PDX Series Drive User Guide

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: 1600.172.03   November, 96

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. Note that it may be necessary for the completed installation to comply with the Low Voltage Directive or Machinery Directive. 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 any part of the equipment 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.

If the equipment is used in any manner that does not conform to the instructions given in this User Guide, then the protection provided by the equipment may be impaired.

EMC INFORMATION

EMC Information is presented in boxed paragraphs (such as this one). Information in this User Guide consists of recommendations only; compliance is not guaranteed. PDX drives are sold as complex components for use by professional system builders. They are not intended for sale to end users.

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.

Table of Contents

User Guide Change Summary

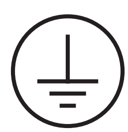
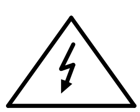
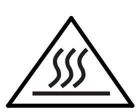
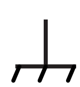
The following is a summary of the primary changes to this user guide since the last version was released. This user guide, version 1600.172.03, supersedes version 1600.172.02.

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.

Major changes introduced at revision 03 are:

EMC Installation instructions have been added to the Installation section.
LVD compliance.

Warning symbols used on the PDX 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: **PDX13, PDX15 and PDX15-D Stepper Drives**

The above products are in compliance with the requirements of directives

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

The PDX Series of drives are sold as complex components to professional assemblers, as components they are not compliant with Electromagnetic Compatibility Directive 89/336/EEC. However, information is offered in this User Guide on how to install these drives in a manner most likely to minimise the effects of drive emissions and to maximise the immunity of drives from externally generated interference.

INTRODUCTION

The PDX series of single-axis packaged ministep drives consists of the PDX13 & PDX15, with peak current ratings of 3A and 5A respectively. A further 5A version, known as the PDX15-D is also available with a built-in regenerative dump circuit. PDX series drives are high-performance, MOSFET, chopper-regulated stepper drives designed for optimum performance in low and medium power applications. A recirculating chopper regulator improves operating efficiency, minimizes power consumption, and reduces motor and drive heating. They are powered from direct on line mains supplies of 110V to 240V (nominal) AC. An internal switch mode power supply is used, incorporating power factor correction to minimise losses in the AC supply.

The PDX series of drives have 4 selectable resolutions between 400 steps/rev and 4000 steps/rev, set using the front panel switches.

PDX drives incorporate an RS232C-programmed indexer, which features a complete range of motion control commands using Digiplan's X-Code command language. Its non-volatile memory may be programmed with up to 7 complete motion control sequences which can be selected using external input signals.

The indexer has dedicated inputs for home-position and end-of-travel limit switches, with provision for 3 programmable user inputs and 2 programmable outputs. Hardware addressing allows up to 8 drives to be connected to a single serial port and up to 255 drives when software addressing is used.

The drives are suitable for use with 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 illuminated.

The drive, indexer 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 for service personnel use only.

SPECIFICATION

PDX Series Drive Specifications

Parameter	Value
Amplifiers	
Type	MOSFET Chopper
Motor resolution	400, 1000, 2000 and 4000 steps/rev (user-selectable)
Protection	
Short circuit	Phase-to-phase, across phases and phase to ground
Peak output current	3A/phase (PDX13), 5A/phase (PDX15) - switch reduceable
Standby current reduction	To 80% or 50% of programmed peak value after 100ms (switch-selected)
Maximum stepping rate	200kHz at 4000 steps/rev
Nominal chopping frequency	20kHz
Command Interface	
Indexer	
Position range	±1 - 99,999,999 steps
Velocity range	0.001 - 50 rev/sec
Acceleration range	0.01 - 999 rev/sec/sec
Communication parameters	RS232C, 3-wire, 9600 Baud, 8 data bits, 1 stop bit, no parity
Motion programs	7, max. 256 characters each
Programmable inputs	3, TTL levels
Programmable outputs	2, TTL levels, 24mA max.
Fault output	N channel MOSFET output (see figure 7) Low (transistor switched to 0V) +0.8V max. at 1.2mA max. High (transistor off) +5V
Power up reset time	1 - 2 secs

Table 1. PDX Series Drive Specifications

PDX Series Drive Specifications (Continued)

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
Input current	
PDX13	2A rms max
PDX15	3A rms max
Recommended supply protection	3A MCB Type ‘C’ characteristics or fuse in live (3A TL HB)
Fuse Values	
PDX13 mains input (FS2 & FS3)	2A TL HB (6.3 x 32 mm)
PDX 15 mains input (FS2 & FS3)	3.15A TL HB (6.3 x 32 mm)
Motor HV fuse FS1	5A QA HB (5 x 20mm)
Dump PCB fuse FS1 (if fitted)	500mA TL LB (5 x 20mm)
Output current range	
PDX13	0.9A - 3A (300mA steps)
PDX15	2.5A - 5A (350mA steps)
Standby reduction	50% or 80%
Environment	
Drive dimensions	Height 250mm(9.8in), Width 50mm (2in), Depth 190mm (7.5in)
Weight	1.8Kg (4lb)
Operating temperature	0° - 50°C (32° - 122°F)
Ingress protection	IP20
Max. power dissipation of drive unit	
PDX13	20 Watts
PDX15	30 Watts
PDX15-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
Inductance range	1mH-10mH

Table 1. PDX Series Drive Specifications (Continued)

Control I/O Specifications

Pin	Name	I/O	Min. High Level	Max. Low Level	Comments
1 2	Step Out* Dim Out	O	4.26V @ 24mA Source	0.44V @ 24mA Sink	Totem pole device
3 4	CW Limit CCW Limit	I	2 Volts	0.8V @ 1.2mA	Has 4K75 pullup to 5V
5	Home	I	2 Volts	0.8V @ 1.2mA	Has 4K75 pullup to 5V
6	-	-	-	-	Reserved
7	Gnd	I/O			Signal common return
8 10	Output 2 Output 1	O	4.26V @ 24mA Sink	0.44V @ 24mA Sink	Totem pole device
9	Fault**	O	5 Volts (Max. 12 Volts)	0.8V @ 50mA Sink	MOSFET
11 12 13	Sequence select 1 Sequence select 2 Sequence select 3	I	2 Volts	0.8V @ 1.2mA	Has 4K75 pullup to 5V
14 15	RS232 Tx*** RS232 Rx***	O I			Conforms to EIA RS232C & CCITT V.28 Spec.
16	Shutdown**	O	4.26V @ 24mA Sink	0.44V @ 24mA Sink	Totem pole device
17 18 19	Encoder channel A Encoder channel B Encoder channel C	I	2 Volts	0.8V @ 1.2mA	Has 4K75 pullup to 5V
20 21 22	Trigger 1 Trigger 2 Trigger 3	I	2 Volts	0.8V @1.2mA	Has 4K75 pullup to 5V
23 24 25	Address setting 1 Address setting 2 Address setting 3	I	2 Volts	0.8V @ 1.2mA	Has 4K75 pullup to 5V

Notes:

* Step output pulses low for each motor step. Minimum pulse width is 2.5µS; maximum frequency is 200kHz.

** Default normal state is high. This output goes low under a fault condition.

*** Factory default RS232 address is 8. When programming the drive preceed commands with suffix 8, e.g. 8R, 8XSP etc.

All inputs and outputs are TTL compatible. These signals should be opto-isolated in electrically “noisy” environments.

Table 2. Control I/O Specifications

INSTALLATION

Installation Options

PDX drives must be installed by competent personnel familiar with the installation, commissioning and operation of motion control equipment. In the final application the equipment must be enclosed to prevent the operator coming into contact with any high voltages. This includes the drive and motor terminations.

The drives are not EMC compliant, they are sold as a complex component for professional assemblers of motion control systems. Where a system is not required to conform to the European EMC directive the installation procedure described in this Section may be followed. Systems which are to conform to the European EMC directive should be assembled using these procedures and additionally the EMC specific installation recommendations, described at the end of this Section. Digiplan cannot guarantee EMC compliance.

Metal equipment cabinets offer the most advantages for siting the equipment since they can provide operator protection, EMC screening and can be readily fitted with interlocks arranged to remove all AC power when the cabinet door is opened. This form of installation also allows the fitting of metal trays beneath the equipment to act as a flame barrier, which must be provided in the final installation, in accordance with LVD requirements.

Power Connections

Input power is taken directly from the AC supply via the front panel mounted IEC 3-way mains inlet socket. Ensure that the drive is reliably earthed. Any mains wiring should have an insulation rating of at least 1350V, and should be kept separate from the motor and signal wiring.

Wiring Guidelines

Proper grounding of electrical equipment is essential to ensure the safety of personnel. In general, all components and enclosures must be connected to earth ground to provide a low impedance path for ground fault or noise-induced currents. All earth ground connections must be continuous and permanent. A central earth stud is recommended.

Motor Selection

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 with the PDX15, 4.5A peak with the PDX13. 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 with the PDX 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.

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 PDX15-D has the same electrical specification as the PDX15 but incorporates a power dump with a continuous rating of 15 watts (170 watts peak).

You will need the PDX15-D for the following situation:

Metric formula - if the deceleration time $\mathbf{t<\{Jw^2 - 0.1\}}$
where t is the deceleration time in seconds
J is the total system inertia in $Kg\text{-}m^2$
w is the maximum speed in revolutions per second

Imperial formula - if the deceleration time $\mathbf{t<\{0.02Jw^2 - 100\}}$
where t is the deceleration time in milliseconds
J is the total system inertia in $oz\text{-}in^2$
w is the maximum speed in revolutions per second

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

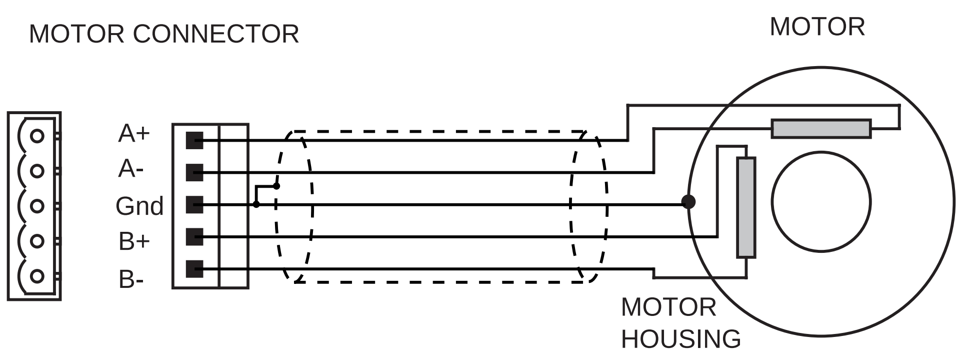
Note that a program to calculate if a power dump is required, is available from Compumotor and Digiplan Technical Support departments (Windows 3.1 required). This program is also available on Compumotor's Bulletin Board Service (Tel: 707/584-4059 in USA).

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. See **Motor Cables** in the **EMC Installation** sub-section.

Motor Connections

Once you have determined the motor's wiring configuration, connect the motor leads to the connector marked "MOTOR" on the front panel. The motor cable earth conductor and screen should be connected to the terminal marked “GND”.



To reverse motor rotation relative to the direction input, interchange connections to A+ and A–.

Figure 1 PDX Drive Motor Connections

Motor Insulation

Motor insulation must be capable of withstanding voltages of at least 500V.

Motor Cable

The recommended gauge for PDX drives is 1mm^2. Use a cable containing five conductors plus the braided screen, the fifth (green) wire being used to provide an earth return to the drive. The temperature rating of the cable should be at least 80°C. The insulation rating should be at least 350V.

Motor Earth

The motor body must be reliably earthed. Also see **Motor Connections** in the **EMC Installation** sub-section. The earth pin on the 5-way plug and socket motor connector is not an appropriate earthing point for the motor safety earth.

WARNING: The case of a motor can become very hot. Precautions may need to be taken to prevent operator contact.

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
Astrosyn, Rapidsyn,	6-lead	Red	Red/Wh	Grn	Grn/Wh	White & Black N.C.
Slo-syn	T.box (x6)	1	3	4	5	2 & 6 N.C.
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
Zebotronics	T.box	1	4	5	8	Link 2 & 3, link 6 & 7
Oriental	6-lead	Black	Green	Red	Blue	Yellow & White N.C.
Sonceboz	8-lead	Green	Grn/Wh	Red	Red/Wh	Link Org & Blk/Wh, link Black & White
Japan Servo	6-lead	Red	Blue	Green	Yellow	2 x White N.C.
Escap	8-lead	Brown	Org/Wh	Red	Yel/Wh	Link Brn/Wh & Org, Link Red/Wh & Yellow.
Bodine	8-lead	Brown	Orange	Yellow	Red	Link Wh/Brn & Wh/Org, link Wh/Yel & Wh/Red.
	T.box	1	3	4	2	Link 5 & 7,link 6 & 8
Digiplan/Compumotor OEM Series	4-lead 8-lead	- Red	- Black	- Green	- White	Internally wired in parallel 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. Motor Connection Data - Windings in Series

N.C. - no connection. For 6-lead motors, connections shown are for one half-winding.

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	
Astrosyn, Rapidsyn, Slo-syn	6-lead	Red	Black	Green	White	Red/Wh & Grn/Wh N.C.
	T.box(x6)	1	6	4	2	3 & 5 N.C.
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	
Zebotronics	T.box	1 & 2	3 & 4	5 & 6	7 & 8	
Oriental	6-lead	Black	Yellow	Red	White	Grn & Blue N.C.
Sonceboz	8-lead	Grn & Blk/Wh	Or & Grn/Wh	Red & White	Blk & Red/Wh	
Japan Servo	6-lead	Red	White*	Green	White*	
Escap	8-lead	Brn & Orange	Brn/Wh & Org/Wh	Red & Yellow	Red/Wh & Yel/Wh	
Bodine	8-lead	Brn & Wh/Or	Wh/Brn & Orange	Yel & Wh/Red	Wh/Yel & Red	
	T.box	1 & 7	3 & 5	4 & 6	2 & 8	
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	

* Use correct White for each phase. † Size 34 only. Size 23 can only be operated in series.

Table 4. Motor Connection Data - Windings in Parallel

Compumotor S and QM Motor Drive Settings

When using Compumotor ‘S’ and ‘QM’ motors you will need to set the PDX drive current settings as shown in Table 5.

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	**PDX13**			**PDX15**			**Peak Motor Current (Amps)**	**Rotor Inertia Kg-cm^2 (oz-in^2)**
	SW6	**SW7**	**SW8**	**SW6**	**SW7**	**SW8**		
S/QM-57-51S	ON	OFF	OFF	*	*	*	1.2	0.088 (0.48)
S/QM-57-51P	ON	OFF	ON	OFF	OFF	OFF	2.3	
S/QM-57-83S	OFF	ON	OFF	*	*	*	1.5	0.234 (1.28)
S/QM-57-83P	ON	ON	ON	OFF	ON	OFF	3.1	
S/QM-57-102S	ON	ON	OFF	*	*	*	1.7	0.32 (1.75)
S/QM-57-102P	ON	ON	ON	ON	ON	OFF	3.5	
S/QM-83-62S	OFF	OFF	ON	*	*	*	2.2	0.64 (3.50)
S/QM-83-62P	ON	ON	ON	OFF	ON	ON	4.4	
S/QM-83-93S	ON	ON	ON	ON	OFF	OFF	2.9	1.23 (6.70)
S/QM-83-93P	X	X	X	ON	ON	ON	5.6	
S/QM-83-135S	ON	ON	ON	ON	ON	OFF	3.5	1.87 (10.24)
S/QM-83-135P	X	X	X	ON	ON	ON	6.9	
S/QM-106-178S	X	X	X	ON†	ON†	ON†	6.0	8.05 (44.0)
S/QM-106-178P	X	X	X	ON†	ON†	ON†	12.0	
S/QM-106-205S	X	X	X	ON†	ON†	OFF†	3.6	9.51 (52.00)
S/QM-106-205P	X	X	X	ON†	ON†	ON†	7.2	
S/QM-106-250S	X	X	X	ON†	ON†	ON†	6.2	12.14 (63.00)
S/QM-106-250P	X	X	X	ON†	ON†	ON†	12.4	

* Minimum drive current too high for motor.
X Unsuitable motor/drive combination.
† 106 (42) size motors must use PDX15-D option (regenerative power dump).

Table 5. PDX Current Drive Settings for Compumotor ‘S’ and ‘QM’ 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 5. Size 34 (83) motors are internally wired in Parallel.

In Table 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	PDX13			PDX15			Peak Motor Current (Amps)	Rotor Inertia Kg-cm^2 (oz-in^2)
	SW6	SW7	SW8	SW6	SW7	SW8		
OEM-57-40S	OFF	ON	ON	OFF	OFF	OFF	2.7	0.07 (0.380)
OEM-57-51S	ON	ON	ON	OFF	ON	OFF	3.3	0.12 (0.650)
OEM-57-83S	ON	ON	ON	ON	ON	OFF	3.8	0.25 (1.360)

Table 6. PDX Current Drive Settings for Compumotor ‘OEM’ Motors

Digiplan SM and STEBON Motor Drive Settings

Table 7 lists the PDX Drive current settings you need to make when using Digiplan ‘SM’ and STEBON motors.

In Table 7, under ‘Motor Type’, a suffix ‘S’ refers to series connected and ‘P’ refers to parallel connected.

Motor Type	PDX13			PDX15			Peak Motor Current (Amps)	Rotor Inertia Kg-cm^2 (oz-in^2)
	SW6	SW7	SW8	SW6	SW7	SW8		
SM-57-51S	OFF	OFF	OFF	*	*	*	1.1	0.11 (0.60)
SM-57-51P	OFF	OFF	ON	*	*	*	2.1	
SM-57-83S	OFF	OFF	ON	*	*	*	2.3	0.23 (1.26)
SM-57-83P	ON	ON	ON	OFF	ON	ON	4.7	
SM-57-102S	OFF	OFF	ON	*	*	*	2.3	0.30 (1.64)
SM-57-102P	ON	ON	ON	OFF	ON	ON	4.6	
SM-83-62S	ON	ON	ON	OFF	ON	OFF	3.2	0.60 (3.30)
SM-83-62P	X	X	X	ON	ON	ON	6.4	
SM-83-93S	ON	ON	ON	OFF	OFF	ON	4.0	1.25 (6.83)
SM-83-93P	X	X	X	ON	ON	ON	7.9	
SM-83-135S	ON	ON	ON	ON	ON	OFF	3.8	2.00 (10.93)
SM-83-135P	X	X	X	ON	ON	ON	7.6	
SM-106-140S	X	X	X	OFF†	ON†	ON†	5.0	3.65 (19.96)
SM-106-140P	X	X	X	ON†	ON†	ON†	9.9	

* Minimum drive current too high for motor.
X Unsuitable motor/drive combination.
† 106 (42) size motors must use PDX15-D option (regenerative power dump).

Table 7. PDX Current Drive Settings for Digiplan ‘SM’ and STEBON Motors

Indexer Connections

Control Signal Connector

Indexer connections are made via the front panel 25-way D-type connector. A typical wiring configuration for this connector is shown in Figure 2.

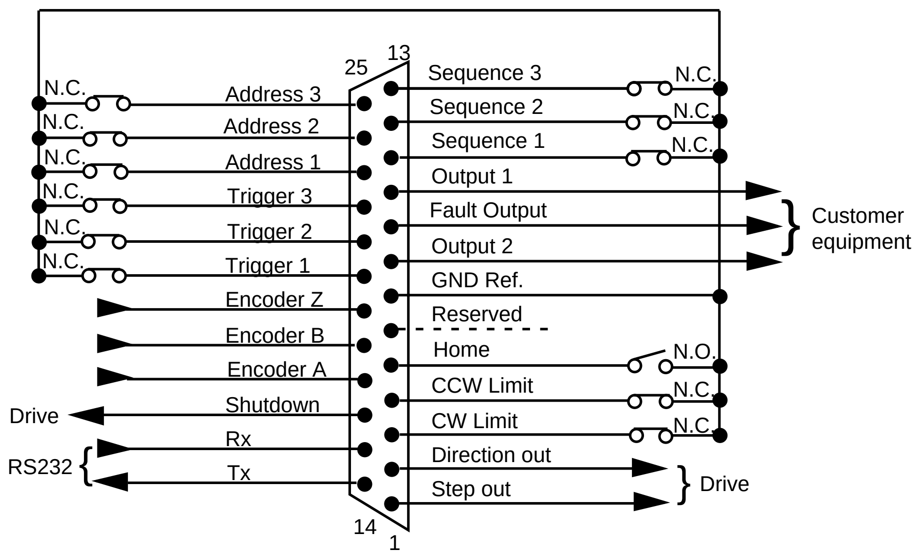


Figure 2 PDX Drive Indexer Connections

Step (Signal 1) & Direction (Signal 2) Outputs

The indexer produces a step and direction output that is identical to the internal step and direction signals. These outputs can be used to slave to another drive or to monitor position and velocity. The direction output's default state is logic high. The step output's default state is a high, pulsing low output. Figure 3 shows the circuit of this output.

The minimum high-level output is 4.26V (source current of 24mA) and the maximum low-level output is 0.44V (sink current of 24mA).

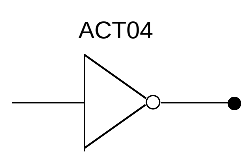


Figure 3 Step and Direction Outputs

CAUTION
All output signals using the ACT04 device have a totem-pole configuration and are not open-collector.

