MPL SYNTAX

Acceleration	A <rate> <cr>	A <display>	
Branch	B <label> <condition>		
Delay	D [<time>] [<sync>] <cr>		
Exit Program	E <condition>		
Function Call	F <label> <condition>		
Go	G <position> <direction> <cr>		
	G <display>		
Home	H [<speed>] <direction>	H <display>	
Index	I [<distance>] #<direction># <cr>		
	I <display>		
Jog	J [<speed>] #<direction># <cr>		
	J<display>		
Loop	L [<label>:<count>] <cr>		
Normalize	N [<position>] <direction> N <cr>		
Output	O [<sync>] <hex>	O <display>	
Program	P <cr> <text>	P <label> <text>	
Quit	Q		
Set or Show	SM <mode> <cr>	SM <display>	
	S <register> <hex> <cr>	S <display>	
Tune Loops	T #<register> [<value>] #<sign># #<cr>		
	T [E] <display>		
Until	U <condition>		
Velocity	V <speed><cr>	V<display>	
Label	@ <label> <text> <cr>	@ <text> <cr>	
Assign ID	= <id> <cr>	= <display>	

<label>	Displayable character used to identify a motion routine
<condition>	<hex> character preceded by an optional (-) to invert the required input levels. A <cr> is always a go condition.
<count>	Number of times for operation to be repeated
<sync>	Synchronization character for coordinating motion: , wait until current motion is complete ; wait until constant speed or motion complete + / - positive/negative <speed>, <position> or <distance> * stop system motion <sync> permitted in most cases
<rate>	Acceleration rate in kHz/sec; 100 Hz/sec, or 100's counts
<speed>	Speed in 10 Hz, 100 Hz or .01%
<distance>	Number of relative encoder counts
<position>	Absolute position in encoder counts
<time>	Time in milliseconds
<text>	Motion routines, comments or editing command characters
<display>	? display last entered value ! display current system value % [<time>] display status each <time> interval (valid for G,H,I,J,O,TE,V only) <sync> permitted in most cases
<mode>	Control mode: 0=idle; 1=velocity; 2 =position; 3=position without resetting position error
<register>	Tuning: P=(position gain); V=(velocity gain), F=(feedforward gain), X =(external output gain), CP=(position loop compensation), CV=(velocity loop compensation) S=Status: X, Y, Z
<value>	Value substituted for, added to or subtracted from <register>
<sign>	+/- add/subtract <value> to/from <register> = set <register> to <value>
<hex>	Hexadecimal characters (0-9, A-F)
<id>	Motion axis identifier (most displayable characters)
<cr>	Carriage return ($0D_H$)

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

MOTION SET-UP COMMANDS
Selecting System Options

Tuning Commands

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

- (Assign Axis ID) Command

<id><cr> Assign axis identifier. <id> is any printable ASCII character greater than a SPACE (20 H) except for !, ? and -.

! Display the ID of the axis currently in use.

? Display the ID value currently stored in program memory.

<cr> Turn off serial bus support.

Machine Code Command

M Select optional firmware. The PMC verifies optional firmware is present before execution.

Set/Show Register (X, Y, Z) Commands

S<r> <hex><cr> Set or examine one of the control status <r> registers (X, Y, Z) to change the configuration of the motion control system. <hex> is the two character 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 ACSII hex characters. e.g. 080000

X Register

Bit 7 Reserved.

Bit 6 MOTION BUS SLAVE selects the Motion Reference Bus as the master reference for creating motion instead of the internal crystal controlled clock. (1 on)

Bit 5 ALTERNATE REFERENCE ENABLE 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 with respect to other moving machinery. (1 on)

Bit 4 MOTION BUS MASTER causes system to become motion bus master by supplying its motion reference pulses to the Motion Reference Bus. (1 on)

Bit 3-2 VELOCITY RANGE SELECT selects velocity range as follows:

RANGE	Bit 3	Bit 2
48k Hz	0	0
reserved	0	1
192k Hz	1	0
384k Hz	1	1

Bit 1 ENABLE LIMITS enables Machine I/O inputs I2 and I3 to be used as - and + limit switch inputs respectively.

