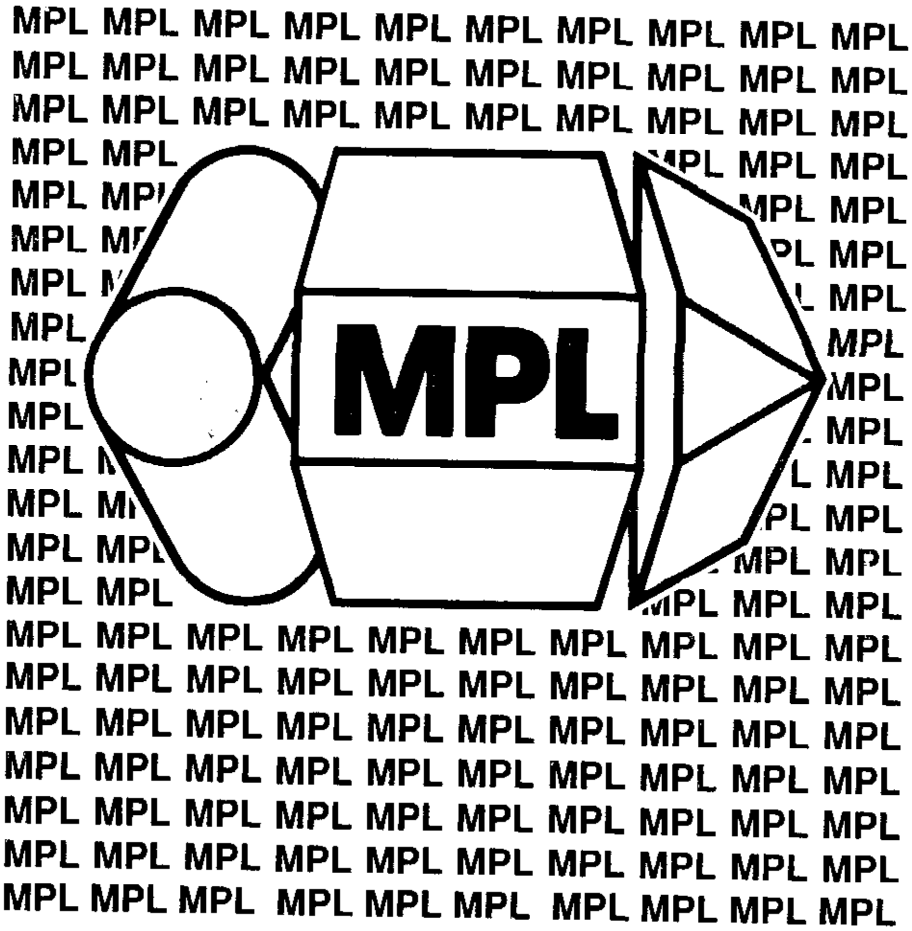
MOTION PROGRAMMING LANGUAGE
MPL Version 3.1



ORMEC

19 Linden Park Rochester, NY 14625 (716) 385-3520

TABLE OF CONTENTS

USING MPL INTERACTIVELY 3
WRITING & EDITING MPL PROGRAMS 3
RUNNING MOTION PROGRAMS 4
COMMAND SYNTAX OVERVIEW 4
Definition of Syntax Variables 6
@ ASSIGN PROGRAM LABEL 8
A ACCELERATION 8
Acceleration (for Overtravel Limits) 9
Acceleration (for Emergency Stop) 9
Acceleration (for Contour E-stop) 9
B BRANCH (GO TO) 9
Conditional Branch on Position 10
C CONTOUR 10
D DELAY 11
Delay (Commanded Motion) 11
Delay (Motion Reference Bus Distance) 11
E EXIT (RETURN) 12
Conditional Exit on Position 12
F FUNCTION (Subroutine) CALL 12
Conditional Function Call on Position 13
G GO 13
H HOME 13
I INDEX 14
J JOG 15
K KILL MOTION 15
L LOOP (Repeat) 16
Nested Program Loops 16
N NORMALIZE 16
NK Checksum on Non-Volatile Memory 17
O OUTPUT 17
P PROGRAM 18
SB SET BAUD 18
SC SHOW CONDITION OF INPUTS 19
SD SET DECELERATION PARAMETERS 19
SL SET OVERTRAVEL LIMITS 20
SM SET or SHOW MODE 20
SP SHOW LAST LABEL PASSED 21
SS SHOW SYSTEM STATUS 21
ST SET PROGRAM TRACE OPTION 21
SW SET WRITE ENABLE 22
SX SET or SHOW X REGISTER 22
SY SET or SHOW Y REGISTER 23
SZ SET or SHOW Z REGISTER 24
T TUNING 26
U UNTIL 26
Conditional Until on Position 27
V VELOCITY 27
= ASSIGN AXIS ID 27
SYSTEM STATUS POLLING 28
Communications Error Checking 28
SELECTING AN AXIS 28
SYNCHRONIZATION TERMINATORS 29
MACHINE I/O INTERFACE 30
PARAMETER RANGES & UNITS 31
ERROR CODES 32

MOTION PROGRAMMING LANGUAGE (MPL)

Version 3.1 of ORMEC's Motion Programming Language (MPL) represents one of the most sophisticated, yet easy to use, motion control programming methods available today. This interpretive, high level language provides automation control systems designers a versatile tool for writing motion control programs. The language is interactive and intuitive to make the process of designing motion control programs for your machinery easy and fast.

This document is designed to give MPL users quick access to command syntax and programming options. For more in-depth information, consult a Programmable Motion Controller (PMC) manual.

USING MPL INTERACTIVELY

MPL is designed to allow the user to set-up and command high performance motion as easily and naturally as you would use a calculator. Once the Programmable Motion Controller (PMC) is powered up, and serial communications is established, the user is presented with a "prompt" of =}. This prompt tells you that the PMC is ready for a command.

Because of the demands of today's *high performance* automation, these commands are designed for maximum speed of execution, with many of them capable of responding in less than a millisecond. In order to maintain this speed of operation, the user defines motion with integer variables which specify the motion in terms of movement of the digital position feedback encoder. Distance is specified in increments of the encoder, velocity is specified as encoder counts per second, and acceleration is specified as a slope defining the rate of rise in encoder frequency per unit time.

To setup and command high performance motion interactively, simply type the commands as documented below when at the =} prompt. But before commanding motion, please note that commanded servomotor motion can be stopped in *several ways:*

1) Hitting the "space bar" <space> at the keyboard.
2) Assertion of the STOP' input at the Machine I/O.
3) Assertion of one of the hardware limits, if enabled.
3) Assertion of one of the software limits, if enabled.
4) Excess position error.

WRITING & EDITING MPL PROGRAMS

MPL commands can be combined in a motion "Program" using the Program command. To write or edit an MPL Program, type P<cr> or P<label><cr> from the keyboard when at the =} prompt.

Editing Functions Used During Program Mode

Cursor Right — TAB (CTRL-I) or CTRL-Y moves the cursor to the right one character. Typing a TAB when the cursor is at the end of a line will move the cursor to the beginning of the next line.

Cursor Left — BACKSPACE (CTRL-H) moves the cursor to the left onc character. Typing a BACKSPACE when the cursor is at the beginning of a line will move the cursor to the end of the previous line.

Cursor Down — LINEFEED (CTRL-J) moves the cursor down a line.

Cursor Up — DEL or CRTL-U moves the cursor up a line. Type a DEL when the cursor is in the middle of a line to move it to the beginning of that line.

End of Line — CTRL-R moves the cursor to the right end of the current line.

