

Compumotor PDS15-E
Step / Direction Drive



\$1850.00

In Stock

Qty Available: 1

Used and in Excellent Condition

Open Web Page

<https://www.artisanng.com/62612-2>

All trademarks, brandnames, and brands appearing herein are the property of their respective owners.



Your **definitive** source
for quality pre-owned
equipment.

Artisan Technology Group

(217) 352-9330 | sales@artisanng.com | artisanng.com

- Critical and expedited services
- In stock / Ready-to-ship

- We buy your excess, underutilized, and idle equipment
- Full-service, independent repair center

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

CE
PDHX-E Series Drive
User Guide

(PDHX15-E, PDHX15E-D)

***For engineering
assistance in Europe:***

Parker Hannifin plc
Electromechanical Division - Digiplan
21 Balena Close
Poole, Dorset
England, BH17 7DX
Direct Lines for Technical Support
Tel: 01202-699000 Fax: 01202-695750
E-mail: tech.help@digiplan.com

Part No: 1600.209.01 September, 1996

***For engineering
assistance in the U.S.:***

Parker Hannifin Corporation
Compumotor Division
5500 Business Park Drive, Suite D
Rohnert Park, CA 94928
USA
Telephone: (800) 358-9070
Fax: (707) 584-3793
FaxBack System: (800) 936-6939
BBS: (707) 584-4059
E-mail: tech_help@cmotor.com

IMPORTANT INFORMATION FOR USERS

Installation and Operation of Digiplan Equipment

It is important that Digiplan motion control equipment is installed and operated in such a way that all applicable safety requirements are met. It is your responsibility as an installer to ensure that you identify the relevant safety standards and comply with them; failure to do so may result in damage to equipment and personal injury. In particular, you should study the contents of this user guide carefully before installing or operating the equipment.

The installation, set-up, test and maintenance procedures given in this User Guide should only be carried out by competent personnel trained in the installation of electronic equipment. Such personnel should be aware of the potential electrical and mechanical hazards associated with mains-powered motion control equipment - please see the safety warning below. The individual or group having overall responsibility for this equipment must ensure that operators are adequately trained.

Under no circumstances will the suppliers of the equipment be liable for any incidental, consequential or special damages of any kind whatsoever, including but not limited to lost profits arising from or in any way connected with the use of the equipment or this user guide.



SAFETY WARNING

High-performance motion control equipment is capable of producing rapid movement and very high forces. Unexpected motion may occur especially during the development of controller programs. **KEEP WELL CLEAR** of any machinery driven by stepper or servo motors. Never touch it while it is in operation.

This product is sold as a motion control component to be installed in a complete system using good engineering practice. Care must be taken to ensure that the product is installed and used in a safe manner according to local safety laws and regulations. In particular, the product must be enclosed such that no part is accessible while power may be applied.

EMC INFORMATION

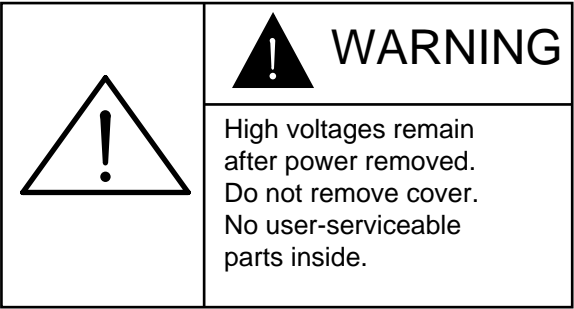
EMC Information is presented in boxed paragraphs (such as this one). Digiplan cannot guarantee compliance unless guidelines are strictly followed.

The information in this user guide, including any apparatus, methods, techniques, and concepts described herein, are the proprietary property of Parker Digiplan or its licensors, and may not be copied, disclosed, or used for any purpose not expressly authorised by the owner thereof.

Since Digiplan constantly strives to improve all of its products, we reserve the right to modify equipment and user guides without prior notice. No part of this user guide may be reproduced in any form without the prior consent of Digiplan.

© *Digiplan Division of Parker Hannifin plc, 1996*
– *All Rights Reserved* –

The following Warning label is fitted to the drive:



Symbols used on the PDHX-E Series of drives have the following meanings:



Refer to the accompanying documentation



Protective conductor terminal



Risk of electric shock



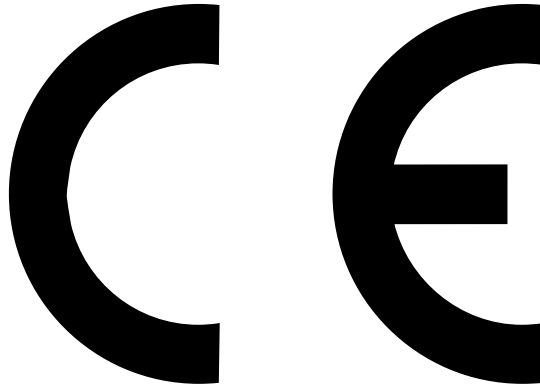
Alternating current



Hot surface



Frame or chassis terminal



Product Type: PDHX15E, PDHX15E-D

The above product is in compliance with the requirements of directives

- **89/336/EEC** **Electromagnetic Compatibility Directive
as amended by Directive 92/31/EEC**

The product is intended for use in the Commercial, Light Industrial and Industrial Environments as defined in the relevant EMC standards.

This product is compliant with the Low Voltage Directive.

- **73/23/EEC** **Low Voltage Directive**
- **93/68/EEC** **CE Marking Directive**

Table of Contents

Section 1. INTRODUCTION	1
Section 2. GETTING STARTED	3
Section 3. INSTALLATION	9
Section 4. HARDWARE REFERENCE	31
Section 5. BASIC MOTION CONTROL CONCEPTS	33
Section 6. COMMUNICATING WITH THE POSITIONER.....	43
Section 7. PROGRAMMING	61
Section 8. MAINTENANCE & TROUBLESHOOTING	85
INDEX	91

Associated Documentation

This User Guide forms part of the documentation required to use the PDHX-E Series of stepper drives, it should be read in conjunction with the **X150/X150E Software Reference User Guide** (Part Number 1600.221.01) which provides details of individual commands.

User Guide Change Summary

When a user guide is updated, the new or changed text is differentiated with a change bar in the outside margin (this paragraph is an example). If an entire section is changed, the change bar is located on the outside margin of the section title.

This is the first issue of the PDHX-E Series Drive User Guide.

Section 1. INTRODUCTION

Product Description

The PDHX15E is a single-axis packaged ministep drive incorporating an X150E position controller. The PDHX15E-D is also available with a built-in regenerative dump circuit. The drive is a chopper-regulated MOSFET design offering high performance in low and medium power applications. The recirculating chopper regulator improves operating efficiency, minimizes power consumption and reduces motor and drive heating. The drive is powered Direct On-Line (DOL) from nominal mains supplies of 110V to 240V AC. An internal switch mode power supply is used, incorporating power factor correction to minimise distortion of the AC supply.

The PD-E series of drives have 4 selectable resolutions between 400 steps/rev and 4000 steps/rev, set using the front panel switches. They are suitable for use with EMC compatible hybrid and permanent magnet stepping motors having 4, 6, or 8 leads.

Motor short-circuit protection is assured across and between phases and between any phase and earth. If a wiring fault occurs the drive fault LED will be lit up - see ***Maintenance and Troubleshooting***.

The drives can be used as stand alone units with separate control inputs and motor connection/power inputs. The built-in controller performs position control and indexing functions using an industry-standard RS232C interface, with RS485 communication as an option. The programming language is based on Digiplan's X-code and the positioner is capable of storing and executing complex motion programs from its non-volatile memory.

The drive and integral power supply are contained in one compact enclosure, cooled by natural convection. Wall or panel vertical mounting is recommended to allow access to the front panel connectors and controls.

EMC Directive compliance has been achieved through careful product design and does involve the use of braided screen cables for the motor feed and controller inputs. However, to maintain convenient interfacing no special cable screening requirements are necessary for the user I/O connector, which may use standard equipment wire.

Section 2. GETTING STARTED

What You Should Have

Upon receipt, you should inspect your PDHX-E Series Drive system for obvious damage to its container. Report any damage as soon as possible. The items listed in Table 2-1 should be present and in good condition. To verify that you have the proper drive model, check the model number listed on the drive serial plate.

Ship Kit Table

Part Description	Part Number
PDHX-E Drive	See drive identification below
PDHX-E User Guide	1600.209.XX
Drive connectors and ferrite absorber kit	PDHXE/FITKIT

Table 2-1. PDHX-E Drive Ship Kit

Identification

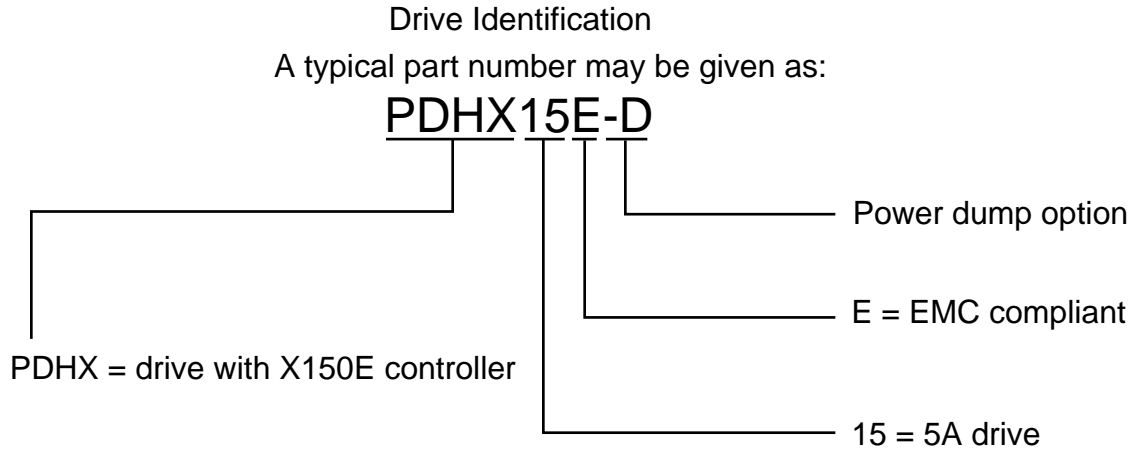


Figure 2-1. Drive Identification

Pre-installation Test

This section provides procedures to help you to connect up your PDHX-E drive system for a pre-installation test. Please note the pre-installation test allows you to become familiar with the operation of the drive prior to permanent installation, it should only be set up and used by competent personnel familiar with installation of motion control equipment.

Figure 2-2 illustrates the pre-installation test configuration for systems operating from a 230V mains supply. Please note the following:

- The motor should be securely clamped to the test bench before any power is supplied.

Minimum RS232 Connections

You will need to make a connection between a terminal and the drive. The simplest connection that can be made is using a RS232 communications link as detailed below.

Drive RS232/RS485 9-way D-type Connector	Computer 9-way D-type	Computer 25-way D-type
3 0V	5	7
4 RxA	3	3
5 TxA	2	2

Table 2-2. Minimum RS232 Connections

Note: The RS232 connections need to be made using a braided screen cable with metal back shells on the D-type connectors to maintain EMC compatibility.

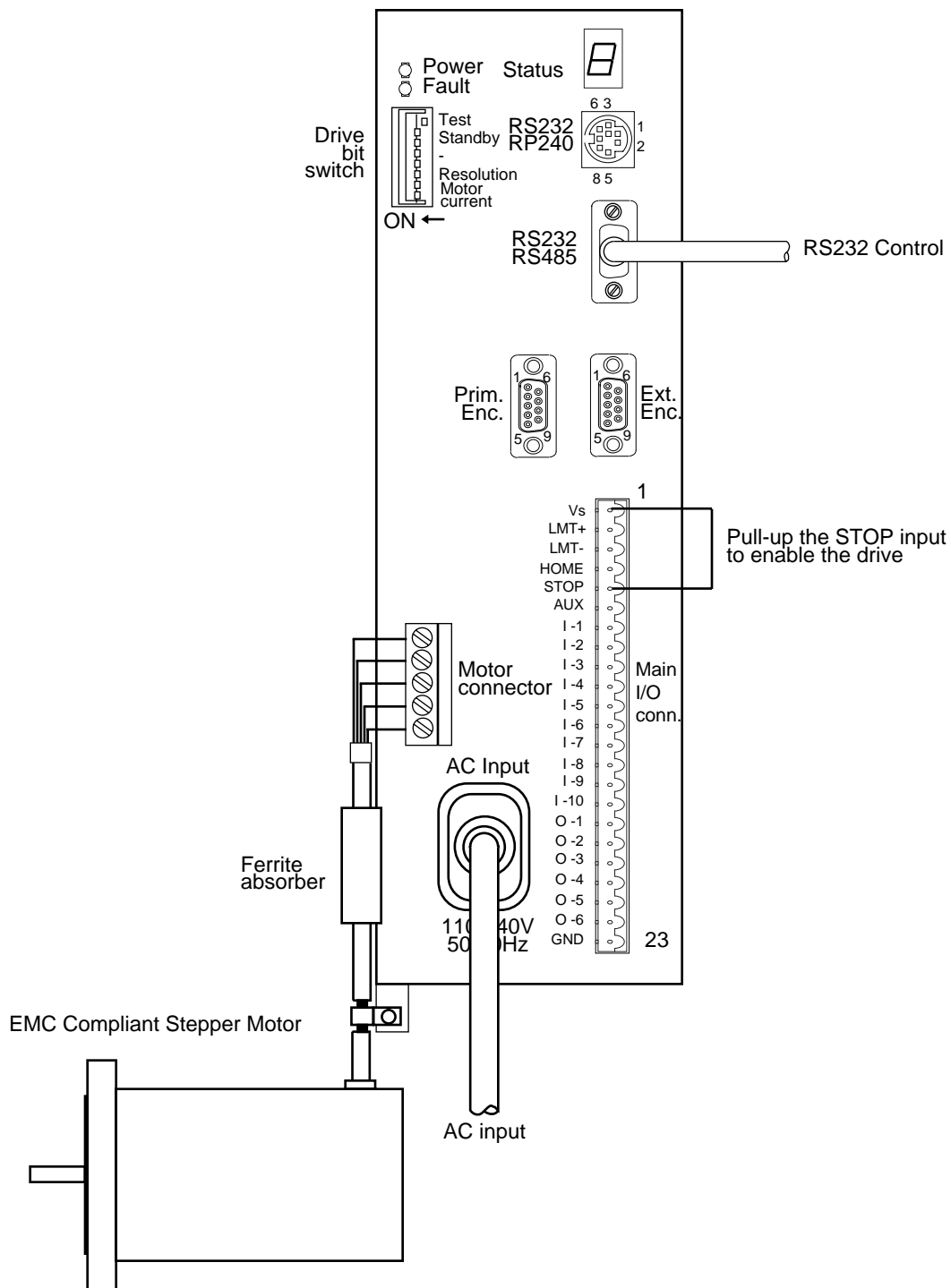


Figure 2-2. Pre-installation Test Configuration

Motor and AC Power Connections

1. Connect the Motor

WARNING - Electric Shock Hazard

Ensure that AC power is disconnected before attempting to connect or disconnect the motor. Lethal voltages are present on the motor connectors.

Fit the drive end motor connector to the corresponding socket, ensuring that the motor cable screen is anchored under the P-clip and the ferrite absorber is fitted in the position shown in Figure 2-2.

2. Connecting to an AC Supply

The PDHX-E drive operates direct-on-line from a 110-240V AC input, with no isolating transformer. A permanent connection, or an IEC309 connector is required to the mains supply.

DO NOT APPLY POWER TO THE DRIVE YET.

Testing the PDHX-E System

1. Ensure a link is made between Vs (pin 1) and Stop (pin 5) on the 23-way main controller connector.
2. Make the RS232 connection between the terminal and controller, as detailed above.
3. Configure the terminal to operate at 9600 baud, 8 data bits, 1 stop bit and no parity. Set the keyboard to capitals.
4. Apply power to the drive.
5. Establish serial communications by sending a 1DR comand. The drive should return a report of the setup of various parameters of the controller. If no response is obtained check the RS232 connections are made correctly.
6. Configure the drive for PDHX-E operation, by typing the following:

1RFS2

7. Correct operation of the drive can be checked by sending the following commands:

LD3<return> disable limit inputs
D 4000<return>
G<return>

-
8. Motor rotation of 1 revolution confirms that the drive is operating correctly. If the motor fails to rotate, re-check all connections. If you do not discover the cause of the problem refer to the ***Maintenance & Troubleshooting*** section.
 9. At the end of testing remove power from the drive.

Section 3. INSTALLATION

Precautions

During installation, take the normal precautions against damage caused by electrostatic discharges. Earthed wrist straps should always be worn.

Power Connections

Input power is taken directly from AC supplies via the front panel mounted IEC 3-way mains inlet socket. A mating connector is supplied with the drive. Connect the terminals marked L and N to the AC supply live and neutral respectively, and the terminal marked E to earth.

Motor Wiring EMC Requirements

You must wire the motor using 1mm² 5-core screened cable with a high-quality braided screen. Cables using a metallised plastic bandage for an earth screen are unsuitable, as there is a problem in terminating the screen in a mechanically stable manner and the screen itself is comparatively fragile - bending it round a tight radius can seriously affect the screening performance.

There must be no break in the 360° coverage that the screen mesh provides around the cable conductors. If a connector must be used it should retain the 360° coverage, possibly by the use of an additional metallic casing where it passes through the bulkhead of the enclosure. The cable screen must *not* be bonded to the cabinet at the point of entry. Its function is to return high-frequency switching current back to the drive. This may require mounting the connector on a sub-panel insulated from the cabinet, or using a connector having an internal screen which is insulated from the connector housing.

Within the cabinet itself, all the motor cables should lie in the same trunking as far as possible. They must be kept separate from any low-level control signal cables. This applies particularly where the control cables are unscreened and run close to the drive.

Motor-end connections

At the motor end, the motor cable screen must form a continuous and complete shield of the motor wires by connecting directly to the metal motor housing. It is preferable to use motors with internally mounted screw terminations, where provision has been made for a conductive gland to connect the motor cable screen to the motor body. If flying-lead motors are used, it is important that the unscreened leads are converted into a braided-screen cable within 100mm conductor length of the motor body. A separate terminal box may be used for this purpose but the braided cable screen must be properly strapped to the motor body using a suitable metal P-clip. The loop created at the motor end of the cable by the exposed motor leads, between where the cable screen is bonded to the motor body and the leads enter the motor body, must be made as small as possible. Take the drive ground connection to the motor earth stud.

Drive-end connections

Fit the ferrite absorber supplied over the motor cable. Connect the five leads in the motor cable to the motor connector as shown in Fig. 3-1. Expose a short length of the cable screen and anchor it to the drive using the metal P-clip located below the mains input connector. Locate the absorber between the P-clip and the motor connector, retaining it in place with the heat-shrink sleeving supplied. The specification of a suitable absorber is given below. Fit the insulated cover supplied with the drive over the motor connector to protect any exposed wires and screw terminals.

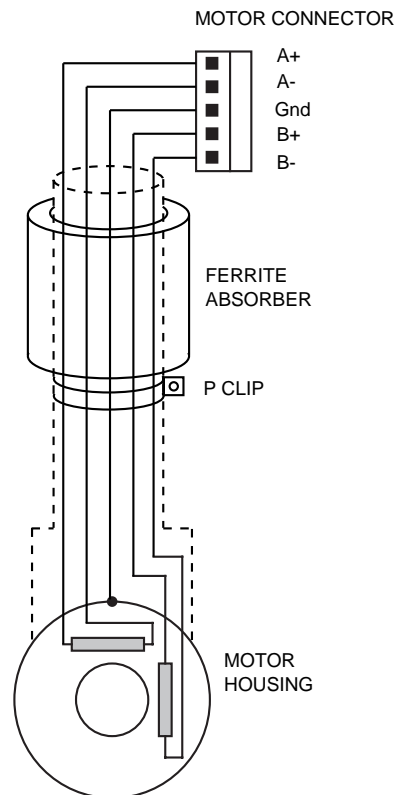


Figure 3-1 PD-E Drive Motor Connections

Ferrite absorber specifications

The absorbers described in these installation instructions are made from a low-grade ferrite material which has high losses at radio frequencies. They therefore act as a high impedance in this waveband.

The recommended components are produced by Parker Chomerics and are suitable for use with cable having an outside diameter up to 10mm. The specification is as follows:

Chomerics part number	H8FE-1115-NC
Outside diameter	17.5mm
Inside diameter	10.7mm
Length	28.5mm
Impedance at 25MHz	80
Impedance at 100MHz	120
Curie temperature	130°C (the device should not be operated near this temperature)

Handling and installing ferrite absorbers

Take care when handling the absorbers - they can shatter if dropped on a hard surface. For this reason the preferred method of installation is to use a short length of 19mm diameter heat-shrink sleeving. This gives a degree of physical protection while the cable is being installed. The sleeving should have a shrink ratio of at least 2.5:1. Cable ties may be used as an alternative, however they give no physical protection to the absorber.

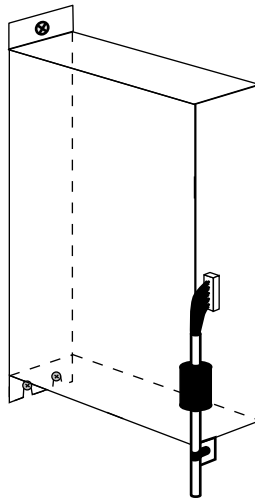


Figure 3-2. Motor wiring installation

Motor Selection

To maintain EMC compliance, the wiring guidelines given in the previous section must be followed. This should be kept in mind when the motor is selected. In particular, if the motor is fitted with flying leads or screened cable with a foil screen, this must be replaced with braided-screen cable as described.

