

# Precision Motion Controls

## PWX Series Motors and Drives

Operator's Manual  
PN 04-01809 Rev. A.2

# **Precision Motion Controls**

160 E. Virginia St. #264 San Jose, CA 95112 Tel : (408) 298-0898 Fax: (408) 298-0899



# Table Of Contents

Introduction .....	2
Description .....	2
Warranty .....	3
Limitation of Warranty .....	3
Terms and Conditions .....	3
Installation .....	4
Unpacking .....	4
Mounting .....	4
Wiring .....	4
Motor Connections .....	4
Line Power .....	5
Indexer Connections .....	5
Switch Settings .....	6
Operation .....	8
Electrical .....	8
Motor Compatibility .....	8
Resonance .....	8
Inputs/outputs .....	8
Sequence Inputs .....	8
Trigger Inputs .....	9
Home, Limit+, Limit- .....	9
Output+, Output- .....	10
RS232 Connections .....	10
Theory of Indexer .....	11
Concept .....	11
Command Execution .....	11
Command Structure .....	11
Indexer Functions .....	11
Preset Move .....	12
Continuous Move .....	12
Loop Capability .....	12
Sequences (Stored Programs) .....	13
Trigger .....	13
Home .....	13
Time .....	14
IF Command .....	14
TS Registration move .....	14
Command Set Description .....	15
Trouble Shooting .....	32
Command Summary .....	34
Appendix	
A. PWX Specification	
B. Dimension And Torque Curves	

## Introduction

### Description

The PWX indexer/drive is a integral microstepping drive and indexer in one package. Communication to the indexer is through a RS232 port at 9600 baud. The indexer supports a comprehensive set of industry standard commands. The unit can store up to 8 programs of 255 bytes each in its battery backed RAM. The indexer has 14 input-output lines; 5 are trigger inputs, 3 output lines, 3 binary program select lines, 2 limits and a home sensor input. When used in stand alone applications 7 of the stored programs can be executed from the program select lines. The indexer drive has its own built in power supply, relieving the customer of the responsibility and cost of supplying one. All of the systems I/O lines are optically isolated eliminating ground loops to connecting equipment. An isolated 5 volt, 500 ma. current power supply is available to the user and may be used for the opto 1 and opto 2 sensor supplies.

The drive has a current monitor in the supply line to provide short circuit protection and increase reliability. The amplifier operates at a 20 kHz pulse-width modulation rate, this is above the human hearing response level and below the point where the motor magnetic's become inefficient. The PWX can be ordered with motors from 60 to 1200 in-oz of torque; the motor case sizes range from NEMA 23 to 42. The drive has programmable resolutions of 25600, 25400, or 20000 steps /revolution with a 200 step per revolution stepper motor. The default power up resolution is 25600 steps/revolution. These choices of an easy method to work in binary, metric, or decimal base respectively. An accessible DIP switch lets the user select current in 62.5 ma. increments to 3.5 amps. A programmable 3rd harmonic current waveform selection provides maximum smoothness of movement for a variety of motors.

### CAUTION:

The drive is not AC line isolated. The motor and driver box are tied to earth ground through the AC line cord for safety. Do not plug the unit into an outlet that does not have earth ground, or use an extension cord that does not have an earth ground.

The motor's terminal screw contains hazardous voltage. Do not probe or touch it.

## **Warranty**

PMC's PWX drives have a one year warranty against manufacturing defects from the date of purchase. If your unit should ever fail, and you wish to send it back for repairs, you should do the following:

1. Get the serial number from the defective unit.
2. Check purchase date to see if the unit is under warranty. If not, obtain a purchase order number for repair costs.
3. Call Precision Motion Controls for a return authorization (408) 298-0898.
4. Ship to :

Precision Motion Controls  
2530 Berryessa Rd. #209  
San Jose , CA 95132

Attention RMA# \_\_\_\_\_

## **Limitation of Warranty**

Sellers liability arising from or connected with item purchased shall be limited exclusively to repair, replacement or refund of the buyers purchase price, at the sellers sole option. In no event shall seller be liable for any incidental, consequential or special damages of any kind, including but not limited to lost profits arising from or in any way connected with this agreement or items sold hereunder. Including without limitation, negligence, failure, to warn, or strict liability.

## **Terms and Conditions**

All acknowledgments, acceptances, and sales of sellers products are subject to and shall be governed exclusively by the terms and conditions stated herein.

## Installation

### Caution!!!

**The drive is a delicate electronic device, always disconnect the AC power prior to connecting or disconnecting the motor to the driver. Inspect the motor winding at the connector and verify they are not shorted to each other before plugging into a 115 VAC outlet. Make sure the external +5 volt is not shorted to the 5 volt RTN. The external +5V supplies 500 ma. of maximum current.**

### Unpacking

When unpacking your unit, verify that the unit was not damaged during shipping. Report any damage found to the shipper. Check the box contents against the packing slip. The box will contain the driver, AC line cord, mounting brackets and manual; if a complete system was bought the motor will also be included. Connect the motor to the driver box and then plug in the AC line cord to a 115 VAC outlet. The power light should be on and the fault light off. Feel the shaft of the motor and verify that the motor is producing torque. If you are able to move the motor shaft, the motor does not have torque; unplug the line cord. Check if any of the motor wires have come out of their connector and the DIP switch current setting (the DIP switch is accessed from the bottom of the box). Contact the factory for a return authorization if the above checks prove negative.

The drive is shipped with the test command programmed in sequence 1 at power up. This will cause the motor to run CW continuously at 0.1 rev/sec. Note the limits are not looked at in this mode so make sure that nothing is connected to the motor shaft. Use command "K" to stop the movement. You can remove the power-up sequence by entering "XP8".

### Mounting

The PWX drive comes with mounting brackets to facilitate mounting. The unit may be mounted in any direction. For motor currents between 2 to 3.5 amps the box should be mounted with a 3 inch opening surrounding the drive and the heat sink fins running in the vertical direction to help cooling. See appendix for a detailed drawing of the motors and box.

### Cooling

The surrounding temperature of the environment should not exceed 40 degrees centigrade. If the driver needs to be in a warmer environment, consult the factory for recommendations. The temperature of the heat sink should not exceed 50 degrees centigrade.

### Wiring

#### Motor Connections

If the drive is purchased with a motor, the motor cable will have the connector attached to it; otherwise a connector with screw mounting will be supplied. The center pin is for connection of the motor cable shield. The center pin is connected to earth ground and the shield is connected to the motor case; if the motor is mounted to a potential other than earth ground do not connect the shield to the center pin.

A+	(Green)
A-	(White)
Shield	(Earth gnd)
B+	(Black)
B-	(Red)

## Line Power

The correct line cord is supplied for connection of 115 VAC power. Make sure that the unit is plugged into a wall socket with earth ground.

## Indexer Connections

The input signals are photo-coupled from the motor ground to minimize coupled noise and to prevent ground loops. The connectors are Phoenix style connectors, this provides easy screw type connections on a removable connector.

Examples of the input and output circuits are shown in figure 1 and figure 2 respectively. Note the inputs have a common anode connected to either the opto1 or opto 2 terminals through a 330 ohm resistor.

