

CHAPTER FIVE

# 5 Software Commands

---

## IN THIS CHAPTER

---

- Descriptions of each command for the TQ10X Controller/Drive
  - Descriptions of the function, range, default, and sample use of each command for the TQ10X Controller/Drive.
-

# Command Descriptions

---

---

## ① A—Acceleration (*Example Command*)

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

### ① *Command Mnemonic*

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

### ② *Command Type*

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

- |   |   |
|---|---|
| <input type="checkbox"/> Command Type: Motion                         | <input type="checkbox"/> Valid Software Version: A    |
| <input type="checkbox"/> Syntax: <a>An                                | <input type="checkbox"/> Units: revs/sec <sup>2</sup> |
| <input type="checkbox"/> Range: n = 0.01-9999.99                      | <input type="checkbox"/> Default Value: A = 100       |
| <input type="checkbox"/> Attributes: Buffered,<br>Savable in Sequence | <input type="checkbox"/> See Also: D, G, MR, V        |
|   | <input type="checkbox"/> 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

$$\text{Min Accel} = \text{Encoder resolution (ER command)} \times .00465$$

<u>Command</u>	<u>Description</u>
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V1Ø</b>	Sets velocity to 10 revs/sec
<b>D2ØØØ</b>	Sets distance to 2,000 encoder counts
<b>G</b>	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.

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

<u>Command</u>	<u>Description</u>
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, BCIL, BCTG, CDG, CPG, CTG, CIL

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.

Refer to *Chapter ④—Tuning* for more information.

<b>Command</b>	<b>Description</b>
<b>BCIL40</b>	Set the integral limit term to 40
<b>BCIG10</b>	Set the integral gain term to 10
<b>1BCIG</b>	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, BCIG, BCPG, BCTG, CDG, CIG, CPG, CTG

This buffered command is used for system tuning. Since integral windup can sometimes cause very aggressive motion, you may want to limit this effect. CIL sets an upper limit on the integral, which in turn limits the integral term (**Integral \* CIG**).

Refer to *Chapter ④—Tuning* for more information.

<b>Command</b>	<b>Description</b>
<b>BCIG10</b>	Set the integral gain term to 10
<b>BCIL40</b>	Set the integral limit to 40
<b>1BCIL</b>	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 TQ10X will generate a fault condition and shut down power output to the motor. 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.

<b>Command</b>	<b>Description</b>
<b>BCPE400</b>	Set the maximum allowable position error to 400 encoder counts
<b>BCPE0</b>	Disable fault generation due to position error
<b>1BCPE</b>	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.

<u>Command</u>	<u>Description</u>
<b>BCPG50</b>	Set the proportional gain term to 50
<b>1BCPG</b>	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  $\mu$ sec
- 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  $\mu$ sec—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.

<u>Command</u>	<u>Description</u>
<b>BCTG0</b>	Set the derivative sampling period to 266 $\mu$ sec
<b>BCTG1</b>	Set the derivative sampling period to 532 $\mu$ sec
<b>BCTG2</b>	Set the derivative sampling period to 798 $\mu$ sec
<b>BCTG3</b>	Set the derivative sampling period to 1064 $\mu$ sec
<b>1BCTG</b>	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.

<u>Command</u>	<u>Response</u>
<b>1BS</b>	*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.

<b>Command</b>	<b>Description</b>
<b>MC</b>	Sets move to continuous mode
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V1Ø</b>	Sets velocity to 10 revs/sec
<b>PS</b>	Pauses system until controller receives <b>C</b> command
<b>G</b>	Accelerates the motor to 10 revs/sec
<b>C</b>	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.

<b>Command</b>	<b>Description</b>
<b>CDG400</b>	Set the derivative gain term to 400
<b>1CDG</b>	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.

<b>Command</b>	<b>Description</b>
<b>CEW10</b>	Configure an In Position Error Window $\pm 10$ encoder counts either side of desired position
<b>1CEW</b>	Reports $\pm$ 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. Integral gain provides a torque that is directly proportional to the sum, over time, of the error values—the *integral* of the error. The controller reads the error value every sampling period, and adds it to the sum of all previous error values. The sum is multiplied by the value set by the **CIG** command (Integral \* CIG), resulting in the *integral term* which contributes to the motor control output every system sampling period. This gain will improve overall accuracy but may increase settling time and, if excessive, may cause a low frequency oscillation around the commanded position.

Refer to *Chapter ④—Tuning* for more information.