CW (Sig nal 3) & CCW (Signal 4) Limit Inputs

The PDX has two dedicated hardware end-of-travel limit inputs (**CCW** and **CW**). When you power up the drive, these inputs are enabled (high). To test the drive without connecting the **CCW** and **CW** limits, you must disable the limits with the **LD3** command. You can use the Limit Switch Status Report (**RA**) and Input Status (**IS**) commands to monitor the limit status. Figure 4 shows the circuit of these inputs.

The maximum low-level input is 0.8V (sink current of 1.2mA) and the minimum high-level input is 2V.

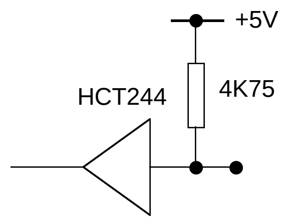


Figure 4 CW and CCW Limit Inputs

Home Position Input (Signal 5)

The PDX has one dedicated home input. The Home input allows you to establish a home reference position. This input is not active during power-up. Figure 5 shows the circuit of the Home input. The maximum low-level input is 0.8V (sink current of 1.2mA) and the minimum high-level input is 2V.

Refer to **Homing Operations** at the end of this section and the **OS** and **GH** commands for more information on the use of this input.

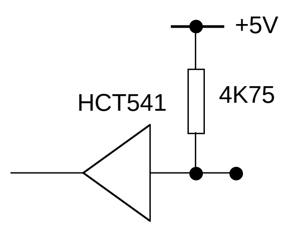


Figure 5 Home Position Input

Reserved (Signal 6)

This signal cannot currently be used to perform any function in this release of the PDX.

Output 1 (Signal 10) and Output 2 (Signal 8)

The PDX has two dedicated programmable outputs. They may be used to signal peripheral devices, for example at the start or completion of a move. The default state for Outputs 1 and 2 is logic low. Refer to the Output (**O**) command for information on using these outputs. Figure 6 shows the circuit used by outputs 1 & 2.

The minimum high-level output is 4.26V (source current of 24mA) and the maximum low-level output is 0.44V (sink current of 24mA).

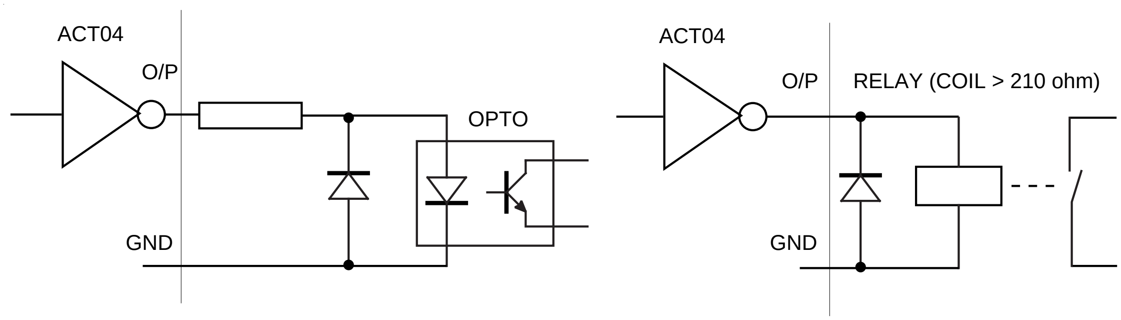


Figure 6 Output Examples

Dedicated Fault Output (Signal 9)

There is a dedicated fault output, which may be used to signal peripheral devices if a unit failure occurs. The Fault output's default state is logic high. Figure 7 shows the circuit of this output.

The output uses a MOSFET, having a maximum low-level output of 0.8V (sink current of 50mA) and a high-level output of 5V.

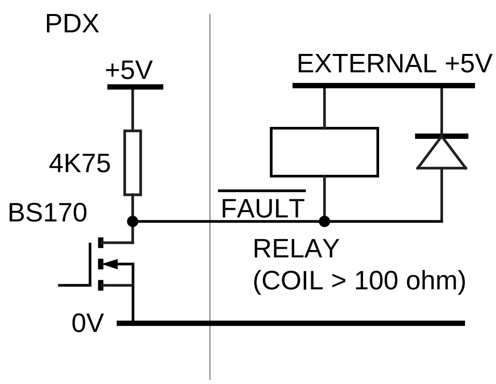


Figure 7 Dedicated Fault Output

Sequence Inputs 1 - 3 (Signals 11 - 13)

The PDX has three dedicated sequence inputs that allow you to control seven different sequences. Refer to the **X** commands for information on how to control these inputs. Sequence No.Ø is not a valid sequence.

Sequences are executed remotely by using one of the following logic patterns (1 represents a +5V signal, Ø represents 0V signal).

Sequence No.	**Ø**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
SEQ Input 1	Ø	1	Ø	1	Ø	1	Ø	1
SEQ Input 2	Ø	Ø	1	1	Ø	Ø	1	1
SEQ Input 3	Ø	Ø	Ø	Ø	1	1	1	1

Ø = low, pulled to ground
1 = high, 5VDC

Table 8. Sequence Input Patterns

The circuit of the sequence inputs is given in Figure 8. The maximum low-level input is 0.8V (current sink of 1.2mA) and the minimum high-level input is 2V.

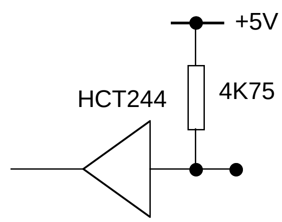


Figure 8 Sequence Inputs

Sequence Interrupted Run Mode

When using sequence interrupted run mode (command XQ) the circuit shown in Figure 9 can be used to setup and load sequences.

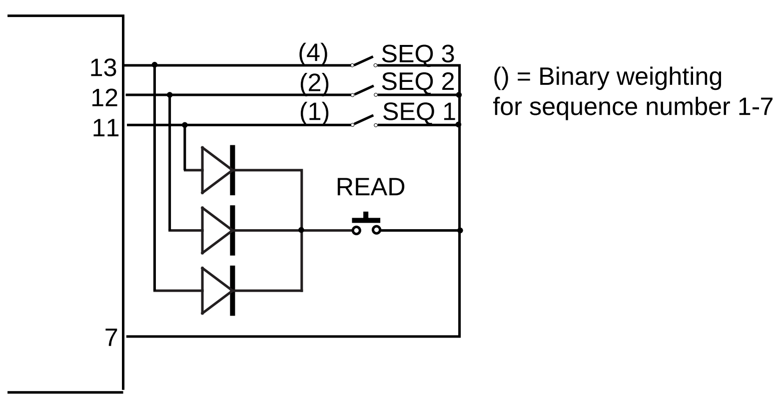


Figure 9 Circuit to Load Sequences

Example Program using External Sequence Selection

Instruction	**Description**
8XE5	Erase the old sequence
8XD5	Define a new sequence number 5
8O00	Ensure both outputs are off
8OSB1	Motor will backup to home switch for backlash compensation
8OSC0	Active state of home input is closed i.e. normally open switch used
8A10	Set acceleration rate for homing move
8GH-2	Execute homing move at a move velocity of 2 rps in the CCW direction
8O10	Set output 1 active (home complete)
8XT	Denote end of sequence
8XE6	Erase the old sequence
8XD6	Define a new sequence number 6
8O00	Ensure both outputs are off
8D40000	Set move distance in motor steps
8A15	Set acceleration/deceleration rate for move
8V5	Set move velocity (rps)
8L5	Repeat 5 times
8G	Perform move
8T1.0	Wait 1 second
8N	End of loop
8O01	Set output 2 active (indexing complete)
8XT	Denote end of sequence
8XP9	Immediate, auto-saved command - PDX will now always run sequence selected by inputs on power-up or after a software reset (Z command) has been transmitted
8Z	Reset controller

When sequence select inputs are configured as Seq1 & Seq3=high and Seq2=low, the homing sequence will run. When inputs are configured as Seq1=low and Seq2 & Seq3=high, the indexing moves will be executed. Note that if the sequence inputs have not changed when the sequence of commands have finished, the same sequence will be called again (see XQ1 command to disable this feature).

RS-232C—Tx (Signal 14), Rx (Signal 15), and Ground (Signal 7)

The PDX uses RS-232C as its communication medium. This indexer does not support handshaking. A three-wire (Rx, Tx, and Signal Ground) configuration is used. Figure 10 represents a typical RS-232C configuration.

The factory default address is 8 (as set by input terminals 23, 24 & 25). See # command to change this by software.

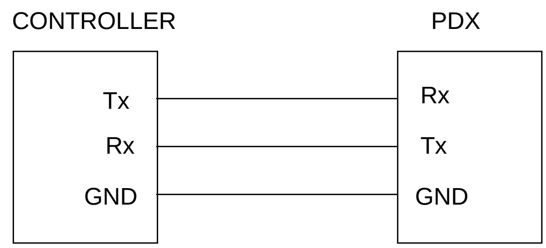


Figure 10 RS232 Connections

CAUTION
RX, TX and GND pinouts are not 2, 3 and 7 as in most interfaces

Shutdown Output (Signal 16)

The drive produces a Shutdown output that is identical to the indexer's internal signal. This output may be used to slave to another drive or to monitor the PDX. The Shutdown output's default state is logic high. Figure 11 shows the shutdown output circuit.

The minimum high-level output is 4.26V (source current of -24mA) and the maximum low-level output is 0.44V (sink current of 24mA).

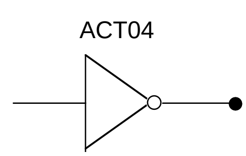


Figure 11 Shutdown Output

Closed Loop Operation

Closed loop moves require an external encoder to provide position correction signals. Motor position may be adjusted to reach the desired position. To implement the closed loop functions, you must connect a single ended, incremental, optical encoder to the PDX. When an encoder is used, the following functions will be added to the system:

- Encoder referenced positioning
- Encoder position servoing
- Motor stall detection
- Higher accuracy homing function - see OSD command
- Multi-axis stop (also available without an encoder - see **FSF** command)

Encoder Inputs A,B,Z (Signals 17 - 19)

The PDX has three dedicated inputs for use with a single ended incremental encoder (NPN open collector). These inputs in conjunction with the **FS** commands will determine the encoder functionality.

Figure 12 shows the encoder input circuits.
The maximum low-level input is 0.8V (current sink of 1.2mA) and the minimum high-level input is 2V.

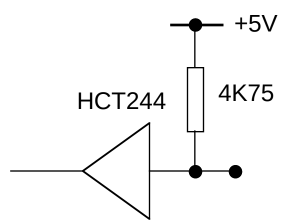


Figure 12 Encoder Inputs

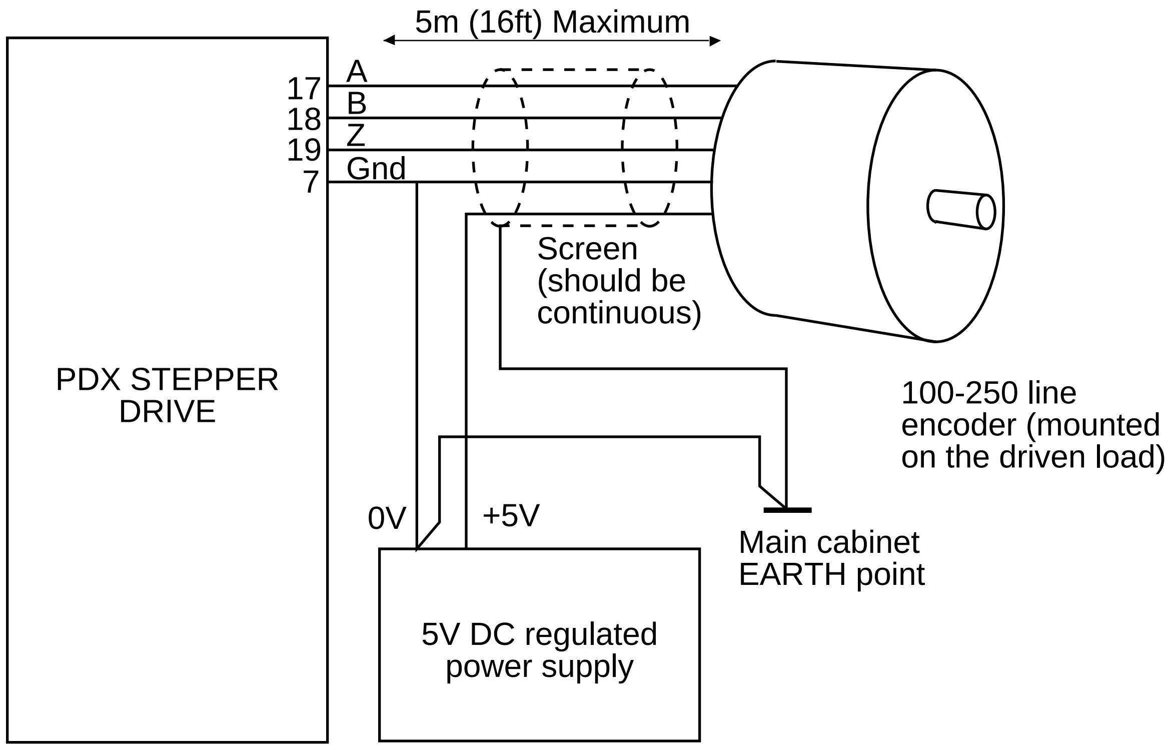


Figure 13 Wiring Example for PDX - Encoder Option

Example Encoder Setup

Drive is setup for 4000 step/revolution. The encoder has 100 lines so the post-quadrature resolution is 400 counts/revolution. The encoder is mounted on the load and there is no gear reduction between the motor/encoder and load (i.e. ratio 1:1). Stall detection and position maintenance are required. Sequence 1 will contain a power-up, initialisation program as follows:

Instruction	Description
8XE1	Erase the old sequence
8XD1	Define a new sequence number 1
8MRD4000	Drive switches 4 & 5 are OFF=4000 steps/rev
8ER400	Post-Quadrature Encoder counts/rev
8DW200	Stall condition if encoder lags motor by >200 **motor** steps
8DB30	Corrective move if encoder is >30 **encoder** steps away from expected final position
8CG8	Try to correct position by total position error on first corrective move attempt
8FSB1	Index distances will use encoder counts/rev as units
8FSC1	Enable position maintenance
8FSH1	Enable stall detection
8FSE1	Turn on output 1 if a stall occurs
8FSG1	Output 2 goes active when the motor is stationary and within position deadband

.....Additional commands required

8XT	Denote end of sequence
8XP1	Immediate, auto-saved command - PDX will always run sequence 1 on power-up or after a software reset (Z command)

All subsequent moves now have distances set in encoder counts/rev and will operate in a closed-loop mode.

Trigger Inputs 1- 3 (Signals 20 - 22)

The drive has three dedicated Trigger inputs, which are pulled up internally. These inputs are used with the Trigger (**TR**) command to control the trigger function. Figure 14 shows the circuit of these inputs.

The maximum low-level input is 0.8V (current sink of 1.2mA) and the minimum high-level input is 2V.

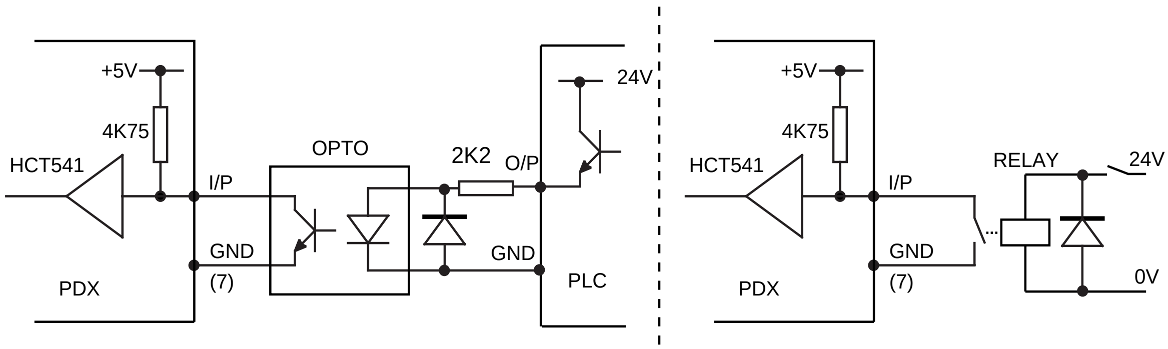


Figure 14 Trigger Input Example Connections

Address Signals 1 - 3 (Signals 23 - 25)

The PDX has three dedicated address inputs that allow you to specify a unique address for each drive in your system. Units may be assigned an address from 1 to 8. Each unit in the configuration must have a unique address. The default address is 8 (all three inputs are internally pulled up). The address inputs are read only during power-up and when Restart (**Z**) commands are issued. Use the matrix below to assign unique address values.

Address No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Address 1	Ø	1	Ø	1	Ø	1	Ø	1
Address 2	Ø	Ø	1	1	Ø	Ø	1	1
Address 3	Ø	Ø	Ø	Ø	1	1	1	1

Ø = low, pulled to ground
1 = high, 5VDC

Table 9. Address Selection Matrix

Figure 15 shows the circuit of one of the address line inputs.
The maximum low-level input is 0.8V (current sink of 1.2mA) and the minimum high-level input is 2V.

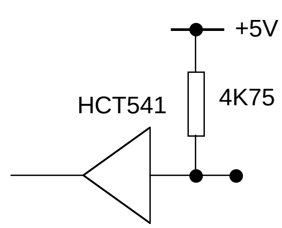


Figure 15 Address Inputs

Daisy Chaining

You may daisy chain up to 8 PDX drives (or up to 255 if software addressing is used - please refer to the # command). Individual drive addresses are set with signals 23, 24, and 25 on the 25-pin D connector. When daisy chained, the units may be addressed individually or simultaneously. You should establish a unique device address for each drive. Refer to the Figure 16 below for daisy chain wiring.

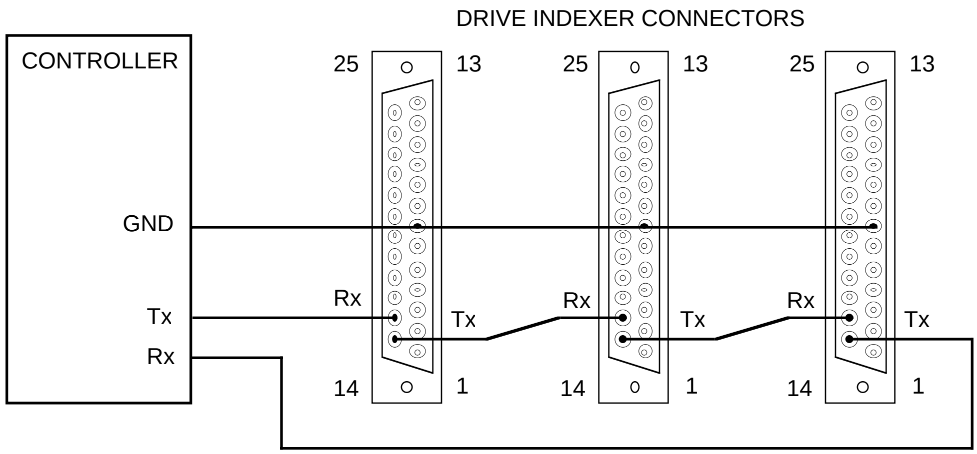


Figure 16 Daisy Chain Connection of Drives

Commands prefixed with a device address control only the unit specified; commands without a device address control 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. For example, 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 PDX, all units in a 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 PDX executes a device-specific command unless the unit number specified with the command matches the drive's unit number. Device-specific commands include both buffered and immediate commands.

Homing Operations

The term ‘homing’ refers to an automatic return to a mechanical reference position which is usually performed when the system is first powered up. All subsequent moves will then be relative to this reference position. The home position is usually determined by an optical or proximity switch, though a mechanical switch can also be used. When very high accuracy is required, the home position can be referenced to the index track of an encoder which eliminates errors due to switch repeatability.

Setting up the homing configuration

There are four commands which determine the homing setup - they are all from the **OS** group (‘Other Switches’).

OSB - Back up to Home

This command determines whether or not the motor backs up to one edge of the home switch region. With backup enabled (**OSB1**), the home position will be more accurate but the homing operation will take longer. With backup disabled (**OSB0**), the home position can lie anywhere within the range of the Home switch. This may be more appropriate when frequent homing operations are needed and speed is more important than accuracy.

OSC - Active state of Home switch

The OSC command determines whether the active state of the Home input is high or low. If a mechanical switch is used, use **OSC0** if the switch is **closed** in the home region, and **OSC1** if the switch is **open** in the home region. The switch is connected between the Home input and 0v. If a proximity detector is used this should have an NPN output, and the command to use is normally **OSC0** assuming that the switch detects metal present in the home region.

OSD - Enable homing to encoder Z channel

In applications using an encoder, the Z channel or index track can be used to give a highly-repeatable home position. The normal home switch is still required to give a coarse home position, whilst the index track gives a precise shaft location within the range of the switch. Variations in the operating point of the switch will then have no effect on the final home position. Use **OSD1** to home to the index track, and **OSD0** to disable this function or when no encoder is fitted.

OSH - Reference edge of home switch

When the system is configured to back up to the home switch (**OSB1**), the OSH command determines which edge of the switch is used. The switch edges are referred to as CW and CCW. The CW edge is defined as the first switch transition seen when travelling away from the CW limit switch in the CCW direction. **OSH0** will set the CW edge as the reference edge, **OSH1** the CCW edge. If backup is disabled (**OSB0**), it makes no difference how OSH is set.

Which way is home?

