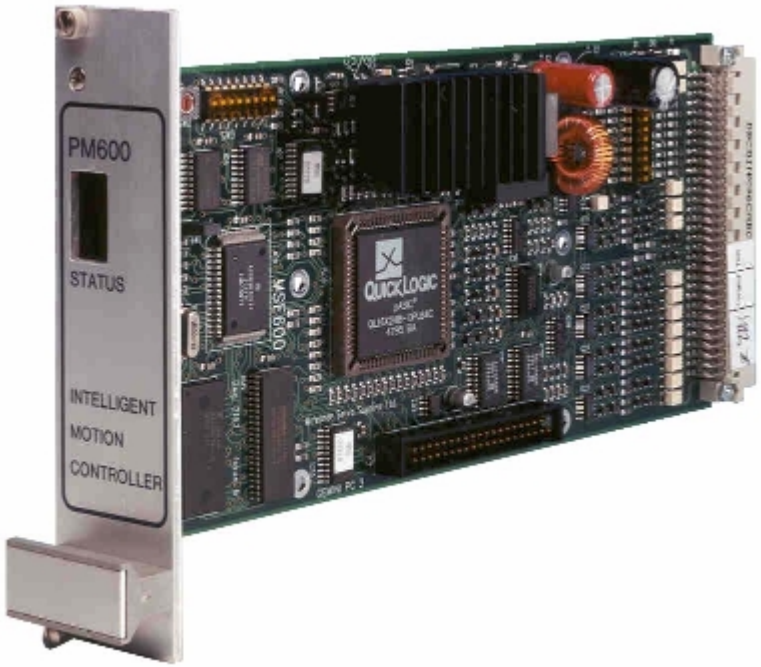
Mclennan Servo Supplies Ltd.

Mclennan



PM600 Motion Controller
USER Manual

PRODUCT MANUAL FOR PM600 Digiloop ®

SAFETY NOTICE !

Position control systems are inherently hazardous. Even a small motor, if coupled to a leadscrew, gearbox, or any other form of mechanism that provides a mechanical advantage, can generate considerable force and could cause serious injury. Incorrect operation can also lead to damage to the motor or associated machinery. It is essential that the purpose of the various safety features built into the PM600 be fully understood and used correctly.

Caution

STATIC SENSITIVE DEVICES

This unit has static sensitive devices. Observe handling precautions: Hold card by edges only. Do not touch components or connector pins. Ship only in anti-static packaging.

Mclennan Servo Supplies Ltd.
22 Doman Road,
Yorktown Industrial Estate,
Camberley,
Surrey,
GU15 3DF
UK

Telephone: +44 (0)1276 26146
FAX: +44 (0)1276 23452
Sales EMAIL: sales@mclennan.co.uk
Technical support EMAIL: tech@mclennan.co.uk

This manual is written for firmware version – V3.25a

The manufacturer reserves the right to update the data used in this manual in line with product development without prior notice.

This manual printed: Wednesday, 28 March 2001

1 Commands – How to talk to the PM600

1.1 COMMAND WORD SYNTAX

The PM600 has a wide range of command options extending beyond the main move functions. The aim is to provide a flexible and comprehensive control device for integration of motion control into larger systems.

1.1.1 Commands

Most commands are two letters, the function of each, being described in the Programmer Reference section. Each command is preceded by the appropriate address to identify the axis for which the command is intended. Where applicable (e.g. move commands, setting of system parameters, etc.) the command should then be followed by the desired value:

aXXnnn<cr>

a = address
XX = command
nnn = value (if required)
<cr> = carriage return.

Command strings should be terminated with a carriage return character (ASCII 0D hex).
Upper or lower case characters may be used for the command. Spaces within the command line are ignored. If no value is given, then zero is assumed.

All commands except for **Control C** and **ESCAPE** are buffered. Commands are executed in consecutive order. Commands will be acted on sequentially, as they have been entered. If any command cannot be executed immediately (because it may need to wait for some condition or a previous command to finish) then the command and any that follow it will be buffered internally (up to 256 characters). The responses for each command are returned as the command is executed.

NOTE: Delete, backspace and cursor movement characters are **NOT** detected by the controller. With some terminals or emulators these keystrokes will be translated as an escape sequence, i.e. a sequence of characters beginning with an escape character (ASCII 1B hex). The controller will detect the escape character and act on it accordingly.

Please note that due to the loop nature of the RS232 communications, all characters that are sent to the controllers are echoed back to the host.

1.1.2 Replies

Response to a command, once it has been accepted, is either an **OK** string or an alpha-numeric string. Responses terminate in a carriage return character (0D) and a line feed character (0A) and are preceeded by the address number and a colon character. An appropriate message is sent if a mistake or conflicting instruction creates an error. The first character of an error message is **!**

1.1.3 Non-volatile Memory

All set-up parameters (control coefficients, acceleration, deceleration, velocities, jog speeds, creep speed, etc.), sequences and profiles will be read from the on board FLASH memory to the controllers normal RAM memory on power-up. The parameters may then be modified in the volatile RAM by the relevant commands, but these modifications will not persist after power down. Use the backup (BA, BC, BD, BP & BS) commands to write the current set-up back to the non-volatile FLASH memory, so that they will be 'remembered' on power up.

CONTROL **C** (ASCII *03*) **Hard Stop**.

Moves, sequences and profiles halted immediately.
Motion stopped at the **LD** rate.
Command buffer cleared.
Sets status to **Idle**.
Sequences and profiles retained in memory.
Operates on all axes.

ESCAPE (ASCII 27) **Soft Stop**

Moves, sequences and profiles halted immediately.
Motion stopped at the **SD** rate.
Command buffer cleared.
Status returns to **Idle**.
Sequences and profiles retained in memory.
Operates on all axes.

AB ABORT

The control of the motor may be aborted by sending **AB**. When aborted, the servo loop is disabled and the status display will show **C** for command abort. A user abort may be reset with the **RS** command. The encoder positions are still read while aborted.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>AB	N/A	N/A		N/A	0

Condition Requirements
None.

Notes:
The response to a CO command will be Command Abort. It will override any other abort situation.

Responses:

! COMMAND ABORT	Command has been accepted.

Example:

1AB Abort axis 1.

AD AUTO-EXECUTE SEQUENCE - DISABLE

Switches off any auto-execute sequences that may have been set by the **AE** command. Note that this setting is written to the non volatile FLASH memory and is therefore retained after power down.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>AD				Disabled	5

Condition Requirements
Idle

Notes:
Value stored in FLASH

Responses:

OK	Command has been accepted.

Example:

1AD Disable auto execute of axis 1.

AE AUTO-EXECUTE SEQUENCE - ENABLE

Set sequence *n* to run on power-up of the controller (auto-execute). This may be used in stand alone systems where there is no permanent host computer or terminal. Note that this setting is written to the non volatile FLASH memory and is therefore retained after power down.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>AEn	Seq. No.	0	7	Disabled	5

Condition Requirements
Idle

Notes:
Value stored in FLASH

Responses:

OK	Command has been accepted.
! INVALID SEQUENCE NUMBER	Argument is out of valid range.
! SEQUENCE UNDEFINED	Sequence specified has not been defined yet.

Example:

1AE5 Sets auto execute of axis 1 to run sequence 5 on power-up.

AI READ ANALOGUE INPUT

Read Analogue Input. If the argument is 1 or 2 then the value will be read from the bi-polarAnalogue inputs 1 or 2 respectively. These return a value between –2047 and +2047 at a scaling of 5mV per unit (±10V). For arguments 3 to 5 will give the unipolar values for Joystick1, Joystick2 and JoystickCT respectively for the range 0 to 4095 for 0V to 5V.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>AIn	N/A	1	5	N/A	0

Condition Requirements **Notes:**
None

Response:
The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by something of the format AD**C1 = or ! NOT VALID ADC CHANNEL**

Example:
If the controller of axis 1 currently has a Command Position of 45280 then the command:
1AI3 in Verbose Mode will respond: **01:Joystick 1 = 2040 (approx 2.5V)**
1AI3 in Quiet Mode will respond: **01:2040**

AG WAIT FOR ANALOGUE INPUT GREATER THAN

Wait until an Analogue Input is greater than a value. If the first argument is 1 or 2 then the value will be read from the bi-polar Analogue inputs 1 or 2 respectively. The value is between –2047 and +2047 at a scaling of 5mV per unit (±10V). If the first argument is 3 to 5, then the unipolar values are for Joystick1, Joystick2 and JoystickCT respectively for the range 0 to 4095 for 0V to 5V.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>AGn/xxx	N/A	1	5	N/A	1

Condition Requirements **Notes:**
None

Response:
OK The condition has come true or
! NOT VALID ADC CHANNEL

Example:
1AG2/1000 Wait until Analogue Input 2 is greater than 2.5V
1AG3/2047 Wait until Joystick 1 Input is greater than 2.5V

AL WAIT FOR ANALOGUE INPUT LESS THAN

Wait until an Analogue Input is less than a value. If the first argument is 1 or 2 then the value will be read from the bi-polar Analogue inputs 1 or 2 respectively. The value is between –2047 and +2047 at a scaling of 5mV per unit (±10V). If the first argument is 3 to 5, then the unipolar values are for Joystick1, Joystick2 and JoystickCT respectively for the range 0 to 4095 for 0V to 5V.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>ALn/xxx	N/A	1	5	N/A	1

Condition Requirements **Notes:**
None

Response:

OK	The condition has come true or
! NOT VALID ADC CHANNEL	

Example:

1AL2/-1000	Wait until Analogue Input 2 is less (more negative) than 2.5V
1AL3/2047	Wait until Joystick 1 Input is less than 2.5V

AM SET ABORT MODE

Set the conditions that cause an abort and disable the control (servo) loop.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>AMabcdefgh	Bits	0	1	00000000	6

where
- **a** - 0 – Abort Stop Input disables control loop
  1 – Abort Stop Input stops all moves only
- **b** - 0 –Abort Stop Input is latched requiring RS command to reset
  1 – Abort Stop Input is only momentary
- **c** - 0 – Stall Error disables control loop
  1 – Stall Error is indicated but control loop remains active
- **d** - 0 – Tracking Error disables control loop
  1 – Tracking Error is indicated but control loop remains active
- **e** - 0 – TimeOut Error disables control loop
  1 – TimeOut Error is indicated but control loop remains active
- **f** - Reserved for future use.
- **g** - Reserved for future use.
- **h** - 0 – Enable output switched OFF during a disabled control loop
  1 – Enable output left ON during a control loop abort

Condition Requirements	Notes:
Idle	Bit **a** = **0** and bit **b** = **1** is not appropriate and should not be used.

Response:

OK	Command has been accepted.
! OUT OF RANGE	Illegal argument range or format

Examples:

	1AM00010100	Set axis 1 to abort on all conditions except Tracking Error, enable output stays ON.
or	1AM11000000	Set axis 1 to abort on all conditions except momentary Abort Stop input only stops moves.

