C H A P T E R ⑤

Software Commands

Chapter Objective

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

Command Descriptions

① A—Acceleration *(Example Command)*

② Command Type: Motion
③ Syntax: <a>An
④ Range: n = 0.01-999.99
⑤ Attributes: Buffered, Savable in Sequence
⑥ Valid Software Version: A
⑦ Units: revs/sec^2
⑧ Default Value: A = 100
⑨ See Also: D, G, MR, V
⑩ Response to aA is *An

① *Command Mnemonic*

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

② *Command Type*

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

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 1-255.

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: Motion
- ❑ Syntax: <a>An
- ❑ Range: n = 0.01-9999.99
- ❑ Attributes: Buffered, Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: revs/sec^2
- ❑ Default Value: A = 100
- ❑ See Also: D, G, MR, V
- ❑ Response to aA is *An

The Acceleration command specifies the rotary acceleration rate to be used for the next Go (**G**) command. The acceleration remains set until you change it. You do not need to reissue this command for subsequent Go (**G**) commands. Accelerations outside the valid range cause the acceleration to remain at the previous valid **A** setting.

If the Acceleration command is entered with only a device address (**1A**), the controller 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. Acceleration cannot be changed on the fly. The minimum acceleration is

Min Accel = Encoder resolution (**ER** command) x .00465

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

B—Buffer Status

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

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

The command buffer is 512 bytes long. A ***B** response will be issued if less than 5% of the command buffer is free.

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

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

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

BCDG—Buffered Configure Derivative Gain

- ❑ Command Type: Set-up
- ❑ Syntax: <a>BCDGn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 240
- ❑ Response to aBCDG is *CDGn
- ❑ See Also: BCIG, BCPG, BCTG, CIG, CPG, CTG

This buffered command is used for system tuning. This term represents the gain applied to the derivative of the position error—in other words, the rate at which the position error is changing. This gain produces a damping effect similar to velocity feedback.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
BCDG400	Set the derivative gain term to 400
1BCDG	Reports derivative gain term (*CDG400)

BCIG—Buffered Configure Integral Gain

- ❑ Command Type: Set-up
- ❑ Syntax: <a>BCIGn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 2
- ❑ Response to aBCIG is *CIGn
- ❑ See Also: BCDG, BCPG, BCTG, CDG, CPG, CTG

This buffered command is used for system tuning. This term represents the gain applied to the integral of the position error—the net accumulation of the position error over time. Thus integral gain will contribute when a position error is not being reduced over time, as may be caused by the effects of friction or gravity. This gain will improve overall accuracy but may increase settling time and, if excessive, may cause a low frequency oscillation around the commanded position.

Before you increase **BCIG**, you must first increase the integral limit (**BCIL**) to an equal or higher value.

Refer to *Chapter ④Tuning* for more information.

Command	Description
BCIL40	Set the integral limit term to 40
BCIG10	Set the integral gain term to 10
1BCIG	Reports integral gain term (*CIG10)

BCIL—Buffered Configure Integral Limit

- ❑ Command Type: Set-up
- ❑ Syntax: <a>BCILn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 2
- ❑ Response to aBCIL is *CILn
- ❑ See Also: BCDG, BCIL, BCPG, BCTG, CDG, CIL,CPG, CTG

This buffered command is used for system tuning. This term represents the limit applied to the integral gain contribution of the PID equation. A high integral gain with a large inertial load can cause a non-ringing overshoot of the commanded position. By limiting the contribution of integral action, this overshoot can be minimized.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
BCIG10	Set the integral gain term to 10
BCIL40	Set the integral limit to 40
1BCIL	Reports integral limit term (*CIL40)

BCPE—Buffered Configure Position Error

- ❑ Command Type: Set-up
- ❑ Syntax: <a>BCPEn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 4000
- ❑ Response to aBCPE is *CPEn
- ❑ See Also: DPE, CPE

