

PS Programming	Pause			VALID Software Version E2
SYNTAX <a>PS	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO C, U		

Description

This command pauses execution of a command string or sequence following the Pause (**PS**) command until the indexer receives a Continue (**C**) command. This command is useful if you need to enter a complete string of commands before you can execute your other commands.

This command is useful for interactive tests and in synchronizing multiple indexes that have long command strings.

Example

<u>Command</u>	<u>Description</u>
P S	Pauses execution of following commands until the indexer receives the Continue (C) command
A 5	Sets acceleration to 5 rev/sec ²
V 5	Sets velocity to 5 rev/sec
D25600	Sets move distance to 25,600 steps
G	Executes the move (Go)
T 2	Delays the move for 2 sec
G	Executes the move (Go)
C	Continues Execution

When the indexer receives the **C** command, the motor moves 25,600 steps twice with a 2 second delay between them.

PX Status	Report Absolute Encoder Position			VALID Software Version E2
SYNTAX aPX	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO W3, PR, FSB		
RESPONSE TO aPX IS ±nnnnnnnnnn				

Description

This command returns a decimal value indicating the absolute position of the incremental encoder. The absolute position is based on the zero position. The zero position is established when you power up the system. The zero position can also be established after the indexer performs a Position Zero (**PZ**) command. Whether in Motor Step mode or Encoder Step mode, the position is reported in encoder steps.

The range of the response is 0 - ±9,999,999,999.

This command is useful in the following situations:

- Encoder Set-up
- Closing the loop with the host though positioning with n steps
- End of move (verification of position)

Example

<u>Command</u>	<u>Description</u>
MN	Presets mode
PZ	Sets the absolute counter to zero
A10	Sets acceleration to 10 rev/sec ²
V5	Sets velocity to 5 rev/sec
D25600	Sets move distance to 25,600 steps
G	Executes the move (Go)
FSB1	Sets indexer to encoder step mode
1PX	After the motor executes the move, the encoder position is reported: The response is *+0000008000, assuming the ER command is set to 4000.

PZ Set-up	Set Absolute Counter to Zero			VALID Software Version E2
SYNTAX <a>PZ	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO MN, MPI, MPA, PR, D, PX		

Description This command sets the absolute position counter to zero. If the motor does not move after the last **PZ** command, the response to the **PR** and **PX** commands will be 0s.

Example	<u>Command</u> MPA A1Ø V5 D256ØØ G 1PR PZ 1PR	<u>Description</u> Make all preset moves with respect to absolute zero position Set Acceleration to 10 rev/sec ² Set Velocity to 5 rev/sec Set move distance to 25,600 steps Executes the move (Go) Report Absolute Position (Response = *+0000025600) Sets the absolute counter to zero Report absolute position (Response = *+0000000000)
----------------	---	--

" Programming	Quote			VALID Software Version E2
SYNTAX <a>"x	UNITS x = characters	RANGE up to any 12 ASCII characters	DEFAULT None	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME		SEE ALSO CR		
RESPONSE TO <a>"x IS x				

Description Any characters entered after the quotation marks (") (up to 12 characters) will be transmitted, exactly as they were 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, or to command other Compumotor devices to move.

Example 1

<u>Command</u>	<u>Description</u>
MN	Set to mode normal (Preset Moves)
A1Ø	Set acceleration to 10 rev/sec ²
V5	Set velocity to 5 rev/sec
D128ØØ	Set distance to 12,800 steps
G	Executes the move (Go)
"MOVE_DONE	After motor finished the move, the Compumotor Indexer will send the message MOVE_DONE out from the RS-232C port.

Example 2

<u>Command</u>	<u>Description</u>
MN	Set to mode normal (Preset Moves)
A1Ø	Set acceleration to 10 rev/sec ²
V5	Set velocity to 5 rev/sec
D128ØØ	Set distance to 12,800 steps
G	Executes the move (Go)
"2XR1	Once the move is done, Run Sequence 1 is commanded on a unit with device address 2.

QØ Set-up	Exit Velocity Profiling Mode			VALID Software Version E2
SYNTAX <a>QØ	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO Q1, RM		

Description The **gØ** command exits the velocity profiling mode. The motor will stop when **gØ** is issued.