1 OPTO 1	Internally connects a series resistor 330 ohm to LIMIT+, LIMIT-, and HOME sensor LED's RESPECTIVELY. The opto1 supply input is designed to be connected to a +5V supply.
2 LIMIT +	Pull down to ground disable. Figure 1 shows a recommended circuit.
3 LIMIT -	Pull down to ground to disable.
4 HOME	The default active polarity(home) is the home input pulled down. You can change the active level . Refer to OSB command.
5 OPTO 2	Internally connects to the 5 TRIGGERS, and 3 SEQUENCES inputs through a 330 ohm resistor.
6 TR1	Pull to ground to activate.
7 TR2	Pull to ground to activate.
8 TR3	Pull to ground to activate.
9 TR4	Pull to ground to activate.
10 TR5	Pull to ground to activate.
11 SEQ1	All sequence inputs must go high for the debounce time prior to the desired selection; use SN command to set value.
12 SEQ2	
13 SEQ3	
14 OUT1+	Collector of photo-coupled transistor.
15 OUT1-	Emitter of photo-coupled transistor. The transistor will have 10 ma. maximum current if it turns on. The maximum voltage across the transistor is 30 volt.
16 OUT2+	
17 OUT2-	
18 OUT3+	
19 OUT3-	
21 TX	Connect to the terminal RS232 receive input.

22 RX                    Connect to the terminal RS232 transmit output.  
 23 RS COM-            Connect to the terminal RS232 ground.  
 24 +5V                 This voltage may be higher than +5. Do not use for critical components. Maximum  
 25 5V RTN              500ma. is provided for external use. A fuse should be used to protect the drive.

FIGURE 1:

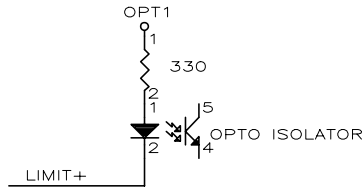
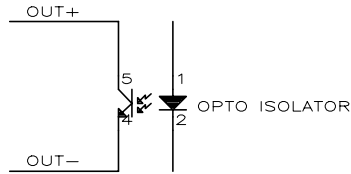


FIGURE 2:



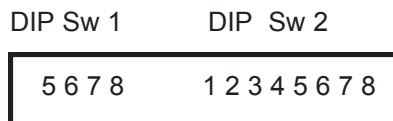
### Switch Settings

Disconnect AC power from the unit prior to attempting to change any switch settings. The switch settings control the maximum winding current, unit ID, and auto standby mode.

There are two DIP switches on the bottom of the box. The following diagrams show the DIP switch on the bottom of the box and the switches functions. Table 1 lists the DIP switch settings for various motor currents and address of the drive.

#### DIP SWITCHES

ON



OFF

TABLE 1

DIP SW 2

Switches	1	2	3	4	5	6
PM - 60 MTR	off	on	off	on	on	on
PM - 90 MTR	on	on	on	off	on	on
PM - 120 MTR	off	off	on	off	on	on
PM - 140 MTR	off	on	off	off	on	on
PM - 260 MTR	on	off	off	on	off	on
PM - 380 MTR	off	on	on	off	off	on
PM - 450 MTR	on	off	off	off	off	on
PM - 1200 MTR	on	on	on	off	off	off

Switches 1 through 6 are used for the current settings and are approximately 62.5 ma. per bit. The off position of the switch selects the current. The PMC supplied motors run from approximately 0.4 to 3.5 amps/phase.

Switch 8      auto standby   off

Set switch 8 to the off position to activate the auto standby mode. The motor current will reduce to 1/2 of its DIP switch value 2 seconds after the motor stops.

The system ID is on DIP Sw 1. The units address for various switch settings are shown in the table below.

System ID	Switches		
	6	7	8
0	on	on	on
1	off	on	on
2	on	off	on
3	off	off	on
4	on	on	off
5	off	on	off
6	on	off	off
7	off	off	off

## Operation

### Electrical

Input Power: 95 to 132 VAC 50/60 Hz

Fuse : 5 amp

Output Power: 0 to 3.5 amps at 170 VDC through a H bridge driver (supply takes 5 minutes to discharge after power is removed).

#### Safety Features:

Short circuit protection is provided by a current monitoring circuit in the power supply line. The instantaneous supply current is monitored and if current in excess of 10 amps is detected the output drivers are disabled. Power to the unit must be turned off and back on if an over-current or over-temperature condition is detected. Check the motor connector to verify nothing is shorting the windings prior to turning power back on.

An internal temperature sensor monitors the temperature and shuts down the supply if the temperature internally exceeds 60 degrees centigrade.

### Motor Compatibility

Motor inductance - approximately ranges from 40 to 150 mH for a PMC supplied motor. The drive has a recommended minimum inductance requirement of 25 mH. Motors with 10 mH can be used but this will generally cause most motors to run too hot; except for the largest frame size ones. Typically as the motor inductance increases the top motor speed will decrease and motor heating will decrease.

“DO Not” connect or disconnect the motor with power applied to the box.

### Resonance

All stepper motor drives are subject to two oscillation regions; low speed (approximately 1 rev/sec) and mid range (approximately 10 to 15 rev/sec).

Microstepping minimizes low speed oscillation and this should not occur with PMC supplied motors. If your load has a high Q resonates around 200 Hz, adding extra inertia to the motor shaft may eliminate the problem.

PMC motors and drives have been matched to eliminate most mid-range instability problems.

### Inputs/outputs

#### Sequence Inputs

The 3 sequence inputs are decoded into 8 sequences . The following table lists sequence input combinations versus stored program. The all sequence high input condition is reserved for no sequence condition (null). Only stored programs 1 to 7 may be addressed by this method. Each program is allocated 256 bytes in the battery backup RAM. The sequence lines must be held for debounce time without changing before it will be accepted. Changes on the sequence input are only valid when the processor is not executing other commands. Thus loops must not be executing; a sequence must not be running, or distance/constant velocity moves must not be in

progress. The only movement that is valid when a sequence input is changed is when the motor is running in the constant velocity mode and no further moves are pending. The sequence inputs must return to the null position prior to going from one sequence to another.

SEQ1	SEQ2	SEQ3	PROGRAMS
ON	ON	ON	7
OFF	ON	ON	6
ON	OFF	ON	5
OFF	OFF	ON	4
ON	ON	OFF	3
OFF	ON	OFF	2
ON	OFF	OFF	1
OFF	OFF	OFF	reserve

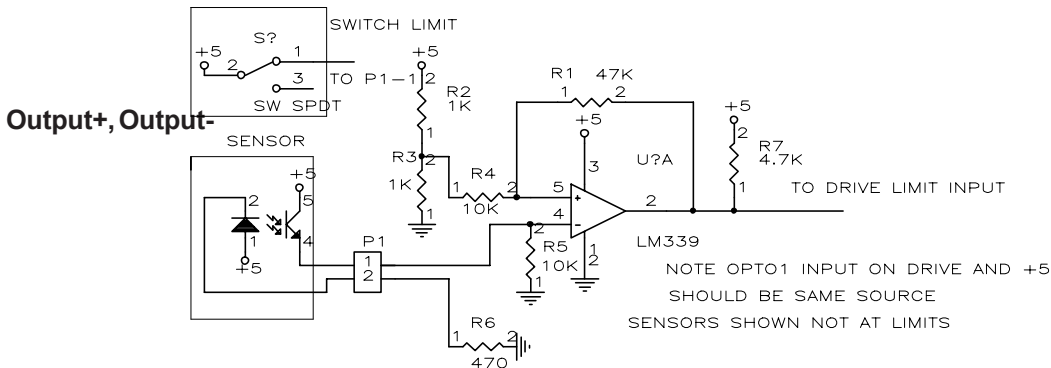
### Trigger Inputs

5 trigger inputs are provided. Pulling the trigger input low puts it to its active state. A trigger input must be in its active state for a minimum of 1 msec. in order to be guaranteed to be recognized. When a trigger command is received by the processor it waits for that signal to become active before proceeding to the next command. There are two trigger commands "TR and CTR"; the TR command is position independent (always executed prior to movement); whereas CTR is position dependent. See the command descriptions for more information and examples.

