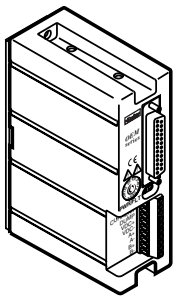
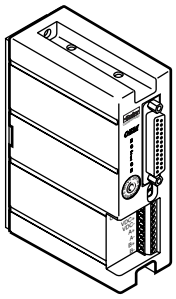
Compumotor

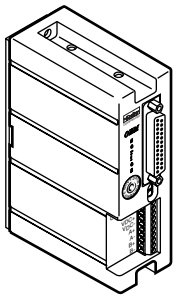
OEM Series
Software Reference Guide
Microstepper Products
OEM750X OEM650X
OEM350X OEM010



OEM750X



OEM650X



OEM350X

OEM010

Compumotor Division
Parker Hannifin Corporation
p/n 88-013785-01 C

Parker

Important User Information

Installation & Operation of Compumotor Equipment

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

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 associated 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 CLEAR of any machinery driven by stepper or servo motors and never touch them while they are in operation.

High voltages exist with in enclosed units, on rack system backplanes, and on transformer terminals. KEEP CLEAR of these areas when power is applied to the equipment.

Parker Compumotor 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 prior consent from Parker Compumotor.

For assistance in the United States, contact:
Compumotor Division of Parker Hannifin
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070
Fax: (707) 584-8015
email: tech_help@cmotor.com
Internet: http://www.compumotor.com

For assistance in Europe, contact:
Parker Digiplan
21 Balena Close
Poole, Dorset
England BH17 7DX
Telephone: 0202-690911
Fax: (707) 584-8015

Part # 88-013785-01C

Compumotor

User Guide Change Summary

Software Reference Guide
OEM750X/010
OEM650X/350X

Revision C
January 2000

This following is a summary of the primary technical changes to this document since the previous version was released. This document, p/n 88-013785-01 **C**, supersedes 88-013785-01 **B**.

Communication Error Checking (pp 47, 65)
New feature — Communication error checking can now be performed on each byte received by the controller. See the "SSE" and the "%" commands (NOTE: This only applies to firmware 92-016638-01).

Software Flow Control (pg 56)
New feature — Software flow control using Xon and Xoff has now been added. See the "XONOFF" command (NOTE: This only applies to firmware 92-016638-01).

Device ID Response (pg 46)
New feature — The device ID can now be included with command responses generated by the controller. See the "SSD" command (NOTE: This only applies to firmware 92-016638-01).

RS485 Communication Option
New option — To extend RS-232 cable length and/or improve noise immunity, RS-485 is available as a custom product request. Contact the Custom Products department for details (NOTE: This only applies to firmware 92-016638-01).

Command Direction Polarity (pg 7)
New feature — The direction the controller considers to be the "positive" direction can be reversed. See the CMDDIR command (Note: This only applies to firmware 92-016638-01).

OEM010 Fault Input (pg 74)
Correction — Previous documentation incorrectly described the OEM010 as having a dedicated fault output (pin 9). This pin actually serves as a dedicated fault input. For the OEM750X, this pin remains unchanged as a dedicated fault output.

Homing Diagrams (pg 81)
Clarification — Homing diagrams have been added to aid troubleshooting system setup.

Firmware Upgrade
Clarification — The new features and options listed in the above change summary are only available for firmware 92-016638-01. All other firmware is not field upgradeable, and will require that the units be returned for hardware and software upgrade at a nominal fee. Contact the Customer Service department for details.

Contents

This page left blank intentionally.

1 Introduction

Chapter Objective

The information in this chapter will enable you to:

❒ Understand which OEM Series products use the software commands listed in this manual.

OEM Series Software Commands

This manual contains descriptions for all software commands applicable to the OEM Series Indexer products listed below. These commands are based on Compumotor's popular and easy-to-use X Series Language. The indexers also provide additional I/O control and communication capabilities.

All of the following products use the same set of software commands.

OEM750X Indexer

The OEM750X Drive/Indexer is the same drive product as the OEM750, but it includes an indexer (position controller). The OEM750X is the same size as the OEM750 and it incorporates the same design features. It is designed for use with step motors that have current ratings up to 7.5 amps, and inductance ratings from 0.2m mH to 80 mH. The OEM750X is the improved replacement for the OEM650X and OEM350X Indexer drives.

Consult the OEM750/OEM750X user Guide (part number 88-016109-01) for question about indexer hardware—dimensions, mounting, installation, etc. Consult this OEM Series Software Reference Guide for software information.

OEM650X Indexer

The OEM650X Drive/Indexer is the same drive product as the OEM650, but it includes an indexer (position controller). The OEM650X is the same size as the OEM650 and it incorporates the same design features. It is designed for use with step motors that have current ratings up to 7.5 amps, and inductance ratings from 1 mH to 10 mH.

Consult the *OEM650/OEM650X User Guide* (part number 88-013157-02) for questions about indexer hardware—dimensions, mounting, installation, etc. Consult this *OEM Series Software Reference Guide* for software information.

OEM350X Indexer

The OEM350X Drive/Indexer is a low power version of the OEM650X. It is designed for use with step motors that have lower current ratings and higher inductance (10 mH to 80 mH) than Compumotor step motors.

Operation of the OEM350X is identical to that of the OEM650X. The two indexers use the same manuals: the hardware user guide mentioned above (p/n 88-013157-02), and this software reference guide. Software commands are identical for both indexers.

OEM010 Indexer

The OEM010 is a stand-alone indexer with the same capabilities as the OEM750X, but for use with other drives. It is compatible with all Compumotor Step and Direction drives (including the Dynaserv) and is suitable for less demanding motion control applications such as:

❒ Rotary Indexing
❒ Step and Repeat
❒ Linear Slide Positioning

The OEM010 uses the same software commands as the OEM750X.

The *Hardware Guide* for the OEM010 is contained in an appendix at the end of this *Software Reference Guide*. Consult the appendix for hardware information about the OEM010. There is no separate hardware reference guide.

2 Software Reference

Chapter Objective

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

Command Descriptions

① A—Acceleration *(Sample)*

② Command Type: Set-Up
③ Syntax: <a>An
④ Range: n = 0.01-999.00
⑤ Attributes: Buffered, Savable in Sequence
⑥ Valid Software Version: A
⑦ Units: revs/sec^2
⑧ Default Value: A = 100
⑨ See Also: D, G, MR, V
⑩ Response to aA is *An

① *Command Mnemonic*

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

② *Command Type*

Set-Up—Set-up commands define application conditions. These commands establish the output data's 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.

③ *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—An *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.

n—An *n* represents an integer. An integer may be used to specify a variety of values (acceleration, velocity, etc.).

s—An *s* indicates that a sign character, either positive or negative (+ or -), is required.

x—An *x* represents any character or string of characters.

④ ***Range***
This is the range of valid values that you can specify for n (or any other parameter specified).

⑤ ***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.

⑥ ***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.

⑦ ***Units***
This field describes what unit of measurement the parameter in the command syntax represents.

⑧ ***Default Value***
The default setting for the command is shown in this box. A command will perform its function with the default setting if you do not provide a value.

⑨ ***See Also***
Commands that are related or similar to the command described are listed here.

⑩ ***Response***
A sample status command and system response are shown. When the command has no response, this field is not shown.

A—Acceleration

- ❑ Command Type: Set-Up
- ❑ Syntax: <a>An
- ❑ Range: n = 0.01-999.00
- ❑ Attributes: Buffered, Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: revs/sec^2
- ❑ Default Value: A = 100
- ❑ See Also: D, G, MR, V
- ❑ Response to aA is *An

The Acceleration command specifies the rotary 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.

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

- ❑ Command Type: Status
- ❑ Syntax: aB
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ Response to aB is *B or *R
- ❑ See Also: BS

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

The command buffer is 512 bytes long. 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.

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

BS—Buffer Size Status

- ❏ Command Type: Status
- ❏ Syntax: aBS
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ Response to aBS is *n
- ❏ See Also: B

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.

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

C—Continue

- ❏ Command Type: Motion
- ❏ Syntax: <a>C
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: PS

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.

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	Accelerates the motor to 10 revs/sec
C	Continues executing commands in the buffer

CG—Correction Gain

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CGn
- ❏ Range: n = 1 - 8
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 8
- ❏ Response to aCG is *CGn
- ❏ See Also: FSB, FSC, DB

This command allows you to set the amount of error (steps) that should be corrected on the initial position maintenance (FSC1 command) correction move (which 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 faster. However, there may be more overshoot and ringing at the end of this type of correction move.

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

CMDDIR—Commanded Direction Polarity

- ❏ Command Type: Set-up
- ❏ Syntax: <a>CMDDIRn
- ❏ Range: 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: E
- ❏ Units: N/A
- ❏ Default Value: 1
- ❏ Also: H,D

The **CMDDIR** command allows you to reverse the direction that the controller considers to be the "positive" direction; this also reverses the polarity of the counts from the feedback device.