<b>Command</b>	<b>Description</b>
<b>CIL40</b>	Set the integral limit term to 40
<b>CIG10</b>	Set the integral gain term to 10
<b>1CIG</b>	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. Since integral windup can sometimes cause very aggressive motion, you may want to limit this effect. CIL sets an upper limit on the integral, which in turn limits the integral term (**Integral \* CIG**).

Refer to *Chapter ④—Tuning* for more information.

<u>Command</u>	<u>Description</u>
<b>CIG10</b>	Set the integral gain term to 10
<b>CIL40</b>	Set the integral limit to 40
<b>1CIL</b>	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.

<u>Command</u>	<u>Description</u>
<b>1SSC1</b>	Set output 1 as "In Position"
<b>1CIT30</b>	Set "In Position" time to 30ms
<b>1CEW20</b>	Set allowable position error to $\pm 20$ 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 TQ10X will generate a fault condition and shut down power output to the motor. If the maximum allowable position error is set to  $\emptyset$ , the function is disabled and no amount of position error will generate the fault condition.

Refer to *Chapter ④—Tuning* for more information.

<u>Command</u>	<u>Description</u>
<b>CPE400</b>	Set the maximum allowable position error to 400 encoder counts
<b>CPE0</b>	Disable fault generation due to position error
<b>1CPE</b>	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.

<u>Command</u>	<u>Description</u>
<b>CPG50</b>	Set the proportional gain term to 50
<b>1CPG</b>	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.

<u>Command</u>	<u>Description</u>
<b>MN</b>	Sets mode to preset mode
<b>A5Ø</b>	Sets acceleration to 50 revs/sec <sup>2</sup>
<b>V5</b>	Sets Velocity to 5 revs/sec
<b>D5ØØØ</b>	Sets distance to 5,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>1CR</b>	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.

<b>Command</b>	<b>Description</b>
<b>CTG0</b>	Set the derivative sampling period to 266 $\mu$ sec
<b>CTG1</b>	Set the derivative sampling period to 532 $\mu$ sec
<b>CTG2</b>	Set the derivative sampling period to 798 $\mu$ sec
<b>CTG3</b>	Set the derivative sampling period to 1064 $\mu$ sec
<b>1CTG</b>	Reports derivative sampling period (*CTG3)

---

## D—Distance

- |   |   |
|---|---|
| <input type="checkbox"/> Command Type: Motion           | <input type="checkbox"/> Valid Software Version: A          |
| <input type="checkbox"/> Syntax: <a>Dn                  | <input type="checkbox"/> Units: encoder counts              |
| <input type="checkbox"/> Range: n = $\pm 1,073,741,823$ | <input type="checkbox"/> Default Value: 4000                |
| <input type="checkbox"/> Attributes: Buffered           | <input type="checkbox"/> Response to aD is *Dn              |
| <input type="checkbox"/> Savable in Sequence            | <input type="checkbox"/> 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.

<b>Command</b>	<b>Description</b>
<b>MN</b>	Sets controller to Normal mode
<b>MPI</b>	Sets controller to Incremental Position mode
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V1Ø</b>	Sets velocity to 10 revs/sec
<b>D4ØØØ</b>	Sets distance to 4000 encoder counts
<b>G</b>	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:  $n = \pm 1,073,741,823$
- 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**).

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

---

## DPE—Display Position Error

- Command Type: Status
- Syntax: aDPE
- Range:  $n = \pm 2,147,483,646$
- 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.

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

---

## DVA—Display Velocity Actual

- Command Type: Status
- Syntax: aDVA
- Range:  $n = 1-20000$
- 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.

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

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

## ER—Encoder Resolution

- |   |  |
|---|--|
| <input type="checkbox"/> Command Type: Set-up                         | <input type="checkbox"/> Valid Software Version: A     |
| <input type="checkbox"/> Syntax: <a>ERn                               | <input type="checkbox"/> Units: n = encoder counts/rev |
| <input type="checkbox"/> Range: n = 400 - 65,532                      | <input type="checkbox"/> Default Value: 4000           |
| <input type="checkbox"/> Attributes: Buffered,<br>Savable in Sequence | <input type="checkbox"/> Response to aER is *ERn       |
|   | <input type="checkbox"/> 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)

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

## F—Disable Communications

- |   |  |
|---|--|
| <input type="checkbox"/> Command Type: Programming            | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>F                         | <input type="checkbox"/> Units: N/A                |
| <input type="checkbox"/> Range: N/A                           | <input type="checkbox"/> Default Value: None       |
| <input type="checkbox"/> Attributes: Immediate<br>Never Saved | <input type="checkbox"/> 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.