Change Line — To change a line in a motion control program, position the cursor at the point to be changed and type the desired information. Underscores (_) may be used to mark program buffer space for future parameter changes or MPL commands.

Kill Line

CTRL-K deletes all characters from the cursor to the end of the line. It can be used to delete an entire line or only unwanted characters at the end of the line. Type CTRL-K with the cursor positioned at the end of a line to delete the "end-of-line" marker and append the next line to current line.

Add Line

Type <cr> at the beginning of a line to insert a "blank" line before it; then type the data for that line. Since "blank" lines are not allowed, immediately typing a second <cr>, or moving the cursor, will eliminate the "blank" line just created. Type <cr> in the middle of a line to "split" the line into two lines.

Exiting

The ESCAPE key is used to exit the program command.

Program Erase

Typing a } in column 1 (immediately after a <cr>) will erase the program buffer, starting at the current location, and exit the program command. **Note**: executing this command will erase all information from the cursor to end of the program buffer.

RUNNING MOTION PROGRAMS

Running a Program

To run an MPL Program, type B<label><cr> from the keyboard when at the =} prompt.

Stopping a Program

To stop an MPL Program, type <escape> at the keyboard. **Note:** Motion underway will continue.

Aborting a Program

To abort an MPL Program and/or Motion, type <space> at the keyboard, or assert the STOP' input at the Machine I/O. This will both stop the program, if one is running, and stop any motion currently underway.

COMMAND SYNTAX OVERVIEW

Acceleration - Set or examine acceleration rate for motion, limits or E-stop.

Standard Motion	A <rate> <cr>
	A <display>
	A [<relative> <rate>] <cr>
End-of-Travel	AL <rate> <cr>
	AL <display>
	AL [<relative> <rate>] <cr>
Stop Contour	AQ <rate> <cr>
	AQ <display>
	AQ [<relative> <rate>] <cr>

Stop Input AS <rate> <cr>

AS <display>

AS [<relative> <rate>] <cr>

Branch - Transfer MPL program execution to a program label.

B <label> [<condition>] <cr>

B <label> [<op> <position> [<direction>]] <cr>

Contour - Create complex contoured motion.

C #<timebase> <distance>#

C #<ref-distance> <distance>#

Delay - Delay time or distance before executing the next command.

Time D [<time>] [<sync>] <cr>

Commanded Motion DM [<distance>] [<sync>] <cr>

Reference Distance DX [<ref-distance>] [<sync>] <cr>

Exit Program - Return from subroutine to statement after function call.

E [<condition>] <cr>

E [<op> <position> [<direction>]] <cr>

Function Call - Call a Subroutine.

F <label> [<condition>] <cr>

F <label> [<op> <position> [<direction>]] <cr>

Go - Move the servomotor to the specified absolute position.

G <position> <direction> <cr>

G <display>

Home - Move to the nearest encoder reference or sensor.

H [<speed>] <direction>

H <display>

H [<relative> <speed>] [<direction>]

Index - Move the specified distance from the current position.

I [<distance>] #<direction># <cr>

I <display>

I [<relative> <distance>] #<direction># <cr>

Jog - Move at the specified jog speed.

J [<speed>] #<direction># <cr>

J <display>

J [<relative> <speed>] #<direction># <cr>

Kill Motion - Kill any system motion unconditionally.

K [<time>] <cr>

Loop - Transfer MPL execution to a program label a specified number of times.

L [<label> <count>] <cr>

L [<label> <loop cnt id> <count>] <cr>

Normalize - Normalize PMC functions.

Absolute Position N [<position>] <direction>

Reset PMC N* <cr>

Reset Checksums NK*<cr>

Display Checksums NK <display>

Output - Set general purpose machine outputs.

O [<sync>] <select>[/<status>]

O <display>

Program - Enter, edit or examine motion programs.		
	P	<program> <text>
	P	<label> <cr> <text>
Set or Show - Specify system and machine parameters.		
Baud Rate	SB	<rate> <cr>
Condition of Inputs	SC	<display>
Set Deceleration Rate	SD	<trigger> [<sync>] <cr>
	SDC<cr>	
	SD	<display>
Software Limits	SL	<name> <position> [<sign>] <cr>
	SL	<display>
Servo System Mode	SM	<mode> <cr>
	SM	<display>
Last Label Passed	SP	<display>
System Status	SS	<display>
Program Trace	ST	<status>
	ST	<display>
Write Enable	SW	<status>
	SW	<display>
Registers	S	<register> <status> <cr>
	S	<display>
Tune Loops - Adjust or examine the current servo loop tuning parameters.		
Adjust Parameter(s)	T	#<register> [<value>] #<sign>##<cr>
Display Parameters	T	<display>
Display Position Error	TE	<display>
Until - Suspend MPL operation until a specified condition is true.		
	U	<condition> <cr>
	U	[<op> <position> [<direction>]] <cr>
Velocity - Set or examine maximum velocity rate.		
	V	<speed> <cr>
	V	<display>
	V	[<relative> <speed>] <cr>
Label - Establish a single-letter program label in the program buffer.		
	@	<label> <text> <cr>
	@	<text><cr>
Assign ID - Label motion axis in non-volatile memory.		
	=	<id> <cr>
	=	<display>

Definition of Syntax Variables

<condition> The specification of discrete machine inputs and status to be tested in the format "<select> [/<status>]". A <cr> is always a go condition.

<count> Number of times for operation to be repeated

<cr> Carriage return ($0D_H$)

<display>
? display last entered value
! display current system value

%[<time>] display status each <time> interval
(valid for G,H,I,J,O,SC,SS,TE,SD & V only)
<sync> permitted in most cases

<distance> Number of relative encoder counts

<direction>
+/- positive/negative <speed>, <position>,<distance>
* stop system motion
(Used on C,G,H,I,J commands)
<sync> permitted in most cases

<hex> Hexadecimal numbers (0-9, A-F)

<id> Motion axis identifier (A-Z)

<label> Displayable character used to identify a motion routine

<loop count id> The character X, Y, or Z specifying the loop counter to be used. Loops can be nested up to three levels.

<mode> Motor Control mode: 0 - idle; 1 - velocity; 2 - position; 3 - position without resetting position error; 4 - master axis controller

<name>
F the forward end-of-travel limit
R the reverse end-of-travel limit
H hardware limit polarity

<op> An operator specifying "greater than" (>) or "less than" (<) a <position> for commands using conditional testing

<position> Absolute motor position in encoder counts

<program> Enter, edit or display MPL program buffer:
{ Initiate programming at beginning of program buffer.
<cr> Initiate programming at end of program buffer.
? Display program buffer from the beginning.
! Display entire program buffer.

<rate> Acceleration rate in kHz/sec; 100 Hz/sec; or 100's counts

<ref-distance> The <ref-distance> parameter specifies the length of a position/ position segment in the Contour command.

<register> P=(position gain), V=(velocity gain), F=(feedforward gain), X=(external output gain), CP=(position loop compensation), CV=(velocity loop compensation)
T=Program Trace W=Program Buffer Write Enable
Status: X, Y, Z

<relative>
P Increase the magnitude of the speed, distance, or rate by the value following the <relative> token
M Decrease the magnitude of the speed, distance, or rate by the value following the <relative> token

<select> A hexadecimal number that specifies which machine inputs or outputs are to be used or changed and which are to be ignored.

<sign>
+/- Add/subtract <value> to/from <register>.
<cr> Set <register> to <value>.

<speed> Speed in increments of 10 cts/sec, 100 cts/sec or .01% of the motion reference bus speed

<status> a two-digit Hexadecimal number (0-9, A-F), specifying register values.