CMDDIR0 = CW (+5V), CCW (0V) Recommended for use with Compumotor compatible step & direction drives using the OEM010

indexer.

CMDDIR1 = CW (0V), CCW (+5V) Default for use with OEM750X.

Thus, using the CMDDIR command, you can reverse the referenced direction of the motion without the need to (a) change the connections to the drive and the feedback device, or (b) change the sign of all the motion-related commands in your program.

Command	Response
1CMDDIR	***CMDDIR1**
CMDDIR0	Set-up commanded direction polarity.
1CMDDIR	***CMDDIR0**

Note: Once you change the commanded direction polarity, you should swap the end-of-travel limit connections to maintain a positive correlation with the commanded direction.

CR—Carriage Return

- ❏ Command Type: Programming
- ❏ Syntax: <a>CR
- ❏ Range: N/A
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ Response to aCR is *[cr]
- ❏ See Also: LF

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.

Command	Description
MN	Sets mode to preset mode
A5Ø	Sets acceleration to 50 revs/sec^2
V5	Sets Velocity to 5 revs/sec
D5ØØØ	Sets distance to 5,000 steps
G	Executes the move (Go)
1CR	Sends a carriage return after move is completed

The motor moves 5,000 steps. When the motor stops, the indexer

sends a carriage return over its interface.

D—Distance

- ❑ Command Type: Motion
- ❑ Syntax: <a>D(s)n
- ❑ Range: n=±2,147,483,648
- ❑ Attributes: Buffered
  Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: steps
- ❑ Default Value: 25,000
- ❑ See Also: A, CMDDIR, G, MN, MPA, MPI, V

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.

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
D25ØØØ	Sets distance to 25000 steps
G	Executes the move

A 25000-step resolution motor will travel 1 rev (CW) after **G** is issued.

DB—Dead Band

- ❑ Command Type: Set-up
- ❑ Syntax: <a>DBn
- ❑ Range: n = 0 - 999,999,999
- ❑ Attributes: Buffered,
  Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: n = steps
- ❑ Default Value: 0
- ❑ Response to aDB is *DBn
- ❑ See Also: FSG, CG

This command 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.

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

- ❏ Command Type: Set-up
- ❏ Syntax: <a>DWn
- ❏ Range: n = 0 - 999,999,999
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: n = steps
- ❏ Default Value: 0
- ❏ Response to aDW is *DWn
- ❏ See Also: FS commands

This command allows precise dead band specification in motor steps. The backlash dead band 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 command is most effective when the encoder is mounted on the load.

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

- ❑ Command Type: Programming
- ❑ Syntax: <a>E
- ❑ Range: N/A
- ❑ Attributes: Immediate Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: Enabled
- ❑ See Also: F, XONOFF, %

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.

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
D5ØØØ	Sets distance to 5000 steps
G	Executes the move (Go—only axes 1 & 4 will move)

ER—Encoder Resolution

- ❑ Command Type: Set-up
- ❑ Syntax: <a>ERn
- ❑ Range: n = 1 - 50,000
- ❑ Attributes: Buffered, Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: n = steps/rev
- ❑ Default Value: 4000
- ❑ Response to aER is *ERn
- ❑ See Also: FS, DW

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 you set the motor resolution (**MR** command) at least five times higher than the encoder resolution.

Command	Description
MR2ØØØØ	Sets indexer to control a motor of 20000 steps/rev
ER4ØØØ	Sets encoder resolution to 4000 encoder steps per 1 motor revolution
1ER	Reports Encoder Resolution (*ER4000)

F—Disable Communications

- ❏ Command Type: Programming
- ❏ Syntax: <a>F
- ❏ Range: N/A
- ❏ Attributes: Immediate Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: None
- ❏ See Also: E, XONOFF, %

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.

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

- ❏ Command Type: Status
- ❏ Syntax: <a>FS
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: n = N/A
- ❏ Default Value: None
- ❏ Response to aFS is *nnnnnnnn
- ❏ See Also: ER, PX

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

Enables position maintenance. This 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 E 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.

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

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSAn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: MPI, MPA, PZ, PR, PX

This command sets the indexer to perform its moves in either absolute or incremental positioning mode.

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.

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

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSBn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: ER, D, FSC

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.

Command	Description
ER4ØØØ	Set up encoder where 4,000 encoder pulses (1,000 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 1 motor revolution) are received

FSC—Enable/Disable Position Maintenance

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSCn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: 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.

Command	Description
ER2ØØØ	Set encoder resolution to 2,000.
FSB1	Set encoder step mode.
FSC1	Enable position maintenance
FSH1	Enable stall detection
FSD1	Enable stop on stall

FSD—Stop on Stall

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSDn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: DW, ER, FSH

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; no deceleration. This command is valid only if stall detection (FSH1) has been enabled. It will have no effect otherwise.

Command	Description
DW1ØØ	Set backlash value to 100 steps.
ER2ØØØ	Set encoder resolution to 2,000 steps/rev.
FSB1	Set indexer to encoder step mode
FSH1	Enable stall detect.
FSD1	Enable stop on stall.

FSE—Turn on Output Number 1 on Stall

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSEn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: SS, DW, ER, FSH, FSF

FSEØ = Do not turn on output #1 on stall

FSE1 = Turn on output #1 on stall

Entering FSE1 will cause the indexer to turn on output number 1 when a stall is detected. This is useful for signaling other components in you 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 uneffected by a stall when **FSEØ** is entered.

This output will also turn on if Stop Motion on Trigger 3 (FSF1) is enabled.

Command	Description
ER2ØØØ	Set encoder resolution to 2,000 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

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSFn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: TR, FSE, 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. Setting up another unit to turn on Output #1 when it detects a stall with the Turn on Output on Stall (**FSE**) command, enables the user to implement a multi-axis stop on stall axis by connecting output 1 of one axis to the trigger 3 input on the other. The input may be used as a trigger, but will stop motion when TR3 is entered. The deceleration rate is set by the **A** setting.

Entering **FSFØ** will turn this feature off.

Command	Description

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

FSG—Turn on Output 2 when within Dead Band

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSGn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: DB, FSB, FSC, FSH

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.

Command	Description
ER4ØØØ	Set encoder resolution to 4,000 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

- ❏ Command Type: Set-up
- ❏ Syntax: <a>FSHn
- ❏ Range: n = 0, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: 0
- ❏ See Also: FS commands, DW, ER

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.

Command	Description
DW1ØØØ	Set dead band window to 1,000 steps
ER2ØØØ	Set encoder resolution to 2,000 steps (500 lines)
FSB1	Set indexer to encoder step mode
FSH1	Enable stall detection
FSD1	Stop motor movement if stall detected.

G—Go

- ❑ Command Type: Motion
- ❑ Syntax: <a>G
- ❑ Range: N/A
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: None
- ❑ See Also: A, D, MC, MN, S, V

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.

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

- ❏ Command Type: Motion
- ❏ Syntax: <a>GHsn
- ❏ Range: n = .01 - 50.00
  s = + or -
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Revs/sec
- ❏ Default Values: n = 0, s = +
- ❏ See Also: OS Commands, RC, V

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 limit signal changes states nearest 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.

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

^H—Delete

- ❑ Command Type: Programming
- ❑ Syntax: ^H
- ❑ Range: N/A
- ❑ Attributes: Immediate
  Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: None

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.

CAUTION

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

Pressing the delete key will not delete the previous character.

H—Set Direction

- ❑ Command Type: Programming
- ❑ Syntax: <a>H(s)
- ❑ Range: s = + or -
- ❑ Attributes: Buffered
  Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: +
- ❑ See Also: CMDDIR, D

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

Command	**Description**
MN	Sets Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D25ØØØ	Sets distance to 25,000 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

- ❑ Command Type: Status
- ❑ Syntax: aIS
- ❑ Range: N/A
- ❑ Attributes: Immediate
  Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: D
- ❑ Response to aIS is *nnnnnnnnnnn

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 not 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.

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

- ❏ Command Type: Motion
- ❏ Syntax: <a>K
- ❏ Range: N/A
- ❏ Attributes: Immediate
  Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: S

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.

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

- ❏ Command Type: Programming
- ❏ Syntax: <a>Ln
- ❏ Range: n = 0 - 65,535
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: number of loops
- ❏ Default Value: None
- ❏ See Also: C, N, U, Y

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.

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

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>LDn
- ❏ Range: n = 0 - 3
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: None
- ❏ See Also: RA, TR, TS

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, Compumotor strongly suggests that you wire the hardware limit switches to prevent the load from jamming into the end-of-travel limit.

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

- ❏ Command Type: Programming
- ❏ Syntax: <a>LF
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: CR
- ❏ Response to <a>LF is [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.

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

- ❏ Command Type: Motion
- ❏ Syntax: <a>MC
- ❏ Range: N/A
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Inactive
- ❏ See Also: A, MN, T, TR, V

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

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

- ❏ Command Type: Motion
- ❏ Syntax: <a>MN
- ❏ Range: N/A
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Active
- ❏ See Also: A, D, G, MC, MPA, MPI

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**).

