
OSC—Define Active State of Home Switch

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

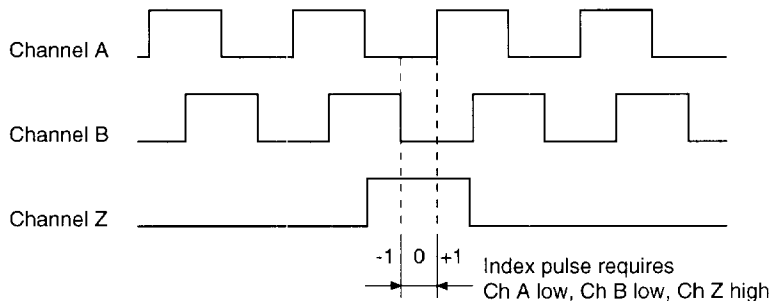
OSD—Enable Encoder Z Channel for Home

- Command Type: Set-up
- Valid Software Version: A
- Syntax: <a>OSDn
- Units: N/A
- Range: n = 0, 1
- Default Value: 0
- Attributes: Buffered,
Savable in Sequence
- 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 and final home position inside the home region. As the next drawing shows, the final home position occurs when Channel A is low, Channel B is low, and Channel Z is high.



For OSD1 to be selected, OSB1 must also be selected.

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

OSH—Reference Edge of Home Switch

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>OSHn | <input type="checkbox"/> Units: NA |
| <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: GH, OSB, OSC, OSD |

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

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

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

<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

The home limit becomes active when the home limit input is closed. The controller 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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aPR | <input type="checkbox"/> Units: Encoder counts |
| <input type="checkbox"/> Range: ±1,073,741,820 | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered,
Savable in Sequence | <input type="checkbox"/> See Also: D, MPA, MPI, MN, PZ, PX
<input type="checkbox"/> Response to aPR is *±nnnnnnnnnn |

This command reports the commanded motor position relative to the power-up position. When a **D** command is issued, the distance is relative to the value of **PR**. The difference between the commanded position (**PR**) and the actual encoder position (**PX**) is the position error (**DPE**). The controller is always trying to minimize the position error. You can reset the encoder position counter to zero by using the position zero (**PZ**) command or reset (**Z**). After **PZ** the encoder position (**PX**) will be set to zero. If there was a position error before

the **PZ** was issued, the value of **PR** will differ from the value of **PX** by the amount of the position error. Increasing integral gain (**CIG**) can help reduce the position error at rest thus insuring the value of **PR** equals the value of **PX**.

<u>Command</u>	<u>Description</u>
1PR	Commanded position report. (*+0000002000)
1PX	Encoder position report (*+0000002005)

The actual motor position is 5 encoder counts from the commanded position.

PS—Pause

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

This command pauses execution of a command string or sequence until the controller 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 controllers that have long command strings.

<u>Command</u>	<u>Description</u>
PS	Pauses execution of commands until the controller receives the Continue (C) command
A5	Sets acceleration to 5 revs/sec ²
V5	Sets velocity to 5 revs/sec
D4000	Sets move distance to 4,000 encoder counts
G	Executes the move (Go)
T2	Delays the move for 2 seconds
G	Executes the move (Go)
C	Continues Execution

When the controller receives the **C** command, the motor moves 4,000 encoder counts twice with a 2 second delay between moves.

PX—Report Absolute Encoder Position

- Command Type: Status
- Syntax: aPX
- Range: ±1,073,741,820
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: Encoder counts
- Default Value: N/A
- See Also: PR, PZ
- Response to aPX *±nnnnnnnnn

This command reports the actual motor position as measured by the encoder. When a **D** command is issued, the distance is relative to the value of **PR** not the value of **PX**. The difference between the commanded position (**PR**) and the actual encoder position (**PX**) is the position error (**DPE**). The controller is always trying to minimize the position error. You can reset the encoder position counter to zero by using the position zero (**PZ**) command or reset (**Z**). After **PZ** the encoder position (**PX**) will be set to zero. If there was a position error before the **PZ** was issued, the value of **PR** will differ from the value of **PX** by the amount of the position error. Increasing integral gain (**CIG**) can help reduce the position error at rest thus insuring the value of **PR** equals the value of **PX**.