### Home, limit+, limit-:

The driver for these 3 inputs need 10 ma. sink capability; and should be valid for a minimum time of 100 ms. The Limit+ input is the limit that is energized when the motor is running in the CW direction. The Limit- should be connected to the sensor at the most CCW position of motor travel. The limits can be disabled by the LD command. To assure proper operation limits should not be able to sit at half levels. It is recommended that the outputs of limit sensors be sent through a comparator with hysteresis prior to being sent to the limit inputs of the drive. See the following figure for a sample circuit.

Figure 4:



There are 3 outputs available for the user. Output+ connects to the collector of the output photocoupler, while output- connects to the emitter of the photocoupler. See figure 2. The photocoupler has a maximum voltage rating from collector to emitter of 30 volts and has a maximum current of 10 ma.

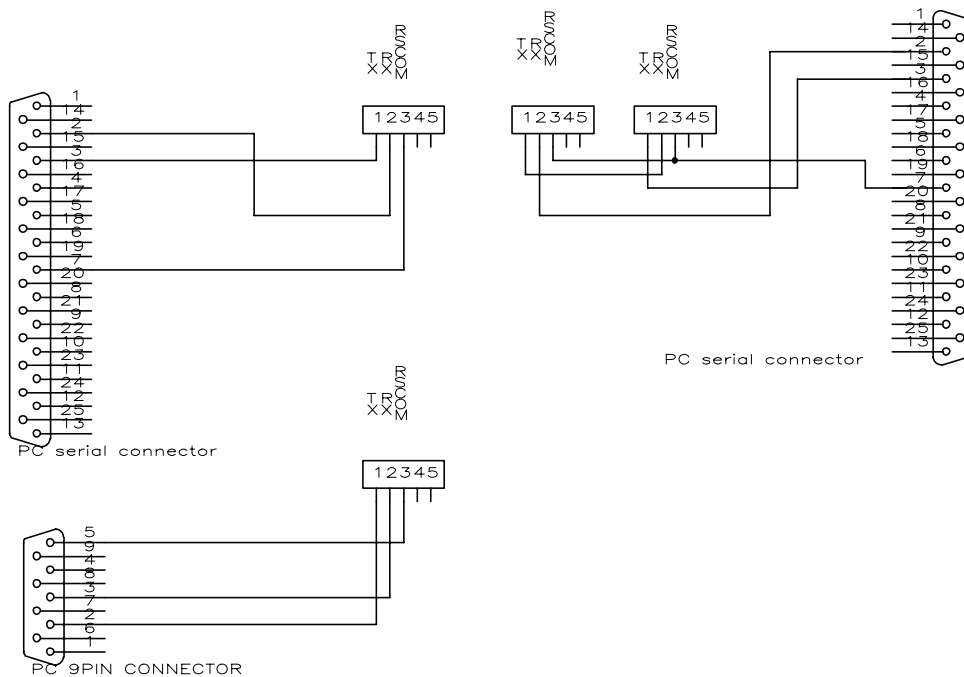
### RS232 connections

A standard 3 pin standard RS232C interface is used; the connections for single and multiple drives are shown in the figures below. A maximum 8 drivers can be daisy chain together. IF only one drive is connected to the terminal the echoing back of characters can be inhibited. The drives will not work properly if echoing is disabled on a multi-drive system. The serial communication setup is listed in the following table.

BAUD RATE	DATABITS	PARITY	STOPBITS
9600	8	NONE	1

The following diagram shows a single axis connection to a terminal and a multi-axis connection. Note on the multi-axis connection the transmit of one drive is connected to the receive of the next. Thus there is only a 2 wire connection between drives (a communication line and a ground line). The data from the computer is passed from one drive to the next and finally echoed back to the computer from the last drive.

Figure 5:



# Theory of Indexer

## Concept

### Command Execution

Commands are executed in different manners depending on how they are received. As a general rule the system will not accept a "G" (go) command while the motor is running. There are three ways that the systems receives commands to be executed. They can be received in the interactive mode (received from a terminal), received from the loop buffer; or executed from a stored program. The interactive mode executes all commands when received except for constant velocity commands (CV, CTM, CTR); these are put in a buffer and are executed after the motor starts movement, the constant velocity commands are executed in order and the previous one must be completed before the next one starts. The loop instruction prevents the execution of any of the commands received after the "L" until the end of loop command (N) is received.

The execution of a sequence has a different priority. Stored programs try to execute everything between the "G" commands and waits until the move up to the go is completed before proceeding to execute anything after the go. The system will then try to process the commands up to the next "G". Thus the follow example will complete the constant velocity profile move and then do the distance move. Note the system must be brought to a stop before the position move is executed.

MC A5 V10 CV2 CV12 CV0 G MPI A5 V3 D50000 G (in the stored program)

### Command Structure

The PWX accepts both upper case and lower case characters and echoes back upper case only. A SPACE or RTN delimiter must be entered after a command is entered.

The commands for the PWX drive can thought of as in 3 categories; status, parameter, and system. Status commands request information from the indexer such as position, velocity, limit status, etc. Parameters are commands that affect movement such as velocity, type of movement, time delay. Parameters are also categorized into 3 different types; immediate, buffer, and register. As the name implies, immediate commands will be executed as soon as they are received, buffer commands are buffered until the previous instruction is finished. A, V, D, MC, MPI, MPA, are register type commands. The indexer will use the register parameter previous stored unless it receives a new one. A system command starts the execution of a function such as the go, reset, stop, and define program.

### Indexer Functions

Indexer functions can be divided to following :

- Preset move**
- Continuous move**
- Loop capability**
- Stored program sequence control**
- Trigger command**
- Homing function**
- If command**
- Registration move**

Each one of these are explained in the following paragraph.

## Preset Move

There are 2 type of preset moves; incremental move, and absolute move. Incremental moves will move the distance specified each time a GO command is executed. Absolute moves will move to the distance specified by the "D" command. The preset move consists of A(acceleration), DC(deceleration), V(velocity), D(distance), and optionally T(time) information.