AO SET ANALOGUE OUTPUT

Set the voltage of an Analogue Output. The first argument is 1 or 2 and the value will be written to the bi-polar Analogue outputs 1 or 2 respectively. The value is between –2047 and +2047 at a scaling of 5mV per unit (±10V).
The Analogue Output channel needs to be available, so AO2/xxx only may be used single channel servo mode, and both may be used in stepper motor modes.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>Aon/xxx	N/A	1	2	N/A	3

Condition Requirements	Notes:
None	

Response:

OK	Command has been accepted or
! NOT VALID DAC CHANNEL	

Examples:

1AO1/1024	Set the Analogue output 1 to +5V
1AO2/-2047	Set the Analogue output 2 to -10V

AP SET ACTUAL POSITION

Set the actual position (position of motor or mechanism) value to that given in the argument.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>APnnn	Steps	-2147483647	2147483647($\pm 2^{32}$)	0	3

Condition Requirements
Idle, Constant velocity or Gearbox

Notes:
Value zero on power-up.

Response:

OK	Command has been accepted.

Examples:

	1AP5000	Set the axis 1 Position to 5000.
or	1AP0	Set the axis 1 Position to zero.

BA BACKUP ALL

Saves parameters, sequence definitions, profile definitions and cam definitions to *flash* memory. These are then restored on power-up.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>BA	n/a	n/a		n/a	5

Condition Requirements
Idle

Notes:

Response:

OK	Command has been accepted.

BC BACKUP CAMS

Saves all cam definitions to *flash* memory. These are then restored on power-up.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>BC	n/a	n/a		n/a	5

Condition Requirements
Servo mode, Idle

Notes:

Response:

OK	Command has been accepted.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

BD BACKUP DIGILOOP PARAMETERS

Saves Digiloop parameters to *flash* memory. These are then restored on power-up.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>BD	n/a	n/a		n/a	4

Condition Requirements
Idle

Notes:

Response:

OK	Command has been accepted.

BO SET BACKOFF

Set number of back-off steps that are executed at the end of a move. The motor will decelerate to the creep speed at the back-off position relative to the required end position. The controller will then complete the move at the creep speed. The controller will therefore always approach the final position at the creep speed and from the same direction. This may be useful in combating backlash in a mechanism.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>BOnnn	Steps	-2147483647	2147483647(2^{32})	0	6

Condition Requirements
Idle

Notes:
Value stored in FLASH by BD command.

Response:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Examples:

1BO500	Set the back-off distance to 500 steps on axis 1.

BP BACKUP PROFILES

Saves all profile definitions to *flash* memory. These are then restored on power-up.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>BP	n/a	n/a		n/a	5

Condition Requirements
Servo mode, Idle

Notes:

Response:

OK	Command has been accepted.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

BS BACKUP SEQUENCES

Saves all sequence definitions to *flash* memory. These are then restored on power-up.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>BS	n/a	n/a		n/a	5

Condition Requirements
Idle

Notes:

Response:

OK	Command has been accepted.

CD CLEAR CAPTURED DATUM

Clears the current captured datum position.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>CD	n/a	n/a		n/a	3

Condition Requirements
None

Notes:

Response:

OK Command has been accepted.

CM SET CONTROL MODE

Sets the current control mode. When changing from servo mode to a steeper mode or vice versa, the controller will be forced into a command abort and will reset all parameters to the default values for the new mode.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>CMnn	n/a	1, 11, 12, 13 or 14		1	8

CM1 Servo mode
CM11 Open loop stepper mode
CM12 Checking stepper mode
CM13 External loop stepper mode
CM14 Closed loop stepper mode

Condition Requirements
Idle

Notes:
Forces a Command Abort if changing between servo/stepper

Response:

OK Command has been accepted (no servo/stepper change).
! COMMAND ABORT Command has been accepted (servo/stepper change forced abort).

CO QUERY CURENT OPERATION

This command will return the current operation being executed by the controller.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>CO	n/a	n/a		n/a	0

Condition Requirements
None

Notes:

Responses:

Mode = and one of the following:

Backoff Executing backoff correction
Cam Executing Cam profile
Cam synchronisation Waiting for synchronisation in cam mode
Command Abort Aborted due to command abort (AB) command
Constant velocity Constant velocity
Correcting Closed loop stepper correcting
Creep Creep steps at end of move
Delay Executing delay command
Gearbox Gearbox mode
Gearbox synchronisation Waiting for synchronisation in gearbox absolute mode
Home to datum Searching for datum

Idle	
Input abort	Aborted due to Abort Stop input
Jogging	Jogging or Joystick move
Move	Move (MA or MR command)
Profile	Executing profile
RS232 abort	Illegal serial character(s) received
Settling	Settling at end of move
Stopping	Decelerating due to limit, Ctrl-C or ESCAPE command
Stall Abort	Aborted due to Stall Error
Tracking Abort	Aborted due to Tracking Abort
Not Complete /Timeout Abort	Aborted due to Timeout Abort
Wait for condition	Waiting for specific pattern on read port

CP SET COMMAND POSITION

Set the command position value to that given in the argument. The command position is the position generated by a move command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>CPnnn	Steps	-2147483647	2147483647($\pm 2^{32}$)	N/A	3

Condition Requirements
Idle, Constant velocity or Gearbox

Notes:
Value zero on power-up.

Response:

OK	Command has been accepted.

Examples:

	1CP5000	Set the axis 1 Command Position to 5000.
or	**1CP0**	Set the axis 1 Command Position to zero.

CR SET CREEP DISTANCE

Set number of creep steps at the end of a move. The motor will decelerate and execute this number of steps at the creep speed.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>CRnnn	Steps	0	2147483647(2^{32})	0(servo) 10(stepper)	6

Condition Requirements
Idle

Notes:
Value stored in FLASH by BD command.

Response:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Examples:

1CR50	Set the creep distance to 50 steps on axis 1.

CV CONSTANT VELOCITY MOVE

A Constant velocity move is used to move continuously at the required speed. Initially the move will ramp up to the speed given in the argument, where the sign dictates the direction of movement. The argument therefore

controls the velocity. Subsequent CV commands may then be sent to change the required velocity, including changes in direction.
The speed is changeable whilst motion is in progress. The **SA** and **SD** rates define the rate at which the change of speed will be made. Constant velocity mode is exited by an **ST** command, **ESCAPE** or Control **C**.
The soft limits are active in CV mode. For continuous applications they must be disabled with the SL command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>CVnnn	steps/sec	-400,000	400,000	N/A	1

Condition Requirements
Idle or Constant velocity.

Notes:

Responses:

OK	Command has been accepted.
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Soft limit for required direction has already been reached
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected

Examples:

1CV2000	Start constant velocity move in positive direction at 2000 steps/sec on axis 1.
1CV-10000	Start constant velocity move in negative direction at 10,000 steps/sec on axis 1.

DC DEFINE CAM

This command will start a Cam profile definition. There are eight cams that may be defined and the argument selects which profile is to be defined (0 to 7). The only command that is used during a Cam profile definition is **XY.** Any other commands except for **EC** will cause a **!ILLEGAL CAM INSTRUCTION** error.
The commands that follow this **DC** command will not be executed, but will be stored in the on board volatile memory until the End Cam definition (**EC**) command is received. If a Control-C or ESCAPE command is received or the controller runs out of memory, the Cam definition will cease, the Cam will not be stored and the controller will return to the idle state. The Cam Modulo (profile length) is calculated automatically.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>DCn	Cam number	0	7	N/A	5

Condition Requirements
Servo mode, Idle

Notes:
If a Cam is defined, that fact is shown on the QA page.
Use BP command to store profiles in non-volatile flash memory.

Responses:

OK: START OF CAM	Command has been accepted.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1DC3	Start definition of Cam 3.
1XY200/500	Second Cam profile point. (First Cam profile co-ordinates 0,0.)
1XY400/500	Next Cam Profile point.
1XY600/-200	"
1XY700/-200	"
1XY750/0	"
1EC	End of Cam profile definition.

DE DELAY

This command will start a delay timer for the length given in the argument. After the expiry of this time, the controller will return to the idle mode.
If the following command should not be executed until the end of this delay time, and does not wait for the idle state itself, then you must follow it with a Wait for End (**WE**) command. This will wait until the controller returns to the idle mode and will therefore sense the expiry of the delay time.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>DEnnn	Milliseconds	1	2147483647(2^{32})	N/A	1

Condition Requirements **Notes:**
Idle

Responses:
OK Command has been accepted.

Example:

1DE5000 Delay for 5 seconds
1WE Wait for end of delay
1WP22222221 Put write port 1 on after delay.

DM SET DATUM MODE

Set the mode of operation for datum searches.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>DMabcdefgh	Bits	0	1	00000000	6

where **a** - 0 – Encoder index input polarity is normal
1 – Encoder index input polarity is inverted
b - 0 –Datum point is captured only once (i.e. after HD command)
1 – Datum point is captured each time it happens
c - 0 – Datum position is captured but not changed
1 – Datum position is set to Home Position (**SH**) after datum search (**HD**)
d - 0 – Automatic direction search disabled
1 – Automatic direction search enabled
e - 0 – Automatic opposite limit search disabled
1 – Automatic opposite limit search enabled
f - Reserved for future use.
g - Reserved for future use.
h - Reserved for future use.

Condition Requirements **Notes:**
Idle

Response:
OK Command has been accepted.
! OUT OF RANGE Illegal argument range or format

Examples:
1DM00100000 Set axis 1 to normal datum capture with automatic setting the captured position to Home position.

DP DEFINE PROFILE

This command will start a Profile definition. There are eight profiles that may be defined and the argument selects which sequence is to be defined (0 to 7). The only command that is used during a Profile definition is **MR** any other commands except for **EP** will cause a **!ILLEGAL PROFILE INSTRUCTION** error.
The commands that follow this **DP** command will not be executed, but will be stored in the on board volatile memory until the End Profile definition (**EP**) command is received. If a Control-C or ESCAPE command is received or the controller runs out of memory, the Profile definition will cease, the Profile will not be stored and the controller will return to the idle state.

Syntax	**Units**	**Range**	**to**	**Initial State**	**Privilege level**
<ad>DPn	Profile No.	0	7	N/A	5

Condition Requirements
Servo mode, Idle.

Notes:
If a Profile is defined, that fact is shown on the QA page.
Use **BP** command to store profiles in non-volatile flash memory.

Responses:

OK	Command has been accepted.
!ILLEGAL PROFILE INSTRUCTION	Command has not been accepted
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1DP	Start Profile definition.
1MR200	First Profile move.
1MR500	Next Profile move.
1MR-500	"
1MR-200	"
1MR50	"
1EP	End of Profile definition.

DS DEFINE SEQUENCE

This command will start a sequence definition. There are eight sequences that may be defined and the argument selects which sequence is to be defined (0 to 7). All valid commands that follow this **DS** command will not be executed, but will be stored in the on board volatile memory until the End Sequence definition (**ES**) command is received. If a command is not suitable for inclusion in a sequence, the controller will respond **!ILLEGAL SEQUENCE INSTRUCTION**.
If a Control-C or ESCAPE command is received or the controller runs out of memory, the sequence definition will cease, the sequence will not be stored and the controller will return to the idle state. . Note that a **BS** command will be needed to copy the sequences to the non volatile FLASH memory, otherwise it will be lost on power down.

Syntax	**Units**	**Range**	**to**	**Initial State**	**Privilege level**
<ad>DSn	Seq. No.	0	7	N/A	5

Condition Requirements
Idle

Notes:
The sequences defined are shown on the QA page.
Use **BS** command to store sequences in non-volatile flash RAM

Responses

OK: START OF SEQUENCE	Command has been accepted.

Example:

1DS4	Start definition of sequence 4.
1SV2000	Set slew speed.
1MA8000	First move (absolute).
1MR5000	Next move (relative).