<u>Command</u>	<u>Description</u>
MN	Set to mode normal
PZ	Sets the absolute counter to zero
A1Ø	Sets acceleration to 10 rev/sec ²
V5	Sets velocity to 5 rev/sec
D46ØØ	Sets move distance to 4,600 encoder counts
G	Executes the move (Go)
1PX	After the motor executes the move, the encoder position is reported: The response is *+0000004600.

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 encoder position counter to zero. If there was a position error before the **PZ** was issued, the new value of **PR** will differ from the value of **PX** by the amount of the position error. Absolute counter will also be set to zero when you cycle power or when you successfully execute a homing (**GH**) function.

<u>Command</u>	<u>Description</u>
MN	Enter position mode (mode normal)
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
D4ØØØØ	Sets move distance to 40000 encoder counts
G	Executes the move (Go)

⑥ SOFTWARE COMMANDS • OEM070

1PX	Reports absolute encoder position (*+0000040000)
PZ	Sets the absolute counter to zero
1PX	Reports absolute encoder position (*+0000000000)

“—Quote

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: "x | <input type="checkbox"/> Units: N/A |
| <input type="checkbox"/> Range: x = up to 17 ASCII characters | <input type="checkbox"/> Default Value: N/A |
| <input type="checkbox"/> Attributes: Buffered, Savable in Sequence | <input type="checkbox"/> See Also: CR, LF |
| | <input type="checkbox"/> 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. On a daisy chain of multiple units, if more than one unit is reporting a message at once, the messages will overlap and be garbled.

<u>Command</u>	<u>Description</u>
PS	Pause execution until Continue (C) is entered
A5	Set acceleration to 5 revs/sec ²
V5	Set velocity to 5 revs/sec
D2000	Set distance to 2,000 encoder counts
G	Executes the move (Go)
"MOVE_DONE	Transmits message
C	Continue move

After the move, the OEM Controller will send the message MOVE_DONE via the RS-232C port

Q0—Exit Velocity Profiling Mode

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>Q0 | <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: 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 Controller to enter Normal mode (**MN**).

Q1—Enter Velocity Profiling Mode

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>Q1 | <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: QØ, RM |

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

<u>Command</u>	<u>Description</u>
ER2000	Set encoder resolution to 2000
Q1	Enter Velocity Streaming mode
RM0000220C	Accelerate to 0.25 revs/sec ²
RM00004418	Accelerate to 0.5 revs/sec ²
RM00008831	Accelerate to 1 revs/sec ²
RM00011062	Accelerate to 2 revs/sec ²
RM00008831	Decelerate to 1 revs/sec ²
RM00004418	Decelerate to 0.5 revs/sec ²
RM0000220C	Decelerate to 0.25 revs/sec ²
RM00000000	Decelerate to 0 revs/sec ²
QØ	Exit Velocity Streaming mode

R—Request Controller 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 Controller Status (**R**) command can be used to indicate the general status of the controller. Possible responses are:

<u>Character</u>	<u>Definition</u>
*R	Ready
*S	Ready, Attention Needed
*B	Busy
*C	Busy, Attention Needed

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

- * Performing a 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

⑤ **SOFTWARE COMMANDS • OEM070**

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

- * Go home failed
- * Limit has been encountered
- * Sequence execution was unsuccessful
- * Sequence memory checksum error
- * Undervoltage
- * Drive recently enabled

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.

<u>Command</u>	<u>Response</u>
1R	*R (Controller ready to accept a command, and no error conditions exist.)

RA—Limit Switch Status Request

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aRA | <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: R, RB |
| | <input type="checkbox"/> 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	*@ (the last move was not terminated by a limit and 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
*H	No	No	No	Yes
*I	Yes	No	No	Yes
*J	No	Yes	No	Yes
*K	Yes	Yes	No	Yes
*L	No	No	Yes	Yes
*M	Yes	No	Yes	Yes
*N	No	Yes	Yes	Yes
*O	Yes	Yes	Yes	Yes

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

<u>Command</u>	<u>Response</u>
1RB	*A (After issuing a 1RB command, the response came back as *A. This means that the controller is currently executing a loop.)

RC—Homing Status Request

<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 condition for which status is indicated is:

Homing Function Failure:

In this condition, the controller 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.

Response Character	Go Home Successful?
*@	YES
*B	NO

<u>Command</u>	<u>Description</u>
1RC	*B Go home was unsuccessful.