<b>Command</b>	<b>Description</b>
<b>1F</b>	Disables the communications interface on unit #1
<b>3F</b>	Disables the communications interface on unit #3
<b>G</b>	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.

<u>Command</u>	<u>Description</u>
<b>MN</b>	Sets Normal mode (preset)
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V10</b>	Sets velocity to 10 revs/sec
<b>D2000</b>	Sets distance to 2,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>A1</b>	Sets acceleration to 1 rev/sec <sup>2</sup>
<b>G</b>	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<sup>2</sup> (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 from 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.

<b>Command</b>	<b>Description</b>
<b>INE1</b>	Configure input #1 as home input
<b>OSB1</b>	Back up the home switch
<b>OSD1</b>	Reference Z channel as final home
<b>GH-2</b>	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**

- |  |  |
|--|--|
| <input type="checkbox"/> Command Type: Programming | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: ^H                | <input type="checkbox"/> Units: N/A                |
| <input type="checkbox"/> Range: N/A                | <input type="checkbox"/> Default Value: None       |
| <input type="checkbox"/> Attributes: Immediate     |  |
| Never Saved  |  |

This command allows you to delete the last character that you entered. The **^H** command will not prevent execution of an immediate command. A new character may be entered at that position to replace the existing character. (**^H** indicates that the Ctrl key is held down when the H key is pressed.) This command prompts the 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
<b>MN</b>	Sets Normal mode
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4000</b>	Sets distance to 4,000 encoder counts
<b>G</b>	Executes the move (Go) in CW direction
<b>H</b>	Reverses direction
<b>G</b>	Executes the move (Go) in CCW direction
<b>MC</b>	Sets mode to continuous
<b>H+</b>	Sets direction to CW
<b>G</b>	Moves continuously in CW direction

---

## IN—Set Input Function

- Command Type: Set-up
- Syntax: aINxn
- Range: n = 1 – 5, x = A – F
- 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 **SSH0**). 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.

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

## IS—Input Status

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

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

- 1—IN1 (0 = Low, 1 = High)
- 2—IN2 (0 = Low, 1 = High)
- 3—IN3 (0 = Low, 1 = High)
- 4—IN4 (0 = Low, 1 = High)
- 5—IN5 (0 = Low, 1 = High)
- 6—CW limit (0 = Low, 1 = High)
- 7—CCW limit (0 = Low, 1 = High)
- 8—Fault (0 = 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.

<b>Command</b>	<b>Response</b>
<b>2IS</b>	*00010001 (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.

---

<b>Command</b>	<b>Description</b>
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V2</b>	Sets velocity to 2 revs/sec
<b>MC</b>	Sets mode to continuous
<b>G</b>	Executes the move (Go)
<b>K</b>	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  $\emptyset$ , 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.

<b>Command</b>	<b>Description</b>
<b>L5</b>	Loop 5 times
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V1<math>\emptyset</math></b>	Sets velocity to 10 revs/sec
<b>D1<math>\emptyset\emptyset\emptyset\emptyset</math></b>	Sets distance to 10,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>N</b>	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 CCW limit—*n = 2*
- Disable CCW and CW limits—*n = 3*

<u>Command</u>	<u>Description</u>
<b>1LDØ</b>	Enables CW and CCW limits. The motor will move only if the limit inputs are bypassed or connected to normally-closed limit switches.
<b>1LD3</b>	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.

<u>Command</u>	<u>Description</u>
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D15ØØØ</b>	Sets distance to 15,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>1LF</b>	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.

<u>Command</u>	<u>Description</u>
<b>MC</b>	Sets mode to continuous
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>G</b>	Executes the move (Go)
<b>T10</b>	Move at 5 revs/sec for 10 seconds
<b>V7</b>	Set velocity to 7 revs/sec
<b>G</b>	Change velocity to 7 revs/sec
<b>T10</b>	Move at 7 revs/sec for 10 seconds
<b>V0</b>	Set velocity to 0 rev/sec (stop)
<b>G</b>	Executes the <b>V0</b> 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**).

<u>Command</u>	<u>Description</u>
<b>MN</b>	Set positioning mode to preset
<b>A5</b>	Set acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Set velocity to 5 revs/sec
<b>D1000</b>	Set distance to 1,000 encoder counts
<b>G</b>	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.

<u>Command</u>	<u>Description</u>
<b>MN</b>	Sets Normal mode (preset)
<b>PZ</b>	Resets absolute counter to zero
<b>MPA</b>	Sets position mode absolute
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V10</b>	Sets velocity to 10 revs/sec
<b>D40000</b>	Sets destination to absolute position 40,000
<b>G</b>	Motor will move to absolute position 40,000
<b>D10000</b>	Sets destination to absolute position +10,000
<b>G</b>	Motor will move to absolute position +10,000

The motor will move 40,000 encoder counts in the CW direction (if starting from position 0) 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.