1MR3000	Next move (relative).
1SV50000	Set new slew speed.
1MA0	Next move (return to start position).
1XS4	Execute sequence 4 (loop to start of this sequence).
1ES	End of sequence definition.

EC END CAM DEFINITION

This command will end a Cam profile definition. The Cam definition must have been started by the Define Cam (**DC**) command. No argument is necessary as the sequence number is specified with the Define Cam (**DC**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>EC	N/A	N/A		N/A	5

Condition Requirements **Notes:**
Servo mode, Define Cam

Responses

OK	Command has been accepted.
! ILLEGAL INSTRUCTION	EC attempted when NOT already defining a Cam.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1DC	Start definition of Cam.
1XY200/500	Second Cam profile point.
1XY400/500	Next Cam Profile point.
1EC	End of Cam definition.

EP END PROFILE DEFINITION

This command will end a Profile definition. The Profile definition must have been started by the Define Profile (**DP**) command. No argument is necessary as the sequence number is specified with the Define Profile (**DP**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>EP	N/A	N/A		N/A	5

Condition Requirements **Notes:**
Servo mode, Define Profile

Responses:

OK	Command has been accepted.
! ILLEGAL INSTRUCTION	EP attempted when NOT already defining a Profile.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1DP	Start Profile definition.
1MR200	First Profile move.
1MR500	Next Profile move.
1EP	End of Profile definition.

ER ENCODER RATIO

Set encoder ratio. The encoder ratio is specified by two arguments separated by a **/** character, and is therefore specified as a fraction with the format: numerator**/**denominator. The incoming position encoder pulses and then scaled by this ratio to derive the Actual Position.
Great care must be exercised in setting this ratio as it will affect the stability of a servo loop (K constants etc.). Beware that if you scale the position up (fraction of greater than 1), then certain positions will then become unobtainable.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>ERnnn/nnn					7
Numerator	N/A	-32768	32767 ($\pm 2^{15}$)	1	
Denominator	N/A	1	32767 (2^{15})	1	

Condition Requirements
Idle.

Notes:
Value stored in FLASH by BD command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1ER400/2000 (1ER1/5) Axis 1 Set encoder gearbox ratio to 1:5 - i.e. for every 5 steps of the position encoder the Actual Position will change by 1 steps. This may be used if for example a 400 step/rev motor is fitted with a 2000step/rev encoder.

ES END SEQUENCE DEFINITION

This command will end a sequence definition. The sequence definition must have been started by the Define Sequence (**DS**) command. No argument is necessary as the sequence number is specified with the Define Sequence (**DS**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>ES	N/A	N/A		N/A	5

Condition Requirements
Define Sequence

Notes:

Responses

OK	Command has been accepted.
! ILLEGAL INSTRUCTION	ES attempted when NOT already defining a sequence.

Example:

1DS2	Start definition of sequence 2.
1MR400	First move (relative).
1MR-400	Next move (relative).
1XS5	Execute sequence 5 (transfer control to start of this sequence 5).
1ES	End of sequence definition.

GA GEARBOX ABSOLUTE MODE

This command will enter gearbox mode when the value of the Input (master) encoder is equal to the value of the Actual Position (slave) encoder. The slave motor will then be driven at a ratio of the Input encoder speed. The ratio is specified by the gear ratio command **GR**. Gearbox mode is exited by a **ST** command, **ESCAPE** or Control **C**.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>GA	N/A	N/A		N/A	1

Condition Requirements
Servo mode, Idle.

Notes:
In absolute gearbox mode SA, SD and SV values are not active.

Responses:

OK	Command has been accepted.
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1GA Axis 1 enter absolute gearbox ratio mode.

GB GEARBOX MODE

Enter gearbox mode. The slave motor is now driven at a ratio of the Input encoder speed. The ratio is specified by the gear ratio command **GR**. Gearbox mode is exited by an **ST** command, **ESCAPE** or Control **C**.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>GB	N/A	N/A		N/A	1

Condition Requirements
Servo mode, Idle.

Notes:
In gearbox mode, SA, SD and SV values are not active.

Responses:

OK	Command has been accepted.
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1GB Axis 1 enter gearbox ratio mode.

GD GEARBOX RATIO DENOMINATOR

Set gearbox ratio denominator. This command may be used in conjunction with GN gearbox numerator. The ratio is therefore specified as a fraction with the format: numerator(**GN**)/(**GD**)denominator. The **GR** gearbox ratio command may also be used to set the ratio.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>GDnnn	N/A	1	32767 (2^{15})	1	4

Condition Requirements
Idle or Gearbox.

Notes:
Value stored in FLASH by BD command.
GR is Not available in sequence definition – use GN & GD.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1GN2 1GD5 Axis 1 Set electronic gearbox ratio to 2:5 - i.e. for every 5 steps of the input encoder the command position will change by 2 steps.

GM GEARBOX MOVE RELATIVE

When in gearbox mode, this command may be used to superimpose a relative move on top of the gearbox slaving. This way, a correction in the synchronism of the two positions may be changed without exiting the gearbox mode. This move is presently done at the creep speed.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>GMnnn	Steps	-2147483647	2147483647($\pm 2^{32}$)	0	1

Condition Requirements
Servo mode, Gearbox mode.

Notes:

Responses:

OK	Command has been accepted.
! NOT ALLOWED IN THIS MODE	Not in Gearbox mode
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1GM100 Superimpose a move of 100 steps (posative) on top of gearbox ratio mode.

GN GEARBOX RATIO NUMERATOR

Set gearbox ratio numerator. This command may be used in conjunction with GD gearbox denominator. The ratio is therefore specified as a fraction with the format: numerator(**GN**)/(**GD**)denominator. The **GR** gearbox ratio command may also be used to set the ratio.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>GNnnn	N/A	-32768	32767 ($\pm 2^{15}$)	1	4

Condition Requirements
Idle or Gearbox.

Notes:
Value stored in FLASH by BD command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1GN2 1GD5 Axis 1 Set electronic gearbox ratio to 2:5 - i.e. for every 5 steps of the input encoder the command position will change by 2 steps.

GR GEARBOX RATIO

Set gearbox ratio. In gearbox modes the ratio is specified by two arguments separated by a **/** character. The ratio is therefore specified as a fraction with the format: numerator**/**denominator. This ratio is also used for input encoder jog scaling.

Syntax	Units	Range	to	Initial State	Privilege level

<ad>GRnnn/nnn				4
Numerator	N/A	-32768	32767 ($\pm 2^{15}$)	1
Denominator	N/A	1	32767 (2^{15})	1

Condition Requirements
Idle or Gearbox.

Notes:
Value stored in FLASH by BD command.
Not available in sequence definition – use GN & GD.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1GR2/5 Axis 1 Set electronic gearbox ratio to 2:5 - i.e. for every 5 steps of the input encoder the command position will change by 2 steps.

HD HOME TO DATUM

This command is used to find a datum point of a mechanism.
Refer to the Datum Search section of this manual and the DATUM MODE **DM** command for details on datum search use.
The **HD-1** command will perform the search in the negative direction.
Soft limits are **not** used during a Home to Datum search.
N/A

Syntax	Units	Range	to	Initial State	Privilege level
<ad>HDn	N/A	-ve	+ve	N/A	3

Condition Requirements
Idle.

Notes:

Responses:

OK	Command has been accepted.
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Soft limit for required direction has already been reached
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected

Examples:

1HD	Search for datum point of axis 1 in positive direction.
1HD-1	Search for datum point of axis 1 in negative direction.

HE, HM, HN, HP HELP

Help pages. The commands **HE** (first help page) and **HN** (help next) and **HP** (help previous) return pages showing Digiloop commands. These help pages give a concise list of the commands available and their function. It also shows the privilege level required to perform the command. **HM** (help modes) give a indication of the bit patterns used in the Abort Mode (AM), Datum Mode (DM) and Jog Mode (JM) commands

Syntax	Units	Range	to	Initial State	Privilege level
<ad>HE	N/A	N/A		N/A	0
<ad>HN	N/A	N/A		N/A	0
<ad>HP	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None.

Responses:

HE

AB <0> Abort
AD <4> Disable auto-execute
AE <4> Enable auto-execute
BA <4> Backup all
BC <4> Backup cams
BP <4> Backup profiles
BD <4> Backup Digiloop parameters
BS <4> Backup sequences
CD <0> Clear motor datum
CP <position> <1> Command position
CR <steps> <1> Set creep steps
CV <speed> <0> Constant velocity mode
DC <cam number> <3> Define cam
DL <2> Disable soft limits
DP <profile number> <3> Define profile
DS <sequence number> <3> Define sequence
EC <0> End cam
EL <2> Enable soft limits
EP <0> End profile
ES <0> End sequence
GA <0> Absolute gearbox mode
GB <0> Relative gearbox mode
For more help pages type: HN for next page; HP for previous page

Example:

1HE Show the first help page of the controller of axis 1.

ID IDENTIFY

This command is used to give the type of controller and its internal software revision.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>ID	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Response:

Mclennan Digiloop Motor Controller V2.21a(0.4)

Example:

1ID Identify controller of axis 1.
01:Mclennan Digiloop Motor Controller V3.25a(0.5)

IF IF FALSE DO NEXT COMMAND

This command will examine the read port inputs and compare them with the specified bit pattern argument. If the inputs are NOT equal to the specified bit pattern (false), then the controller will execute the next command it receives, in its buffer or in its sequence. If the bit pattern IS equal (true) then the controller will skip over, i.e. not execute the next command. If the next command is skipped, the controller will give the response **'*SKIPPED*'** instead of **'*OK*'** or any other response for that command.
The bit pattern is specified as a eight digit binary number of either **0, 1** or **2** characters starting with read port 8, through to 1. A **0** defines that the input must be low (**0**), a **1** defines that the input must be high (**1**) and a **2** defines that the input is not relevant or 'don't care'. If less that eight digits are specified in the argument, then the preceding ones are assumed as low (**0**).
This command may be used to introduce a conditional response to some machine functions, and may be used to create 'clever' sequences. See also the If True (**IT**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>IFbbbbbbb	Bits	0	2	N/A	1

Condition Requirements
None

Notes:

Response:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

This following sequence has been constructed to repeat a loop of moving in 400 step intervals, until read port 4 goes high (possibly activated by a switch).

1DS3	Start sequence definition
1MR400	Move 400 steps
1IF22221222	This condition is FALSE so next command is executed (i.e. NOT skipped).
1XS3	Condition was FALSE; therefore execute this sequence i.e. repeat this loop
(**1XS0)**	Return to main or another sequence (optional).
1ES	End sequence

The sequence starts by moving 400 steps. The **IF** command will then compare with the readports, in this case it is only bit 4 that is relevant. If the condition is FALSE (the switch is not on), then the next command is executed which will restart the current sequence of move 400 steps and compare. If the condition becomes TRUE (the switch goes on), then the **XS3** command will be skipped and go on to the one after. This could be the end of the sequence or a command to then do another sequence such as returning to a master sequence.

IN INITALISE

This command will set all the programmable parameters back to their initial values, clear sequences and profiles. This is used to re-initialise all the volatile memory values to 'safe' values - e.g. if the controller was to be used in a new application. A **BA** command is required to then write these values into the *flash* non-volatile memory.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>IN	N/A	N/A		N/A	8

Condition Requirements
Idle.