<sync>	Synchronization character for coordinating MPL program operation with commanded motion: , Wait until commanded motion is stopped. ; Wait until motion is constant or stopped. : Wait until motion is accelerating or decelerating.
<text>	Motion routines, comments or editing command characters
<time>	Time in milliseconds
<timebase>	The <timebase> parameter specifies the length of a position/time segment in the Contour command.
<value>	Value substituted for, added to or subtracted from <register>

< > - variable item # # - repeatable item [ ] - optional item

@ ASSIGN PROGRAM LABEL

Establish a single-letter program label in the program buffer.

@<label><cr>	Establish a single-letter program name <label> in the program buffer for future reference. The "@" routine (signified by @@), if present, will automatically execute on powerup or software reset of the PMC.
@<label><text><cr>	After the single-letter program <label>, additional <text> may be added as comments to the MPL routine.

Examples:

@Motion<cr>	Creates starting point for program "M" which causes motion. The rest of the word "Motion" is a comment.
@@_Powerup<cr>	Identifies the start point of the program which will automatically execute on powerup or software reset.

A ACCELERATION

Set or examine acceleration rate for motion, limits, or E-stop.

A<rate><cr>	Set acceleration/deceleration rate in motion buffer.
A?	Display acceleration rate currently in motion buffer.
A!	Display currently commanded system acceleration rate (zero *if at rest* or *top speed*).
A<relative><rate><cr>	Add or subtract value 'rate' from the acceleration rate. Add to acceleration rate by specifying <relative> variable as 'P' for plus, and subtract by specifying 'M' for minus.

Examples:

A3500<cr>	Set acceleration rate to 3500 counts/sec/msec, or Hz/msec
AP50<cr>	Add 50 Hz/msec to the current acceleration rate.

Acceleration (for Overtravel Limits)

AL<rate><cr>	Set deceleration rate to be used for deceleration when an overtravel limit is encountered.
AL?	Display limit deceleration rate currently set.
AL<relative><rate><cr>	Add or subtract value ‘rate’ from the limit deceleration rate. Add by specifying <relative> as ‘P’ for plus, and subtract by specifying ‘M’ for minus.
Example:	
AL6000<cr>	Set deceleration rate used for overtravel limit stops to 6000 Hz/msec.

Acceleration (for Emergency Stop)

AS<rate><cr>	Set deceleration rate to be used for emergency stops.
AS?	Display currently set emergency stop deceleration rate
AS<relative><rate><cr>	Add or subtract ‘rate’ from the emergency stop deceleration rate. Add by specifying <relative> as ‘P’ for plus, and subtract by specifying ‘M’ for minus.
Example:	
AS400<cr>	Set deceleration rate for the stop input to 400 Hz/msec.

Acceleration (for Contour Command E-stop)

AQ<rate><cr>	Set deceleration rate used for emergency or fault stop during Contour command. Units are counts/second per msec in internal mode, and 100’s of counts of the external motion reference bus in external mode.
AQ?	Display currently set emergency or fault stop deceleration for use during Contour Command.
AQ<relative><rate><cr>	Add or subtract ‘rate’ from the contour command E-stop deceleration rate. Add by specifying <relative> as ‘P’ for plus, and subtract by specifying ‘M’ for minus.

B BRANCH

Transfer MPL program execution to a program label.

B<label><cr>	Unconditionally transfer MPL program execution to a program <label> with no return.
B<label><condition>	Transfer MPL program execution to a program <label> if the machine input <condition> is met. *See the Machine I/O Interface Section.*
Examples:	
BQ<cr>	Unconditionally branch to program label ‘Q’
BQ20<cr>	Branch to program label ‘Q’ if input IN20’ is “on”
BQ1/0<cr>	Branch to program label ‘Q’ if input IN1’ is “off”

BQ3/2<cr>

Branch to program label 'Q' only if input IN2' is "on" and IN1' is "off".

Conditional Branch on Position

B<label><op><position>[<direction>]<cr>

Transfer MPL program execution to a program <label> with no return based on a specified test of absolute position. The operator <op> specifies "greater than" (>) or "less than" (<) <position> as the condition to be tested against. Position is an integer which specifies absolute position in counts. The variable <direction> makes position a positive or negative integer.

Examples:

BL>200+<cr>

Branch to program label 'L' if the PMC-based system is at an absolute position greater than 200 counts in the positive direction. (i.e. 201, 202 ...)

BL<100-<cr>

Branch to program label 'L' if the PMC-based system is at an absolute position less than 100 counts in the negative direction. (i.e. -101, -102, -103 ...)

C CONTOUR

Create high performance profiled motion

C<timebase><distance>

The Contour command allows specification of a general motion-time profile in <timebase> segments over a range from 3.94 to 336.000 msec. These linear "position vs. time" segments may be commanded in real time by a host computer through the serial communications interface or "Programmed" in the program buffer for execution by the PMC.

C<ref-distance><distance>

The Contour command also allows specification of a general motion-motion profile in <ref-distance> increments of the motion reference bus over a range from 252 to 16,128 motion reference pulses. These linear "position vs. position" segments may be commanded in real time by a host computer through the serial communications interface or "Programmed" in the MPL program buffer for later execution by the PMC. This capability allows "multi-axis contouring" capability which can be referenced to a common "master axis controller" or to an external source of motion information.

Note: This command is designed to create high performance profiled motion which is defined and coded by a host computer. The coded Contour profile is then either downloaded into the program buffer or sent to the PMC via the serial communications interface in real time. It is not practical to define Contour data manually. Consult the *PMC Installation and Operation Manual* for detailed data definitions for this command.

D DELAY

Delay a specified time interval or number of counts before executing next command.

Delay (Time)

D[T]<time><sync><cr>

Delay the specified time before executing the next command. The resolution of the PMC's internal timer is 4 msec and, due to the asynchronous nature of the delay command, there is an uncertainty of 4 msec. Therefore since time is "rounded up," a D1 command will delay 4 to 8 msec. The "T" is optional, and present only to be consistent with the "DM" and "DX" commands.

Example:

D16<cr>

Delay for 16 msec.

See "Synchronization Terminators" for how this command can be used to synchronize MPL program operation with the motion being created.

Delay (Commanded Motion)

DM <distance> [<sync>] <cr>

Delay a specified distance (commanded motion in counts) before executing the next command. **Note**: The system must be at steady state speed to use this command.

Example:

DM1000<cr>

Delay 1000 counts of commanded motion before executing next command.

See "Synchronization Terminators" for how this command can be used to synchronize MPL Program execution with the motion being created.

Delay (Motion Reference Bus Distance)

DX <distance> [<sync>] <cr>

Delay a specified number of reference clock counts before executing the next command. **Note**: The system must be at steady state speed to use this command.

Example:

DX5000<cr>

Delay for 5000 counts of the reference clock. If in internal 192kHz mode, this command would delay 5000 counts of the 192kHz clock. If in external mode, this command would delay 5000 counts of the motion reference bus.

See "Synchronization Terminators" for how this command can be used to synchronize MPL program execution with the motion being created.

E EXIT

Return from subroutine to statement after function call.

E<cr> Unconditionally exit (return from) an MPL (sub)routine.

E<condition> Exit an MPL routine if machine input <condition> is true. *See the Machine I/O Interface Section.*

Examples:

E<cr> Unconditional exit (return from subroutine).

E8<cr> Exit (return) if Machine Input IN8' is "on"

E3/1<cr> Exit (return) if Machine Input only if both IN1' is "on" and IN2' is off.