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

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>MPA
- ❏ Range: N/A
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Inactive
- ❏ See Also: D, MN, MPI, PZ

This command sets the positioning mode to absolute. 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.

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 Ø) and then move 15,000 steps in the CCW direction to reach the absolute position 10,000.

MPI—Mode Position Incremental

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>MPI
- ❏ Range: N/A
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Status: Active
- ❏ See Also: D, MN, MPA

This command sets the positioning mode to incremental. 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.

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).

MR—Motor Resolution

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>MRn
- ❏ Range: n = 200 - 1,024,000
- ❏ Attributes: Buffered
  Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: steps/rev
- ❏ Default Value: 25,000
- ❏ See Also: A, V

The Motor Resolution (**MR**) command sets the number of steps per revolution. **MR** allows the indexer to control drives of different resolutions while maintaining specified acceleration and velocity values. This command does not change the actual resolution of the motor. The resolution of the motor is dependent on the drive. This command determines the Velocity (**V**) and Acceleration (**A**) for motors with different resolutions. The MR command only accepts values that correspond to valid motor resolution options (see list below). If an invalid motor resolution value is entered, the **MR** command will be ignored and the last valid motor resolution setting entered will be used.

Valid OEM Indexer Motor Resolutions (Steps/Rev)

200	12,800	25,600	507,904
400	18,000	36,000	614,400
1,000	20,000	50,000	655,360
2,000	21,600	50,800	819,200
5,000	25,000	278,528	1,024,000
10,000	25,400	425,984	

Command	Description
MN	Set positioning mode to preset
MR25ØØØ	Sets indexer to control a motor of 25000 steps/rev
A5	Set acceleration to 5 revs/sec^2
V1Ø	Set velocity to 10 revs/sec
D25ØØØ	Set distance of move to 25000 steps
G	Executes the move (Go)

A 25,000 step per revolution motor/drive will turn 25000 steps (one revolution) CW at an acceleration of 5 revs/sec^2 and a velocity of 10 revs/sec after the **G** command.

If the same command set is sent to a motor/drive with a resolution of 50,000, the motor will turn 25,000 steps (1/2 revolution), but the actual acceleration would be 2.5 revs/sec^2 and the actual velocity would be 5.0 revs/sec.

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

N—End of Loop

- ❏ Command Type: Programming
- ❏ Syntax: <a>N
- ❏ Range: N/A
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: C, L, PS, U

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.

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

- ❏ Command Type: Programming
- ❏ Syntax: <a>Onn
- ❏ Range: Ø, 1 or X (See Below)
- ❏ Attributes: Buffered Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: on, off, or unchanged
- ❏ Default Value: ØØ
- ❏ See Also: OS, S, TR, TS

The Output (**O**) command turns the programmable output bits on and off. This is used for signaling 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

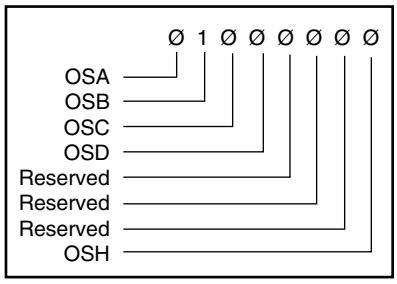
n=X = Leaves output bits unchanged

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

- ❏ Command Type: Status
- ❏ Syntax: <a>OS
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: OS(A-H)
- ❏ Response to aOS is 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

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSAn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: LD, OSC

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.

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

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSBn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: 1
- ❏ See Also: GH, OSC, OSH

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 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 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 limit input is still active. If the deceleration overshoots the active home region the motor will reverse direction and travel in the active home region and the Go Home (**GH**) command is given no motion will occur.

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

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSCn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: GH, OSB, OSH

OSCØ: Active state of home input is n = Ø (closed)

OSC1: Active state of home input is n=1 (open)

OSCØ requires that a normally open (high) switch be connected to the home limit input. **OSC1** requires that a normally closed (low) switch be connected to the home limit input.

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

OSD—Enable Encoder Z Channel for Home

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

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 limit) 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.

Command	Description
OSD1	Recognizes Z channel as final home reference

OSH—Reference Edge of Home Switch

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>OSHn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: GH, OSB, OSC

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 traveling 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 traveling from the CCW limit in the CW direction.

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 limit becomes active when the home limit input (pin #5 on 25-pin D connector) is closed. The indexer recognizes the CCW edge of the switch as the home limit and backs up to that edge to complete the Go Home move.

PR—Absolute Position Report

- ❏ Command Type: Status
- ❏ Syntax: aPR
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: D, MPA, MPI, MN, PZ
- ❏ Response to aPR is *±nnnnnnnnnn

PR reports motor position relative to the power-up position. The response is ±nnnnnnnnnn (range = 0 - ±2,147,483,648). If 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.

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

- ❏ Command Type: Programming
- ❏ Syntax: <a>PS
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: C

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.

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
D25ØØØ	Sets move distance to 25,000 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 25,000 steps twice with a 2-second delay between moves.

PX—Report Absolute Encoder Position

- ❏ Command Type: Status
- ❏ Syntax: aPX
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ Response to aPX is *nnnnnnn
- ❏ See Also: W3, PR, FSB

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

•Closing the loop with the host though positioning with n steps

•End of move (verification of position)

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
D256ØØ	Sets move distance to 25,600 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 *+0000008000, assuming the ER command is set to 4000.

PZ—Set Absolute Counter to Zero

- ❑ Command Type: Programming
- ❑ Syntax: <a>PZ
- ❑ Range: N/A
- ❑ Attributes: Buffered, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: D, MN, PR, PX

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

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

- ❏ Command Type: Programming
- ❏ Syntax: "x
- ❏ Range: x = up to 17 ASCII characters
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: CR, LF
- ❏ Response to "x is 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.

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 OEM Indexer will send the message MOVE_DONE via the RS-232C port

QØ—Exit Velocity Profiling Mode

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>QØ
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: Q1, RM

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

Q1—Enter Velocity Profiling Mode

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>Q1
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: QØ, RM

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

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

R—Request Indexer Status

- ❏ Command Type: Status
- ❏ Syntax: aR
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: RA, RB, RC, XSR, XSS
- ❏ Response to aR is *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:

* Drive faulted
* 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.

Command	Response
R	***R** (Indexer ready to accept a command, and no error conditions exist.)

RA—Limit Switch Status Report

- ❑ Command Type: Status
- ❑ Syntax: aRA
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: R, RB
- ❑ Response to aRA is *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*		*Current Limit Status*	
	CW Limit	*CCW Limit*	*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 Ready Status (**R**) will return a ***S**.

Command	**Response**
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

- ❏ Command Type: Status
- ❏ Syntax: aRB
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: L, PS, R, RA, ST, TR
- ❏ Response to aRB is *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.

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

- ❏ Command Type: Status
- ❏ Syntax: aRC
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ Response to aRC IS *x
- ❏ See Also: R, RA, RB, FS, GH

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

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

- ❏ Command Type: Motion
- ❏ Syntax: <a>RMn
- ❏ Range: n = ØØØ - FFF
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: revs/sec
- ❏ Default Value: None
- ❏ See Also: D, H, ĢØ, Ģ1

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) • *(resolution constant)* = **decimal # 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*
200	655.3600
400	655.3400
1000	652.7801
2000	648.5135
5000	639.9805
10000	639.9805
12800	655.3400
18000	614.3812
20000	597.3151
21600	645.1003
25000	639.9804
25400	650.2202
25600	655.3399

36000	767.9766
50000	1066.6341
50800	1083.7002

The resolution that the **RM** 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 OEM Indexer to exit Velocity Streaming 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 OEM Indexer to enter Normal mode (**MN**).

Command	**Response**
Q1	Enter Velocity Streaming mode
RMØØAØ	Accelerate to 0.25 revs/sec^2
RMØ14Ø	Accelerate to 0.5 revs/sec^2
RMØ28Ø	Accelerate to 1 revs/sec^2
RMØ5ØØ	Accelerate to 2 revs/sec^2
RMØ28Ø	Decelerate to 1 revs/sec^2
RMØ14Ø	Decelerate to 0.5 revs/sec^2
RMØØAØ	Decelerate to 0.25 revs/sec^2
RMØØØØ	Decelerate to 0 revs/sec^2
QØ	Exit Velocity Streaming mode

RV—Revision Level

- ❑ Command Type: Status
- ❑ Syntax: aRV
- ❑ Range: N/A
- ❑ Attributes: Buffered, Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also:
- ❑ Response to aRV is *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 Parker Compumotor's Applications Department.

Command	**Response**
1RV	***92-Ø16678-Ø1E**

The product is identified by *92-Ø16678-Ø1E, and the revision level is identified by E.

S—Stop

- ❑ Command Type: Motion
- ❑ Syntax: <a>S
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: A, K, QØ, SSH

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.

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

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SNn
- ❏ Range: 1 - 1000
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: n = mS
- ❏ Default Value: 50
- ❏ See Also: XP

The Scan (**SN**) command allows you to define the *debounce time* (in milliseconds) for external sequence selection inputs. The debounce time is the amount of time that the sequence inputs must remain constant for a proper reading from a remote controller, such as a programmable logic controller (PLC). If you are using a PLC you should change the debounce time to match the *on time* of the PLC outputs.