Notes:

Response:

OK	Command has been accepted.

Example:

1IN	Set all parameters on axis 1 back to their initial values.

IP SET INPUT POSITION

Set the Input Encoder position value to that given in the argument.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>IPnnn	N/A	-2147483647	2147483647($\pm 2^{32}$)	N/A	3

Condition Requirements
Idle, Constant velocity or Gearbox

Notes:
Value zero on power-up.

Response:

OK Command has been accepted.

Examples:

1IP5000 Set the axis 1 Input Encoder Position to 5000.

IT IF TRUE DO NEXT COMMAND

This command will examine the read port inputs and compare them with the specified bit pattern argument. If the inputs are equal to the specified bit pattern (true), then the controller will execute the next command it receives, in its buffer or in its sequence. If the bit pattern is NOT equal (false) then the controller will skip over, i.e. not execute the next command. If the next command is skipped, the controller will give the response **'*SKIPPED*'** instead of **'*OK*'** or any other response for that command.
The bit pattern is specified as a eight digit binary number of either **0, 1** or **2** characters starting with read port 8, through to 1. A **0** defines that the input must be low (**0**), a **1** defines that the input must be high (**1**) and a **2** defines that the input is not relevant or 'don't care'. If less that eight digits are specified in the argument, then the preceding ones are assumed as low (**0**).
This command may be used to introduce a conditional response to some machine functions, and may be used to create 'clever' sequences. See also the If False (**IF**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>ITbbbbbbb	Bits	0	2	N/A	1

Condition Requirements
None

Notes:

Response:

OK Command has been accepted.
! OUT OF RANGE Argument is out of valid range.

Example:

If the following states are present on the inputs:

PORT :	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
STATE :	High	Low	Low	High	High	Low	Low	Low

1IT22222200 This condition is TRUE so next command is executed (i.e. NOT skipped).
1MR200 Move 200 steps
1IT22222201 This condition is FALSE so next command is skipped (i.e. is NOT executed).
1MR400 This command is skipped

If the following states are present on the inputs:

PORT :	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
STATE :	High	Low	Low	High	High	Low	Low	High

1IT22222200 This condition is FALSE so next command is skipped (i.e. is NOT executed).
1MR200 This command is skipped
1IT22222201 This condition is TRUE so next command is executed (i.e. NOT skipped).
1MR400 Move 400 steps

I.E. In the above example, read port 1 is used to select a move lengthand read port 2 will disable the move:

PORT :	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	
STATE :	(Ignored)	(Ignored)	(Ignored)	(Ignored)	(Ignored)	(Ignored)	Low	Low	Move 200 steps
STATE :	(Ignored)	(Ignored)	(Ignored)	(Ignored)	(Ignored)	(Ignored)	Low	High	Move 400 steps

JM SET JOG MODE

Set the mode of operation for jog switch, joystick and encoder jog moves. There are three methods of commanding JOG or manual moves:
Jog switch inputs using fast (**SF**) and slow jog speeds (**SJ**) such as with a jog box.
Joystick input measures an analogue voltage such as from an analogue joystick or potentiometer
Input encoder jog uses the quadrature signals on the Input encoder such as with a trackersball

Syntax	Units	Range	to	Initial State	Privilege level
<ad>JMabcdefgh	Bits	0	1	10000000	6

where
- **a** - 0 – Jog switch inputs disabled
  1 – Jog switch inputs enabled
- **b** - 0 – Joystick input disabled
  1 – Joystick input enabled
- **c** - 0 – Input encoder jog input disabled
  1 – Input encoder jog input enabled
- **d** - Reserved for future use.
- **e** - Reserved for future use.
- **f** - Reserved for future use.
- **g** - Reserved for future use.
- **h** - Reserved for future use.

Condition Requirements **Notes:**
Idle

Response:

OK	Command has been accepted.
! OUT OF RANGE	Illegal argument range or format

Examples:
1JM01000000 Set axis 1 to use Joystick only (jog switches and Input encoder jog disabled).

JS SET JOYSTICK SPEED

Set the normal speed for all following manual joystick moves. This speed is that if the joystick output was at 10% of full swing. Please note that few joysticks can achieve full voltage swing and therefore this sets the speed for full movement of the joystick.

Syntax	Units	Range	to	Initial State	Privilege level
<ad> JSnnn	Steps/sec	1	400000	10000	4

Condition Requirements **Notes:**
Idle. Value stored in FLASH by BD command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:
1JS2000 Sets jog speed of axis 1 controller to 2000 Steps/sec.

KF SET FEEDFORWARD COEFFICIENT

Set velocity feedforward servo coefficient. This compensates for the position offset caused by the velocity lag introduced by **KV**. For positioning moves **KF** is normally set at zero, but for Profiles and Cam moves where the actual position should not lag behind the command position, **KF** should be set equal to **KV**.
In dual encoder feedback systems **KX** also causes a velocity lag. The value of complete **KF** compensation needed is equal to **KX** multiplied by the ratio of Auxiliary encoder pulses to Position Encoder pulses plus the value of **KV**. It is not usually necessary for complete compensation of the velocity lag as this adversely effects the settling time of the system.

This command is only appropriate in the servo motor control mode (not for stepper motor control)

Syntax	Units	Range	to	Initial State	Privilege level
<ad>KFnnn	Number	0	32767	0	7

Condition Requirements
Servo mode, Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Examples:

1KF500 Set velocity feedforward on axis 1 to 500.

KP (IN SERVO MODE) SET PROPORTIONAL GAIN COEFFICIENT

Set proportional gain servo coefficient. The stiffness and accuracy of the servo loop are controlled by the magnitude of the proportional gain.
See next section for use in stepper motor mode

Syntax	Units	Range	to	Initial State	Privilege level
<ad>KPnnn	Number	0	32767	10	7

Condition Requirements
Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Examples:

1KP100 Set the proportional gain on axis 1 to 100.

KP (IN STEPPER MODE) SET PROPORTIONAL GAIN COEFFICIENT (CORRECTION GAIN)

Set proportional gain for auto-correction moves. . The amount of attempted correction for each iteration is the difference between the Command Position and the encoder read Actual Position, scaled by this value. If the result is less than one step then it will use one step of correction. Each successful iteration, separated by settling time, should therefore result in less of an error and the next attempt will be less. Error correction will continue until the error is within the window (**WI**) or the Time Out (**TO**) period has expired.
Care must be taken with remote feedback encoders, not to set too higher value that might give rise to an oscillatory system.
See previous section for use in servo mode

Syntax	Units	Range	to	Initial State	Privilege level
<ad>KPnnn	%	0	100	70	7

Condition Requirements
Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Examples:

1KP100 Set the proportional gain on axis 1 to 100.

KS SET SUM GAIN COEFFICIENT

The Sum servo coefficient is the sum of the integral and proportional components of the servo control loop. The accuracy of the servo loop depends on having a non-zero value of **KS** at the expense of transient response. This command is only appropriate in the servo motor control mode (not for stepper motor control)

Syntax	Units	Range	to	Initial State	Privilege level
<ad>KSnnn	Number	0	32767	0	7

Condition Requirements
Servo mode, Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Examples:
1KS50 Set the Sum gain on axis 1 to 50.

KV SET VELOCITY FEEDBACK COEFFICIENT

The value of this coefficient defines the magnitude of the velocity feedback signal derived from the position encoder. This coefficient influences the transient response by producing a damping effect. It effects the system by reducing overshoot and enhancing stability, but too high a value can create a *buzzy* system, and ultimately an unstable system.
This command is only appropriate in the servo motor control mode (not for stepper motor control)

Syntax	Units	Range	to	Initial State	Privilege level
<ad>KVnnn	Number	0	32767	0	7

Condition Requirements
Servo mode, Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Examples:
1KV500 Set the Velocity feedback on axis 1 to 500.

KX SET EXTRA VELOCITY FEEDBACK COEFFICIENT

The Extra Velocity Feedback coefficient. It is used in Dual Encoder feedback mode. The value of this coefficient defines the magnitude of the velocity feedback signal derived from the auxiliary (third) encoder. This coefficient influences the system transient response by producing a damping effect.
This command is only appropriate in the servo motor control mode (not for stepper motor control)

Syntax	Units	Range	to	Initial State	Privilege level
<ad>KXnnn	Number	0	32767	0	7

Condition Requirements
Servo mode, Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Examples:

1KX5000 Set the Extra Velocity feedback on axis 1 to 5000.

LC LIST CAM

This command will list a previously defined Cam profile.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>LCn	Cam Number	0	7	N/A	0

Condition Requirements **Notes:**
Servo mode, Idle

Responses

The command will either respond with the axis address identifier followed by each line of the Cam definition Profile, or an error message:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! CAM UNDEFINED	Cam has not been defined yet.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

A controller that had previously been programmed with:

1DC3	Start Cam 3 definition.
1XY200/500	First Cam Profile point.
1XY400/500	Next Cam Profile point.
1XY600/-200	Next Cam Profile point.
1XY700/-200	Next Cam Profile point.
1XY750/0	Next Cam Profile point.
1EC	End of Cam profile definition.

The command 1LC3 would give:

01:Cam 3:
XY 0 / 0
XY 200 / 500
XY 400 / 500
XY 600 / -200
XY 700 / -200
XY 750 / 0

LD SET LIMIT DECELERATION

Set the deceleration rate for stopping when hitting a Hard Limit or a Soft Limit. It is also used during a soft stop such as after an ESCAPE. This value would normally be set to a high value to prevent limit overrun, but may be used to reduce the harshness of stopping on a limit. This should also be used to prevent stepper motors desynchronising and losing steps through excessive deceleration.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>LDnnn	Steps/sec^2	1	20000000	2000000 (servo) 50000 (stepper)	4

Condition Requirements **Notes:**

Idle	Value stored in FLASH by BD command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1LD1000000 Sets Limit Deceleration of axis 1 controller to 1000000 Steps/sec^2.

LL SET LOWER SOFT LIMIT POSITION

This command will set the Lower Soft Limit Position to the value given in the argument. Subsequent moves by the Move Absolute (**MA**) or Move Relative (**MR**), constant velocity and manual Jog moves will not be allowed below this Lower Limit if the Soft Limits are enabled.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>LLnnn	Steps	-2147483647	2147483647 (±2^{32})	-2000000000	3

Condition Requirements	Notes:
Idle	Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! LIMITS CONFLICT	Attempting to set lower limit above or equal to upper limit

Example:

1LL-4000 Set the axis 1 Lower Soft Limit Position to -4000.

LP LIST PROFILE

This command will list a previously defined velocity Profile.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>LPn	Profile No.	0	7	N/A	0

Condition Requirements	Notes:
Servo mode, Idle	

Responses

The command will either respond with the axis address identifier followed by each line of the Profile definition, or an error message:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! PROFILE UNDEFINED	Profile has not been defined yet.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

A controller that had previously been programmed with:

1DP6	Start definition of Profile.
1MR2000	First move.
1MR7000	Next move.
1MR1000	Next move.
1MR0	Next move.
1EP	End of Profile definition.

The command 1LP6 would give:

01:Profile 6:

MR 2000
MR 7000
MR 1000
MR 0

LS LIST SEQUENCE

This command will list a previously defined Sequence.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>LSn	Seq Number	0	7	N/A	0

Condition Requirements **Notes:**
Idle

Responses

The command will either respond with the axis address identifier followed by each line of the Sequence definition, or an error message:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.
! SEQUENCE UNDEFINED	Sequence has not been defined yet.