The homing operation is initiated by the **GH** (Go Home) command, and this will determine the initial search direction The command must be followed by the search speed, so **GH1** will start the system looking for home at

1 rps - the direction is assumed to be CW if no sign is included. **GH-2** will begin a home search at 2 rps in the CCW direction. If the ‘wrong’ direction is chosen, i.e. the system moves away from the home region, it will reverse when it encounters a limit switch and resume the search in the opposite direction. If the next switch seen is the opposite limit switch, it is assumed that the home switch has failed and the operation is aborted.

Note that under certain conditions the direction given as part of the GH command may be ignored - see below.

Homing approach patterns with backup

Figure 17 shows the various approach patterns with backup enabled (**OSB1**). These patterns show the route taken when starting from either side of the home switch and in either direction. Regardless of the approach direction, the motor decelerates to rest at the programmed **A** rate when it sees the selected edge of the Home switch. A secondary move is then made to position the system slightly *outside* the home region. The final approach to home is made at 0.1 rps towards the switch. Since this final move is within the start-stop region of the motor, no deceleration is required and the system stops accurately on the switch edge. Note that the final move is always made over the same distance in the same direction, regardless of which way the home region was approached.

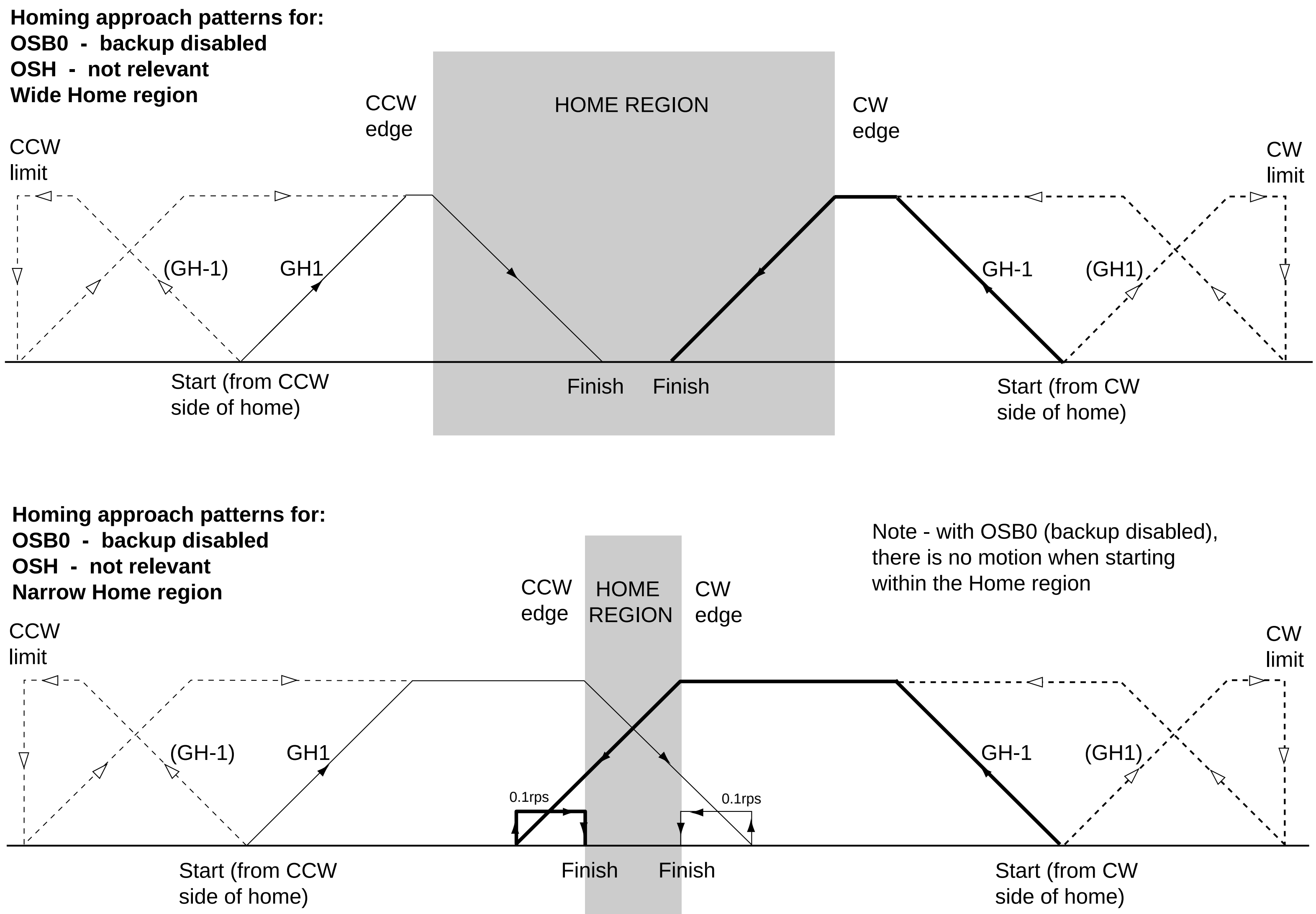


Figure 17 Homing Approach Patterns With Backup

If the GH command is sent with the system already in the home region, the final two stages of the normal homing operation are performed as shown in Figure 18. The motor is first positioned slightly outside the home region and then approaches the selected edge at 0.1 rps. In this instance the direction required to approach the selected edge can be determined, so the indexer will disregard any direction given with the GH command - it will always make its initial move towards the CW or CCW edge as selected by OSH.

Homing approach patterns for:
OSB1 - backup enabled
System starts inside home region
Search direction depends on selected edge (OSH)
GH1 and GH-1 have same effect

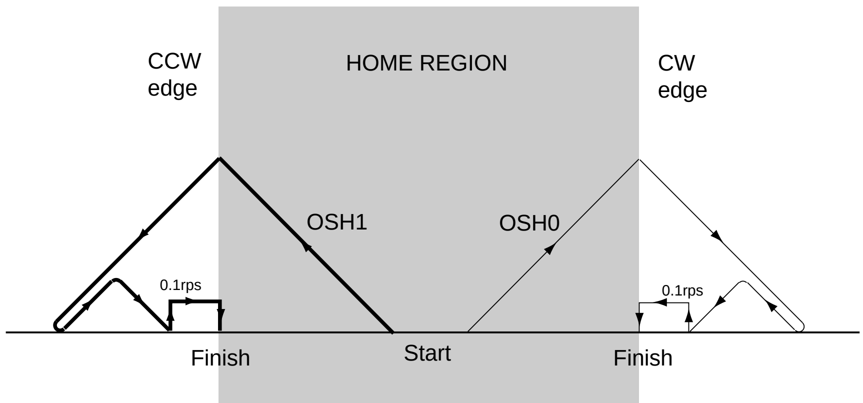


Figure 18 System Already in the Home Region

Homing approach patterns without backup

Figure 19 shows the approach patterns when backup is disabled (**OSB0**). The motor decelerates at the programmed **A** rate when it sees the first edge of the home switch, and will be considered to be 'at home' if it stops with the switch still active. If it overshoots the switch, it will back up at 0.1 rps until it sees the *first* edge of the switch. In this mode, the final home position can therefore lie *anywhere within the range of the home switch*. If the system is already in the home region when the GH command is issued, there will be no movement since the 'at home' condition is already satisfied.

Homing to the encoder Z channel

The approach patterns when homing to the index track are basically the same as those shown in Figs A and B. However, instead of stopping at the selected switch edge, the motor continues until the Z channel pulse appears and this sets the home position. Note that to home to the index track you must have backup enabled (**OSB1)** and be operating in Encoder Step Mode (**FSB1**).

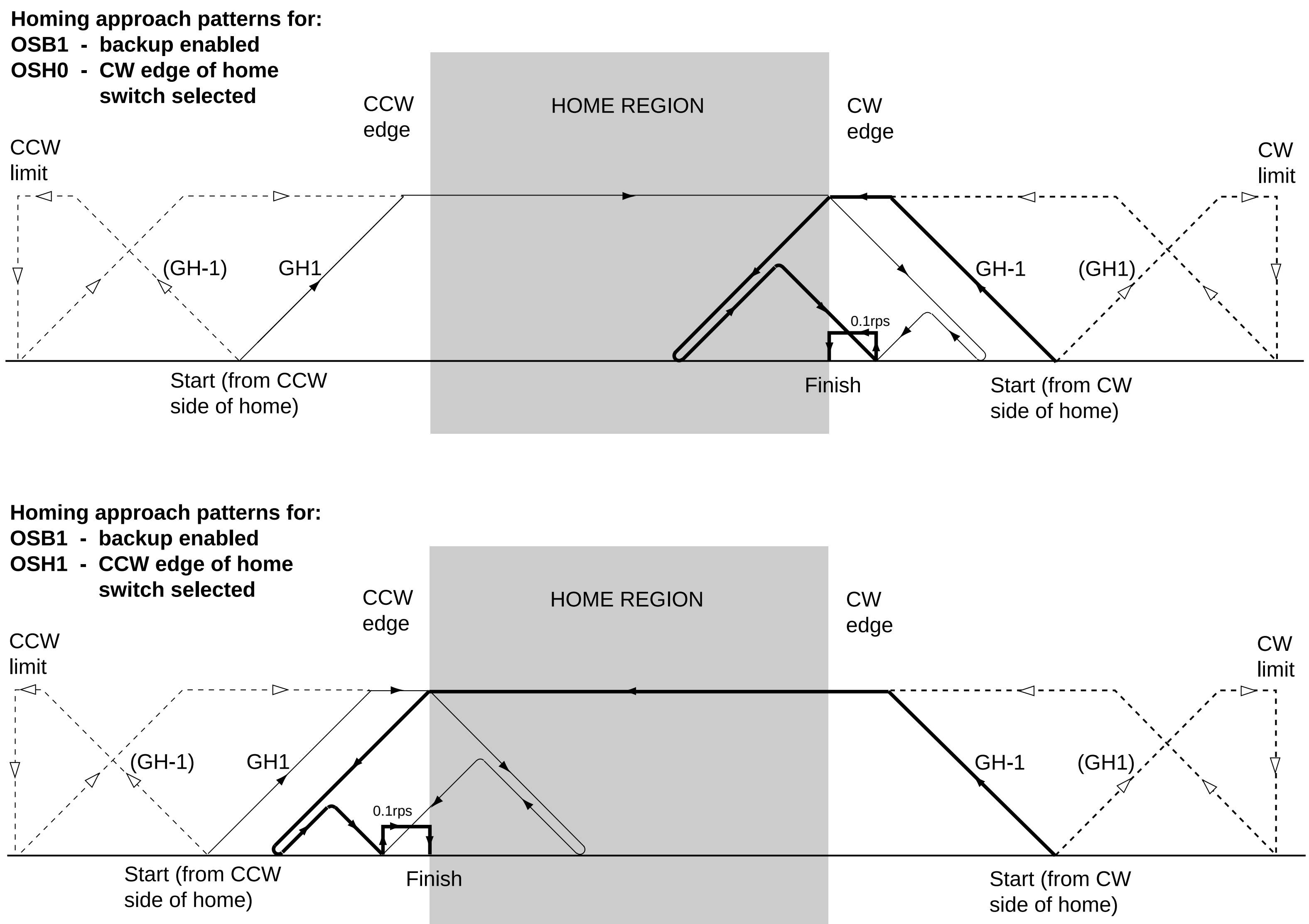


Figure 19 Homing Approach Patterns Without Backup

Application considerations

In all applications, you need to confirm the active state of the home switch using **OSC**.

For *fast homing,* where high accuracy is not needed, use **OSB0**. The setting of OSD and OSH is irrelevant. If the home position is always approached at the same speed and from the same direction using the same acceleration rate (**A**), the home position will show reasonable repeatability even with a wide home region.

For *higher accuracy*, use **OSB1.** This will increase the homing cycle time. You can select the active edge of the home switch with OSH, but in most cases it will not make much difference which you choose. It is usually only important that the home position is repeatable - the actual position only affects the distance of the initial move away from home. Where the home position must be a specific mechanical location, for instance a park position, you will need to adjust the home sensor or its target. In this case the ability to select the CW or CCW edge may be useful.

For *total repeatability* to the same motor step, an encoder is required. Use **OSB1** and **OSD1**, and operate in Encoder Step Mode (**FSB1**).

The *search speed* issued with the GH command may need experiment for the best results. Try a high speed if long homing moves are typical - the time saved may offset the longer final search time caused by increased overshoot beyond the switch.

The *home switch position* is frequently chosen to be near one end of the working range - this tends to simplify fitting and adjusting the home sensor. However, try to avoid positioning it so close to a limit switch that the limit may be hit during deceleration.

Mechanical/ Environmental

Enclosure Considerations

The drive and its switch mode power supply are contained in a single case measuring 250mm (9.84 inches) High, by 50mm (1.97 inches) Wide, by 190mm (7.5 inches) Deep. *Note: Enclosure depth does not take connector dimensions into account. These need an additional 60mm.*

You should install the drive system in an enclosure to protect it against atmospheric contaminants such as oil, moisture, dirt etc. This also prevents operator access and assits the EMC performance by limiting emissions and adding to the rf immunity.

Ideally, you should install the system in a cabinet. In the USA, the National Electrical Manufacturers Association (NEMA) has established standards that define the degree of protection that electrical enclosures provide. The enclosure should conform to NEMA Type 12 standards if the intended environment is industrial and contains airborne contaminants. Proper layout of components is required to ensure sufficient cooling of equipment within the enclosure.

Environmental Specifications

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

- Operating Temperature: 0° to 50°C (32° to 122°F)
- 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.

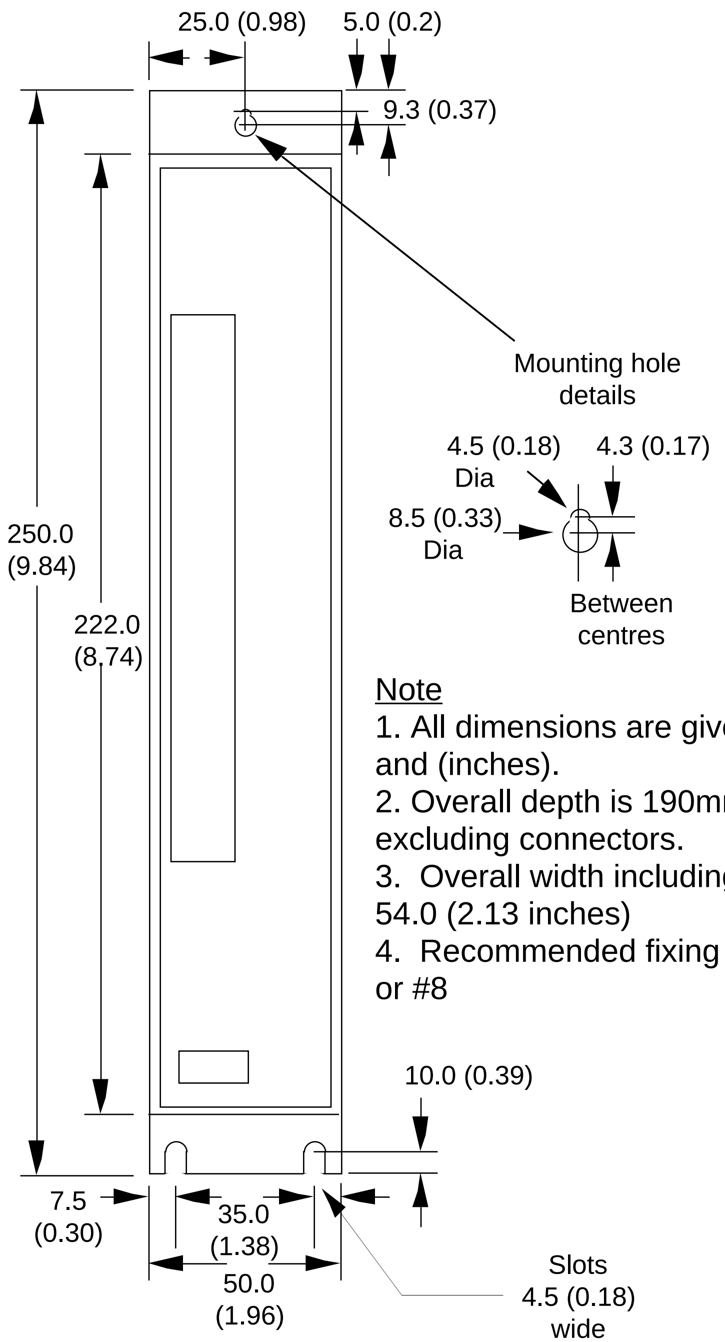
In exceptional circumstances, such as running the motor continuously at maximum current, forced-air cooling may be needed to maintain the local ambient temperature within specification.

The mains input to the drive should be Installation Category II maximum.

The PDX series of drives can be used in a Pollution Degree 2 environment i.e., one in which only non-conductive pollution occurs.

Installation Considerations

The drive is designed to be installed vertically as shown in Figure 20. 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.



<u>Note</u>
1. All dimensions are given in millimetres and (inches).
2. Overall depth is 190mm (7.5 inches) excluding connectors.
3. Overall width including screw heads 54.0 (2.13 inches)
4. Recommended fixing screw size M4 or #8

Figure 20 Drive Mounting Hole Locations

EMC Compliant Installation

It should be stressed that although these recommendations are based on the expertise acquired during the development of fully compliant products, and on tests carried out on each of the product types, it is impossible for Digiplan to guarantee the compliance of any particular installation. This will be strongly influenced by the physical and electrical details of the installation and the performance of other system components. Nevertheless it is important to follow *all* the installation instructions if an adequate level of compliance is to be realisable.

External Enclosures

The measures described in these recommendations are primarily for the purpose of controlling conducted emissions. To control radiated emissions, all drives and rack systems must be installed in a steel equipment cabinet which will give adequate screening against radiated emissions. This external enclosure is also required for safety reasons. With the exception of drive front panels in rack-based units, there must be *no user access* while the equipment is operating. This is usually achieved by fitting an isolator switch to the door assembly. PDX drives and filters must be mounted to a conductive panel. If this has a paint finish, it will be necsssary to remove the paint in certain areas where specified.

To achieve adequate screening of radiated emissions, all panels of the enclosure must be bonded to a central earth point. The enclosure may also contain other equipment such as motion controllers, and the EMC requirements of these must be considered during installation. Always ensure that drives and rack systems are mounted in such a way that there is adequate ventilation.

Before mounting the drive, remove the paint from the rear face of the lower mounting lug as shown in Fig. 22, and if necessary from the corresponding area on the rear panel of the enclosure. This is to guarantee a good high-frequency connection between the drive case and the cabinet. Use petroleum jelly on the exposed metal to minimise the risk of future corrosion.

AC Supply Filtering

These recommendations are based on the use of proprietary mains filter units which are readily available. However, the full EMC test includes a simulated lightning strike which will damage the filter unless adequate surge suppression devices are fitted. These are not normally incorporated into commercial filters since the lightning strike test can be destructive. This test is normally carried out on the overall system and not on individual components, therefore the surge protection should be provided at the system boundary.

Try to arrange the layout of drive and filter so that the AC input cable is kept away from the filter output leads. It is preferable for the current path to be as linear as possible without doubling back on itself - this can negate the effect of the filter. Mount the filter within 50mm of the drive, and run the input cable and any earth cables close to the panel.

PDX drives incorporate a switch-mode power supply operating directly from the AC input. The substantial filtering effect of a mains isolation transformer is therefore not available, and additional external filtering is required. The solution offered uses a single filter in order to control both differential and common-mode emissions. The manufacturer’s part number for a suitable filter is:

CORCOM 6EQ1

Mount the filter within 50mm of the drive as shown in Fig 22. Again ensure that there is no paint on the rear panel behind the filter mounting lugs - it is vital that there is good large-area contact between the filter and the panel.

Mains Cable

Connect the incoming AC supply cable to the push-on terminals on the first filter, with the earth lead connected to a local earth stud or bus bar. Connect the earth terminal on the case of each filter to the earth stud. Route the supply cable so that it runs close to the rear panel within the cabinet.

3-core 1mm^2 screened cable (with a braided screen) must be used between the output of the filter and the input to the drive. Connect the earth wire to the earth stud, and arrange all the earth leads so thay they run close to the panel. Expose a short length of the screen and anchor the cable close to the filter with a P-clip. Remove any paint from the panel behind the P-clip. Fit a ferrite absorber over the cable and wire up the power connector - no connection is made to the screen at the drive end. Locate the absorber as close as possible to the connector using heat-shrink sleeving.