Example See **Q1** example

Q1 Set-up	Enter Velocity Profiling Mode			VALID Software Version E2
SYNTAX <a>Q1	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO Q0, RM		

Description The **Q1** command enters the indexer in velocity profiling mode. Subsequent **RM** commands will cause an immediate change in motor velocity. Use **Q0** to exit this mode.

Example	<u>Command</u>	<u>Description</u>
	Q 1	Enter Velocity streaming mode
	RM0190	Accelerate to 1 rev/sec
	RM0320	Accelerate to 2 revs/sec
	RM0460	Accelerate to 3 revs/sec
	RM0640	Accelerate to 4 revs/sec
	RM0460	Decelerate to 3 revs/sec
	RM0320	Decelerate to 2 revs/sec
	RM0190	Decelerate to 1 rev/sec
	RM0000	Decelerate to 0 revs/sec
	Q 0	Exit velocity streaming mode

Motor movement will stop when **Q0** command is entered.

R Status	Request Indexer Status			VALID Software Version E2
SYNTAX aR	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO RA, RB, RC		
RESPONSE TO aR IS *x				

Description The Request Indexer Status (**R**) command can be used to indicate the general status of the indexer. Possible responses are as follows:

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

The following conditions will cause a response indicating that the indexer is busy:

- Performing a preset move
- Accelerating/decelerating during a continuous move
- A time delay is in progress. (**T** command)
- In **RM** mode
- Paused
- Waiting on a Trigger
- Going Home
- Running a sequence
- Executing a loop

The following conditions will cause a response indicating that an error exists.

- A feedback error condition exists.
- Go home failed
- Limit has been encountered
- Sequence execution was unsuccessful

When the response indicates that attention is required, more details on the error condition are available by using the **RA**, **RB**, or **RC** commands.

It is not recommended that this command be used in tight polling loops which could result in microprocessor over load. Time delays can alleviate this problem.

This command is not intended to be used to determine if a move is complete. Rather, it should be used after the move is complete to determine if there might be other errors or faults.

Use a buffered status request command or a programmable output to indicate move completion.

Example

<u>Command</u>	<u>Response</u>
1R	*R (Indexer ready to accept a command and no error conditions require attention)

RA Status	Limit Switch Status Report			VALID Software Version E2
SYNTAX aRA	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO R, RB		
RESPONSE TO aRA IS *x				

Description

The Limit Switch Status Report (**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 16 characters representing the conditions listed below.

Response Character	Last Move Terminated by		Current Limit Status	
	CW Limit	CCW Limit	CW Limit	CCW Limit
	*@	NO	NO	OFF
*A	YES	NO	OFF	OFF
*B	NO	YES	OFF	OFF
*C	YES	YES	OFF	OFF
*D	NO	NO	ON	OFF
*E	YES	NO	ON	OFF
*F	NO	YES	ON	OFF
*G	YES	YES	ON	OFF
*H	NO	NO	OFF	ON
*I	YES	NO	OFF	ON
*J	NO	YES	OFF	ON
*K	YES	YES	OFF	ON
*L	NO	NO	ON	ON
*M	YES	NO	ON	ON
*N	NO	YES	ON	ON
*O	YES	YES	ON	ON

The **RA** command is useful when the motor will not move in either or both directions. The report back will indicate whether or not the last move was terminated by one or both end-of-travel limits.

NOTE: This command is not intended to be used to determine if a move is complete. Rather, it should be used after the move is complete to determine if there might be other errors or faults.

Example

Command
1RA

Response

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

RB Status	Loop, Pause, Shutdown, Trigger Status Report			VALID Software Version E2
SYNTAX aRB	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Not Saved
EXECUTION TIME <2mS		SEE ALSO R, RA, TR, PS, L, ST		
RESPONSE TO aRB IS *x				

Description

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 are not being executed due to a Pause (PS) command (waiting for a C command).

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

Trigger Active: At least one trigger is active.

Response Character	Loop Active	Pause Active	Shutdown Active	Trigger Active
*@	NO	NO	NO	NO
*A	YES	NO	NO	NO
*B	NO	YES	NO	NO
*C	YES	YES	NO	NO
*D	NO	NO	YES	NO
*E	YES	NO	YES	NO
*F	NO	YES	YES	NO
*G	YES	YES	YES	NO
*H	NO	NO	NO	YES
*I	YES	NO	NO	YES
*J	NO	YES	NO	YES
*K	YES	YES	NO	YES
*L	NO	NO	YES	YES
*M	YES	NO	YES	YES
*N	NO	YES	YES	YES
*O	YES	YES	YES	YES

NOTE: This command is not intended to be used to determine if a move is complete. Rather, it should be used after the move is complete to determine if there might be other errors or faults.