Conditional Exit on Position

E<op>[<position>[<direction>]]<cr>

Return from subroutine to statement after function call based on a specified test of absolute position. The operator <op> specifies "greater than" (>) or "less than" (<) <position> as the condition to be tested against. Position is an integer which specifies absolute position in counts. The variable <direction> makes position a positive or negative integer.

Example:

E>2000+<cr> Exit the subroutine if the PMC-based system is at an absolute position greater than 2000 counts in the positive direction.

F FUNCTION (Subroutine) CALL

Call a subroutine. Functions can be nested three deep.

F<label><cr> Unconditionally transfer MPL program execution to a program <label>. When an Exit command is executed, MPL operation resumes at the line following the "F-command".

F<label><condition> Transfer MPL program execution to a program <label> if the machine input <condition> is met. See the Machine I/O Interface Section below. When an Exit command is executed, MPL operation resumes at the next program line after the "F-command".

See the Machine I/O Interface Section.

Examples:

FE8<cr> Call subroutine "E" if input IN8' is "on".

FE3/0<cr> Call subroutine "E" if inputs IN2' and IN1' are "off".

Conditional Function Call on Position

F<label>[<op><position>[<direction>]]<cr>

Transfer MPL program execution to a program <label> based on a specified test of absolute position. The operator <op> specifies "greater than" (>) or "less than" (<) <position> as the condition to be tested against. Position is an integer which specifies absolute position in counts. The variable <direction> makes position a positive or negative integer.

Example:

FA>200+<cr> Call subroutine "A" if the PMC-based system is at an absolute position greater than 200 counts in the positive direction.

G GO

Move to the specified absolute position of the system.

G<position>+<cr> Move to the absolute position of the system that is specified. The sign (+ or -) of the position follows the numerical specification.

G+<cr> Go to the absolute zero position of the system.

G200-<cr> Move to absolute position -200.

G! Display the current absolute position of the system.

G%<cr> Repeatedly display the current absolute position of the system.

G%<time><cr> Repeatedly display the current absolute position of the system at the repeat rate specified by <time> in msec.

G? Display the currently commanded absolute position.

G* Stop system motion.

See the Synchronization Terminators Section for how this command can be used to synchronize MPL program execution with motion being created.

H HOME

Move to the nearest encoder reference or sensor.

H<speed><cr> Set homing speed in motion buffer.

H? Display homing speed currently in motion buffer.

H- Move at the previously specified homing speed in a negative direction until an encoder reference or sensor is reached.

H<speed>+ Set the homing speed in the motion buffer, and home in a positive direction.

H! Display current system speed.

H%<cr> Repeatedly display current system speed.

H* Stop system motion.

H[<rel><speed>]<cr> Add or subtract <speed> from the homing speed. Add to the home speed by specifying <relative> as "P" for plus, or reduce it by entering "M" for minus.

Examples:

H15+ Move to the homing position in the positive direction with a velocity of 1.5 kHz.

HM4<cr> Decrease the magnitude of the homing speed by 0.4 kHz.

See the Synchronization Terminators Section for how this command can be used to synchronize MPL program execution with motion being created.

I INDEX

Move the specified distance from the current position.

I<distance><cr> Set the relative index distance in motion buffer.

I? Display index distance currently in motion buffer.

I+<cr> Move in a positive direction the previously specified index distance. Entering the [+] initiates the motion; the <cr> terminates the **I** command.

I<distance>-<cr> Set the index distance in the motion buffer, and move in a negative direction the specified number of counts.

I! Display the number of counts remaining in the current move.

I%<cr> Repeatedly display the counts remaining in current move.

I* Stop system motion.

I[<rel><dist>]<cr> Add or subtract <dist> from the index distance. Add to the index distance by specifying <rel> as "P" for plus, or subtract by entering "M" for minus.

Examples:

I250<cr> Set the index distance to 250 counts.

I+ Index the system the previously set distance in the positive direction.

IP300<cr> Increase the magnitude of the index distance in the motion buffer by 300 counts.

I%200<cr> Display the number of remaining counts in the current move. Update the display every 200 msec.

See the Synchronization Terminators Section for how this command can be used to synchronize MPL program execution with motion being created.

J JOG

Move at the specified jog speed.

J<speed><cr>	Set jog speed in motion buffer.
J?	Display jog rate currently in motion buffer.
J+<cr>	Jog in a positive direction at the jog speed in the motion buffer.
J<speed>+<cr>	Set jog speed in motion buffer, and jog in a positive direction.
J!	Display current system speed.
J%<cr>	Repeatedly display current system speed.
J*	Stop system motion.
J[<rel><speed>]<cr>	Add or subtract <speed> from the jog speed. Add by specifying <rel> as "P" for plus, or subtract by entering "M" for minus.

Examples:

J36+	Jog in the positive direction at 3.6 kHz.
J+	Jog in the positive direction at previously specified jog speed.
JM100<cr>	Decrease the magnitude of the system jog rate by 100kHz.
JP10-	Increase the magnitude of the system jog rate by 10kHz and start motion in the negative direction.

See the Synchronization Terminators Section for how this command can be used to synchronize MPL program execution with motion being created.

K KILL MOTION

Kill any system motion unconditionally.

K<cr>	Kill system motion immediately.
K<time><cr>	Kill system motion specifying the time in msec allowed for system deceleration at the AL rate. If system motion has not stopped after the time has expired, an immediate stop will be commanded. This command is particularly convenient when working interactively with a system referencing its motion to the motion reference bus, and the motion reference is stopped.

Examples:

K1000<cr>	Kill any system motion after one second is allowed for deceleration.
K<cr>	Kill system motion immediately.

L LOOP (Return)

Transfer MPL execution to a program label a number of times.

L<label><count><cr> Transfer MPL program execution to a program <label> a specified number of times, and then continue program execution with the next command in the program buffer.

L<cr> Clear the loop counter.

Nested Program Loops

Utilize up to three additional levels of nested program loops. The <loop counter ID> variables X,Y and Z specify the additional loop counter to be used. **Note:** To nest Program loops, the counter IDs must be used.

Examples:

@A	Establish program label A
G0+	Go to absolute position 0
U1	Wait Until machine input IN1' is asserted
@B	Establish program label B
I100,+	Index 100 counts in the positive direction
D50,	Delay 50 msec after index is complete
LBX5	Loop back to label B five times
LA2	Loop back to label A twice

Note: This command can only be used in an MPL program and not interactively.

N NORMALIZE

Normalize PMC functions.

N<cr> Start SCI automatic baud rate selection. Additional <cr>'s will cause the PMC to automatically determine the baud rate. It will send a prompt when the rate is determined.

N* Reset PMC software.

N<position><sign> Define the current physical position. This command resets the internal absolute position counter to the position specified. The <sign> (+ or -) follows the numerical specification and terminates the command.

Example:

N2000+	Set absolute position counter to +2000.

NK - Checksum on Non-Volatile Memory

The **NK** command is a special command which calculates, displays and verifies checksums on the PMC's non-volatile memory. If the verify fails, the PMC generates an error flash code with its diagnostic LED, and returns an error prompt of "-}". The NK command is automatically executed at powerup.

NK?	Displays checksums stored in memory.
NK!	Calculates and displays checksums. Displays what PMC thinks checksums should be.
NK<cr>	Perform automatic non-volatile memory verify.
NK*	Recalculates and updates checksums. Errors are cleared; LED and prompt return to normal.

O OUTPUT

Set general purpose machine outputs.

O[<sync>]<select>[/<status>]<cr>

Set the four general purpose machine outputs. The hexadecimal <select> parameter specifies which outputs will be affected by the command. The <status> parameter specifies whether the selected outputs will be turned "on" or "off". This allows a single command to set any output pattern, as well as leave unselected outputs unchanged.