If motors other than those listed are to be used please contact Digiplan.

Usually optimum performance will be obtained when the current rating of the motor is between 1 and 1.5 times the drive rating (refer to specification). For maximum high speed torque a motor rating of 7.5A peak should be used (PDHX15-E). The drives can be derated to accommodate motors with lower current ratings however, the high speed torque will be reduced.

Do not use a drive setting which gives an output current greater than the motor rating.

With 4 lead motors the bipolar rating is quoted and this should match the criteria stated above.

With 6 lead motors the unipolar rating is quoted, but for best performance of the PDHX-E Drives, the centre tap of each winding should be left unconnected and the connections made between the winding ends. This will give a bipolar rating 70% of the quoted motor unipolar rating.

With 8 lead motors the bipolar rating of the motor, which is normally quoted refers to a parallel winding connection. With the windings connected in series the current rating of the motor connection will be 50% that of the bipolar rating, and the motor will give improved low-speed torque, but reduced high-speed torque.

Please note that under exceptional operating conditions, for instance prolonged running at high speed, under certain fault conditions or if the drive current setting is too high, the motor temperature may become higher than normal. This may represent a safety hazard if the operator has access to the motor.

Regenerative Power Dump Option

Applications which involve rapid deceleration of high-inertia loads may require that the drive is fitted with a power dissipation circuit. The PDHX15E-D has the same electrical specification as the PDHX15E but incorporates a power dump with a continuous rating of 15 watts (170 watts peak). You will need the PDHX15E-D in the following situation:

Metric formula - if the deceleration time $t < \{J\omega^2 - 0.1\}$

where t is the deceleration time in seconds

J is the total system inertia in Kg-m²

ω is the maximum speed in revolutions per second

Imperial formula - if the deceleration time $t < \{0.02J\omega^2 - 100\}$

where t is the deceleration time in milliseconds

J is the total system inertia in oz-in²

ω is the maximum speed in revolutions per second

If the expression in brackets is negative, the power dump option is not required. This option is strongly recommended for size 42 (106) motors.

Note that a program to calculate if a power dump is required is available from Compumotor and Digiplan Technical Support departments (Windows 3.1 required). This program is also available on Compumotor's Bulletin Board Service (Tel: 707/584-4059 in USA).

Long Motor Leads

Using a motor with long leads will cause the cabling resistance to become significant when compared to the resistance of the motor. The DC volt drop of the cable and motor connection when measured at the drive, should not exceed 5 volts in order to limit power dissipation in the drive and maintain maximum system performance.

To maintain EMC compatibility, the motor leads must be assembled and connected as defined under ***Motor Wiring EMC Requirements***.

Motor Connections

Motor connection details can be determined from Tables 3-1 and 3-2. For example, Figure 3-3 shows the connection required to be made to a STEBON SDT 852-150-90 motor with its windings connected in parallel. The motor is viewed from the terminal box end with the end plate removed.

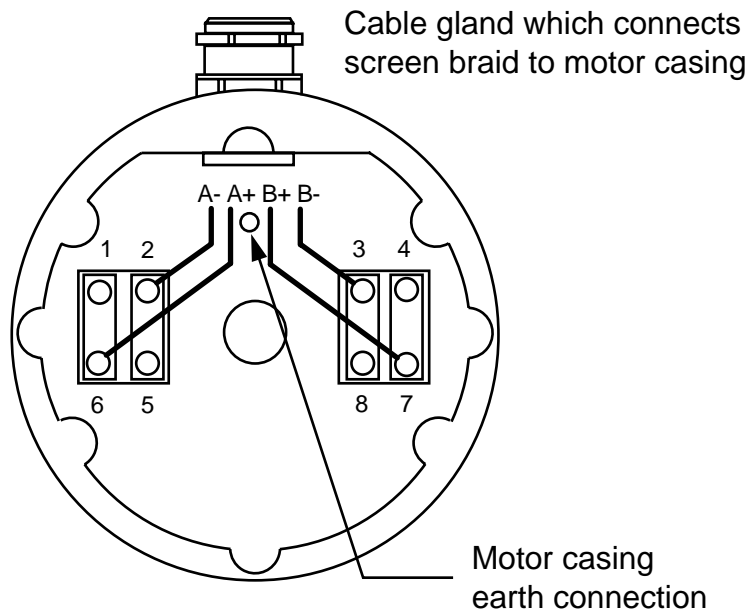


Figure 3-3 Connections for a STEBON Motor

N.C. - no connection.						
MAKE	TYPE	A+	A–	B-	B+	NOTES
Pacific Scientific	6-lead	Black	Orange	Red	Yellow	White/Blk/Org, White/Red/Yel N.C.
	8-lead	Black	Orange	Red	Yellow	Link Wh/Blk & Wh/Org Link Wh/Red & Wh/Yel
	T.box	1	3	2	4	Link 5 & 6, link 7 & 8
Slo-syn	8-lead	Red	Red/Wh	Grn	Grn/Wh	Link Black & White, link Org & Blk/Wh
	T.box (x8)	1	3	5	4	Link 2 & 6, link 7 & 8
Stebon, Digiplan SM	8-lead	Red	Yel	Pink	Blk	Link Blue & violet, link White & Grey
	T.box	1	2	3	4	Link 5 & 6, link 7 & 8
M.A.E.	6-lead	Grn/Wh	Grn	Red	Red/Wh	White & Black N.C.
	8-lead	Black	Orange	Red	Yellow	Link Wh/Blk & Wh/Org, Link Wh/Red & Wh/Yel
	T.box	6	5	8	7	Link 1 & 3, link 2 & 4
Digiplan/Compumotor OEM Series	4-lead	-	-	-	-	Internally wired in parallel
	8-lead	Red	Black	Green	White	Link blue & yellow Link orange & brown
Digiplan/Compumotor QM & S Motors	8-lead	Red	Black	White	Green	Link Yel & Blue Link Org & Brown
Digiplan/Compumotor QM & S Motors (except 106-205)	T.box	1	3	4	5	Link 2 & 6 Link 7 & 8
Digiplan/Compumotor QM & S106-205	T.box	1	3	7	8	Link 5 & 6 Link 2 & 4

Table 3-1. Motor Connection Data - Windings in Series

For 6-lead motors, connections shown are for one half-winding.

N.C. - no connection.

MAKE	TYPE	A+	A-	B-	B+	NOTES
Pacific Scientific	6-lead	Black	Wh/Blk/ Orange	Red	Wh/Red/ Yellow	Or & Yellow N.C.
	8-lead	Black & Wh/Or	Or & Wh/Blk	Red & Wh/Yel	Yel & Wh/Red	
	T.box	1 & 5	3 & 6	2 & 7	4 & 8	
Slo-syn	8-lead	Red & White	Blk & Red/Wh	Grn & Blk/Wh	Org & Grn/Wh	
	T.box(x8)	1 & 2	3 & 6	4 & 7	5 & 8	
Stebon, Digiplan SM	8-lead	Rd & Blue	Yel & Violet	Wh & Pink	Black & Grey	
	T.box	1 & 6	2 & 5	3 & 8	4 & 7	
M.A.E.	6-lead	Grn/Wh	White	Red	Black	Grn & Red N.C
	8-lead	Black & Wh/Or	Or & Wh/Blk	Red & Wh/Yel	Yel & Wh/Red	
	T.box	3 & 6	1 & 5	4 & 8	2 & 7	
Digiplan/Compumotor OEM Series†	4-lead	Red	Black	Green	White	
	8-lead	Red & Blue	Yellow & Black	Green & Orange	Brown & White	
Digiplan/Compumotor QM & S Motors	8-lead	Red & Blue	Blk & Yellow	Wh & Brn	Green & Org.	
Digiplan/Compumotor QM & S Motors (except 106-205)	T.box	1 & 2	3 & 6	4 & 7	5 & 8	
Digiplan/Compumotor QM & S 106-205	T.box	1 & 5	3 & 6	2 & 7	4 & 8	

† Size 34 only. Size 23 can only be operated in series.

Table 3-2. Motor Connection Data - Windings in Parallel

Drive Current Setting

SWITCH SETTINGS			PEAK CURRENT
6	7	8	
ON	ON	ON	5.0A
OFF	ON	ON	4.6A
ON	OFF	ON	4.3A
OFF	OFF	ON	3.9A
ON	ON	OFF	3.6A
OFF	ON	OFF	3.2A
ON	OFF	OFF	2.9A
OFF	OFF	OFF	2.5A

Table 3-3. Peak Current Settings

Compumotor S and QM Motor Drive Settings

When using Compumotor 'S' and 'QM' motors you will need to set the drive current settings as shown in Table 3-4.

The 'S' motor and 'QM' motor are electrically identical e.g. an S57-51 is the same as QM57-51. In the following table, under motor type, a suffix 'S' refers to series connected and 'P' refers to parallel connected.

Motor Type	Switch settings			Peak Motor Current (Amps)	Rotor Inertia Kg-cm ² (oz-in ²)
	SW6	SW7	SW8		
S/QM-57-51S	*	*	*	1.2	0.088 (0.48)
S/QM-57-51P	OFF	OFF	OFF	2.3	
S/QM-57-83S	*	*	*	1.5	0.234 (1.28)
S/QM-57-83P	OFF	ON	OFF	3.1	
S/QM-57-102S	*	*	*	1.7	0.32 (1.75)
S/QM-57-102P	ON	ON	OFF	3.5	
S/QM-83-62S	*	*	*	2.2	0.64 (3.50)
S/QM-83-62P	OFF	ON	ON	4.4	
S/QM-83-93S	ON	OFF	OFF	2.9	1.23 (6.70)
S/QM-83-93P	ON	ON	ON	5.6	
S/QM-83-135S	ON	ON	OFF	3.5	1.87 (10.24)
S/QM-83-135P	ON	ON	ON	6.9	
S/QM-106-178S	ON†	ON†	ON†	6.0	8.05 (44.0)
S/QM-106-178P	ON†	ON†	ON†	12.0	
S/QM-106-205S	ON†	ON†	OFF†	3.6	9.51 (52.00)
S/QM-106-205P	ON†	ON†	ON†	7.2	
S/QM-106-250S	ON†	ON†	ON†	6.2	12.14 (63.00)
S/QM-106-250P	ON†	ON†	ON†	12.4	

* Minimum drive current too high for motor.

X Unsuitable motor/drive combination.

† 106 (42) size motors must use PDHX15E-D option (regenerative power dump).

Table 3-4. PD-E Series Current Drive Settings for Compumotor 'S' and 'QM' Motors

Digiplan SM Motor Drive Settings

Table 3-5 lists the PDHX15-E Drive current settings you need to make when using Digiplan 'SM' and STEBON motors. In Table 3-5, under 'Motor Type', a suffix 'S' refers to series connected and 'P' refers to parallel connected.

Motor Type	Switch settings			Peak Motor Current (Amps)	Rotor Inertia Kg-cm ² (oz-in ²)
	SW6	SW7	SW8		
SM-57-51S	*	*	*	1.1	0.11 (0.60)
SM-57-51P	*	*	*	2.1	
SM-57-83S	*	*	*	2.3	0.23 (1.26)
SM-57-83P	OFF	ON	ON	4.7	
SM-57-102S	*	*	*	2.3	0.30 (1.64)
SM-57-102P	OFF	ON	ON	4.6	
SM-83-62S	OFF	ON	OFF	3.2	0.60 (3.30)
SM-83-62P	ON	ON	ON	6.4	
SM-83-93S	OFF	OFF	ON	4.0	1.25 (6.83)
SM-83-93P	ON	ON	ON	7.9	
SM-83-135S	ON	ON	OFF	3.8	2.00 (10.93)
SM-83-135P	ON	ON	ON	7.6	
SM-106-140S	OFF†	ON†	ON†	5.0	3.65 (19.96)
SM-106-140P	ON†	ON†	ON†	9.9	

* Minimum drive current too high for motor.

X Unsuitable motor/drive combination.

† 106 (42) size motors must use PDHX15E-D option (regenerative power dump).

Table 3-5. PD-E Series Current Drive Settings for Digiplan 'SM' Motors

Compumotor OEM Motor Drive Settings

The 34 frame size motors (OEM-83-62/93/135) have identical drive current settings to the 'QM' motors listed in Table 3-4. Size 34 (83) motors are internally wired in Parallel. In Table 3-6, under 'Motor Type', a suffix 'S' refers to series connected. The parallel connection can not be used for size 23 (57) motors.

Motor Type	Switch settings			Peak Motor Current (Amps)	Rotor Inertia Kg-cm ² (oz-in ²)
	SW6	SW7	SW8		
OEM-57-40S	OFF	OFF	OFF	2.7	0.07 (0.380)
OEM-57-51S	OFF	ON	OFF	3.3	0.12 (0.650)
OEM-57-83S	ON	ON	OFF	3.8	0.25 (1.360)

Table 3-6. PD-E Series Current Drive Settings for Compumotor 'OEM' Motors

Controller Signals

There are 5 front-panel connectors associated with control inputs and outputs as shown in the diagram. Their functions are as follows:

1. **RS232/RP240** (8-way mini-DIN socket). This socket is normally used for the RP240 Remote Operator Panel but may also be used as an alternative RS232C connection from the host controller. The RS232 connections are in parallel with those on socket 2.
2. **RS232/RS485** (9-way D-connector). This is the connector normally used for serial communication. The initial release of the X150E controller will be supplied with RS232 communication as standard, RS485 being a factory-fitted option. Subsequent issues will allow either RS232 or RS485 to be user-selected.
3. **External encoder** (9-way D-connector). An encoder fitted to this input is used for the connection of an external encoder for following applications.
4. **Primary encoder** (9-way D-connector). A motor or load mounted encoder is fitted to this input.

Note: The ICON command controls the connection of the encoder inputs. Bit 'ENC' must be set to 0 to use the encoder inputs.

5. **Main I/O connector** (23-way two-part screw terminal connector). All the user-programmable inputs and outputs, as well as registration, stop, home and limit inputs, appear on this socket.

Controller Signal Connections

When you make connections to the following ports:

RS232/RP240
RS232/RS485
Ext. Enc.
Prim. Enc.

Observe the following wiring recommendations:

Connections must be made using braided shield signal cable. A foil screen is not suitable for achieving the required EMC performance. The shield of the cable must be 360° bonded through a suitable connector housing to the mating half of the drive connector. The connectors must be conductive bright tin plated, not zinc passivated (i.e. those of a gold/yellow appearance). D-type connectors must be of the 'dimple-D' type, which offer better screening performance than conventional plain sided D-type connectors.

Encoder screened leads that are routed to the same area as the motor must be arranged to avoid creating a loop around the mains inlet cable, see Figure 3-4.

Main I/O Connector

No screening requirements are necessary for EMC purposes, standard equipment wire may be used. Although, care should be taken to avoid looping the wires around the mains inlet wire or other drive connections.

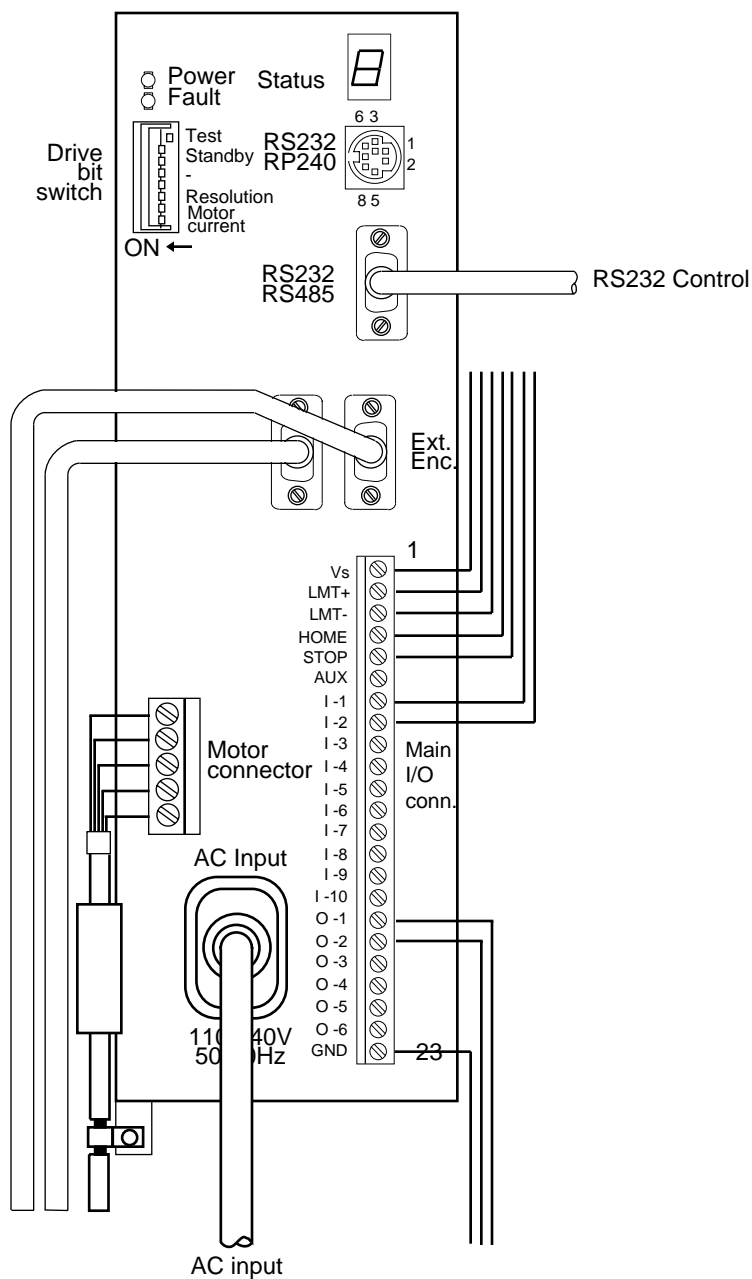


Figure 3-4. PDHX15E Wiring Layout

CONNECTOR
Ext. encoder
9-way D-type

Pin	Signal Name	Function	Signal Type
1	A+	Secondary Encoder A	C
2	A-		
3	B+	Secondary Encoder B	C
4	B-		
5	Z+	Secondary Encoder Z	C
6	Z-		
7	5V ENC.	+5V DC, 150mA max.	D
8	GND	Encoder GND	D
9	n/c		

Pri. encoder
9-way D-type

1	A+/CLK+	Primary Encoder A	C
2	A-/CLK-		
3	B+/DIR+	Primary Encoder B	C
4	B-/DIR-		
5	Z+	Primary Encoder Z	C
6	Z-		
7	5V ENC.	+5V DC, 150mA max.	D
8	GND	Encoder GND	D
9	n/c		

RS232/RP240
mini DIN

1	+5V (RP240)	+5V DC, 150mA max	
2	0V (RP240)		
3	RxB (RP240)		
4	n/c		
5	RxA (Terminal)	RS232 Receive*	E
6	TxB (RP240)		
7	Screen		
8	TxA (Terminal)	RS232 Transmitt†	E

RS232/RS485
9-way D-type

1	Rx+	RS485 Receive +	J
2	Tx+	RS485 Transmit +	J
3	0V		
4	RxA (Terminal)	RS232 Receive*	E
5	TxA (Terminal)	RS232 Transmitt†	E
6	Rx-	RS485 Receive -	J
7	Tx-	RS485 Transmit -	J
8	n/c		
9	+5V	+5V DC, 150mA max.	I

*Pin 5 of mini DIN connected internally to pin 4 of 9-way D-type

†Pin 8 of mini DIN connected internally to pin 5 of 9-way D-type

Table 3-7. Controller Signals

MAIN CONNECTOR

1	Vs	+24/5V (100mA)	DC OUTPUT
2	LMT+	+Limit Switch Input	A
3	LMT-	-Limit Switch Input	A
4	HOME	Home Switch Input	A
5	STOP	Stop Input	A
6	AUX	Registration Input	A
7	I - 1	User Input 1	A
8	I - 2	User Input 2	A
9	I - 3	User Input 3	A
10	I - 4	User Input 4	A
11	I - 5	User Input 5	A
12	I - 6	User Input 6	A
13	I - 7	User Input 7	A
14	I - 8	User Input 8	A
15	I - 9	User Input 9	A
16	I - 10	User Input 10	A
17	O - 1	User Output 1	B
18	O - 2	User Output 2	B
19	O - 3	User Output 3	B
20	O - 4	User Output 4	B
21	O - 5	User Output 5	B
22	O - 6	User Output 6	B
23	GND	24/5V GND	DC GND RETURN

Table 3-7. Controller Signals - Continued

Signal Types

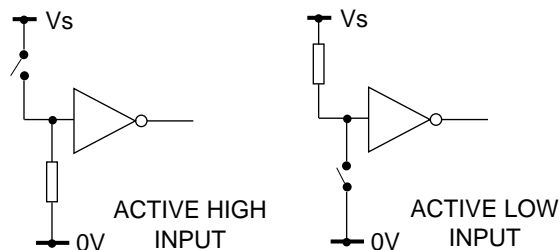
Signal Types identified by letters A, B, C, D, E, F, G and I in Table 3-7 are defined below.

Input Circuit Conventions

Inputs may be configured to be active high or active low.

An active high input has a pull-down resistor inside the controller, and must be connected to Vs to turn the input ON, i.e. make the input active. Alternatively, the input may be turned on by connecting it to an external +5V or +24V supply.

An active low input has a pull-up resistor to Vs inside the controller, and must be connected to 0V to make the input active.