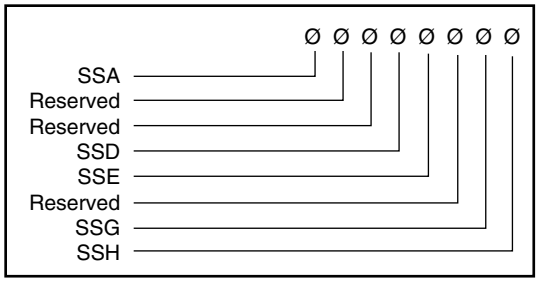
This command allows you to select the best possible trade-off between noise immunity and speed for a given application. If you make your scan time too short, the OEM Indexer may respond to an electrical glitch. If you issue the Scan command with only a device address (**1SN**), the indexer will respond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

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

SS—Software Switch Function Status

- ❏ Command Type: Status
- ❏ Syntax: aSS
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: SSA, SSG, SSH
- ❏ Response to aSS is *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
- **SSD** = Ø
- **SSE** = Ø
- **SSG** = Ø
- **SSH** = Ø

SSA—RS-232C Echo Control

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSAn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also:

This command turns the RS-232C echo (transmission of characters received from the remote device by the OEM 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 OEM 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.*

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

SSD—Add Device ID Prefix

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSDn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Saveable in Sequence
- ❏ Valid Software Version: E
- ❏ Units: N/A
- ❏ Default Value: Ø
- ❏ See Also: E, F, XONOFF, #, %

This command is used to add the controller's device ID number (address) to all response messages (except where noted). **SSD1** will enable the device ID prefix, and **SSDØ** will disable the device ID prefix.

SSDØ = Disable device ID prefix
SSD1 = Enable device ID prefix

Command responses that are not prefixed with the device ID are listed below:

Command	**Description**
CR	Carriage Return
LF	Line feed
“	Quote command
W1	Signed binary position report
#	Remote address numbering
XU	Upload sequence
%	Reset communication

NOTE: The device ID prefix always has 3 digits with a range of 1 - 255.

Example, device ID 23:

Command (SSDØ)	**Response**
23V	*V1
23SS	*00000000
“Hello	Hello

Command (SSD1)	**Response**
23V	*023V1
23SS	*02300000000
“Hello	Hello

SSE—Enable/Disable Communication Error Checking

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSEn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Saveable in Sequence
- ❏ Valid Software Version: E
- ❏ Units: N/A
- ❏ Default Value: Ø
- ❏ See Also: %

SSEØ = Disables error checking
SSE1 = Enables error checking

This command setting determines whether or not each byte received at the controller is checked for communication errors. **SSE1** enables error checking for all bytes received at controller, and **SSEØ** disables error checking. See the **%** command for the types of errors detected.

Command	Description
SSE1	Enables error checking
SSEØ	Disables error checking

SSG—Clear/Save the Command Buffer on Limit

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSGn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: LD

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.

Command	Description
SSG1	Saves buffer on limit
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D25ØØØ	Sets distance to 25,000 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

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>SSHn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø
- ❏ See Also: S

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.

Command	Description
SSHØ	Clears buffer on stop
A1Ø	Sets acceleration to 10 revs/sec^2
V5	Sets velocity to 5 revs/sec
D25ØØØ	Sets distance to 25,000 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

- ❏ Command Type: Programming
- ❏ Syntax: <a>STn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: Ø

The Shutdown (**ST1**) command rapidly decreases motor current to zero. The system ignores move commands that are issued after **ST1**. 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. **STØ** reduces motor heating, and allows you to manually position the load. 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.

For the **ST** command to work, the shutdown output from the indexer must be wired to the shutdown input of the drive.

Command	Description
ST1	Shuts off current to the motor

SV—Servoing Parameter

- ❏ Command Type: Set-up
- ❏ Syntax: <a>SVn
- ❏ Range: n = 0 - 3
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: None
- ❏ Response to aCG is *n
- ❏ See Also: FSC, ST

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.

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

T—Time Delay

- ❏ Command Type: Programming
- ❏ Syntax: <a>Tn
- ❏ Range: n = 0.01 - 99999.99
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: seconds
- ❏ Default Value: None

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.

Command	Description
MN	Sets Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D25ØØØ	Sets distance to 25,000 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

- ❏ Command Type: Programming
- ❏ Syntax: <a>TRnnn
- ❏ Range: n = Ø, 1, or X
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: See Below
- ❏ Default Value: None
- ❏ See Also: TS

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

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
D25ØØØ	Sets distance to 25,000 steps
G	Executes the move (Go)

TS—Trigger Input Status

- ❑ Command Type: Status
- ❑ Syntax: aTS
- ❑ Range: n = 0, 1, or X
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: See Below
- ❑ Default Value: None
- ❑ See Also: TR
- ❑ Response to aTS is *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.

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

- ❑ Command Type: Programming
- ❑ Syntax: <a>U
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: C, PS

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.

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

- ❏ Command Type: Motion
- ❏ Syntax: <a>Vn
- ❏ Range: n = 0.01 - 50.00
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: revs/sec
- ❏ Default Value: 1
- ❏ See Also: A, D, G, GH, MR

The **V** command defines the maximum speed at which the motor will run when given the Go (**G**) command. The actual speed of the motor or output frequency of the indexer will vary, depending on the resolution setting (see table below).

Resolution	Maximum Velocity
200 - 25,600 steps/rev	50 rps
36,000 steps/rev	40 rps
50,000 - 50,800 steps/rev	30 rps
278,528 steps/rev	4.5 rps
425,984 steps/rev	3.0 rps
507,904 steps/rev	2.5 rps
614,400 steps/rev	2.0 rps
819,200 steps/rev	1.5 rps
1,024,000 steps/rev	1.25 rps

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

*If the value specified for the **V** command is not valid, the OEM 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**).*

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**), when a Go (**G**) command is complete—the indexer moves to the next command in the buffer—once the specified velocity is reached.

W1—Signed Binary Position Report

- ❏ Command Type: Status
- ❏ Syntax: aW1
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: PR, W3
- ❏ Response to aW1 is *xxxx

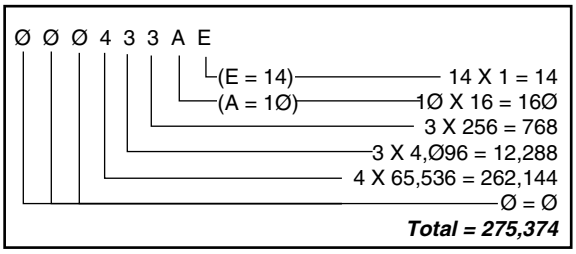
Report back gives immediate binary representation of position relative to start of the current move. The format of the response is a four character response (nnnn) that is interpreted as a 32-bit binary number. The computer interprets the number to give a numerical position in steps. The format is in 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.*

W3—Hexadecimal Position Report

- ❏ Command Type: Status
- ❏ Syntax: aW3
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: PR, W1
- ❏ Response to aW3 is *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 168 (232) (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 164	1ØØØØ hex	=	65,536
		Result **=**	***-25,000***

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

XC—Sequence Checksum

- ❏ Command Type: Status
- ❏ Syntax: aXC
- ❏ Range: N/A
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: None
- ❏ See Also: XD, XE

XC computes the BBRAM checksum. After the unit is programmed, the response can be used for system error checking. The three-decimal reposnse (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 OEM Indexer is not re-programmed, the checksum response should always be the same.

Command	Response
1XC	***149**

XD—Sequence Definition

- ❏ Command Type: Programming
- ❏ Syntax: <a>XDn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: None
- ❏ See Also: XE, XR, XRP, XT

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 255 characters. Immediate commands cannot be entered into a sequence. Sequence can only be permanently saved with the -M2 (BBRAM) option.

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 defintion of Sequence #1
XR1	Executes Sequence #1

XE—Sequence Erase

- ❏ Command Type: Programming
- ❏ Syntax: <a>XEn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: None
- ❏ See Also: XD, XR, XRP, XT

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

Command	Description
XE1	Deletes Sequence 1
XD1	Defines Sequence 1

XONOFF—Enable/Disable XON/XOFF

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>XONOFFn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Saveable in Sequence
- ❏ Valid Software Version: E
- ❏ Units: N/A
- ❏ Default Value: Ø
- ❏ See Also: E, F, %

Use the XONOFF command to enable or disable XON/XOFF (ASCII handshaking).

XONOFF1 = Enable XON/XOFF
XONOFFØ = Disable XON/XOFF

XONOFF1 enables XON/XOFF, which allows the controller to recognize ASCII handshaking control characters. When XON/XOFF is enabled, ASCII 17 or ^Q is a signal to start sending characters, and ASCII 19 or ^S is a signal to stop sending characters. **XONOFFØ** disables XON/XOFF.

NOTE: Disable flow control when connecting multiple units together in a daisy chain.