RFS—Return Servo Gains to Factory Settings

<input type="checkbox"/> Command Type: Status	<input type="checkbox"/> Valid Software Version: A
<input type="checkbox"/> Syntax: aRFS	<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 aRFS is *x

This command can be used to return all the servo gains to the factory defaults immediately. This command is useful when the current servo gain values are inappropriate and re-tuning is required. By returning to factory defaults the gains will be in proper relationship to one another allowing easy fine tuning of the system. The factory defaults are:

Proportional Gain	(CPG)	16
Integral Gain	(CIG)	2
Derivative Gain	(CDG)	240
Derivative Sampling Period	(CTG)	0

Position Error	(CPE)	4000
Integral Limit	(CIL)	2

<u>Command</u>	<u>Description</u>
1RFS	All servo gains returned to the factory defaults

RM—Rate Multiplier in Velocity Streaming

- Command Type: Motion
- Valid Software Version: A
- Syntax: <a>RMn
- Units: revs/sec
- Range: n = 0 - FFFFFFFF
- Default Value: None
- Attributes: Immediate, Never Saved
- See Also: D, H, Q0, Q1

The **RM** command sets an immediate velocity where n represents an 8-digit hexadecimal value. The value for n is determined with the following formula:

$$(\text{desired revs/sec}) \cdot (\text{encoder resolution}) \cdot 17.432576 = \text{decimal \#}$$

for velocity value to be rounded off to the closest whole number.

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 Controller to exit Velocity Streaming mode. To recover from a limit stop in **RM** mode, **Q0** 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 (0), turning **RM** mode off, changing the direction, and re-enabling **RM** mode. Exiting **RM** mode with **Q0** causes the OEM Controller to enter Normal mode (**MN**).

<u>Command</u>	<u>Response</u>
ER2000	Set encoder resolution to 2000
Q1	Enter Velocity Streaming mode
RM0000220C	Accelerate to 0.25 revs/sec ²
RM00004418	Accelerate to 0.5 revs/sec ²
RM00008831	Accelerate to 1 revs/sec ²
RM00011062	Accelerate to 2 revs/sec ²
RM00008831	Decelerate to 1 revs/sec ²
RM00004418	Decelerate to 0.5 revs/sec ²
RM0000220C	Decelerate to 0.25 revs/sec ²
RM00000000	Decelerate to 0 revs/sec ²
Q0	Exit Velocity Streaming mode

RSE—Report Servo Errors

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Status | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: aRSE | <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: R, RA, RB, RC |
| | <input type="checkbox"/> Response to aRSE is *x |

The Report Servo Errors (**RSE**) command can be used to indicate the general status of the controller. Possible responses are:

<u>Character</u>	<u>Definition</u>
*0	No errors
*2	Excessive position error
*4	Drive fault
*6	Commanded shutdown
*8	Undervoltage, or drive was recently enabled

During a fault condition, the RSE command can be used to interrogate the controller for the reason of the fault. The following can cause a latched fault condition—the red LED will be illuminated, and the fault output will be active:

- Drive commanded shutdown via the **OFF** or **ST1** commands
- Position error exceeded the value set in the **CPE** command
- Drive fault (overvoltage, overtemperature, etc.)

The next two conditions are not latched, and do not illuminate the red LED. They will cause a ***8** response to **RSE**, until the next move command is issued.

- Undervoltage (voltage at drive's DC input drops below 21.5VDC)
- Drive was recently enabled

<u>Command</u>	<u>Response</u>
1RSE	*0 (Controller ready to accept a command, and no error conditions exist.)

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-nnnnnn-nnnn |

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	92-016637-01A

The product is identified by *92-016637-01A, and the revision level is identified by A.

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, ST |

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 controller will stop the current move but it will not clear the command buffer.

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

The **S** command is not buffered. As soon as the controller 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.

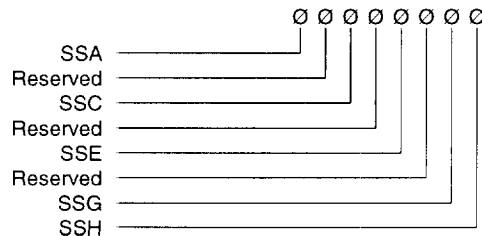
This command allows you to select the best possible trade-off between noise immunity and speed for a given application. If you make your scan time too short, the OEM070 may respond to an electrical glitch. If you issue the Scan command with only a device address (**1SN**), the controller will respond 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, SSC, 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**.