Example:

A controller that had previously been programmed with:

1DS2	Start definition of sequence 2.
1MA2000	First move (absolute).
1MR7000	Next move (relative).
1DE1000	Delay for 1 second.
1MA0	Next move (return to start position).
1XS2	Execute sequence 2 (loop to start of this sequence).
1ES	End of sequence definition.

The command 1LS2 would give:

01:Sequence 2:
MA 2000
MR 7000
DE 1000
MA 0
XS 2

MA MOVE TO ABSOLUTE POSITION

This command will move the motor to the position given in the argument. This position is relative to the Command Position of zero.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>MAnnn	Steps	-2147483647	2147483647 ($\pm 2^{32}$)	N/A	1

Condition Requirements **Notes:**
Idle

Responses

OK	Command has been accepted.
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Move attempted that exceeds the Soft limit in the required direction
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected

! TIMEOUT ABORT A timeout abort has been detected

Example:
If axis 1 has a current Command Position of 5000 then the command:
1MA4000 Will move 1000 steps in the negative direction to arrive at a Command position of 4000.

MD MOVE TO DATUM POSITION

This command will move the motor to the datum position if one has already been captured.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>MD	N/A	N/A		N/A	1

Condition Requirements **Notes:**
Idle, Valid captured datum.

Responses

OK	Command has been accepted.
! NO VALID DATUM	A datum point has not yet been found
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Move attempted that exceeds the Soft limit in the required direction
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected

Example:
If axis 1 has a current valid Datum Position of 12496 then the command:
1MD Will move to the position of 12496.

MR MOVE TO RELATIVE POSITION

This command will move the motor to the position given in the argument relative to the current Command Position.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>MRnnn	Steps	-2147483647	2147483647 ($\pm 2^{32}$)	N/A	1

Condition Requirements **Notes:**
Idle

Responses

OK	Command has been accepted.
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Move attempted that exceeds the Soft limit in the required direction
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected

Example:
If axis 1 has a current Command Position of 5000 then the command:
1MR4000 Will move 4000 steps in the positive direction to arrive at a Command position of 9000.

NP SET NEW PIN SECURITY NUMBER

Set the PIN security number to that given in the argument. You must enter the existing PIN using the PI command first.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>NPnnn	Number	0000	9999	9999	9

Condition Requirements **Notes:**
Idle

Response:
OK Command has been accepted.

Examples:
1PI0001 Enter existing PIN number.
1NP0666 Set the axis 1 PIN number to 0666.

OA OUTPUT ACTUAL POSITION

This command will give the current Actual Position read from the position encoder. This position is derived from the incoming position encoder pulses (scaled by the encoder ratio (**ER**).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OA	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Response:
The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by **Actual pos =**

Example:
If the controller of axis 1 currently has an Actual Position of 70551 then the command:
1OA in Verbose Mode will respond: **01:Actual pos = 70551**
1OA in Quiet Mode will respond: **01:70551**

OC OUTPUT COMMAND POSITION

This command will give the current Command Position.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OC	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None.

Response:
The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by **Command pos =**

Example:
If the controller of axis 1 currently has a Command Position of 45280 then the command:
1OC in Verbose Mode will respond: **01:Command pos = 45280**
1OC in Quiet Mode will respond: **01:45280**

OD CAPTURED DATUM POSITION

This command will give the current captured datum position.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OD	N/A	N/A		N/A	0

Condition Requirements
None.

Notes:

Response:
The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by **Datum position =**
! NO VALID DATUM if no datum has been captured.

Example:
If the controller of axis 1 currently has a datum position of 28456 then the command:
1OD in Verbose Mode will respond: **01:Datum position = 28456**
1OD in Quiet Mode will respond: **01:28456**

OF OUTPUT FOLLOWING ERROR BETWEEN COMMAND AND ACTUAL POSITIONS

This command will give the difference between the current Command Position and the current encoder read Actual Position. Numerically it is the Command Position (**CP**) - Actual Position (**AP**).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OF	N/A	N/A		N/A	0

Condition Requirements
None.

Notes:

Response:
The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by **Following error =**

Example:
If the controller of axis 1 currently has a Current position of 1000 and an Actual Position of 1050 then the command:
1OF in Verbose Mode will respond: **01:Following error = -50**
1OF in Quiet Mode will respond: **01:-50**

OI OUTPUT INPUT POSITION

This command will give the current encoder read Input Position. This position is derived from the incoming Input encoder pulses.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OI	N/A	N/A		N/A	0

Condition Requirements
None

Notes:
Used in *electronic gearbox* and *Cam profiles* etc.

Responses:
The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by **Input pos =**

Example:

If the controller of axis 1 currently has an Input Position of 30401 then the command:

1OI in Verbose Mode will respond: **01:Input pos = 30401**

1OI in Quiet Mode will respond: **01:30401**

OS OUTPUT STATUS

This command will give a binary string that will represent the current status of the controller.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OS	N/A	N/A		N/A	0

Condition Requirements **Notes:**

None

Response:

The response is a string of 8 numeric characters of either 0 or 1. If the communications are in Verbose Mode, the reply is preceded by **Status = .** Each bit is described as follows:

Status = abcdefgh where:

a - 0 – Controller is busy (doing something)
1 – Controller is idle

b - 0 – OK
1 – Error (abort, tracking, stall, timeout etc.)

c - 0 – Upper hard limit is OK
1 – Upper hard limit is ON

d - 0 – Lower hard limit is OK
1 – Lower hard limit is ON

e - 0 – Not jogging or joystick moving
1 – Jogging or joystick moving

f - 0 – Not at datum sensor point
1 – On datum sensor point

g - 0 – For future use
1 – For future use

h - 0 – For future use
1 – For future use

Example:

If the controller of axis 1 is currently moving to a position (using a **MA** command):

1OS in Verbose Mode will respond: **01:Status = 00000000**

1OS in Quiet Mode will respond: **01:00000000**

If the controller of axis 1 is currently stopped on the upper hard limit:

1OS in Verbose Mode will respond: **01:Status = 10100000**

1OS in Quiet Mode will respond: **01:10100000**

OT OUTPUT (THIRD) AUXILIARY POSITION

This command will give the current encoder read Auxiliary Position. This position is derived from the incoming position encoder pulses of the second encoder (dual encoder feedback).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OT	N/A	N/A		N/A	0

Condition Requirements **Notes:**

None

Response:

The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by **Auxiliary pos =**

Example:

If the controller of axis 1 currently has an Auxiliary Position of 20501 then the command:

1OT in Verbose Mode will respond: **01:Auxiliary pos = 20501**

1OT in Quiet Mode will respond: **01:20501**

OV OUTPUT VELOCITY

This command will give the current velocity of the Actual Position (position encoder), unless in open loop stepper mode where the velocity of the command position is used. This value is averaged over the time given in the argument in milliseconds. You would therefore choose a time to give the accuracy you require at the expense of the time to complete the command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>OVnnn	mS	1	1000	N/A	0

Condition Requirements **Notes:**

None

Response:

The response is a string of numeric characters. If the communications are in Verbose Mode, the reply is preceded by **Velocity =**

Example:

If the controller of axis 1 currently at 20000 steps per second but instantaneously currently very slightly lagging then:

1OV250 in Verbose Mode will respond: **01:Velocity = 19994**

1OV250 in Quiet Mode will respond: **01:19994**

Notice that as the average time is a quarter of a second, then the speed is a multiple of four.

PI ENTER PIN NUMBER

This command allows you to enter the PIN security number. This then allows you to change the Privilege level, using the privilege level (**PL**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>PInnnn	N/A	0000	9999	N/A	0

Condition Requirements **Notes:**

None

Response:

OK Command has been accepted.

Example:

If the controller of axis 1 currently has a security PIN number of 4423

1PI4423 will allow the privalege level to be changed.

PL SET PRIVALEGE LEVEL

Set the privilege level. This command allows you to set a privilege level that will restrict the commands available to the user. This may be used to prevent accidental changing of important set-up parameters.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>PLn	N/A	0	9	8	0

Condition Requirements **Notes:**
The PIN number must have been entered (PI) Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1PL1 Sets the privilege level to 1 (queries and moves only).

PT PROFILE TIME

This command allows you to enter the time to complete each element in a profile definition.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>PTnnn	mS	0	32000	1000	4

Condition Requirements **Notes:**
Servo mode, Idle

Response:

OK	Command has been accepted.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1PT50 will set the time for each element of the profile to be 50 mS

QA QUERY ALL PARAMETERS

Query All. Returns all of the current settings and modes of the controller along with the current positions in a single page format.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>QA	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Response:

The response is alpha-numeric strings of characters. Each line gives the parameter names and their values. See example for the format.

Example:

1QA Will generate a response of the form:

```
Mclennan Digiloop Motor Controller V2.10a(1.2)
Address = 1
Privilege level = 4
Mode = Aborted
Kf = 1000     Kp = 500      Ks = 2000     Kv = 1000     Kx = 0
Slew speed = 100000
Acceleration = 200000          Deceleration = 400000
Creep speed = 400              Creep steps = 0
Jog speed = 100                Joystick speed = 10000
Settling time = 200
Window = 4                     Threshold = 2000
```

Tracking = 4000
Lower soft limit = -2147483647 Upper soft limit = 2147483647
Soft limits enabled
Lower hard limit on Upper hard limit on
Jog enabled Joystick disabled
Gbox num = 1 Gbox den = 1
Command pos = 0 Motor pos = 1
Pos error = -1 Input pos = 0
Valid sequences: none Autoexec disabled
Valid cams: none
Valid profiles: none Profile time = 1000 ms
Read port: %00000000 Last write: %00000000

QK QUERY K COEFFICIENTS

Query servo loop coefficients. Returns the current settings of the KP, KS, KV, KF and KX coefficients.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>QK	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Response:
The response is an alpha-numeric string of characters showing the parameter name and its value. See example for the format.

Example:

1IN	Set to initial values.
1KP2909	Set proportional gain to 2909.
1KV357	Set velocity feedback to 357.
1KS3258	Set Sum coefficient to 3258.
1QK	Will generate a response of the form:

KP=2909,KS=3258,KV=357,KF=0,KX=0

QL QUERY CURRENT PRIVALEGE LEVEL

This command will give the current privilege level. The higher the level, the more commands you can use.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>QL	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Responses:
The response is a numeric character.

Example:
If the controller of axis 1 currently has a privilege level of, then the command:
1QL will respond: **01:Privilege level = 6**

QM QUERY MODES

This command will give the current Control Mode (**CM**), Abort Mode (**AM**), Datum Mode (**DM**), Jog Mode (**JM**).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>QM	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Responses:
The response is the axis address identifier, followed by the following (see example for format):
CM = currently set control mode (in decimal)
AM = currently set abort mode (as binary bit pattern)
DM = currently set datum mode (as binary bit pattern)
JM = currently set jog mode (as binary bit pattern)

Example:
If the controller of axis 1 is set to servo motor controller
1QM may give a response of:
01:CM = 1 AM = 00000000 DM = 00010000 JM = 11000000

QP QUERY POSITIONS

Query the current position information. Returns the current values for Command Position (CP), Actual Position (AP), Input (IP) Position Auxiliary Position (TP) and Datum Position (OD)

Syntax	Units	Range	to	Initial State	Privilege level
<ad>QP	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Response:
The response is an alpha-numeric string of characters showing all the current position variables. See example for the format.