Bit 0 DIRECTION INVERT transposes the meaning of + and - in motion commands.

Y Register

Bit 7-6 ACCEL PROFILE SELECT specifies an acceleration profile by setting bits 6 & 7 as follows:

Bit 7	Bit 6	type
0	0	- linear
0	1	- s-curve
1	0	- parabolic
1	1	- reserved

Bit 5 EXTERNAL START causes a motion to start upon receiving an external signal. Bit 4 will indicate which signal will 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 EXTERNAL DECEL causes deceleration to occur on the machine sensor input* (SENSIN) instead of a calculated distance (after full speed is attained). Ordinarily, deceleration is initiated when the remaining distance is equal to the acceleration distance. (1 on)

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 INDEX EXTEND specifies that speed should remain at the level set by the **J** command during deceleration 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. (1 on)

Bit 0 SHARP JOG STOP selects a sharp (immediate) stop upon jog deceleration rather than the deceleration rate specified by the **A** command. (1 on)
Note: The Y Register is in the motion buffer and therefore, altering it during a motion will only effect the next command motion.

Z Register

Bit 7 BINCOM ENABLE enables binary host communications. Binary input or output is actually selected by setting the high order bit of the ASCII command character preceeding the data transfer. That character will then be followed by the appropriate number of two's complement binary interpreted bytes (most significant byte first). These bytes are not echoed regardless of the setting of Bits 0 & 1.

Command	# bytes	terminator
A	2	None
D	2	None
G	4	Yes
H	2	Yes
I	4	Yes
J	2	Yes
N	4	None
S	3	None
V	2	None
T	6	None

The high order bit is ignored for all other commands not in above table. The <cr> terminator is not required with the commands indicated above and the] prompt is returned after the specified number of bytes have been sent.
Note: The Go and Normalize commands expect their arguments in two's complement form and the <cr> initiates action. The <sign> character is not recognized in binary mode. Binary output from the PMC (in two's complement form) is requested by setting the high order bit of the ASCII "?", "!" or "%" terminator. The number of binary output bytes is specified by the above table.

Bit 6 HEXCOM specifies that all input data be interpreted and output data be displayed in ASCII hexadecimal. (1 on)

Bits 5-4 Reserved

Bits 3-2 Reserved

Bit 1 PROGRAM TRACE displays each MPL command at the SCI (if active) as it is executed. (1 on)

Bit 0 NO ECHO prevents echo of SCI characters. (1 on)

SYNTAX ERROR CODES

A1 An invalid command has been used.
A2 An invalid terminator or designator has been used.
A3 Reserved
A4 The input value is out of the allowable range. See the "Ranges & Units" table for allowable values.
A5 Invalid address for the (axis identifier) command has been entered.
A6 An invalid HEX value, either a condition code or an output value, has been entered.
A7 The requested velocity range is not available.
A8 No valid program in the optional socket.
A9 The requested machine code is not recognized.

MOTION ERROR CODES

B1 Currently unused.
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 Currently unused.
B5 An ESCAPE character was received during a synchronization command.
B6 A motion command was entered with DRVOFF (Drive Off) asserted.
B7 Reserved
B8 Attempt to move forward with forward limit (+LIMIT) asserted.
B9 Attempt to move in reverse with reverse limit (-LIMIT) asserted.

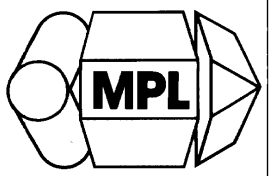
PROGRAMMING ERROR CODES

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 or EEPROM memory.
C4 An ESCAPE character was received during execution of an MPL program.
C5 Currently unused.
C6 A Program command cannot be executed during program execution.

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.

MOTION PROGRAMMING LANGUAGE REFERENCE CARD Version 2D



ORMEC SYSTEMS CORP

Ormec's Motion Programming Language (MPL) provides the systems designer with a versatile tool for writing simple programs that create high performance motion. This Reference Card is designed to give MPL users quick access to command syntax and programming options to make programming even simpler to implement. For more in-depth information, consult an ORMEC manual.

