

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 | ⑥ Valid Software Version: A |
| ③ Syntax: <a>An | ⑦ Units: revs/sec ² |
| ④ Range: n = 0.01-999.00 | ⑧ Default Value: A = 100 |
| ⑤ Attributes: Buffered,
Savable in Sequence | 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.

COMMANDS • OEM SOFTWARE

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/sec²
- 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.

<u>Command</u>	<u>Description</u>
A10	Sets acceleration to 10 revs/sec ²
V10	Sets velocity to 10 revs/sec
D2000	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.

COMMANDS • OEM SOFTWARE

*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.

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

BS—Buffer Size Status

<input type="checkbox"/> Command Type: Status	<input type="checkbox"/> Valid Software Version: A
<input type="checkbox"/> Syntax: aBS	<input type="checkbox"/> Units: N/A
<input type="checkbox"/> Range: N/A	<input type="checkbox"/> Default Value: N/A
<input type="checkbox"/> Attributes: Immediate, Never Saved	<input type="checkbox"/> Response to aBS is *n
	<input type="checkbox"/> 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.

<u>Command</u>	<u>Response</u>
1BS	*122 (122 bytes available in the buffer)

C—Continue

<input type="checkbox"/> Command Type: Motion	<input type="checkbox"/> Valid Software Version: A
<input type="checkbox"/> Syntax: <a>C	<input type="checkbox"/> Units: N/A
<input type="checkbox"/> Range: N/A	<input type="checkbox"/> Default Value: N/A
<input type="checkbox"/> Attributes: Immediate, Never Saved	<input type="checkbox"/> 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.

<u>Command</u>	<u>Description</u>
MC	Sets move to continuous mode
A10	Sets acceleration to 10 revs/sec ²
V10	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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>CGn | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: n = 1 - 8 | <input type="checkbox"/> Default Value: 8 |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> Response to aCG is *CGn |
| | <input type="checkbox"/> 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 $n/8 \times 100\%$. 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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-up | <input type="checkbox"/> Valid Software Version: E |
| <input type="checkbox"/> Syntax: <a>CMDDIRn | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: 0, 1 | <input type="checkbox"/> Default Value: 1 |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> 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.

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

COMMANDS • OEM SOFTWARE

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.

<u>Command</u>	<u>Response</u>
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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>CR | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered | <input type="checkbox"/> Response to aCR is *[cr] |
| <input type="checkbox"/> Savable in Sequence | <input type="checkbox"/> 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.

<u>Command</u>	<u>Description</u>
MN	Sets mode to preset mode
A50	Sets acceleration to 50 revs/sec ²
V5	Sets Velocity to 5 revs/sec
D5000	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 D0 and G in Incremental preset mode will cause the W3 command to report back with *00000000.

<u>Command</u>	<u>Description</u>
MN	Sets indexer to Normal mode
MPI	Sets indexer to Incremental Position mode
A10	Sets acceleration to 10 revs/sec ²
V10	Sets velocity to 10 revs/sec
D25000	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

COMMANDS • OEM SOFTWARE

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
DB100	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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>DWn | <input type="checkbox"/> Units: n = steps |
| <input type="checkbox"/> Range: n = 0 - 999,999,999 | <input type="checkbox"/> Default Value: 0 |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> Response to aDW is *DWn |
| | <input type="checkbox"/> 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
DW100	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 than 100 motor steps.
1DW	Reports Deadband Window (*DW100)

E—Enable Communications

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

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.

<u>Command</u>	<u>Description</u>
F	Disables all units (axes) on the communications interface
1E	Enables serial interface on Device 1
4E	Enables serial interface on Device 4
A10	Set acceleration to 10 revs/sec ²
V5	Set velocity to 5 revs/sec
D5000	Sets distance to 5000 steps
G	Executes the move (Go—only axes 1 & 4 will move)

ER—Encoder Resolution

- Command Type: Set-up
- Valid Software Version: A
- Syntax: <a>ERn
- Units: n = steps/rev
- Range: n = 1 - 50,000
- Default Value: 4000
- Attributes: Buffered,
Savable in Sequence
- 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.