This buffered command defines the maximum allowable position or following error. If the actual position error ever exceeds the allowable position error, the controller will generate a fault condition and shut down the drive. If the maximum allowable position error is set to Ø, the function is disabled and no amount of position error will generate the fault condition.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
BCPE400	Set the maximum allowable position error to 400 encoder counts
BCPE0	Disable fault generation due to position error
1BCPE	Reports maximum position error setting (*CPE0)

BCPG—Buffered Configure Proportional Gain

- ❑ Command Type: Set-up
- ❑ Syntax: <a>BCPGn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 16
- ❑ Response to aBCPG is *CPGn
- ❑ See Also: BCDG, BCIG, BCTG, CDG, CIG, CTG

This buffered command is used for system tuning. This term represents the gain applied directly to the position error. The proportional gain sets how actively the system will respond to position error. High proportional gain will give a stiff, responsive system, but may result in overshoot and oscillation.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
BCPG50	Set the proportional gain term to 50
1BCPG	Reports proportional gain term (*CPG50)

BCTG—Buffered Configure Derivative Sampling Period

- ❑ Command Type: Set-up
- ❑ Syntax: <a>BCTGn
- ❑ Range: n = 0-255
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: 266 µsec
- ❑ Default Value: 0
- ❑ Response to aBCTG is *CTGn
- ❑ See Also: BCDG, BCIG, BCPG, CDG, CIG, CPG

This buffered command is used for system tuning. Use **BCTG** to adjust the derivative sampling period. The *system* sampling period—266 µsec—is the period between updates of position error. The *derivative* sampling period is an integer multiple of the system sampling period. In general, a longer derivative sampling period gives a more constant derivative term and improves stability. Many systems require a low **BCTG** value to prevent oscillations, however. Therefore, start with a low value and increase it incrementally.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
BCTG0	Set the derivative sampling period to 266 µsec
BCTG1	Set the derivative sampling period to 532 µsec
BCTG2	Set the derivative sampling period to 798 µsec
BCTG3	Set the derivative sampling period to 1064 µsec
1BCTG	Reports derivative sampling period (*CTG3)

BS—Buffer Size Status

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

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

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

C—Continue

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

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

Command	Description
MC	Sets move to continuous mode
A1Ø	Sets acceleration to 10 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
PS	Pauses system until controller receives **C** command
G	Accelerates the motor to 10 revs/sec
C	Continues executing commands in the buffer

The motor will not execute the **G** command until the **C** command is issued

CDG—Configure Derivative Gain

- ❑ Command Type: Set-up
- ❑ Syntax: <a>CDGn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Immediate Automatically Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 240
- ❑ Response to aCDG is *CDGn
- ❑ See Also: CIG, CPG, CTG, RFS

This command is used for system tuning. This term represents the gain applied to the derivative of the position error, in other words, the rate at which the position error is changing. This gain produces a damping effect similar to velocity feedback.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
CDG400	Set the derivative gain term to 400
1CDG	Reports derivative gain term (*CDG400)

CEW—Configure In Position Error Window

- ❑ Command Type: Set-up
- ❑ Syntax: <a>CEWn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: encoder counts
- ❑ Default Value: 50
- ❑ Response to aCEW is *CEWn
- ❑ See Also: CIT, SSC

This command, together with the **CIT** command, can be used to configure an In Position window, which can be used to indicate that the preceding move has terminated.

The In Position condition is met when:

- The controller algorithm has finished (no input position command)
- The CEW condition is met (the position error is less than that specified by the **CEW** command).
- The above condition has been true for the length of time specified by the **CIT** command

The position error range, specified by *n* in **CEWn**, is the maximum number of encoder counts allowed on either side of the desired position. For example, if *n* = 10, then the In Position window is 20 encoder counts wide.