XP—Set Power-up Sequence Mode

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>XPn
- ❏ Range: n = 0 - 9
- ❏ Attributes: Buffered, Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: 0
- ❏ See Also: XQ, XSP, XSR

This command executes a single sequence or multiple sequences on power-up. If *n* = 1-7, the sequence whose value = *n* being 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 OEM 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 **XQ1**command to stop the OEM Indexer from selecting the next sequence until all the sequence select inputs are first opened.

Command	Description
XP1	Executes Sequence #1 on power-up
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
Z	Resets the indexer

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

XQ—Sequence Interrupted Run Mode

- ❏ Command Type: Set-Up
- ❏ Syntax: <a>XQn
- ❏ Range: n = Ø, 1
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: Ø
- ❏ See Also: XP

n = 1: Interrupted Run mode is set (on)
n = Ø: Interrupted Run mode is reset (off)

This command can be used only when the OEM Indexer is stand-alone power-up sequencing in **XP9** mode. In **XP9** mode, if **XQ1** is executed, the OEM Indexer will not accept a sequence select input until all sequence select inputs are OFF (closed). After all lines have simultaneously been brought to a low state (OFF), 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.

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 high (open) before selecting any other sequences.

XR—Run a Sequence

- ❏ Command Type: Programming
- ❏ Syntax: <a>XRn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Sequence
- ❏ Default Value: 0
- ❏ See Also: XD, XE, XRP, XT

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.

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.

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

- ❏ Command Type: Programming
- ❏ Syntax: <a>XRPn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: Sequence
- ❏ Default Value: 0
- ❏ See Also: XD, XE, XR, XT

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.

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

- ❏ Command Type: Programming
- ❏ Syntax: aXSD
- ❏ Range: N/A
- ❏ Attributes: Buffered, Savable in Sequence
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: XD, XE, XT
- ❏ Response to aXSD is *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.

Command	Response
1XSD	*1

XSP—Sequence Status Power-up

- ❏ Command Type: Status
- ❏ Syntax: aXSP
- ❏ Range: N/A
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: XP, XQ, XSR
- ❏ Response to aXSP is *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 sequences is Ø - 9.

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

XSR—Sequence Status Run

- ❏ Command Type: Status
- ❏ Syntax: aXSR
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: XR, XRP
- ❏ Response to aXSR is *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** = Invalid sequence
* **3** = Erased
* **4** = Bad checksum
* **5** = Running
* **6** = Killed,stopped

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

XSS—Sequence Status

- ❑ Command Type: Status
- ❑ Syntax: aXSSn
- ❑ Range: n = 1 - 7
- ❑ Attributes: Buffered, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: Sequences
- ❑ Default Value: None
- ❑ See Also: XD, XE, XT
- ❑ Response to aXSS is *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.

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

XT—Sequence Termination

- ❑ Command Type: Programming
- ❑ Syntax: <a>XT
- ❑ Range: N/A
- ❑ Attributes: Buffered, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: XD, XE, XR, XRP

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 (*-M2 Option Only*) automatically by issuing this command.

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
D25ØØØ	Sets distance to 25,000 steps
G	Executes the move (Go)
XT	Ends sequence definition

XU—Upload Sequence

- ❏ Command Type: Status
- ❏ Syntax: aXUn
- ❏ Range: n = 1 - 7
- ❏ Attributes: Buffered, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: Sequences
- ❏ Default Value: N/A
- ❏ See Also: F, XD, XE, XT
- ❏ Response to aXUn is contents of sequence 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 (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.

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

- ❏ Command Type: Programming
- ❏ Syntax: <a>Y
- ❏ Range: N/A
- ❏ Attributes: Immediate, Never Saved
- ❏ Valid Software Version: A
- ❏ Units: N/A
- ❏ Default Value: N/A
- ❏ See Also: L, N

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 processes reach 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.

Command	Description
L	Loops indefinitely
A1Ø	Sets acceleration to 10 revs/sec2
V5	Sets velocity to 5 revs/sec
D25ØØØ	Sets distance to 25,000 steps
T2	Waits 2 seconds
G	Executes the move (Go)
N	Ends loop
Y	Stops loop

The loop requires the motor to move 25,000 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

- ❑ Command Type: Programming
- ❑ Syntax: <a>Z
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: K, S

The Reset (**Z**) command is equivalent to cycling DC 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 1,000 ms and ignores all commands. Any changes that you do not save before issuing this command will be lost. This command sets all position counters to zero and returns all values except the **XP** command to factroy defaults.

Command	Description
1Z	Resets indexer with address 1

#—Remote Address

- ❏ Command Type: Set-up
- ❏ Syntax: <a>#n
- ❏ Range: n=0-255
- ❏ Attributes: Immediate, Automatically Saved
- ❏ Valid Software Version: A
- ❏ Units:
- ❏ Default Value: 0
- ❏ See Also: E,F

This command allows the user to set the unit address via software command rather than hardware input. This allows the user to override the hardware address lines, allowing addresses up to 255. Upon receipt of the command, the OEM Indexer will pass on the daisy chain the address *plus one*, thus enabling automatic addressing of all units on the daisy chain. The address may also be set individually if preferred.

#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.

%—Reset Communication

- ❑ Command Type: Status
- ❑ Syntax: %
- ❑ Range: N/A
- ❑ Attributes: Immediate, Never Saved
- ❑ Valid Software Version: E
- ❑ Units: N/A
- ❑ Default Value: N/A
- ❑ See Also: E, F, SSE, XONOFF

When a communication error is detected, all external commands are ignored by echoing an **&** for each byte received from the host. This command is used to re-establish communication, and to identify the cause of the communication error.

In a daisy-chained environment, units located downstream from the unit detecting a communication error will also disable external command processing. Units upstream in a daisy chain are not affected.

NOTE: Error detection will only occur if **SSE1** is enabled. Detection of a communication error has no effect on internal command processing, or sequence execution. A communication error will not stop motion.

Possible responses are as follows:

Character	**Definition**
*	No errors
*Ø	Unit upstream (daisy chained)
*2	Framing error

Command	**Response**
%	*2 (Either host or controller has lost synchronization)

NOTE: For daisy chained environments, the response values are in reverse order.

%	***Ø*Ø*Ø*2***** (First 5 units report no error, 6th unit detected a framing error, and the last 3 units turned communication off because of unit 6.)

Summary of Commands

A—Acceleration
B—Buffer Status
BS—Buffer Size Status
C—Continue
CG—Correction Gain
CMDDIR—Commanded Direction Polarity
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
GH—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
MR—Motor Resolution
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
Q1—Enter Velocity Profiling Mode
QØ—Exit 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
SSD—Add Device ID Prefix
SSE—Enable/Disable Communication Error Checking
SSG—Clear/Save the Command Buffer on Limit
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
XONOFF—Enable/Disable XON/XOFF
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
%—Reset Communication

Appendix: OEM010 User Guide

This appendix is designed to help you install, develop, and maintain your system. Each section begins with a list of specific objectives that should be met after you have read the chapter. This section will help you find and use the information in this appendix.

Assumptions

To use this product and its instructions effectively, you should have a fundamental understanding of the following information.

- ❏ Electronics concepts (voltage, switches, current, etc.)
- ❏ Motion control concepts (torque, velocity, distance, force, etc.)

Appendix Contents

Section A1: Introduction

This chapter provides a description of the product and a brief account of its specific features.

Section A2: Installation

This chapter contains a ship kit list of items you should have received with your OEM010. Instructions to mount and connect the system properly are included. Upon completion of this chapter, your system should be completely installed and ready to perform basic operations.

Section A3: Specifications

This chapter contains information on system performance specifications (environmental, etc.).

Section A4: Troubleshooting

This chapter contains information on identifying and resolving system problems. Descriptions of LED signals, debugging tools, problems/solutions table are included.

Installation Process Overview

To ensure trouble-free operation, pay special attention to the environment in which the equipment will operate, the layout and mounting, and the recommended wiring and grounding. These recommendations will help you easily and safely integrate the OEM010 into your manufacturing facility. If your environment contains conditions that may adversely affect solid-state equipment (electrical noise or atmospheric contamination), be sure to follow any special instruction

to ensure the safety and long life of your equipment.

Installation Preparation

Before you install this product, complete the following steps:

1. Review this user guide. Become familiar with the user guide's contents so that you can quickly find the information you need.

2. Develop a basic understanding of all system components, their functions, and inter-relationships.

3. Complete the basic system configuration and wiring instructions (in a simulated environment, not a permanent installation) provided in *Section A2, Installation*.

4. Perform as many basic functions as you can with the preliminary configuration. Try to simulate the task(s) that you expect to perform when you permanently install your application (however, do not attach a load at this time). This will give you a realistic preview of what to expect from the complete configuration.

5. After you have tested the system's functions and become familiar with the system's basic features, carefully read Section *A2*.

6. After you have read Section *A2* and clearly understand what must be done to properly install the system, begin the installation process. Do not deviate from the instructions provided.

7. Before you customize your system, check all of the system functions and features to ensure that you have completed the installation process correctly.