The indexer uses the acceleration and deceleration terms to calculate the distance needed to ramp up and ramp down. It will use the acceleration as the deceleration value if a deceleration term isn't entered. The power up default deceleration value is zero. If the calculated distance to ramp is greater than the distance to move, a triangle type move is generated and the commanded velocity is never reached. If the total distance to travel exceeds the ramp distance a trapezoidal move is performed. The motor accelerates to commanded velocity, and stays at that velocity till the distance need to ramp down from the final position is reached.

MPA (absolute move), or MPI (incremental move) initiates a preset move. Once they are issued, the indexer will stay in that mode until it is changed.

Separate distance moves can be accomplished by linking them with go commands. T (time) commands will always be executed before a move.

## Continuous Move

Continuous moves are initiated with the MC command. Distance information is ignored. Immediate and buffer commands are executed differently, therefore, they will accomplish different kinds of moves. The following examples highlight the differences.

The format for continuous velocity moves are as follows:

```
MC A5 V5 CV10 CTM5 CV2 CV0 G
```

The motor will accelerate to 5 rps with acceleration 5 rps<sup>2</sup>, then accelerate to 10 rps, stay at that speed for 5 seconds, decelerate to 2 rps, and then decelerate to a stop.

```
MC A5 V5 CV10 CTM10 CV5 CV0 G VC1
```

The motor will accelerate to 5 rps with a 5 rps<sup>2</sup> acceleration, it will try to accelerate to 10 rps, most likely it will receive the VC1 prior to executing the CTM10 command; if so it will then execute this command immediately to decelerate to 1 rps. After the VC1 is done the move continues with the CTM10, CV5, CV0. The CV commands are buffered and place dependent command whereas VC1 is an immediate command (executed as soon as received). The VC command will override the existing CV that is in process.

## Loop Capability

Looping is a convenient way to accomplish repetitive moves. The loop command starts with an "Lxx" (x= number of times to loop) and terminates with a "N" command. The Indexer will loop xx times. As Lxx is entered the following commands are going through a preliminary check, and buffered for later execution. The following is an example of a loop command.

```
L20 MPI A5 V5 D51200 G T10 H G N
```

The motor will move 2 revolution at a velocity of 5 rev/sec; wait for 10 sec.; then change direction and move back to the origin. The above move will repeat 20 times. Only K, S, Z, will stop the on going loop.

Setting the loop equal to one (L1) is a way of entering a complex distance move, and delaying the start until the entire command is entered. The Y command will terminate a loop after the completion of the next time through the loop.

```
L1 MPI A5 V5 D51200 G H D51200 G N
```

Without the loop command the motor would have taken off as soon as the first g was received.

### **Sequences (Stored Programs)**

Commands can be downloaded and stored in a sequence only if the drive is not waiting for a trigger, or not executing a time. The system can store up to eight programs of 256 bytes each in battery backed RAM. A sequence can contain loops, call another sequence, and contain different types of motion (MPI, MPA, MC); however the motor must come to a stop between moves and each different type must contain a go. The following sequence is legitimate.

Sequence 1

```
XD1 MC A5 V10 L3 CV5 CV15 G N CV0 G MPI A15 V6 D50000 G XT
```

Mixed mode and loop. Ramps to 10 RPS then loops 3 times from 5 to 15 RPS. The motor then decelerates to zero and then initiates a distance move of 50000 steps in the clockwise direction.

Note the motor must come to a stop before the mode is changed. The CV0 brings the motor to a stop, then the mode is changed. In a sequence the "G" acts as a separator of command sequences. That is everything before the go must be completed before the next command after the "G" is executed.

### **Trigger**

Triggers are a way to synchronize the start of a move or the continuation of a move with external events. The TR command can be used to delay the start of a move until the trigger input changes. The CTR is used in conjunction with continuous velocity moves and will prevent the next change in velocity from occurring until the trigger input is active. Thus they act as a variable time delay.

```
MPI A10 V5 D50000 TR1 G
```

Waits for trigger 1 to go active prior to doing the distance move.

```
MC A20 V10 CTR2 CV5 G
```

Accelerates to 10 rev/sec and then stays at that velocity until trigger 2 goes active and then decelerates to 5 rev/sec.

### **Home**

The Home routine travels between the limits looking for the home sensor. The motor first travels towards the positive limit until it finds the home sensor or the plus limit. If the limit is detected it then travels in the CCW direction towards the minus limit. Once the home is detected the motor moves in the CW direction at 0.1 rev/sec until the home sensor is no longer detected; it then moves CCW at 0.1 rev/sec until the home sensor is detected again. If the GH command is used in a sequence, surround it by a loop of 1, then you may put any other movement commands after it. The following sequences are legitimate.

```
xd1 osb0 gh1 xr2 xt  
xd2 mpi a5 v5 d50000 g xt
```

```
Xd1 osb0 l1 gh1 n mpi a5 v5 d50000 g xt
```

The following sequence is illegitimate.

```
xd1 osb0 gh1 mpi a5 v5 d50000 g xt
```

## **Time**

There are two types of time commands; T, and CTM. The difference between T and CTM is similar to the difference between V and CV. See the following 2 examples. Note the CTM command is place dependent and the T command is executed when encountered.

```
MPI A5 V5 D50000 T10 G
```

The motor will wait for 10 seconds after receiving the "T", then execute the movement.

```
MC A5 V5 CV1 CTM10 CV10 G
```

The motor will change from 5 rev/sec to 1 rev/sec, wait for 10 second then change to 10 rev/sec.

## **IF Command**

The If command should only be used in stored program.

The If command looks to see if the specified trigger is present and if it is not, it skips the next command. If the trigger is active, the command following the if is executed. The next command could be a jump to another program if the if statement is not inside a loop. The following is an example of a jump.

```
MPI A5 V10 D500000 G IF1 XR1 XR2
```

The motor will move 500000 steps and then check trigger input 1. If it is active it will execute sequence 1 else it will execute sequence 2.

The If command can also be used to terminate a loop based on an external event.

```
L20 MPI A5 V10 D500000 G IF2 Y N
```

The proceeding program would move the motor 500000 steps then check trigger input 2; if the input is active it executes the y command which terminates the loop by setting the loop count to 0. If the trigger input is not active, the move is repeated. This is done 20 times or until trigger 2 is active.

## **TS Registration Move**

The registration move is a fixed distance move in which the distance to be moved can be changed to a distance reference from the occurrence of trigger 1 becoming active.

```
MPI A5 V10 D500000 TS100000 G
```

the motor will move 500000 steps unless trigger 1 occurs during the movement. If trigger 1 occurs the motor will move 100000 steps from the point at which trigger 1 occurred; it will continue at the velocity the motor was moving when the trigger occurred. A command similar to this but only decelerates the motor to a stop is the RS command. If trigger 1 is encountered, the motor decelerates to a stop.

```
MPI A5 V10 D500000 RS1 G
```

## Command Set Description

The following shows is a list of the commands, their name and an example of their syntax and a description of their functions. Most also show an example and an explanation of the example. The form that the commands are presented in is shown below.

Abbreviation Name  
Syntax {a}= address of device                    category  
description  
example

x represents a numeric number of 0 to 9  
s represents sign + or -  
{a} = 0-7 drive ID address

Responses from the drive for status have the following format

\*{a}:data

The “\*” signifies that the following information is data from the drive and not characters being echoed back. The “a” is the drive ID from which the status is coming from. The data then follows. If the data is in hex format the actual data has a prefix of “0x”. Thus a position report in hex format would be as follows 0x3fe456.