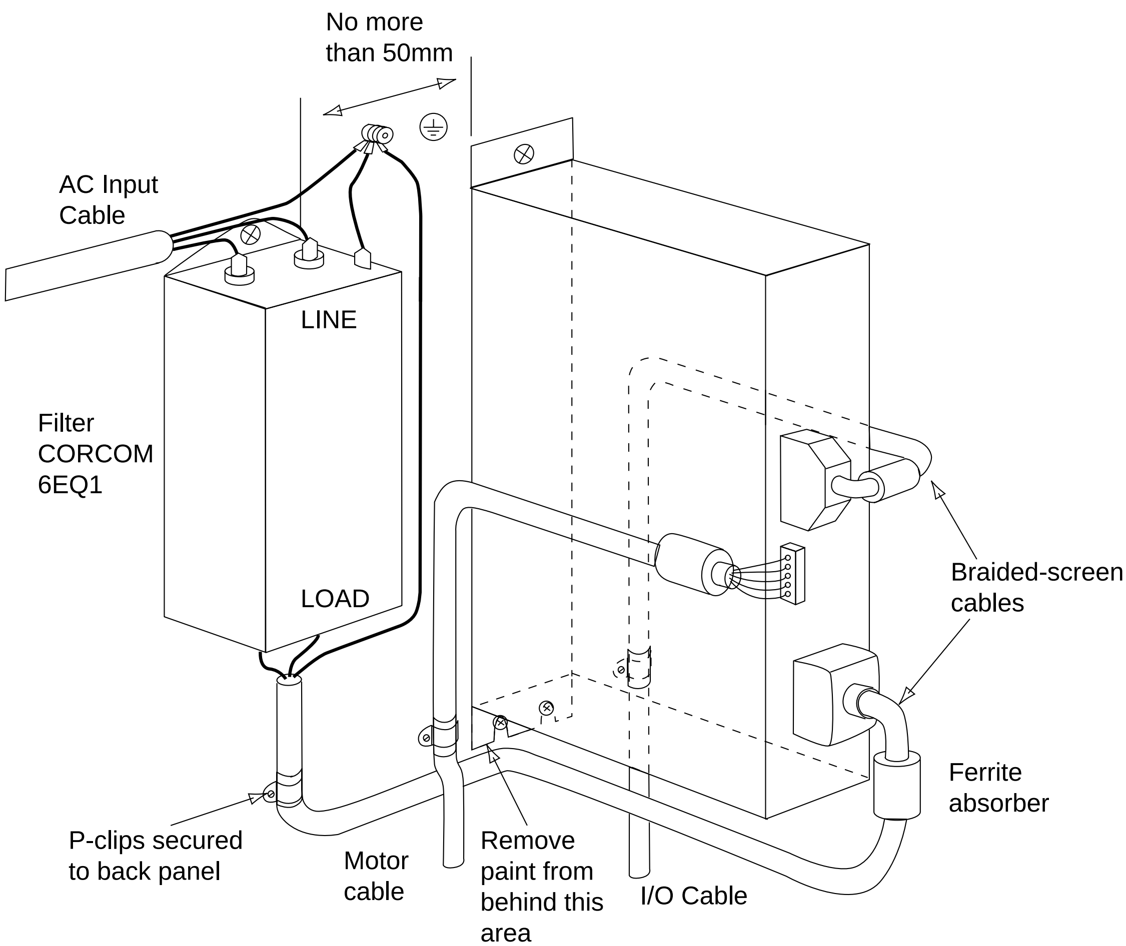


Figure 22. EMC Installation

Motor cables

All motor connections must be made using a high quality braided-screen cable. Cables using a metallised plastic bandage for an earth screen are unsuitable and in fact provide very little screening. There is a problem in terminating to 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° high optical coverage that the screen 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 chopping current back to the drive. This may require mounting the connector on a sub-panel insulated from the main cabinet, or using a connector having an internal screen which is insulated from the connector housing.

Motor cables (continued)

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 or rack system.

Stepper motors

It is preferable to use motors with screw terminations whenever possible. If flying-lead motors are used, it is important that the unscreened leads are converted into a braided-screen cable within 100mm 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. Motors fitted with terminal boxes also allow local selection of series or parallel connection, reducing the cost of the cable running back to the drive.

Motor connections

Use 5-core $1mm^2$ screened cable for the motor connections. At the drive end, fit a ferrite absorber over the cable before wiring to the motor connector. No connection is made to the cable screen at this end. Locate the absorber as close as possible to the connector using heat-shrink sleeving.

Run the motor cable back to the rear panel and down between the drive and the filters. Expose a short length of braiding and anchor to the rear panel with a P-clip. Note that the motor cable should preferably be kept at least 300mm away from I/O cables carrying control signals.

Termination at the motor must be made using a 360° bond to the motor body, and this may be achieved by using a suitable clamp. Many stepper motors are designed to accommodate an appropriate conductive terminal gland which can be used for this purpose.

Control signal wiring

High-quality braided screen cable should be used for control connections. In the case of the PDS drive which has differential step-direction inputs, it is preferable to use cable with twisted pairs to minimise magnetic coupling. No connection is made to the cable screen at the D-connector on the drive. Fit a ferrite absorber close to the D-connector and run the cable back to the rear panel as shown in Fig. 22. Expose a short length of the braided screen and anchor to the rear panel with a P-clip.

Control signal wiring (continued)

The PDHX indexer-drive has I/O signals operating at TTL levels and these are unlikely to meet EMC immunity requirements if taken outside the enclosure. Should this be necessary, route the signal via a separate opto-isolator which allows the external signal to operate at 24V levels.

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 like a high resistive 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	80Ω
Impedance at 100MHz	120Ω
Curie temperature	130°C (the device should not be operated near this temperature)

Handling and installing the ferrite absorbers

Take care when handling the absorbers - they can shatter if dropped on a hard surface. For this reason the suggested 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.

Setting Up

Drive Switch Settings

These setting are intended to be changed only by qualified service personnel. Operator access to the drive should not be permitted. Take care, unexpected motion may occur at any time, especially during the commissioning of motion control equipment.

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

Table 10. Switch Settings

Selftest Switch 1

The selftest switch is used to check the operation of the drive. Set to the ON position will cause the motor to rotate at 1 rev/sec.
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

Switches 4 and 5 determine the step resolution of the drive, as defined in Table 11. Use the MRD command to configure the indexer to the same setting. *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 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 12.

SWITCH SETTINGS			PDX13 PEAK CURRENT	PDX15 PEAK CURRENT
6	**7**	**8**		
ON	ON	ON	3.0A	5.0A
OFF	ON	ON	2.7A	4.6A
ON	OFF	ON	2.4A	4.3A
OFF	OFF	ON	2.1A	3.9A
ON	ON	OFF	1.8A	3.6A
OFF	ON	OFF	1.5A	3.2A
ON	OFF	OFF	1.2A	2.9A
OFF	OFF	OFF	0.9A	2.5A

Table 12. Peak Current Settings

Preliminary Testing for Service Personnel Only

Initially do not make any connections to the 25-way D-type connector. With the power OFF, ensure that the drive current is set correctly for the motor, confirm that the motor is correctly wired and free to rotate. Ensure the motor is securely clamped in position. Apply power and, if it is safe to do so, check the motor shaft to see if there is holding torque. The red FAULT LED should be OUT and the green POWER LED should be LIT. If all is well, close switch 1 (self test) and the motor should start turning. The velocity is preset to approximately 1 revolution per second. This should confirm that the drive-motor wiring is OK. Note: when running self test an internal clock source is used and the indexer is by-passed. At the end of this test power down and turn OFF switch 1.

To test the indexer, connect the RS232 serial port (terminals 14, 15 & 7) to your computer terminal and load a communications program (e.g. X-WARE). Power up the drive. If using X-WARE, select the OEMX/PDX option from the PRODUCT menu option and then select the CHECK OUT option on the SETTINGS popup menu. If the program reports back “Checks Out Fine”, select the CONNECT option from TERMINAL menu on the main menu bar. If the program reports “Bad RS232 Setting”, confirm that your wiring is correct.

If you are using a Terminal Emulation program other than XWARE, set your terminal program to 9600 Baud, No parity, 8 Data bits, 1 Stop bit and choose the appropriate serial port (COM#) of your computer.

Decouple the motor from the load for safety, note the settings of drive switch 4 and 5 (resolution) and type the following commands:

8LD3	This disables end of travel limits (required if these are not wired)
8MRD<resolution>	Sets the indexer resolution - options are 400, 1000, 2000, or 4000
8G	Tell the drive to start the motor moving

Working directly (“on-line”) using a terminal emulation program each single command must be followed by a space or a <CR>. Do not edit these commands, if you make a typing mistake, type a space followed by the correct command.

The indexer must be set to match the drive switch setting for resolution. On entering the ‘G’ command (GO), the motor should produce a 1 revolution move in the clockwise direction - at a velocity of 1 rev/sec.

You are now ready to start your application design.

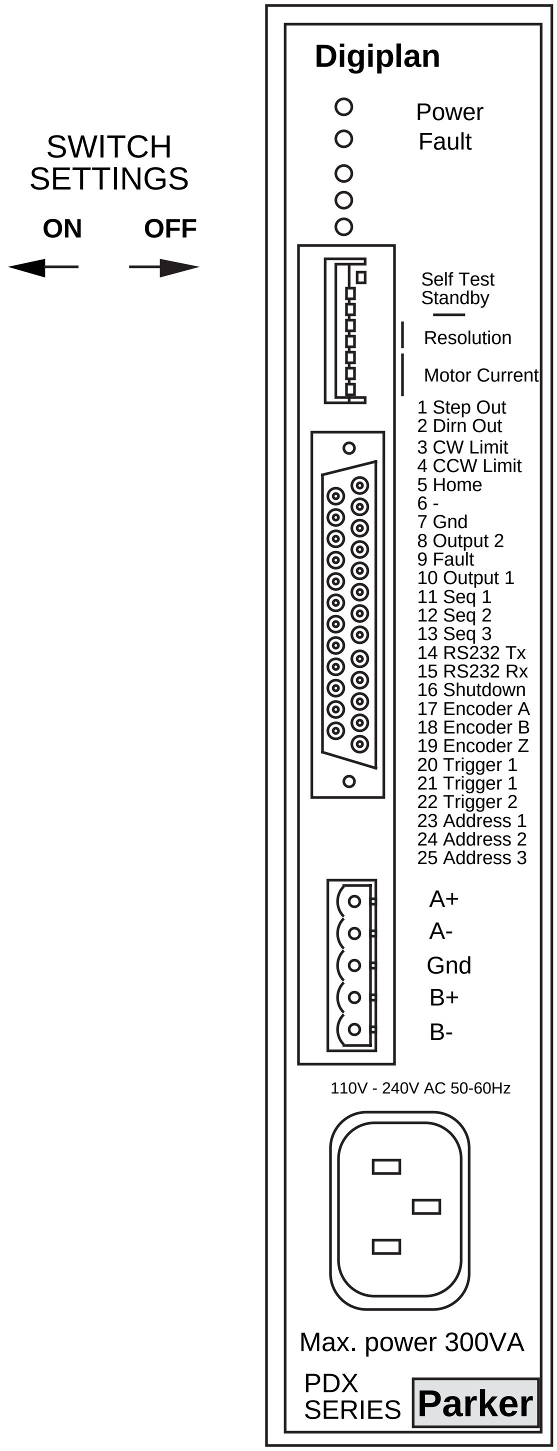


Figure 21. Front Panel Layout

Software Reference

Chapter Objectives

Use this chapter as a reference for the function, range, default, and sample use of each command for the Indexer.

Command Description

Command Mnemonic

The beginning of each command entry contains the command's mnemonic value and full name.

Command Type

Set-Up—Set-up commands define application conditions. These commands establish the output data format from the indexer.
Motion—Motion commands affect motor motion, such as acceleration, velocity, distance, go home, stop, direction, mode, etc.
Programming—Programming commands affect programming and program flow for trigger, output, all sequence commands, time delays, pause and continue, enable and disable, loop and end-loop, line feed, carriage return, and backspace.
Status—Status commands respond (report back) with data. These commands instruct the system to send data out from the serial port for host computer use.

Syntax

The proper syntax for the command is shown here. The specific parameters associated with the command are also shown. If any of these parameters are shown in brackets, such as <a>, they are optional. The parameters are described below.

- **a** indicates that a device address must accompany the command. Only the device specified by this parameter will receive and execute the command. Valid addresses are 1-8 when selected by inputs, and 1-255 when selected by software.
- **n** represents an integer. An integer may be used to specify a variety of values (acceleration, velocity, etc.).
- **s** indicates that a sign character, either positive or negative (+ or -), is required.
- **x** represents any character or string of characters.

Range

This is the range of valid values that you can specify for n (or any other parameter specified).

Attributes

This first attribute indicates if the command is *immediate* or *buffered.* The system executes immediate commands as soon as it receives them. Buffered commands are executed in the order that they are received with other buffered commands. Buffered commands can be stored in a sequence.

The second attribute explains how you can save the command.

- Savable in Sequence
- Never Saved
- Automatically Saved

Savable in Sequence commands are saved when they are defined in a sequence (see **XT** command). *Savable in Sequence* commands can be stored in system memory (nonvolatile) and retained when power is removed from the system. A command that is *Never Saved* is executed without being saved into the system's permanent memory. *Automatically Saved* commands are automatically saved into memory upon execution.

Valid Software Version

This field contains the current revision of the software in which the command resides at the time this user guide was released.

Units

This field describes what unit of measurement the parameter in the command syntax represents.

Default Value

A command will perform its function with the default setting if you do not provide a value.

See Also

Commands that are related or similar to the command described are listed here.

Response

A sample system response is shown. When the command has no response, this field is marked N/A.

Alphabetic List of Commands

A Acceleration

Software Version: A

Syntax	Units	Range	Default Value
<a>An	revs/sec^2	n = 0.01-999.00	100

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Motion	D, G, MRD, V	*An

The Acceleration command specifies the acceleration rate to be used for the next Go (**G**) command. The acceleration remains set until you change it. You do not need to reissue this command for subsequent Go (**G**) commands. Accelerations outside the valid range cause the acceleration to remain at the previous valid **A** setting.
If the Acceleration command is entered with only a device address (**1A**), the indexer will respond with the current acceleration value. If a move is commanded without specifying an acceleration rate, the previously commanded acceleration rate will be used.

Example

Command	Description
A1Ø	Sets acceleration to 10 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D2ØØØ	Sets distance to 2,000 steps
G	Executes the move

B Buffer Status

Software Version: A

Syntax	Units	Range	Default Value
aB	N/A	N/A	N/A
Attributes	**Command Type**	**See Also**	**Response**
Immediate, Never Saved	Status	BS	*B or *R

The buffer status command will report the status of the command buffer. The command buffer is 512 bytes long. If the command buffer is empty or less than 95% full, the controller will respond with ***R**.

A ***B** response will be issued if less than 5% of the command buffer is free.

*R **=** More than 5% of the buffer is free

*B **=** Less than 5% of the buffer is free

This command is commonly used when a long series of commands will be loaded remotely via RS-232C interface. If the buffer size is exceeded, the extra commands will not be received by the controller until more than 5% of the command buffer is free.

Example

Command	Response
1B	*B (less than 5% of the command buffer is free)

BS Buffer Size Status

Software Version: A

Syntax	Units	Range	Default Value
aBS	N/A	N/A	N/A
Attributes	**Command Type**	**See Also**	**Response**
Immediate, Never Saved	Status	B	*n

This command reports the number of bytes remaining in the command buffer. When entering long string commands, check the buffer status to be sure that there is enough room in the buffer. Otherwise, commands may be lost. Each character (including delimiters) uses one byte. The range for the response is 0 - 512 bytes.

Example

Command	Response
1BS	*122 (122 bytes available in the buffer)

C Continue

Software Version: A

Syntax	Units	Range	Default Value
<a>C	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Motion	PS, U	N/A

The Continue (**C**) command ends a pause state. It enables your indexer to continue executing buffered commands. After you enter a Pause (**PS**) or the Pause and Wait for Continue (**U**) command, you can clear it with a Continue (**C**) command. This command is useful when you want to transmit a string of commands to the buffer before you actually execute them.

Example

Command	Description
MC	Sets move to continuous mode
A1Ø	Sets acceleration to 10 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
PS	Pauses system until indexer receives C command
G	No action until C command received,
C	Continues executing commands in the buffer; accelerates the motor to 10 revs/sec

CG Correction Gain

Software Version: A

Syntax	Units	Range	Default Value
<a>CGn	N/A	n = 1 - 8	8

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	FSB, FSC, DB	*CGn

This command allows you to set the amount of error (steps) that should be corrected on the initial position maintenance (**FSC1** command) correction move. This takes place whenever the motor is stationary and outside the dead-band region (set with the **DB** command). This function is valid only in the Encoder Step mode (**FSB1**) and Position Maintenance (**FSC1**).

The percentage of error that the Position Maintenance function will attempt to correct on its correction moves is n/8 x 100%. If you set n to 1, the system will correct the error slowly (1/8 of the error is corrected on the first try). This type of correction is performed smoothly. If you set n to 8, the system will correct the error more quickly. However, there may be more overshoot and ringing at the end of the move.

Example

Command	Description
CG3	The system corrects 3/8 of the final-position error on the initial correction move
1CG	Reports Correction Gain (***CG3**)

CR	Carriage Return	Software Version: A

Syntax	Units	Range	Default Value
<a>CR	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	LF	*[cr]

The Carriage Return (**CR**) command determines when the indexer has reached a particular point in the execution buffer. When the indexer reaches this command in the buffer, it responds by issuing a carriage return (ASCII 13) over its interface back to the host computer or terminal. If you place the **CR** command after a Go (**G**) command, it indicates when a move is complete. If you place the **CR** command after a Trigger (**TR**) command, it indicates when the trigger condition is met.

You can use Carriage Return (**CR**) and Line Feed (**LF**) commands with the Quote (**"**) command to display multiple-line messages via the RS-232C interface.

Example

Command	Description
MN	Sets mode to preset mode
MRD4ØØØ	Sets motor resolution to 4000 steps/rev (check drive resolution)
A5Ø	Sets acceleration to 50 revs/sec^2
V5	Sets Velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 steps
G	Executes the move (Go)
1CR	Sends a carriage return after move is completed

The motor moves 4,000 steps. When the motor stops, the indexer sends a carriage return over its interface.

D Distance

Software Version: A

Syntax	Units	Range	Default Value
<a>D(s)n	steps	s=+/-2,147,483,648	25,000

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Motion	A, G, MN, MPA, MPI, V	*Dn

The Distance (**D**) command defines either the number of steps the motor will move or the absolute position it will seek after a Go (**G**) command is entered. In incremental mode (**MPI**), the value set with the Distance (**D**) command will be the distance (in steps) the motor will travel on all subsequent Go (**G**) commands.

In Absolute mode (**MPA**), the distance moved by the motor will be the difference between the current motor position and the position (referenced to the zero position) set with the **D** command. The **D** command has no effect on continuous moves (**MC**).

If **D** is entered with only a device address (**1D**), the indexer will respond with the current distance value. If a move is commanded without specifying a distance, the previously commanded distance will be applied to the move.

Entering DØ and G in Incremental preset mode will cause the W3 command to report back with *ØØØØØØØØ.

Example

Command	Description
MN	Sets indexer to Normal mode
MPI	Sets indexer to Incremental Position mode
A1Ø	Sets acceleration to 10 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D4ØØØ	Sets distance to 4000 steps
G	Executes the move

With the drive operating at 4000 steps/rev, the motor will travel 1 rev (CW) after **G** is issued.

DB Dead Band

Software Version: A

Syntax	Units	Range	Default Value
<a>DBn	n = steps	n = 0 - 999,999,999	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	FSG, CG	*DBn

This command is used in positioner maintenance and specifies a positioning range (in encoder steps) that the motor may not exceed after completing a move. If the motor's position is closer to the desired position than the number specified, no position maintenance correction will be performed. If the motor's position is not within the allowable range, position maintenance is performed (if enabled by the Enable Position Maintenance [**FSC1**] command).

The purpose of the **DB** command is to prevent the motor from searching for a set position when it is within an allowable dead band range.

Example

Command	Description
DB1ØØ	Sets Position Maintenance to activate if the motor's end-of-move position is off by more than 100 encoder steps.
1DB	Reports Deadband (*DB100)

DW Dead Band Window

Software Version: A

Syntax	Units	Range	Default Value
<a>DWn	n = steps	n = 0 - 999,999,999	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	FS commands	*DWn

This command allows systems with backlash to use stall detect (**FSH** command) features. If a non-zero dead band is selected, stall detection will not occur until the error exceeds the dead band width. This function is most effective when the encoder is mounted on the load.

Example

Command	Description
FSB1	Set indexer to Encoder Step mode
FSH1	Enable Stall Detect
DW1ØØ	Set Dead Band Window to 100 motor steps. 100 motor steps of Backlash are expected by the indexer. A stall will not be detected until the encoder lags the motor position by more then 100 motor steps.
1DW	Reports Deadband Window (*DW100)

E Enable Communications

Software Version: A

Syntax	Units	Range	Default Value
<a>E	N/A	N/A	Enabled

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Programming	F	N/A

The Enable Communications (**E**) command allows the indexer to accept commands over the serial communications interface. You can re-enable the communications interface with this command if you had previously disabled the RS-232C interface with the Disable Communications Interface (**F**) command. If several units are using the same communications interface, the **E** and **F** commands can help streamline programming.

Example

Command	Description
F	Disables all units (axes) on the communications interface
1E	Enables serial interface on Device 1
4E	Enables serial interface on Device 4
A1Ø	Set acceleration to 10 revs/sec^2
V5	Set velocity to 5 revs/sec
D4ØØØ	Sets distance to 4000 steps
G	Executes the move (Go—only axes 1 & 4 will move)

ER Encoder Resolution

Software Version: A

Syntax	Units	Range	Default Value
<a>ERn	steps/rev	n = 1 - 50,000	4000

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	FS, DW	*ERn

The encoder resolution defines the number of encoder steps the indexer will see per revolution of the motor. The number of lines on an encoder should be multiplied by 4 to arrive at the correct ER value per revolution of the motor. (In other words, one line of an encoder will produce 4 encoder steps.)

For accuracy and closed-loop stability, it is strongly recommended that the motor resolution (**MRD** command) is at least four times the encoder resolution.