<u>Command</u>	<u>Description</u>
MR20000	Sets indexer to control a motor of 20000 steps/rev
ER4000	Sets encoder resolution to 4000 encoder steps per 1 motor revolution
1ER	Reports Encoder Resolution (*ER4000)

F—Disable Communications

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

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.

<u>Command</u>	<u>Description</u>
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
- Valid Software Version: A
- Syntax: <a>FS
- Units: n = N/A
- Range: N/A
- Default Value: None
- Attributes: Buffered,
Savable in Sequence
- 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

1FS

Response

*11000000 (The indexer is in absolute encoder step mode with all other FS functions turned OFF.)

FSA—Set Indexer to Incremental/Absolute Mode

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

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

COMMANDS • OEM SOFTWARE

FSA0 = Incremental mode

FSA1 = Absolute mode

In Incremental mode (**FSA0**), 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.

<u>Command</u>	<u>Description</u>
FSA1	Sets Indexer to Absolute mode
PZ	Resets the absolute position counter to zero
A10	Sets acceleration to 10 rev/sec ²
V5	Sets velocity to 5 rev/sec
D25600	Move motor to absolute position 25,600
G	Executes the move (Go)
D64000	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

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

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

FSB0 = 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
ER4000	Set up encoder where 4,000 encoder pulses (1,000 lines) are produced per 1 motor rev.
FSB1	Set moves to encoder step mode
A10	Set acceleration to 10 rev/sec ²
V5	Set velocity to 5 rev/sec
D4000	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
- Valid Software Version: A
- Syntax: <a>FSCn
- Units: N/A
- Range: n = 0, 1
- Default Value: 0
- Attributes: Buffered, Savable in Sequence
- See Also: FSB, ER, DB

FSC1 = Enable Position Maintenance

FSC0 = 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

COMMANDS • OEM SOFTWARE

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
ER2000	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 FSD0 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
DW100	Set backlash value to 100 steps.
ER2000	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

FSE0 = 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 your system that a stall has occurred. This command will be valid only if Stall Detect (**FSH1**) and encoder step mode (**FSB1**) have been enabled.

Output number 1 is unaffected by a stall when **FSE0** is entered.

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

Command	Description
ER2000	Set encoder resolution to 2,000 steps/rev.
DW200	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
- Valid Software Version: A
- Syntax: <a>FSFn
- Units: N/A
- Range: n = 0, 1
- Default Value: 0
- Attributes: Buffered,
Savable in Sequence
- See Also: TR, FSE, A

FSF0 = 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 **FSF0** will turn this feature off.

Command	Description
----------------	--------------------

COMMANDS • OEM SOFTWARE

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

FSG—Turn on Output 2 when within Dead Band

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

FSG0— 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
ER4000	Set encoder resolution to 4,000 steps/rev.
DB50	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
- Valid Software Version: A
- Syntax: <a>FSHn
- Units: N/A
- Range: n = 0, 1
- Default Value: 0
- Attributes: Buffered, Savable in Sequence
- See Also: FS commands, DW, ER

FSH0 = 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
DW1000	Set dead band window to 1,000 steps
ER2000	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

- | | |
|--|---|
| <input type="checkbox"/> Command Type: Motion | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>G | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: None |
| <input type="checkbox"/> Attributes: Buffered
Savable in Sequence | <input type="checkbox"/> 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.

COMMANDS • OEM SOFTWARE

<u>Command</u>	<u>Description</u>
MN	Sets Normal mode (preset)
A5	Sets acceleration to 5 revs/sec ²
V10	Sets velocity to 10 revs/sec
D2000	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/sec² (Total distance moved = 4,000 steps).

GH—Go Home

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

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

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

COMMANDS • OEM SOFTWARE

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
D25000	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

- | | |
|--|---|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aIS | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate | <input type="checkbox"/> See Also: D |
| Never Saved | <input type="checkbox"/> Response to aIS is *nnnnnnnnnn |

This command reports the status of all hardware inputs. The response is 10 ASCII digits (0 or 1) plus a device address (1 - 8), corresponding to the following I/O bits:

- 1—Trigger bit 1 (0 = Low, 1 = High)
- 2—Trigger bit 2 (0 = Low, 1 = High)
- 3—Trigger bit 3 (0 = Low, 1 = High)
- 4—Home enable (0 = Low, 1 = High)
- 5—FLT (0 = Faulted, 1 = Normal)
- 6—CCW limit (0 = Low, 1 = High)
- 7—CW limit (0 = Low, 1 = High)
- 8—Sequence Select 1 (0 = Low, 1 = High)
- 9—Sequence Select 2 (0 = Low, 1 = High)
- 10—Sequence Select 3 (0 = 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.