---

A Acceleration  
syntax {a}Axx.xx                    parameter

Acceleration command; sets the acceleration in revolutions per second. The acceleration and deceleration rates are the same if deceleration value has not been entered; acceleration = deceleration. The acceleration rate is from 0.01 to 99.99 revolutions/sec<sup>2</sup>.

MC A5 V10 G

The motor will accelerate at 5 rev/sec<sup>2</sup> to 10 rev/sec after the “g” is entered.

---

AC Change acceleration  
Syntax {a}CAXX.XX                    parameter

Changes the acceleration parameter immediately; intended use is for interactive control from a host while the motor is moving in the constant velocity mode. This command is discarded if the motor is not moving when received.

Continue  
Syntax {a}C system

Enables execution of a program after a pause was encountered. This allows distance moves to be downloaded from a terminal; but execution will not start until the C command is received. The same thing can be accomplished by storing the move in a sequence and then running the sequence when ready; or making the sequence surrounded by a loop of 1.

PS MPI A5 V10 D500000 G C

---

CA Change acceleration  
syntax {a}CAxx.xx parameter

Changes the acceleration in the constant velocity mode, this is only a temporary change for the rest of the move and does not change the original acceleration value. In a string of commands it is position dependent; i.e. all preceding commands must have been executed before it becomes effective.

1) MC A5 V5 CA10 CV2 CV5 G

The motor accelerates at a rate of 5 rev/sec<sup>2</sup> to a velocity of 5 rev/sec and then decreases to 2 rev/sec at a 10 rev/sec<sup>2</sup> deceleration rate; after reaching 2 rev/sec it then accelerates to 5 rev/sec at 10 rev/sec<sup>2</sup> and remains there.

---

CO Output continuous velocity mode  
Syntax {a}COxx parameter

Turns output 1,2, or 3 on or off. The first digit signifies the output and the second digit indicates whether the output is on (1) or off (2). This command is only valid in the constant velocity mode and is place dependent. Their system interaction is similar to the CTR (trigger) and CTM (time) commands.

CO11 turn output 1 on  
CO10 turn output 1 off  
CO21 turn output 2 on  
CO30 turn output 3 off

MC A5 V5 CO11 CV1 CO10 G

Accelerates to 5 rev/sec at 5 rev/sec<sup>2</sup> and then turns on output 1; then decelerates to 1 rev/sec and turns off output 1 while in the continuous velocity mode.

---

CTM Time delay constant velocity  
Syntax {a}CTMxx.xx parameter

Time delay while moving, this command is place sensitive and is executed after the preceding command is completed.

1) MC V10 A5 CTM5 CV5 G

Accelerate to 10 rps, delay 5 seconds, then decelerate to 5 rps.

---

CTR    Trigger Constant velocity  
Syntax {a}CTRx            parameter

Waits for the trigger input to go active low before proceeding. The trigger input is specified by the first character after the "R" and the active level is low. This command is used in the constant velocity mode and is position dependent. The velocity preceding the command must be reached before the system looks for the trigger.

Example:  
MC A5 V10 CTR3 CV2 G

The system accelerates to 10 rev/sec and then waits for trigger input 3 to go low; then decelerates to 2 rev/sec.

---

CV    Change velocity  
syntax {a}CVxx.xx            parameter

Changes the velocity of the motor in the constant velocity mode. The command is position dependent and the previous commands must be executed before it is executed. Range is from 0.001 to 40.00 rev/sec.

MC A5 V10 CV2 CV7 G

The motor will accelerate at 5 rev/sec<sup>2</sup> to 10 rev/sec after the "G" is entered then decelerate to 2 rev/sec then accelerate to 7 rev/sec.

---

D    Distance  
Syntax {a}D+xxxxxxx parameter

Sets the distance to move in motor steps. The default resolution on power up or reset is 25600 steps per revolution. The range of movement is + 32 million steps. A positive distance moves the motor in the "CW" direction in the incremental mode. When the system is in the MPA mode the distance entered refers to a position with respect to the zero reference.

example:

MPI A8 V10 D51200 g

The motor moves 51200 steps (2 rev's) in the clockwise direction with an acceleration of 8 rev/sec<sup>2</sup> and a maximum velocity of 10 rev/sec.

---

DC    Deceleration Value  
Syntax {a}DCxxx.xx            parameter

Sets the rate of deceleration. On power-up this value is set to zero. If it has a value of zero the acceleration (A) value is used for deceleration. Values range from 999 to 0.01 rev/sec<sup>2</sup>. This allows for different acceleration and deceleration rates.

example:

MPI A100 V20 D500000 DC50 G

This performs a distance move of 500000 steps and accelerates to 20 rev/sec at a rate of 100 rev/sec<sup>2</sup>, it then decelerates at a rate of 50 rev/sec<sup>2</sup>.

---

DD     Disable done

Syntax {a}DDx            system

Disable done; normally a "DN" is sent when the motor comes to a stop. This is the power up default condition. This function can be enabled or disabled with this command.

DD0 enable done response

DD1 disable done response

---

DN     Done

syntax DN                    system

Response sent when the motor comes to a stop after a movement command is issued. The system sends this out each time a command is completed up to the G command and the velocity is zero. Thus the following 2 examples a sequence with 3 G's or a loop of three around a sequence with 1 g would each produce three DN's.

Examples:

MPI A5 V8 D50000 G D25000 G D25000 G (if in stored program)

MPI A5 V8 L3 d50000 G N

Response:

DN DN DN

---

DS     Secondary distance

Syntax DSxxxxxxx

In an incremental move this sets the distance to move at the secondary velocity. This is intended to allow a drive to move the final portion of the move at a different velocity.

Example

MPI A50 V15 D500000 DS50000 VS2 G

The motor will move a total distance of 500000 steps. The motor will accelerate to 15 rev/sec with an acceleration of 50 rev/sec<sup>2</sup>, it will then decelerate at a point so that at 450000 steps it is going at 2 rev/sec. Also see "VS" command , this sets the secondary speed.

---

#### E Enable serial communications

Syntax {a}E system

Enable serial communications. If serial communications are disabled with the “F” command they must be enabled with this command before the unit will accept any more commands. this command is used when multiple systems are daisy chained. See “F” command

---

#### F Disable serial communications

Syntax {A}F system

Disable serial communication should be used on daisy chained systems with multiple units. When status from a particular unit is requested all other units should be disabled, this prevents these units from confusing status information from one drive with commands. To enable communications issue the “E” command.

---

#### G Go

syntax {a}G system

Initiates the start of motor movement. A “G” will use the V, A, and D parameters of the last move if the parameters have not been changed. If the motor is stopped no further movement will occur until a G is received. If modes are changed all parameters necessary for that move should be issued prior to the GO, otherwise unpredictable movements may occur.

Example 1:

MC A5 V10 G

The motor accelerates at 5 rev/sec<sup>2</sup> to 10 revs/sec; the motion starts when the G is received.