The successful completion of these steps will prevent subsequent performance problems and allow you to isolate and resolve potential system difficulties before they affect your system's operation.

Warnings & Cautions

Warning and caution notes alert you to problems that may occur if you do not follow the instructions correctly. Situations that may cause bodily injury are presented as warnings. Situations that may cause system damage are presented as cautions.

CAUTION

System damage will occur if you power up the system improperly.

A1 Introduction

Section Objective

The information in this section will enable you to:

❒ Understand the product's basic functions and features

OEM010 Description

The OEM010 is a stand-alone indexer with the same capabilities as the OEM750X, OEM650X and OEM350X, but for use with other drives. It is compatible with all Compumotor Step and Direction drives (including the Dynaserv) and is suitable for less demanding motion control applications such as:

❒ Rotary Indexing
❒ Step and Repeat
❒ Linear Slide Positioning

The OEM010 uses the same software commands as the OEM750X, OEM650X and OEM350X.

Features

The OEM010 requires an external 5VDC power supply. The OEM010 provides the following features:

❒ Microprocessor-controlled microstepping provides smooth operation over a wide range of speeds

❒ 3 user-defined inputs/2 user-defined outputs

❒ Step, direction, and shutdown outputs are compatible with all Compumotor drives

❒ Compatible with optical encoders for position verification

❒ RS-232C communication for programming or direct operation

❒ Can daisy chain up to 8 units via hardware/up to 255 in software

❒ 1.5 MHz step output

❒ 3 Inputs for remote sequence selection

❒ Can store 7 programmed sequences in 2k of BBRAM memory

❒ Dedicated Home, CW, CCW limit inputs

❒ Flat or side panel mountable

A2 Installation

Section Objectives

The information in this section will enable you to:

❒ Verify that each component of your system has been delivered safely and completely

❒ Become familiar with components and their interrelationships

❒ Mount unit within recommended thermal specifications

OEM010 Ship kit

Inspect the OEM010 upon receipt for obvious damage to its shipping container. Report any such damage to the shipping company. Parker Compumotor cannot be held responsible for damage incurred in shipment. You should receive an indexer and a software reference guide. Compare your order with the units shipped.

Part	*Part Number*
Indexer	OEM010
OEM Series Software Ref. Guide	88-013785-01

OEM010 Quick Test

1. Connect the OEM010 to a suitable Step and Dir input drive or monitor the Step and Dir output on an oscilloscope.

2. Connect the OEM010 to an RS-232C communications device (i.e., computer, PLC, etc.). The OEM010's communication parameters are listed below:
   - ❑ Baud Rate: 9600
   - ❑ Data Bits: 8
   - ❑ Stop Bit: 1
   - ❑ Parity: None

Terminals should be set for full duplex mode. Reference the **XONOFF** command for handshaking support.

CAUTION

RS-232C signals are not on pins 2, 3, and 7 on the 25-pin D connector.

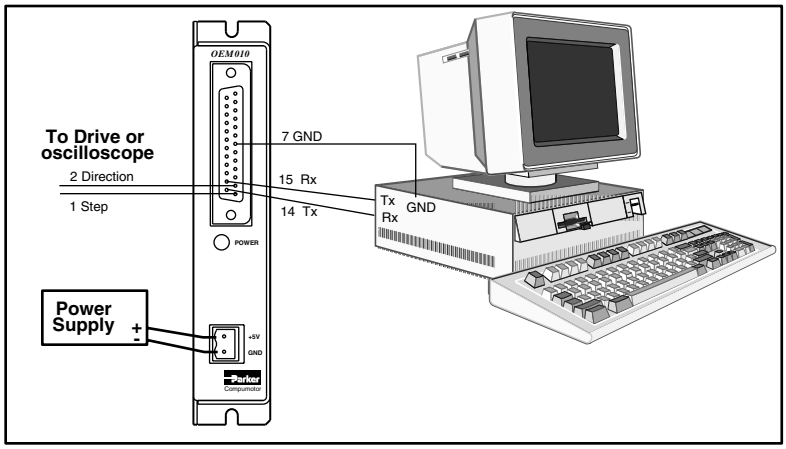
3. Apply power. The OEM's green power LED should be on.

This test assumes that your indexer's motor resolution is set to 25,000 steps/rev. This is the default motor resolution setting for the OEM010.

4. Enter and run the following command sequence to test the system.

Command	Description
MN	Sets unit to Normal mode
LD3	Disables CW & CCW Limits
A1Ø	Set acceleration to 10 rps^2
V1Ø	Set velocity to 10 rps
D25ØØØ	Set move distance to 1 CW revolution
G	Initiate move (Go)
H	Reverse move direction (CCW)
G	Initiate move (Go)

5. After verifying that the motor moves CW and CCW, turn off power.



OEM010 Test Configuration

OEM010 Mounting

If you mount the OEM010 in an enclosure, observe the following guidelines:

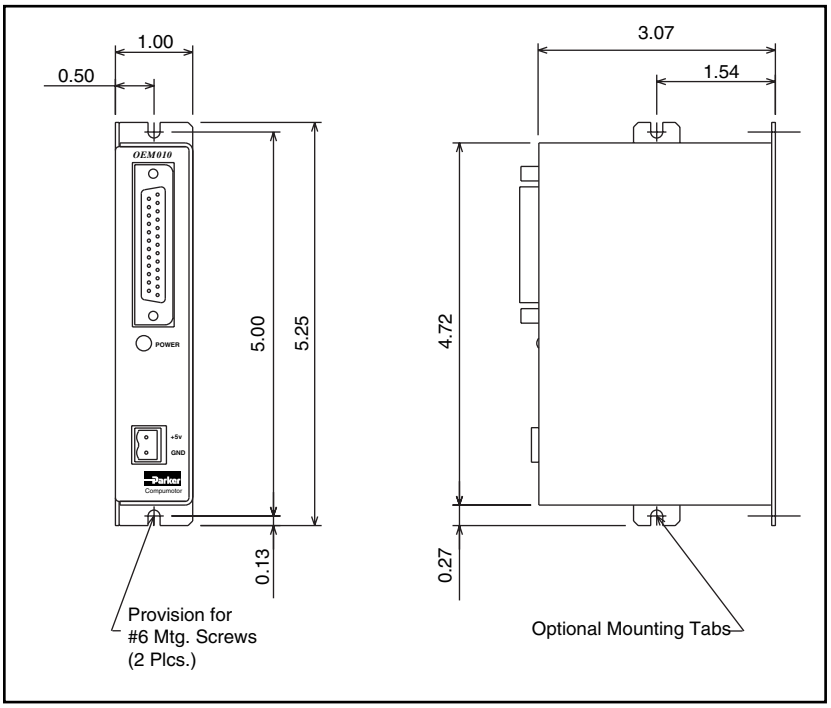
- ❏ Do not mount large, heat-producing equipment directly beneath the OEM010.
- ❏ Do not mount the OEM010 directly above a drive (the drive produces more heat than an indexer).

Refer to the subsequent instructions and diagrams in this section for specific mounting information.

The OEM010 produces almost no heat and can be mounted almost anywhere. The OEM010 is not water-proof, dust-proof, or splash proof, so please provide suitable indexer protection.

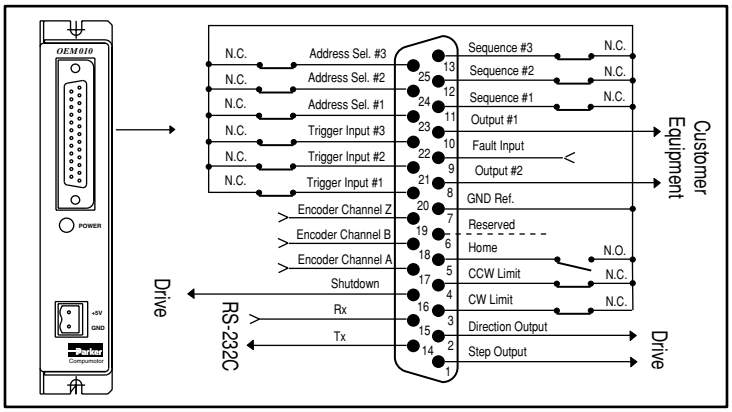
OEM010 Dimensions

The OEM Indexer mounting is designed to minimize panel area or footprint (refer to the figure below).



OEM010 Mounting Dimensions

OEM010 Inputs and Outputs



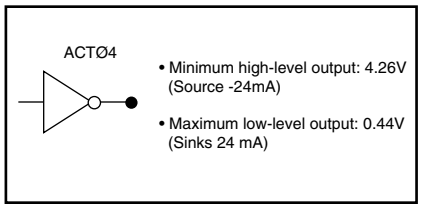
OEM010 Inputs & Output Schematic

CAUTION

I/O is not OPTO isolated, I/O GND is common to GND.