<u>Command</u>	<u>Response</u>
2IS	*00010000002 (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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Motion | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>K | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate
Never Saved | <input type="checkbox"/> 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.

<u>Command</u>	<u>Description</u>
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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>Ln | <input type="checkbox"/> Units: number of loops |
| <input type="checkbox"/> Range: n = 0 - 65,535 | <input type="checkbox"/> Default Value: None |
| <input type="checkbox"/> Attributes: Buffered
Savable in Sequence | <input type="checkbox"/> 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

COMMANDS • OEM SOFTWARE

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.

<u>Command</u>	<u>Description</u>
L5	Loop 5 times
A5	Sets acceleration to 5 revs/sec ²
V10	Sets velocity to 10 revs/sec
D10000	Sets distance to 10,000 steps
G	Executes the move (Go)
N	Repeats 10,000-step move five times

LD—Limit Disable

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

The Limit Disable (**LD**) command allows you to enable/disable the end-of-travel limit switch protection. The **LD0** 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 = 0 (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.

<u>Command</u>	<u>Description</u>
1LD0	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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>LF | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered | <input type="checkbox"/> See Also: CR |
| Savable in Sequence | <input type="checkbox"/> 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
D15000	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

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

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.

COMMANDS • OEM SOFTWARE

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)
T10	Move at 5 revs/sec for 10 seconds
V7	Set velocity to 7 revs/sec
G	Change velocity to 7 revs/sec
T10	Move at 7 revs/sec for 10 seconds
V0	Set velocity to 0 rps (stop)
G	Executes the V0 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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Motion | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>MN | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Status: Active |
| <input type="checkbox"/> Attributes: Buffered
Savable in Sequence | <input type="checkbox"/> 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
D1000	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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>MPA | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Status: Inactive |
| <input type="checkbox"/> Attributes: Buffered
Savable in Sequence | <input type="checkbox"/> 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
V10	Sets velocity to 10 revs/sec
D25000	Sets destination to absolute position 25,000
G	Motor will move to absolute position 25,000
D10000	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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>MPI | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Status: Active |
| <input type="checkbox"/> Attributes: Buffered
Savable in Sequence | <input type="checkbox"/> 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.

COMMANDS • OEM SOFTWARE

Command	Description
MN	Set positioning mode normal (preset)
MPI	Set positioning mode incremental
A5	Sets acceleration to 5 revs/sec ²
V10	Sets velocity to 10 revs/sec
D10,000	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
- Valid Software Version: A
- Syntax: <a>MRn
- Units: steps/rev
- Range: n = 200 - 1,024,000
- Default Value: 25,000
- Attributes: Buffered
- See Also: A, V
- Savable in Sequence

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
MR25000	Sets indexer to control a motor of 25000 steps/rev
A5	Set acceleration to 5 revs/sec ²
V10	Set velocity to 10 revs/sec
D25000	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/sec² 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/sec² 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 (**IMR**), the indexer will respond with the current distance value (***MRn**).

N—End of Loop

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

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.

<u>Command</u>	<u>Description</u>
MN	Sets move to Normal mode
A5	Sets acceleration to 5 revs/sec ²
V5	Sets velocity to 5 revs/sec
D10000	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
- Valid Software Version: A
- Syntax: <a>Onn
- Units: on, off, or unchanged
- Range: Ø, 1 or X (See Below)
- Default Value: ØØ
- Attributes: Buffered
- See Also: OS, S, TR, TS
- Savable in Sequence

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

COMMANDS • OEM SOFTWARE

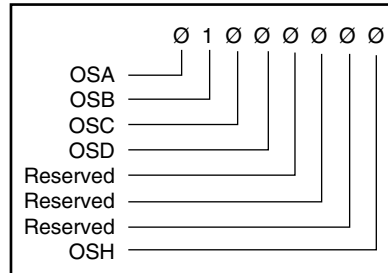
n=X = Leaves output bits unchanged

<u>Command</u>	<u>Description</u>
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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>OS | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> See Also: OS(A-H) |
| | <input type="checkbox"/> 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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>OSAn | <input type="checkbox"/> Units: See Below |
| <input type="checkbox"/> Range: n = Ø, 1 | <input type="checkbox"/> Default Value: Ø |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> 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.