Example

Command	Description
MRD4ØØØ	Sets indexer to control a motor of 4000 steps/rev
ER1ØØØ	Sets encoder resolution to 1000 encoder steps per 1 motor revolution
1ER	Reports Encoder Resolution(*ER1000)

F Disable Communications

Software Version: A

Syntax	Units	Range	Default Value
<a>F	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Programming	E	N/A

The Disable Communications (**F**) command is useful when you are programming multiple units on a single interface. Axes that are not intended to process global commands should be disabled using device specific **F** commands. This allows you to program other units without specifying a device identifier on every command. If you do not disable other units in a daisy chain, uploading programs may cause other units on the daisy chain to perform uploaded commands.

Example

Command	Description
1F	Disables the communications interface on unit 1
3F	Disables the communications interface on unit 3
G	All indexers (except 1 & 3) will execute a move (Go)

FS Encoder Functions Report

Software Version: A

Syntax	Units	Range	Default Value
aFS	n = N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	ER, PX	*nnnnnnnn

This command allows you to request the status of encoder functions set by **FS** commands. The response contains one ASCII digit per function set by the **FS** command, each of which is a zero or a one. The digits correspond to the functions, left to right, A through H. The digit 1 corresponds to a function that has been turned on, or enabled. The digit 0 corresponds to a function that has been turned off, or disabled.

A**—**Incremental = OFF (0); Absolute = ON (1)

Defines the move distances (D) as either incremental from current position, or as absolute (referenced to the absolute zero position).

B—Motor step mode = OFF (0); Encoder step mode = ON (1)

Defines the distance (D) parameter in units of motor steps or encoder steps

C—Position Maintenance: 0 = OFF, 1 = ON

Position maintenance, when enabled will cause the indexer to servo the motor to the desired position if not in the correct position at the end of a move, or if the motor is forced out of position while at rest.

D—Terminate move on Stall Detect: 0 = OFF, 1 = ON

Instructs the indexer to abort any move if a stall is detected.

E—Turn on Output 1 on Stall Detect: 0 = OFF,1 = ON

Instructs the indexer to set output 1 if a stall is detected.

F—Multiple axis stop: 0 = OFF, 1 = ON

Instructs the indexer to abort any move if a signal is received on the Trigger 3 input. If output on stall is enabled (FSE1), the indexer will also turn on Output 1 when a trigger is seen. Used when daisy chaining multiple axes together.

G—Turn on Output 2 when within dead band: 0 = OFF, 1 = ON

H—Enable Stall detect: 0 = OFF, 1 = ON.

Example	Command	Response
	1FS	*11ØØØØØØ (The indexer is in absolute encoder step mode with all other FS functions turned OFF.)

FSA Set Indexer to Incremental/Absolute Mode

Software Version: A

Syntax	Units	Range	Default Value
<a>FSAn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	MPI, MPA, PZ, PR, PX	N/A

This command sets the indexer to perform its moves in either absolute or incremental positioning mode. It is an alternative to the **MPI** and **MPA** commands.

FSAØ = Incremental mode
FSA1= Absolute mode

In Incremental mode (**FSAØ**), all moves are made with respect to the position at the beginning of the move. This mode is useful for repeating moves of the same distance.

In Absolute mode (**FSA1**), all moves are made with respect to the absolute zero position. The absolute zero position is set to zero when you power up the indexer or execute the Position Zero (**PZ**) command.

The Absolute mode is useful when you need to move to specific locations.

Example

Command	Description
FSA1	Sets Indexer to Absolute mode
PZ	Resets the absolute position counter to zero
A1Ø	Sets acceleration to 10 rev/sec^2
V5	Sets velocity to 5 rev/sec
D256ØØ	Move motor to absolute position 25,600
G	Executes the move (Go)
D64ØØØ	Move motor to absolute position 64,000
G	Executes the move (Go)

The motor moves 25,600 steps. Then the motor moves an additional 38,400 steps in the same direction to reach the absolute position of 64,000

FSB Set Indexer to Motor/Encoder Step Mode

Software Version: A

Syntax	Units	Range	Default Value
<a>FSBn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	ER, D, FSC	N/A

This command sets up the indexer to perform moves in either motor steps or encoder steps.

FSBØ = Motor step mode
FSB1 = Encoder step mode

In Motor Step mode, the distance command (**D**) defines moves in motor steps.

In Encoder Step mode, the distance command defines moves in encoder steps.

You must set up the indexer for the correct encoder resolution The Encoder Resolution (**ER**) command is used to define the number of encoder steps per revolution of the motor.

Example

Command	Description
MRD4ØØØ	Set the motor resolution to 4000 Steps/Rev
ER1ØØØ	Set up encoder where 1,000 encoder pulses (250 lines) are produced per 1 motor rev.
FSB1	Set moves to encoder step mode
A1Ø	Set acceleration to 10 rev/sec2
V5	Set velocity to 5 rev/sec
D4ØØØ	Set distance to 4,000 encoder steps
G	Executes the move (Go)

The motor will turn in the CW direction until 4,000 encoder pulses (equal to 4 motor revolutions) are received.

FSC Enable/Disable Position Maintenance

Software Version: A

Syntax	Units	Range	Default Value
<a>FSCn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	FSB, ER, DB	

FSC1 = Enable Position Maintenance
FSCØ = Disable Position Maintenance

Enabling position maintenance will cause the indexer to servo the motor until the correct encoder position is achieved. This occurs at the end of a move (if the final position is incorrect) or any time the indexer senses a change in position while the motor is at zero velocity. You must have an encoder connected, and set the indexer in Encoder Step mode in order to enable position maintenance.

Position maintenance will be disabled (turned OFF) automatically if a stall is detected while doing position maintenance.

Position maintenance may be turned off temporarily by issuing a **K** command. The next move will re-enable it. If using position maintenance, the user should also enable **FSD1** and **FSH1** to make certain motion stops if encoder feedback is lost. *NOTE: **FSC1** will work only if **FSB1** is enabled.*

Example

Command	Description
ER1ØØØ	Set encoder resolution to 1,000.
FSB1	Set encoder step mode.
FSC1	Enable position maintenance
FSH1	Enable stall detection
FSD1	Enable stop on stall

FSD Stop on Stall

Software Version: A

Syntax	Units	Range	Default Value
<a>FSDn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	DW, ER, FSH	N/A

Entering **FSDØ** will cause the indexer to attempt to finish the move when a stall is detected, even if the load is jammed.

Entering **FSD1** will cause the indexer to stop the move in progress when a stall is detected. The move is stopped immediately; there is no deceleration. This command is valid only if stall detection (**FSH1**) has been enabled.

Example

Command	Description
DW1ØØ	Set backlash value to 100 steps.
ER1ØØØ	Set encoder resolution to 1,000 steps/rev.
FSB1	Set indexer to encoder step mode
FSH1	Enable stall detect.
FSD1	Enable stop on stall.

FSE Turn on Output 1 on Stall

Software Version: A

Syntax	Units	Range	Default Value
<a>FSEn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	SS, DW, ER, FSH, FSF	N/A

FSEØ = Do not turn on output 1 on stall
FSE1 = Turn on output 1 on stall

Entering **FSE1** will cause the indexer to activate output number 1 when a stall is detected. This is useful for signalling other components in your system that a stall has occurred. This command will be valid only if Stall Detect (**FSH1**) and encoder step mode (**FSB1**) have been enabled.

Output number 1 is unaffected by a stall when **FSEØ** is entered.

This output will also activate during deceleration if Stop Motion on Trigger 3 (**FSF1**) is enabled. The default state of both outputs is low (0 volts).

Example

Command	Description
ER1ØØØ	Set encoder resolution to 1,000 steps/rev.
DW2ØØ	Set backlash dead band to 200 motor steps.
FSB1	Set indexer to encoder step mode
FSH1	Enable stall detect.
FSE1	Turn on output number 1 when a stall is detected.

FSF Stop Motion on Trigger 3 Software Version: A

Syntax	Units	Range	Default Value
<a>FSFn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	TR, FSE, SSH	N/A

FSFØ = Do not terminate move on Trigger 3
FSF1 = Terminate move when Trigger 3 is low.

Entering **FSF1** will cause any move in progress to be stopped whenever Trigger 3 is brought low. The move is stopped with a deceleration ramp set by the **A** command. The input may be used as a trigger, but only to stop motion. Entering **FSFØ** will turn this feature off.

The command buffer is cleared on stop. The SSH command can be used to continue to process commands.

Example

Command	Description
FSF1	Trigger 3 is now dedicated as a remote stop input.

FSG Turn on Output 2 when within Dead Band

Software Version: A

Syntax	Units	Range	Default Value
<a>FSGn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	DB, FSB, FSC, FSH	N/A

FSGØ— Do not turn on output 2 when the motor is within dead band.
FSG1—Turn on output 2 when within dead band.

The dead band is set using the **DB** command.

FSB1 and **FSC1** must be used for this command to function correctly. The output is updated by position maintenance.

Example

Command	Description
ER1ØØØ	Set encoder resolution to 1,000 steps/rev.
DB5Ø	Dead band is set to 50 steps.
FSB1	Set indexer to encoder step mode
FSC1	Enable Position Maintenance
FSG1	Enable post move position loss detection.

FSH Enable Stall Detect

Software Version: A

Syntax	Units	Range	Default Value
<a>FSHn	N/A	n = 0, 1	0

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	FS commands DW, ER	N/A

FSHØ = Disable Stall Detect
FSH1 = Enable Stall Detect

This command must be used to detect a stall condition. After enabling stall detection, stop on stall (**FSD1**) and output on stall (**FSE1**) can be used.

It is necessary to define the Dead band Window (**DW**) command and the Encoder Resolution (**ER**) command before this feature will operate properly. Stall Detection is only possible when an encoder is being used.

Stall Detect (**FSH1**) will function only if encoder step mode (**FSB1**) is enabled.

Example

Command	Description
DW5ØØ	Set dead band window to 500 steps
ER1ØØØ	Set encoder resolution to 1,000 steps (250 lines)
FSB1	Set indexer to encoder step mode
FSH1	Enable stall detection
FSD1	Stop motor movement if stall detected.

G Go

Software Version: A

Syntax	Units	Range	Default Value
<a>G	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Motion	A, D, MC, MN, S, V	N/A

The Go (**G**) command instructs the motor to make a move using motion parameters that you have previously entered. You do not have to re-enter Acceleration (**A**), Velocity (**V**), Distance (**D**), or the current mode (**MN** or **MC**) commands with each **G** (if you do not need to change them). In the Incremental Preset mode (**MPI**), a **G** will initiate the steps you specified with the **D** command.

A **G** command in the Absolute Preset mode (**MPA**) will not cause motion unless you enter a change in distance (**D**) first.

In Continuous mode (**MC**), you only need to enter the Acceleration (**A**) and Velocity (**V**) commands prior to **G**. The system ignores the Distance (**D**) command in this mode.
No motor motion will occur until you enter **G** in both the Normal (**MN**) and Continuous (**MC**) modes.

If motion does not occur with **G**, an activated end-of-travel limit switch may be on. Check the hard limit switches or use the limit disable command (**LD3**—see **RA** command also). The next buffered command will not be executed until after the move is completed.

Example

Command	Description
MN	Sets Normal mode (preset)
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D2ØØØ	Sets distance to 2,000 steps
G	Executes the move (Go)
A1	Sets acceleration to 1 rev/sec^2
G	Executes the move (Go)

Assuming the indexer is in Incremental Preset mode, the motor turns 2,000 steps and repeats the 2,000-step move using the new acceleration value of 1 rev/sec^2 (Total distance moved = 4,000 steps).

GH	Go Home	Software Version: A

Syntax	**Units**	**Range**	**Default Value**
<a>GHsn	Revs/sec	n = .01 - 50000 s = + or -	n = 0, s = +
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Savable in Sequence	Motion	OS Commands, RC, V	N/A

The Go Home (**GH**) command instructs the Indexer to search for an absolute position in the positive or negative (+ or -) direction. It defines home as the position where the home input signal changes states on the edge selected with the **OSH** command.

With the **OSB** command enabled, and if the selected edge for final home position is the first edge encountered, the motor will decelerate to 0 velocity, when that edge is detected. The motor will then reverse direction and stop on the selected edge. If the selected edge for the final home position is the second edge encountered the motor will travel until that edge is detected. The motor will then decelerate to 0 velocity. The controller will then position the motor 1/32 of a revolution on the outside of the selected edge. Finally, the motor will creep at 0.1 rps in the direction of the active home region, until home is detected. If the motor is already in the active home region and **GH** is issued, the motor will travel in the direction of the edge for the final home position. The motor will decelerate to 0 velocity, reverse direction, and approach home slowly until home is detected.

With the **OSB** command disabled, the motor will decelerate to 0 velocity after reaching the active home region, and will be considered to be at home if the home limit input is still active. If the deceleration overshoots the active home region, the motor will reverse direction and travel until home is reached. If the motor is already in the active home region and the **GH** is issued, no motion will occur.

The Indexer will reverse direction if an end-of-travel limit is activated while searching for home. However, if a second end-of-travel limit is encountered in the new direction, the Go Home procedure will stop and the operation will be aborted. The Status (**RC**) command will indicate if the homing operation was successful.

The Go Home command will use acceleration set by the **A** command. The Go Home velocity will not affect the standard velocity (**V**) value.

Completion of a successful Go Home move will set the absolute position counter to zero (see **PZ** command).

Example	Command	Description
	GH-2	The motor moves CCW at 2 revs/sec and looks for the Home Limit input to go active.

^H	Delete	Software Version: A

Syntax	Units	Range	Default Value
^H	N/A	N/A	None
Attributes	**Command Type**	**See Also**	**Response**
Immediate, Never Saved	Programming	-	N/A

This command allows you to delete the last character that you entered. The **^H** command will not prevent execution of an immediate command. A new character may be entered at that position to replace the existing character. (**^H** indicates that the Ctrl key is held down when the H key is pressed.) This command prompts the indexer to backup one character in the command buffer, regardless of what appears on the terminal. On some terminals, the Ctrl and the left arrow (<—) keys produce the same character. Pressing the delete key will not delete the previous character.

CAUTION
This command will not delete characters beyond the last delimiter issued.

H	Set Direction	Software Version: A

Syntax	Units	Range	Default Value
<a>H(s)	N/A	s = + or -	+
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Savable in Sequence	Programming	D	N/A

The Set Direction (**H**) command changes or defines the direction of the next move that the system will execute. This command does not affect moves already in progress.

H+ = Sets move to CW direction
H– = Sets move to CCW direction
H = Changes direction from the previous setting

In preset moves, a Distance (**D**) command entered after the **H** command overrides the direction set by the **H** command. In Continuous mode(**MC**), only the **H** command can set the direction of motion.

Example

Command	Description
MN	Sets Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 steps
G	Executes the move (Go) in CW direction
H	Reverses direction
G	Executes the move (Go) in CCW direction
MC	Sets mode to continuous
H+	Sets direction to CW
G	Moves continuously in CW direction

IS Input Status

Software Version: A

Syntax	Units	Range	Default Value
aIS	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Status	D	*nnnnnnnnnn

This command reports the status of all hardware inputs. The response is 10 ASCII digits (Ø or 1) plus a device address (1 - 8), corresponding to the following I/O bits:

1 Trigger bit 1 (Ø = Low, 1 = High)
2 Trigger bit 2 (Ø = Low, 1 = High)
3 Trigger bit 3 (Ø = Low, 1 = High)
4 Home enable (Ø = Low, 1 = High)
5 FLT (Ø = Faulted, 1 = Normal)
6 CCW limit (Ø = Low, 1 = High)
7 CW limit (Ø = Low, 1 = High)
8 Sequence Select 1 (Ø = Low, 1 = High)
9 Sequence Select 2 (Ø = Low, 1 = High)
10 Sequence Select 3 (Ø = Low, 1 = High)
11 Device Address (will return 1 ASCII digit, 1-8)

This is <u>not</u> a software status. It will report the actual hardware status of the inputs. **IS** can help you troubleshoot an application, to verify that limit switches, trigger inputs and home switches work.

Example

Command	Response
2IS	*ØØØ1ØØØØØØ2 (The input status of device 2 is reported: I/O bits 1-3 and 5-10 are low (grounded), and I/O bit 4, home enable, is high)

K	Kill	Software Version: A

Syntax	Units	Range	Default Value
<a>K	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Motion	S	N/A

The Kill (**K**) command is an emergency stop command and should only be used as such. This command causes indexing to cease immediately. There is *NO* deceleration of the motor. The Kill command may cause the motor to stall and lose torque with large loads at high speed. The load could be driven past limit switches and cause damage to the mechanism and possibly to the operation.

In addition to stopping the motor, the **K** command will terminate a loop, end a time delay, abort down-loading a sequence (**XD**), and clear the command buffer.

Example

Command	Description
A5	Sets acceleration to 5 revs/sec^2
V2	Sets velocity to 2 revs/sec
MC	Sets mode to continuous
G	Executes the move (Go)
K	Stops the motor instantly

L Loop Software Version: A

Syntax	Units	Range	Default Value
<a>Ln	number of loops	n = 0 - 65,535	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	C, N, U, Y	N/A

When you combine the Loop (**L**) command with the End-of-Loop (**N**) command, all of the commands between **L** and **N** will be repeated the number of times indicated by n. If you enter **L** without a value specified for n, or with a Ø, subsequent commands will be repeated continuously. If you specify a value greater than 65,535, the loop will be repeated continuously.

The **N** command prompts the indexer to proceed with further commands after the designated number of loops have been executed. The **Y** command stops loop execution after completing the current loop cycle. The Immediate Pause (**U**) command allows you to temporarily halt loop execution after completing the current loop cycle. You can use the Continue (**C**) command to resume loop execution. Nested loops are not supported.

Example

Command	Description
L5	Loop 5 times
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D1ØØØØ	Sets distance to 10,000 steps
G	Executes the move (Go)
N	Repeats 10,000-step move five times

LD	**Limit Disable**	Software Version: A

Syntax	**Units**	**Range**	**Default Value**
<a>LDn	See Below	n = 0 - 3	0
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Savable in Sequence	Set-Up	RA, TR, TS	N/A

The Limit Disable (**LD**) command allows you to enable/disable the end-of-travel limit switch protection. The **LDØ** condition does not allow the motor to turn without properly installing the limit inputs. If you want motion without wiring the limits, you must issue **LD3**.

- Enable CCW and CW limits—n = Ø (Default)
- Disable CW limit—n = 1
- Disable CCW limit—n = 2
- Disable CCW and CW limits—n = 3

WARNING

For your safety, we suggest that you wire the hardware limit switches to prevent the load from hitting the mechanical limits.

Example

Command	Description
1LDØ	Enables CW and CCW limits. The motor will move only if the limit inputs are bypassed or connected to normally-closed limit switches.
1LD3	Allows you to make any move, regardless of the limit input state.

LF Line Feed

Software Version: A

Syntax	Units	Range	Default Value
<a>LF	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	CR	[LF]

When you issue the Line Feed (**LF**) command, the system transmits a line feed character over the communications link. When the indexer reaches this command in the buffer, it responds by issuing a line feed (ASCII 10) over its interface back to the host computer. If you place the **LF** command after a Go (**G**) command, it indicates when a move is complete. If you place the **LF** command after a Trigger (**TR**) command, it indicates when the trigger condition is met.

You can use the Carriage Return (**CR**) and **LF** commands with the Quote (**“**) command to display multiple-line messages via the RS-232C interface.

Example

Command	Description
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D15ØØØ	Sets distance to 15,000 steps
G	Executes the move (Go)
1LF	Transmits a line feed character over the communications interface after the move is completed

MC Mode Continuous

Software Version: A

Syntax	Units	Range	Default Value
<a>MC	N/A	N/A	Inactive

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Motion	A, MN, T, TR, V	N/A

The Mode Continuous (**MC**) command causes subsequent moves to ignore any distance parameter and move continuously. You can clear the **MC** command with the Mode Normal (**MN**) command.

The indexer uses the previously defined Acceleration (**A**) and Velocity (**V**) commands to reach continuous velocity.

Using the Time Delay (**T**), Trigger (**TR**), and Velocity (**V**) commands, you can achieve basic velocity profiling.

Example

Command	Description
MC	Sets mode to continuous
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
G	Executes the move (Go)
T1Ø	Move at 5 revs/sec for 10 seconds
V7	Set velocity to 7 revs/sec
G	Change velocity to 7 revs/sec
T1Ø	Move at 7 revs/sec for 10 seconds
VØ	Set velocity to 0 rps (stop)
G	Executes the VØ command

The motor turns at 5 revs/sec for 10 seconds, then moves at 7 revs/sec for 10 seconds before decelerating to a stop.

MN	**Mode Normal**	Software Version: A

Syntax	**Units**	**Range**	**Default Value**
<a>MN	N/A	N/A	Active
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Savable in Sequence	Motion	A, D, G, MC, MPA, MPI	N/A

The Mode Normal (**MN**) command sets the positioning mode to preset. In Mode Normal, the motor will move the distance specified with the last distance (**D**) command. To define the complete move profile, you must define Acceleration (**A**), Velocity (**V**), and the Distance (**D**). The **MN** command is used to change the mode of operation from Mode Continuous (**MC**) back to normal or preset. To use the **MPA** or **MPI** command, you must be in Normal Mode (**MN**).

Example

Command	Description
MN	Set positioning mode to preset
A5	Set acceleration to 5 revs/sec^2
V5	Set velocity to 5 revs/sec
D1ØØØ	Set distance to 1,000 steps
G	Executes the move (Go)

Motor turns 1,000 steps CW after the **G** command is issued.