MOTION PARAMETER COMMANDS
Defining Motion Parameters

To define a motion using MPL, the user interacts with integer values contained in a motion buffer which specify the motion in terms of the digital position encoder's position pulses. Distance is specified as a number of digital encoder increments, velocity is specified as the desired frequency from the digital position encoder, and acceleration is specified as a slope of frequency per unit time.

Velocity Commands

V<speed><cr>	Set index speed in motion buffer.
V?	Display index speed currently in motion buffer.
V!	Display current system speed.
V%	Repeatedly display current system speed.

Acceleration Commands

A<rate><cr>	Set acceleration rate in motion buffer.
A?	Display acceleration rate currently in motion buffer.
A!	Display current system acceleration rate (zero if at rest or top speed).

Index Commands

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

Jog Commands

J<speed><cr>	Set jog speed in motion buffer.
J?	Display jog rate currently in motion buffer.

Home Command

H<speed><cr>	Set homing speed in motion buffer.
H?	Display homing speed currently in motion buffer.

MOTION ACTION COMMANDS
Creating Motion

Index Commands

I+<cr>	Move in a positive direction the previously specified index distance. Entering the (+) initiates the motion; the <cr> terminates the I command.
I<dist><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.

Go Commands

G<cr>	Move to the absolute zero 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!	Display the absolute position of the system.
G%<cr>	Repeatedly display the absolute position of the system.
G?	Display the commanded absolute position.
G*	Stop system motion.

Jog Commands

J+<cr>	Jog in a positive direction at the jog speed in the motion buffer. Any character except a <cr> or period (.) stops the motion.
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.

Home Commands

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.

PROGRAM BUFFER COMMANDS
Entering, Editing or Displaying a Program

Program Commands

P!	Enter program mode with the cursor at the beginning of the program buffer.
P<L>	Enter program mode with the cursor at program label <L>.
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.

@ Label Commands

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

Editing Functions Used During Program Mode

Cursor Right	TAB (CTRL-I) or CTRL-Y moves the cursor to the right one character at a time. Moving the cursor to the end of the line and continuing to tab will move the cursor to the beginning of the next line.
Cursor Left	BACKSPACE (CTRL-H) or DELETE moves the cursor to the left one character at a time. Moving the cursor to the beginning of the line and continuing to backspace will move the cursor to the beginning of the previous line.
Cursor Down	LINEFEED moves the cursor down a line at a time.
Changes	To change a mot on control program, put the cursor at the point to be changed and overtype the desired information. Periods (.) may be used to overtype additional undesired characters or to reserve program buffer space for additional future commands or parameter changes.
Insert Line	Typing CTRL-V allows text to be inserted in the program buffer at the point of the cursor. After a CTRL-V, all characters typed are put into a 40 character RAM buffer until a second CTRL-V is typed or the RAM buffer is full. At that time, space is made in the program buffer and the characters are written to program buffer memory. Because of the time required to rewrite EEPROM, this operation may take several seconds, and so a bell will sound both the beginning and the end of the insert. If the insert operation is a result of the 40 character buffer being full, the insert operation is continued at the second bell. An ESCAPE can be used to exit from insert mode without inserting any characters.
Kill Line	CTRL-K deletes unwanted characters in the program buffer. When CTRL-K is typed, all characters from the cursor to the end of the line (next carriage return) will be deleted. A bell will sound the beginning and the end of this operation.
Exiting	The ESCAPE key is used for exiting the program command.
Program Erase	Typing a 1 in column 1 (immediately after a <cr>, will erase the program buffer, starting at the current location, and exit the program command. **Note that executing this command will erase all information from the cursor to the end of the program buffer.** Erasing the buffer can take several seconds, so the PMC will output a "." every 3 to 4 seconds to indicate it is still working.

PROGRAM CONTROL COMMANDS
Utilizing Subroutines & Creating Complex Motion Control Applications

Loop (Repeat) Commands

L<L> <count>	Transfer MPL program execution to a program label <L> a specified number of times, and then continue program execution with the next command in the program buffer. **Caution:** Program loops cannot be nested.
L<cr>	Clear the loop counter.