<u>Command</u>	<u>Description</u>
<b>MN</b>	Set positioning mode normal (preset)
<b>MPI</b>	Set positioning mode incremental
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V10</b>	Sets velocity to 10 revs/sec
<b>D10000</b>	Sets distance of move to 10,000 encoder counts
<b>G</b>	Move 10,000 encoder counts CW
<b>G</b>	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  
Never Saved
- 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

<u>Command</u>	<u>Description</u>
<b>MN</b>	Sets move to Normal mode
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D10000</b>	Sets move distance to 10,000 encoder counts
<b>L5</b>	Loops the following commands five times
<b>G</b>	Executes the move (Go)
<b>N</b>	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 immediately disables the drive. This command can be used to shut down the drive in an emergency. This command is functionally identical to the ST1 command.

<u>Command</u>	<u>Description</u>
<b>OFF</b>	De-energize the drive
<b>1IS</b>	*00000001 (fault bit active)

---

## ON—Energize Drive

- Command Type: Programming
- Syntax: <a>ON
- 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*.

<u>Command</u>	<u>Description</u>
<b>ON</b>	Energize the drive
<b>1IS</b>	*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.

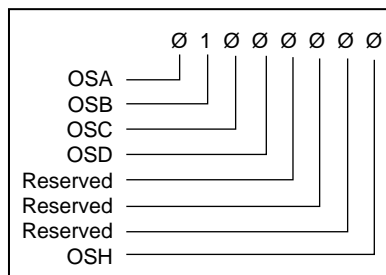
- n=1** = Turns output bits on
- n=Ø** = Turns output bits off
- n=X** = Leaves output bits unchanged

<u>Command</u>	<u>Description</u>
<b>MN</b>	Set to Mode Normal
<b>A1Ø</b>	Set acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D2ØØØØ</b>	Set move distance to 20,000 encoder counts
<b>OØ1</b>	Set programmable output 1 off and output 2 on
<b>G</b>	Executes the move (Go)
<b>OØØ</b>	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.

<u>Command</u>	<u>Description</u>
<b>OSA1</b>	Sets active state for normally open limit switches
<b>OSCØ</b>	Sets active state of home input closed (low)
<b>OSH1</b>	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.

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

## OSC—Define Active State of Home Switch

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

Command	Description
<b>OSC1</b>	Sets the active state of the home input to open

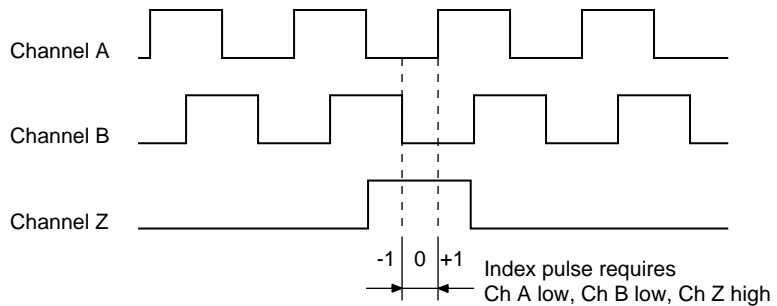
## OSD—Enable Encoder Z Channel for Home

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

**OSD0** = Do not reference Z Channel during homing

**OSD1** = Reference Z Channel during homing

The encoder Z channel is used (in conjunction with a load activated switch connected to the home limit) to determine the home position. The switch determines the home region, and the Z channel determines the exact 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.

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

---

## OSH—Reference Edge of Home Switch

- Command Type: Set-Up
- Syntax: <a>OSHn
- Range: n = Ø, 1
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: NA
- Default Value: Ø
- 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>
<b>OSB1</b>	Sets back up to home switch active
<b>OSCØ</b>	Sets active state of home input closed (low)
<b>OSH1</b>	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

- Command Type: Status
- Syntax: aPR
- Range: ±1,073,741,820
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: Encoder counts
- Default Value: N/A
- See Also: D, MPA, MPI, MN, PZ, PX
- 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>
<b>1PR</b>	Commanded position report. (*+ØØØØØØØ2ØØØ)
<b>1PX</b>	Encoder position report (*+ØØØØØØØ2ØØØ5)

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>
<b>PS</b>	Pauses execution of commands until the controller receives the Continue (C) command
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4000</b>	Sets move distance to 4,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>T2</b>	Delays the move for 2 seconds
<b>G</b>	Executes the move (Go)
<b>C</b>	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>
<b>MN</b>	Set to mode normal
<b>PZ</b>	Sets the absolute counter to zero
<b>A10</b>	Sets acceleration to 10 rev/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 rev/sec
<b>D4600</b>	Sets move distance to 4,600 encoder counts
<b>G</b>	Executes the move (Go)
<b>1PX</b>	After the motor executes the move, the encoder position is reported: The response is *+0000004600.