Example:
1QP May generate a response of the form:
01:CP = -1026 AP = -1026 IP = 1050 TP = 0 OD = -2050

QS QUERY SPEEDS

Query the current settings for the speeds and accelerations. Returns the current settings of SV, SC, SA,SD and LD.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>QS	N/A	N/A		N/A	0

Condition Requirements **Notes:**
None

Response:
The response is an alpha-numeric string of characters showing all the speed related variables. See example for the format.

Example:
1SC700 Set creep speed to 700 steps/sec.
1SV16200 Set slew speed to 16200 steps/sec.
1SA50000 Set deceleration to 50,000 steps/sec^2.
1SD100000 Set deceleration to 100,000 steps/sec^2.

1SD200000 Set limit deceleration to 200,000 steps/sec^2.
1QS Will generate a response of the form:

01:SC = 700 SV = 16200 SA = 50000 SD = 100000 LD = 200000

RP READ INPUT PORT

This command will examine the read port inputs and return their current state as an eight digit numeric string of either **0** or **1** characters. The string starts with read port 8. A **0** indicates that the input is low (0V or open-circuit) and a **1** indicates that the input is high (+24V).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>RP	N/A	N/A		N/A	0

Condition Requirements
None

Notes:
If an **RP** command is executed with the read ports open circuit, a reply of **00000000** will be returned

Responses

The response is a string of 8 numeric characters of either 0 or 1. If the communications are in Verbose Mode, the reply is preceded by Port: **.**

Example:

If the following states are present on the inputs:

PORT :	8	7	6	5	4	3	2	1
STATE :	LOW	LOW	LOW	HIGH	LOW	LOW	LOW	HIGH

Then

RP in Verbose Mode will respond: **01:Port: 00010001**
1RP in Quiet Mode will respond: **01: 00010001**

RS RESET

This command will reset the *tracking abort, stall abort, time out abort* or *user(command) abort* conditions and re-enable the servo control loop. It will also set the Command position to be equal to the Actual position (except open loop stepper mode).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>RE	N/A	N/A		N/A	3

Condition Requirements
None

Notes:

Responses

RESET or ! NOT ABORTED Command has been accepted
! INPUT ABORT Stop Input still active.

Example:

1RS Reset abort on axis 1 controller.

SA SET ACCELERATION

Set the acceleration rate for changes of velocity for all following moves.

Syntax	Units	Range	to	Initial State	Privilege level
< ad>SAnnn	Steps/sec^2	1	20000000	2000	4

Condition Requirements

Notes:

Idle	Value stored in FLASH by BD command.

Condition Requirements **Notes:**

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1SA10000 Sets acceleration of axis 1 controller to 10000 Steps/sec^2.

SC SET CREEP SPEED

Set the creep speed for all following moves. This is the speed that at which moves with a non-zero creep distance will stop. It is also the speed that slow datum search will be moved at (**HD** command).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>SCnnn	Steps/sec	1	400000	800	4

Condition Requirements **Notes:**

Idle Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1SC700 Sets creep speed of axis 1 controller to 700 Steps/sec.

SD SET DECELERATION

Set the deceleration rate for changes of velocity for all following moves.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>SDnnn	Steps/sec^2	1	20000000	3000	4

Condition Requirements **Notes:**

Idle Value stored in FLASH by BD command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1SD100000 Sets deceleration of axis 1 controller to 100000 Steps/sec^2.

SE SET SETTLING TIME

Set the settling time for all following moves. This time elapses at the end of each move to allow the motor to settle. The end of a move is defined by the **OF** (following error or position difference) value being less than the **WI** (end of move window) value for the **SE** (settling) time. If the following error exceeds the window, then the settling counter will be reset and therefore it must be within the window for the whole length of the settling time.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>SEnnn	milliseconds	0	20000	100	6

Condition Requirements
Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1SE1000 Sets settling time of axis 1 controller to 1 second.

SF SET FAST JOG SPEED

Set the fast speed for all following manual *jog switch* moves. The jog movement will accelerate up to this speed when a jog input and the jog fast inputs are active.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>SFnnn	Steps/sec	1	200000	500	4

Condition Requirements
Idle.

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1JF1000 Sets fast jog speed of axis 1 controller to 1000 Steps/sec.

SH SET HOME POSITION

Set the Home position value to that given in the argument. The Home Position may be used during a datum search to automatically set the datum point to the given value, when using the Home to Datum (**HD**) command, if the correct Datum Mode is set (see Datum Search section and **DM** command).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>SHnnn	Steps	-2147483647	2147483647($\pm 2^{32}$)	0	6

Condition Requirements
Idle.

Notes:
Value stored in FLASH by BD command.

Response:

OK	Command has been accepted.

Examples:

1SH-34277 Set the axis 1 Home Position to -34277.

SJ SET JOG SPEED

Set the normal speed for all following manual *jog switch* moves. The jog movement will be at this speed when a jog input is active, but not the jog fast input.

Syntax	Units	Range	to	Initial State	Privilege level
<ad> SJnnn	Steps/sec	1	20000	100	4

Condition Requirements
None.

Notes:
Value stored in FLASH by BD command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1SJ50 Sets jog speed of axis 1 controller to 50 Steps/sec.

SL ENABLE/DISABLE SOFT LIMITS

This command is used to enable or disable the soft limit protection. If the soft limits are disabled, further movement is NOT bounded by the upper and lower soft limits. Hard limits will still be active and cannot be disabled.

Syntax	Units	Range	to	Initial Value	Privilege level
<ad>SLb	N/A			Enabled (1)	6

where **b** - 0 –Soft limits disabled
1 – Soft limits enabled

Condition Requirements
Idle.

Notes:
Value stored in FLASH by BD command.

Response:

OK	Command has been accepted.

Example:

1SL0 Sets the soft limits OFF (disabled) for controller axis 1.
1SL1 Sets the soft limits ON (enabled) for controller axis 1.

ST STOP

This command will stop any current move, decelerate the motor speed down at the **SD** rate, then stop and return to *idle* mode.
This command is buffered and is only responded to when it reached in the command queue. Care must therefore be taken that there are no commands that hold up the queue between the move command and the **ST** command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>ST	N/A	N/A		N/A	0

Condition Requirements
Not Idle

Notes:
Will exit constant velocity mode or gearbox mode.

Responses

OK	Command has been accepted.
! NOT ALLOWED IN THIS MODE	The controller is already stopped (idle).

Example:

1CV1000 Will start axis 1 moving in constant velocity mode (1000 steps/sec).
1ST This will then stop the current move of axis 1.

SV SET VELOCITY

Set the Slew (maximum) velocity for all following moves.

Syntax	Units	Range	to	Initial State	Privilege level

<ad> SVnnn	Steps/sec	1	400000	1000	4

Condition Requirements
Idle

Notes:
Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1SV5000	Sets slew speed of axis 1 controller to 5000 Steps/sec.

TH SET THRESHOLD

This command will set the motor stalled threshold. Failure of an encoder is indistinguishable from a stalled motor, and messages from the PM600 refer to *stall abort* rather than encoder failure.
A stalled motor (or encoder failure) is detected by looking for changes in the position encoder signals (or equivalently the changes in observed motor position). If the motor does not move, and the voltage output value from the PM600 exceeds the value set by the **TH** command for a time of 256ms, then the PM600 will set its output to zero and set Stall Abort condition. The threshold is expressed as a percentage of full scale output of the Analogue output.
The servo system will have coulomb friction and the voltage required to overcome this friction, varies from system to system, so the value of **TH** must be large enough not to nuisance trigger but small enough to detect any failure.
If a *stall abort* condition occurs, the front panel status display shows a **S**, and movement is stopped. Subsequent moves will not function but will return the response **! STALL ABORT** until reset by either a Reset (**RS**) command or by powering off. The stall abort function may be enabled or inhibited by using the **AM** (abort mode) command.
The response to a **CO** command is **! STALL ABORT**.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>THnnn	%	1	100	50	6

Condition Requirements
Idle.

Notes:
Value stored in FLASH by BD command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1TH40	Set the Threshold before *motor stalled* condition for axis 1 to 40%.

TO SET TIME-OUT/NOT COMPLETE TIME

This command will set the Not Complete/Time-Out time. This is the maximum time allowed at the end of a move, from when the Command Position reaches its target, until the move has settled and completed (including auto-correct stepper mode). If the error correction is not completed within this time then a Time Out will be detected and Abort if set using the Abort Mode (**AM**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>TOnnn	milliseconds	1	60000	8000	6

Condition Requirements
Idle.

Notes:
Value stored in FLASH by BD command.

Response:

OK	Command has been accepted.

Examples:

1TO4000 Set the axis 1 Time out to 4 seconds (4000mS).

TP SET (THIRD) AUXILIARY POSITION

Set the Auxiliary (third) Position value to that given in the argument. This position is derived from the incoming position encoder pulses of the second encoder (dual encoder feedback).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>TPnnn	Steps	-2147483647	2147483647($\pm 2^{32}$)	N/A	3

Condition Requirements
Idle, Constant velocity or gearbox

Notes:
Value zero on power-up.

Response:

OK Command has been accepted.

Examples:

1TP-5000 Set the axis 1 Auxiliary Position to -5000.

TR SET TRACKING WINDOW

This command will set the tracking window. The *Tracking window* is the allowable difference between the *Command Position* and the *Actual Position*. When the motor is stationary this is the allowable static error. During a move, a changing *command position* is generated. The *Tracking Window* operates on the difference between the *actual position* and this moving *command position*. The servo system will have a *following error,* so the value of **TR** must be large enough not to nuisance trigger but small enough to detect any failure.
If the *tracking* window is exceeded the front panel display will show a **t**, The Error output signal will be activated and (if abort is enabled) the controller *aborts*.
The abort function may be enabled or inhibited by using the **AM** (abort mode) command. If aborted, subsequent moves will not function but will return the response **! TRACKING ABORT** until reset by either a Reset (**RS**) command or by powering off.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>TRnnn	Steps	0	2147483647 (2^{32})	4000	6

Condition Requirements
Idle

Notes:
Value stored in FLASH by BD command

Responses:

OK Command has been accepted.
! OUT OF RANGE Argument is out of valid range.

Example:

1TR400 Set the Tracking Window for axis 1 to 400 steps.

TUNE TUNE SERVO COEFFICIENTS

An approximate set of servo coefficients can usually be derived by invoking the **TUNE** command. The controller will *exercise* the motor over a small displacement for a few seconds and obtain a set of values for the *K* coefficients that should be stable and provide a reasonable disturbance rejection.
The tuning algorithm may fail if there is excessive backlash, if the low frequency loop gain is either very small or very large or the feedback encoder phasing is wrong. Further optimisation of system response may be required to achieve the desired performance.

The **TUNE** command only affects **KP, KV, KS and KV** therefore its use in a double encoder system is inappropriate and may produce a **! TUNE FAILURE** error.
The TUNE command is only appropriate in the servo motor control mode (not for stepper motor control)

Syntax	Units	Range	to	Initial State	Privilege level
<ad>TUNE	N/A	N/A		N/A	7

Condition Requirements **Notes:**
Servo mode, Idle

Responses:

Reversals = nn Amplitude = mm
n and m are parameters relating to the system response.
These parameters are used by optimisation programs.