<u>Command</u>	<u>Description</u>
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

<input type="checkbox"/> Command Type: Set-Up	<input type="checkbox"/> Valid Software Version: A
<input type="checkbox"/> Syntax: <a>OSBn	<input type="checkbox"/> Units: See Below
<input type="checkbox"/> Range: n = Ø, 1	<input type="checkbox"/> Default Value: 1
<input type="checkbox"/> Attributes: Buffered, Savable in Sequence	<input type="checkbox"/> 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.

<u>Command</u>	<u>Description</u>
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 = 0, 1
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: See Below
- Default Value: 0
- See Also: GH, OSB, OSH

OSC0: Active state of home input is n = 0 (closed)

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

OSC0 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.

<u>Command</u>	<u>Description</u>
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

OSD0 = 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.

<u>Command</u>	<u>Description</u>
OSD1	Recognizes Z channel as final home reference

OSH—Reference Edge of Home Switch

- Command Type: Set-Up
- Syntax: <a>OSHn
- Range: n = 0, 1
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: See Below
- Default Value: 0
- See Also: GH, OSB, OSC

OSH0:

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.

<u>Command</u>	<u>Description</u>
OSB1	Sets back up to home switch active
OSC0	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 Valid Software Version: A
- Syntax: aPR Units: N/A
- Range: N/A Default Value: N/A
- Attributes: Buffered, See Also: D, MPA, MPI, MN, PZ
 Savable in Sequence 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 (FSB0), 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.

<u>Command</u>	<u>Description</u>
PZ	Resets the absolute counter to zero
LD3	Disable both CW & CCW limits ²
A10	Set Acceleration to 10 revs/sec ²
V5	Set velocity to 5 revs/sec
D2000	Set move distance to 2000 steps

COMMANDS • OEM SOFTWARE

G Executes the move (Go)
1PR Request absolute position report. Response should be *+00000002000

PS—Pause

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

<u>Command</u>	<u>Description</u>
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
D25000	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
- Valid Software Version: A
- Syntax: aPX
- Units: N/A
- Range: N/A
- Default Value: N/A
- Attributes: Buffered, Savable in Sequence
- 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/sec ²
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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>PZ | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Never Saved | <input type="checkbox"/> 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
- Valid Software Version: A
- Syntax: “x
- Units: N/A
- Range: x = up to 17 ASCII characters
- Default Value: N/A
- Attributes: Buffered, Savable in Sequence
- 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
D2000	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

Q0—Exit Velocity Profiling Mode

- Command Type: Set-Up
- Valid Software Version: A
- Syntax: <a>Q0
- Units: N/A
- Range: N/A
- Default Value: N/A
- Attributes: Immediate, Never Saved
- See Also: Q1, RM

The **g0** command exits the Velocity Profiling mode. The motor will stop when **g0** is issued. Entering this command will cause the OEM Indexer to enter Normal mode (**MN**).

Q1—Enter Velocity Profiling Mode

- Command Type: Set-Up
- Valid Software Version: A
- Syntax: <a>Q1
- Units: N/A
- Range: N/A
- Default Value: N/A
- Attributes: Immediate, Never Saved
- See Also: Q0, RM

q1 activates Velocity Profiling mode. Subsequent **RM** commands will immediately change motor velocity. **g0** exits this mode.