Example 2:

G

If the indexer is in the constant velocity mode it will accelerate the motor using the last “A” value received to the velocity of the last V command received. In the incremental position mode (MPI) it will move the distance specified by the last “D” command received. In the absolute position mode (MPA) the motor will not move since it is already at the last specified absolute distance.

---

#### GH Go Home

Syntax {a}GHxx.xx system

Seeks the home position at first in the CW direction at the velocity specified . If a limit is reached before the home sensor is encountered the motor changes direction and goes until home or the other limit sensor is detected. For the home condition to be satisfied the limit and home sensor cannot be active at the same time. Also see the OSB command that works in conjunction with this.

## GH1.1

The motor turns in the clockwise direction at 1.1 rev/sec looking for the home sensor. If the CW limit is detected the motor reverses direction and goes until the home sensor or CCW limit is detected.

---

### H Change direction

syntax {a}H+ parameter

Changes the direction of the previous move if the sign parameter is not included. In the incremental distance or velocity mode a "+" forces clockwise rotation while a - forces a counter clockwise rotation.

Example:

1) MPI A5 D50000 V3 G H G

The motor moves 50000 steps clockwise then moves 50000 steps counter clockwise.

2) MC H+ A3 V5 CTM5 CV0 G H G

The motor accelerates to 5 rev/sec in the CW direction and remains there for 5 seconds; then decelerates to zero. The motor then accelerates to 5 rev/sec in the CCW direction and remains there.

---

### IF If

Syntax {a}IFx system

The IF command executes the command following it if the specified trigger is present, otherwise it skips the next command. If the "IF" command is inside of a loop, the command following the if cannot be a XR command. The "XR" command is not allowed inside of a loop. The following example shows an example of an "IF" command in a stored program. When stored program 1 is run the drive moves 100000 steps then checks to see if trigger 1 is present, if it is not, it calls program 1 (a continuous repeat) if trigger 1 is present it then jumps to program 2. The command should only be used in stored program.

example

XD1 A5 V5 D100000 G IF1 XR2 XR1 g

---

### K Kill

Syntax {a}K system

This is an emergency stop; acts like a software limit. The command stops the motor and terminates sequence or loop that is in progress. After a kill is issued all movement parameters must be issued for the next movement.

L Loop  
Syntax {a}Lxxxx system

The L signifies the start of a loop; the commands between the L and the N command will be repeated 1 to 64000 times. Note a loop of one is a way to put in a complex command with multiple goes and prevent execution from starting until the N is received. A loop can be terminated by issuing a "K", "Z", or "S" command.

Example:  
L3 MPI A5 V12 D50000 G D-100000 G N

The motor moves 50000 steps CW then 100000 step CCW at 12 RPS this move is done a total of 3 times.

---

LD Limit Disable  
Syntax {a}LDx Parameter

The system powers up with all limits enabled. The system requires normal closed switches at the limits. Thus broken wires in the limit chain will cause a limit condition to be registered at the indexer. If you do not have the limits wired up, the LD3 command must be issued before the move. Nothing connected to the limit inputs is interpreted as being at a limit.

- 0= enable all limits
  - 1 = disable CW limit
  - 2 = disable CCW limit
  - 3 = disable CW and CCW limit
- 

LS(x) limit sensed

Response sent if a move is terminated by a limit (L+ or L-); the x is either a + or - indicating which limit caused the termination.

---

MC Constant velocity mode  
Syntax {a}MC system

The motor runs continuously at the velocities specified until a zero velocity command is received. The CV commands are used in conjunction with this command to create motor profiling. The CA command can also be used to change accelerations between velocities. See also CTM and CTR for time and trigger options. Once a zero velocity is reached a new "G" command must be entered to initiate movement.

Example:  
MC A10 V5 CV2 CA40 CV15 CV0 G

Accelerates to 5 RPS at an acceleration of 10 rev/sec<sup>2</sup> then decelerates to 2 RPS, changes acceleration rate to 40 and accelerates to 15 RPS and then decelerates to zero.

---

MPI Incremental position mode  
Syntax {a}MPI system

Position mode, moves a specified distance. All moves reference the distance specified with respect to the motor position prior to the start of the move.

---

MPA Absolute position mode  
Syntax {a}MPA system

Moves to the specified position relative to home position. A distance parameter must be given for each execution of a MPA move; otherwise no move will occur.

---

N End of loop  
Syntax {a}N system

Signifies the end of a loop; all commands between a "L" and here are repeated the number of times specified by the "L" command.

L3 MPI A5 V12 D50000 G D-100000 G N

The motor move 50000 steps CW then 100000 step CCW at 12 RPS this move is done a total of 3 times.

---

O Output  
Syntax {a}Oxx parameter

Turns output 1,2, or 3 on or off. The first digit signifies the output and the second digit program the output on (1) or off (0). These commands are executed as soon as encountered. Their system interaction is similar to the TR (trigger) and T(time) commands.

O11 turn output 1 on  
O10 turn output 1 off  
O21 turn output 2 on  
O30 turn output 3 off

MC A5 V5 O11 G

Turn on output 1 then accelerate to 5 rev/sec at 5 rev/sec<sup>2</sup> in the continuous velocity mode.

---

OSBX home polarity, edge and backup qualifiers  
Syntax {a}OSBx

0 sensor = 0 (i.e. home position indicated by break in home connection input), stops at sensor moves off sensor in CW direction and then comes back at 0.1 CCW.

1 sensor = 1 ,stops at sensor moves off sensor in CW direction and then comes back at 0.1 CCW.

2 sensor = 0 ,stops at sensor moves off sensor in CCW direction and then comes back at 0.1 CW.

3 sensor = 1 ,stops at sensor moves off sensor in CCW direction and then comes back at 0.1 CW.

---

PS Pause  
Syntax {a}PS

Stops any further commands from being processed until a Continue (C) is received

---

PR Report motor position  
Syntax {a}PR system

Reports position with respect to the home position. The response has a range of +- 32,000,000.

---

PZ Position zero  
Syntax {a}PZ system

Set absolute position counter to zero. Establishes the present motor position as zero.

Response:

\*{a}:nnnnnnn

---

RA Report Status  
Syntax {a}RAx system

Status report sent as 2 hex values

- 1) b0 limit+
- b1 limit-
- b2 home
- b3 trig1 input
- b4 trig2 input
- b5 trig3 input
- b6 trig4 input



cancel it.

MPI A5 V5 D256000 RS1 G

Example:

A position move is performed where the drive accelerates to 5 rev/sec and moves 2560000 steps unless trigger 1 occurs. The occurrence of trigger 1 causes the drive to decelerate to a stop.

---

RV Report revision

Syntax {a}RV

Reports the sign-on message which includes the software revision. You should know this before calling the factory for assistance.

---

S Stop

Syntax {a}S system

Stop, decelerate to a stop at the current acceleration rate. This is an immediate command and takes precedence over other commands the system may be executing. It will terminate loop or sequence moves in process. This command inside of a loop or sequence will terminate it upon its encounter. After a stop is issued all movement parameters need to be reentered for the next movement.

Example:

L2 MPI A20 V5 D50000 S G N

This movement will never be executed since the stop command causes the termination of the loop.