**Figure 3-5. Input Circuit Configurations**

In either case, the input is OFF (i.e. inactive) when in the open circuit condition. This condition is described as the logic 0 state and will return a ‘0’ in response to the input status request IS.

Signal Type A

Apply to all dedicated and user definable inputs. Signal type A inputs can be operated at 5V or 24V and can be configured, by software, for operation as active high or active low inputs. The input circuit for “A inputs” is shown in Figure 3-6.

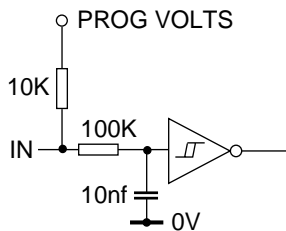


Figure 3-6. Signal Type A Input Circuit

Inputs can be programmed to operate as pull-down (sink) inputs or pull-up (source) inputs. The pull-up voltage may also be programmed as 5V or 24V. This is achieved by using the circuit arrangement shown in Figure 3-6, where PROG VOLTS can be programmed to 0V (pull-down), 5V or 24V (pull-up).

Table 3-8 defines the required settings for 5V and 24V operation and the different input characteristics between the two settings.

	5V Operation		24v Operation	
Signal type	B		A	
Input Characteristics	Active high	Active low	Active high	Active low
Logic 1 level (Input ON)	>3.0v	<2.5v	>9.0v	<5.7v
Logic 0 level (Input OFF)	<2.5v	>3.0v	<5.7v	>9.0v
Hysteresis	0.5v	0.5v	3.3v	3.3v
Max. input voltage range	0-30v	0-30v	0-30v	0-30v

Table 3-8. 5V and 24V Operation for Signal Type A Inputs

Signal type A inputs may be configured as Pull down (sink) inputs or as Pull up (source) inputs. The inputs are programmed using the ICON command. ICON is also used to set the pull up voltage level and the value of Vs supplied to the customer.

Signal Type B

Signal type B outputs can also be defined in software, using the OCON command. Outputs can be programmed as NPN or PNP.

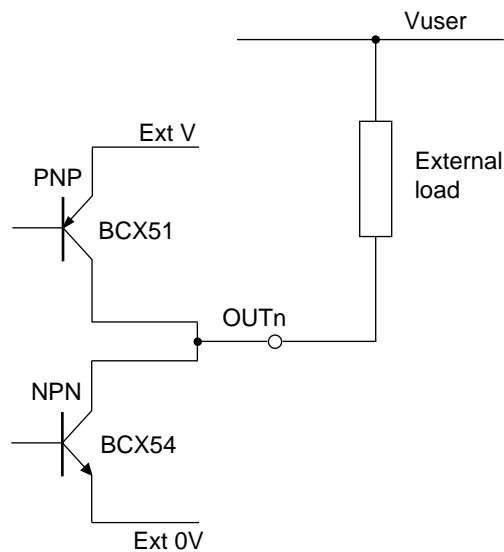


Figure 3-7. NPN or PNP Output Configuration

Output voltage levels will be determined by the value of the external voltage (V_{user}) and the value of the connected load.

Caution - Possibility of equipment damage

The output circuit uses a connected pair of output transistors, consequently when the NPN transistor is used, but is turned OFF, the external user voltage (V_{user}) must be no greater than the external voltage (V_s).

The characteristics of each output type are listed in Table 3-9.

	NPN	PNP
Output Characteristics		
Circuit configuration	Common emitter	Common emitter
Output current max.	300mA at $1V_{sat}$	100mA at $(V_S-2)V_{sat}$
Max. OFF signal voltage	30V	30V
Mode	Current sinking	Current sourcing

Table 3-9. Signal Type B Output Characteristics

Signal Type C

Signal type C inputs are used for differential, optically isolated, encoder TTL level inputs. The maximum frequency handled by these inputs is 100KHz.

Signal Type D

Encoder supply of +5V DC at 150mA.

Signal Type E

RS232 data signals.

Signal Type F

RP240 RS232 data signals.

Signal Type G

RP240 +5V DC supply (150mA max).

Signal Type I

Signal ground.

Signal Type J

RS485 data signals.

Address Switch

The address of the drive is set using an internally mounted bit switch. The switch may be accessed via the ventilation slots on the top of the drive - see Figure 3-8. Address selection is described in ***Communicating With The Positioner***.

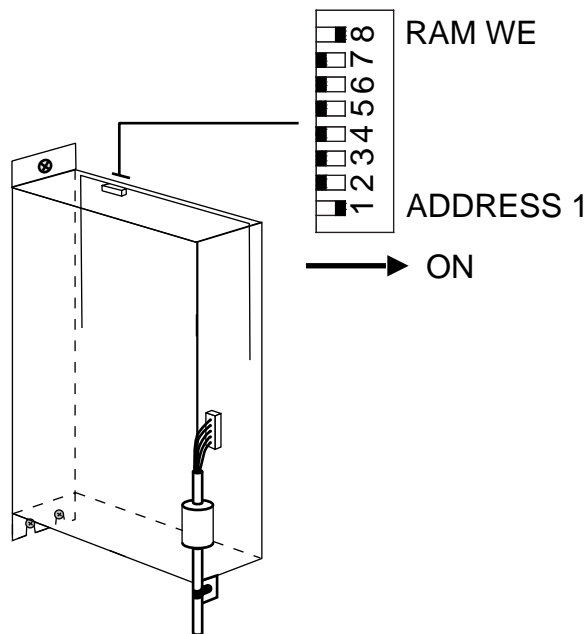


Figure 3-8. Connector and Switch Positions

Mechanical/Environmental

Enclosure Considerations

The drive, its switch mode power supply and positioner are contained in a single case measuring 292mm high, 75mm wide and 224mm deep.

Note: Enclosure depth does not take connector dimensions into account. These need an additional 60mm.

Environmental Specifications

Digiplan recommends you operate and store your PDHX Drive system under the following conditions:

- Operating Temperature: 0° to 40°C (32° to 104°F), or 0° to 50°C (32° to 122°F) if there is no user access to the case
- Relative Humidity: 0% to 95% (non-condensing)
- Storage Temperature: -40° to 85°C (-40° to 185°F)

The recommended orientation of the drive enclosure is back panel, vertical mounting.

In exceptional circumstances, such as running the motor continuously at maximum current, forced-air cooling may be needed to maintain the local ambient temperature within specification.

Installation Considerations

The drive should be installed vertically. Air vents on the top and bottom panels allow convection cooling. At least 50mm minimum clearance around the air vents is recommended for unobstructed ventilation and reliable operation.

The drive is designed for Installation Category II (see IEC 664). This means that transient voltages on the local mains supply are not expected to exceed 2,500V. The environment should conform to Pollution Degree 2, i.e. no conductive foreign matter (either solid or liquid) should be allowed to enter the case.

Setting up

Drive Switch Settings

The switch for setting the drive operating conditions is at the top left of the front panel.

SWITCH NUMBER	FUNCTION	COMMENTS
1	Selftest	Default OFF
2	Standby current reduction	Default ON
3	Not used	-
4, 5	Step resolution	Default all OFF
6, 7, 8	Peak current setting	Default all ON

Table 3-10. Drive Switch Settings

Selftest Switch1

The selftest switch is used to check the operation of the drive. Set to the ON position to cause the motor to rotate at slow speed. The default setting of selftest is OFF i.e. not selected.

Standby Switch 2

Switch 2 determines the level of standby current. With switch 2 ON, the current will reduce by 50% at standby. With switch 2 OFF, current will be reduced to 80% of the programmed value at standby.

Step Resolution Switch 4, 5

DIP Switches 4 and 5 determine the step resolution of the drive, as defined in Table 3-11. Note: cycle the power for switch settings to take effect.

SWITCH SETTINGS		RESOLUTION IN STEPS/REV
4	5	
ON	ON	400
ON	OFF	1000
OFF	ON	2000
OFF	OFF	4000

Table 3-11. Step Resolution Settings

Peak Current Setting Switch 6, 7, 8

DIP Switches 6, 7 and 8 determine the peak current setting of the drive, as defined in Table 3-12.

SWITCH SETTINGS			PEAK CURRENT
6	7	8	
ON	ON	ON	5.0A
OFF	ON	ON	4.6A
ON	OFF	ON	4.3A
OFF	OFF	ON	3.9A
ON	ON	OFF	3.6A
OFF	ON	OFF	3.2A
ON	OFF	OFF	2.9A
OFF	OFF	OFF	2.5A

Table 3-12. Peak Current Settings

Thumbwheel Interface

This section assumes that you are using Parker's TM8 thumbwheel module.

You can use up to 16 digits of the thumbwheel. The controller uses a multiplexed BCD input scheme to read the thumbwheel data. The commands and format that allow for the thumbwheel data entry are :

DRDxyz	Read distance via thumbwheel.
VRDxyz	Read velocity via thumbwheel.
LRDxyz	Read loop count via thumbwheel.
TRDxyz	Read time delay via thumbwheel.
VARDn,xyz	Read variables via thumbwheel.
XRDxyz	Read sequence count via thumbwheel.
FRDxyz	Read following ratio via thumbwheel.

To request 1 digit x=y=the desired digit number.

To request all digits x=0 and y=7 (or do not set xyz fields).

To request a block of digits $0 < x < y < 7$.

The Z field scales the thumbwheel value by 10. When reading digits from the thumbwheel, the least significant digit will be filled first. The z field allows you to position the decimal point were needed.

Note : Either all the fields (xy and z) or none of the fields must be used. Refer to the software section for further explanation.

Caution - Possibility of damage to the TM8 module

THE TM8 REQUIRES A 5V SUPPLY. THE ICON COMMAND MUST BE USED TO SELECT A PULL-UP LEVEL OF +5V OTHERWISE DAMAGE WILL OCCUR. DO NOT CONNECT THE TM8 MODULE TO THE DRIVE UNTIL YOU ARE SURE THE SUPPLY IS SET TO +5V.

Note: The TM8 module is not EMC compliant, consequently when it is used within a permanent installation EMC screening precautions must be taken. Generally, this will require the TM8 module to be mounted within the equipment cabinet housing the drive.

1. Ensure that the controller is configured with NPN outputs using the OCON command.
2. Using the ICON command, set the following conditions:
 - Pull-up level of 5V
 - Function set to pull-up
 - VOL=0 (5V DC operation)
 - ENC set to 0
3. You can define any block of 5 inputs for use with the thumbwheel. Therefore you must decide which inputs you intend to use for data (4 inputs required) and sign bit (1 input required).
4. Configure the defined inputs for TM8 operation using the INnN command.
5. Configure the defined input for the TM8 sign bit using INnW command.

6. Define appropriate outputs for TM8 operation using the OUTnJ command.

The TM8 is now ready for operation.

Example

The TM8 is wired as shown in Figure 3-9.

Inputs 6 to 9 are to be used as data inputs.
Input 10 is to be used as a sign input.
Outputs 3 to 5 are to be used as strobe outputs.

Set up as follows:

- 1. Set **1OCON 000000** all NPN outputs
- 2. Set **1ICON 00000100** bank 3 selected as pull-up
- 3. Set **1IN6N** Input 6 is defined as data input.
1IN7N Input 7 is defined as data input.
1IN8N Input 8 is defined as data input.
1IN9N Input 9 is defined as data input.
1IN10W Input 10 is defined as sign bit.
- 4. Set **1OUT3J** Output 3 is defined as data strobe.
1OUT4J Output 4 is defined as data strobe.
1OUT5J Output 5 is defined as data strobe.

The thumbwheel is now ready for use. Set the thumbwheel to the desired value and enter the information using one of the data entry commands.

For example set the thumbwheel to +12345678

Command	Description	
1DRD	Request distance data from all TM8 digits	
1D	Display the distance read (D+12345678)	If you do not receive this response, return to step 1 and retry

For further information relating to Data entry commands refer to the software section.

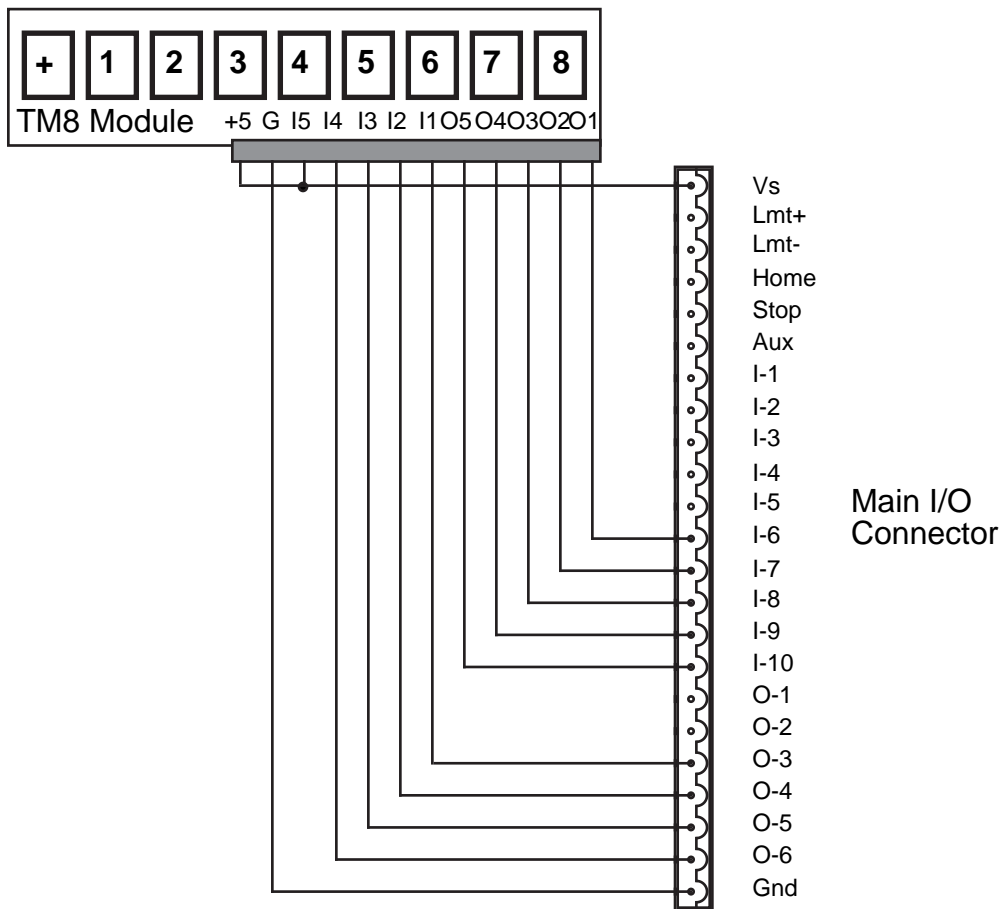


Figure 3-9. Connections to TM8 Thumbwheel Module

Using two TM8 Thumbwheels

Wire the TM8s as shown in Figure 3-10.

Configure as previously described.

Output 6 is used to select which thumbwheel is to be read.

SET **1OUT6A** Programmable output.

Enter the following commands.

Command	Description
OXXXXX0	Output 6 is low TM8 module #1 is disabled, TM8 module #2 is enabled.
DRD	Request distance data from thumbwheel #2.
1D	Display the distance read.
OXXXXX1	Output 6 is high TM8 module #1 is enabled, TM8 module #2 is disabled.

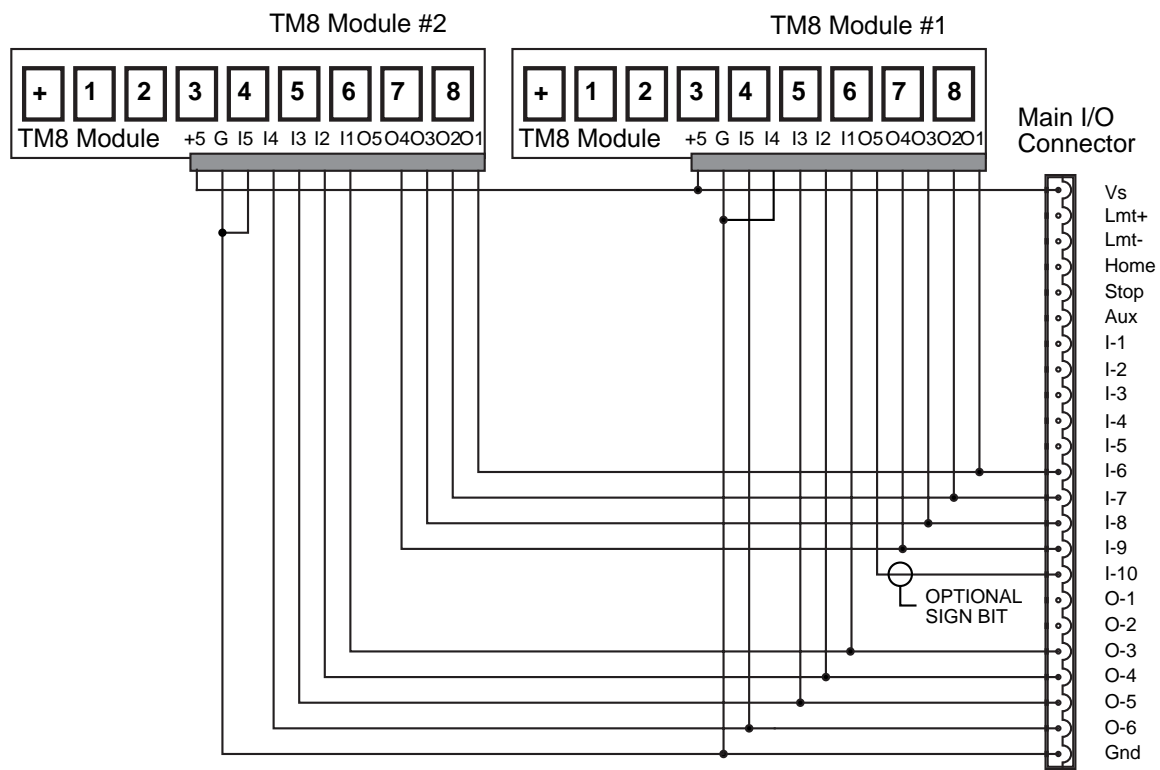


Figure 3-10. Connections to Dual TM8 Thumbwheel Modules

Section 4. HARDWARE REFERENCE

PDHX15-E Drive Specifications

Parameter	Value
Amplifiers	
Type	MOSFET Chopper
Motor resolution	400, 1000, 2000 and 4000 steps/rev (user-selectable)
Short circuit protection	Phase-to-phase, across phases and phase to ground
Peak output current	5A/phase, switch reduceable
Standby current reduction	To 80% or 50% of programmed peak value after 100ms (switch-selected)
Nominal chopping frequency	20kHz
AC Power input	
Drive supply voltage	95V to 264V AC (absolute limits)
Supply frequency range	47 to 63Hz
Power factor	Better than 0.9 over input voltage range and output power range
Maximum input power	300VA
Input current:	3A rms max
Recommended supply protection	3A MCB type 'C' characteristics
Output current range	2.5A - 5A (350mA steps)
Standby reduction	50% or 80%
Environmental	
Drive dimensions	Height 292mm, Width 75mm , Depth 224mm
Weight	2.9Kg
Operating temperature range	0° - 40°C (32° - 104°F), or 50°C (122°F) if no user access to case
Ingress protection	IP20
Max. power dissipation of drive unit	
PDHX15E	30 Watts
PDHX15E-D	45 Watts
Motors	
Type	2-Phase hybrid or permanent magnet (normally 1.8°)
Number of leads	4, 6, or 8 (5 lead not suitable)
Minimum Motor Inductance	1mH
Optimum Inductance range	1mH-10mH
Power/Motor connection	refer to Installation

Controller Specification

Parameter	Value
Command Input	RS232C (3-wire) or RS485 (2-wire, single-ended or 4-wire, differential)
RS232 Type	3-wire (Tx, Rx, Gnd). Minimum voltage swing = $\pm 3V$
Parameters	9600 baud, 8 data bits, 1 stop bit, no parity
Connector	8-way mini DIN or 9-way D-type
Configuration	Up to 32 interfaces can be controlled from a single RS232C port. Device address set up by DIL switch
RS485/422 Type	2-wire (single ended) or 4-wire (differential)
Parameters	9600 baud, 8 data bits, 1 stop bit, no parity
Connector	9-way D-type
Configuration	Up to 32 interfaces can be controlled from a single RS485/422 port. Device address set up by DIL switch
Operating Ranges	
Position	± 1 to 268,435,455 steps
Velocity	0.0001 to 200 revs/sec
Acceleration	0.06 to 999,999 revs/sec ²
Maximum Encoder Frequency	100 kHz (lines/sec before multiplication)

Resolution Ranges	
Feedback encoder	1 to 32,767 counts/rev
User-programmed	1 to 32,767 steps/rev
Co-ordinate System	Incremental or absolute
Operating Modes	Preset, preset with speed change, continuous, registration, electronic gearbox following, preset following
Motion Program Storage	
Memory Type	Battery-backed RAM
Memory Capacity	8000 characters total
Number of Programs	64
Program Length	Variable up to memory limit
Program Selection	a) Via RS232C/RS485 b) Automatic execution at power up c) Binary address on sequence select inputs d) RP240 e) Thumbwheel (TM8) f) PLC
Opto-isolated I/P's	Home, Limits, Aux-in, Stop, 10 user-definable inputs (also used for program selection)
Optically-isolated O/P's	6 programmable
Type of output	NPN or PNP (software selectable)

Table 4-1. Controller Specification

Fuses

PDHxE drives are fitted with fuses which limit circuit damage in the event of a fault occurring, they are not user replaceable. If the drive fails to operate correctly or you suspect a fuse has blown return the drive for repair. See **Returning The System** in the **Maintenance and Troubleshooting** section. Warranty is void if the case is opened.