---

## PZ—Set Absolute Counter to Zero

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

This command sets the absolute 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>
<b>MN</b>	Enter position mode (mode normal)
<b>MPA</b>	Makes preset moves from absolute zero position
<b>PZ</b>	Sets absolute position counter to zero
<b>A1Ø</b>	Sets acceleration to 10 rev/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 rev/sec
<b>D4ØØØØ</b>	Sets move distance to 40000 encoder counts
<b>G</b>	Executes the move (Go)
<b>1PX</b>	Reports absolute encoder position (*+0000040000)
<b>PZ</b>	Sets the absolute counter to zero
<b>1PX</b>	Reports absolute encoder position (*+0000000000)

---

## “—Quote

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

Up to 17 characters entered after the quotation marks (“) will be transmitted, exactly as they are entered, over the RS-232C link. A space entered by the space bar indicates the end of the command. A space is always sent after the last character in the string. This command is used during buffered moves or sequences to command other devices to move, or to send the message to a remote display. 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>
<b>PS</b>	Pause execution until Continue (C) is entered
<b>A5</b>	Set acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Set velocity to 5 revs/sec
<b>D2ØØØØ</b>	Set distance to 2,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>“MOVE_DONE</b>	Transmits message
<b>C</b>	Continue move

After the move, the TQ10X will send the message MOVE\_DONE via the RS-232C port

---

## QØ—Exit Velocity Profiling Mode

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

The **QØ** command exits the Velocity Profiling mode. The motor will stop when **QØ** is issued. Entering this command will cause the TQ10X to enter Normal mode (**MN**).

---

## Q1—Enter Velocity Profiling Mode

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

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

<u>Command</u>	<u>Description</u>
<b>ER2ØØØ</b>	Set encoder resolution to 2000
<b>Q1</b>	Enter Velocity Streaming mode
<b>RMØØØØ22ØC</b>	Accelerate to 0.25 revs/sec <sup>2</sup>
<b>RMØØØØ4418</b>	Accelerate to 0.5 revs/sec <sup>2</sup>
<b>RMØØØØ8831</b>	Accelerate to 1 revs/sec <sup>2</sup>
<b>RMØØØ11Ø62</b>	Accelerate to 2 revs/sec <sup>2</sup>
<b>RMØØØØ8831</b>	Decelerate to 1 revs/sec <sup>2</sup>
<b>RMØØØØ4418</b>	Decelerate to 0.5 revs/sec <sup>2</sup>
<b>RMØØØØ22ØC</b>	Decelerate to 0.25 revs/sec <sup>2</sup>
<b>RMØØØØØØØØ</b>	Decelerate to 0 revs/sec <sup>2</sup>
<b>QØ</b>	Exit Velocity Streaming mode

---

## R—Request Controller Status

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

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

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

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

## RA—Limit Switch Status Request

- Command Type: Status
- Syntax: aRA
- Range: N/A
- Attributes: Immediate,  
Never Saved
- Valid Software Version: A
- Units: N/A
- Default Value: N/A
- See Also: R, RB
- Response to aRA is \*x

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

<i>Response Character</i>	<i>Last Move Terminated By</i>		<i>Current Limit Status</i>	
	<i>CW Limit</i>	<i>—CCW Limit</i>	<i>CW Limit</i>	<i>—CCW Limit</i>
*@	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>
<b>1RA</b>	*@ (the last move was not terminated by a limit and no limits are currently active.)

---

## RB—Loop, Pause, Shutdown, Trigger Status Request

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

This command receives a response from \*@ to \*O, as defined below. The four conditions for which status is indicated are as follows:

**Loop Active:** A loop is in progress.

**Pause Active:** Buffered commands waiting for a Continue (C).

**Shutdown Active:** The motor is shutdown by the ST1 command.

**Trigger Active:** At least one trigger is active.

<i>Response Character</i>	<i>Loop Active</i>	<i>Pause Active</i>	<i>Shutdown Active</i>	<i>Trigger Active</i>
*@	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

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

The RC command has the same response format of RA and RB. The 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.