SSA—RS-232C Echo Control

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

This command turns the RS-232C echo (transmission of characters received from the remote device by the OEM070) on and off.

SSAØ = Echo on

SSA1 = Echo off

In the Echo On (**SSAØ**) mode, characters that are received by the controller are echoed automatically. In the Echo Off (**SSA1**) mode, characters are not echoed from the OEM070. This command is useful if your computer cannot handle echoes. In a daisy chain, you must have the echo on (**SSAØ**) to allow controllers 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 controller are not echoed back to the host.)

SSC—Output #1 on In Position

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

With **SSC** set to 1, output 1 will turn on when the motor is within the In Position window for the specified time defined by the **CEW** and **CIT** commands

b = 0: Normal

b = 1: Output 1 is configured as an "In Position" output

<u>Command</u>	<u>Response</u>
SSC1	Set output 1 as an In Position output

SSE Enable/Disable Communication Error Checking

- Command Type: Set-Up
- Valid Software Version: E
- Syntax: <a>SSEn
- Units: N/A
- Range: 0 (disable), 1 (enable)
- Default Value: 0 (disable)
- Attributes: Buffered, Savable in Sequence
- See Also: %

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 the controller, and **SSE0** disables error checking. See the % command for the types of errors detected.

SSG—Clear/Save the Command Buffer on Limit

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

SSG0 = 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 **SSG0** 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.

<u>Command</u>	<u>Description</u>
SSG1	Saves buffer on limit
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D4000	Sets distance to 4,000 encoder counts
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

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Set-Up | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>SSHn | <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: S |

SSHØ = Clears command buffer on stop

SSH1 = Saves command buffer on stop

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

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

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

ST—Shutdown

- | | |
|---|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>STn | <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: OFF, ON |

The ST1 command immediately disables the drive through the enable output, which would be connected to the enable input of the drive. This command can be used to shut down the drive in an emergency. This command is functionally identical to the OFF command.

The ST0 command immediately re-enables the drive. This command

⑤ SOFTWARE COMMANDS • OEM070

is used to re-enable the drive after a commanded shutdown or after a fault condition such as excessive position error. This command is functionally identical to the ON command.

For details on the enable output see *Chapter ② Installation*. Also, refer to the drive product user guide for the proper use of enable input.

Command	Description
ST1	Disable the drive product
ST0	Enable the drive product

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 controller 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
D4000	Sets distance to 4,000 encoder counts
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>TRnnnnn Units: See Below
- Range: n = 0, 1, or X Default Value: None
- Attributes: Buffered, See Also: IN
Savable in Sequence

This command allows you to specify a trigger configuration to be

matched before continuing execution of the move, where *nnnnn* corresponds to triggers 1, 2, 3, 4 and 5 respectively. The possible values for *n* are as follows:

- n = 1** Wait for the trigger input to be high (opened)
- n = 0** Wait for the trigger input to be low (grounded)
- n = X** Ignore the trigger input

The lowest numbered input will be the first trigger. For example, if input 3,4 and 5 are configured with the IN command as triggers, they will be trigger 1,2 and 3 respectively.

<u>Command</u>	<u>Description</u>
1IN3A	Configure input 3 as trigger input 1
1IN5A	Configure input 5 as trigger input 2
TR10	Wait for input 1 to be opened and input 2 to be grounded before going on to the next command
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D4000	Sets distance to 4,000 encoder counts
G	Executes the move (Go)

Motion will not occur until trigger conditions are true.

U—Pause and Wait for Continue

- Command Type: Programming
- Valid Software Version: A
- Syntax: <a>U
- Units: N/A
- Range: N/A
- Default Value: N/A
- Attributes: Immediate, Never Saved
- 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 controller continues to execute the program (at the point where it was interrupted). The controller continues processing when it receives the **C** command. This command is typically used to stop a machine while it is unattended.

<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
LØ	Loops indefinitely
D46ØØ	Sets distance to 4,600 encoder counts
G	Executes the move (G)
T1Ø	Waits 10 seconds after the move
N	Ends loop
U	Halts execution until the controller receives the Continue command (C)

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 controller receives the **C** command. In reaction to the **T1Ø** command in the loop, there may be a 10 second delay before motion resumes after the **C** is executed, depending on when the **U** command is completed.