Example

Command
1RB

Response

*A (After issuing 1RB, the response came back as *A. This means that the indexer is currently within a loop.)

RC Status	Closed Loop Status			VALID Software Version E2
SYNTAX aRC	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO R, RA, RB		
RESPONSE TO aRC IS *x				

Description

The **RC** command has the same response format of **RA** and **RB**. The four conditions for which status is indicated are:

Static Position Loss:

In this condition, the indexer has detected motion of the load while the motor was stopped. The indexer was not able to correct the position, resulting in Position Maintenance being disabled.

Post Move Position Loss:

In this condition, the indexer has detected a deviation between actual and desired position at the end of a move which exceeds the backlash/dead band parameter. This may involve a Stall, or slipping of the load short of a stall.

Homing Function Failure:

In this condition, the indexer has encountered both End-of-Travel limits or one of several possible Stop commands or conditions. Go Home motion was concluded, but not at Home.

Stall:

In this condition, the indexer has detected a deviation between motor and encoder position larger than one pole of the motor while running, or a deviation larger than that plus the backlash parameter following a direction change.

NOTE: This command is not intended to be used to determine if a move is complete. Rather, it should be used after the move is complete to determine if there might be other errors or faults.

**RC
Response
Table**

Response Character	Stall Detected?	Go Home Successful?
*@	NO	YES
*A	YES	YES
*B	NO	NO
*C	YES	NO

Example

Command
1RC

Response
*A (This means that while attempting the last move, the indexer detected a stall.)

RM Motion		Rate Multiplier in Velocity Streaming Mode			VALID Software Version E2
SYNTAX <a>RMn	UNITS revs/sec	RANGE n = 0000 - 4E20 (0.0 - 50.0)	DEFAULT 0000	ATTRIBUTES Immediate Never Saved	
EXECUTION TIME			SEE ALSO Q1, Q0		

Description

The **RM** command followed by 4 hexadecimal digits represents a velocity. The 4 hex digit range is 000 to 4E20, representing 0.0 to 50.0 rps, in units of 1/400 (0.0025) rps.

The velocity change is essentially instantaneous; there is no acceleration/deceleration ramp between velocities. A limit switch-closure will stop movement while in velocity profiling mode, but does not cause the Indexer to exit velocity streaming mode. **RM** (profiling) mode is unidirectional. The direction will be the last activated direction either from an actual move or from a **D** or **H** command. Bi-directional moves using this mode can be made by returning to velocity zero, switching off **RM** mode, changing the direction, and re-enabling **RM** mode. This extra overhead should be acceptable given the need to change to velocity zero when changing directions in real situations.

Situations requiring non-linear accelerations may use the **Q0**, **Q1**, and **RM** commands. **Q1** is used to enter the velocity profiling mode, and **Q0** is used to exit. While in this mode the **RM** command is used to generate velocity values that are immediately implemented while the motor is moving. This means that the **RM** command must be sent to the AX at the time the change in velocity is required. This creates a stair-step effect in velocity change. By implementing a large number of very small instantaneous velocity changes, a smooth, non-linear acceleration ramp can be achieved.

Example	Command	Description
	Q 1	Enter Velocity streaming mode
	RM0190	Accelerate to 1 rev/sec
	RM0320	Accelerate to 2 revs/sec
	RM04B0	Accelerate to 3 revs/sec
	RM0640	Accelerate to 4 revs/sec
	RM04B0	Decelerate to 3 revs/sec
	RM0320	Decelerate to 2 revs/sec
	RM0190	Decelerate to 1 rev/sec
	RM0000	Decelerate to 0 revs/sec
	Q 0	Exit velocity streaming mode

RS		Status of Sequence Execution			VALID
Status					Software Version E2
SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES	
aRS	N/A	N/A	N/A	Immediate Never Saved	
EXECUTION TIME <2mS			SEE ALSO R, RA, RB, RC		
RESPONSE TO aRS IS *x					

Description The **RS** command indicates the status of the latest sequence execution. Possible responses are as follows:

Response Character	Sequence Started	Sequence Ended
* @	NO	NO
* A	YES	NO
* B	NO	YES

Whenever a sequence is started, the sequence start bit is set and the sequence end bit is cleared (this only occurs if the sequence is valid and is actually run). Bit 0: Sequence started; Bit 1: Sequence Ended. Whenever a sequence is ended, the start bit is cleared and the end bit is set. Any abrupt move termination (e.g., limit activation), or a **K** or **S** command clears both bits.