Output 1 can be configured with the **SSC** command to show the state of the In Position detector. This allows the user to trigger external hardware from the In Position condition.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
CEW10	Configure an In Position Error Window ±10 encoder counts either side of desired position
1CEW	Reports ± number of encoder counts (*CEW10)

CIG—Configure Integral Gain

- ❑ Command Type: Set-up
- ❑ Syntax: <a>CIGn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Immediate Automatically Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 2
- ❑ Response to aCIG is *CIGn
- ❑ See Also: CIL, CDG, CPG, CTG, RFS

This command is used for system tuning. This term represents the gain applied to the integral of the position error—the net accumulation of the position error over time. Thus integral gain will contribute when a position error is not being reduced over time, as may be caused by the effects of friction or gravity. This gain will improve overall accuracy but may increase settling time and, if excessive, may cause a low frequency oscillation around the commanded position.

Before you increase **CIG**, you must first increase the integral limit (**CIL**) to an equal or higher value.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
CIL40	Set the integral limit term to 40
CIG10	Set the integral gain term to 10
1CIG	Reports integral gain term (*CIG10)

CIL—Configure Integral Limit

- ❑ Command Type: Set-up
- ❑ Syntax: <a>CILn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Immediate Automatically Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 2
- ❑ Response to aCIL is *CILn
- ❑ See Also: CIG,CDG, CPG, CTG, RFS

This command is used for system tuning. This term represents the limit applied to the integral gain contribution of the PID equation. A high integral gain with a large inertial load can cause a non-ringing overshoot of the commanded position. By limiting the contribution of integral action, this overshoot can be minimized.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
CIG10	Set the integral gain term to 10
CIL40	Set the integral limit to 40
1CIL	Reports integral limit term (*CIL40)

CIT—Configure In Position Time

- ❑ Command Type: Setup
- ❑ Syntax: <a>CITn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: Milliseconds
- ❑ Default Value: 20
- ❑ Response to aCIT is *CITn
- ❑ See Also: CEW, SSC

This command is used to specify the time period that the servo is to be within the In Position window before the "In Position" signal is generated. The range is 0 to 32,767, and is the number of milliseconds to be used as the testing time frame.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
1SSC1	Set output 1 as "In Position"
1CIT30	Set "In Position" time to 30ms
1CEW20	Set allowable position error to ±20 encoder counts

CPE—Configure Position Error

- ❑ Command Type: Set-up
- ❑ Syntax: <a>CPEn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Immediate Automatically Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 4000
- ❑ Response to aCPE is *CPEn
- ❑ See Also: DPE, RFS

This command defines the maximum allowable position or following error. If the actual position error ever exceeds the allowable position error, the controller will generate a fault condition and shut down the drive. If the maximum allowable position error is set to Ø, the function is disabled and no amount of position error will generate the fault condition.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
CPE400	Set the maximum allowable position error to 400 encoder counts
CPE0	Disable fault generation due to position error
1CPE	Reports maximum position error setting (*CPE0)

CPG—Configure Proportional Gain

- ❑ Command Type: Set-up
- ❑ Syntax: <a>CPGn
- ❑ Range: n = 0-32,767
- ❑ Attributes: Immediate Automatically Saved
- ❑ Valid Software Version: A
- ❑ Units: N/A
- ❑ Default Value: 16
- ❑ Response to aCPG is *CPGn
- ❑ See Also: CDG, CIG, CTG, RFS

This command is used for system tuning. This term represents the gain applied directly to the position error. The proportional gain sets how active the system will be to position error. High proportional gain will give a stiff, responsive system, but may result in overshoot and oscillation.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
CPG50	Set the proportional gain term to 50
1CPG	Reports proportional gain term (*CPG50)

CR—Carriage Return

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

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

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

Command	Description
MN	Sets mode to preset mode
A5Ø	Sets acceleration to 50 revs/sec^2
V5	Sets Velocity to 5 revs/sec

D5ØØØ	Sets distance to 5,000 encoder counts
G	Executes the move (Go)
1CR	Sends a carriage return after move is completed

The motor moves 5,000 encoder counts. When the motor stops, the controller sends a carriage return over its interface.