Section 5. BASIC MOTION CONTROL CONCEPTS

Chapter Objectives

This chapter is preliminary to the programming section and describes some of the basic concepts of motion control systems in general and the PDHX-E in particular.

Motion Profiles

In any motion control application the most important requirement is precise shaft rotation, whether it be with respect to position, time or velocity. The type of motion profile needed will depend upon the motion control requirement. The following sections describe the basic types of motion profiles.

Preset Moves

A preset move referred to in this manual is a move of a specified distance (in user steps). Preset moves allow the user to position in relation to the motor's previous stopped position (incremental moves) or in relation to a defined zero reference position (absolute moves). Preset moves are selected by putting the positioner into incremental mode using the MPI command and absolute moves are made using the MPA command.

Incremental Preset Moves

If the positioner is in the incremental mode (MPI command), a preset move will move the shaft of the motor the specified distance (in user steps) from its starting position. For example, to move the motor shaft 1.5 revolutions, a preset move with a distance of +6000 steps would be specified, assuming a 4000 step per rev encoder resolution setup. Every time this move is executed, the motor shaft will move 1.5 revolutions positive from its current position. The direction of the move can be specified at the same time as the distance by using the optional sign (D+6000 or D-6000), or it can be defined separately with the H command (H+ or H-).

Absolute Preset Moves

A preset move in absolute mode (MPA command) will move the shaft of the motor the specified distance (in user steps) from the absolute zero position. The absolute position can be set to zero with the PZ or SP commands, for instance at the end of a GO HOME move (GH command). The absolute zero position is initially the power-up position, and will remain that way until changed with a PZ command. Any preset move performed while in the absolute mode will position the motor shaft the defined distance (in user steps) from the absolute zero position. For example, with the positioner at the absolute zero position a move with a distance of +4000 will cause the motor shaft to turn 1 revolution in the positive direction. If a move with the same defined distance is executed immediately after this move, the motor shaft will not turn, since it is already +4000 steps from the absolute zero position.

The direction of an absolute preset move will depend upon the shaft position at the beginning of the move and the position that it is being commanded to move to. For example, if the motor shaft is at absolute position +12,800, and position commanded is +5000, the motor shaft will move in the negative direction a distance of 7800 steps to absolute position +5000.

The positioner saves the mode that it was in at power down and powers up again in the same mode. Issuing the MPA command will set the mode to absolute. Issuing the MPI command will switch the mode from absolute to incremental. The positioner retains the absolute

position referenced, even while in incremental mode. However, the counter does have an upper limit just slightly more than 268,400,000 counts.

Continuous Moves

Continuous moves (MC command) cause the motor to accelerate and attain the specified velocity and then continue to move. To change velocity while the motor is moving use the V command. Issuing a stop (the S command) will cause the motor to decelerate to a stop at the last defined acceleration rate. The distance parameter is not used, although it is saved in case the mode is changed back to preset.

This mode is useful for applications which require constant spinning of the load, when the motor must stop after a period of time has elapsed rather than after a fixed distance, or when the motor must be synchronised to external events such as trigger input signals.

Speed Change Mode

The MQ command allows speed changes to be made during a preset move. The speed changes can be triggered at a set distance using the TRD command or by an input using the TRE command.

Registration Moves

A move may be programmed to end a specified distance after a registration pulse appears at AUX-IN. The mode is selected using the TRR command with the required registration distance in the MQ or MC modes. Its use is easiest to understand in the context of an example registration move:

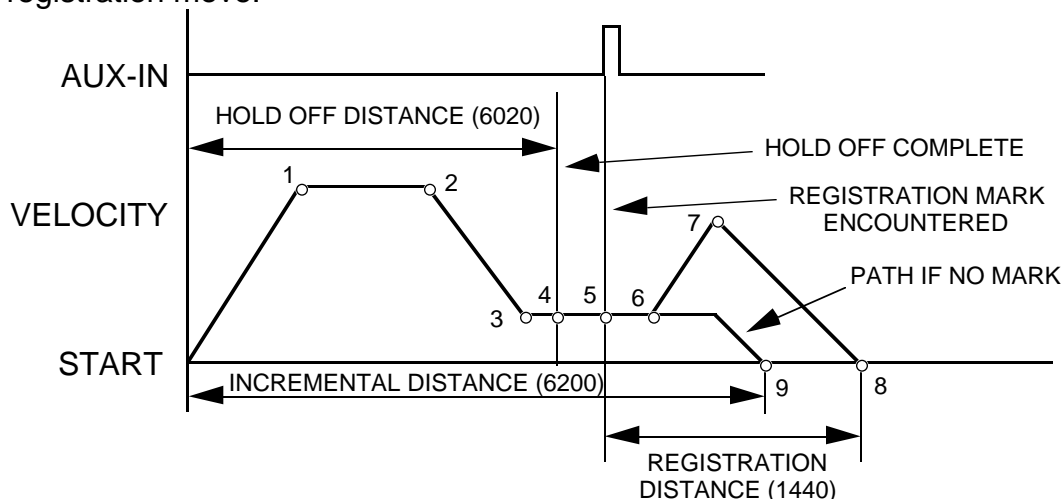


Figure 5-1. Example Registration Move

Example Program

MQ	Select MQ mode
D6200	Set move distance to 6200 steps
A100	Set all accelerations to 100 revs/sec ²
V10	Set velocity to 10 revs/sec
G	Go
TRD4000	At distance = 4000 steps
V1	Change velocity to 1 rev/sec
TRD6020	At distance 6020 look for registration pulse
TRR1440	Travel 1440 steps from registration pulse

**V50
TRIP**

At a velocity of 50 revs/sec
Set defined Output when in position

Program Operation

When executing this program, the axis will first accelerate at 100 steps/rev² to 10 revs/sec (1) and when it reaches a distance of 4000 steps (2), its speed will be reduced to 1 rev/sec (3) whilst it is waiting for the registration pulse. Shortly after the pulse is received (6), the axis will accelerate towards 50 revs/sec, but will not reach the speed before starting to decelerate (7) to stop at 1440 steps from where the pulse occurred (8).

If the pulse was not received after 6020 steps (4 - the hold off distance) and before 6200 steps (the incremental distance), the axis will stop at 6200 steps (9). This sets a window of 180 steps during which the pulse is expected.

Program Criteria

The distance command (D6200)	This command sets the distance the axis will travel if no registration pulse is received and the maximum distance at which the pulse will be recognised.
Trigger distance command (TRD4000)	The distance at which the axis will start to decelerate to the slower speed of 1 rev/sec (V1) in readiness for the registration pulse is set by this command.
Trigger distance command (TRD6020)	This command sets the minimum distance at which the system will recognise the registration pulse.
The TRR command (TRR1440)	This command sets up the registration mode and the registration distance (the distance the axis will travel after the pulse is received)
Registration move velocity (V50)	To save time in travelling the registration distance, the axis then accelerates towards 50 revs/sec (V50), but it does not reach that speed before starting to decelerate to stop at 1440 steps from where the pulse occurred.

Motion Profiles

Velocity, acceleration and distance must be defined before any preset move can be executed. The value of these parameters determines the type of motion profile as either triangular or trapezoidal.

Triangular Profile

A triangular profile will result when the velocity and acceleration are set such that the designed velocity is not attained before half of the specified distance has been travelled. This results from either a very low acceleration or a very high velocity or both over a relatively short distance. For example, if the acceleration is set to 1 rev/sec/sec, velocity is set to 5 revs/sec and distance is set to 16000 steps (2 revs), a triangular motion profile will result. This is because by the time the motor shaft has reached a velocity of 2 revs/sec, it will also

have travelled half of the defined distance due to the acceleration setting of 1 rev/sec/sec. The motion profile for this move would look like this.

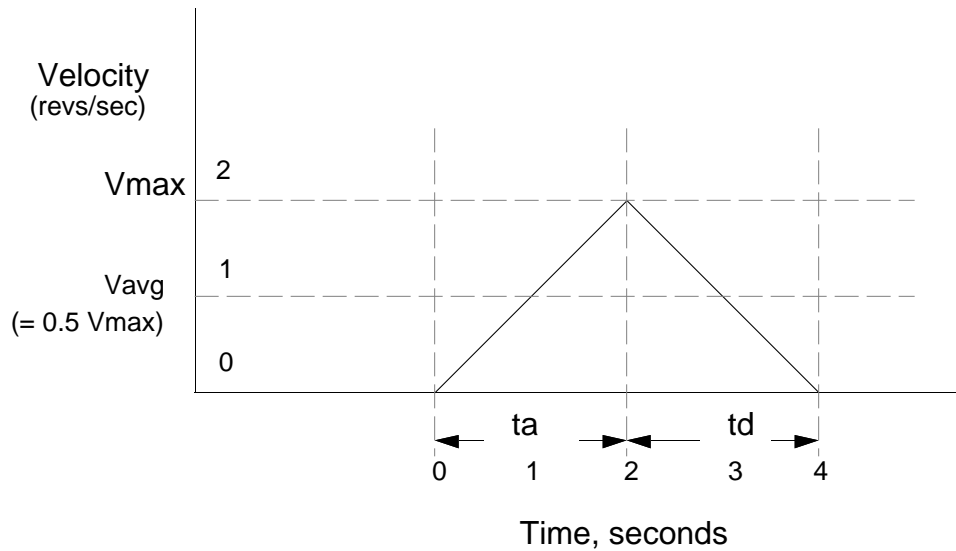
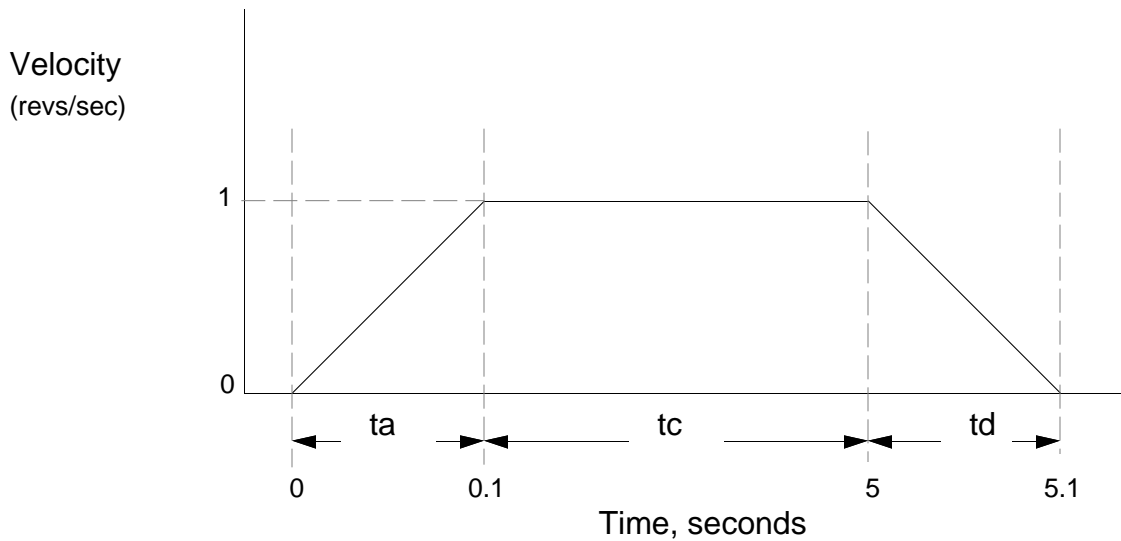


Figure 5-2. Triangular Profile

Trapezoidal Profile

A trapezoidal move profile results when the defined velocity is attained before the motor shaft has moved half of the specified distance. This is due to a defined velocity that is low, a defined acceleration that is high, a move distance that is long, or a combination of all three. For example, if the acceleration is set to 10 revs/sec/sec, velocity is set to 1 rev/sec, and distance is specified as 20000 steps (5 revs), the resulting motion profile would look like this:



ta = Accelerate
tc = Constant velocity
td = Decelerate

Figure 5-3. Trapezoidal Profile

User Profiles

The user may define a move profile using the MC command to establish the continuous mode. Velocity can be programmed on the fly by the V command.

The positioner also has a mode MQ, which is like the preset move mode in that the move distance is pre-defined, but it is possible to change speed in the middle of the move as required based on a distance, input or time delay trigger.

Encoder Following

The commands SIM and CCS may be used to allow the motor of one axis to follow the encoder of another axis or an externally-generated step and direction signal. When the control module is to be used in following mode, the input from the encoder to be followed should be connected to the External Encoder connector pins 1 - 9 (as shown in Figure 2-2). These inputs can be configured using the CCS command as an encoder input (x1, x2 or x4).

The SIM command may be used to select:

- a) Normal indexer operation (SIM0), used for reverting to normal operation by overriding previous SIM commands
- b) Encoder following with indexer motion commands inoperative (SIM1).
- c) Encoder following with indexer commands operative (SIM 2), allowing the superimposing of indexer moves.
- d) Software scaled encoder following (SIM3).
- e) Software scaled encoder following with direction reversal (SIM4).
- f) Preset following index mode (SIM 5).

For SIM1 and SIM2 operation the motor output rate follows the encoder input, using hardware scaling, at a ratio of 1 or less.

Software Scaling

This is the scaling of the encoder input when using the SIM3 or SIM4 commands to achieve following at a ratio greater or less than 1. Unlike hardware scaling, exact following ratios can be achieved by controlling both the numerator and denominator parts of the fraction used to set the scaling ratio, thus ratios such as 3:1 can be obtained.

The scaling ratio is set using the CMR command value divided by the CUR command value to give :

$$\text{Motor Output Rate} = \text{input rate} \times \frac{\text{CMR}}{\text{CUR}}$$

Both CMR values and CUR values are individually limited to a range of 1 to 32,767, and when combined as a fraction are further limited, by software scaling to having both numerator and denominator in the range 1 to 255. This results in a maximum division ratio of 1/255, or at the other extreme a maximum multiplication ratio of 255. Software scaling does not allow the superimposing of indexer moves.

When using SIM3 or SIM 4 operation for short moves it is possible to predict the number of steps the motor will take using the formula :

Number of motor steps to move =

$$\text{INT}\left(\frac{\text{CMR} \times \text{no. of pulses received} + \text{prev. remainder}}{\text{CUR}}\right)$$

where INT means "take the integer part of", with the value rounded towards zero whether positive or negative. The controller can repeatedly apply this formula to establish an iterative calculation. In situations where short moves have been programmed the ratio of CMR/CUR may only approximate the number of steps the motor is required to move, but since the remainder is carried forward no steps are lost. This allows the CMR/CUR value to better approximate the following ratio in subsequent moves.

In summary, short moves may only approximate the defined following ratio, but no positioning accuracy is lost in later moves.

Buffered Clock Mode

SIM3 or SIM4 work in a buffered clock mode, which allows the controller to buffer an unprofiled following-input pulse stream until the output velocity equals the following input velocity. The controller accelerates the output velocity to match the input velocity using an exponential acceleration profile, the time constant of which is set using the CAG command. The default time constant value of CAG is 1.00ms to accept already profiled follower inputs.

The input pulses are buffered (or stored) at the input resolution, the actual number being stored at any particular instant being termed the following error. This can lead to input pulses being lost above a maximum speed, due to excessive following error.

The maximum number of input pulses that can be buffered is

+/-32767 which requires CAG to be less than $\frac{32767}{\text{input pulse rate}}$ to prevent overflow.

Preset Following Index Mode

The following mode SIM5 selects indexing at a speed determined by the external input. In this mode the controller sets the motor velocity to a speed in rps determined as a percentage of the following input speed in rps. The percentage following factor is set by the FOL command which can be varied between 0.0 and 5000.0%.

When using this mode the acceleration is fixed to whatever is defined by the A command, and the V command value has no effect since the FOL command percentage value will now control the velocity.

The velocity is dependent on the user resolution (CUR value) and the motor resolution (CMR value). CUR and CMR set the encoder / motor resolution ratio so that the input shaft speed will match the output shaft speed with FOL set to 100%.

The FOL value can be entered from a variable, for example:

1FOL(VAR1)

Pulses must be present at the input when the G command is executed.

Position Maintenance

By attaching an encoder to the load, Position Maintenance can be used to correct position errors between motor and encoder at the end of a move. The controller will detect the difference between the number of steps the motor was commanded to move and the number of steps reported by the encoder. This position error is used, at the end of a move, to drive the motor in a direction to give the correct encoder reading.

To make use of Position Maintenance an encoder will need to be fitted at the load end of the system and connected to the controller. If the encoder were connected directly to the stepper motor shaft the stepper motor would need to have a step resolution at least 4X that of the encoder. For example, using the PDHX-E ministepping drive set to 4000 steps/rev will require the use of a 250 step/rev quadrature encoder, which after processing will generate 1000 steps/rev (with CCS0 i.e. normal x4 decode). In a practical application the motor may drive the load via a gearbox or the encoder may sense the load in a manner that alters the number of steps being generated. The aim is to achieve at least 4 motor steps for every encoder edge, thereby maintaining the minimum X4 step resolution of the motor over the encoder.

Position Maintenance is enabled using the FSC command and can be operated in Encoder Step mode or Motor Step mode using the FSB1 command. Note: If motor steps are chosen the position resolution will still be determined by the encoder.

Position Maintenance Using The PDHX-E

The PDHX-E can use Position Maintenance to correct for positional errors at the end of a move. If a 4,000-step move is made, the motor will attempt to go 4,000 steps (or 1 rev). At the end of the move the controller will read the encoder to check it reads 1,000 steps or one revolution. If not, it will correct the motor's final position by commanding the motor to move until the encoder reads 1,000 steps. In practice, mechanical alignment errors and lost motion within the system will not allow precise matching of motor and encoder, consequently an error band (Dead Band Window) is defined using the DW command. With DW10 the motor may have an error of 10 mini steps before Position Maintenance will attempt to correct the error. If the number is set too small, the motor may oscillate about its correct position.

If a Dead Band Window has been set and Position Maintenance is used to correct a position error the motor will only be moved by an amount that just re-positions the motor back within the Dead Band Window. Greater re-positioning accuracy will only be achieved by making the Dead Band Window smaller.

Speed Of Correction

Once a move has been completed and the controller decides position maintenance correction is required, it will move the motor at a fixed speed, set by the Maintenance Velocity (MV) command.

Use of Position Maintenance

Used with a ministepping drive, Position Maintenance is generally used to correct for load positioning errors at the end of a move. A typical application would be the correction of slippage in pinch rollers. A stepper motor is used to drive the pinch rollers and the encoder

senses the movement of material passing through the rollers via a drive wheel, as shown in Figure 5-4. If slippage occurs driving the required length of material through the rollers the encoder count will disagree with the motor step count, therefore Position Maintenance can be used to give the motor more steps until the correct encoder count is achieved.

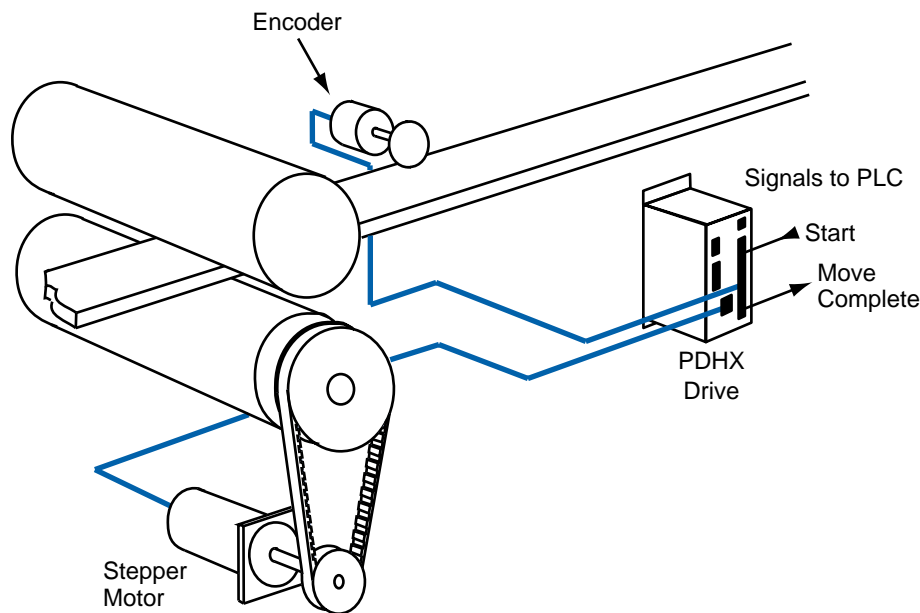


Figure 5-4. Position Maintenance Used to Correct for Slippage

Stall Detect

Once an encoder is fitted to the motor, Stall Detection can be used. A stall occurs when the error between the commanded position and the actual position, determined by the encoder, exceeds the value set in the maximum allowable position error (CPE) command. The value of CPE will be in motor steps and should not be set less than 40 steps in 4000 step/rev mode. This is to allow for rotor lead/lag during acceleration and deceleration. If the motor becomes desynchronised, the minimum resulting error will be 80 steps in 4000 step/rev mode.

Stop-on-stall

You can enable the Stop-on-stall function with the FSD1 command. The move will terminate, without any delay, as soon as the stall is detected. The function can be used in Motor Step or Encoder Step mode.

Output-on-stall

A drive output can be defined to act as an Output-on-stall by assigning it the letter L. When a stall condition occurs the output will be activated, but it will not latch the stall condition. By selecting an output as an output-on-stall you are not causing the motor to stop on a stall. The motor will not stop on a stall unless you enable it with an FSD1 command.