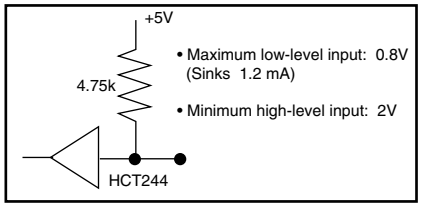
Step (Signal 1) & Direction (Signal 2) Outputs

The OEM010 produces a step and direction output that is compatible with all step and dir input drives. The Direction output's default state is logic high. The Step output's default state is a high, pulsing low output. The figure represents a typical configuration of this output. See also the CMDDIR command.

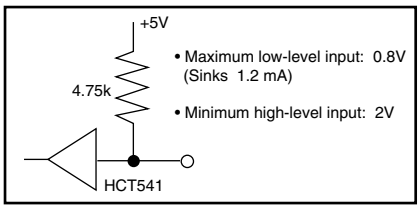


CW (Signal 3) & CCW (Signal 4) Limit Inputs

The OEM010 has two dedicated hardware end-of-travel limits (CCW and CW). When you power up the OEM010, these inputs are enabled (high). To test the OEM010 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 limits' status. The figure represents a typical configuration of these inputs. Minimum pulse width 1 ms.

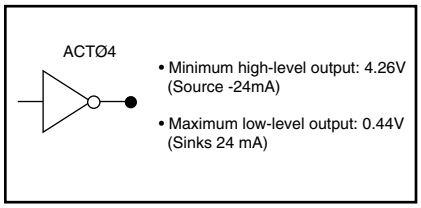


Home Position Input (Signal 5)



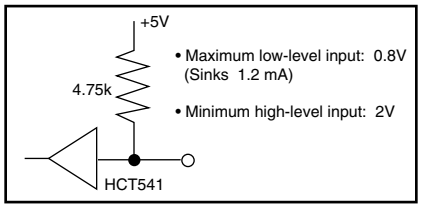
The OEM010 has one dedicated home input. The Home Limit input allows you to establish a home reference input. This input is not active during power-up. Refer to the Go Home (**GH**) command for more information on setting up and using this function. Minimum pulse width is 1 ms. The figure represents a typical configuration.

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



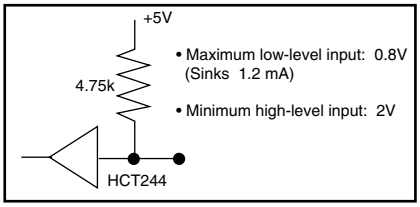
The OEM010 has two dedicated programmable outputs. They may be used to signal peripheral devices upon the start or completion of a move. The default state for Outputs #1 and #2 is logic low. Refer to the Output (**O**) command for information on using these outputs.

Dedicated Fault Input (Signal 9)



The OEM010 has one dedicated fault input**.** Input status can be checked with the **IS** or **R** command. An active low on the input will perform an immediate kill of step pulse generation. The figure represents a typical configuration of the input.

Sequence Inputs #1 - #3 (Signals 11 - 13)



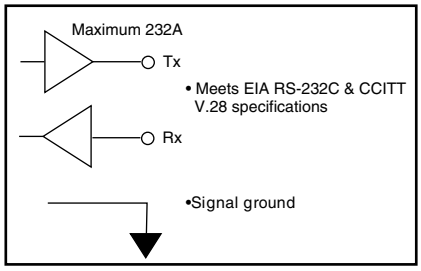
The OEM010 has three dedicated sequence inputs that allow you to control seven different sequences. During power-up, the inputs are pulled up internally, which activates ***power-up sequence #7***. Sequence #Ø is not a valid sequence. Sequences are executed remotely by using one of the following logic patterns in conjunction with the **XP** command.

Sequence #	**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

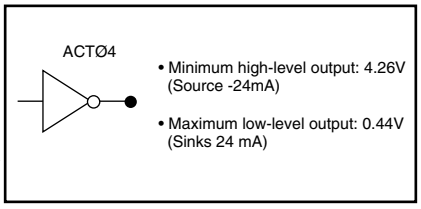
RS-232C—Tx (Signal 14), Rx (Signal 15), and Ground (Signal 7)

The OEM010 uses RS-232C as its communication medium. This indexer does not support handshaking. A typical three-wire (Rx, Tx, and Signal Ground) configuration is used. The figure represents a typical RS-232C configuration.



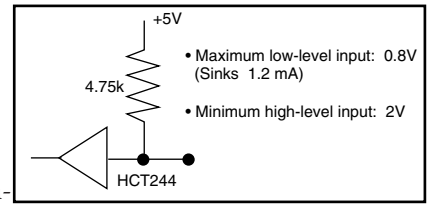
Shutdown Output (Signal 16)

The OEM produces a Shutdown output that is used to remotely disable a drive. This function is controlled by the **ST command.** The Shutdown output's default state is logic high: output is high when the motor is <u>not</u> shutdown (STØ).



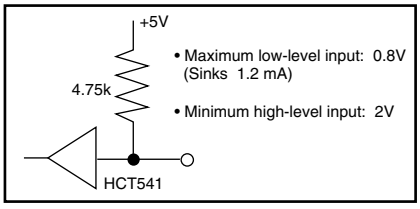
Encoder Inputs A,B,Z (Signals 17 - 19)

The OEM650X has three dedicated inputs for use with a single ended incremental encoder. With differential encoders, leave A-, B-, and Z- isolated and tie the encoder GND to pin 7 (GND). These inputs in conjunction with the **FS** commands will determine the encoder functionality.



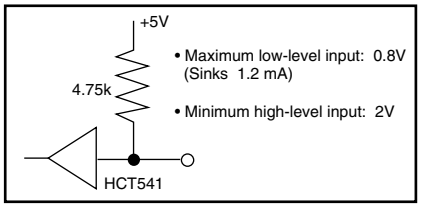
Trigger Inputs #1 - #3 (Signals 20 - 22)

The OEM010 has three dedicated Trigger inputs. These inputs are pulled up internally. These inputs are used with the Trigger (**TR**) command to control the OEM010's trigger function. The figure represents a typical configuration of these inputs. Minimum pulse width is 1 ms.



Address Signals #1 - #3 (Signals 23 - 25)

The OEM010 has three dedicated address inputs that allow you to specify a unique address for each OEM010 in your configuration. Units may be assigned a valid 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.

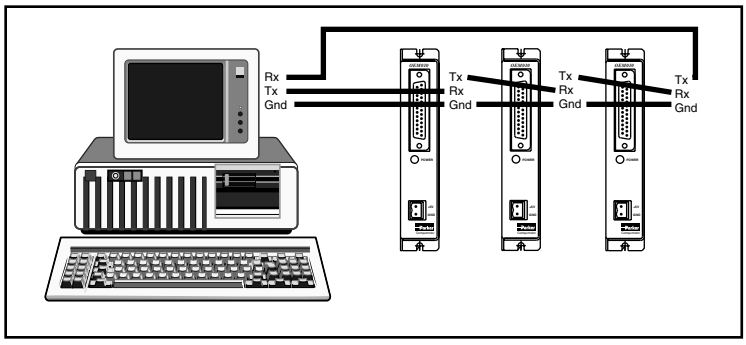


Address #	1	2	3	4	5	6	7	8
Address #1	Ø	1	Ø	1	Ø	1	Ø	1
Address #2	Ø	Ø	1	1	Ø	Ø	1	1
Address #3	Ø	Ø	Ø	Ø	1	1	1	1

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

Daisy Chaining

You may daisy chain up to 8 OEM010s. 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 OEM010. Refer to the figure below for OEM010 daisy chain wiring.



Daisy Chain wiring

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. 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 OEM010, 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.

A3 Specifications

Section Objectives

The information in this section will enable you to:

❒ Verify and identify product specifications

Parameter	Value
Performance	
Stepping accuracy	±0 steps from preset total
Velocity accuracy	±0.02% of maximum rate 0.01 rev/sec
Velocity repeatability	±0.02% of maximum rate
Power	
	+5 VDC ±5%, 500 mA maximum
Inputs	
RS232 Interface	3-wire implementation(Tx, Rx, Gnd)
Control lines	0.80 VDC maximum low level (sinks 1.2 mA, Minimum pulse width: 1 millisecond
Sequence select	Pulse width set by **SN** command
Encoder	
	Single ended. Accepts two phase quadrature incremental encoders 5 volt TTL compatible
Outputs	
Step , Dir , Shutdown	Active high: <0.44 VDC max low, >4.26 VDC Max high 1.5 MHz maximum output frequency
Control Outputs	4.26 VDC Minimum high (source -24mA) 0.44 VDC maximum low (sinks 24 mA)
Environmental	
Operating	0°C to 50°C (32°F – 122°F)
Storage	-30°C to 85°C (-22°F – 185°F)
Humidity	0 to 95% non condensing
Contaminants	The OEM010 is not water-proof, oil-proof, or dust-proof.

A4 Troubleshooting

Section Objectives

The information in this section will enable you to:

❑ Maintain the system to ensure smooth, efficient operation
❑ Isolate and resolve system problems

Reducing Electrical Noise

For detailed information on reducing electrical noise in your system, refer to the current Compumotor Catalog.