<u>Command</u>	<u>Description</u>
Q1	Enter Velocity Streaming mode
RM00A0	Accelerate to 0.25 revs/sec ²
RM0140	Accelerate to 0.5 revs/sec ²
RM0280	Accelerate to 1 revs/sec ²
RM0500	Accelerate to 2 revs/sec ²
RM0280	Decelerate to 1 revs/sec ²
RM0140	Decelerate to 0.5 revs/sec ²
RM00A0	Decelerate to 0.25 revs/sec ²
RM0000	Decelerate to 0 revs/sec ²
Q0	Exit Velocity Streaming mode

R—Request Indexer Status

<input type="checkbox"/> Command Type: Status	<input type="checkbox"/> Valid Software Version: A
<input type="checkbox"/> Syntax: aR	<input type="checkbox"/> Units: N/A
<input type="checkbox"/> Range: N/A	<input type="checkbox"/> Default Value: N/A
<input type="checkbox"/> Attributes: Immediate, Never Saved	<input type="checkbox"/> See Also: RA, RB, RC, XSR, XSS
	<input type="checkbox"/> 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:

<u>Character</u>	<u>Definition</u>
*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

COMMANDS • OEM SOFTWARE

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

R

Response

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

RA—Limit Switch Status Report

- Command Type: Status
- Valid Software Version: A
- Syntax: aRA
- Units: N/A
- Range: N/A
- Default Value: N/A
- Attributes: Immediate, Never Saved
- 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**.

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

RB—Loop, Pause, Shutdown, Trigger Status Request

- | | |
|---|---|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aRB | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate, Never Saved | <input type="checkbox"/> See Also: L, PS, R, RA, ST, TR |
| | <input type="checkbox"/> 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

COMMANDS • OEM SOFTWARE

*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 **1RB**

Response

***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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aRC | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> Response to aRC IS *x |
| | <input type="checkbox"/> 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
IRC	*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
- Valid Software Version: A
- Syntax: <a>RMn
- Units: revs/sec
- Range: n = 000 - FFF
- Default Value: None
- Attributes: Immediate, Never Saved
- See Also: D, H, Q0, Q1

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:

$(revs/sec) \cdot (resolution\ constant) = \text{decimal \# for velocity value to be rounded off to the closest whole number.}$

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

COMMANDS • OEM SOFTWARE

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

- | | |
|---|---|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aRV | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> See Also:
<input type="checkbox"/> 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.

<u>Command</u>	<u>Response</u>
1RV	*92-Ø16678-Ø1E

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

S—Stop

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Motion | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>S | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate,
Never Saved | <input type="checkbox"/> 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.

COMMANDS • OEM SOFTWARE

<u>Command</u>	<u>Description</u>
MC	Sets move in continuous mode
A1	Sets acceleration to 1 revs/sec ²
V10	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

<input type="checkbox"/> Command Type: Set-Up	<input type="checkbox"/> Valid Software Version: A
<input type="checkbox"/> Syntax: <a>SNn	<input type="checkbox"/> Units: n = mS
<input type="checkbox"/> Range: 1 - 1000	<input type="checkbox"/> Default Value: 50
<input type="checkbox"/> Attributes: Buffered, Savable in Sequence	<input type="checkbox"/> 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.

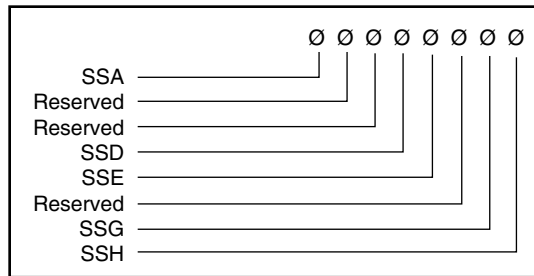
This command allows you to select the best possible trade-off between noise immunity and speed for a given application. If you make your scan time too short, the OEM Indexer may respond to an electrical glitch. If you issue the Scan command with only a device address (**1SN**), the indexer will respond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

<u>Command</u>	<u>Description</u>
SN10	Sets scan time of sequence select inputs to 10 ms

SS—Software Switch Function Status

<input type="checkbox"/> Command Type: Status	<input type="checkbox"/> Valid Software Version: A
<input type="checkbox"/> Syntax: aSS	<input type="checkbox"/> Units: N/A
<input type="checkbox"/> Range: N/A	<input type="checkbox"/> Default Value: N/A
<input type="checkbox"/> Attributes: Buffered, Savable in Sequence	<input type="checkbox"/> See Also: SSA, SSG, SSH
	<input type="checkbox"/> 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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>SSAn | <input type="checkbox"/> Units: See Below |
| <input type="checkbox"/> Range: n = Ø, 1 | <input type="checkbox"/> Default Value: Ø |
| <input type="checkbox"/> Attributes: Buffered,
Automatically Saved | <input type="checkbox"/> 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.*