CTG—Configure Derivative Sampling Period

- ❑ Command Type: Set-up
- ❑ Syntax: <a>CTGn
- ❑ Range: n = 0-255
- ❑ Attributes: Immediate Automatically Saved
- ❑ Valid Software Version: A
- ❑ Units: 266 µsec
- ❑ Default Value: 0
- ❑ Response to aCTG is *CTGn
- ❑ See Also: CDG, CIG, CPG, RFS

This command is used for system tuning. Use **CTG** to adjust the derivative sampling period. The *system* sampling period—266 µsec—is the period between updates of position error. The *derivative* sampling period is an integer multiple of the system sampling period. In general, a longer derivative sampling period gives a more constant derivative term and improves stability. Many systems require a low **CTG** value to prevent oscillations, however. Therefore, start with a low value and increase it incrementally.

Refer to *Chapter ④ Tuning* for more information.

Command	Description
CTG0	Set the derivative sampling period to 266 µsec
CTG1	Set the derivative sampling period to 532 µsec
CTG2	Set the derivative sampling period to 798 µsec
CTG3	Set the derivative sampling period to 1064 µsec
1CTG	Reports derivative sampling period (*CTG3)

D—Distance

- ❑ Command Type: Motion
- ❑ Syntax: <a>Dn
- ❑ Range: n = ±1,073,741,823
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: encoder counts
- ❑ Default Value: 4000
- ❑ Response to aD is *Dn
- ❑ See Also: A, G, MN, MPA, MPI, V, H

The Distance (**D**) command defines either the number of encoder counts 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 encoder counts) 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 present motor position and the position (referenced to the zero position) set with the **D** command. In either mode, the direction is controlled by the direction (+ or -) that precedes the distance value. The **D** command has no effect on continuous moves (**MC**).

In Mode Normal (**MN**) the position may not exceed the maximum distance range of 1,073,741,823 encoder counts. If the motor approaches the absolute maximum (plus or minus), the controller will not execute any **GO** commands that would cause the distance to exceed the absolute maximum. To proceed further, use the **PZ** command to reset the absolute counter to zero, and then resume operations.

If **D** is entered with only a device address (**1D**), the controller 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.

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

A servo motor with a 4000 count encoder will travel 1 rev (CW) after **G** is issued.

DPA—Display Position Actual

- ❑ Command Type: Status
- ❑ Syntax: aDPA
- ❑ Range: n = ±1,073,741,823
- ❑ Attributes: Immediate Device Specific, Never saved
- ❑ Valid Software Version: A
- ❑ Units: Encoder counts
- ❑ Default Value: NA
- ❑ Response to aDPA is *DPAn
- ❑ See Also: D, PZ

Single display of actual motor position as measured by the encoder. This command is functionally identical to the **PX** command. The response is the position in encoder counts as referenced to the last power reset or position zero command (**PZ**).

Command	Description
1DPA	Report the actual motor position of axis 1 (*+0004000000)

DPE—Display Position Error

- ❑ Command Type: Status
- ❑ Syntax: aDPE
- ❑ Range: n = ±2,147,483,646
- ❑ Attributes: Immediate Device Specific, Never saved
- ❑ Valid Software Version: A
- ❑ Units: Encoder counts
- ❑ Default Value: NA
- ❑ Response to aDPE is *DPEn
- ❑ See Also: D, DPA

Single display of position error. The response is the difference in encoder counts of the actual motor position and the commanded motor position. This information is used by the control algorithm to control the torque command to the motor. It is normal for the position error to be present during a move but large position errors are usually caused by inappropriate gain settings. Issuing a DPE command before the motor has settled into final position will also result in larger position errors.

Command	Description
1DPE	Report the position error of axis 1 (*+0000000005).