V—Velocity

- Command Type: Motion
- Syntax: <a>Vn
- Range: n = 0.01 - 200.00
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: revs/sec
- Default Value: 1
- See Also: A, D, G, GH, MR

The **V** command defines the maximum speed at which the motor will run when given the Go (**G**) command. The maximum encoder frequency the controller can accept is 960 kHz. 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 issued the controller moves to the next command in the buffer.

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 OEM070 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 controller will respond with the current velocity value (***Vn**).*

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

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, Savable in Sequence	<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 OEM070 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. Sequences can only be permanently saved with the -M2 (BBRAM) option. Without the -M2 option sequences can be saved in operating RAM, and *will* be retained after a reset (Z) but not after a power cycle.

<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 encoder counts
G	Executes the move (Go)
XT	Ends definition of Sequence #1
XR1	Executes Sequence #1

XE—Sequence Erase

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

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

<u>Command</u>	<u>Description</u>
XE1	Deletes Sequence 1
XD1	Defines Sequence 1

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: IN, XQ, XSP, XSR

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

If $n = 8$, the sequence whose number appears on the sequence select inputs (configured with the **IN** command) 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 (configured with the **IN** command) will be executed on power-up. When the first sequence is finished in **XP9** mode, the

OEM070 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 = 0:** 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 OEM070 from selecting the next sequence until all the sequence select inputs are first opened.

Sequences can only be permanently saved with the -M2 (BDRAM) option. Without the -M2 option sequences can be saved in operating RAM, and *will* be retained after a reset (Z) but not after a power cycle.

<u>Command</u>	<u>Description</u>
XP1	Executes Sequence #1 on power-up
XE1	Erases Sequence #1
XD1	Defines Sequence #1
LD3	Disables CW & CCW limits
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D4000	Sets distance to 4,000 encoder counts
G	Executes the move (Go)
XT	Ends definition of Sequence #1
Z	Resets the controller

The motor moves 4,000 encoder counts during power-up (with -M2 option only) or reset (Z).

XQ—Sequence Interrupted Run Mode

- Command Type: Set-Up
- Valid Software Version: A
- Syntax: <a>XQn
- Units: Sequences
- Range: n = 0, 1
- Default Value: 0
- Attributes: Buffered, Savable in Sequence
- See Also: XP

- n = 1: Interrupted Run mode is set (on)
- n = 0: Interrupted Run mode is reset (off)

This command can be used only when the OEM070 is stand-alone power-up sequencing in **XP9** mode. In **XP9** mode, if **XQ1** is executed, the OEM070 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 controller will 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.

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

If you execute Sequence #1 during power up by setting the sequence select inputs (configured with the **IN** command) inputs properly, Interrupted Run mode will be set.

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.

Sequences can only be permanently saved with the -M2 (BBRAM) option. Without the -M2 option sequences can be saved in operating RAM, and will be retained after a reset (Z) but not after a power cycle.

Command	Description
XE1	Erases sequence #1
XD1	Defines sequence #1
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D10000	Sets distance to 10,000 encoder counts
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 controller executes the command buffer. The pause condition is invoked only if the sequence is valid. This allows you to execute a sequence without the delay of buffering that sequence.

Command	Description
XE5	Erases Sequence #5
XD5	Defines Sequence #5
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D10000	Sets distance to 10,000 encoder counts
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:

- n = 0: 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 (A sequence already exists as sequence 1)

XSP—Sequence Status Power-up

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

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

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

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

- * \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 aXSSn is *x |

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.

NOTE: In your communication program, use sufficient time delays after downloading a sequence before you send more commands to

⑤ SOFTWARE COMMANDS • OEM070

the OEM070. In particular, after sending the **XT** command, wait at least 12.5 msec before sending a **Z** command.

<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
D4ØØØ	Sets distance to 4,000 encoder counts
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 controller 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 controller processes reach the last command of the current loop. At that time, the controller executes the command that follows the End Loop (**N**) command. You cannot restart the command loop unless you enter the entire command structure, including the Loop (**L**) and End Loop (**N**) commands.

<u>Command</u>	<u>Description</u>
L	Loops indefinitely
A10	Sets acceleration to 10 revs/sec ²
V5	Sets velocity to 5 revs/sec
D4000	Sets distance to 4,000 encoder counts
T2	Waits 2 seconds
G	Executes the move (Go)
N	Ends loop
Y	Stops loop

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

Z—Reset

- | | |
|--|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>Z | <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: K, S |

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

When you use the **Z** command, the controller is busy for 1,000 ms and ignores all commands. This command sets all position counters to zero and returns all values except the **XP** command to factory defaults.