Bits in the <select> parameter that are set to 1 allow the corresponding machine output to be changed. Conversely, select bits set to 0 cause the corresponding machine outputs to be unaffected by the output command.

If no <status> parameter is specified, all the selected outputs will be turned "on". Specifying a <status> parameter of "0" will cause all the selected outputs to be turned "off". Multiple outputs can be turned "on" and "off" in a single command by specifying "1's" and "0's" in the corresponding bit of the <status> parameter.

Examples:

O3<cr>	Turn "on" outputs OUT1' and OUT2' (TTL low). The other outputs are left unchanged.
O,3/0<cr>	Delay until motion is stopped, then turn "off" outputs OUT1' and OUT2' (TTL high). The other outputs are left unchanged.
OC/0<cr>	Turn "off" outputs OUT4' and OUT8' (TTL high), leaving outputs OUT1' and OUT2' unchanged.
O3/2<cr>	Turn "on" output OUT2' and "off" output OUT1', leaving outputs OUT4' and OUT8' unchanged. *See the Machine I/O Interface Section.*
O?	Display the state of the general purpose machine outputs.
O!	Same as O?

P PROGRAM

Enter, edit or examine motion programs.

P{	Enter program mode with the cursor at the beginning of the program buffer.
P<label><cr>	Enter program mode with the cursor at a program <label>.
P<cr>	Enter program mode with cursor at the end of the program buffer (for adding routines to memory).
P?	Display program (for viewing only, no editing) one line at a time from the beginning of the program buffer.
P!	Displays entire program buffer. The ESCAPE character will abort this command.

Host Programming Command

The host (binary) programming command is designed for compact programming of a PMC from a host computer interface. To use it you must first enable binary communications mode by setting Bit 7 of the Z Register to 1 and Bit 6 of the Z Register to 0. See the SZ command.

P <label>	A program label is required to enter binary programming, and when it is found, the PMC will output an "@" character. A **P{** command can also be utilized, and programming will start at the beginning of the program buffer.

Binary Programming Buffer Commands

<lf>	Move to the first character of the next line. No output.
<tab>	Echo current character and move to the next character.
<esc>	Terminate programming mode.

Note: any other character overwrites the program buffer.

SB SET BAUD RATE

Select baud rate. Specify the PMC to either autobaud or save and use a specific baud rate parameter in non-volatile memory.

SB <baud> <cr> Specify system baud rate according to table below.

0	Specifies PMC autobaud sequence.
1	38,400
2	19,200
3	9600
4	4800
5	2400
6	1200
7	600
8	300

SB? Display the last entered baud specification (to take effect at the next cycle of power or hard reset).

SB! Display the current <baud> value.

Note: For operator convenience, the new baud rate selected by the set baud command does not take effect until the next time the PMC is reset.

SC SHOW CONDITION OF INPUTS

Display the current system inputs.

SC! Display the condition of the machine inputs using a two digit hexadecimal number.

SC? Same as SC!

SC% Continuously display the condition of the machine inputs.

SD SET DECELERATION RATE

Set up a trigger to begin a deceleration from top velocity; calculate a deceleration rate

SD <trigger> [<sync>] <cr> Select a trigger to be used to begin the deceleration from top velocity. The trigger may be one of the following values:

D - remaining Distance (Go and Index motions only)
E - Encoder Reference
I - Immediate trigger
O - Turn OFF trigger
S - Sensor Input

The synchronization character can be used to coordinate this command with completion of a motion, motion reaching a constant speed or reaching the end of a constant speed.

SDC <cr> Calculate the deceleration rate to be used for the next system deceleration triggered by the **SD** command. If this command is not used, the deceleration rate will be equal to the last acceleration rate. **Note:** when the **SDC** command is issued, the current value of the 'A' command will be used as the deceleration rate.

Examples:

SDS;<cr> Wait for top velocity, then set the deceleration trigger to be the sensor input

SDI;<cr> Wait for system to accelerate to top velocity, then decelerate immediately

SD? Display the deceleration trigger set by the **SD** command. This value is reset to 'O' when the system comes to rest.

SDC Calculate the new deceleration rate for the **SD** trigger.

SDD; Set up a new deceleration rate and trigger it based on the decel point in the current Go or index motion.

Note: (1) If an index or go motion has its deceleration trigger and/or rate improperly changed by the **SD** command, the motion may not end as expected. (2) The **SD** trigger command can only be used at top velocity. (3) The **SDC** command can be issued at any time and will use the value set by the **A** command to calculate the deceleration rate.

SL SET OVERTRAVEL LIMITS

Set software absolute position limits or hardware limit polarity.

SL <name> <position> [<sign>] <cr>

Set limit parameters. The variable <name> specifies which limit is to be set. "F" sets the forward travel limit; "R" sets the reverse travel limit; and "H" sets the polarity of the hardware overtravel limits. <position> is an integer which specifies the absolute position of the limit, either positive "+" or negative "-" by specifying <sign>.

SL? Display the limit parameter settings.

Examples:

SLF100000<cr>	Set the forward limit to position +100000.
SLR50000-<cr>	Set the reverse limit to position -50000.
SLR<cr>	Set the reverse limit to position 0.
SLH0	Specify that the hardware overtravel limits are asserted for TTL low level input signals.
SLH1	Specify that the hardware overtravel limits are asserted for TTL high level input signals.

Note: To cause the PMC to use the software limits, Bit 1 of the Z Register must be set using the SZ command. To use hardware limits, Bit 1 of the X status register must be set by using the SX command.

SM SET or SHOW MODE

Select the servo control mode. (fully supported only while motion is stopped)

SM<mode><cr> Set servo control mode.

<0> IDLE (servodrive off and servo loops disabled)
<1> VELOCITY control mode (servo using analog tach only)
<2> POSITION control mode (servo using tach and encoder)
<3> POSITION mode without clearing position error
<4> MASTER Axis Controller mode

SM! Display the current servo control mode.
0 - IDLE; 1 - VELOCITY; 2 - POSITION; 4 - MASTER

SM? Display last entered mode.

SP SHOW LAST LABEL PASSED

Display the last program label passed.

SP! Displays the last label passed, followed by a number indicating the number of commands executed since the label was passed.

SP? Same as SP!

SS SHOW SYSTEM STATUS

Display the motion profile register which provides information on system status.

SS? Displays the motion profile register as two hex digits as defined below.

SS! Same as SS?

Bit 7 "Drive On" set to 1 indicates the servodrive is enabled.

Bit 6 "Position Summing Junction Overflow" (1 => overflow)

Bit 5 Reserved

Bit 4 Reserved

Bit 3 Reserved

Bit 2 "Direction" set to 1 indicates that the last (or current) motion was (or is) Forward.

Bit 1 "Top velocity" set to one indicates whether that servo-system is currently at top velocity.

Bit 0 "Motion" set to 1 indicates that the system is in motion.

ST SET PROGRAM TRACE OPTION

Enable or disable the program buffer trace option.

ST<hex> Set program trace "on" or "off". Entering "ST0" disables program buffer trace, while "ST1" displays each MPL command at the Serial Communications Interface (SCI) as it is executed.

ST! Displays the current state of the program buffer trace option.

ST? Same as ST!

SW SET WRITE ENABLE

Enable or disable write protection on the program buffer contents.

SW<hex>

Set the program buffer write enable "on" or "off". Entering "SW0" does not allow changes to the program buffer. Entering "SW1" allows the program buffer to be modified. Even with the program buffer write protected, you may view its contents via any of the **P** commands, however any attempt to change the contents will result in a C0 error. Powerup default is SW0.