---

SCA Automatic standby mode

Syntax {a}SCAx parameter

SCA0 disable auto-standby

SCA1 enable auto-standby

The automatic standby feature is enabled with a "SCA1" and disabled with a "SCA0". After the motor is stopped for 2 seconds the current is reduced to 1/2 of its nominal value; The current automatically goes to its nominal value when a go command is received. This will reduce motor heating but affects motor position. Reducing the current may cause a 5 to 10 step position error. If retention of the exact motor position is necessary this command should not be used.

SNx Scan sequence  
Syntax {a}SNx parameter

Scans the sequence inputs and debounces the inputs with the time specified. Values of debounce time are 0 to 5.00 sec. Time is in 10 msec. increments. The recorded value is shown in the RB command.

SN113  
Sequence input must be steady for 1.13 seconds before it is recognized as valid and is executed.

---

SSAx Serial communication echo  
Syntax {a}SSAx parameter

Enables and disables the echoing of characters received over the serial link. On power up the echo is on. A "1" turns the echo off. The echo must be on if more than one drive is on the serial link. All characters are echoed back as upper case.

SSA0 = echo on  
SSA1 = echo off

---

STx Shut down;  
Syntax {a}STx parameter

0 = torque on  
1 = shut down

Sets the motor current to zero. This removes holding torque from the motor. The motor can now be easily rotated for manual positioning.

---

SZ Step size  
Syntax {a}SZx parameter

0 = 25600 steps per revolution  
1 = 25400 steps per revolution  
2 = 20000 steps per revolution

Sets the number of steps per revolution to a binary (25600) multiple, metric (25400) multiple, or a decimal (20000) multiple. This eases the calculations for number of steps vs. distance moved when used with lead screws. For a lead screw with 1 turn per inch and the motor set to 20000 step/rev each

step would equal 0.00005 inch (pitch of lead screw/steps/rev). The same lead screw used with 25400 steps would yield 0.0001 cm/step ((pitch of lead screw/steps/rev)\*2.54 cm/in)).

---

T Time delay  
Syntax {a}Txx.xx parameter

The program delays this many seconds prior to performing the next command and prior to the execution of the go command. The time is in 10 msec. increments. Used to put delays in a movement or to allow completion of an external event before executing the next command. A "T" command preceding a go is executed prior to start of movement, it is not place dependent and the time starts as soon as the "T" command is received.

Example:

1) MPI T5 A5 V10 D2750 G

Delay 5 seconds then move 2750 steps.

2) MC T5 V10 A5 CV5 G

Delay 5 seconds, then accelerate to 10 rps, then decelerate to 5 rps.

---

TEST Test drive/motor  
Syntax {a}TEST parameter

This command runs the motor continuously in the clockwise direction at 0.1 rev/sec. The motor should not be hooked to anything at this time since the limits are not active. To disable issue reset "Z" command.

---

TR Wait for trigger  
Syntax {a}TRx parameter

Waits for the trigger input to go active low before proceeding. The trigger input is specified by the first character after the "R" character. This is an immediate type of command and is executed as soon as seen.

Example:

MC A5 V10 TR3 CV2 G

The system waits for trigger input 3 to go low; accelerates to 10 rev/sec and then decelerates to 2 rev/sec.

TS Trigger distance stop  
Syntax {a}TSxxxxxx

If trigger 1 occurs during the a distance move that contains this command, the distance move is changed to the distance specified in the TS command with reference to the trigger point. If the trigger does not occur the distance is not modified. This could be used to compensate for belt slippage or stretching on a conveyor.

Example:

MPI A5 V5 D256000 TS50000 G

The motor will move 256000 steps unless trigger 1 occurs. If trigger 1 occurs the drive will move 50000 steps from that point.

---

U Upload all stored programs  
Syntax {a}U parameter

The response to this command is to send back the program # and contents of that program for all programs. Unused location have the program # followed by a blank line. This is a listing of all programs.

---

VXX.XXX Velocity command  
Syntax {a}Vxx.xx parameter

Motor speed in revolutions per second. Acceptable parameters are from 0.001 to 38.000 revolutions per second.

---

VC Velocity Change  
Syntax {a}VCxx.xx parameter

Changes velocity when running in the constant velocity mode. This is an immediate command and will override an existing velocity command in progress. It does not change the original default velocity setting. Acceptable parameters are from 0.001 to 40.00 revolutions per second.

Example:

Assume the following command sequence is being executed and while the system is decelerating from 5 to 2 RPS a VC10 is received. The system will immediately start accelerating to 10 RPS. Once 10 RPS is reached it will then go on to 15 RPS.

MC A10 V5 CV2 CV15 G

---

VS Velocity secondary  
Syntax {a}VSxx.xxx

rev 2.23

Sets the velocity for the second half of a dual profile position move. This command is used in conjunction with the V, D, and DS commands.

Example:

Do a move which involves a high speed move 15 rev/sec to get a part in position and then move it slowly at 1 rev/sec for 50000 steps to inspect it. the total distance is 250000 steps.

MPI A100 V15 D250000 DS50000 VS1 G

---

WV Select wave shape  
Syntax {a}WVx system

- 0 pure sine wave
- 1 2% harmonic distortion
- 2 -2% harmonic distortion
- 3 4% harmonic distortion
- 4 -4% harmonic distortion

Used to compensate for smoothness of motion due to difference of motor parameters. Default is 0 (pure sine wave).

---

XC Request check sum  
Syntax {a}XCx system

Response is the check sum for requested sequence. This can be used to check against the check sum when the sequence was made as a verification of program integrity. The check sum is the sum of the data in memory.

**XD** Define sequence  
Syntax {a}XDx system

Start sequence definition, any previous commands in the buffer are erased and the commands following this to the XT command are stored in battery backed RAM. (Note a XT must be issued. All entries are put into the buffer until the xt command is received.) Error checking is not done on the entries but illegal commands will not be executed when the sequence is executed. Verify your entry with the XU Command.

Example:

XD1 MPI A33 V10 D25600 G D-25000 G XT

A move that will go 1 rev. clockwise and then 1 rev. counter clockwise is stored in battery backed RAM under sequence number one.

---

**XP** Power-up sequence  
Syntax {a}XPx system

Execute the sequence specified on power-up. The sequence specified (0 - 7) will be executed on power up. The sign-on message will not be displayed when a power up sequence is run. The Command XP8 disables the power-up sequence.

XP2 will tell the system to run sequence 2 after power-up or reset.

---

**XR** Run sequence  
Syntax {a}XRx system

Execute the sequence in battery backed RAM. Note It is "not" legitimate to call a sequence from inside a loop. Do not put a XR command inside a sequence prior to rev 2.13 of software.

XR1 will run sequence #1

XT End of sequence.  
Syntax {a}XTx system

This stores the commands from the XD to this command in battery backed RAM. See XD Command for example.

---

XU Upload sequence  
Syntax {a}XUx parameter

Response is the program in sequence "x" where x is a number from 0 to 7. The stored sequence does not contain the XD and XT commands. See XD command for sequence details.

---

Y TerminateLoop  
Syntax {a}Y system

Terminates loop; a loop that is in process is terminated at the end of the present loop. This command sets the loop counter to zero.