Interaction of Stall Detect and Position Maintenance

If both Stall Detect and Position Maintenance are enabled and the motor is moved away from its expected position, a stall will occur or position maintenance will attempt to bring the motor back to position. The decision as to what happens is made as follows:

If DPE (position error) $>$ CPE (maximum allowable position error) a stall condition exists, if stop-on-stall is enabled the motor will stop.

If DPE (position error) $>$ DW (deadband window), position maintenance will occur.

A value of DPE less than DW will leave the motor in its present position, a value of DPE between DW and CPE will be corrected using position maintenance and a value of DPE the same as or greater than CPE will cause a stall condition.

Program Storage

The program memory is battery backed up RAM with memory retention of 10,000 hours. The RAM has 8K characters available for sequence storage and a write protect facility is provided.

Programs are stored in variable length buffers and the total length cannot exceed 8K.

Write Protection

A write protection facility is incorporated for the protection of stored sequences and parameters stored by the SV command. If switch 8 (on top of the drive) is ON then the memory is not write protected. An attempted SV command when the switch is OFF will result in the following error message being displayed:

#15 BACKUP RAM IS WRITE PROTECTED

Motion Program Selection

The system may be set up so that no sequence is executed at power-up. In this case sequence execution would be initiated from the controller over the RS232C.

Alternatively, a single sequence to be automatically executed at power-up can be programmed. After that sequence is executed, control can pass to another sequence or to the RS232C interface.

Sequence selection may also be initiated via thumbwheel, RP240, and PLC inputs. Up to 64 user defined sequences can be selected for execution in this way.

Section 6. COMMUNICATING WITH THE POSITIONER

Chapter Objectives

The information contained in this chapter will enable you to set up communications with the positioner using an RS232C serial link. If more than one positioner is present in the system, details are provided on the connection of multiple positioners using the RS-232 Daisy Chain.

Command Interface

The interface is a three wire implementation (Tx, Rx, Ground) of RS232C. The Tx and Rx lines requires a minimum voltage swing of ± 3 volts.

Hardware handshaking is not supported in any form. The computer or terminal sending characters to the positioner should have its handshaking disabled by either hardware or software.

Communication Parameters

The positioner communications protocol is fixed and as follows:-

9600 baud, 8 data bits, no parity, 1 start bit, 1 stop bit

Device address: 1-32

Echo function: all characters received are immediately re-transmitted. This function can be enabled using the SSA0 command or disabled using the SSA1 command.

Once entered, a parameter will be remembered until power down. It is not necessary to keep re-stating the same parameter value.

When you power up the positioner, all parameters are assigned default values which the 1DR command will show you. You can change any of these and then make them the new default value by typing the SV (Save command).
(see also commands RFS, RIFS).

XON, XOFF software handshaking is supported.

Installing the RS232C

RS232C connections from the controller to the positioner are as shown in Figure 6-2. Note that the Tx and Rx lines are cross-connected so that transmit output is connected to receive input.

Address Selection

A unique address must be assigned to each positioner in the chain. Normally positioner 1 will be the first in the serial communications link. The positioner's address is assigned by configuring the DIL switch on top of the drive in a binary pattern as shown in Table 6-1.

The DIL switch can be accessed through the ventilation slots in the top of the drive using a small screwdriver or similar tool. Make sure that the AC power is disconnected before attempting to change the switch settings. SW1 is nearest to the front of the drive.

Switch settings for addresses 1 to 16 (switch 5 off)				
Address	Sw 1	Sw 2	SW 3	Sw 4
1 (17)	ON	OFF	OFF	OFF
2 (18)	OFF	ON	OFF	OFF
3 (19)	ON	ON	OFF	OFF
4 (20)	OFF	OFF	ON	OFF
5 (21)	ON	OFF	ON	OFF
6 (22)	OFF	ON	ON	OFF
7 (23)	ON	ON	ON	OFF
8 (24)	OFF	OFF	OFF	ON
9 (25)	ON	OFF	OFF	ON
10 (26)	OFF	ON	OFF	ON
11 (27)	ON	ON	OFF	ON
12 (28)	OFF	OFF	ON	ON
13 (29)	ON	OFF	ON	ON
14 (30)	OFF	ON	ON	ON
15 (31)	ON	ON	ON	ON
16 (32)	OFF	OFF	OFF	OFF

For addresses in brackets (17 to 32), turn on Switch 5.

Table 6-1. Positioner Address Selection

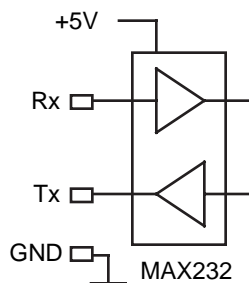


Figure 6-1. RS232 Input

Controller Daisy Chain Wiring

You may daisy chain up to 32 axes. Individual drive addresses are set with the Controller DIP switches (see Table 6-1). When daisy chained, the units may be addressed individually or simultaneously. You should establish a unique address for each axis. Refer to Figure 6-2 for Controller daisy chain wiring configuration.

Commands prefixed with a device address command only the unit specified. Commands without an address command all units on the daisy chain. The general rule is: *Any command that causes the drive to transmit information from the RS-232C port (such as a status or report command), must be prefixed with a device address.* This prevents daisy chained units from all transmitting at the same time.

Attach device identifiers to the front of the command. The Go (G) command instructs all units on the daisy chain to go, while 1G tells only unit #1 to go.

When you use a single communications port to control more than one Controller, all units in the daisy chain receive and echo the same commands. Each device executes these commands, unless this command is preceded with an address that differs from the units on the daisy chain. This becomes critical if you instruct any indexer to transmit information. To prevent all of the units on the line from responding to a command, you must precede the command with the device address of the designated unit.

No Controller executes a device-specific command unless the unit number specified with the command matches the Controller unit number. Device-specific commands include both buffered and immediate commands.

You must use status-request commands in an orderly fashion. Commands should only be issued when the host is ready to read the response. You should not send new commands until you receive a response from the previous status-request command. In particular, you should not issue an immediate-status command until the host receives a buffered command status response. If this is not followed, the command responses will be intertwined, rendering the data useless.

If you enable the Interactive mode (EX1), the Controller that is currently echoing will respond with a prompt (>). This prevents all the Controllers from sending out > in a daisy chain. Typically, you should disable the Interactive mode when you use a host computer with the Controller. The default for the EX command is enabled (EX1).

Figure 6-2 shows a multiple-drive configuration (daisy-chain) of RS-232C ports from one controlling terminal or computer.

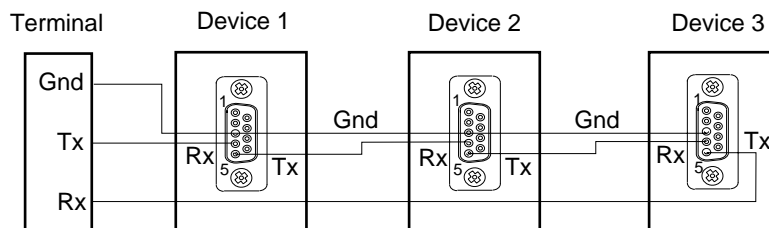


Figure 6-2. RS-232C Daisy Chain Configuration

Example

Three Controllers are connected to an RS-232C daisy chain. Send the following commands:

Command	Description
MN	Sets unit to Preset mode for all three controllers
A5	Sets acceleration to 5 rps ² for all three controllers
V10	Sets velocity to 10 rps for all three controllers
LD3	Disables limits (if they are not connected)
1D25000	Sets axis 1 distance to 25,000 steps
2D50000	Sets axis 2 distance to 50,000 steps
3D100000	Sets axis 3 distance to 100,000 steps
G	Moves all axes

Unit 1 moves 25,000 steps, unit 2 moves 50,000 steps and unit 3 moves 100,000 steps. All three units use the same acceleration and velocity rates.

Interactive operation with a PC/PLC

To create a reliable, trouble free interactive RS232 communications control program between a PC/PLC and the serial interface you should note the following points :

- • Use the X-WARE package for terminal emulation on an IBM PC or equivalent computer system.
- • The interface should only be operated in computer mode to suppress error messages and extended text output.
- • The program should operate the interface with a response check during critical processes, such as sequence download. The response check will need to detect :
 - No response at all
 - Incorrect characters
 - More characters than expected
 - Less characters than expected
 - Control characters such as EOF caused by errors
 - Correct response

The response expected should be either echoback, or character returned by interrogation commands, or the sequence of the two.

- If there is no check for echoback on each character, you should include a delay of up to 1ms between each character sent . This may require characters to be sent individually rather than as strings.
- You should not use the simple BASIC INPUT statement as this 'hangs' the link without time out if an error causes no carriage-return character to be received from the interface.
- Interpreted BASIC should not be used, as it is too slow for most applications.
- Interrupt error handling, such as the BASIC 'ON ERROR" statement, should be enabled.
- Commands such as SV, Z, RFS and RIFS take several seconds to execute. This also applies to status commands like DR and HELP. It is recommended you insert a suitable delay after these commands to allow them to fully execute their intended function without corrupting immediately following commands, although the use of these commands in an interactive situation is unlikely.
- If problems arise the communications error responses actioned by the R command will help diagnosis. In some cases a recovery can be attempted by sending the command again (such as the R command itself). But this may not always be appropriate, for example LD3 altered to L3 would not be fully corrected by simply sending LD3 again.
- A routine that breaks some of these rules may appear to work satisfactorily, but is not necessarily safe. The worst case communications response condition may involve a

number of coincidences, such as sending a status request just as the interface hits a limit, just as it starts decelerating etc. Any motion control requirement explicitly takes priority over communications under these circumstances, and can therefore cause a communications problem unless you wait for echoback. Also different issues of software may have different execution time characteristics which can make a routine based on delays work at one software issue, and not another.

Accurate defensive programming is required if safety constraints are to be met and possible damage to hardware prevented.

Interactive Programming

To help you achieve reliable, trouble free interactive programming between a Controller Positioner and a PC/PLC the following guidelines will be useful.

- The interface should be operated in terse mode (EX0) to suppress unsolicited error messages and extended text output. This will also allow for extra communication time.
- The program should include an interface response check, especially during critical processes such as sequence download. Checks should be made for:
 - *No response at all
 - *Incorrect characters
 - *More or less characters than expected
 - *Control characters such as EOF formed by errors
 - *The correct response

Where the correct response is the echoback of the transmitted characters or characters returned by interrogation commands or a combination of the two.

- The command buffer is 256 characters long. When it has filled up, the interface will echo a <CTRL>G instead of the correct character. Subsequent characters received will 'overwrite' the last character in the buffer until a 'free' space occurs.
- Given the above, it is preferable to transmit one character and wait for correct echoback before transmitting the next character. This enables an error to be rectified immediately.
- When monitoring the echoback characters in verbose mode (EX1), some are echoed as a combination of characters, for example:
<CR> is echoed as <CR> <LF>.

This is not the case in terse mode (EX0).

- If there is no check for echoback on each individual character, an explicit delay of at least 1ms between each character must be included. This may mean sending characters individually rather than in strings.
- Certain interface commands, notably ON, Z etc. take some time to execute and consequently momentarily interfere with communications. An explicit delay (several seconds is typical) should be inserted after these commands before another is sent.
- This interference of communications can involve temporarily not servicing the RS-232 interrupt on the Controller, allowing subsequent characters received by the UART, during this condition, to cause an overrun error within the UART.

- The status request command (R) can be used to provide helpful diagnostics. The responses generated by the interface are as follows:
 - *R<CR> - ready for a command with no errors
 - *S<CR> - ready for a command with function error
 - *T<CR> - ready for a command with previous comms error
 - *U<CR> - ready for a command with function error and previous comms error
 - *B<CR> - busy performing a move with no errors
 - *C<CR> - busy performing a move with function error
 - *D<CR> - busy performing a move with previous comms error
 - *E<CR> - busy performing a move with function error and previous comms error
- The average execution time of commands is approximately 1.5ms. The execution time is dependent upon the activity of the indexer. When performing a move, especially during acceleration and deceleration, there is generally less time available for command execution. This time will delay the execution of subsequent commands, and hence reduce the rate at which the command input buffer is cleared.
- The interface will correctly respond to XON/XOFF protocol, but will not be able to generate the XON/XOFF signals when the command input buffer is full.
- If the control program does not use the echo back facility provided by the interface it may be disabled by sending the command SSA1. This will speed up the character handling routine a little, but will stop any commands from being transmitted to axes further down the chain, so should be used with care.

The fact that a routine that breaks some of these rules 'works' is irrelevant. Worst case communication response conditions may involve a number of coincidences such as sending a status request just as the interface hits a limit, just as it starts to decelerate, just as the servo is responding to a load glitch etc. Any motion control requirement explicitly takes priority over communications under these circumstances and can therefore cause a communication problem unless you wait for echoback.

If the Controller echoback is turned off by accident, causing an apparent loss of communication (hanging), it may be re-enabled by typing the following sequence:

CTRL-Q<space>E<space>1R<carriage return>

Note:

[1] This sequence is case sensitive - make sure you are in upper case.

[2] CTRL-Q means hold the 'control' key down and press the Q key at the same time. This sends the X-ON ASCII character. The Controller obeys the X-ON X-OFF communications protocol which is used to hold up communications if an interface is sending too much data too quickly. This means that if X-OFF is sent down the chain (the character CTRL-S) all communications from the axes stop until CTRL-Q (X-ON) is sent down the chain. This includes echoback of commands. If CTRL-Q is never sent the system will 'hang'.

[3] You may type CTRL-S by mistake. This condition can be cleared by cycling the power.

[4] The E command is the enable RS-232 command which can easily be miss-typed as the F command which will lock out the keyboard. Cycling the power will not cure this problem since SV cannot be typed following a keyboard lock out.

[5] The <space>1 in <space>1R is important. It sets axis 1 as the echoback device (as would <carriage return>1). If you make a typing error, such as TRD<space>3000 instead of TRD3000 the <space>3 sets axis 3 as the echoback device, where R can be any command.

Certain 'tidy up' commands can be useful. For example:

- If you are not getting the typed characters echoed back you may need to use the SSA0 command.
- To get a prompt (\$ or >) you may need to use the EX1 command.
- To get out of a \$ prompt (indicating a sequence definition) type XT.
- If buffered commands are being ignored, type S in case the indexer is busy. You should avoid typing Z or cycling the power.

In summary:

- Don't type CTRL-S
- Avoid accidental use of the command F
- Don't address a non-existent axis
- Don't rubout a device address
- Type carefully

None of the above conditions is an error so the status LED on the Controller will not indicate a fault and the axis may well be energised or even moving.

Programming With the RP240

Prompting an Operator or Displaying Information

In many motion control applications, the most important requirement is the operator interface. Presenting information to an operator in a desired format is often difficult. The RP 240 has two visual indicators to help present information to the operator. The simplest indicator is the 8 LEDs on the panel. These LEDs can be turned on or off with the DLED command. The LEDs can be used in conjunction with the outputs to show the state of an output, or they can be used to show status, such as motor moving, specific sequences in progress, etc.

If a DLED10001011 command is issued, the LEDs shown below would be illuminated. These eight LEDs can be labelled, using the slide in card provided, to represent cycle status, output status, etc.

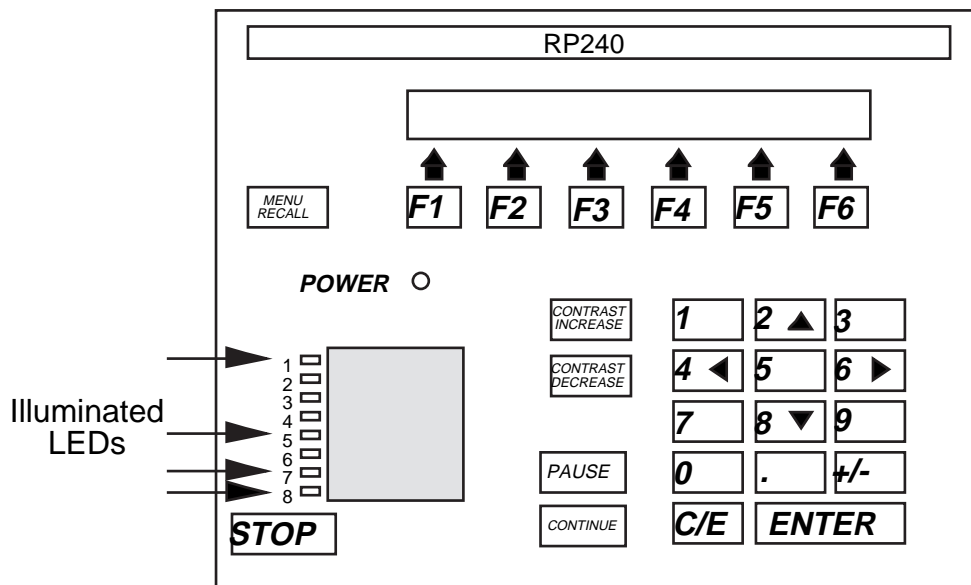


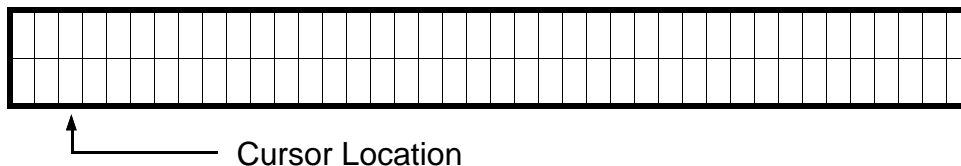
Figure 6-3. Status LEDs

The other indicator is the two line, 40 character LCD display. This display can be controlled with specific Extended X Language commands. The Position Cursor (DPC) command allows the user to program the location of the cursor on the LCD display. The 'Display Text Data on RP240 Display' (DTXT) command allows the user to place text, beginning at the current cursor location, on the LCD display. For example: A user wishes their operator to see the message ENTER THE CYCLE COUNT. The user wants this message placed on line two, starting after two spaces in from the left. Following the message by three spaces the cycle count is to be entered, and stored in variable 1. Below are the steps required to accomplish this.

The cursor does not appear on the display. The cursor is displayed when the VARn=NUM or DPC commands are used.

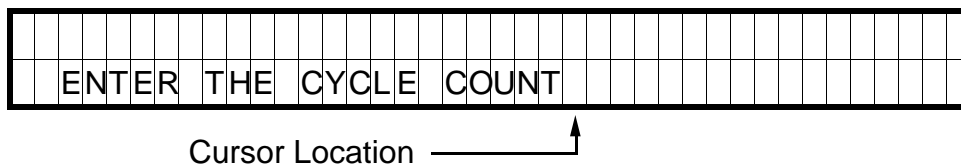
Step 1

Issue the DPC202 command (line 2, position 02)



Step 2

Issue the DTXT"ENTER THE CYCLE COUNT" command.



Issue the DPC225 command (line 2, position 25)

Diagram illustrating the input line structure for the CYCLE COUNT field. The input line is 40 characters long. The first 20 characters are used for the text "ENTER THE CYCLE COUNT". A cursor arrow points to the 21st character position, indicating the start of the numeric input field.

Once the operator has been provided with the prompting message, the actual cycle count data is provided using the Read and Enable Numeric Keypad (VARn=NUM) or Read and Enable Function Keys (VARn=FUN). The VARn=NUM command will enable the numeric keypad and allow the operator to enter information. The numbers, as entered, will be displayed at the current cursor location. Once the ENTER key is pressed, the number will be accepted by the Controller.

Step 4

Issue the VAR1=NUM command.

Diagram illustrating the input line structure for the CYCLE COUNT field. The input line is 40 characters long. The first 20 characters are used for the text 'ENTER THE CYCLE COUNT'. The 21st character is an underscore '_'. A cursor arrow points to the 21st character position, labeled 'Cursor Location'.

Step 5

Press a 2, followed by a 5.

The diagram shows a 40-character display. The first 38 characters are 'ENTER THE CYCLE COUNT 25_'. The 39th character is a blank space, and the 40th character is a blank space. A cursor is located at the end of the display, pointing to the 39th character position.

If the wrong value is entered press the C/E key and re-enter the value.

Step 6

Press the C/E key.

[illegible]

Press a 0 followed by an ENTER.

[illegible]

Once the cycle count has been entered, clear the display and enter text which allows an operator to choose a particular function (in this case a choice of parts).

Step 8

DCLR0 command is issued.

A diagram showing a horizontal bar representing a memory buffer. The bar is divided into 32 small squares, each representing a byte. Above the bar, an arrow points to the first square, labeled "Cursor Location".

Step 9

Issue the DPC105 command (line 1, position 05)

A diagram showing a horizontal array of 32 cells, arranged in two rows of 16 cells each. An arrow labeled "Cursor Location" points to the first cell in the top row.

Step 10

Issue the DTEXT"WHICH PART DO YOU WANT TO RUN?" command.

A diagram illustrating a memory buffer. A horizontal row of 32 small squares represents the buffer. The first 28 squares contain the text "WHICH PART DO YOU WANT TO RUN?". The last 4 squares are empty. An arrow labeled "Cursor Location" points to the 28th square, which is the last square containing a character.

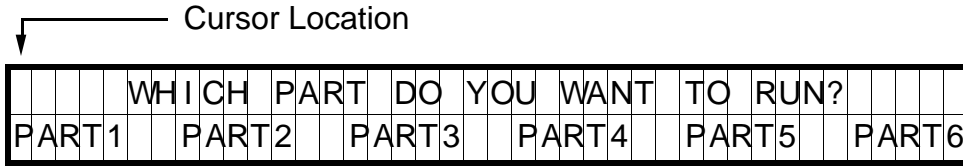
Step 11

Issue the DPC200 command (line 2, position 0)

[illegible]

Step 12

Issue the DTXT"PART1 PART2 PART3 PART4 PART5 PART6" command. This text serves as an operator menu for the function keys.