<b>Response Character</b>	<b>Go Home Successful?</b>
*@	YES
*B	NO

<b>Command</b>	<b>Description</b>
<b>1RC</b>	*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,<br>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	( <b>CPG</b> )	16
Integral Gain	( <b>CIG</b> )	2
Derivative Gain	( <b>CDG</b> )	240
Derivative Sampling Period	( <b>CTG</b> )	0
Position Error	( <b>CPE</b> )	4000
Integral Limit	( <b>CIL</b> )	2

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

## RM—Rate Multiplier in Velocity Streaming

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

$( \text{desired revs/sec} ) \cdot ( \text{encoder resolution} ) * 17.432576 = \text{decimal \# for velocity value to be rounded off to the closest whole number.}$

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 TQ10X 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 TQ10X to enter Normal mode (**MN**).

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

## RSE—Report Servo Errors

- Command Type: Status
- Syntax: aRSE
- Range: N/A
- Attributes: Immediate, Never Saved
- Valid Software Version: A
- Units: N/A
- Default Value: N/A
- See Also: R, RA, RB, RC
- 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>
<b>*0</b>	No errors
<b>*2</b>	Excessive position error
<b>*4</b>	Drive fault
<b>*6</b>	Commanded shutdown
<b>*8</b>	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 **NOT ENABLED** 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
- Drive was recently enabled

<u>Command</u>	<u>Response</u>
<b>1RSE</b>	<b>*0</b> (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,<br>Savable in Sequence | <input type="checkbox"/> See Also: D, H, QØ, Q1            |
|   | <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>
<b>1RV</b>	<b>92-Ø16637-Ø1A</b>

The product is identified by \*92-Ø16637-Ø1A, 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,<br>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>
<b>MC</b>	Sets move in continuous mode
<b>A1</b>	Sets acceleration to 1 revs/sec <sup>2</sup>
<b>V1Ø</b>	Sets velocity to 10 revs/sec
<b>G</b>	Executes the move (Go)
<b>S</b>	Stops motor (motor comes to a stop at a deceleration rate of 1 revs/sec <sup>2</sup> )

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

---

## SN—Scan

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

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

This command allows you to select the best possible trade-off between noise immunity and speed for a given application. If you make your scan time too short, the TQ10X 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**).

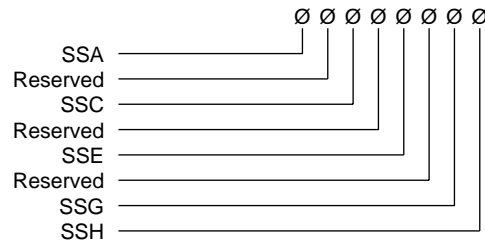
<u>Command</u>	<u>Description</u>
<b>SN1Ø</b>	Sets scan time of sequence select inputs to 10 ms

---

## SS—Software Switch Function Status

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

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



---

## SSA—RS-232C Echo Control

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

This command turns the RS-232C echo (transmission of characters received from the remote device by the TQ10X) 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 TQ10X. 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>
<b>SSA1</b>	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
- Syntax: aSSCn
- Range: n = 0 or 1
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: N/A
- Default Value: 0
- See Also: CEW, CIT
- Response to aSSC is \*SSCn

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>
<b>SSC1</b>	Set output 1 as an In Position output
<b>1SSC</b>	Reports output 1 configuration (*SSC1)

---

## SSE—Enable/Disable Communication Error Checking

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

The 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
- Syntax: <a>SSGn
- Range: n = Ø, 1
- Attributes: Buffered, Savable In Sequence
- Valid Software Version: A
- Units: See Below
- Default Value: Ø
- See Also: LD

**SSGØ** = Clears command buffer on limit

**SSG1** = Saves command buffer on limit

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

current move is aborted, but the remaining commands in the buffer continue to be executed.

<u>Command</u>	<u>Description</u>
<b>SSG1</b>	Saves buffer on limit
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4ØØØ</b>	Sets distance to 4,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>O11</b>	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,<br>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>
<b>SSHØ</b>	Clears buffer on stop
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4ØØØ</b>	Sets distance to 4,000 encoder counts
<b>L5Ø</b>	Loops 50 times
<b>G</b>	Executes the move (Go)
<b>T.5</b>	Pauses the motor 500 ms
<b>N</b>	Ends Loop
<b>S</b>	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,<br>Savable in Sequence | <input type="checkbox"/> See also: OFF, ON         |

The ST1 command immediately disables 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 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*.

<u>Command</u>	<u>Description</u>
<b>ST1</b>	Disable the drive
<b>ST0</b>	Enable the drive

---

## T—Time Delay

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

The Time (**T**) command causes the 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.