OK	Command has been accepted.
! TUNE FAILURE	Auto tuning failed
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Soft limit for required direction has already been reached
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:
1TUNE Tune coefficients on axis 1 controller.

UC UNDEFINE CAM

This command will undefine or cancel a Cam definition. **Note** that this will only remove the cam definition from the volatile memory and to change the non-volatile flash memory this command must be followed by a backup cam (**BC**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>UC	Cam number	0	7	N/A	5

Condition Requirements **Notes:**
Servo mode, Idle. Value stored in FLASH by BC command.

Responses

OK	Command has been accepted.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:
1UC Delete Cam from axis 1 controller.

UL SET UPPER SOFT LIMIT POSITION

This command will set the Upper Soft Limit Position to the value given in the argument. Subsequent moves by the Move Absolute (**MA**), Move Relative (**MR**), Constant Velocity or manual Jog moves will not be allowed above this Upper Limit if the Soft Limits are enabled (see **SL** command).

Syntax	Units	Range	to	Initial State	Privilege level
<ad>ULnnn	Steps	-2147483647	2147483647 ($\pm 2^{32}$)	2000000000	3

Condition Requirements **Notes:**
Idle Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! LIMITS CONFLICT	Attempting to set upper limit below or equal to lower limit

Example:

1UL8000 Set the axis 1 Upper Soft Limit Position to 8000.

UP UNDEFINE PROFILE

This command will undefine or cancel a Profile definition. **Note** that this will only remove the profile definition from the volatile memory and to change the non-volatile flash memory this command must be followed by a backup profile (**BP**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>UP	Profile No.	0	7	N/A	5

Condition Requirements
Servo mode, Idle.

Notes:
Value stored in FLASH by BP command.

Responses:

OK	Command has been accepted.
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1UP Delete Profile from axis 1 controller.

US UNDEFINE SEQUENCE

This command will undefine or cancel a sequence definition. **Note** that this will only remove the sequence definition from the volatile memory and to change the non-volatile flash memory this command must be followed by a backup sequence (**BS**) command.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>US	Sequence No.	0	7	N/A	5

Condition Requirements
Idle.

Notes:
Value stored in FLASH by BS command.

Responses:

OK	Command has been accepted.
! OUT OF RANGE	Argument (sequence number) is out of valid range.

Example:

1US6 Delete sequence 6 from axis 1 controller.

WA WAIT FOR INPUT PORT CONDITION

This command will examine the read port inputs and compare them with the specified bit pattern argument. It will wait until the inputs are equal to the specified bit pattern before issuing its **'OK'** response and moving on to the next command.
The bit pattern is specified as a eight digit binary number of either **0, 1** or **2** characters starting with read port 8, through to 1. A **0** defines that the input must be low (0V or open-circuit), a **1** defines that the input must be high (+24V) and a **2** defines that the input is not relevant or 'don't care'. If less that 8 digits are specified in the argument, then the preceding ones are assumed as low (**0**).

Syntax	Units	Range	to	Initial State	Privilege level

<ad>WAbbbb	Bit pattern	8 digits of 0, 1 or 2	N/A	1

Condition Requirements **Notes:**
None.

Responses

OK	Command has been accepted.
! INVALID BINARY	Invalid argument i.e. bit specified was not 0, 1 or 2 OR the number of bits was greater than 8.

Example:

1WA22112210 Will wait until the following condition is on the read input port before continuing:

PORT :	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
STATE :	(Ignored)	(Ignored)	High	High	(Ignored)	(Ignored)	High	Low

WE WAIT FOR END

This command will wait for the end of a move or delay. It will wait until any current move or delay has finished and detects the return to the *idle* state. The 'OK' response will not be issued until the move or delay has been completed. Therefore **WE** can be used to execute I/O commands after a move is complete.

Syntax	**Units**	**Range**	**to**	**Initial State**	**Privilege level**
<ad>WE	N/A	N/A		N/A	1

Condition Requirements **Notes:**
Idle (by definition).

Response:

OK	Command has been completed.

Examples:

1MR4000	Move 4000 steps positive.
1WE	Wait for End of above move
1WP22222221	Turn LED on (write port 1) when move has finished.
1DE1000	Delay for 1 second.
1WE	Wait for End of Delay
1WP22222220	Turn LED off (write port 1).

WI SET WINDOW

This command will set the window for end of move checking. At the end of a move, when the Actual Position comes within the **WI** range of this final target, the **SE** (settling time) counter counts down. When the settling time reaches zero the controller will either accept the next command or go to the *idle* condition.
If the Position overshoots the window before to the settling time reaches zero, the settling time counter is reset and started again.

Syntax	**Units**	**Range**	**to**	**Initial State**	**Privilege level**
<ad> WInnn	Steps	0	2147483647 (2^{32})	4	6

Condition Requirements **Notes:**
Idle Value stored in FLASH by BD command.

Responses

OK	Command has been accepted.
! OUT OF RANGE	Argument is out of valid range.

Example:

1WI2 Set the Window for axis 1 to 2 steps.

WP WRITE TO OUTPUT PORT

Write to output port. The PM600 controller has eight user output ports, known as write ports 1 to 8. This command will set the write port outputs to a state defined by the specified bit pattern argument. The bit pattern is specified as an eight digit binary number. The digits will be either **0, 1** or **2** characters starting with write port 8 through to 1

Format: Eight digit binary string
consisting of 0s, 1s or 2s.
0 = *Off* 0V or open-circuit
1 = *On* +24V (depending on the voltage of Write Port V_{source})
2 = *Don't change*

Syntax	Units	Range	to	Initial State	Privilege level
<ad>WPbbbb	Bit pattern	8 digits of 0, 1 or 2		N/A	1

Condition Requirements
None.

Notes:
Initial state on power-up: all **0** = *Off*
The last *write* is shown on the **QA** page.

Responses:

OK Command has been accepted.
! INVALID BINARY Invalid argument i.e. bit specified was either not 0, 1 or 2 or the number of bits was greater than eight.

Example:
If a PM600 on axis 1 currently has the following states on its output write ports:

	PORT:	8	7	6	5	4	3	2	1
	STATE:	off	off	on	on	off	on	on	on
1WP12001200		Will set the outputs to:							
	PORT:	8	7	6	5	4	3	2	1
	STATE:	on	off	off	off	on	on	off	off
		1	2 (unchanged)	0	0	1	2 (unchanged)	0	0

WS WAIT FOR SYNCHRONISATION

This command will make the PM600 wait and not execute any more commands until the Input position equals the Motor Position. This command is used in Absolute gearbox mode.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>WS	N/A	N/A		N/A	1

Condition Requirements
Synchronised in absolute gearbox.

Notes:

Response:

OK Command has been completed.

Example:

1GA Axis 1 enter absolute gearbox mode.
1WS Axis 1 wait for synchronisation.
1WP22222221 Axis 1 switch output ON

XC EXECUTE CAM

This command will execute the defined Cam profile. The argument sets the number of times that the Cam repeats. A zero value will cause the Cam to repeat continuously.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>XCnnn	Cam number	0	7	N/A	2

Condition Requirements
Servo mode, Idle

Notes:

Responses

OK	Command has been accepted.
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Soft limit for required direction has already been reached
! CAM UNDEFINED	Cam profile hasn't yet been defined
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1XC1	Axis 1, execute Cam number 1.

XP EXECUTE PROFILE

This command will execute the defined Profile. The move occurs at a rate, defined in milliseconds by the **PT** command, for each **MR** segment to be completed.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>XPn	Profile No.	0	7	N/A	2

Condition Requirements
Servo mode, Idle

Notes:

Responses:

OK	Command has been accepted.
! HARD LIMIT	Hard limit for required direction is already activated
! SOFT LIMIT	Soft limit for required direction has already been reached
! PROFILE UNDEFINED	Profile hasn't yet been defined
! INPUT ABORT	An input abort has been detected
! STALL ABORT	A stall abort has been detected
! TRACKING ABORT	A tracking abort has been detected
! TIMEOUT ABORT	A timeout abort has been detected
! NOT ALLOWED IN STEPPER MODE	Only works in servo mode

Example:

1PT1001XP5	Axis 1, execute Profile number 5. Each segment takes 100 mS.

XS EXECUTE SEQUENCE

This command will start execution of a sequence. The argument selects which sequence is to be executed (0 to 7). The sequence must have already been defined with a Define Sequence **DS** command.
If the Execute Sequence (**XS**) command is encountered during a sequence, it will explicitly transfer control to the beginning of the sequence specified, whether it is the sequence already running or another sequence. It

may therefore be used to make a loop type sequence or jump to any other sequence. Please note that it should not be considered as a subroutine. It is like a GOTO rather than a GOSUB.
A sequence execution may be stopped before completion, or if in a continuous loop, by a Control-C or ESCAPE command.

Control-C will stop any movement immediately, exit the sequence and return to idle.
Escape will decelerate any move to a stop, exit the sequence and return to idle.

Syntax	Units	Range	to	Initial State	Privilege level
<ad>XSn	Sequence No.	0	7	N/A	2

Condition Requirements **Notes:**
Idle, Gearbox or Constant velocity

Responses:

! INVALID SEQUENCE NUMBER Argument (sequence number) is out of valid range.
! SEQUENCE UNDEFINED Sequence specified has not been defined yet.

Other responses may be generated by commands within the sequence. At the completion of the sequence, the response to the last command is sent.

Example:

1XS1 Execute sequence 1

XY CAM CO-ORDINATES

Set Cam co-ordinates. In Cam mode the slave motor is driven at a ratio of the Input encoder speed. This Cam profile is specified by two arguments separated by a ***/*** character.
The first point is always x=0, y=0. Co-ordinate pairs must be defined in order of increasing x co-ordinate.
The x co-ordinate of the last pair defines the *modulo*, that is the repeat distance. In the example given below the modulo is 750, so that the y values for x=200, x=950, x=1700, etc. are the same. Exit from *cam mode* can be achieved by either ESCAPE or ST commands.
To obtain the most accurate cam action the feedforward coefficient should be made equal to the velocity coefficient. KF=KV.
Cam positions are absolute, not relative, so that the motor position should be around zero before starting cam. The motor will only start to move when the *input* position divided by the *cam modulo* is equal to the equivalent motor position defined by the *cam*.

Syntax
<ad>XYnnn/nnn

	Units	Range	to	Initial State	Privilege level
x-value	N/A	0	32767 (2^{15})	N/A	5
y-value	N/A	-32768	32767 ($\pm 2^{15}$)	N/A	

Condition Requirements **Notes:**
Servo mode, Define Cam.

Responses:

OK Command has been accepted.
! OUT OF RANGE Argument, either X or Y is out of valid range.
! CAM FULL No memory space for further definition
!ILLEGAL CAM INSTRUCTION Command may only be used in cam definition
! NOT MONOTONIC Cam co-ordinate non-monotonic.
! NOT ALLOWED IN STEPPER MODE Only works in servo mode

Example:
Cam profiles are *piecewise linear*, with the first co-ordinate implicitly (x=0, y=0). A cam profile would be defined using the following commands:

1DC Open Cam definition.
1XY200/500 Second Cam co-ordinate.

1XY400/500	Next Cam co-ordinate.
1XY600/-200	Next Cam co-ordinate.
1XY700/-200	Next Cam co-ordinate.
1XY750/0	Last Cam co-ordinate.
1EM	End Cam Definition.