DVA—Display Velocity Actual

- ❑ Command Type: Status
- ❑ Syntax: aDVA
- ❑ Range: n = 0 – 20000
- ❑ Attributes: Immediate Device Specific, Never saved
- ❑ Valid Software Version: A
- ❑ Units: (rev/sec)*100
- ❑ Default Value: NA
- ❑ Response to aDVA is *DVAn
- ❑ See Also: DPA, V

Single display of actual motor velocity in revolutions per sec. There is an implied decimal point before the last two digits.

Command	Description
MC	Mode continuous
V2	Velocity of 2 rev/sec
G	Go
1DVA	Report the actual motor velocity of axis 1 (*00200)

E—Enable Communications

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

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

Command	Description
F	Disables all units (axes) on the communications interface
1E	Enables serial interface on Device 1
4E	Enables serial interface on Device 4
A1Ø	Set acceleration to 10 revs/sec^2
V5	Set velocity to 5 revs/sec
D5ØØØ	Sets distance to 5000 encoder counts
G	Executes the move (Go—only axes 1 & 4 will move)

ER—Encoder Resolution

- ❑ Command Type: Set-up
- ❑ Syntax: <a>ERn
- ❑ Range: n = 400 - 65,532
- ❑ Attributes: Buffered, Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: n = encoder counts/rev
- ❑ Default Value: 4000
- ❑ Response to aER is *ERn
- ❑ See Also: CPE

The encoder resolution defines the number of encoder counts the controller 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 counts due to quadrature detection)

Command	Description
ER4ØØØ	Sets encoder resolution to 4000 encoder counts per 1 motor revolution
1ER	Reports Encoder Resolution (*ER4000)

F—Disable Communications

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

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

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

G—Go

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

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

In the Normal mode (**MN**), moves can be either incremental or absolute. In the Incremental Preset mode (**MPI**), a **G** will initiate the move distance you specified with the **D** command. A **G** command in the Absolute Preset mode (**MPA**) will not cause motion unless the distance (**D**) value is different from the present motor position (**PR**) .

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

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

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

GH—Go Home

- ❑ Command Type: Motion
- ❑ Syntax: <a>GHn
- ❑ Range: n = ± .01 - 200
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: Revs/sec
- ❑ Default Value: n = 0
- ❑ See Also: OS, RC, V,IN

The Go Home (**GH**) command instructs the Controller to search for an absolute home position in the positive or negative (+ or -) direction. It defines home as the position where the home limit signal changes states. To use the GH command, one of the general purpose inputs (pins 1-5) must be configured as a home input (**IN** command).

Homing can be as simple as decelerating to a stop when the edge of the home limit is detected. By using the **OS** commands, the homing process can be tailored to meet the application needs.

OSB—*Back up to home* makes homing more repeatable by backing off the home switch and re-approaching at low speed. The final approach to home switch is always form the CW direction.

OSC—*Define active state of home* allows the use of a normally closed or normally open limit switch.

OSD—*Enable Z Channel for home* uses the Z channel of the encoder, in conjunction with a home switch, to determine the final home position. The Z channel is a more accurate home position than the edge of a switch.

OSH—*Reference edge of home switch* allows either edge of the home switch to be used as the final edge position.

The Controller 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 Homing 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
INE1	Configure input #1 as home input
OSB1	Back up the home switch
OSD1	Reference Z channel as final home
GH-2	The motor moves CCW at 2 revs/sec and looks for the Home Limit input to go active.

Since the motor is turning CCW, it will see the CW edge of the limit first. It will decelerate to a stop and turn at 0.1 rev/sec in the CW direction until it detects the Z channel.

^H—Delete

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

This command allows you to delete the last character that you entered. The **^H** command will not prevent execution of an immediate command. A new character may be entered at that position to replace the existing character. (**^H** indicates that the Ctrl key is held down when the H key is pressed.) This command prompts the controller 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.

This command will *not* delete characters beyond the last delimiter issued. Pressing the delete key will not delete the previous character.