Problem Isolation

When your system does not function properly (or as you expect it to operate), the first thing that you must do is identify and isolate the problem. When you accomplish this, you can effectively begin to resolve and eradicate the problem.

The first step is to isolate each system component and ensure that each component functions properly when it is run independently. You may have to dismantle your system and put it back together piece by piece to detect the problem. If you have additional units available, you may want to use them to replace existing components in your system to help identify the source of the problem.

Determine if the problem is mechanical, electrical, or software-related. Can you repeat or re-create the problem? Random events may appear to be related, but they may not be contributing factors to your problem. Investigate the events that occur before the subsequent system problem.

You may be experiencing more than one problem. You must solve one problem at a time. Document all testing and problem isolation procedures. You may need to review and consult these notes later. This will also prevent you from duplicating your testing efforts.

Once you isolate the problem, take the necessary steps to resolve it. Use the solutions in this chapter. If your system's problem persists, call Compumotor at 800-358-9070.

Front Panel LEDs

There is one LED on the front panel of the OEM010. The POWER LED is green and turns on when there is +5V applied to the power input. Should the LED go out at any time, your OEM010 is in need of repair.

RS-232C Problems

Use the following procedure to troubleshoot communication problems that you may have with the OEM010.

1. Be sure the host computer's transmit (Tx) wire is wired to the peripheral's receive (Rx) connection, and the host computer's receive (Rx) wire is wired to the peripheral's transmit (Tx) connection. Switch the receive and transmit wires on either the host or peripheral if the problem persists.

CAUTION

OEM010 Rx, Tx, and GND pin outs are not 2, 3, and 7 like most devices.

2. Confirm that the host and peripheral are configured for the same baud rate, 8 data bits, 1 stop bit, and no parity.

3. Use DC common or signal ground as a reference, not earth ground.

4. Cable lengths should not exceed 50 ft. unless you are using some form of line driver, optical coupler, or shield. As with any control signal, be sure to shield the cable-to-earth ground at one end only.

5. To test the terminal or terminal emulation software and the RS-232C cable for proper three-wire communication, unhook the OEM010 and enter a character. You should not receive an echoed character. If you do, you are in half duplex mode. Connect the host's transmit and receive lines together and send another character. You should receive the echoed character. If not, consult the manufacturer of the host's serial interface for proper pin outs.

6. (Note: This only applies to firmware 92-016638-01.) If the controller echoes back **&** for each byte sent to the controller, a data communication error has occurred. To re-establish communication, see the **%** command.

7. (Note: This only applies to firmware 92-016638-01.) To extend cable length and/or improve noise immunity, the RS-485 option is available as a custom product request. Contact the Custom Products department for details.

8. (Note: This only applies to firmware 92-016638-01.) If commands are being lost intermittently while data streaming, reference the **XONOFF** command for using software flow control.

Software Debugging Tips

This section offers helpful tips for debugging programs and understanding errors and fault conditions. The OEM010 has several tools that you can use to debug a problem in the system. The software tools are listed below:

RA—Limit Switch Status Report
R—Report Status
IS—Input Status Report
BS—Buffer Status Report
B—Buffer Status Report

Homing Diagrams

The following diagrams are examples of the many possible homing set-ups. Your parameters may vary and the results may vary slightly depending on your settings.

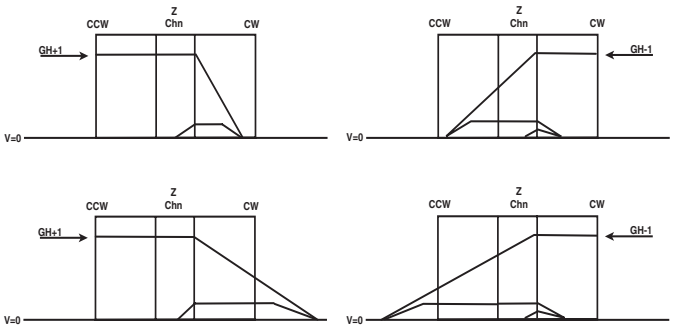
The CW side of the home pulse is the side closest to the CW limit. The CCW side of the home pulse is the side closest to the CCW limit.

The long pulse diagrams are indicative of situations where the motor decelerates while remaining inside the home pulse width due to the rapid homing deceleration or a very wide home pulse. The short pulse diagrams are indicative of situations where the motor decelerates through the home pulse width due to slow deceleration or a very narrow pulse width.

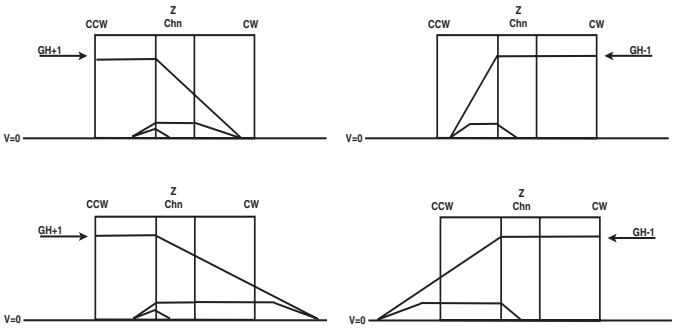
If an end-of-travel limit is hit during the initial homing, refer to the homing diagram for the opposite direction of travel.

The diagrams are drawn as a general guide. Velocity levels and slopes are drawn to indicate the general move profile the motor will make during the go home move. The vertical axis is velocity and the horizontal axis the position in relation to the home input transitions. Some lines are drawn as closely as possible together to indicate identical velocities, yet remain discernible.

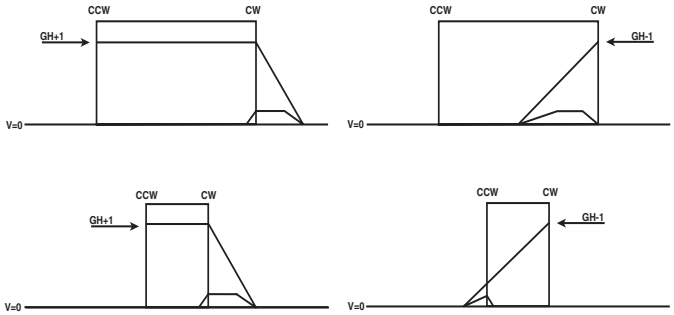
OSB1
OSD1
OSHØ
FSB1



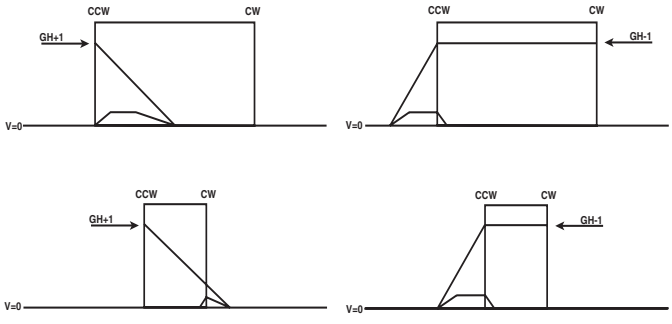
OSB1
OSD1
OSH1
FSB1



OSB1
OSDØ
OSHØ



OSB1
OSDØ
OSH1



Returning the System

If your OEM010 system is faulty, you must return the drive and motor for replacement or repair. A failed drive can damage motors. If you must return your OEM010 to effect repairs or upgrades, use the following steps:

① Get the serial number and the model number of the defective unit(s), and a purchase order number to cover repair costs in the event the unit is determined by Parker Compumotor to be out of warranty.

② Before you ship the drive to Parker Compumotor, have someone from your organization with a technical understanding of the OEM010 and its application include answers to the following questions: What is the extent of the failure/reason for return?

- How long did it operate?
- How many units are still working?
- How many units failed?
- What was happening when the unit failed (i.e., installing the unit, cycling power, starting other equipment, etc)?
- How was the product configured (in detail)?
- What, if any, cables were modified and how?
- With what equipment is the unit interfaced?
- What was the application?
- What was the system sizing (speed, acceleration, duty cycle, inertia, torque, friction, etc.)?
- What was the system environment (temperature, enclosure, spacing, unit orientation, contaminants, etc.)?
- What upgrades are required (hardware, software, user guide)?

③ Call Parker Compumotor's Applications Engineering Department [(800) 358-9070] for a Return Material Authorization (RMA) number. Returned products cannot be accepted without an RMA number.

④ Ship the unit to: Parker Compumotor Corporation
5500 Business Park Drive, Suite D
Rohnert Park, CA 94928

This page left blank intentionally.

Index

Summary of Commands

A—Acceleration
B—Buffer Status
BS—Buffer Size Status
C—Continue
CG—Correction Gain
CMDDIR—Commanded Direction Polarity
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
GH—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
MR—Motor Resolution
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
Q1—Enter Velocity Profiling Mode
QØ—Exit 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
SSD—Add Device ID Prefix
SSE—Enable/Disable Communication Error Checking
SSG—Clear/Save the Command Buffer on Limit
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
XONOFF—Enable/Disable XON/XOFF
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
%—Reset Communication