<u>Command</u>	<u>Description</u>
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
- Valid Software Version: E
- Syntax: <a>SSDn
- Units: N/A
- Range: n = Ø, 1
- Default Value: Ø
- Attributes: Buffered, Saveable in Sequence
- 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:

<u>Command</u>	<u>Description</u>
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:

<u>Command (SSDØ)</u>	<u>Response</u>
23V	*V1
23SS	*00000000
"Hello	Hello

<u>Command (SSD1)</u>	<u>Response</u>
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
- Valid Software Version: A
- Syntax: <a>SSHn
- Units: See Below
- Range: n = 0, 1
- Default Value: 0
- Attributes: Buffered, Automatically Saved
- See Also: S

SSH0 = Clears command buffer on stop

SSH1 = Saves command buffer on stop

In Normal Operation (**SSH0**) 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.

<u>Command</u>	<u>Description</u>
SSH0	Clears buffer on stop
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D25000	Sets distance to 25,000 steps
L50	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
- Valid Software Version: A
- Syntax: <a>STn
- Units: See Below
- Range: n = 0, 1
- Default Value: 0
- Attributes: Buffered, Savable in Sequence

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.

<u>Command</u>	<u>Description</u>
ST1	Shuts off current to the motor

SV—Servoing Parameter

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>SVn | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: n = 0 - 3 | <input type="checkbox"/> Default Value: None |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> Response to aCG is *n |
| | <input type="checkbox"/> 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.

COMMANDS • OEM SOFTWARE

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

T—Time Delay

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

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
D25000	Sets distance to 25,000 steps
T10	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 Valid Software Version: A
- Syntax: <a>TRnnn Units: See Below
- Range: n = Ø, 1, or X Default Value: None
- Attributes: Buffered, See Also: TS
 Savable in Sequence

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

<u>Command</u>	<u>Description</u>
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.

<u>Command</u>	<u>Response</u>
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

COMMANDS • OEM SOFTWARE

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
- Valid Software Version: A
- Syntax: <a>Vn
- Units: revs/sec
- Range: n = 0.01 - 50.00
- Default Value: 1
- Attributes: Buffered, Savable in Sequence
- 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 (**IV**), 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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aW1 | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate,
Never Saved | <input type="checkbox"/> See Also: PR, W3 |
| | <input type="checkbox"/> 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).

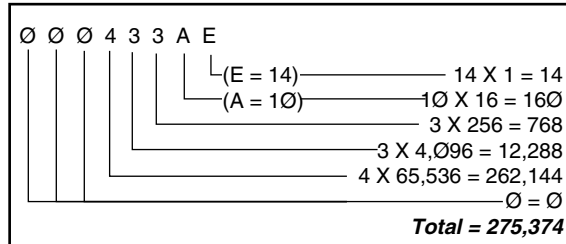
If you use a terminal to communicate with the indexer, the response may not be a printable character. The response must be decoded with a computer. *This command is useful if you want to receive a position report while the motor is moving.*

W3—Hexadecimal Position Report

- | | |
|--|---|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aW3 | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate,
Never Saved | <input type="checkbox"/> See Also: PR, W1 |
| | <input type="checkbox"/> Response to aW3 is *xxxxxxxx |

COMMANDS • OEM SOFTWARE

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 *000433AE. 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 | 10000 hex | = | 65,536 |
| | Result | = | -25,000 |

Command

1W3

Response

*FFFA19C (24,163 steps from start of move)

XC—Sequence Checksum

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aXC | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: None |
| <input type="checkbox"/> Attributes: Buffered,
Never Saved | <input type="checkbox"/> 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 response (000 - 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.

<u>Command</u>	<u>Response</u>
1XC	*149

XD—Sequence Definition

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>XDn | <input type="checkbox"/> Units: Sequences |
| <input type="checkbox"/> Range: n = 1 - 7 | <input type="checkbox"/> Default Value: None |
| <input type="checkbox"/> Attributes: Buffered,
Never Saved | <input type="checkbox"/> 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.