<u>Command</u>	<u>Description</u>
1Z	Resets controller with address 1

#—Address Numbering

- Command Type: Set-up
- Valid Software Version: A
- Syntax: <a>#n
- Units: Address number
- Range: n=1-255
- Default Value: 1
- Attributes: Immediate, Automatically Saved
- See Also: E,F

This command sets the individual unit address for each OEM070, allowing addresses up to 255. Upon receipt of the command, the OEM070 will assign itself the address in the command and 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)

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

<u>Command</u>	<u>Description</u>
#1	#5 (for a daisy chain of 4 units, the units will assign themselves addresses 1 through 4 and return #5 as confirmation).

%—Reset Communication

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

When the OEM070 detects a communication error, it ignores all *external* commands and echoes an **&** for each byte it receives from the host. You can use this % command 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:

<u>Character</u>	<u>Definition</u>
*	No errors
*0	Unit upstream (daisy chained)
*1	Overrun, data received too fast
*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.

%	*0*0*0*1***** (First 5 units report no error, 6 th unit detected an overrun error, and the last 3 units turned communication off because of unit 6)
---	--

Summary of Commands

A—Acceleration	OSC—Define Active State of Home Switch
B—Buffer Status	OSD—Enable Encoder Z Channel for Home
BCDG—Buffered Configure Derivative Gain	OSH—Reference Edge of Home Switch
BCIG—Buffered Configure Integral Gain	PR—Position Report
BCIL—Buffered Configure Integral Limit	PS—Pause
BCPE—Buffered Configure Position Error	PX—Report Absolute Encoder Position
BCPG—Buffered Configure Proportional Gain	PZ—Set Absolute Counter to Zero
BCTG—Buffered Configure Derivative Sampling Period	“—Quote
BS—Buffer Size Status	Q1—Enter Velocity Profiling Mode
CDG—Configure Derivative Gain	QØ—Exit Velocity Profiling Mode
CEW—Configure In Position Error Window	R—Request Controller Status
CIG—Configure Integral Gain	RA—Limit Switch Status Report
CIL—Configure Integral Limit	RB—Loop, Pause, Shutdown, Trigger Status Report
CIT—Configure In Position Time	RC—Homing Status Report
CPE—Configure Maximum Position Error	RFS—Return Servo Gains to Factory Settings
CPG—Configure Proportional Gain	RM—Rate Multiplier in Velocity Streaming Mode
CR—Carriage Return	RSE—Report Servo Errors
CTG—Configure Filter Time Constant	RV—Revision Level
D—Distance	S—Stop
DPA—Display Position Actual	SN—Scan
DPE—Display Position Error	SS—Software Switch Function Status
DVA—Display Velocity Actual	SSA—RS-232C Echo Control
E—Enable Communications	SSC—Output #1 on In Position
ER—Encoder Resolution	SSE—Enable/Disable Communication Error Checking
F—Disable Communications	SSG—Clear/Save the Command Buffer on Limit
G—Go	SSH—Clear/Save Command Buffer on Stop
GH—Go Home	ST—Shutdown
^H—Delete	T—Time Delay
H—Set Direction	TR—Wait For Trigger
IN—Set Input Functions	U—Pause and Wait for Continue
IS —Input Status	V—Velocity
K—Kill	XC—Sequence Checksum
L—Loop	XD—Sequence Definition
LD—Limit Disable	XE—Sequence Erase
LF—Line Feed	XP—Set Power-up Sequence Mode
MC—Mode Continuous	XQ—Sequence Interrupted Run Mode
MN—Mode Normal	XR—Run a Sequence
MPA—Mode Position Absolute	XRP—Sequence Run With Pause
MPI—Mode Position Incremental	XSD—Sequence Status Definition
N—End of Loop	XSP—Sequence Status Power-up
O—Output	XSR—Sequence Status Run
OFF—Servo Disable	XSS—Sequence Status
ON—Servo Enable	XT—Sequence Termination
OS—Report Homing Function Set-Ups	XU—Upload Sequence
OSA—Define Active State of End-of-Travel Limits	Y—Stop Loop
OSB—Back Up To Home	Z—Reset
	#—Address Numbering
	%—Reset Communication