3 Status display


Idle


Move


Constant velocity


Gearbox


Home to datum


Profile running


Cam running


Jog or Joystick move


Stopping


Upper limit activated


Lower limit activated


Wait for signal/condition


Stop Input Abort


Command abort (AB)


Stall error


Tracking error


Time Out error (not got there)


Communication error

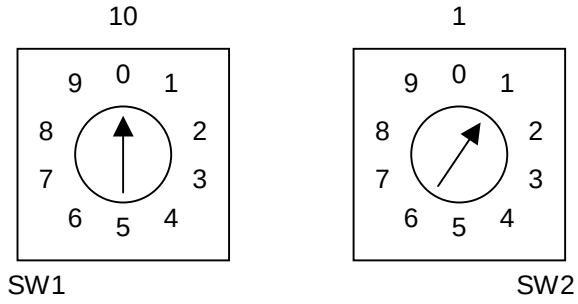

Tune


Controller failed self test

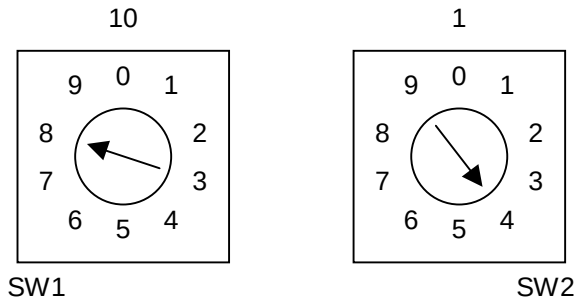
4 Switch settings

4.1 Axis address number SW1 & SW2

Rotary switches SW1 and SW2 are used to set the units axis address. This is the address of the serial commands that it will respond to. Typically, each controller in a system will be set to differing addresses. The left hand switch SW1 sets the decade value and SW2 sets the units. They may be set using a small screwdriver.



This example shows the address set to 1 (01).

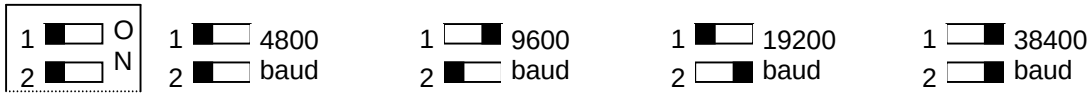


This example shows the address set to 84.

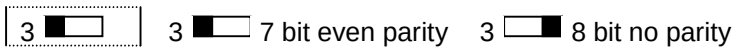
4.2 Communication Configuration Switch SW3

DIP Switch SW3 is used to set the serial communication parameters. These should be set to match those of your host terminal or PC, and all PM600 units should be set the same. If the unit receives characters that do not match the set parameters it will cause a communication error abort.

Baud rate



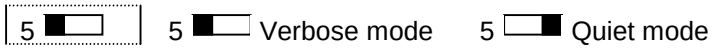
Word format



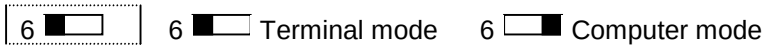
Hardware handshake (RTS / CTS)

(Not Yet Implemented – leave off)

Reply format



Transmission format

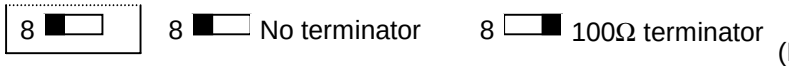


(Not Yet Implemented – leave off)

Interface type used

(Not Yet Implemented – leave off)

RS485 terminator select



(Not Yet Implemented – leave off)

Reply format:

Verbose Mode - Responses are given in full (e.g. **1OS** gives 01:Status = 00000000). Unsolicited responses are allowed (e.g. **01:! STALL ABORT**).

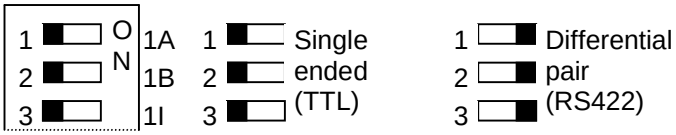
Quiet Mode - Responses are given abridged (e.g. **1OS** gives 01:00000000). Unsolicited responses are NOT allowed (e.g. **01:! STALL ABORT**).

4.3 Encoder Termination Configuration Switch SW4

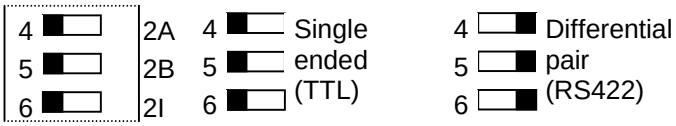
Switch SW4 selects the termination for the encoder signals. These may be set to OFF for open ended, TTL or open collector type encoder outputs or ON for 5V differential, RS422 type encoder outputs
There are switches for Channel-A, Channel-B and Index point inputs. The A and B switches should be set as a group for each encoder input

The correct termination should be set for maximum noise immunity.

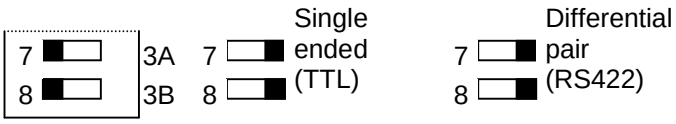
Encoder 1



Encoder 2



Encoder 3 (has no index input)



5 Error messages

! BACKUP FAILURE	Unable to store parameters/sequence/cam/profile in flash
! CAM FULL	Define Cam memory is already full
! CAM UNDEFINED	Attempting to run Cam that hasn't yet been defined
! COMMAND ABORT	Cannot move when aborted by Command Abort
! CORRUPT BACKUP	Unable to get valid data from flash memory
! HARD LIMIT	Cannot move in direction of activated hard limit
! ILLEGAL ABORT MODE	Abort Mode parameters not correctly set
! ILLEGAL CAM INSTRUCTION	Cam Definition commands can only be XY and EC
! ILLEGAL INSTRUCTION	Instruction not recognised - probably typing mistake
! ILLEGAL PROFILE INSTRUCTION	Profile Definition commands can only be MR and EP
! ILLEGAL SEQUENCE INSTRUCTION	Command cannot be store in a Sequence Definition
! INPUT ABORT	Cannot move when aborted by Stop Abort Input
! INVALID BINARY	Binary expressions must be up to 8 characters of 0, 1 or 2
! INVALID CAM DEFINITION	Possibly due to no length Cam Definition
! INVALID CAM NUMBER	Only Cam Definitions 0 to 7 available
! INVALID MODE STRING	Attempting to set an invalid mode
! INVALID PIN	Attempting to enter wrong PIN number
! INVALID PROFILE DEFINITION	Possibly due to no length Profile Definition
! INVALID PROFILE NUMBER	Only Profile Definitions 0 to 7 available
! INVALID SEQUENCE NUMBER	Only Sequence Definitions 0 to 7 available
! JOYSTICK FAILURE	Joystick input out of range or not zero on power-up
! LIMITS CONFLICT	Attempting to set Soft Limits incorrectly
! NO VALID DATUM	There is no captured datum position to query or move to.
! NOT ABORTED	Attempting to reset a controller that is not aborted
! NOT ALLOWED IN STEPPER MODE	Attempting a servo only command
! NOT ALLOWED IN THIS MODE	Command not allowed whilst doing current command
! NOT COMPLETE/TIMEOUT ABORT	Cannot move whilst aborted; last move was not completed
! NOT MONOTONIC	All X values must increase during Cam Definition
! ONLY ALLOWED IN GBOX MODE	Attempting command that only functions in Gearbox mode
! OUT OF RANGE	Argument to command is outside allowable limits
! PRIVILEGE VIOLATION	Attempting command of higher privilege level than you have
! PROFILE FULL	Profile Definition is already full
! PROFILE UNDEFINED	Attempting to run Profile that hasn't yet been defined
! RS232 ABORT	Cannot move when aborted by receiving illegal character
! SEQUENCE FULL	Sequence Definition is already full
! SEQUENCE UNDEFINED	Attempting to run Sequence that hasn't yet been defined
! SOFT LIMIT	Attempting to move beyond a soft limit
! STALL ABORT	Cannot move when aborted by stall detect
! TRACKING ABORT	Cannot move when aborted by tracking error
! TUNE FAILURE	Auto Tune has not worked
Self-test failure	Power On Self Test detected failure
SKIPPED	Command was skipped due to preceding IT or IF command

6 External connections

Electrical connections to the PM600 controller are made via the standard 96 way DIN41612 connector. The pin assignments are as follows:

	A	B	C	
01	+VLL supply	+VLL supply	+VLL supply	01
02	Idle Output Source	Error Output Source	Analogue Input 1	02
03	Idle Output	Error Output	Analogue Input 2	03
04	Joystick 1 (AI3)	Joystick 2 (AI4)	Joystick Centre (AI5)	04
05	+5V output	+5V output	+5V output	05
06	Encoder 1 A+	Encoder 2 A+	Encoder 3 A+	06
07	Encoder 1 A-	Encoder 2 A-	Encoder 3 A-	07
08	Encoder 1 B+	Encoder 2 B+	Encoder 3 B+	08
09	Encoder 1 B-	Encoder 2 B-	Encoder 3 B-	09
10	Encoder 1 I+	Encoder 2 I+	Abort Stop Input	10
11	Encoder 1 I-	Encoder 2 I-	Abort Stop Isolated 0V	11
12	Analogue Output 1	Analogue Output 2	Jog Select Input	12
13	Analogue 0V	Analogue 0V	Jog Positive Input	13
14	Channel 1 Enable Source	Channel 2 Enable Source	Jog Fast Input	14
15	Channel 1 Enable Output	Channel 2 Enable Output	Jog Negative Input	15
16	Channel 1 Step Output	Channel 2 Step Output	Jog Channel Output	16
17	Channel 1 Direction Output	Channel 2 Direction Output	Jog Isolated 0V	17
18	Channel 1 Upper Hard Limit	Channel 2 Upper Hard Limit	RS232 Transmit (Out)	18
19	Channel 1 Lower Hard Limit	Channel 2 Lower Hard Limit	RS232 Receive (In)	19
20	Channel 1 Datum Approach	Channel 2 Datum Approach	RS232 RTS (Out)	20
21	Channel 1 Datum Stop	Channel 2 Datum Stop	RS232 CTS (In)	21
22	Channel 1 Limit/Datum Iso 0V	Channel 2 Limit/Datum Iso 0V	RS485-A	22
23	Write Port 1	Read Port 1	RS485-B	23
24	Write Port 2	Read Port 2	M-bus Data	24
25	Write Port 3	Read Port 3	M-bus Clock	25
26	Write Port 4	Read Port 4	Mezzanine Option 1	26
27	Write Port 5	Read Port 5	Mezzanine Option 2	27
28	Write Port 6	Read Port 6	Mezzanine Option 3	28
29	Write Port 7	Read Port 7	Mezzanine Option 4	29
30	Write Port 8	Read Port 8	Mezzanine Option 5	30
31	Write Port Source	Read Port Isolated 0V	Mezzanine Option 6	31
32	0VLL ground	0VLL ground	0VLL ground	32

We recommend the use of the MSB603 motherboard to facilitate connections.

7 MSB603 Motherboard

The MSB603 motherboard facilitates easy connections to the PM600. The connectors have been arranged into functional groups.
Encoders – 10 way IDC connectors
RS232 – 10 way IDC connectors
Digital I/O – 20 way IDC connector
Power – screw-terminal plugs
Others – 0.1” Molex type connectors