<u>Command</u>	<u>Description</u>
XE1	Erases sequence #1
XD1	Defines sequence #1
MN	Sets to Normal mode
A10	Sets acceleration to 10 revs/sec ²
V5	Sets acceleration to 5 revs/sec
D10000	Sets distance to 10,000 steps
G	Executes the move (Go)
XT	Ends definition of Sequence #1
XR1	Executes Sequence #1

XE—Sequence Erase

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

COMMANDS • OEM SOFTWARE

This command allows you to delete a sequence. The sequence that you specify (n) will be deleted when you issue the command. *Computer 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 = ∅**: 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
A1∅	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D25∅∅∅	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
- Valid Software Version: A
- Syntax: <a>XQn
- Units: Sequences
- Range: $n = ∅, 1$
- Default Value: ∅
- Attributes: Buffered, Savable in Sequence
- See Also: XP

- $n = 1$: Interrupted Run mode is set (on)
- $n = ∅$: 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

COMMANDS • OEM SOFTWARE

then read the sequence select lines and execute the sequence whose number appears there. This paused mode will continue until an **XQ0** command is executed. You may use **S** or **K** command to stop sequence execution. **XQ1** must be the first command entered in the sequence.

<u>Command</u>	<u>Description</u>
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
- Valid Software Version: A
- Syntax: <a>XRn
- Units: Sequence
- Range: n = 1 - 7
- Default Value: 0
- Attributes: Buffered, Savable in Sequence
- 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 GO-SUB). An **XR** command placed within a loop will be ignored.

<u>Command</u>	<u>Description</u>
XE1	Erases sequence #1
XD1	Defines sequence #1
A10	Sets acceleration to 10 revs/sec ²
V5	Sets acceleration to 5 revs/sec
D10000	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
- Valid Software Version: A
- Syntax: <a>XRPn
- Units: Sequence
- Range: n = 1 - 7
- Default Value: 0
- Attributes: Buffered,
Savable in Sequence
- 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.

<u>Command</u>	<u>Description</u>
XE5	Erases Sequence #5
XD5	Defines Sequence #5
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D10000	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
- Valid Software Version: A
- Syntax: aXSD
- Units: N/A
- Range: N/A
- Default Value: N/A
- Attributes: Buffered,
Savable in Sequence
- 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:

COMMANDS • OEM SOFTWARE

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.

<u>Command</u>	<u>Response</u>
1XSD	*1

XSP—Sequence Status Power-up

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aXSP | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Never Saved | <input type="checkbox"/> See Also: XP, XQ, XSR |
| | <input type="checkbox"/> 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.

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

XSR—Sequence Status Run

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aXSR | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate,
Never Saved | <input type="checkbox"/> See Also: XR, XRP |
| | <input type="checkbox"/> 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.

- * \emptyset = Last sequence was successful
- * **1** = In a loop
- * **2** = Invalid sequence
- * **3** = Erased
- * **4** = Bad checksum
- * **5** = Running
- * **6** = Killed,stopped

<u>Command</u>	<u>Response</u>
1XSR	* \emptyset (Sequence ran O.K.)

XSS—Sequence Status

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aXSSn | <input type="checkbox"/> Units: Sequences |
| <input type="checkbox"/> Range: n = 1 - 7 | <input type="checkbox"/> Default Value: None |
| <input type="checkbox"/> Attributes: Buffered,
Never Saved | <input type="checkbox"/> See Also: XD, XE, XT |
| | <input type="checkbox"/> 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.

- \emptyset = Empty
- 1 = Bad Checksum
- 3 = O.K.

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

<u>Command</u>	<u>Response</u>
1XSS1	* \emptyset (Sequence #1 of device 1 is not defined.)

XT—Sequence Termination

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>XT | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Never Saved | <input type="checkbox"/> 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.