SW!

Displays the current state of the program buffer write enable option.

SW?

Same as SW!

Example:

SW0

Write protect the program buffer.

SX SET or SHOW X REGISTER

Select system parameters by specifying the condition of the X Status Register

SX<hex><cr>

Set or examine the X status register to change the configuration of the motion control system. <hex> is the hexadecimal representation of the selected byte.

S?

Display status registers in the order X,Y,Z. The data is displayed with labels. e.g. X=08 Y=00 Z=00

S!

Display status registers in the order X,Y,Z. The data is displayed with six consecutive ASCII hex characters. e.g. 080000

X Register

Bit 7

Setting SMOOTH ACCELERATION PROFILE to a 1 provides extremely smooth acceleration for all PMC motion commands. Note that only the linear acceleration profile is supported while using this mode.

Bit 6

Setting MOTION BUS SLAVE to a 1 selects the Motion Reference Bus as the master reference for creating motion instead of the internal crystal controlled clock;

Bit 5

Setting ALTERNATE REFERENCE ENABLE to a 1 causes each odd motion reference pulse to be sent directly to the Position Summing Junction and each even motion reference pulse to be used as the internal distance reference; This output is useful with the MOTION BUS SLAVE bit for establishing a nominal motor speed of 50% with respect to other moving machinery.

Bit 4

Setting MOTION BUS MASTER to a 1 causes the PMC to supply its motion reference pulses to the Motion Reference Bus for reference by other PMCs.

Bits 3-2 — VELOCITY RANGE SELECT selects velocity range:

RANGE	Bit 3	Bit 2
48k Hz	0	0
256K Hz	0	1
192k Hz	1	0
384k Hz	1	1

Bit 1 — Setting ENABLE HARDWARE OVERTRAVEL LIMITS to a 1 causes Machine I/O inputs IN4' and IN8' to be used as - and + overtravel limit switch inputs respectively. When one of these limits is encountered, the **AL** deceleration value is used to stop motion.

Bit 0 — Setting DIRECTION INVERT to a 1 transposes the meaning of + and - in motion commands.

Examples:

SX88<cr>	Select smooth acceleration mode in 192 kHz range.
SX08<cr>	Terminate smooth acceleration mode in 192 kHz range.
SX18<cr>	Become a "motion bus master" in the 192 kHz range.

SY SET or SHOW Y REGISTER

Select system parameters by specifying the condition of the Y Status Register

SY<hex><cr> — Set or examine the Y status register to change the configuration of the motion control system. <hex> is the hexadecimal representation of the selected byte.

S? — Display status registers in the order X,Y,Z. The data is displayed with labels. e.g. X=08 Y=00 Z=00

S! — Display status registers in the order X,Y,Z. The data is displayed with six consecutive ASCII hex characters. e.g. 080000

Y Register

Bit 7-6 — Bits 6 & 7 of the Y Register are used to select one of the available Acceleration Profiles as follow:

Bit 7	Bit 6	Acceleration Profile
0	0	linear
0	1	s-curve (polynomial)
1	0	parabolic
1	1	reserved

Bit 5 — Setting EXTERNAL START to a 1 causes a commanded motion to not start until the PMC receives an external signal. Bit 4 indicates signal used to initiate motion.

Bit 4 — EXTERNAL START SELECT specifies either the machine sensor input signal (SENSIN) or the encoder reference (ENCR) to start a motion. (1=machine sensor, 0=encoder reference) Bit 4 will be ignored unless Bit 5 (EXTERNAL START) is set.

Bit 3 — Setting EXTERNAL DECEL to a 1 causes deceleration to occur when the machine sensor input (SENSIN) is received, instead of a calculated distance after full speed is attained. Ordinarily, deceleration is initiated when the remaining distance is equal to the acceleration distance.

Bit 2

EXTERNAL STOP SELECT specifies either the machine sensor input signal (SENSIN) or the encoder reference (ENCR) to stop motion during a home command or an INDEX EXTEND. (1=machine sensor, 0=encoder reference)

Bit 1

Setting INDEX EXTEND to a 1 specifies that during deceleration, the speed should remain at the level set by the **J** command rather than continuing to zero. INDEX EXTEND is used in conjunction with EXTERNAL STOP SELECT or with a machine input condition to stop the motion.

Bit 0

Setting SHARP JOG STOP to a 1 causes the PMC to stop immediately at the end of a jog command rather than the deceleration rate specified by the **A** command.

Note: The Y Register is in the motion buffer and therefore, altering it during a motion will only affect the next commanded motion. ***See the PMC manual for diagrams showing some of the possible motion profiles created by using the Y Register.***

SZ SET or SHOW Z REGISTER

Select system parameters by specifying the condition of the Z Status Register

SZ<hex><cr>

Set or examine the Z status register to change the configuration of the motion control system. <hex> is the hexadecimal representation of the selected byte.

S?

Display status registers in the order X,Y,Z. The data is displayed with labels. e.g. X=08 Y=00 Z=00

S!

Display status registers in the order X,Y,Z. The data is displayed with six consecutive ASCII hex characters. e.g. 080000

Z Register

Bit 7-6

Bits 6 & 7 of the X Register are used to select one of the available Communications Modes as follow:

Bit 7	Bit 6	Communications Mode
0	0	Decimal In - Decimal Out
0	1	Hexadecimal In - Hexadecimal Out
1	0	Binary In - Binary Out
1	1	Hexadecimal In - Binary Out

The syntax of MPL commands which may have their parameters changed in binary input mode is listed below:

A <null><byte><byte>
AL <null><byte><byte>
AS <null><byte><byte>
AQ <null><byte><byte>
D <null><byte><byte><cr>
DM <null><byte><byte><byte><byte><cr>

DR <null><byte><byte><byte><byte><cr>
G <null><byte><byte><byte><byte><cr>
H <null><byte><byte><term>
I <null><byte><byte><byte><byte><term>
J <null><byte><byte><term>
K <null><byte><byte><cr>
N <null><byte><byte><byte><byte>
S <null><byte><byte><byte>
SL<null><byte><byte><byte><byte><byte><byte><byte><byte>
T <null><byte><byte><byte><byte><byte><byte>
V <null><byte><byte>

In the descriptions above:

<null> refers to the ASCII null character (00_H).

<cr> refers to the ASCII carriage return ($0D_H$).

<byte> refers to one byte of binary data, and for the specification of numbers larger than one byte, the numbers are normally represented in unsigned binary format with the most significant byte first. An exception to this is the G command, in which the numbers are expressed in two's complement binary notation, with the most significant byte first.

<term> represents any of the motion terminators (+,-,*).

In the SL command, the first four bytes represent the value of the upper overtravel limit, and the second four bytes represent the value of the lower overtravel limit.

In the T command, the six bytes represent the six gain and compensation adjustments in the order P,V,F,X,CP, & CV.

Bit 5 Setting BAUD RATE & AXIS ID WRITE ENABLE to 1 allows the user to set a baud rate with the "SB command", or to set an axis ID in non-volatile memory with the "= command".

Bit 4 Setting the MACHINE I/O MOTION ROUTINE ADDRESS SELECT to 1 changes the motion routine address select from 5 bits to 1 bit, allowing either the "0" or "1" routine to be selected from hardware using the SEL input, instead of 32 routines, which requires the use of four general purpose machine inputs for address lines.

Bit 3 Setting IN MOTION ENABLE to 1 causes Machine Output OUT8' to be automatically turned "on" whenever the system is in motion and "off" when the system is stopped.