Branch (Go To) Commands

B<L> <cr>	Unconditionally transfer MPL program execution to a program label <L> with no return.
B<L> <condition>	Transfer MPL program execution to a program label <L> with no return if the machine input <condition> is met. (See Machine Input Condition explanation below.)

Function (Subroutine) Call Commands

F<L> <cr>	Unconditionally transfer MPL program execution to a program label <L>. When an Exit command is executed, MPL operation resumes at the line following the "F-command".
F<L> <condition>	Transfer MPL program execution to a program label <L> if the machine input <condition> is met. (See Machine Input Condition explanation below.) When an Exit command is executed, MPL operation resumes at the next program line after the "F-command". **Caution:** Functions cannot be nested.

Exit (Return) Commands

E<cr>	Unconditionally exit an MPL routine.
E<condition>	Exit an MPL routine if the machine input <condition> is true. (See Machine Input Condition explanation below.)

Quit Command

Q	Terminate execution of MPL from the program buffer and return to the interactive mode.

Machine Input Conditions

<Condition> is a hexadecimal character specifying which of the four machine inputs should be checked. For example, 6 specifies that inputs I1 and I2 are to be checked. The test is for the specified inputs to be low, unless a negative condition <-condition> is used which indicates that the specified bits should be high. If <condition> is true, the statement will be executed. If it's not true, the next command will be executed.

INTERFACE/SYNCHRONIZATION COMMANDS
Synchronizing Motion

Delay Commands

D<time><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.

Until Commands

U<condition>	Wait until the specified machine input <condition> is true before executing the next command. (See Machine Input Condition explanation under PROGRAM CONTROL COMMANDS for more information.)

Synchronization Characters

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 to synchronize with the motion being created.

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

MACHINE INPUT/OUTPUT COMMANDS
Examining and Manipulating Machine Outputs

Output Commands

O<hex>	Set the four general purpose machine outputs to the specified state <hex>, where <hex> is a hexadecimal digit which specifies which outputs are "on" or "off".
O?	Display the current state of the general purpose machine outputs.
O!	Display the current state of the general purpose machine inputs and outputs.
O%<cr>	Repeatedly display the current state of the general purpose machine inputs and outputs.

MOTION SET-UP COMMANDS
Selecting System Options

Normalize Commands

N<cr>	Start SCI automatic baud rate selection.
N*	Reset PMC software.
N<position>+	Define the current physical position. This command resets the internal absolute position counter to the position specified. The sign (+ or -) of the position follows the numerical specification and terminates the command.

Set/Show Mode Commands

SM<mode><cr>	Set control mode <0> IDLE (servodrive off and servo loops disabled) <1> VELOCITY control mode <2> POSITION control mode <3> POSITION mode without clearing position error
SM!	Display current motion status and current control mode. Lower Nibble: 0 IDLE; 1 VELOCITY; 2 POSITION mode Upper Nibble Bit Assignments: bit 4 - direction (1 forward), bit 5 motion (1 in motion); bit 6 position summing junction overflow (1 overflow)
SM?	Display last entered mode.

PARAMETER RANGES & UNITS

	Range	Default	Units
Motion Parameters			
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
Acceleration	1-65,535	4000	kHz/sec
Jog	192kHz 1- 1,920	100	100 Hz
Velocity	Mode 1- 1,920	400	100 Hz
Home	1- 1,920	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 Ext	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	0	counts
Delay	0-65,535	0	msec
<Label>	2 byte 7D$_H$	—	—
Tuning Parameters			
Position Loop Gain	0-255	2	—
Velocity Loop Gain	0-255	2	—
Feedforward	0-255	0	—
External Output Gain	0-255	0	—
Velocity Loop Compensator	0-F	0	—
Position Loop Compensator	0-F	0	—
	10-1F*	0	—
Register Parameters			
X Register	—	08	—
Y Register	—	00	—
Z Register	—	00	—
Mode	—	0	—

*—Two ranges are shown for the Position Loop Compensator because the presence of bit 0 in the upper nibble designates whether the time optimal positioning compensation is enabled (1 => enabled).

ORMEC SYSTEMS CORP. • 19 Linden Park • Rochester, NY 14625 • 716-385-3520

RCBM/MM-000