COMMANDS • OEM SOFTWARE

<u>Command</u>	<u>Description</u>
XE1	Erases Sequence #1
XD1	Defines Sequence #1
MN	Sets to Normal mode
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)
XT	Ends sequence definition

XU—Upload Sequence

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aXUn | <input type="checkbox"/> Units: Sequences |
| <input type="checkbox"/> Range: n = 1 - 7 | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Never Saved | <input type="checkbox"/> See Also: F, XD, XE, XT |
| | <input type="checkbox"/> 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.

<u>Command</u>	<u>Description</u>
2F	Turns off communication to unit #2
3F	Turns off communication to unit #3
1XU1	Uploads sequence #1 from unit #1

Y—Stop Loop

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>Y | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate,
Never Saved | <input type="checkbox"/> 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

#—Remote Address

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Set-up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>#n | <input type="checkbox"/> Units: |
| <input type="checkbox"/> Range: n=0-255 | <input type="checkbox"/> Default Value: 0 |
| <input type="checkbox"/> Attributes: Immediate,
Automatically Saved | <input type="checkbox"/> 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

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: E |
| <input type="checkbox"/> Syntax: % | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: N/A | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Immediate,
Never Saved | <input type="checkbox"/> 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:

<u>Character</u>	<u>Definition</u>
*	No errors
*Ø	Unit upstream (daisy chained)
*2	Framing error

<u>Command</u>	<u>Response</u>
%	*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	OSH—Reference Edge of Home Switch
B—Buffer Status	PR—Absolute Position Report
BS—Buffer Size Status	PS—Pause
C—Continue	PX—Report Absolute Encoder Position
CG—Correction Gain	PZ—Set Absolute Counter to Zero
CMDDIR—Commanded Direction	“—Quote
Polarity	Q1—Enter Velocity Profiling Mode
CR—Carriage Return	QØ—Exit Velocity Profiling Mode
D—Distance	R—Request Indexer Status
DB—Dead Band	RA—Limit Switch Status Report
DW—Deadband Window	RB—Loop, Pause, Shutdown, Trigger
E—Enable Communications	Status Request
ER—Encoder Resolution	RC—Closed Loop Status
F—Disable Communications	RM—Rate Multiplier in Velocity Streaming
FS—Encoder Functions Report	Mode
FSA—Set Indexer to Incremental/	RV—Revision Level
Absolute Mode	S—Stop
FSB—Set Indexer to Motor/Encoder	SN—Scan
Step Mode	SS—Software Switch Function Status
FSC—Enable/Disable Position	SSA—RS-232C Echo Control
Maintenance	SSD—Add Device ID Prefix
FSD—Stop on Stall	SSE—Enable/Disable Communication
FSE— Turn On Output Number 1 on	Error Checking
Stall	SSG—Clear/Save the Command Buffer on
FSF—Stop Motion on Trigger 3	Limit
FSG—Turn On Output Number 2 when	SSH—Clear/Save Command Buffer on
in Deadband	Stop
FSH—Enable Stall Detect	ST—Shutdown
G—Go	SV—Servoing Parameter
GH—Go Home	T—Time Delay
^H—Delete	TR—Wait For Trigger
H—Set Direction	TS—Trigger Input Status
IS—Input Status	U—Pause and Wait for Continue
K—Kill	V—Velocity
L—Loop	W1—Signed Binary Position Report
LD—Limit Disable	W3—Hexadecimal Position Report
LF—Line Feed	XC—Sequence Checksum
MC—Mode Continuous	XD—Sequence Definition
MN—Mode Normal	XE—Sequence Erase
MPA—Mode Position Absolute	XONOFF—Enable/Disable XON/XOFF
MPI—Mode Position Incremental	XP—Set Power-up Sequence Mode
MR—Motor Resolution	XQ—Sequence Interrupted Run Mode
N—End of Loop	XR—Run a Sequence
O—Output	XRP—Sequence Run With Pause
OS—Report Homing Function Set-	XSD—Sequence Status Definition
Ups	XSP—Sequence Status Power-up
OSA—Define Active State of End-of-	XSR—Sequence Status Run
Travel Limits	XSS—Sequence Status
OSB—Back Up To Home	XT—Sequence Termination
OSC—Define Active State of Home	XU—Upload Sequence
Switch	Y—Stop Loop
OSD—Enable Encoder Z Channel for	Z—Reset
Home	#—Remote Address Numbering
	%—Reset Communication