Bit 2 Setting FAULT ENABLE to 1 causes Machine Output OUT4 to be automatically turned "on" whenever there is a fault, and "off" when the first character of the next command is received.

Bit 1 Setting SOFTWARE LIMITS ENABLE to 1 causes the software limits, set by the "SL command" to limit system motion.

Bit 0 Setting NO ECHO to 1 prevents echo of SCI characters, causing the PMC's serial communications to operate in the half-duplex mode.

T TUNING

Tune the servo loops or examine the current tuning parameters.

TP<x><cr> Set position loop gain to <x>.

TV<x><cr> Set velocity loop gain to <x>.

TF<x><cr> Set feedforward gain in <x>.

TX<x><cr> Set external output gain to <x>.

TCP<x><cr> Set position loop compensation to <x>.

TCV<x><cr> Set velocity loop compensation to <x>.

Note: <x> indicates a value to use in each of the operations specified above. <x> is entered as a decimal for all gain values (ASCII hex if HEX COM is set). The compensation values are always entered as hex arguments.

T<reg><x>+<cr> Add value <x> to the specified tuning register <reg>. e.g. **P++V5—** increments position gain by 2 and decrements velocity gain by 10. Note: <x> defaults to 1 if it is not specified.

T? Display current values for each of the tuning parameters. Values are labelled (e.g. P=02 V=02 F=00 X=00 CP=04 CV=00). Gain values are output in decimal ASCII unless HEXCOM is set. The compensation values are always output in hex ASCII.

T! Display current values for each of the tuning parameters. All values are output in hex ASCII. e.g. 020200000400

TE? Display the normalization error (error cancelled when the last Normalize command was executed).

TE! Display current error relative to normalization point. This is equal to the digital error in the counters minus the normalization error.

TE%<cr> Repeatedly display the current error.

U UNTIL

Suspend MPL operation until a specified condition is true.

U<condition> Wait until the specified machine input <condition> is true before executing the next command. *See the Machine I/O Interface Section for more information.*

Examples:

U1<cr> Wait until input IN1' is asserted (TTL low) before executing the next command.

U2/0<cr> Wait until input IN2' is not asserted (TTL high) before executing the next command.

U81<cr> Wait until inputs IN80' and IN1' are asserted (TTL low) before executing the next command. *See the Machine I/O Interface Section.*

Conditional Until on Position

U<op>[<position>[<direction>]]<cr>

Wait before executing the next command based on a specified test of absolute position. The operator <op> specifies "greater than" (>) or "less than" (<) <position> as the condition to be tested against. Position is an integer which specifies absolute position in counts. The variable <direction> makes position a positive or negative integer.

Example:

U>2000+<cr>	Wait until the PMC-based system moves to an absolute position greater than 2000 counts in the positive direction before executing the next command in the program.

V VELOCITY

Set or examine maximum velocity rate.

V<speed><cr>	Set index speed in motion buffer.
V?	Display index speed currently in motion buffer.
V!	Display currently commanded system speed.
V%	Repeatedly display current system speed.
V[<rel><speed>]<cr>	Add or subtract <speed> from the velocity parameter. Add by specifying <rel> as "P" for plus, or subtract by entering "M" for minus.

Examples:

V304<cr>	Set velocity to 30.4 kHz for next motion.
VP100<cr>	Increase the magnitude of the velocity by 10 kHz for the next index.
V%1000<cr>	Display the current system speed and update the display every 1000 msec.

= ASSIGN AXIS ID

Label motion axis in non-volatile RAM.

=<id><cr>	Assign axis identifier. An <id> can be specified as any upper case alphabetic character.
=!	Display the ID of the axis currently in use.
=?	Display the ID currently stored in program memory.
==<cr>	Turn off serial bus support.

Example:

=N<cr>	Assigns axis ID "N" to a PMC.

Notes: (1) For up to 14 axis systems, it is recommended that the axis ID be assigned using jumpers on the PMC. *See the PMC Installation and Operation manual.* (2) Bit 5 of the Z Status Register must be set to use this command.

SYSTEM STATUS POLLING

System status polling is provided to allow access to important information during program execution. This information is acquired by sending a two character sequence to the PMC with which communications are established. These two character sequences perform the equivalent function that the indicated standard MPL command would if issued at the command level. The advantage of the system status polling commands is that the PMC does not have to be at command level for these commands to be used.

Syntax: <attn> <sys poll>

<attn>	Ctrl] (ASCII $1D_H$)		(Equivalent MPL command)
<sys poll>	c	system inputs	(SC!)
	e	current following error	(TE!)
	g	current system position	(G!)
	h	hex communications checksum	none
	i	distance remaining in motion	(I!)
	l	last error code	none
	o	state of outputs & ready line	none
	p	last label passed	(SP!)
	r	X, Y, Z status registers	(S!)
	s	motion paramcters	(SS!)
	v	current system velocity	(V!)
	x	current axis id	(=!)

Note: The <sys poll> command must be lower case.

Communications Error Checking

The PMC maintains a 16 bit checksum of all its received serial communications, excluding the two byte system status poll sequences. Serial communications integrity may be checked by a host computer by maintaining its own 16 bit checksum of communications characters sent, and then periodically issuing an "<attn>h" system status poll. This status poll will return either a four byte hexadecimal number or a two byte binary number, depending on communications mode selected. The host computer then compares this data with its own internal data to determine whether a communications error has been made. The PMC also resets its checksum whenever the host computer issues this system status poll command.

SELECTING AN AXIS

When multi-serial communications is implemented, a PMC will not communicate with the Serial Communications Interface until it recognizes its Axis ID being selected at that interface. This means that once the multi-serial communications is enabled, the user must send a two character "select axis" sequence before the PMC will communicate. Unless the PMCs have executed Set Baud commands, a host computer or terminal must first send an "autobaud sequence" to enable all the PMCs on the serial communications bus to determine the host baud rate. The host computer or terminal can then send the "select axis" sequence as shown below to communicate with a specific PMC.

Syntax: <CTRL]> <id>

<CTRL]> is ASCII ($1D_H$)

<id> is a single upper case ASCII character used as the Axis ID of the unit selected (normally starting with the letter A)

Executing the above command selects the specified PMC for communications with the host. Its communications driver ICs are enabled, and it will send and receive data characters with the host computer and properly operate the flow control line. The normal "prompt" of =} will be changed to **<id>**}, displaying the Axis ID of the selected PMC. e.g. **A**} for "PMC-A"

SYNCHRONIZATION TERMINATORS

The fact that MPL operates independently of the motion it creates is a powerful feature in that it allows machine I/O to be manipulated or successive motions to be set up while motion is taking place. Synchronization characters are provided to allow MPL language execution to synchronize with the motion which is underway.

<,> The <,> character can be used with the Index, Go, Jog, Home, Delay, Output , Normalize and **SD** commands to synchronize them with the completion of any commanded motion which may be in progress.

<;> The <;> character can be used with the Jog, Home, Delay, Output & **SD** commands, to synchronize them with motion reaching a constant speed or completion. e.g. The command I+;<cr> starts a positive index and then delays until the acceleration is complete before executing the next MPL command.

<:> The <:> character can be used with the Index, Jog, Home, Delay, Output and **SD** commands, to synchronize them with motion reaching the end of constant speed. e.g. The command I+;:<cr> starts a positive index and then delays until the deceleration begins before executing the next MPL command.

Examples:

I,- Wait for the last motion to end before indexing the same distance as previously commanded in the negative direction.

D;<cr> Wait for the system to reach steady state speed or the motion to be completed.

I300,+ After the last motion is complete, index the system 300 counts in the positive direction.