Step 13

Issue the VAR2=FUN command.

The VARn=FUN command enables the function (F1 - F6), and the MENU RECALL keys. When any of these keys are pressed, the number corresponding to the function key will be transmitted from the RP240 to the Controller. Function keys 1 - 6 will return the values 1 - 6. The MENU RECALL key will return a zero.

Step 14

Press F1

After the F1 key is pressed, the RP240 will transmit the value 1 to the Controller. This value will be stored in variable 2.

Processing Information

Variables can be used in conditional statements like IF, WHILE and REPEAT or as data values in D, V, A, L, or T commands. In step 7 above, the value 20 was entered and stored in variable 1 to be used as the cycle count. The following is a command example:

<u>Command</u>	<u>Description</u>
D2400	Set distance to 2400 steps
V1	Set velocity to 1 rps
L(VAR1)	Loop the number of times as specified by variable 1
G	Initiate motion
T1	Wait one second
N	End the loop

In step 14 above, the operator pressed the F1 key. This value was stored in variable 2, and was used to select PART1. The following is a command example:

	<u>Command</u>	<u>Description</u>
REPEAT	loop	
IF(VAR2=1)	XR11 NIF	Branch to sequence 11 if F1 is pressed
IF(VAR2=2)	XR12 NIF	Branch to sequence 12 if F2 is pressed
IF(VAR2=3)	XR13 NIF	Branch to sequence 13 if F3 is pressed
IF(VAR2=4)	XR14 NIF	Branch to sequence 14 if F4 is pressed
IF(VAR2=5)	XR15 NIF	Branch to sequence 15 if F5 is pressed
IF(VAR2=6)	XR16 NIF	Branch to sequence 16 if F6 is pressed
UNTIL(VAR2=0)		End loop on menu recall pressed

For further information on the commands used in the example above, refer to the software reference guide for the Controller.

Enabling STOP and PAUSE Keys

In addition to the function keys and numeric keypad, there are three other keys. The STOP key, and the PAUSE and CONTINUE keys are enabled by default, and can be disabled by user commands.

Typically, if an application uses the STOP key, the key will be enabled (DSTP1) in the power-up sequence. However, the STOP key can be enabled and disabled in any sequence, and at any time.

The PAUSE and CONTINUE keys can be enabled in any sequence. The DCNT command enables the PAUSE and CONTINUE keys.

If SSH is enabled (SSH1), the STOP key will function as a controlled stop. After motion is halted, the Controller will resume command processing with the command directly following the command that was stopped with the STOP key.

Sample Program

This section provides an example of an RP240 application program. Refer to the Software Reference Guide for a more detailed description of the commands.

Power-up Sequence

A configuration sequence can be used to initialise the Controller to a state compatible with the RP240 and used to initialise variables.

```
XE63
XD63
LD3 1DSTP1
VAR20=5
VAR21=1
XR21
XT
```

Sequence

In this example sequence #21 provides the main menu for a demonstration program. Control is transferred to another sequence based on a function key input.

```
XE21
XD21
1DCLR0
1DPC106
1DTXT"DIGIPLAN'S RP240 DEMO PROGRAM"
1DPC200
1DTXT"ACCESS"
1DPC235
1DTXT"EXIT"
VAR1=FUN
IF(VAR1=1)
    XG22
ELSE
    IF(VAR1=6)
        1DCLR0
        HALT
    ELSE
        1DPC220
        1DTXT"WRONG BUTTON!"
        T1
        XG21
NIF
```

NIF
XT

Sequence #22

Sequence #22 prompts you for a password, and then transfers control to sequence #23. If the password is incorrect, control is passed to sequence #21.

```
XE22
XD22
1DCLR0
1DPC100
1DTXT"ENTER CODE NUMBER xxxx"
1DPC118
VAR1=NUM
IF(VAR1<1234 OR VAR1>1234)
    1DPC220
    1DTXT"WRONG NUMBER"
    T1
    XG21
ELSE
    XG23
NIF
XT
```

Sequence #23

Sequence #23 asks you to make a selection via the function keys. Control is passed to a sequence based upon the function key pressed.

```
XE23
XD23
1DCLR0
1DPC101
1DTXT"JOG I/O TEACH MAKE FEED EXIT"
1DPC201
1DTXT"AXIS TEST MODE MOVE TO LENGTH"
VAR1=FUN
IF(VAR1=1)
    XR24
ELSE
    IF(VAR1=2)
        XR25
    ELSE
        IF(VAR1=3)
            XR26
        ELSE
            IF(VAR1=4)
                XR27
            ELSE
                IF(VAR1=5)
                    XG28
                ELSE
                    IF(VAR1=6)
                        XG21
                    NIF
                NIF
            NIF
        NIF
    NIF
NIF
XG23
XT
```

Sequence #24

Sequence #24 simulates jogging. You can select CW or CCW jog options of either high or low velocity. The default jog velocities are stored in variables 20 and 21, which were assigned in sequence #63. The MENU RECALL key is used to exit Jog mode.

```
XE24
XD24
A75
VAR1=0
1DCLR0
1DPC101
1DTXT"CW CW CCW CCW STOP"
1DPC201
1DTXT"LOW HIGH LOW HIGH MOTION"
MC
VAR1=FUN
REPEAT
    VAR3=0
    WHILE(VAR3=0)
        IF(VAR1=1) H+ V(VAR21) G NIF
        IF(VAR1=2) H+ V(VAR20) G NIF
        IF(VAR1=3) H- V(VAR21) G NIF
        IF(VAR1=4) H- V(VAR20) G NIF
        IF(VAR1=5) 1DCLR0 1DPC212 1DTXT"INVALID SELECTION" T1 XG23 NIF {all on one line}
        VAR3=1
    NWHILE
    REPEAT
        VAR1=FUN
    UNTIL(VAR1=6)
    SB
    T0.3
    VAR1=FUN
UNTIL(VAR1=0)
V0
MN
XT
```

Sequence #25

Sequence #25 provides input status information and changes the state of LEDs.

```
XE25
XD25
1DCLR0
1DPC100
1DTXT"INPUT STATUS"
1DPC200
1DTXT"UPDATES"
1DPC233
1DTXT"OUTPUTS"
REPEAT
    VAR1=FUN
    1DPC113
    IF(IN_1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_X1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XX1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XXX1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XXXX1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XXXXX1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XXXXXX1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XXXXXXX1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XXXXXXXX1) 1DTXT"1" ELSE 1DTXT"0" NIF
    IF(IN_XXXXXXXXX1) 1DTXT"1" ELSE 1DTXT"0" NIF
UNTIL(VAR1=6)
1DCLR0
```

```

1DPC100
1DTXT"THE LEDS BELOW WILL CHANGE STATE"
1DLED00000001 T1
1DLED00000011 T1
1DLED00000111 T1
1DLED00001111 T1
1DLED00011111 T1
1DLED00111111 T1
1DLED01111111 T1
1DLED11111111 T1
1DLED00000000 T1
VAR=0
XT

```

The 10 input bits in sequence #25 correspond to the total number of Controller user inputs.

Sequence #26

Sequence #26 sets up the learning mode for moves.

```

XE26
XD26
MPA
MN
VAR30=1
VAR29=1
VAR28=33
REPEAT
    1DCLR0
    1DTXT"HOW MANY MOVES TO TEACH?_x"
    1DPC200
    1DTXT"(MAXIMUM OF THREE MOVES)"
    1DPC125
    VAR30=NUM
UNTIL(VAR30>0 AND VAR30<4)
L(VAR30)
PZ
1DCLR0
1DTXT"PLEASE MAKE MOVE# "
1DVO29,2,0,0
1DPC200
T1
XR24
XR(VAR28)
VAR29=VAR29+1
VAR28=VAR28+1
N
VAR1=0
XT

```

Sequence #27

Sequence #27 runs the pre-programmed moves defined by sequence #28.

```

XE27
XD27
A100
MPI
PZ
V10
VAR29=40
L(VAR30)
XR(VAR29)
G
VAR29=VAR29+1
N

```

VAR1=0
XT

Sequence #28

Sequence #28 prompts the operator to select the feed length, maximum speed and to count the number of items to be cut.

```
XE28 XD28
1DCLR0 1DPC100
1DTXT"SELECT FEED LENGTH, MAX SPEED, AND COUNT"
T2 1DCLR1
REPEAT
  VAR30=0
  1DPC102
  1DTXT"ENTER FEED LENGTH IN INCHES: (0-99.9)"
  1DPC212
  1DTXT"FEED LENGTH=xx.x"
  1DPC224
  VAR2=NUM
  IF(VAR2>99.9 OR VAR2<0)
    1DCLR0
    1DPC215
    1DTXT"OUT OF RANGE"
    T2
    1DCLR2
    VAR30=1
  NIF
  IF(VAR30=0)
    1DCLR0
    1DPC112
    1DTXT"FEED LENGTH="
    1DPC125
    1DVO2,2,1,0
    1DPC202
    1DTXT"YES"
    1DPC236
    1DTXT"NO"
    VAR22=FUN
  NIF
  IF(VAR22>1)
    1DCLR0
    1VAR30=1
  NIF
  UNTIL(VAR30=0)
  1DCLR0
  REPEAT
    VAR30=0
    1DPC101
    1DTXT"ENTER MAX SPEED (RPM): (0-2400RPM)"
    1DPC215
    1DTXT"MAX SPEED=xxxx"
    1DPC225
    VAR3=NUM
    IF(VAR3>2400 OR VAR3<0)
      1DCLR0
      1DPC215
      1DTXT"OUT OF RANGE"
      T2
      1DCLR2
      VAR30=1
    NIF
    IF(VAR30=0)
      1DCLR0
      1DPC114
      1DTXT"MAX SPEED"
```

```

1DPC125
1DVO3,4,2,0
1DPC202
1DTXT"YES"
1DPC236
1DTXT"NO"
VAR22=FUN
NIF
IF(VAR22>1)
1DCLR0
VAR30=1
NIF
UNTIL(VAR30=0)
1DCLR0
REPEAT
VAR30=0
1DPC103
1DTXT"ENTER TOTAL NUMBER OF CUTS: (1-100)"
1DPC210
1DTXT"TOTAL # OF CUTS=xxx"
1DPC226
VAR4=NUM
IF(VAR>4100 OR VAR4<1)
1DCLR2
1DPC215
1DTXT"OUT OF RANGE"
T2
1DCLR2
VAR30=1
NIF
IF(VAR30=0)
1DCLR0
1DPC113
1DTXT"# OF CUTS="
1DPC124
1DVO4,3,0,0
1DPC202
1DTXT"YES"
1DPC236
1DTXT"NO"
VAR22=FUN
NIF
IF(VAR22>1)
1DCLR0
VAR30=1
NIF
UNTIL(VAR30=0)
1DCLR0
1DPC201
1DTXT"START"
1DPC235
1DTXT"EXIT"
REPEAT
VAR22=FUN
IF(VAR22<6 AND VAR22>1)
1DCLR0
1DPC212
1DTXT"INVALID SELECTION"
T2
1DCLR0
NIF
UNTIL(VAR22=6 OR VAR22=1)
IF(VAR22=6)
XG23
NIF
1DCLR0

```

```
VAR5=VAR2*25000
VAR6=VAR3/60
VAR7=VAR4
LD3 MN A100 V(VAR6) D(VAR5) 1DPC211 1DTXT"# LEFT TO CUT:"
L(VAR4) 1DPC225 1DVO7,3,0,0
G VAR7=VAR7-1
T.1 N
1DCLR0 1DPC213 1DTXT"JOB COMPLETED!" T1.5 1DCLR0 XG23
XT
```

Short service sequences used to set up distances in the self learn mode.

```
XE33 XD33
VAR6=POS XT
XE34 XD34
VAR7=POS XT
XE35 XD35
VAR8=POS XT
XE40 XD40
D(VAR6) XT
XE41 XD41
D(VAR7) XT
XE42 XD42
D(VAR8) XT
```


Section 7. PROGRAMMING

Commands

The complete command list for the X150/X150E controller is contained in a separate document entitled **X150/X150E Software Reference User Guide** (No. 1600.221.XX).

Communicating with the Positioner

The positioner is designed to work in one of two modes:

- a) The human interactive mode
- b) The computer host mode

The command EX1 enables the first of these and the command EX0 the second. In the human interactive mode, the positioner will respond to a dumb terminal in an intelligent, user friendly manner and provide the user with help facilities and error messages if errors occur. In the host computer mode all unsolicited output and help information is suppressed so that communication may be handled by software resident in the host computer.

If a computer is to be used for the terminal function, terminal emulation software will be required. The X-Ware program available from Digiplan for use with IBM PC's and compatibles is suitable for this purpose. X-Ware is supplied with the product.

Individual Commands

An individual positioner command controls a single parameter, function, or action such as acceleration, velocity, position, time delay, pause, loop, go, etc. There are two classes of individual commands, Immediate and Buffered.

Individual commands are variable in length. They can consist of one or more letters with a delimiter (carriage return or space character) and one or more letters and numbers with a delimiter. Each command is entered as a character/delimiter combination. Some commands include a sign (\pm) to denote direction of motion. The number of characters used depends on the type of command entered.

Typical commands have the form:

1E
S
A10
V3
D46000

When two or more individual commands are entered on the same line, they are separated by spaces, and multiple command entries will be displayed on a single line of your terminal screen.

The example below shows a set of individual command entries with space delimiters on the same line:

MN A10 V2 D25000 L10 G N

If spaces are used as the delimiter and a large number of multiple command entries are made, you could exceed the ability of your terminal to display characters as a single line (80 characters per line is a typical value). When the carriage return is used as a delimiter, the cursor returns to the beginning of the next line.

NOTE: The Positioner will normally echo carriage return with carriage return line feed in terminal mode. If you get double line spacing you need to turn off the auto line feed facility on your terminal. In computer mode (EX0) only the characters received are echoed.

The example that follows shows the effect of using carriage return as the delimiter when the terminal is not in auto line feed.

**MN
A10
V2
D25000
L10
G
N**

Immediate Commands

Commands that are identified as "Immediate" are executed immediately on receipt and will take priority over whatever operation is in progress. These commands include various Stop commands that clear the command buffer, and various Status Request commands that have no effect on the command buffer.

Buffered Commands

Commands that are identified as "Buffered" are received by the positioner and stored (in the command buffer) if the positioner is not free to execute them. Stored commands are executed as soon as the positioner gets to them, in the order that they were received.

Buffer Capacity

Any combination of individual commands and command groups can be entered in the buffer until the total number of characters currently stored (including the delimiters) equals 2000. The positioner uses a first-in-first-out serial buffer. As the commands are read from the buffer, additional commands can be entered to replace them. Therefore, the possibility exists that a command set could actually consist of more than 2000 characters. Examples of command sets that might be used for different applications are presented later.

The sequence buffer is different from the command buffer in that it can be copied to battery backed up RAM using the SV command, while the command buffer is not battery backed up, so its contents are lost on power down.

Multiple Positioner Commands

When multiple positioners are on the communications line, commands like the previous example are executed by all positioners on the line. To send commands to a single positioner, the device address of that positioner should be put in front of the instruction. The device address is set using the DIP switch on the individual positioner.

Example:

Three positioners are on an RS232C chain. They are sent the following commands:

MN A10 V10 1D25000 2D50000 3D100000 G

Unit 1 moves 25000 steps, unit 2 moves 50000 steps, and unit 3 moves 100000 steps. All three use the same move profile rates.

During sequence download to one axis in a chain, all of the other axes must have their communication switched off by use of the F command. The alternative is to make each command in the sequence device specific by adding the device address.

Programming Modes

Normal Mode

The command MN selects normal mode. In this mode moves are set by the D command. Within normal mode there are two options, absolute mode and incremental mode:-

Absolute Mode

The command MPA puts the positioner into absolute position mode. Distances entered using the D command are interpreted as absolute positions relative to absolute zero

Example:

MPA	Sets absolute mode
PZ	Sets current position to be absolute zero
D4000	Sets move to absolute position 4000
G	Execute move
D2500	Sets move to absolute position 2500
G	Executes move of 1500 steps CCW

Incremental Mode

The command MPI puts the positioner into incremental positioning mode. D command distances are interpreted as the number of steps to move from the current position.

Example:

1MPI	Sets incremental mode
1PR	Response=1500 (current position)
1D4000	Sets move to be 4000 steps CW
1G	Executes move to position 5500
1D-3000	Sets move to 3000 steps CCW
1G	Executes move to position 2500

Continuous Mode

The command MC selects continuous mode in which the motor runs continuously at the specified velocity until stopped or a new velocity is programmed. Distance data is ignored.

Example:

MC	Sets continuous mode
D4000	Ignored in continuous mode
A50	Sets acceleration to 50 revs/sec ²
V20	Sets velocity to 20 revs/sec
G	Motor runs at 20 revs/sec continuously
.	
.	
V10	Motor speed changes to 10 revs/sec
.	
.	
S	Motor stops

Speed Change Mode

The MQ command allows speed changes to be made during a preset move. The speed changes can be triggered at a set distance using the TRD command or by an input using the TRE command.

Example:

3MQ	Set MQ mode for axis 3
3A50	Set axis 3 acceleration to 50 revs/sec ²
V15	Set velocity to 15 revs/sec
D8000	Set move distance to 8000 steps
G	Start move
TRD3000	Go to next command at D = 3000 steps
V5	Change velocity to 5 revs/sec
TRE_X1	Go to next command when Input 2 goes high
V2	Change velocity to 2 revs/sec

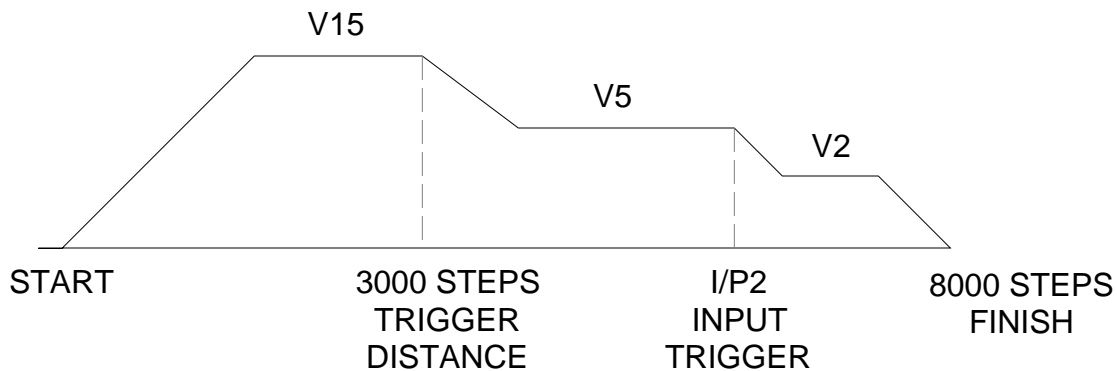


Figure 7-1. Speed Change Mode

Command Types

Start Move Commands

The positioner reads and stores individual commands in the order they are entered. Each command is read and executed before the next command is read. The GO command "G" is the principal buffered command that initiates shaft motion. When the buffer reads a GO command, the motor will then move in accordance with the move parameters that reside in the positioner at the time the Go command is read.

When the Go command is read, the positioner will incorporate any motion parameter command that was encountered prior to the Go command. For example, the "A " command would be combined with the Go command to change the acceleration value as follows:

A123 G

These commands instructs the Positioner to change the acceleration value to 123 rev/sec/sec and then execute the move using the last mode, velocity, and distance values that were previously entered.

When the RS232C positioner is initially activated, a series of commands of the form " MN An Vn Dn G " is required. These command sets up the initial operating conditions for the positioner. After the positioner has been initialised, single move parameter commands can be used.

If only "G " is entered, the positioner will repeat the previous motion pattern. For example, a string of Go commands:

G G G G

would instruct the positioner to move the motor the same way four times.

There is no need to re-enter all the shaft motion commands to change one of the variables. If one or more of the motion parameters is changed by a command entry and the Go command is then entered, the prior pattern will be repeated using the new motion parameter(s) that were entered prior to the latest GO command.

For example:

A14 G V2.6 G D-27634 G

would change acceleration to 14 and move, then change velocity to 2.6 and move with 14 as the acceleration value, then change position and move again using 14 and 2.6 as the acceleration and velocity values.

The CONTINUE command "C " is used following execution of a PAUSE command, "PS " or "U ". CONTINUE will allow execution of the next command waiting in the command buffer (if any command is stored).

Loop Commands

The LOOP command allows a cycle to be repeated continuously "L" or a given number of times "Ln" up to 65535. The END-OF-LOOP command "N" indicates where the loop ends. The END-OF-LOOP command can be used to indicate that the positioner should proceed

with further commands after the designated numbers of loops have been executed, or in combination with the "Y" command to indicate where execution is to stop. The "U" command may be used to temporarily halt Loop execution, the C command will then cause the loop to resume execution.

Stop Move Commands

The STOP MOVE commands are presented in order of severity of response. SOFTWARE RESET "K" command is the most abrupt and severe stop command you can use. The STOP AT END OF CURRENT LOOP command "Y" is the least abrupt and severe stop command

Stop at End of Current Loop Y

Also the 'stop sequence' command.

This command will not halt processing until command processing reaches the character N at the end of the command loop. At that time, the Positioner will execute the next command in the buffer after N, if any. The command loop cannot be restarted without re-entering the commands.

Controlled Stop command S

In the PRESET or CONTINUOUS mode this command will decelerate the motor to a stop at the last used acceleration rate.

An "S " command will always cause a deceleration to velocity zero at the last used acceleration.

The "S " command clears any remaining commands in the command buffer unless prevented from doing so via the SSH command.