<u>Command</u>	<u>Description</u>
<b>MN</b>	Sets Normal mode
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4000</b>	Sets distance to 4,000 encoder counts
<b>T10</b>	Pauses motor movement 10 seconds
<b>G</b>	Executes the move (Go)
<b>T5</b>	Pauses the motor for 5 seconds after the move
<b>G</b>	Executes the move (Go)

---

## TR—Wait For Trigger

- Command Type: Programming
- Syntax: <a>TRnnnnn
- Range: n = 0, 1, or X
- Attributes: Buffered,  
Savable in Sequence
- Valid Software Version: A
- Units: See Below
- Default Value: None
- See Also: IN

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.

<b>Command</b>	<b>Description</b>
<b>IN3A</b>	Configure input 3 as trigger input 1
<b>IN5A</b>	Configure input 5 as trigger input 2
<b>TR1Ø</b>	Wait for input 1 to be opened and input 2 to be grounded before going on to the next command
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4ØØØ</b>	Sets distance to 4,000 encoder counts
<b>G</b>	Executes the move (Go)

Motion will not occur until trigger conditions are true.

## U—Pause and Wait for Continue

- |   |  |
|---|--|
| <input type="checkbox"/> Command Type: Programming          | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>U                       | <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: 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.

<b>Command</b>	<b>Description</b>
<b>MN</b>	Sets move to Normal mode
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>LØ</b>	Loops indefinitely
<b>D46ØØ</b>	Sets distance to 4,600 encoder counts
<b>G</b>	Executes the move (G)
<b>T1Ø</b>	Waits 10 seconds after the move
<b>N</b>	Ends loop
<b>U</b>	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

- |  |  |
|--|--|
| <input type="checkbox"/> Command Type: Motion                      | <input type="checkbox"/> Valid Software Version: A |
| <input type="checkbox"/> Syntax: <a>Vn                             | <input type="checkbox"/> Units: revs/sec           |
| <input type="checkbox"/> Range: n = 0.01 - 200.00                  | <input type="checkbox"/> Default Value: 1          |
| <input type="checkbox"/> Attributes: Buffered, Savable in Sequence | <input type="checkbox"/> 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 TQ10X 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**).*

<b>Command</b>	<b>Description</b>
<b>MC</b>	Sets move to continuous
<b>A5</b>	Sets acceleration to 5 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>G</b>	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,<br>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 TQ10X is not re-programmed, the checksum response should always be the same.

<b>Command</b>	<b>Response</b>
<b>1XC</b>	*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,<br>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 256 characters. Immediate commands cannot be entered into a sequence.

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

---

## XE—Sequence Erase

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

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

<u>Command</u>	<u>Description</u>
<b>XE1</b>	Deletes Sequence 1
<b>XD1</b>	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 TQ10X 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, an enable input is asserted, or the unit is powered down. The possible settings for this command are:

- 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 TQ10X from selecting the next sequence until all the sequence select inputs are first opened.

<u>Command</u>	<u>Description</u>
<b>XP1</b>	Executes Sequence #1 on power-up
<b>XE1</b>	Erases Sequence #1
<b>XD1</b>	Defines Sequence #1
<b>LD3</b>	Disables CW & CCW limits
<b>A10</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4000</b>	Sets distance to 4,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>XT</b>	Ends definition of Sequence #1
<b>Z</b>	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
- Syntax: <a>XQn
- Range: n = Ø, 1
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: Sequences
- Default Value: Ø
- See Also: XP

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

This command can be used only when the TQ10X is stand-alone power-up sequencing in **XP9** mode. In **XP9** mode, if **XQ1** is executed, the TQ10X 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 **XQØ** command is executed. You may use **S** or **K** command to stop sequence execution. **XQ1** must be the first command entered in the sequence.

<b>Command</b>	<b>Description</b>
<b>XE1</b>	Erases sequence #1
<b>XD1</b>	Defines sequence #1
<b>XQ1</b>	Turns Interrupted Run mode on
<b>LD3</b>	Disables CW & CCW limits
<b>XT</b>	Ends Sequence #1
<b>XP9</b>	Sets power-up sequences as sequence select inputs
<b>Z</b>	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
- Syntax: <a>XRn
- Range: n = 1 - 7
- Attributes: Buffered, Savable in Sequence
- Valid Software Version: A
- Units: Sequence
- Default Value: 0
- See Also: XD, XE, XRP, XT