MPA Mode Position Absolute

Software Version: A

Syntax	Units	Range	Default Value
<a>MPA	N/A	N/A	Inactive

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-Up	D, MN, MPI, PZ	N/A

This command sets the positioning mode to absolute, and operates in the same way as **FSA1**. In this mode all move distances are referenced to absolute zero. In Mode Position Absolute (**MPA**), giving two consecutive Go (**G**) commands will cause the motor to move only once, since the motor will have achieved its desired absolute position at the end of the first move.

MPA is most useful in applications that require moves to specific locations while keeping track of the beginning position.

You can set the absolute counter to zero by cycling power or issuing a Position Zero (**PZ**) command. You must be in Normal mode (**MN**) to use this command. In continuous mode (**MC**), **MPA** is ignored.

Example

Command	Description
MN	Sets Normal mode (preset)
PZ	Resets absolute counter to zero
MPA	Sets position mode absolute
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D25ØØØ	Sets destination to absolute position 25,000
G	Motor will move to absolute position 25,000
D1ØØØØ	Sets destination to absolute position +10,000
G	Motor will move to absolute position +10,000

The motor will move 25,000 steps in the CW direction (if starting from position zero) and then move 15,000 steps in the CCW direction to reach the absolute position 10,000.

MPI Mode Position Incremental

Software Version: A

Syntax	Units	Range	Default Value
<a>MPI	N/A	N/A	Active

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-Up	D, MN, MPA	N/A

This command sets the positioning mode to incremental, and operates in the same way as **FSA0**. In incremental mode all move distances specified with the Distance (**D**) command will be referenced to the current position. Mode Position Incremental (**MPI**) is most useful in applications that require repetitive movements, such as feed to length applications.

You must be in normal mode (**MN**) to use this command. In continuous mode (**MC**), this command is ignored.

Example

Command	Description
MN	Set positioning mode normal (preset)
MPI	Set positioning mode incremental
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D1Ø,ØØØ	Sets distance of move to 10,000 steps
G	Move 10,000 steps CW
G	Move another 10,000 steps CW

The motor moves 10,000 steps CW after each G command (total move is 20,000 steps).

MRD Motor Resolution Definition Software Version: D

Syntax	Units	Range	Default Value
<a>MRDn	steps/rev	n = 400 - 4000	4000

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence, automatically saved	Set-Up	A, V, D	N/A

The Motor Resolution Definition command is used to match the indexer to the motor resolution. This command does not change the actual resolution of the motor, which is set by switches 4 and 5. The **MRD** command only accepts values that correspond to valid motor resolution options of 400, 1000, 2000 or 4000 steps/rev. If an invalid motor resolution value is entered, the **MRD** command will be ignored and the last valid motor resolution setting entered will be used.

Under certain circumstances MRD could be set to 25000. The motor will not run under indexer control, and you must set this value to be within the valid range.

Example

Command	Description
MN	Set positioning mode to preset
MRD4ØØØ	Motor reset to 4000 steps/rev
A5	Set acceleration to 5 revs/sec^2
V1Ø	Set velocity to 10 revs/sec
D8ØØØ	Set distance of move to 8000 steps
G	Executes the move (Go)

The indexer resolution and motor/drive resolution must match to get the commanded velocity and acceleration. This command does NOT affect distance. If **MRD** is executed with only a device address (**1MRD**), the indexer will respond with the current value (***MRDn**).

N End of Loop Software Version: A

Syntax	Units	Range	Default Value
<a>N	<a>N	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	C, L, PS, U	N/A

This command marks the end of a loop. You can use this command in conjunction with the Loop (**L**) command. All buffered commands that you enter between the **L** and **N** commands are executed as many times as the number that you enter following the **L** command.

Example

Command	Description
MN	Sets move to Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D1ØØØØ	Sets move distance to 10,000 steps
L5	Loops the following commands five times
G	Executes the move (Go)
N	Ends the loop

O Output Software Version: A

Syntax	Units	Range	Default Value
<a>Onn	on, off, or unchanged	Ø, 1 or X (See Below)	ØØ

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	OS, S, TR, TS	N/A

The Output (**O**) command turns the programmable output bits on and off. This is used for signalling remote controllers, turning on LEDs, or sounding whistles. The output can indicate that the motor is in position, about to begin its move, or is at constant velocity, etc.

n=1 = Turns output bits on
n=Ø = Turns output bits off
n=X = Leaves output bits unchanged

Both outputs must always be specified when using this command.

Example

Command	Description
MN	Set to Mode Normal
A1Ø	Set acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D2ØØØØ	Set move distance to 20,000 steps
OØ1	Set programmable output 1 off and output 2 on
G	Executes the move (Go)
OØØ	After the move ends, turn off both outputs

OS Report Homing Function Set-Ups

Software Version: A

Syntax	Units	Range	Default Value
<a>OS	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	OS(A-H)	*nnnnnnnn

This command results in a report of which software switches have been set by **OS** commands. The reply is eight digits. This command reports **OSA** through **OSH** Set-up status in binary format. The digit 1 represents ON (enabled), the digit Ø represents OFF (disabled). The default response is ***Ø1ØØØØØØ**.

OSA Define Active State of End-of-Travel Limits

Software Version: A

Syntax	Units	Range	Default Value
<a>OSAn	See Below	n = Ø, 1	Ø

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-Up	LD, OSC	N/A

OSAØ: Normally Closed Contacts
OSA1: Normally Open Contacts

This command sets the active state of the CW and CCW end-of-travel limit inputs. It enables you to use either normally closed or normally open switches.

Example

Command	Description
OSA1	Sets active state for normally open limit switches
OSCØ	Sets active state of home input closed (low)
OSH1	Selects the CCW side of the home signal as the edge on which the final approach will stop

OSB Back Up To Home

Software Version: A

Syntax	Units	Range	Default Value
<a>OSBn	See Below	n = Ø, 1	1

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-Up	GH, OSC, OSH	N/A

OSBØ: Back up to home
OSB1: Back up selected edge

With Back Up to Selected Home (**OSB**) command enabled, and if the selected edge for final home position is the first edge encountered the motor will decelerate to 0 velocity, when that edge is detected. The motor will then reverse direction and stop on the selected edge. If the selected edge for the final home position is the second edge encountered the motor will travel until that edge is detected. The motor will decelerate to a 0 velocity. The controller will then position the motor a short distance from the outside of the selected edge. Finally the motor will creep at 0.1 rps in the direction of the active home region, until home is detected. If the motor is already in the active home region and the Go Home (**GH**) command is given, the motor will travel in the direction of the edge for the final home position. The motor will decelerate to 0 velocity, reverse direction and approach home at the creep velocity until home is detected.

With **OSB** disabled, the motor will decelerate to 0 velocity after encountering the active home region, and will be considered to be at home if the home input is still active. If the deceleration overshoots the active home region the motor will reverse direction and travel back at creep speed until the home switch becomes active. If the system is already in the active home region and the Go Home (**GH**) command is given, no motion will occur.

Example

Command	Description
OSB1	Sets back up to home switch active
OSCØ	Sets active state of home input closed (low)
OSH1	Selects the CCW side of the home signal as the edge on which the final approach will stop

OSC Define Active State of Home Switch

Software Version: A

Syntax	Units	Range	Default Value
<a>OSCn	See Below	n = Ø, 1	Ø
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Savable in Sequence	Set-Up	GH, OSB, OSH	N/A

OSCØ: Active state of home input is closed
OSC1: Active state of home input is open
OSCØ requires that a normally open switch be connected to the home limit input.
OSC1 requires that a normally closed switch be connected to the home limit input.

Example

Command	Description
OSC1	Sets the active state of the home input to open

OSD Enable Encoder Z Channel for Home

Software Version: A

Syntax	Units	Range	Default Value
<a>OSDn	N/A	n = 0, 1	0
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Savable in Sequence	Set-up	OSB, OSC, OSH, GH	N/A

OSDØ = Do not reference Z Channel during homing
OSD1 = Reference Z Channel during homing

The encoder Z channel is used (in conjunction with a load activated switch connected to the home input) to determine the home position. The switch determines the home region, and the Z channel determines the exact home position inside the home region. For **OSD1** to be selected, **OSB1** must also be selected, and Encoder Step Mode (**FSB1**) must be enabled.

Example

Command	Description
OSD1	Recognizes Z channel as final home reference

OSH Reference Edge of Home Switch

Software Version: A

Syntax	Units	Range	Default Value
<a>OSHn	See Below	n = Ø, 1	Ø

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-Up	GH, OSB, OSC	N/A

OSHØ: Selects the CW side of the Home signal as the edge on which the final approach will stop

OSH1: Selects the CCW side of the home signal as the edge on which the final approach will stop

The CW edge of the Home switch is the first switch transition seen by the indexer when travelling from the CW limit in the CCW direction. If n = 1, the CCW edge of the Home switch will be referenced as the Home position. The CCW edge of the Home switch is the first switch transition seen by the indexer when travelling from the CCW limit in the CW direction.

Example

Command	Description
OSB1	Sets back up to home switch active
OSCØ	Sets active state of home input closed (low)
OSH1	Selects the CCW side of the home signal as the edge on which the final approach will stop

The home input becomes active when the home switch is closed. The indexer recognizes the CCW edge of the switch as the reference edge and backs up to that edge to complete the Go Home move.

PR Absolute Position Report

Software Version: A

Syntax	Units	Range	Default Value
aPR	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	D, MPA, MPI, MN, PZ	*±nnnnnnnnnn

PR reports motor position relative to the power-up position. The response is ±nnnnnnnnnn (range = 0 - ±2,147,483,648). In the encoder step mode (**FSB1**), the position will be reported in encoder steps. If you are in motor step mode (**FSBØ**), the position will be reported in motor steps. The response to this command will be reported after the move is complete. You can reset the position counter to zero by using the Position Zero (**PZ**) command.

Example

Command	Description
PZ	Resets the absolute counter to zero
LD3	Disable both CW & CCW limits
A1Ø	Set Acceleration to 10 revs/sec^2
V5	Set velocity to 5 revs/sec
D2ØØØ	Set move distance to 2000 steps
G	Executes the move (Go)
1PR	Request absolute position report. Response should be ***+ØØØØØØ2ØØØ**

PS Pause

Software Version: A

Syntax	Units	Range	Default Value
<a>PS	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	C	N/A

This command pauses execution of a command string or sequence until the indexer receives a Continue (**C**) command. **PS** lets you enter a complete command string before running other commands. **PS** is also useful for interactive tests and synchronizing multiple indexers that have long command strings.

Example

Command	Description
PS	Pauses execution of commands until the indexer receives the Continue (C) command
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D5ØØØ	Sets move distance to 5000 steps
G	Executes the move (Go)
T2	Delays the move for 2 seconds
G	Executes the move (Go)
C	Continues Execution

When the indexer receives the C command, the motor moves 5,000 steps twice with a 2-second delay between moves.

PX Report Absolute Encoder Position

Software Version: A

Syntax	Units	Range	Default Value
aPX	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	W3, PR, FSB	*nnnnnnn

This command returns a decimal value indicating the absolute position of the incremental encoder. The absolute position is based on the zero position. The zero position is established when you power up the system. The zero position can also be established after the indexer performs a Position Zero (**PZ**) command. Whether in Motor Step mode or Encoder Step mode, the position is reported in encoder steps. The range of the response is 0 - ±9,999,999,999. This command is useful in the following situations:

- Encoder Set-up
- End of move (verification of position)

Example

Command	Description
MN	Presets mode
PZ	Sets the absolute counter to zero
A1Ø	Sets acceleration to 10 rev/sec2
V5	Sets velocity to 5 rev/sec
D8ØØØ	Sets move distance to 8000 steps
G	Executes the move (Go)
FSB1	Sets indexer to encoder step mode
1PX	After the motor executes the move, the encoder position is reported: The response is *+0000002000, assuming the ER command is set to 1000 and the drive is set to 4000 steps/rev.

PZ Set Absolute Counter to Zero

Software Version: A

Syntax	Units	Range	Default Value
<a>PZ	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Never Saved	Programming	D, MN, PR, PX	N/A

This command sets the absolute position counter to zero. Absolute counter will also be set to zero when you cycle power to the unit or when you successfully execute a homing (**GH**) function.

Example

Command	Description
MPA	Makes preset moves from absolute zero position
PZ	Sets absolute position counter to zero
A1Ø	Sets acceleration to 10 rev/sec^2
V5	Sets velocity to 5 rev/sec
D25ØØ	Sets move distance to 2500 steps
G	Executes the move (Go)
1PR	Reports absolute position (*+0000002500)
PZ	Sets the absolute counter to zero
1PR	Reports absolute position (*+0000000000)

“	Quote	Software Version: A

Syntax	Units	Range	Default Value
“x	N/A	x = up to 17 ASCII	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	CR, LF	x

Up to 17 characters entered after the quotation marks (**“**) will be transmitted, exactly as they are entered, over the RS-232C link. A space entered by the space bar indicates the end of the command. A space is always sent after the last character in the string. This command is used during buffered moves or sequences to command other devices to move, or to send the message to a remote display.

Example

Command	Description
PS	Pause execution until Continue (C) is entered
MN	Set to mode normal (Preset Moves)
A5	Set acceleration to 5 revs/sec^2
V5	Set velocity to 5 revs/sec
D2ØØØ	Set distance to 2,000 revs
G	Executes the move (Go)
“MOVE_DONE	After the move, the Indexer will send the message MOVE_DONE via the RS-232C port
CR	
LF	

Q Enter/Exit Velocity Profiling Mode

Software Version: A

Syntax	Units	Range	Default Value
<a>Q1	N/A	N = Ø,1	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Set-Up	RM	N/A

Q1 activates Velocity Profiling mode. Subsequent **RM** commands will immediately change motor velocity.

The **QØ** command exits the Velocity Profiling mode. The motor will stop when **QØ** is issued. Entering this command will cause the Indexer to enter Normal mode (**MN**).

Example

Command	Description
Q1	Enter Velocity Streaming mode
RMØØAØ	Change speed to 0.25 revs/sec
RMØ14Ø	Change speed to 0.5 revs/sec
RMØ28Ø	Change speed to 1 revs/sec
RMØ5ØØ	Change speed to 2 revs/sec
RMØ28Ø	Change speed to 1 revs/sec
RMØ14Ø	Change speed to 0.5 revs/sec
RMØØAØ	Change speed to 0.25 revs/sec
RMØØØØ	Change speed to 0 revs/sec
QØ	Exit Velocity Streaming mode

R Request Indexer Status

Software Version: A

Syntax	Units	Range	Default Value
aR	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Status	RA, RB, RC, XSR, XSS	*x

The Request Indexer Status (**R**) command can be used to indicate the general status of the indexer. Possible responses are:

Character	Definition
***R**	Ready
***S**	Ready, Attention Needed
***B**	Busy
***C**	Busy, Attention Needed

When the indexer is not prepared to accept another command, the following conditions will cause an indexer is busy (***B**) response:

* Performing a preset move
* Accelerating/decelerating during a continuous move
* A time delay is in progress. (**T** command)
* In RM mode
* Paused
* Waiting on a Trigger
* Going Home
* In Power-on sequence mode
* Running a sequence
* Executing a loop

The following conditions will cause an error (***S** or ***C**) response:

* Go home failed
* Limit has been encountered
* Sequence execution was unsuccessful
* Sequence memory checksum error

When the response indicates that attention is required, the **RA**, **RB**, **RC**, **XSR**, or **XSS** commands can provide details about the error.

It is not recommended that this command be used in tight polling loops that could result in microprocessor overload. Time delays can alleviate this problem.

This command is not intended to be used to determine if a move is complete. It should be used after a move is complete to determine if errors or faults exist. Use a buffered status request (**CR** or **LF**) command or a programmable output to indicate move completion.

Example	<u>Command</u>	<u>Response</u>
	R	*R (Indexer ready to accept a command, and no error conditions exist.)

RA Limit Switch Status Report

Software Version: A

Syntax	Units	Range	Default Value
aRA	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Status	R, RB	*x

The **RA** command responds with the status of the end-of-travel limits during the last move as well as the present condition. This is done by responding with one of 12 characters representing the conditions listed below.

Response Character	***Last Move Terminated By CW Limit—CCW Limit***		***Current Limit Status CW Limit—CCW Limit***	
*@	No	No	Off	Off
*A	Yes	No	Off	Off
*B	No	Yes	Off	Off
*D	No	No	On	Off
*E	Yes	No	On	Off
*F	No	Yes	On	Off
*H	No	No	Off	On
*I	Yes	No	Off	On
*J	No	Yes	Off	On
*L	No	No	On	On
*M	Yes	No	On	On
*N	No	Yes	On	On

The **RA** command is useful when the motor will not move in either or both directions. The report back will indicate if the last move was terminated by one or both end-of-travel limits. This command is not intended to be used to determine if a move is complete. It should be used after a move to determine if errors or faults exist. If you are hitting a limit switch, the Indexer Status (**R**) will return a ***S**.

Example	<u>Command</u>	<u>Response</u>
	1RA	*@ (By issuing a 1RA command, the indexer with the address of 1 responded with *@, indicating that the last move was not terminated by a limit and that no limits are currently active.)

RB Loop, Pause, Shutdown, Trigger Status Request

Software Version: A

Syntax	Units	Range	Default Value
aRB	N/A	N/A	N/A
Attributes	**Command Type**	**See Also**	**Response**
Immediate, Never Saved	Status	L, PS, R, RA, ST, TR	*x

This command receives a response from ***@** to ***O**, as defined below. The four conditions for which status is indicated are as follows:
Loop Active: A loop is in progress.
Pause Active: Buffered commands waiting for a Continue (**C**).
Shutdown Active: The motor is shutdown by the **ST1** command.
Trigger Active: At least one trigger is active.

Response Character	***Loop Active***	***Pause Active***	***Shutdown Active***	***Trigger Active***
*@	No	No	No	No
*A	Yes	No	No	No
*B	No	Yes	No	No
*C	Yes	Yes	No	No
*D	No	No	Yes	No
*E	Yes	No	Yes	No
*F	No	Yes	Yes	No
*G	Yes	Yes	Yes	No
*H	No	No	No	Yes
*I	Yes	No	No	Yes
*J	No	Yes	No	Yes
*K	Yes	Yes	No	Yes
*L	No	No	Yes	Yes
*M	Yes	No	Yes	Yes
*N	No	Yes	Yes	Yes
*O	Yes	Yes	Yes	Yes

This command is not intended to be used to determine if a move is complete. It should be used after the move is complete to determine if errors or faults exist.

Example	Command	Response
	1RB	*A (After issuing a 1RB command, the response came back as *A. This means that the indexer is currently executing a loop.)

RC Closed Loop Status

Software Version: A

Syntax	Units	Range	Default Value
aRC	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	R, RA, RB, FS, GH	*x

The **RC** command has the same response format of **RA** and **RB**. The four conditions for which status is indicated are:

Static Position Loss:

In this condition, the indexer has detected motion of the load while the motor was stopped. The indexer was not able to correct the position, resulting in Position Maintenance being disabled.

Post Move Position Loss:

In this condition, the indexer has detected a deviation between actual and desired position at the end of a move which exceeds the backlash/dead band parameter. This may involve a Stall, or slipping of the load short of a stall.

Homing Function Failure:

In this condition, the indexer has encountered both End-of-Travel limits or one of several possible Stop commands or conditions. Go Home motion was concluded, but not at Home.

Stall:

In this condition, the indexer has detected a deviation between motor and encoder position larger than one pole of the motor while running, or a deviation larger than that plus the backlash parameter following a direction change.

NOTE: This command is not intended to be used to determine if a move is complete. Rather, it should be used after the move is complete to determine if there might be other errors or faults.

Response Character	Stall Detected?	Go Home Successful?
*@	NO	YES
*A	YES	YES
*B	NO	NO
*C	YES	NO

The indexer will respond with either the @ or A character if the last move was not a 'Go Home' move.

Example

Command	Description
1RC	*A (This means that while attempting the last move, the indexer detected a stall.)

Stalls, Stop, FSF, SSD1 and TRIG3 move terminate are treated as stops.

RM Rate Multiplier in Velocity Streaming

Software Version: A

Syntax	Units	Range	Default Value
<a>RMn	revs/sec	n = ØØØ - FFF	None

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Motion	D, H, Q	N/A

The **RM** command sets an immediate velocity where n represents a 4-digit hexadecimal value. The value for n is determined with the following formula:
(revs/sec) x (resolution constant) = **decimal no. for velocity value to be rounded off to the closest whole number.**

In the formula, revs/sec is the desired speed. The *resolution constant* is the value taken from the following table.

Motor Resolution	***Resolution Constant***
400	655.3400
1000	652.7801
2000	648.5135
4000	648.5333

The resolution that the **MRD** command defines determines which resolution constant will be used in the formula. The resulting decimal number must be converted to a hexadecimal number to obtain the value for n.

The velocity change is instant - there is no acceleration/deceleration ramp between velocities. A limit switch closure will stop movement in Velocity Profiling mode, but does not cause the Indexer to exit this mode. To recover from a limit stop in **RM** mode, **QØ** must be issued and the direction must be changed. Velocity Profiling mode is uni-directional. The last direction set either from a move or from a Distance (**D**) or Direction (**H**) command will be used. Bi-directional moves can be made in this mode by returning to velocity zero (**Ø**), turning **RM** mode off, changing the direction, and re-enabling **RM** mode. Exiting **RM** mode with **QØ** causes the Indexer to enter Normal mode (**MN**).