NOTE: Normally, the motor is decelerated to a stop at the same rate as it was accelerated.

Controlled Stop command LS

The LS or 'Limit Stop' command will decelerate the motor to a stop at the deceleration rate defined by the LA or 'Limit Acceleration' command.

Kill Command K

This command stops positioner commands to the motor. In addition it terminates a loop, ends a time delay, and aborts a sequence download (XD command). The command buffer is also cleared.

Software Power-on Reset Z

The "Z " command is equivalent to cycling the AC power to the Positioner; that is, it disables the communications interface and returns all internal settings to their power-on values. The command buffer is cleared. Like the "K " command, "Z " causes an immediate cessation of torque signal to the motor. Z also de-energises and resets the drive ready for re-initialisation at power up.

NOTE: When the "Z " command is used, the Positioner is busy for up to 2 sec and will ignore any commands. The status indicator will be blank for this period.

Pause Commands

There are four types of pause commands available as follows:-

Pause and wait for continue command C to continue. A common use of the Buffered Pause PS is to hold execution of individual commands in a Command Loop until the entire loop has been loaded.

Tnnn

Pause in processing commands for a given number of seconds and then continue.

U

Hold (pause) now and wait for continue. This is an immediate command.

TRE

Pause in processing commands until the designated TRIGGER condition is met.

Triggers are used to synchronise positioner operations with external events. They can be used to implement a "handshaking" function with other devices. See command details for exact syntax.

Status Request Commands

All status request commands result in data being returned to the controller from the positioner. To prevent multiple positioners from all responding at once, the status request commands are given the classification "Device Specific" meaning that the device address of the responding positioner must be placed in front of the command. No positioner will execute a Device Specific command without its device number.

The use of status request commands must be conducted in an orderly fashion. Commands should only be issued when the host is ready to read the response. New commands should not be sent until the response is received. In particular, after a buffered status command, an immediate status command should not be sent until the response has been received by the host. If this procedure is not followed, the command responses will be intertwined, rendering the information useless.

There are two status commands that can be used to request position.

The DPA Command will report how many steps the motor has moved relative to the absolute zero position. This position is calculated by summing the total number of moves commanded since the positioner was at the absolute zero position. If position reports are needed that are relative to the beginning of each individual move, the PR command should be used.

Using the Evaluation Commands

The IF, WHILE and UNTIL commands contain fields that are evaluated to provide conditional program flow. The evaluation statement is contained in parentheses after the IF, WHILE and UNTIL commands. A space or underscore (_) indicates each comparison.

```
IF(condition)..commands..ELSE..commands..NIF
REPEAT..commands..UNTIL(condition)
WHILE(conditional)..commands..NWHILE
```

The conditions take the form of a set of evaluation statements:

1. INnnnnn_nnnnnnnnnn: compare to input status

LMT+ LMT- HOME AUX_IN STOP _
IN1 IN2 IN3 IN4 IN5 IN6 IN7 IN8 IN9 IN10

2. Variable compares:

VARn>VARn
VARn=VARn
VARn<VARn

A constant, POS (the present value of the position counter), ABS (the present value of the absolute encoder) or FEP (follower encoder position) can be substituted for the variable.

3. AND: Boolean AND operation
4. OR: Boolean OR operation

Example

<u>Command</u>	<u>Description</u>
VAR4=4000	Set variable 4 to 4000
IF(IN_10 OR VAR1=3)	If input 1 is active and input 2 inactive or the user variable 1 equals 3 then execute sequence 1, else execute sequence 2
XR1	Execute sequence 1
ELSE	ELSE command
XR2	Execute sequence 2
NIF	End of IF
REPEAT	Repeat command
VAR5=POS	Set variable 5 to current indexer position
XR5	Execute sequence 5
T5.0	Delay 5 seconds
H	Change direction
UNTIL(VAR5>VAR4 AND IN_00)	If inputs 1 and 2 are inactive and the position counter is greater than VAR4, execute the next command. Else, jump to the repeat command
VAR5=POS	Set variable 5 to current indexer position
VAR6=ABS	Set variable 6 to the absolute encoder position
WHILE(VAR6>VAR5 AND IN_00)	If the absolute position encoder is greater than VAR5 and inputs 1 and 2 are inactive, execute the following commands, else execute the command following the NWHILE command
XR7	Execute sequence 7

Drive Energise/De-energise Function

Energising and de-energising the drive is subject to the commands given and error status as follows:

The drive will energise if:

The ST0 or ON command is given when the WATCHDOG signal is inactive.

The drive will de-energise if :

The ST1 or OFF command is given.

The WATCHDOG signal become operative.

The de-energising signal must be restored, then the ST0 or ON command must be given to re-energise the drive.

Homing Function

The GH command causes the motor to rotate in the direction and at the speed specified until the home position is reached. In practice, it continues a little beyond this point but you can locate the home position accurately by then using the commands:

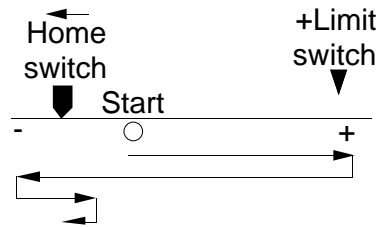
MPA D0 G

(The GH command always makes the home position zero)

The SP command can be used after homing to change the home position from zero if required.

If the home position is not found in the direction given in the GH command, the limit switch position will be reached and the switch will be operated. The motor will then reverse and seek the home position in the opposite direction. For recognised home detection, the home switch must always be activated from the direction specified in the command (shown by the arrow in Figure 7-2), so as illustrated in this figure, a limit switch reversal will take place if the direction given in the GH command is away from the home position (GH+ in the illustration).

Positive homing from + side of home switch:-



Positive homing from - side of home switch:-

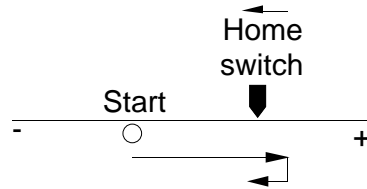
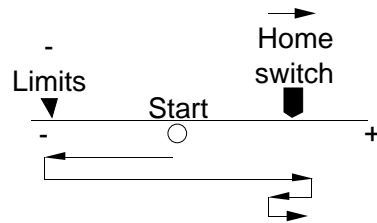


Figure 7-2. Positive Homing

Negative homing from - side of home switch:-



Negative homing from + side of home switch:-

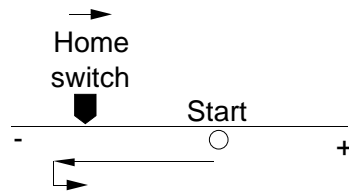


Figure 7-3. Negative Homing

Basic Programming Guide

This section uses examples of simple programs as an introduction to programming and explains the use of the command language.

The system configuration is a stepper motor. The address select switch has been set for unit 1. The motor starting position is 8000 steps CW from the home position.

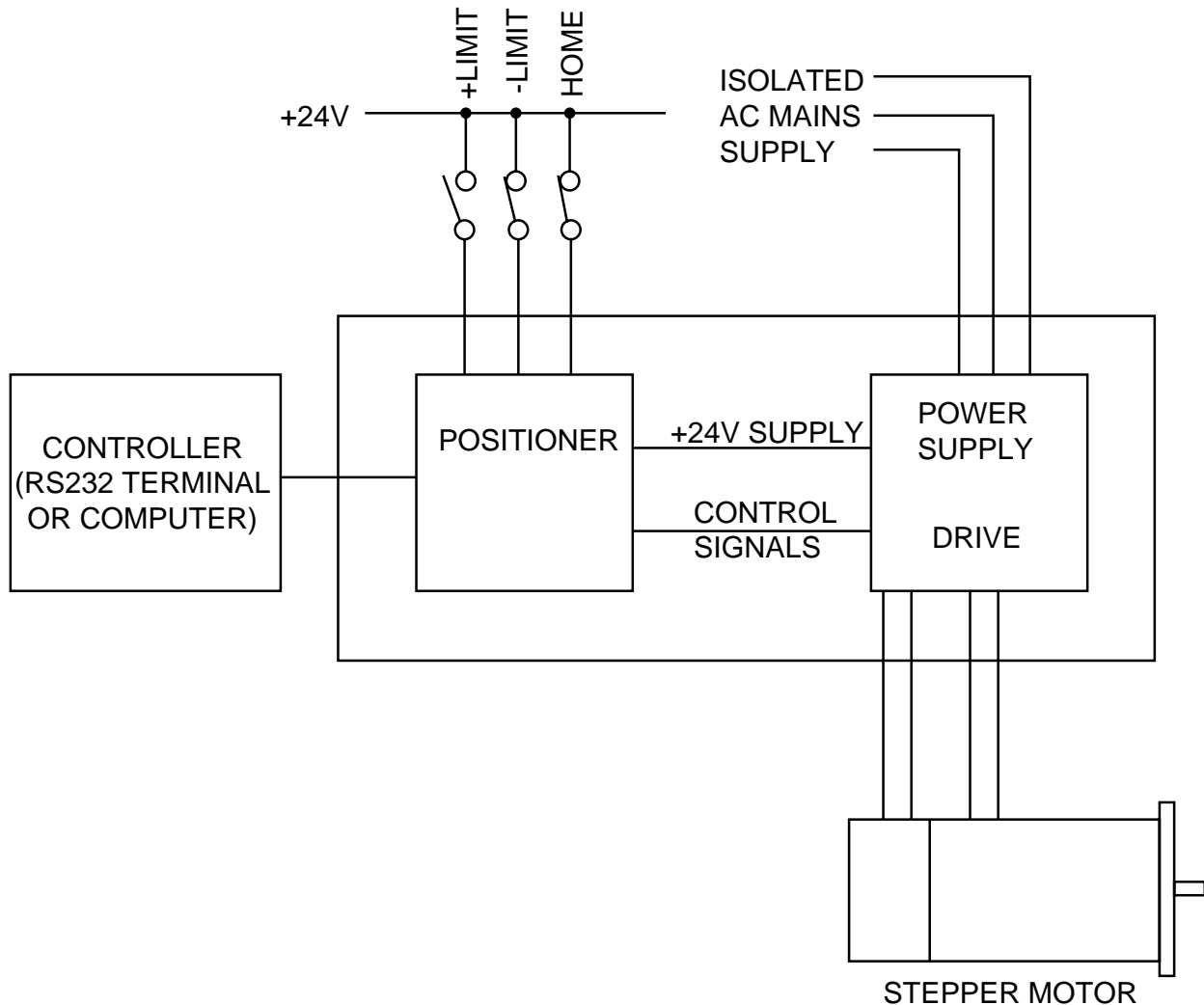


Figure 7-4. System Used in Example Programs

Example 1

In this simple example the requirement is to cause the motor to perform two revolutions in a CW direction and stop.

The following commands in the order shown will fulfil the requirement:-

<u>Command</u>	<u>Purpose</u>
MN	Set move mode to normal
1A10	Set acceleration motor 1 = 10
1V5	Set velocity motor 1 = 5
1D8000	Set move distance motor 1 = CW 2 revs
1G	Start move - motor 1 moves 2 revs CW

Description

Commands cause motor 1 to accelerate at 10 revs/sec² to a velocity of 5 rps, travelling CW for 8000 steps (2 revs) at the G command.

Example 2

This example causes axis 1 to go to the home position, then turn 5 revolutions CCW at 10 revs/sec and set the resulting position as absolute zero. The motor finally turns 8 revolutions CW to finish at absolute position 1600.

<u>Command</u>	<u>Purpose</u>
ST1	Energise the drive
A10	Set acceleration to 10 revs/sec ²
V10	Set velocity to 10 revs/sec
GH-2	Send axis 1 to go home (2 revs/sec CCW)
1D	Set axis 1 to go 5 revs CCW
G -20000	The axis moves to the required absolute position
MPA	Set absolute position mode
PZ	Set position to absolute zero
1D32000	Set move to abs. position 1600 (8 revs CW)
G	Move to position 1600 (1 rev CW from starting position)
1PR	Report absolute position

Description

Commands 2 - 6 cause the motor to go to the home position and then turn 5 revolutions CCW. Command 7 sets the absolute positioning mode and Command 8 sets the resulting position as absolute zero. Commands 9 and 10 cause the motor to turn 8 revolutions CW to absolute position 32000.

Programmable I/O Functions

Programmable Inputs

You can program each input or define them to correspond to a function. There are 15 functions that can be defined by the user. To define an input to a particular function use the IN command. The user definable inputs are labeled IN1 to IN10 on the front panel. The remaining inputs are dedicated as Limit+, Limit-, Home, Stop and Aux-In.

All input functions have been assigned a letter - refer to the IN command in the software section.

To program an input to perform a function, use the IN command followed by the input number and the corresponding function letter.

For example

1IN7H Set input 7 as direction toggle.

The active input level and the type of input (pull-up or pull-down) can be programmed using the ICON command.

To view the current status of the user definable inputs type IN. All user definable inputs will be displayed.

Programmable Input Levels

Inputs can be programmed to operate as pull-down (sink) inputs or pull-up (source) inputs. The pull-up voltage may also be programmed as 5V or 24V. This is achieved by using the circuit arrangement shown in Figure 7-5, where PROG VOLTS can be programmed to 0V (pull-down), 5V or 24V (pull-up).

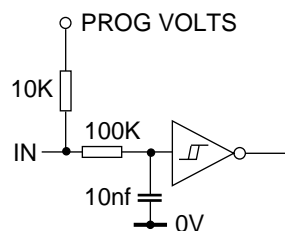


Figure 7-5 Input Circuit

Inputs are configured in 3 banks of 5 inputs, allowing each bank to be programmed independently. The three banks are:

<u>BANK 1</u>	<u>BANK 2</u>	<u>BANK3</u>
IN1	LMT+	IN6
IN2	LMT-	IN7
IN3	HOME	IN8
IN4	STOP	IN9
IN5	AUX_IN	IN10

If you need further information refer to the ICON command.

Verifying Input Wiring

To verify that you have wired the inputs correctly for your application use the IS command. This command reports the active state of all inputs regardless of the function assigned to them.

<u>Command</u>	<u>Response</u>
1IS	000000_0000000000

This indicates that no inputs are active.

The voltage level, sink or source operation for the inputs can be configured using the ICON command. However the IS command will indicate the status of any input regardless of the input configuration.

Command

1IS

Response

000000_1000000000

This indicates that input 1 is active.

Set Up Example

The limit inputs are to be connected to PNP normally closed proximity switches. The Home switch is connected to a normally open PNP proximity switch. Input 4 is to be configured as a kill input active when 0V is applied to the input. Inputs 6,7 and 8 are sequence select lines and input 9 is a sequence strobe line which is connected to the PNP output stages of a PLC. The stop input is active when 0V is applied to the input. All inputs operate at 24V dc.

This is achieved using the ICON command, as follows:

ICON 11111110

All bank 1, 2 and 3 inputs are programmed to +24V DC sense, active low. Vs is set to +24V and a motor mounted encoder (if used) is connected to the 'Primary Encoder' connector.

Configure the inputs as follows :

1IN4C	Configure input 4 as kill input
1IN6N	Configure input 6 as a data input
1IN7N	Configure input 7 as a data input
1IN8N	Configure input 8 as a data input
1IN9N	Configure input 9 as a data input
1IN10W	Configure input 10 as a sign bit
1OUT3J	Configure output 3 as a strobe output
1OUT4J	Configure output 4 as a strobe output
1OUT5J	Configure output 5 as a strobe output

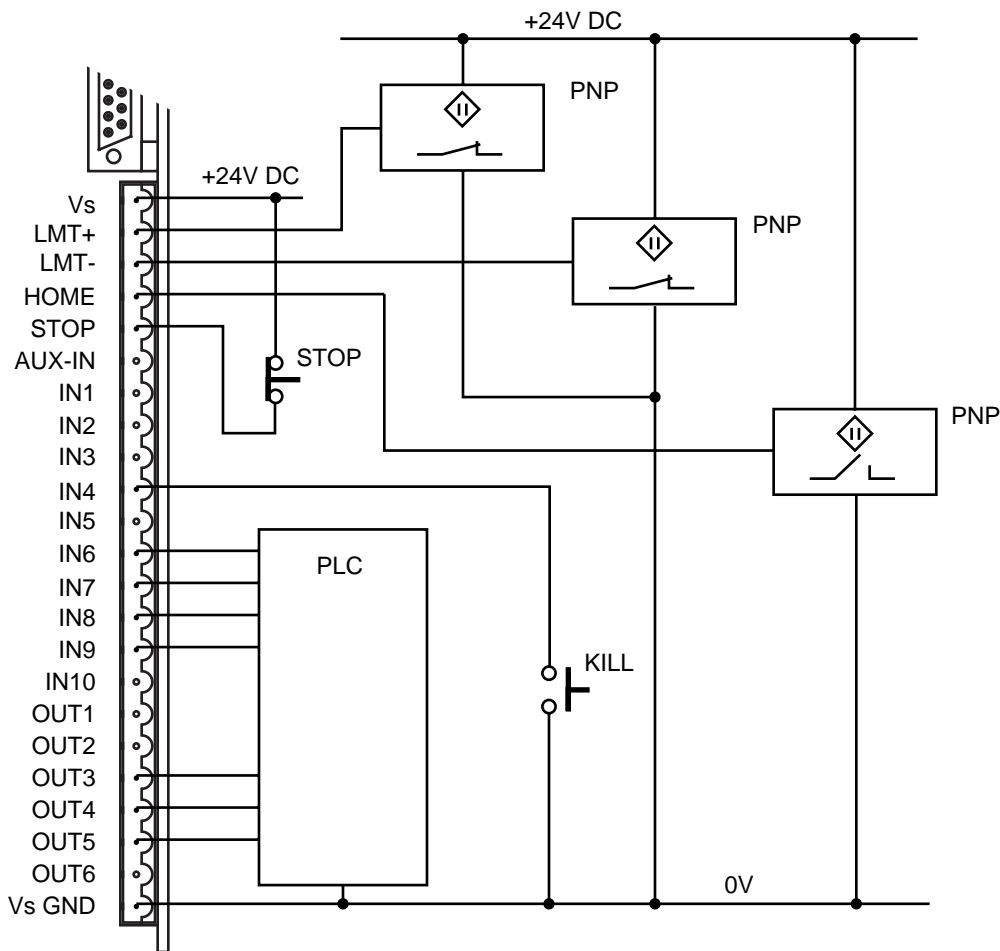


Figure 7-6 Input Example

Programmable Output

You can program each output, or configure it, to perform to a particular function. There are 6 user definable outputs which can be programmed as either NPN or PNP open collector outputs.

To allocate an output to a particular function use the OUT command. The user definable outputs are labelled OUT1 to OUT6 on the front panel. All output functions have been assigned a letter, refer to the OUT command in the software section.

To program an output to perform a function, use the OUT command followed by the output number and the corresponding function letter.

For example

1OUT3T Set output 3 as in position.

The choice of NPN or PNP output can be programmed using the OCON command.

To view the current status of the user definable output type 1OUT. All user definable outputs will be displayed.

The response of the output to an active level will depend upon the programmed state of that output.

Note: Although it has been stated that all output functions can be defined using the OUT command there is one exception to this:

OUTPUT 1 can be configured as a composite fault output using the SSD command. When SSD is set to 1 then output 1 is configured as a composite fault output and overrides the function defined by OUT. SSD must be set to 0 for output 1 to be configured using the OUT command.

Programmable Output Levels

Outputs can be programmed as NPN or PNP open collector outputs. The circuit arrangement shown in Figure 7-7 is used. Output voltage levels will be determined by the value of the external voltage (V_s) and the value of the connected load.

The selection of NPN or PNP outputs is made using the OCON command. For example:

1OCON000111

Sets OUT1-3 to be NPN and OUT4-6 to be PNP. Only 6 bits can be set using this command.

Note: the output circuit uses a connected pair of output transistors, consequently when the NPN transistor is used, but is turned OFF, the external user voltage (V_{user}) must be no greater than V_s .

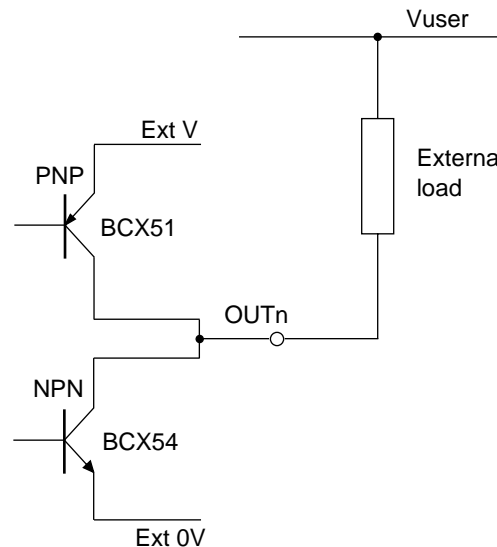


Figure 7-7 Output Example

Verifying Output Operation

You can directly control each output using the output command O. Using this command you can force any output to the desired state, in order to do this the output must be configured as a programmable output (function A).

	Command	Description
> 1O1X0	Turn output 1 on, ignore output 2 and turn off output 3.	

Using Programmable Outputs

Programmable outputs can be used to interface with other parts of the system in order to indicate such things as in position, sequence in progress, fault or a user defined function.

The following steps demonstrate how you can configure these outputs.

Example

Output 1 is to be configured as a composite fault output. Output 3 is to indicate when the motor is moving and Output 5 is to indicate when the motor is in position.