H—Set Direction

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

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

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

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

Command	Description
MN	Sets Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D4ØØØ	Sets distance to 4,000 encoder counts
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

IN—Set Input Function

- ❑ Command Type: Set-up
- ❑ Syntax: aINxn
- ❑ Range: x = A – F, n = 1 – 5
- ❑ Attributes: Buffered
  Savable in sequence
- ❑ Valid Software Version: A
- ❑ Units: NA
- ❑ Default Value: AAAAA
- ❑ Response to aIN is *xxxxx
- ❑ See Also: IS, TR, XP, #, K, GH

This command configures the function of each of the 5 general purpose inputs. You can configure each input to perform one of the following functions:

Function A—*Trigger Input*
Used with the **TR** command as a comparison input. The TR command defines active high, low, or "don't care." Up to 5 inputs can be configured as trigger inputs.

Function B—*Sequence Select Input*
Executes predefined sequences from remote inputs based on the **XP** command. Active state (sequence selected) is high. Up to 3 inputs can be configured as sequence select inputs.

Function C—*Kill Input*
Immediately halts execution of the move. Same as kill (**K**) command. Active state (kill initiated) is high. Up to one input can be configured as a kill input.

Function D—*Stop Input*
Decelerates the motor to a stop using the value specified in the Acceleration (**A**) command (dumps the sequence or command buffer if configured by **SSHØ**). Same as Stop (**S**) command. Active state (stop initiated) is high. Up to one input can be configured as a stop input.

Function E—*Home Input*
Defines the motor home or origin position as executed by the **GH** command and configures with the Homing Function (**OS**) commands. **OSC** determines the active state of the input. Up to one input can be configured as a home input.

Function F—*Go Input*
Accelerates the motor to a velocity and distance specified in the Acceleration (**A**), Velocity (**V**), and Distance (**D**) commands. Same as Go (**G**) command. Active state (go initiated) is high. Up to one input can be configured as a go input.

Some of the functions (stop, kill, home, go) can only have one input configured to that function. If you try to configure another input to that function, the controller will not recognize the new function and revert back to the previous definition. For example, if input 1 is a kill function, and you want input 5 to be the kill function, you must first change input 1 to another function.

Function *B* (sequence select) can configure up to three inputs as sequence select inputs. If you try to configure more than three inputs, the controller will only recognize the first three. It will revert back to the previous definitions for the additional inputs.

Command	Description
1INA1	Configure input one as trigger input 1
1INB2	Configure input two as sequence select input 1
1INB3	Configure input three as sequence select input 2
1INC4	Configure input four as kill input
1INF5	Configure input five as go input
1IN	Reports input configuration (*ABBCF)

IS—Input Status

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

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

1—IN1 (Ø = Low, 1 = High)
2—IN2 (Ø = Low, 1 = High)
3—IN3 (Ø = Low, 1 = High)
4—IN4 (Ø = Low, 1 = High)
5—IN5 (Ø = Low, 1 = High)
6—CW limit (Ø = Low, 1 = High)
7—CCW limit (Ø = Low, 1 = High)
8—Fault (Ø = Low, 1 = High)

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

Command	Response
2IS	***ØØØ1ØØØ1** (The input status of device 2 is reported: I/O bits 1-3 and 5-7 are low (grounded), and I/O bits 4 (IN4), and 8 (Fault), are high)

K—Kill

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

This command causes commanded motion to cease immediately. There is *NO* deceleration of the motor. The motor will servo around the position where the kill was entered. 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.

WARNING

The Kill (**K**) command is not an emergency stop. The motor is not disabled. Motion caused by instability or incorrect wiring will not be stopped. An emergency stop should cut power to the amplifier or interrupt the hardware enable input (pin 10), and mechanically prevent the motor from turning.

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

L—Loop

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

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

The **N** command prompts the controller to proceed with further commands after the designated number of loops have been executed. The **Y** command stops loop execution after completing the current loop cycle. The Immediate Pause (**U**) command allows you to temporarily halt loop execution after completing the current loop cycle. You can use the Continue (**C**) command to resume loop execution.