Example

Command	Response
Q1	Enter Velocity Streaming mode
RMØØAØ	Run at 0.25 revs/sec^2
RMØ14Ø	Change speed to 0.5 revs/sec^2
RMØ28Ø	Change speed to 1 revs/sec^2
RMØ5ØØ	Change speed to 2 revs/sec^2
RMØ28Ø	Change speed to 1 revs/sec^2
RMØ14Ø	Change speed to 0.5 revs/sec^2
RMØØAØ	Change speed to 0.25 revs/sec^2
RMØØØØ	Change speed to 0 revs/sec^2
QØ	Exit Velocity Streaming mode

RV Revision Level

Software Version: A

Syntax	Units	Range	Default Value
aRV	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	D, H, Q	*nn-nnnn-nn<xn>

The Revision (**RV**) command responds with the software part number and its revision level. The response is in the form shown below:

92-nnnn-nn**<xn***>[cr]
(**part number**, ***revision level***)

The part number identifies which product the software is written for, as well as any special features that the software may include. The revision level identifies when the software was written. You may want to record this information in your own records for future use. This type of information is useful when you consult Digiplan's Applications Department.

Example

Command	Response
1RV	*92-Ø13431-Ø1A

The product is identified by *92-Ø13431-Ø1A, and the revision level is identified by A.

S	Stop	Software Version: A

Syntax	Units	Range	Default Value
<a>S	N/A	N/A	N/A
Attributes	**Command Type**	**See Also**	**Response**
Immediate, Never Saved	Motion	A, K, Q, SSH	N/A

This command decelerates the motor to a stop using the last defined Acceleration (**A**) command. This command clears the command buffer (at the end of a move, if one is in progress). The Sequence Definition (**XD**) command is aborted and a time delay is terminated. If **SSH1** is set the indexer will stop the current move but it will not clear the command buffer.

The Stop (**S**) command does not stop the motor in Velocity Streaming or Rate Multiplier (**RM**) mode. If you are in the **RM** mode, issue an Exit Velocity Profiling Mode (**QØ**) command to stop the motor.

Example

Command	Description
MC	Sets move in continuous mode
A1	Sets acceleration to 1 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
G	Executes the move (Go)
A5	Sets Acceleration to 5 revs/sec^2
S	Stops motor (motor comes to a stop at a deceleration rate of 5 revs/sec^2)

The S command is not buffered. As soon as the indexer receives the S command, it stops motion.

SN Scan

Software Version: A

Syntax	Units	Range	Default Value
<a>SNn	n = mS	1 - 1000	50

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-Up	XP	*SNn

The Scan (**SN**) command allows you to define the *debounce time* (in milliseconds) for external sequence selection inputs. This is the time during which the sequence inputs must remain constant before the new input pattern is accepted. If you are using a PLC you should set the debounce time to be less than the *on time* of the PLC outputs.

This command allows you to select the best possible trade-off between noise immunity and speed for a given application. If you make your scan time too short, the Indexer may respond to an electrical glitch. If you issue the Scan command with only a device address (**1SN**), the indexer will respond with the current debounce time (***SNn**).

Example

Command	Description
SN1Ø	Sets scan time of sequence select inputs to 10 ms

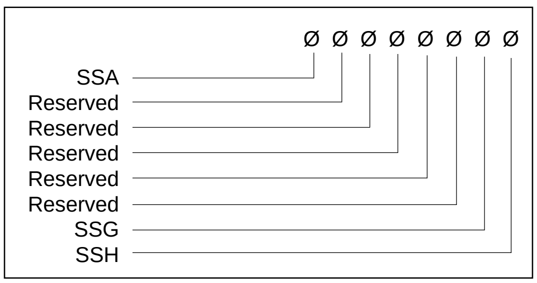
SS — Software Switch Function Status

Software Version: A

Syntax	Units	Range	Default Value
aSS	N/A	N/A	see below

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	SSA, SSG, SSH	*nnnnnnnn

This command reports the status of the **SS** commands. From left to right, the 8-character response corresponds to **SSA** through **SSH**.



Default Values:

- **SSA** = Ø
- **SSG** = Ø
- **SSH** = Ø

SSA	RS-232C Echo Control	Software Version: A

Syntax	Units	Range	Default Value
<a>SSAn	N/A	n = Ø, 1	Ø
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Automatically Saved	Set-Up	E, F	N/A

This command turns the RS-232C echo (transmission of characters received from the remote device by the Indexer) on and off.
SSAØ = Echo on
SSA1 = Echo off

In the Echo On (**SSAØ**) mode, characters that are received by the indexer are echoed automatically. In the Echo Off (**SSA1**) mode, characters are not echoed from the Indexer. This command is useful if your computer cannot handle echoes. In a daisy chain, you must have the echo on (**SSAØ**) to allow indexers further down the chain to receive commands. *Status commands do not echo the command sent, but transmit the requested status report.*

Example

Command	Description
SSA1	Turns echo off (Characters sent to the indexer are not echoed back to the host.)

SSG Clear/Save the Command Buffer on Limit

Software Version: A

Syntax	Units	Range	Default Value
<a>SSGn	N/A	n = Ø, 1	Ø
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Automatically Saved	Set-Up	LD	N/A

SSGØ = Clears command buffer on limit
SSG1 = Saves command buffer on limit

In most cases, it is desirable that upon activating an end-of-travel limit input all motion should cease until the problem causing the over-travel is rectified. This will be assured if all commands pending execution in the command buffer are cleared when hitting a limit. This is the case if **SSGØ** is specified. If **SSG1** is specified and a limit is activated, the current move is aborted, but the remaining commands in the buffer continue to be executed.

Example

Command	Description
SSG1	Saves buffer on limit
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 steps
G	Executes the move (Go)
O11	Turn on outputs 1 and 2

If a limit switch is encountered while executing the move, outputs 1 and 2 will still go on.

SSH Clear/Save Command Buffer on Stop

Software Version: A

Syntax	Units	Range	Default Value
<a>SSHn	N/A	n = Ø, 1	Ø

Attributes	Command Type	See Also	Response
Buffered, Automatically Saved	Set-Up	S	N/A

SSHØ = Clears command buffer on stop
SSH1 = Saves command buffer on stop

In Normal Operation (**SSHØ**) the Stop (**S**) command or a dedicated stop input will cause any commands in the command buffer to be cleared. If you select the Save Command Buffer On Stop (**SSH1**) command, a Stop (**S**) command will only stop execution of a move in progress. It will not stop execution of any commands that remain in the buffer.

Example

Command	Description
SSHØ	Clears buffer on stop
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 steps
L5Ø	Loops 50 times
G	Executes the move (Go)
T.5	Pauses the motor 500 ms
N	Ends Loop
S	Stops motion

When S is issued, the indexer will clear the buffer and stop the move.

ST Shutdown

Software Version: A

Syntax	Units	Range	Default Value
<a>STn	N/A	n = Ø, 1	Ø

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	-	N/A

The Shutdown (**ST1**) command rapidly decreases motor current to zero. The system ignores move commands that are issued after **ST1,** and **ST1** reduces motor heating, and allows you to manually position the load. Torque on the motor is not maintained after you issue **ST1**.
The **STØ** command rapidly increases motor current to normal. Once you restore the current, you can execute moves. The motor position counter is set to the current position when you enter an **STØ** command. If you re-enable the drive using **STØ**, you must wait 500 ms before using other commands.

Example

Command	Description
ST1	Shuts off current to the motor

SV Servoing Parameter

Software Version: A

Syntax	Units	Range	Default Value
<a>SVn	N/A	n = 0 - 3	None

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-up	FSC, ST	N/A

The Servoing Parameter (SV) command provides four different ways of simultaneously changing state of the motor shutdown and position maintenance functions. The four commands are as follows:

SVØ This command causes the position maintenance function to be turned off, but does not turn off motor power. It is identical in function to the FSCØ command.

SV1 This command causes the position maintenance function to be turned off and the motor to be shut down simultaneously.

SV2 This command causes the position maintenance function to be turned on and turns the motor power back on if it was turned off due to SV1 or SVØ command. The encoder position will be read and this newest position will be maintained.

SV3 This command causes the position maintenance function to be turned on and turns the motor power on if it was turned off. The indexer will servo back to the rest position held before the position maintenance function was disabled.

Example

Command	Description
SV1	Simultaneously turns off Position Maintenance function and shuts down the motor.

T	Time Delay	Software Version: A

Syntax	Units	Range	Default Value
<a>Tn	seconds	n = 0.01 - 99999.99	None
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Savable in Sequence	Programming	FSF, SSH	N/A

The Time (**T**) command causes the indexer to wait the number of seconds that you specify before it executes the next command in the buffer. This command is useful whenever you need to delay the motor's actions or when you wish to move the motor in continuous velocity for preset time.

Time delays are not processed when 'Save Command Buffer on Stop' (SSH) is active and the stop input is activated (FSF).

Example

Command	Description
MN	Sets Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 steps
T1Ø	Pauses motor movement 10 seconds
G	Executes the move (Go)
T5	Pauses the motor for 5 seconds after the move
G	Executes the move (Go)

TR Wait For Trigger

Software Version: A

Syntax	Units	Range	Default Value
<a>TRnnn	See Below	n = Ø, 1, or X	None

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	TS	N/A

This command allows you to specify a trigger configuration to be matched before continuing execution of the move, where *nnn* corresponds to triggers 1, 2, and 3 respectively. The possible values for *n* are as follows:

n = 1 Wait for the trigger input to be high (opened)
n = Ø Wait for the trigger input to be low (grounded)
n = X Ignore the trigger input

All three inputs must always be specified when using this command.

Trigger commands are not processed when ‘Save Command Buffer on Stop’ (SSH) is active and the stop input is activated (FSF).

Example

Command	Description
TR1ØX	Wait for input 1 to be opened and input 2 to be grounded before going on to the next command— input 3 will be ignored
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 steps
G	Executes the move (Go)

TS — Trigger Input Status — Software Version: A

Syntax	Units	Range	Default Value
aTS	See Below	n = 0, or 1	None

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Status	TR	*nnn

This command retrieves the state of the trigger inputs. The response is in the form nnn, where nnn reports the status of triggers 1, 2, and 3 respectively. The possible values for n are as follows:

n = 1: Input is high (opened)
n = Ø: Input is low (closed)

TS checks the status of the trigger inputs when it appears the execution is being halted by a **TR** command. To make sure that your trigger pattern is met, you can check with the **TS** command.

Example

Command	Response
1TS	*1Ø1

Trigger bits 1 and 3 are high (opened) and trigger bit 2 is low (closed).

U — Pause and Wait for Continue — Software Version: A

Syntax	Units	Range	Default Value
<a>U	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Programming	C, PS	N/A

This command causes the indexer to complete the move in progress, then wait until it receives a Continue (**C**) to resume processing. Since the buffer is saved, the indexer continues to execute the program (at the point where it was interrupted). The indexer continues processing when it receives the **C** command. This command is typically used to stop a machine while it is unattended.

Example

Command	Description
MN	Sets move to Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
LØ	Loops indefinitely
D256ØØ	Sets distance to 25,600 steps

G	Executes the move (G)
T1Ø	Waits 10 seconds after the move
N	Ends loop
U	Halts execution until the indexer receives the Continue command

This command string pauses when the **U** command is entered. A **C** command resumes execution where it was paused. In this example, the loop stops at the end of a move, and resumes when the indexer receives the **C** command. In reaction to the **T1Ø** command in the loop, there may be a 10-second delay before motion resumes after the **C** is executed, depending on when the **U** command is completed.

V Velocity

Software Version: A

Syntax	Units	Range	Default Value
<a>Vn	revs/sec	n = 0.01 - 50.00	1

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Motion	A, D, G, GH, MRD	*Vn

The **V** command defines the maximum speed at which the motor will run when given the Go (**G**) command.

Once you define the velocity, that value will be valid until you define another velocity, cycle the power, or issue a **Z** (Reset) command.

*If the value specified for the **V** command is not valid, the Indexer ignores that value and defaults to the value specified in the last **V** command. If **V** is issued with only a device address (**1V**), the indexer will respond with the current velocity value (***Vn**).*

Example

Command	Description
MC	Sets move to continuous
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
G	Go (Begin motion)

In preset Mode Normal (MN), the maximum velocity may be limited when the resulting move profile is triangular. In Mode Continuous (MC), the indexer moves to the next command in the buffer—once the specified velocity is reached.

W1 Signed Binary Position Report

Software Version: A

Syntax	Units	Range	Default Value
aW1	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Status	PR, W3	*xxxx

Report back gives immediate binary representation of position relative to start of the current move. The format of the response is a four digit ASCII number (xxxx) that should be interpreted as a 32-bit binary number. The computer interprets the number to give a numerical position in steps. The format is in 2’s complement notation. Moves in the negative direction (CCW) will report back negative numbers (bit 31 is set to 1).

If you use a terminal to communicate with the indexer, the response may not be a printable character. The response must be decoded with a computer. *This command is useful if you want to receive a position report while the motor is moving*.

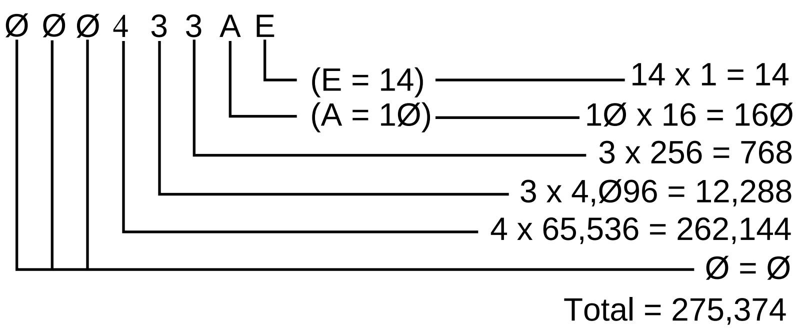
If this command is transmitted in quick succession, it can lead to the processor not being able to respond fast enough. It is recommended that for a single axis, at least 200ms is allowed between command transmission and reception of all data characters. Also at least 500ms should be allowed before the command is repeated. These delay times must be increased if more than 1 axis is being interrogated.

W3 Hexadecimal Position Report

Software Version: A

Syntax	**Units**	**Range**	**Default Value**
aW3	N/A	N/A	N/A
Attributes	**Command Type**	**See Also**	**Response**
Immediate, Never Saved	Status	PR, W1	*xxxxxxxx

This command responds with an immediate hexadecimal character position report back in 2's complement format. The position response indicates the motor position relative to the current move. The format of the response is an asterisk, followed by an 8-digit ASCII hexadecimal number. Assume the response was *ØØØ433AE. The decimal value would be:



If the first digit of the response is **F**, the response represents a "2's complement" negative number. Use the following steps to interpret a negative number (starting with **F**).

The Binary Approach

1. Convert the hexadecimal response to binary form.
2. Complement the binary number.
3. Add 1 to the binary result.
4. Convert the binary result to decimal value with a minus sign placed ahead of the decimal value.

The Computer Approach

Subtract the hexadecimal number from 16^8 (2^{32}) (4,294,967,296).

The Easy Way

1. Delete all the leading **F**s, and convert to decimal.
2. Convert and subtract the next largest power of 16.

*If the indexer response to **W3** is ***FFFF9E58**:*

1.	Leave off the Fs:	9E58 hex	=	40,536
2.	Subtract from 16^4	1ØØØØ hex	=	65,536
		Result	***=***	***-25,000***

If this command is transmitted in quick succession, it can lead to the processor not being able to respond fast enough. It is recommended that for a single axis, at least 200ms is allowed between command transmission and reception of all data characters. Also at least 500ms should be allowed before the command is repeated. These delay times must be increased if more than 1 axis is being interrogated.

Example	Command	Response
	1W3	*FFFFA19C (24,163 steps from start of move)

XC Sequence Checksum

Software Version: A

Syntax	Units	Range	Default Value
aXC	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Never Saved	Status	XD, XE	N/A

XC computes the BBRAM checksum. After the unit is programmed, the response can be used for system error checking. The three-decimal response (ØØØ - 255) is followed by a [cr]. The response does not indicate the number of bytes programmed. This response is designed to be used for comparison. As long as the Indexer is not re-programmed, the checksum response should always be the same.

Example	Command	Response
	1XC	*149

XD Sequence Definition

Software Version: A

Syntax	Units	Range	Default Value
<a>XDn	Sequence No.	n = 1 - 7	N/A

Attributes	Command Type	See Also	Response
Buffered, Never Saved	Programming	XE, XR, XRP, XT	N/A

This command begins sequence definition. All commands between the **XD** command and Sequence Termination (**XT**) command are defined as a sequence. The sequences will automatically be defined when **XT** is issued. If a sequence you are trying to define already exists, you must erase that sequence before defining it using the Erase Sequence (**XE**) command. A sequence cannot be longer than 256 characters. Immediate commands cannot be entered into a sequence.

Example

Command	Description
XE1	Erases sequence 1
XD1	Defines sequence 1
MN	Sets to Normal mode
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets acceleration to 5 revs/sec
D1ØØØØ	Sets distance to 10,000 steps
G	Executes the move (Go)
XT	Ends definItion of Sequence 1
XR1	Executes Sequence 1

XE Sequence Erase

Software Version: A

Syntax	Units	Range	Default Value
<a>XEn	Sequences	n = 1 - 7	N/A
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Never Saved	Programming	XD, XR, XRP, XT	N/A

This command allows you to delete a sequence. The sequence that you specify (n) will be deleted when you issue the command. *You should always delete a sequence before defining it.*

Example

Command	Description
XE1	Deletes Sequence 1
XD1	Defines Sequence 1

XP Set Power-up Sequence Mode

Software Version: A

Syntax	Units	Range	Default Value
<a>XPn	Sequence No.	n = 0 - 9	0
Attributes	**Command Type**	**See Also**	**Response**
Buffered, Automatically Saved	Set-Up	XQ, XSP, XSR	N/A

This command executes a single sequence or multiple sequences on power-up. If *n* = 1-7, the sequence whose number corresponds to n will be executed on power up. Control will then be passed to the RS-232C interface.

If *n* = 8, the sequence whose number appears on the sequence select inputs (**SEQ1 - SEQ3**) will be executed on power-up. Control will then be passed to the RS-232C interface.

If *n* = 9, the sequence whose number appears on the Sequence Select inputs (**SEQ1 - SEQ3**) will be executed on power-up. When the first sequence is finished in **XP9** mode, the Indexer will scan the Sequence Select inputs again and execute the next sequence. This cycle will continue until a Stop (**S**) or Kill (**K**) command is issued, a limit is encountered, or the unit is powered down.

The possible settings for this command are as follows:

n = Ø: No sequence is executed on power-up
n =1-7: Sequence 1 - 7 is executed on power-up

n = 8: Sequence select inputs are read (single run) on power-up
n = 9: Sequence select inputs are read (continuous run) on power-up

In **XP9** mode, you can use the **XQ** command to stop the Indexer from selecting the next sequence until all the sequence select inputs are first opened. Note that **XQ1** must be included in every sequence controlled in this way.

Example

Command	Description
XE1	Erases Sequence 1
XD1	Defines Sequence 1
LD3	Disables CW & CCW limits
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets velocity to 5 revs/sec
D25ØØØ	Sets distance to 25,000 steps
G	Executes the move (Go)
XT	Ends definition of Sequence 1
XP1	Executes Sequence 1 on power-up
Z	Resets the indexer

The motor moves 25,000 steps during power-up or reset (Z).

XQ Sequence Interrupted Run Mode

Software Version: A

Syntax	Units	Range	Default Value
<a>XQn	Sequence No.	n = Ø, 1	Ø

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Set-Up	XP	N/A

XQ1: Interrupted Run mode is set (on)
XQØ: Interrupted Run mode is reset (off)

This command can be used only when the Indexer is stand-alone power-up sequencing in **XP9** mode. In **XP9** mode, if **XQ1** is set, the Indexer will not accept a sequence select input until all sequence select inputs are low (closed to GND). After all lines have simultaneously been brought to a low state, the indexer will then read the sequence select lines and execute the sequence whose number appears there. This paused mode will continue until an **XQØ** command is executed. You may use **S** or **K** command to stop sequence execution. **XQ1** must be the first command entered in the sequence, and must be included in every sequence to be controlled by the external inputs.

Example

Command	Description
XE1	Erases sequence 1
XD1	Defines sequence 1
XQ1	Turns Interrupted Run mode on
LD3	Disables CW & CCW limits
XT	Ends Sequence 1
XP9	Sets power-up sequences as sequence select inputs
Z	Resets the indexer to start sequence scanning

If you execute Sequence 1 during power up by setting the SEQ1-SEQ3 inputs properly, Interrupted Run mode will be set. All sequence select input lines must be low (closed to GND) before selecting any other sequences.

XR	Run a Sequence	Software Version: A

Syntax	Units	Range	Default Value
<a>XRn	Sequence No.	n = 1 - 7	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	XD, XE, XRP, XT	N/A

This command loads a pre-defined sequence into the command buffer (clears the buffer first) and executes these commands as a normal set of commands. **XR** automatically recalls the sequence from BBRAM. Before executing the specified sequence, the Indexer is put into the power-on default state with respect to the following conditions:

- The values for distance, velocity, and acceleration are zeroed.
- Move mode is Normal; position mode is Relative (incremental).
- The direction is set to CW.

XR can be used within one sequence to start execution of another sequence; however, all commands in the first sequence following **XR** will be ignored (in this respect an XR acts like a GOTO not a GOSUB). An **XR** command placed within a loop will be ignored.

Example

Command	Description
XE1	Erases sequence 1
XD1	Defines sequence 1
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets acceleration to 5 revs/sec
D1ØØØØ	Sets distance to 10,000 steps
G	Executes the move (Go)
XT	Ends Sequence 1 definition
XR1	Executes Sequence 1

Sequence 1 is defined (XD1) and executed (**XR1**).