I,+,- Wait for last motion to be completed; index in the positive direction. After motion stops, index in the negative direction.

D300;<cr> Delay 300 msec after reaching steady state speed.

D25:<cr> Delay 25 msec after the end of constant speed

MACHINE I/O INTERFACE

The Machine I/O Interface of ORMEC's PMC-900 Series of Programmable Motion Controllers provides 16 TTL level digital I/O points. These I/O points include 11 discrete inputs and 5 discrete outputs, and are designed to be compatible with industry standard "OPTO-22 style" optically isolated I/O Modules. For logical compatibility with those modules (including the LEDs on them), the inputs and outputs are considered "asserted" or "on" when at 0 volts (TTL low) and "not asserted" or "off" when at 5 volts (TTL high).

General Purpose Machine Inputs

Eight of the eleven machine inputs may be read and used to control "program flow" by the Branch, Exit, Function & Until commands. These commands can specify any combination of the eight inputs to be tested using a <select> parameter of two hexadecimal characters. A single command can test for individual selected inputs to be either "on" or "off" using the test <status> parameter. (also two hexadecimal characters)

Bits in the <select> parameter that are set to 1 cause the corresponding machine input to be "tested" by the MPL command. Bits set to 0 cause the corresponding machine input to be ignored by the command.

Bits in the optional <status> parameter that are set to 1 cause the command to "test" for the corresponding input to be "on", as long as it is selected for test by the <select> parameter. Conversely, setting a <status> *bit to 0 causes the command to* "test" for the corresponding machine input to be "off". If the optional <status> parameter is not specified, all inputs selected by the <select> parameter will be "tested" for the "on" condition.

Special Purpose Machine Inputs

Three of the discrete Machine Inputs have a prespecified affect on the operation of the PMC. Asserting the STOP input will stop both commanded motion and MPL program execution. Asserting the EXECUTE input will cause an MPL program to start. This program to be started is selectable from the Machine I/O using the select (SEL) input, or a five bit address.

Machine Outputs

The Output command can be used to selectively affect any or all of the four General Purpose Machine Outputs, as defined by the <select> parameter. In addition, a single Output command can be used to turn "on" or "off" multiple outputs by specifying their desired state with the <status> parameter.

Setting a bit in the <select> parameter to 1 causes the corresponding machine output to be affected by the command. Conversely, setting a bit in the <select> parameter to a 0 causes the corresponding output to be unchanged by the command. Setting a bit in the optional <status> parameter to a 1 or 0 causes the Output command to turn "on" or "off" the corresponding machine output, as long as it was selected by the <select> parameter. If this parameter is not specified, all the selected outputs will be turned on.

The fifth machine output is the READY output which is asserted whenever the PMC is at the interactive level and "ready" to execute a command or program.

Other Discrete I/O Options

The Machine I/O Interface can also be configured for special automatic operation by setting up option parameters in the X and Z registers. *See the SX and SZ Command Section for details.*

PARAMETER RANGES & UNITS

Motion Parameters		Range	Default	Units
Acceleration		1-65,535	-	100 Hz/sec
Jog	**48kHz**	2- 4,800	-	10 Hz
Velocity	**Mode**	2- 4,800	-	10 Hz
Home		2- 4,800	-	10 Hz
A - Acceleration		1-65,535	100	kHz/sec
AL		1-65,535	4000	kHz/sec
AQ	**192kHz**	1-65,535	4000	kHz/sec
AS	**Mode**	1-65,535	1000	kHz/sec
Jog		1- 1,920	100	100 Hz
Velocity		1- 1,920	400	100 Hz
Home		1- 1,920	20	100 Hz
A - Acceleration		1-65,535	100	kHz/sec
AL		1-65,535	4000	kHz/sec
AQ	**256kHz**	1-65,535	4000	kHz/sec
AS	**Mode**	1-65,535	1000	kHz/sec
Jog		1- 2,560	100	100 Hz
Velocity		1- 2,560	400	100 Hz
Home		1- 2,560	20	100 Hz
Acceleration		1-65,535	-	kHz/sec
Jog	**384kHz**	1- 3,840	-	100 Hz
Velocity	**Mode**	1- 3,840	-	100 Hz
Home		1- 3,840	-	100 Hz
Acceleration		0-65,534	-	100 counts
Jog	**External**	2-10,000	-	.01%
Velocity	**Mode**	2-10,000	-	.01%
Home		2-10,000	-	.01%
Index		1-2,147,483,648	500	counts
Go		0-1,073,741,824	0	counts
Normalize		0-1,073,741,824	-	counts
Delay		0-65,535	-	msec
Delay (commanded motion)		0-4,294,901,759	-	counts
Delay (ref-distance)		0-4,294,901,759	-	counts
<Label>		20_H to $7F_H$	-	-
Tuning & Register Parameters				
Position Loop Gain		0-255	2	-
Velocity Loop Gain		0-255	2	-
Feedforward Gain		0-255	0	-
External Output Gain		0-255	0	-
Velocity Loop Compensator		0-F_H	0	-
Position Loop Compensator		0-F_H	0	-
X Register		0-FF_H	08	-
Y Register		0-FF_H	00	-
Z Register		0-FF_H	10	-
Mode		0-4	0	-

ERROR CODES

Syntax Error Codes

A0 Command not allowed in current velocity range.
A1 An invalid command has been used.
A2 An invalid terminator or designator has been used.
A3 An invalid loop counter ID has been used. Only X, Y, and Z are valid.
A4 The input value is out of the allowable range. See the Parameter Ranges & Units Section for allowable values.
A5 Invalid address for the = (axis identifier) command has been entered. Valid addresses are printable ASCII characters other than the "space" character.
A6 An invalid HEX value, either a condition code or an output value, has been entered.
A7 Reserved
A8 No valid program in the optional socket.
A9 The requested machine code is not recognized.

Motion Error Codes

B0 Command allowed only at rest or at top velocity.
B1 Index distance exceeded.
B2 Command not valid while the system is in motion. A motion designator or programming command was entered when the system was in motion.
B3 Motion cannot be initiated with STOP' low.
B4 Deceleration cannot be triggered while system is accelerating, decelerating or at rest.
B5 An ESCAPE character was received during a synchronization command.
B6 A motion command was entered with DRVOFF (Drive Off) asserted.
B7 Attempt to Index or Go past a software limit.
B8 Attempt to Index or Go with forward limit (+LIMIT') asserted.
B9 Attempt to Index or Go in reverse with reverse limit (-LIMIT') asserted.

Programming Error Codes

C0 Program buffer write protected. An SW1 command must be issued before it can be modified.
C1 Program buffer overflow.
C2 Program label undefined.
C3 The program memory has a storage fault. The last entered programming character was not saved in program buffer due to hardware failure of the RAM, EEPROM or zero-power RAM memory.
C4 An ESCAPE character was received during execution of an MPL program.
C5 Reserved.
C6 A Program command cannot be executed during program execution.
C7 Reserved.
C8 Attempt to modify non-volatile memory with active checksum error.
C9 Function calls nested too deep.

Miscellaneous Error Codes

D0 An ESCAPE character was received during a Delay command.
D1 STOP' signal at the Machine I/O Interface was asserted.
D2 Input operation aborted.
D3 Parameter write protected. Bit 5 of the Z register must be set to disable the write protect feature before a new axis ID or baud rate can be specified.
D4 Reserved
D5 Reserved
D6 Reserved
D7 A SPACE character was received while in motion. Motion stopped.
D8 Contour segment entered while engaged in a contour segment stop.
D9 PSJ overflow causing the PMC to disable the servodrive.