This command loads a pre-defined sequence into the command buffer (clears the buffer first) and executes these commands as a normal set of commands. **XR** automatically recalls the sequence from BBRAM. Before executing the specified sequence, the TQ10X is put into the power-on default state with respect to the following conditions:

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

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

<b>Command</b>	<b>Description</b>
<b>XE1</b>	Erases sequence #1
<b>XD1</b>	Defines sequence #1
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets acceleration to 5 revs/sec
<b>D1ØØØØ</b>	Sets distance to 10,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>XT</b>	Ends Sequence #1 definition
<b>XR1</b>	Executes Sequence #1

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

---

## XRP—Sequence Run With Pause

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

This command is identical to the Sequence Run (**XR**) command, except that it automatically generates a pause condition. You must clear this condition with the Continue (**C**) command before the 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.

<u>Command</u>	<u>Description</u>
<b>XE5</b>	Erases Sequence #5
<b>XD5</b>	Defines Sequence #5
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D1ØØØØ</b>	Sets distance to 10,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>XT</b>	Ends definition of Sequence #5
<b>XRP5</b>	Runs Sequence #5 with a pause
<b>C</b>	Indexer executes Sequence #5

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

---

## XSD—Sequence Status Definition

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

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

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

**XSD** verifies that the last sequence definition was successful.

<u>Command</u>	<u>Response</u>
<b>1XSD</b>	*1 (A sequence already exists as sequence 1)

---

## XSP—Sequence Status Power-up

- Command Type: Status
- Syntax: aXSP
- Range: N/A
- Attributes: Immediate, 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  $\emptyset$  - 9.

<u>Command</u>	<u>Description</u>
<b>1XSP</b>	*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** = Valid sequence
- \* **3** = Erased
- \* **4** = Bad checksum
- \* **5** = Running
- \* **6** = Killed, stopped, sequence never estimated

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

---

## XSS—Sequence Status

- Command Type: Status
- Syntax: aXSSn
- Range: n = 1 - 7
- Attributes: Immediate, Never Saved
- Valid Software Version: A
- Units: Sequences
- Default Value: None
- See Also: XD, XE, XT
- 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>
<b>1XSS1</b>	* $\emptyset$ (Sequence #1 of device 1 is not defined.)

---

## XT—Sequence Termination

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

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

**NOTE:** In your communication program, use sufficient time delays after downloading a sequence before you send more commands to the TQ10X. In particular, after sending the **XT** command, wait at least 12.5 msec before sending a **Z** command.

<u>Command</u>	<u>Description</u>
<b>XE1</b>	Erases Sequence #1
<b>XD1</b>	Defines Sequence #1
<b>MN</b>	Sets to Normal mode
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4ØØØ</b>	Sets distance to 4,000 encoder counts
<b>G</b>	Executes the move (Go)
<b>XT</b>	Ends sequence definition

---

## XU—Upload Sequence

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

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

When using a daisy-chain, **XU** must be used cautiously as the contents of the sequence will go to all controllers in the loop between the 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>
<b>2F</b>	Turns off communication to unit #2
<b>3F</b>	Turns off communication to unit #3
<b>1XU1</b>	Uploads sequence #1 from unit #1

---

## Y—Stop Loop

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

The Stop Loop (**Y**) command takes you out of a loop when the loop completes its current pass. This command does not halt processing of the commands in the loop until the 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.

<b>Command</b>	<b>Description</b>
<b>L</b>	Loops indefinitely
<b>A1Ø</b>	Sets acceleration to 10 revs/sec <sup>2</sup>
<b>V5</b>	Sets velocity to 5 revs/sec
<b>D4ØØØ</b>	Sets distance to 4,000 encoder counts
<b>T2</b>	Waits 2 seconds
<b>G</b>	Executes the move (Go)
<b>N</b>	Ends loop
<b>Y</b>	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 <b>Y</b> command.

## Z—Reset

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

The Reset (**Z**) command is equivalent to cycling DC power to the 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.

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

## #—Address Numbering

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

This command sets the individual unit address for each TQ10X, allowing addresses up to 255. Upon receipt of the command, the TQ10X 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.

<b>Command</b>	<b>Description</b>
<b>#1</b>	#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: Set-Up
- Syntax: %
- Range: N/A
- Attributes: Immediate, Never Saved
- Valid Software Version: A
- Units: N/A
- Default Value: N/A
- See Also: E,F,SSE

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

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

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

Possible responses are:

<u>Character</u>	<u>Definition</u>
*	No errors
*0	Unit upstream (daisy chained)
*1	Overrun, data received to 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, 6th unit detected an overrun error, and the last 3 units turned communication off because of unit 6)

## Summary of Commands

---

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