XRP Sequence Run With Pause

Software Version: A

Syntax	Units	Range	Default Value
<a>XRPn	Sequence No.	n = 1 - 7	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Programming	XD, XE, XR, XT	N/A

This command is identical to the Sequence Run (**XR**) command, except that it automatically generates a pause condition. You must clear this condition with the Continue (**C**) command before the indexer executes the command buffer. The pause condition is invoked only if the sequence is valid. This allows you to execute a sequence without the delay of buffering that sequence.

Example

Command	Description
XE5	Erases Sequence 5
XD5	Defines Sequence 5
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets velocity to 5 revs/sec
D1ØØØØ	Sets distance to 10,000 steps
G	Executes the move (Go)
XT	Ends definition of Sequence 5
XRP5	Runs Sequence 5 with a pause
C	Indexer executes Sequence 5

Upon issuing XRP5, Sequence 5 is entered in the command buffer, but is not executed. Issue a C command to execute Sequence 5.

XSD — Sequence Status Definition — Software Version: A

Syntax	Units	Range	Default Value
aXSD	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Savable in Sequence	Status	XD, XE, XT	*n

This command reports the status of the previous sequence definition (**XD**...**XT**). The response is 0 - 2. The valid values and descriptions of possible responses are shown below:

n = Ø: Download O.K.
n = 1: A sequence already exists with the number you have specified.
n = 2: Out of memory. The sequence buffer is full.

XSD verifies that the last sequence definition was successful.

Example

Command	Response
1XSD	*1

XSP — Sequence Status Power-up — Software Version: A

Syntax	Units	Range	Default Value
aXSP	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Never Saved	Status	XP, XQ, XSR	*n

The Sequence Status Power-up (**XSP**) determines which, if any, sequence will be executed on power-up. After setting a power-up sequence using the Sequence Power-up (**XP**) command, you can check to make sure that proper sequence will be executed on power-up with **XSP**. The command reports which sequence the system will execute during power-up. The range of values for n is 0 - 9

Example

Command	Description
1XSP	*3 (Indicates that sequence 3, if it exists, will be executed upon power-up or reset.)

XSR — Sequence Status Run — Software Version: A

Syntax	Units	Range	Default Value
aXSR	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Status	XR, XRP	*n

This command allows you to check whether or not the last sequence issued was executed successfully without hitting limits, Stop (**S**), or Kill (**K**). The valid values and responses are shown below.

* **Ø** = Last sequence was successful
* **1** = In a loop
* **2** = Valid sequence
* **3** = Erased
* **4** = Bad checksum
* **5** = Running
* **6** = Killed, stopped

Example	Command	Response
	1XSR	*Ø (Sequence ran O.K.)

XSS Sequence Status

Software Version: A

Syntax	Units	Range	Default Value
aXSSn	Sequences	n = 1 - 7	None

Attributes	Command Type	See Also	Response
Buffered, Never Saved	Status	XD, XE, XT	*n

XSS reports whether the sequence specified by n (representing one of the sequences 1 - 7) is empty, has bad checksum, or is OK.

Ø = Empty
1 = Bad Checksum
3 = O.K.

XSS verifies the existence of sequences and if that portion of memory has been corrupted.

Example	Command	Response
	1XSS1	*Ø (Sequence 1 of device 1 is not defined.)

XT Sequence Termination

Software Version: A

Syntax	Units	Range	Default Value
<a>XT	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Buffered, Never Saved	Programming	XD, XE, XR, XRP	N/A

XT is a sequence terminator. This command flags the end of the sequence currently being defined. Sequence definition is not complete until this command is issued. Properly defined sequences are saved into BBRAM automatically by issuing this command.

Example

Command	Description
XE1	Erases Sequence 1
XD1	Defines Sequence 1
MN	Sets to Normal mode
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets velocity to 5 revs/sec
D8ØØØ	Sets distance to 8,000 steps
G	Executes the move (Go)
XT	Ends sequence definition

XU Upload Sequence

Software Version: A

Syntax	Units	Range	Default Value
aXUn	Sequences	n = 1 - 7	N/A

Attributes	Command Type	See Also	Response
Buffered, Never Saved	Status	F, XD, XE, XT	n

This command sends the contents of sequence n to the host computer via the RS-232C interface, terminated by a carriage return [cr]. The contents of that sequence will appear on the computer CRT. All command delimiters in the sequence will be shown as spaces (ASCII 2ØH). Any device identifiers that were included in the original sequence will also be eliminated (they are not stored in the sequence).

When using a daisy-chain, **XU** must be used cautiously as the contents of the sequence will go to all controllers in the loop between the indexer that is uploading and the host. The **F** command may be used to turn off communication on units you are not uploading from.

Example

Command	Description
2F	Turns off communication to unit 2
3F	Turns off communication to unit 3
1XU1	Uploads sequence 1 from unit 1

Y	Stop Loop	Software Version: A

Syntax	Units	Range	Default Value
<a>Y	N/A	N/A	N/A
Attributes	**Command Type**	**See Also**	**Response**
Immediate, Never Saved	Programming	L, N	N/A

The Stop Loop (**Y**) command takes you out of a loop when the loop completes its current pass. This command does not halt processing of the commands in the loop until the indexer reaches the last command of the current loop. At that time, the indexer executes the command that follows the End Loop (**N**) command. You cannot restart the command loop unless you enter the entire command structure, including the Loop (**L**) and End Loop (**N**) commands.

Example

Command	Description
L	Loops indefinitely
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 steps
T2	Waits 2 seconds
G	Executes the move (Go)
N	Ends loop
Y	Stops loop

The loop requires the motor to move 4,000 steps CW and then wait for 2 seconds. The loop terminates at the end of the loop cycle it is executing when it receives the **Y** command.

Z Reset

Software Version: A

Syntax	Units	Range	Default Value
<a>Z	N/A	N/A	N/A

Attributes	Command Type	See Also	Response
Immediate, Never Saved	Programming	K, S	N/A

The Reset (**Z**) command is equivalent to cycling power to the indexer. This command returns all internal settings to their power-up values. It clears the command buffer. Like the Kill (**K**) command, the **Z** command immediately stops output pulses to the motor.

When you use the **Z** command, the indexer is busy for 1 second and ignores all commands. This command sets all position counters to zero and returns all values except the **XP** command to factory defaults.

Example

Command	Description
1Z	Resets indexer with address 1

Remote Address

Software Version: A

Syntax	Units	Range	Default Value
<a>#n	N/A	n=0-255	0

Attributes	Command Type	See Also	Response
Immediate, Automatically Saved	Set-up	E,F	N/A

This command allows the user to set the unit address via software command rather than hardware input. It will override the hardware address lines, allowing addresses up to 255 to be used. Upon receipt of the command, the Indexer will pass on the daisy chain the address *plus one*, automatically addressing all units on the daisy chain. The address of a specific indexer may be changed by including the current address, e.g. 2#15 will change the address of indexer 2 to 15. However, all subsequent units will become automatically re-addressed unless communication has been disabled with the F command.

#1 - Automatic addressing of all units
Response - #(number of units plus one)

#0 - Return to hardware addressing
Response - #0

If the unit addresses exceed 255 , then the response will be #?. A <CR> must be used with this command.

CAUTION
When using long daisy chains, significant delays are possible in command transmission and execution.

Summary of Commands

A	Acceleration
B	Buffer Status
BS	Buffer Size Status
C	Continue
CG	Correction Gain
CR	Carriage Return
D	Distance
DB	Dead Band
DW	Deadband Window
E	Enable Communications
ER	Encoder Resolution
F	Disable Communications
FS	Encoder Functions Report
FSA	Set Indexer to Incremental/Absolute Mode
FSB	Set Indexer to Motor/Encoder Step Mode
FSC	Enable/Disable Position Maintenance
FSD	Stop on Stall
FSE	Turn On Output Number 1 on Stall
FSF	Stop Motion on Trigger 3
FSG	Turn On Output Number 2 when in Deadband
FSH	Enable Stall Detect
G	Go
G	Go Home
^H	Delete
H	Set Direction
IS	Input Status
K	Kill
L	Loop
LD	Limit Disable
LF	Line Feed
MC	Mode Continuous
MN	Mode Normal
MPA	Mode Position Absolute
MPI	Mode Position Incremental
MRD	Motor Resolution Definition
N	End of Loop
O	Output
OS	Report Homing Function Set-Ups
OSA	Define Active State of End-of-Travel Limits
OSB	Back Up To Home
OSC	Define Active State of Home Switch
OSD	Enable Encoder Z Channel for Home
OSH	Reference Edge of Home Switch
PR	Absolute Position Report
PS	Pause
PX	Report Absolute Encoder Position
PZ	Set Absolute Counter to Zero
"	Quote
Q	Velocity Profiling Mode
R	Request Indexer Status
RA	Limit Switch Status Report
RB	Loop, Pause, Shutdown, Trigger Status Request
RC	Closed Loop Status
RM	Rate Multiplier in Velocity Streaming Mode
RV	Revision Level
S	Stop
SN	Scan
SS	Software Switch Function Status
SSA	RS-232C Echo Control
SSG	Clear/Save the Command Buffer
SSH	Clear/Save Command Buffer on Stop
ST	Shutdown
SV	Servoing Parameter
T	Time Delay
TR	Wait For Trigger
TS	Trigger Input Status
U	Pause and Wait for Continue
V	Velocity
W1	Signed Binary Position Report
W3	Hexadecimal Position Report
XC	Sequence Checksum
XD	Sequence Definition
XE	Sequence Erase
XP	Set Power-up Sequence Mode
XQ	Sequence Interrupted Run Mode
XR	Run a Sequence
XRP	Sequence Run With Pause
XSD	Sequence Status Definition
XSP	Sequence Status Power-up
XSR	Sequence Status Run
XSS	Sequence Status
XT	Sequence Termination
XU	Upload Sequence
Y	Stop Loop
Z	Reset
#	Remote Address Numbering

SAMPLE COMMUNICATIONS PROGRAMS

SIMPLE PDX COMMUNICATIONS PROGRAM FOR IBM PC OR COMPATIBLE (TURBO C)

```
#include <dos.h>
#include <bios.h>
#include <conio.h>

#define COM1            0           /* Port number  */
#define PORT            0x3F8       /* Port address */
#define DATA_READY      0x01        /* Line Status for Received data */
#define TRUE            1
#define FALSE     (!TRUE)

#define SETTINGS (0xE0 | 0x03 | 0x00 | 0x00)    /* 9600 BAUD, 8 Data bit
                                                   No Parity, 1 Stop bit */
void main(void)
{
    int ky, end = FALSE;
    char ch;

    clrscr();
    bioscom (0, SETTINGS, COM1);                /* Setup port */
    cprintf("LIMITED PDX TERMINAL [ESC] to exit ...\n\r");
    while (!end)             /* Main loop */
      {
      while (inportb (PORT+5) & DATA_READY)     * Data received ? */
        if ((ch = inportb (PORT)) != 0)
          {
          putch(ch);                            /* Show character */
          if (ch == 0x0D) putch ('\n');
          }
        if (kbhit())                            /* Keyboard pressed */
          {
          if ((ky = getch()) == '\x1B')         /* Was it ESCape ? */
              end = TRUE;                       /* Signal quit */
          outportb (PORT, ky);                  /* Transmit keypress */
          }
      }
}
```

BUFFERED COMMUNICATION PROGRAM USING COM1 ON AN IBM PC OR COMPATIBLE (BORLAND TURBO C)

```
#include  <conio.h>
#include  <dos.h>
#include  <stdio.h>
#include  <bios.h>
#include  "pdx.h"                                /* Header file for serial interface */

static char CircBuf [BSize];                     /* Declare circular buffer */
unsigned int BufStart = 0, BufEnd = 0;           /* Set buffer pointers up */
int Port;                                        /* Contains serial port address */
char *PDX = "\r1D\r1V\r1A\r1R\r";                /* Settings commands for PDX */

void interrupt (*orgvect)();                     /* Interrupt function pointer */

/* Function prototypes */
void Enable_Int (void);                          /* Start everything */
void Disable_Int (void);                         /* Reset serial port */
void interrupt RX_Handler (void);                /* Data Received handler */
int Send (char c);                               /* Transmit a character */
void SendAString (char *string);                 /* Transmit a string */
int Receive (void);                              /* Get received data from buffer */

void main (void)
{
        int c, end = FALSE;

        clrscr();
        Enable_Int();                            /* Start Interrupts */
        fprintf(stdout, "PDX TERMINAL\npress [ESC] to quit...\n\n");
        SendAString (PDX);                       /* Interrogate PDX status */
        do {                                     /* Main loop */
             if (kbhit())                        /* Keyboard pressed ? */
                {
                if ((c = getch()) == 0x1B)
                    end = TRUE;                  /* if Escape pressed end program */
                Send (c);                        /* Transmit character */
                }
            if ((c = Receive()) != FALSE)        /* Anything received from PDX ? */
                {
                fputc (c & 0x07F, stdout);       /* Display character */
                 if (c == 0x0D) fputc ('\n', stdout);  /* If CR do Carriage Return */
                }
            } while (!end);
            Disable_Int();                       /* Stop interrupts and tidy up */
}

void Enable_Int (void)                           /* Turn on communications interrupts */
{
        int c, far *Addr;

        bioscom (0, SETTINGS, 0);                /* Setup comms parameters */
        Addr = MK_FP (0x0040, 0x00);             /* Find address of COM1 */
        Port = *Addr;
        orgvect = getvect (0x0C);                /* Install the interrupt handler */
        disable();
        setvect (0x0C, RX_Handler);
```

```
        c = inportb (Port + MCR) | MC_INT;                  /* Setup Modem Control Register */
        outportb (Port + MCR, c);
        outportb (Port + IER, RX_INT);                      /* Enable interrupts for data received */
        c = inportb (IMR) & IRQ4;                           /* Tell PIC to interrupt on COM1 */
        outportb (IMR, c);
        enable();
}

void Disable_Int (void)                                     /* Turn off communications interrupts */
{
           int c;

        disable();
        c = inportb (IMR) | ~IRQ4;                          /* Stop PIC interrupts for COM1 */
        outportb (IMR, c);
        outportb (Port + IER, 0);                           /* Reset Interrupt Enable Register */
        outportb (Port + MCR, 0);                           /* Reset Modem Control Register */
        setvect  (0x0C, orgvect);                           /* Restore original vector */
        enable();
}

void interrupt RX_Handler (void)                            /* Interrupt handler - data to CircBuf */
{
        disable();
        if ((inportb (Port + IIR) & RX_MASK) == RX_ID)      /* Data received ? */
           {
           CircBuf[BufEnd++] = inportb (Port);
           BufEnd &= BSize - 1;
           }
        outportb (ICR, EOI);                                /* Acknowledge end of hardware interrupt */
        enable();
}

int Send (char c)                                           /* Output character to the serial port */
{
        long int TimeOut = 0xFFFFL;

        while ((inportb (Port + LSR) & XMTRDY) == 0)        /* Wait for transmitter */
        if (!(--TimeOut)) return (FALSE);
        disable();
        outportb (Port, c);                                 /* Do transmit */
        enable();
        return (TRUE);
}

void SendAString (char *string)                             /* Output a string to the serial port */
{
        while (*string)
           {
           delay(25); Send (*string++);
           }
}

int Receive (void)                                          /* Get the current value in the buffer */
{
        int RX;

        if (BufEnd == BufStart) return (FALSE);             /* Nothing received/Overflow */
```

```
        RX = (int) CircBuf[BufStart++];
        BufStart %= BSize;
        return (RX);
}
```

PDX.H HEADER FILE FOR BUFFERED COMMUNICATION PROGRAM (BORLAND TURBO C)

```
#define BAUD            0xE0        /* BAUD rate of 9600        */
#define DATA            0x03        /* 8 Data bits              */
#define NOPARITY        0x00        /* No parity                */
#define STOP            0x00        /* 1 Stop bit               */
#define SETTINGS (BAUD | DATA | NOPARITY | STOP)

#define IER             1           /* Interrupt Enable         */
#define IIR             2           /* Interrupt ID             */
#define MCR             4           /* Modem control            */
#define LSR             5           /* Line Status              */

#define XMTRDY          0x20        /* Line Status Register TX ready    */
#define MC_INT          0x0F        /* Modem Control Register Setup     */

#define RX_INT          0x01        /* Interrupt when data is received  */

#define RX_ID           0x04        /* Interrupt Ident. Register Data Received   */
#define RX_MASK         0x07        /* Interrupt Ident. Register Mask            */

/* 8259 Programmable Interrupt Controller port addresses */
#define IMR             0x21        /* Interrupt Mask Register port      */
#define ICR             0x20        /* Interrupt Control Port            */
#define EOI             0x20        /* 8259 end of hardware interrupt    */
#define IRQ4            0xEF        /* COM1 mask for PIC                 */

/* General definitions */
#define FALSE           0
#define TRUE            (!FALSE)
#define BSize           0x4000      /* Serial buffer size */
```

QUICK BASIC COMMUNICATIONS PROGRAM USING COM1 ON AN IBM PC OR COMPATIBLE

```
DECLARE SUB CommsError ()
DECLARE SUB ReportStat ()

FALSE = 0
TRUE = NOT (FALSE)

Setup the RS232 port to the following:-
9600 BAUD, No Parity, 8 Data bits, 1 Stop bit. Buffer size is 512 bytes
OPEN "COM1:9600,N,8,1,RS,CS,DS,CD,RB512" FOR RANDOM AS #1
PRINT #1, ""                                   'Clear the buffer
CLOSE #1

'Re-establish communication port settings and data stream
OPEN "COM1:9600,N,8,1,RS,CS,DS,CD,RB512" FOR RANDOM AS #1

Finish = FALSE
CLS
DO
        INPUT "Enter the move or status string : "; STEPMOVE$      'Read in command
        IF STEPMOVE$ = "QUIT" THEN
                Finish = TRUE                                      'Enter QUIT to stop program
        ELSE
                FOR C = 1 TO LEN(STEPMOVE$)                        'Set loop count to string length
                B$ = MID$(STEPMOVE$, C, 1)                         'Get one char from string
                PRINT #1, B                                        'Transmit char to RS232 serial port
                R$ = INPUT$(1, #1)                                 'Read char in from port (Echoed char)
                IF R$ <> B$ THEN CommsError
                NEXT C

                PRINT #1, CHR$(13);                                'Send a carriage return
                R$ = INPUT$(1, #1)                                 'Get Echoed response

                FOR cnt = 1 TO 500                                 'A short delay before response
                NEXT cnt

                CLS
                IF LOC(1) > 0 THEN ReportStat                      'Now check for rec buffer not empty
                END IF
LOOP UNTIL (Finish = TRUE)
CLOSE #1
END
```

```
SUB CommsError
        PRINT "There has been a communications error"
        CLOSE #1
        STOP
END SUB

SUB ReportStat
        PRINT : PRINT "The reported status is as follows :-";
        PRINT : PRINT
        DO WHILE NOT (LOC(1) = 0)
                Report$ = INPUT$(1, #1)                 'Input status string data
                PRINT Report$;                          'Display status string
        LOOP
        PRINT
END SUB
```

MAINTENANCE & TROUBLESHOOTING

Maintenance

Routine maintenance is not necessary, but occasional checking of the following points is recommended. Take care, unexpected motion may occur at any time whilst troubleshooting motion control equipment.

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.

Troubleshooting

Use the following information to help in identifying the problem. If the problem persists, call one of the numbers at the front of this User Guide for engineering assistance.

Fault LED

The red LED indicates one of the following fault conditions:

- Motor wiring short-circuit either across phases or between phases.
- Motor wiring short-circuit phase to GND (earth).
- Motor supply overvoltage or undervoltage
- Internal supply failure.
- Drive internal overtemperature
- A motor with too low an inductance is connected to the drive.

The fault LED will also be lit up if the motor is decelerating a large inertia load too quickly. The drive will shutdown under this condition. If this occurs, you can can either reduce the deceleration rate/load inertia, or use a PDX15-D (has a power dump option). See INSTALLATION section for more information on the regenerative power dump option for the 5 amp drive (PDX15-D).

Note that this LED comes on momentarily when power is removed from the drive. This is caused by the supply rails dropping to zero within the drive and gives an indication that the fault circuitry is functioning correctly. You should always ensure that this 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 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.
- Has the indexer shutdown (ST1) the drive?.
- 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, try running the motor using the self test switch. If the motor fails to move, it is possible that;

- the load is jammed. You should *hear* the drive attempting to move the motor. Remove power from the drive and verify that you can move the load manually away from the point of the jam.

If the motor runs from the self test switch, check one of the following:

- Are the end of travel limits active (See **LD** command)?
- Is the indexer set to a valid motor resolution, (see **MRD**)?
- Is the indexer waiting for a trigger input to change state?
- If the fault **output** is driving a device of very low impedance, this might be indicating a fault condition to the indexer. See the **IS** and **R** commands to verify.

Can't establish serial communications

If the motor rotates using the self-test switch check the following:

- Is the communications cable connected correctly at both ends? Confirm the continuity of each conductor.
- Is the communications software configured correctly (9600 BAUD, 8N1)?
- If the commands are prefixed with an address (e.g. **<u>1</u>R**), is the address correct? The default address of a PDX drive is 8. If there is more than one drive on the serial daisy chain, each must have a unique address - see # command.
- Using a DC voltmeter, measure the voltage between trigger input 1 (pin 20) and Gnd (pin 7). The value should be between 4.9 - 5.1V DC.

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.

Returning the System

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