---

Z Reset  
Syntax {a}Z system

Software equivalent of power-up initialization sequence. This will terminate any motion in process and initialize all variables.

## Trouble Shooting

System does not echo characters.

- 1) Power up your terminal and then power up the PWX drive. The drive will send out the software revision on power up. This should be displayed on your terminal if the drive is not running power-up sequence.
- 2) Check the baud rate, # of data bits, and # of stop bits on your terminal. They should be 9600, 8 and 1 respectively.
- 3) Check that wiring is per figure 4 of manual; the RS232 ground must be connected to ground on your system.
- 4) Verify power LED on drive is illuminated.

System echoes back characters but does not move.

- 1) Verify the motor has torque on it. If it does not, turn off power and check the wiring between the motor and drive.
- 2) Verify that the error light is not on. If it is, turn off and back on to see if this clears the condition. If it comes back on, verify the heat sink is not over 50 degrees centigrade. Check the AC line voltage. Turn off the power and unplug the motor and turn back on; if this corrects the problem your motor is shorted.
- 3) Verify that you are not at a limit (get status with RA command).
- 4) Verify that you are using the correct system ID number.
- 5) Issue "RB" status command, if status is not returned issue "E" command, then issue "C" command.
- 6) Verify the drive isn't waiting for trigger, or executing time command. Use "K" command to clear those conditions.

You entered a stored program and the system quit working

- 1) You forgot to terminate the stored program with an XT. All commands are still going to the program buffer. Issue a XT to terminate.
- 2) Turn Power off and back on. Verify that all Loops in a stored program have terminating character "N".

The motor does not appear to move the correct distance for the number of steps requested

- 1) Issue the RA4 command and verify the drive is programmed for the steps/revolution you want
- 2) Note reset (hardware or software) initialize the system to 25600 steps per revolution.

The motor runs the wrong direction in the distance mode.

- 1) Check the wiring between the motor and driver; verify the phases are wired correctly.
- 2) Check if you have distance and an "H" command in the same command statement. If the h command comes after the distance command the movement will be in the opposite direction of the "D" direction; if the "D" command follows the "H" command the movement will be in the direction of the "D" command.

## Drive Operation

Limit sensors Limit sensors terminate movements and programs. During a distance move the limit also reset the distance parameter to zero.

Stop command The stop command is considered an abnormal abort of a movement and will set the distance parameter to 1000. It is recommended that a CV0 be issued to stop the motor in the constant velocity mode.

DN The drive will issue this when it considers a move complete. A move is considered complete when the motor comes to a stop.

Reset Power on reset or the "Z" command initializes the system to the MPI mode, step size equal to 25600, wave shape to zero, limits to enabled, distance to 0, auto shutdown to the DIP switch setting, and velocity to 0.

## PWX Drive

### Command Format

PWX indexer command format is composed of ASCII characters. Each command is composed of 4 sections in the following form. [Device Address][Command][Parameters][Delimiter]

Device address: single character from 0 to 7

Command: a 1 to 3 digit command beginning with a alpha character

Parameter: numeric value for command (i.e. distance, velocity or acceleration)

Delimiter: space or carriage return to signify the end of the command

### Command Summary

A(num)	Set acceleration (0.01 to 99.99 rev/sec <sup>2</sup> )	OSB (num)	Home sensor qualifier (num ) determines polarity and reference edge of home sensor
AC(num)	Changes acceleration in continuous velocity mode (num) = 0.01 to 99.99	PS	Pause; inhibits further commands from being processed until Continue is received
C	Continue enables program execution after a PS "pause" command	PR	Report position, responds with a ASCII value from +32,000,000
CA(num)	Changes acceleration in a predefined move (num) = 0.01 to 99.9 rev/sec <sup>2</sup>	PZ	Zero absolute position counter
CO(num)	Output continuous velocity; turns output num(1-3); on(1) or off(0)	RA(num)	(num)=1 Report hex value limit and trigger (num)=2 Report hex value, pause, trigger, limit, sequence, loop (num)=3 sequence inputs hex value 0-7 (num)=4 step size and wave shape selection
CTM(num)	Constant velocity time delay num = 0.01 to 99.99 seconds	RB	System activity
CTR(num)	Trigger constant velocity mode; (num) trigger input	RC	Returns current position; responds in hex
CV(num)	Changes velocity in a predefined move (num) = 0.001 to 37 rev/sec	RS(num)	Registration trigger stop. num = 1 stop on trigger, 2= do not stop
D(num)	Distance number of indexer steps to move (num) = + 32,000,000 counts	RV	Returns sign-on and software revision
DC(num)	Deceleration value num = 0.01 to 999	S	Stop; immediately decelerate motor and stop
DD(num)	Disable done ; 0= send DN response when motor stops; 1= do not send response	SCA(num)	Automatic Standby Mode. (num)=0 motor current always at full value. (num)=1 motor current to 1/2 after 2 second of no movement
DN	Done response sent from indexer when move is complete; (if enabled)	SN(num)	Debounce sequence inputs for num = 0 - 5 seconds
DS(num)	Secondary distance num = 1 to 32000000	SSA(num)	Echo on =0; off=1
E	Enable serial communications	ST(num)	Shut down; sets motor current to zero 1= off , 0 = on
F	Disable serial communications	SZ(num)	Step size (num)=0 to 2; 25600, 25400, or 20000 steps/revolution
G	Go command initiates start of movement	T(num)	Time delay (num) = 0.01 to 99.99 seconds
GH(num)	Go home with velocity specified (num) = 0.01 to 50.00 rev/sec	TEST	Test. Run at 0.1 rev/sec continuously
H	Changes direction	TR(num)	Wait for trigger
IF(num)	If trigger (num) input active execute next command	TS(num)	Trigger registration distance. num = 1 to 32000000
K	Kill terminates any move immediately	U	Upload all stored programs
L(num)	Loop from L to N (num)= 1 to 64000	V(num)	Velocity, range = .001 to 37.00 rev/sec
LD(num)	Limit disable, (num)=0 to 3 to select limit	VC(num)	Change velocity; interactive command for continuous mode
LS	Limit sensed; response sent when limit encountered and limits are enabled	VS(num)	Velocity secondary num = .001 to 38.00
MC	Mode Continuous, runs constantly at specified velocity	WV(num)	Waveform select (num) = 0 to 2 selects % of 3rd harmonic correction for current waveform
MPI	Incremental Mode; distance commands are relative to the present position	XC(num)	Request check sum; sends check sum for program (num) = 0-7
MPA	Absolute Mode; all distances are relative to the home position	XD(num)	Start sequence definition (num)=0-7
N	Loop command delimiter	XP(num)	Run sequence (0-7) on power-up. num = 8 disable power-up seq.
O(num)	Output immediate; turns output num(1-3); on(1) or off(0)	XR(num)	Run sequence (num) = 0-7
		XT	End sequence and save
		XU(num)	Upload sequence
		Y	Terminate loop when N is next encountered
		Z	Reset, equivalent to power up

# Precision Motion Controls

2530 Berryessa Rd. #209 San Jose, CA 95132

Tel : (408) 298-0898

Fax: (408) 298-0899