1SSD1	Set output 1 as composite fault
1OUT3B	Set output 3 as moving / not moving
1OUT5T	Set output 5 as in position output

Typing 1OUT will display

```
;Output 01 Function F
;Output 02 Function A
;Output 03 Function B
;Output 04 Function A
;Output 05 Function T
;Output 06 Function A
```

Sequences

Introduction

A sequence is a series of commands that is executed in order whenever the sequence is run. Immediate commands cannot be stored in a sequence since they cannot be stored in the command buffer. *The only "saveable in sequence" commands are buffered commands.*

The positioner has a section of non-volatile RAM totalling 8000 characters allocated for storing up to 64 sequences. The sequence buffers are of variable length, so they can store long sequences of up to 8000 characters or several shorter ones, provided that the total length of all sequences does not exceed the allocated 8000 character space.

During normal operation a command can specify the execution of any sequence stored in memory.

Sequence identifiers: number from 1 - 64.

When a sequence is programmed, the positioner automatically calculates a checksum for the sequence and stores the length of the sequence and its checksum in an internal directory.

The command XSS can be used to verify the existence of a sequence.

If a power-on sequence is used, the sequence and its sequence number are stored in the non-volatile RAM.

Programming Sequences

Sequences are programmed by first sending a sequence definition command XD, which places the commands following XD in the sequence buffer. This continues until an end of sequence command (XT) is received. Any condition which prevents proper recording of the sequence is saved and can be accessed with the status command XSD. The programmed sequence can be tested with the run sequence command XR, which executes any specified sequence.

Note: A sequence cannot be defined whilst there are buffered commands still waiting to be executed i.e., while the drive is in an active state.

The XU command can be used to read back the entered sequence.

To begin the programming of a sequence, the XD command immediately followed by sequence identifier number (1 to 64) and a delimiter must be entered. An XT command ends the sequence. The save command SV should be used to store the sequence into the positioner's non-volatile memory.

The commands entered after the XD command and before the XT command will be executed in the order in which they were entered when the sequence is run.

Example:

NOTE: The ">" prompt is replaced by the "\$" prompt when in sequence edit mode.

XD1	Begin definition of sequence No.1
A10	Set acceleration to 10 revs/sec/sec
V10	Set velocity to 10 revs/sec
D4000	Set move distance to 4000 steps
G	Go
H	Reverse next move
G	Go
XT	End of sequence

* The definition of sequence No.1 is terminated by XT but the sequence is NOT stored in non-volatile memory until the save command SV is given.

NOTE: Turn off switch 8 (on top of the drive) to write-protect sequences.

Whenever the above sequence is run, the motor will turn 1 revolution CW and then 1 revolution CCW.

Running Sequences

By Command Input

A sequence can be run by entering the XR command immediately followed by a sequence identifier number (1 - 64) and a delimiter.

Example:

XR3[space] runs sequence No.3

Standalone Operation

Stored sequences may be automatically executed when you power up the system or executed by remote switches, RP240, TM8 and thumbwheel.

A sequence selected as a binary pattern, set on inputs using the INnO command can be run in response to a sequence strobe set on an input using the INnB command.

Example

Inputs 1 to 4 are to be used as sequence select lines.

Input 5 is to be used as a sequence strobe line.

Programs have been stored in sequences 1 to 16.

1IN1O	Set input 1 as sequence data input
1IN2O	Set input 2 as sequence data input
1IN3O	Set input 3 as sequence data input
1IN4O	Set input 4 as sequence data input
1IN5B	Set input 5 as sequence strobe input.

A binary pattern is configured on the inputs corresponding to the decimal sequence number. On a logic 0 to 1 transition of the sequence strobe input, the sequence corresponding to the binary pattern will be executed.

If a sequence is already running when the sequence strobe is activated the sequence strobe will be ignored. A sequence in progress must be terminated before execution of the next sequence strobe.

Power On Sequence

One of the stored sequences can be designated to be executed when the device is powered up. The sequence to be executed can be selected by prior RS232C command (XP1-64). The single sequence specified is executed once; control then passes to the normal command processor loop. Alternatively, power-on execution can be disabled with either the XP0 or XZ commands.

It is possible to have the positioner execute a sequence on power up by defining the commands within a sequence using the XD, XT and XP commands. To define a sequence you will need to enter:

XP1 XE1 XD1 MN A10 V5 D40000 G XT 1SV

In this example the operator would not power the unit on until ready for the sequence to begin. The XP1 command enables a single run of sequence No.1 after power on. The XD1 command signifies the start of the definition of sequence No.1 and the XT command signifies the end of the sequence No.1 definition. When the operator applies power to the positioner, sequence number 1 will be executed (the motor will turn 10 revolutions in the positive direction). The sequence can be stopped using the S command or, if necessary, by using the emergency stop (K command).

Changing a Programmed Sequence

Once a sequence is defined it cannot be redefined until it has been deleted. A new sequence cannot be downloaded over an existing one.

To change a sequence, it must be deleted from memory and then redefined as a new sequence using the same number.

First, the sequence must be deleted from memory using the XE command. It will not be deleted from back-up RAM unless the SV command is used after the XE.

Example:

XE1 erases sequence No.1

The changed sequence should then be entered. It will not be saved to non-volatile memory until the SV command is issued.

Examining the Contents of a Sequence

If the operator wishes to check the contents of the sequence, he would enter:

1XU1

This would cause the positioner to send the contents of sequence number 1 to the terminal's screen. The 1 preceding the XU command is the device address which must be present since the XU command is a "device specific" command. After issuing this command the following will be displayed on the terminal's screen:

;Sequence 1 is: MN A10 V5 D40000 G

This is the command string that is performed when sequence number 1 is executed.

If the operator does not wish to have the sequence execute on power up, it can be cancelled by issuing either the XP0 or XZ command.

Sequence no.1 can also be run manually using the XR command.

Sequence Status Requests

Sequence status commands include XC, XSD, XSP, XSR, XSS. The XC command can be used to verify that the memory in the back-up RAM has not been corrupted in any way since last power down. The XC command will cause the positioner to send a checksum in the form of three decimal digits (000 - 255) to the terminal. The checksum may be recorded when sequences have been defined, so that it may be used for comparison later, such as each time the positioner is powered up.

Sequence Debugging Tools

After creating your sequences, you may need to debug them to ensure that they are performing properly. The Controller provides several debugging tools.

- In trace mode, you can trace a sequence as it is executing
- You can single step through a program
- Error messages can be enabled explaining why the Controller has stopped execution

Trace Mode

You can use the trace mode to debug a sequence or a program of sequences. The trace mode allows you to track, command by command, the entire sequence. It displays to your terminal, over the RS-232C serial link, all of the commands as they are executed. The following example demonstrates the trace mode.

Step 1

Create the following sequence:

<u>Command</u>	<u>Description</u>
XE1	Erases sequence #1
XD1	Begins definition of sequence #1
A10	Acceleration is 10 rps ²
V5	Velocity is 5 rps
L5	Loop 5 times
GOSUB3	Jump to sequence #3
N	Ends the loop
XT	Ends the definition of sequence #1

Step 2

Define the sequence #3.

<u>Command</u>	<u>Description</u>
XE3	Erases sequence #3
XD3	Begins definition of sequence #3
D50000	Sets the distance to 50000 steps
G	Initiates motion
XT	Ends the definition of sequence #3

Step 3

Enter the following command to enable the Trace mode.

<u>Command</u>	<u>Description</u>
1XTR1	Enable the trace mode

Step 4

You will now execute the sequence. The commands will be displayed on the terminal as each command in the sequence is run. Enter the following command:

<u>Command</u>	<u>Description</u>
XR1	Run sequence #1 - response is:
*SEQUENCE 001 COMMAND A10	
*SEQUENCE 001 COMMAND V5	
*SEQUENCE 001 COMMAND L5	
*SEQUENCE 001 COMMAND GOSUB3 LOOP COUNT 1	

```

*SEQUENCE 003 COMMAND D50000 LOOP COUNT 1
*SEQUENCE 003 COMMAND G LOOP COUNT 1
*SEQUENCE 003 COMMAND XT LOOP COUNT 1
*SEQUENCE 001 COMMAND N LOOP COUNT 1
*SEQUENCE 001 COMMAND GOSUB3 LOOP COUNT 2
*SEQUENCE 003 COMMAND D50000 LOOP COUNT 2
*SEQUENCE 003 COMMAND G LOOP COUNT 2
*SEQUENCE 003 COMMAND XT LOOP COUNT 2
*SEQUENCE 001 COMMAND N LOOP COUNT 2
*SEQUENCE 001 COMMAND GOSUB3 LOOP COUNT 3
*SEQUENCE 003 COMMAND D50000 LOOP COUNT 3
.
.
.
.
.
*SEQUENCE 003 COMMAND G LOOP COUNT 5
*SEQUENCE 003 COMMAND XT LOOP COUNT 5
*SEQUENCE 001 COMMAND N LOOP COUNT 5
*SEQUENCE 001 COMMAND XT

```

The format for the trace mode display is:

Sequence Number	Command	Loop Count
-----------------	---------	------------

Step 5

To exit the trace mode, enter the following command:

<u>Command</u>	<u>Description</u>
XTR0	Exits Trace mode

Single-Step Mode

You can debug your program with another level of debugging using the Single-Step mode. Single-Step mode allows you to execute one command at a time when you want the command to be executed. Use the XST command to enable Single-Step mode. Once you are in the mode, you can execute a sequence one command at a time. To execute a command, you must use the # sign. By entering a # followed by a delimiter, you will execute the next command in the sequence. If you follow the # sign with a number (n) and a delimiter, you will execute the next n commands. The following steps illustrate the use of Single Step mode:

Step 1

Enter Single-Step mode

XST1

Step 2

Begin execution of sequence #1

XR1

Step 3

You will not execute any commands until you use the # command. Type the following:

<u>Command</u>	<u>Description</u>
#	Executes one command

The response will be:

*SEQUENCE 001 COMMAND A10

Step 4

To execute more than one command at a time, follow the # sign with the number of commands you want executed.

Command	Description
#3	Executes 3 commands, then pauses sequence execution

To complete the sequence, use the # sign until all the commands are completed. To exit Single-Step mode, type:

XST0

Section 8. MAINTENANCE & TROUBLESHOOTING

Maintenance

Routine maintenance is not necessary, but occasional checking of the following points is recommended.

Motor Maintenance

Periodically check the motor to ensure that no bolts or couplings have become loose during operation, and check the motor cable or leads periodically for signs of wear. Do not make very tight bends or pull on the cable during normal operation. Check all cable connectors.

Drive Maintenance

Check that the drive is clear of loose material and has a free flow of air through the ventilation slots. Enclosures must be connected to earth ground to provide a low-impedance path for ground-fault or noise-induced currents. Check the security of the ground connections.

Fuses

PDHxE drives are fitted with fuses which limit circuit damage in the event of a fault occurring, they are not user replaceable. If the drive fails to operate correctly or you suspect a fuse has blown return the drive for repair. Warranty is void if the case is opened.

Troubleshooting

Fault LED

The red LED indicates one of the following fault conditions:

1. Motor wiring short-circuit either across phases or between phases.
2. Motor wiring short-circuit phase to GND (earth).
3. Motor supply overvoltage or undervoltage.
4. Internal supply failure.
5. Drive internal overtemperature.

The fault LED will also light up if the motor is decelerating a high - inertia load too quickly. The drive will shut down under this condition. If this occurs, you can either reduce the deceleration rate/load inertia, or use a PDS15E-D (power dump option). See INSTALLATION section for more information on the regenerative power dump option.

Note that the fault LED comes on momentarily when power is removed from the drive. This is caused by the supply rails dropping below a safe operating level, and is an indication that the fault circuitry is functioning correctly. You should always ensure that the fault LED is extinguished before re-applying the power.

Motor Fails to Move

Test the motor to see if it has holding torque. If there is no holding torque, here are some probable causes:

- There is no power.
- Current DIP switch selection is not set properly.
- There are bad connections or bad cables in the motor circuit. Disconnect the power to the drive and remove the motor connector. Using a meter, check the continuity in the motor circuit between pins A+ and A- of the motor connector. Repeat for pins B+ and B-.
- Check the resistance of the motor and cables to make sure that shorts do not exist between phases or to earth GND. The resistance across each motor phase should be consistent and there should be no connection between motor phases and between each phase and earth ground.
- Check the motor cables for signs of damage.
- The shutdown input may be active.
- If the power LED is out and the motor will not energise, the drive must be returned for repair.

If the unit has holding torque and the motor shaft still fails to move, here are some possible causes:

- The load is jammed. You should *hear* the drive attempting to move the motor. Remove power from the driver and verify that you can move the load manually away from the point of the jam. Try running the motor using the self-test switch.

Motor Stalls

A motor stall during acceleration may be caused by one or more of the following factors:

- The torque requirements may be excessive.
- The acceleration ramp may be too steep - lower acceleration may be required. Check the torque/speed curves in the published data and make sure you are trying to run the motor within the system capabilities.
- The load inertia and rotor inertia may be grossly mismatched.

If the motor stalls during the constant velocity portion of a move, the shaft and/or coupler may be damaged or binding due to improper coupling or excessive motor load.

A stall may occur if the switch setting for the motor current selection is incorrect. The motor may not be receiving enough current to drive the load.

Motor is Jerky or Weak

Check that there are no mechanical problems at the load causing variable loading conditions. Disconnect the motor from the load and run it without a load connected. Check the switch current settings.

Motor Overheats

If the motor exceeds its maximum motor case temperature rating, failure will eventually result. Check your switch settings to ensure that the current setting is correct for the motor you are using.

Motor Runs the Wrong Way

Turn off the power and interchange the connections between A+ and A- on the motor connector.

Controller problems

The 7-segment display on the drive front panel gives an indication of a number of possible fault conditions. The fault codes are listed below. Under normal operating conditions the display will show 0.

Code 0 - No faults, drive is energised.

Code 1 - No faults, drive is de-energised.

Code 2 - Drive fault. When code 2 is displayed, the Drive Fault LED should also be on. Please refer to 'Fault LED' earlier in this section.

Code 3 - Changed memory map (factory use only)

Code 4 - not applicable to the PDHX15-E.

Code 5 - Checksum error. This indicates a failure in the battery-backed RAM. It may be caused by the unit remaining unused for an exceptionally long period (e.g. several months).

Codes 6 & 7 - not applicable to the PDHX15-E.

Code 8 - Watchdog failure - the processor has 'hung', which will shut down the drive. This may be caused for instance by excessive interference from neighbouring equipment. Cycle power to restart.

Code E - Exception error. This is caused by a hardware or firmware malfunction resulting in incorrect operation of the user program.. Cycle power to recover.

Code P - Power failure. The motor supply in the drive has fallen below the minimum level.

Code U - UART error. This is normally caused by ASCII characters being transmitted in an incorrect format (baud rate, number of start or stop bits). Reset by typing the address followed by R.

No communication

Perhaps the commonest controller problem, especially during initial commissioning, is apparent lack of communication with the terminal or host controller.

If the Controller echoback is turned off by accident, causing an apparent loss of communication (hanging), it may be re-enabled by typing the following sequence:

CTRL-Q<space>E<space>1R<carriage return>

Note:

[1] This sequence is case sensitive - make sure you are in upper case.

[2] CTRL-Q means hold the 'control' key down and press the Q key at the same time. This sends the X-ON ASCII character. The Controller obeys the X-ON X-OFF communications protocol which is used to hold up communications if an interface is sending too much data too quickly. This means that if X-OFF is sent down the chain (the character CTRL-S) all communications from the axes stop until CTRL-Q (X-ON) is sent down the chain. This includes echoback of commands. If CTRL-Q is never sent the system will 'hang'.

[3] You may type CTRL-S by mistake. This condition can be cleared by cycling the power.

[4] The E command is the enable RS-232 command which can easily be mis-typed as the F command which will lock out the keyboard. Cycling the power will not cure this problem since SV cannot be typed following a keyboard lock out.

Certain 'tidy up' commands can be useful. For example:

- If you are not getting the typed characters echoed back you may need to use the SSA0 command.
- To get a prompt (\$ or >) you may need to use the EX1 command.
- To get out of a \$ prompt (indicating a sequence definition) type XT.
- If buffered commands are being ignored, type S in case the indexer is busy. You should avoid typing Z or cycling the power.

In summary:

- Don't type CTRL-S
- Don't type the command F
- Don't address a non-existent axis
- Don't rubout a device address
- Type carefully

None of the above conditions is an error so the status LED on the Controller will not indicate a fault and the axis may well be energised or even moving.

Returning the System

In the event that the drive is considered to be faulty, it should be returned for repair. Do not attempt to open the case - there are no user-serviceable parts inside. Opening the case not only voids the warranty but may also invalidate the EMC compliance.

Contact the Parker Automation Technology Centre or the machinery manufacturer who supplied the product. Equipment for repair should NOT be returned directly to Digiplan without prior authorisation. Repairs will be carried out by Digiplan but will be processed via your supplier.

Digiplan may at their discretion authorise direct shipment to and from Poole or Rohnert Park, but only by prior arrangement with your supplier. Existing UK and USA customers who purchase equipment directly from Digiplan should contact Poole or Rohnert Park for further information (contact numbers are at the front of this User Guide).

Index

"\$" prompt 78

AC power specification 31
Address switch location 24
Amplifiers specification 31

Bipolar rating 12
Buffer Capacity 62
Buffered Commands 62
Bulkhead connections 9

Cabinet connections 9
Case dimensions 25
Checksum 77
Command buffer 62
Communications protocol 43
Computer host mode 61
Connector pin-outs 20
Connectors (front panel) 18
Controller operating range 32
Controller resolution ranges 32
Cooling considerations 25
Current output specification 31
Current setting switch 26

Dead Band Window 39
Default values 43
Drive end connections 10
Drive Maintenance 85
Dump option 12
Dump option calculations 12
DW command 39

Echoback 46
Electrostatic precautions 9
Enclosure Considerations 25
Encoder following 37
Encoder selection for Position Maintenance 39
Environment specification 31
Environmental 25
External encoder D-connector 18

Fault LED 85
Ferrite absorber kit 3
Ferrite absorber specifications 10
Ferrite absorbers installation 11
FSC command 39
Fuses 32

Handshaking 43
Human interactive mode 61

Identification of drive type 3
Immediate Commands 62
Input circuit 73
Input circuit conventions 21
Installation Considerations 25

Loop Commands 65

Main I/O connector 18
Maintenance 85
Motor current (optimum) 11
Motor end connections 9
Motor Fails to Move 86
Motor is Jerky or Weak 87
Motor Leads 12
Motor Maintenance 85
Motor Overheats 87
Motor selection 11
Motor specification 31
Motor Stalls 86
Motor wiring EMC requirements 9
MV command 39

Non-volatile RAM 77
NPN Outputs 23

OEM Motor drive 17
Output on stall 40
Overheating 87

Parallel connection of motors 15
Pause Commands 67
PNP Outputs 23
Position Maintenance 39
Position Maintenance application 39
Position Maintenance correction velocity 39
Power connections 9
Pre-installation test 4
Preset following index mode 37, 38
Primary encoder D-connector 18
Product description 1
Program storage 32
Programmable output 75

QM Motor drive settings 16

Registration 34

Returning the System 89

Reverse motor direction 87

Roller slippage correction 40

RS232 connections (getting started) 4

RS232/RP240 mini-DIN socket 18

RS232/RS485 D-connector 18

S Motor drive settings 16

Selftest selection switch 26

Sequence 77

Sequence buffer 62

Sequence buffers 77

Sequence identifiers 77

Sequence selection 79

Sequence selection example 79

Series connection of motors 14

Setting up 25

Ship kit 3

Signal type A inputs 22

Signal type B outputs 22

Signal type C inputs 23

Signal type D inputs 23

Signal type E inputs 24

Signal type F inputs 24

Signal type G inputs 24

Signal type I inputs 24

Signal type J inputs 24

Signal types 21

Software Scaling 37

Specification 32

Specification 31

Stall Detect 40

Stalling 86

Standby selection switch 26

Status Request Commands 67

Step resolution switch 26

Stop Move Commands 66

Stop on stall 40

Switch Settings 25

Temperature, operating 25

Temperature, storage 25

Test circuit 5

Test configuration (pre-installation) 4

Test connections 6

Testing the PDHX-E 6

Thumbwheel interface 27

TM8 (Dual) example 29

TM8 Example 28

TM8 Thumbwheel 27

Troubleshooting 85

Unipolar rating 12

Write-protect sequences 78

XON, XOFF 43

Amendments identified for this User Guide

Certain errors have been identified within revision 01 of this User Guide. They will be corrected in revision 02, but the corrected text is identified below for your convenience:

Page 4 - Table 2-2

Drive RS232/RS485 9-way D-type Connector	3 0V	5	Computer 9-way D-type Computer 25-way D-type
	4 RxA	3	
	5 TxA	2	
			3
			2
			7

Table 2-2. Minimum RS232 Connections

(connections 2 and 3 have been transposed in column 3).

Page 87 - Error code 4

Code 4 - Excessive positioning error. If the display is flashing it means the drive is still energised. When de-energised, the flashing will stop. Check the mechanics and system parameters to correct this error.

Note: When SSK=0 the STOP input will stop the drive but keep it energised (flashing 4). When SSK=1 the STOP input will stop and de-energise the drive (code 4 permanently displayed).

(Code 4 - previously marked as not applicable).

Page 87 - Error code 8

Code 8 - STOP input is active. If the display is flashing it means the drive is still energised. When de-energised, the flashing will stop.

Note: When SSK=0 the STOP input will stop the drive but keep it energised (flashing 8). When SSK=1 the STOP input will stop and de-energise the drive (code 8 permanently displayed).

(Code 8 - previously identified as a watchdog failure).

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at [artisanng.com](https://www.artisanng.com) for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisanng.com | [artisanng.com](https://www.artisanng.com)