Command	Description
L5	Loop 5 times
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D1ØØØØ	Sets distance to 10,000 encoder counts
G	Executes the move (Go)
N	End of loop

The commands in the loop will be executed 5 times.

LD—Limit Disable

- ❑ Command Type: Set-Up
- ❑ Syntax: <a>LDn
- ❑ Range: n = 0 - 3
- ❑ Attributes: Buffered Savable in Sequence
- ❑ Valid Software Version: A
- ❑ Units: See Below
- ❑ Default Value: Ø
- ❑ See Also: RA

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

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

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

LF—Line Feed

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

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

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

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

MC—Mode Continuous

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

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

The controller 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. After a new parameter is entered a Go (**G**) command is required. Acceleration (**A**) cannot be changed on the fly.

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

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

MN—Mode Normal

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

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

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

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

MPA—Mode Position Absolute

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

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

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

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

Command	Description
MN	Sets Normal mode (preset)
PZ	Resets absolute counter to zero
MPA	Sets position mode absolute
A5	Sets acceleration to 5 revs/sec^2
V1Ø	Sets velocity to 10 revs/sec
D4ØØØØ	Sets destination to absolute position 40,000
G	Motor will move to absolute position 40,000
D1ØØØØ	Sets destination to absolute position +10,000
G	Motor will move to absolute position +10,000

The motor will move 40,000 encoder counts in the CW direction (if starting from position Ø) and then move 30,000 encoder counts in the CCW direction to reach the absolute position 10,000.

MPI—Mode Position Incremental

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

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

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

Command	Description
MN	Set positioning mode normal (preset)
MPI	Set positioning mode incremental
A5	Sets acceleration to 5 revs/sec^2

V1Ø	Sets velocity to 10 revs/sec
D1ØØØØ	Sets distance of move to 10,000 encoder counts
G	Move 10,000 encoder counts CW
G	Move another 10,000 encoder counts CW

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

N—End of Loop

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

This command marks the end of a loop. You must 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	Description
MN	Sets move to Normal mode
A5	Sets acceleration to 5 revs/sec^2
V5	Sets velocity to 5 revs/sec
D1ØØØØ	Sets move distance to 10,000 encoder counts
L5	Loops the following commands five times
G	Executes the move (Go)
N	Ends the loop

The move will be executed five times.

OFF—De-Energize Drive

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

The OFF command immmediately 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 ST1 command.

Command	Description
OFF	De-energize the drive
1IS	*00000001 (fault bit active)

ON—Energize Drive

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

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

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

Command	Description
ON	Energize the drive
1IS	*00000000 (fault bit inactive)

O—Output

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

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

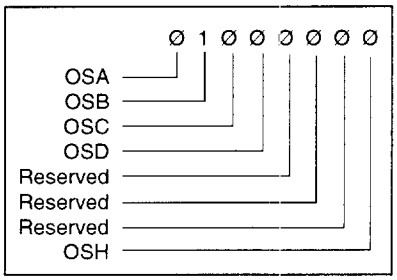
n=1 = Turns output bits on
n=Ø = Turns output bits off
n=X = Leaves output bits unchanged

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

OS—Report Homing Function Set-Ups

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

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



OSA—Define Active State of End-of-Travel Limits

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

OSAØ: Normally Closed Contacts
OSA1: Normally Open Contacts
This command sets the active state of the CW and CCW end-of-travel limit inputs. It enables you to use either normally closed or normally open switches.

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

OSB—Back Up To Home

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

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

This command is used to make homing more repeatable. With Back Up to Selected Home (**OSB**) command enabled, homing is a two step process. First it approaches the home switch at high speed (set by the GH command) until it sees the switch. Then it decelerates and returns to the switch at slow speed. Since it always makes its final approach from the same direction, the system will act differently whether the active edge of the switch is the first or second edge encountered.

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 1/2 of a revolution on the outside of the selected edge. Finally the motor will creep at 0.1 rev/sec in the direction of the active home region, 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 back to the active home region.

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