Example	Command	Response
	1RS	* A (Sequence in progress)

RV Status	Revision Level			VALID Software Version E2
SYNTAX aRV	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Never Saved
EXECUTION TIME <2mS		SEE ALSO None		
RESPONSE TO aRV IS *nn-nnnn-nn<xn>				

Description

The Revision (**RV**) command responds with the software part number and its revision level. The response is in the form shown below:

```
*nn-nnnn-nn<xn>[cr]
(nn-nnnn-nn = part number; <xn> = 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.

Example

<u>Command</u>	<u>Response</u>
1RV	*92-7212-01E2

The product is identified by 92-7212-01. The revision level is identified by E2.

S Motion	Stop			VALID Software Version E2
SYNTAX <a>S	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO K, SSH, QØ, A, SSG		

Description This command decelerates the motor to a stop using the last defined Acceleration (A) command. This command normally clears any remaining commands in the command buffer, unless prevented from doing so by the Clear/Save The Command Buffer On Stop (SSH1) command. When the SSH1 command is present, the S command stops only the current move. The indexer executes the next command in the buffer. The Stop (S) command does not stop the motor if in Rate Multiplier (RM) mode. If you are in the RM mode, issue an Exit Velocity Profiling Mode (QØ) command to stop the motor.

Example	<table border="0"> <tr> <td><u>Command</u></td> <td><u>Description</u></td> </tr> <tr> <td>MC</td> <td>Sets move in continuous mode</td> </tr> <tr> <td>A 1</td> <td>Sets acceleration to 1 rev/sec²</td> </tr> <tr> <td>V 1 Ø</td> <td>Sets velocity to 10 rev/sec</td> </tr> <tr> <td>G</td> <td>Executes the move (Go)</td> </tr> <tr> <td>S</td> <td>Stops motor (motor comes to 0 rev/sec at a deceleration rate of 1 rev/sec²)</td> </tr> </table>	<u>Command</u>	<u>Description</u>	MC	Sets move in continuous mode	A 1	Sets acceleration to 1 rev/sec ²	V 1 Ø	Sets velocity to 10 rev/sec	G	Executes the move (Go)	S	Stops motor (motor comes to 0 rev/sec at a deceleration rate of 1 rev/sec ²)
<u>Command</u>	<u>Description</u>												
MC	Sets move in continuous mode												
A 1	Sets acceleration to 1 rev/sec ²												
V 1 Ø	Sets velocity to 10 rev/sec												
G	Executes the move (Go)												
S	Stops motor (motor comes to 0 rev/sec at a deceleration rate of 1 rev/sec ²)												

The S command is not buffered since it is an immediate command. As soon as the indexer receives the S command, it stops motion.

SC Programming		Standby Current			VALID Software Version E2
SYNTAX <a>SCn	UNITS N/A	RANGE n = 1 - 8	DEFAULT 8	ATTRIBUTES Buffered Savable in Sequence	
EXECUTION TIME <2ms			SEE ALSO SCA		

Description

This command reduces motor current when the motor is not moving. The percentage of maximum current going to the motor is listed below:

n	Percentage of Maximum Current
1	13%
2	25%
3	38%
4	50%
5	63%
6	75%
7	88%
8	100%

This command keeps the motor cooler and saves energy; however, you sacrifice some holding torque at zero speed. Once the motor starts moving, full motor current is restored.

This command is valid only while the motor is stationary. When the motor starts moving, you must re-issue the command to reduce current after the move.

Example

<u>Command</u>	<u>Description</u>
SC1	Sets standby current level to 13%

When the motor stops moving, current is immediately set to 1/8 of the maximum current.

SCA Programming	Standby Current Automatically			VALID Software Version E2
SYNTAX <a>SCAN	UNITS N/A	RANGE n = 1 - 8	DEFAULT 8	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO SC		

Description This command reduces current going to the motor after move is complete. The percentage of maximum current going to the motor is $n/8 \times 100\%$ of maximum current. When the motor starts moving again, full current is restored to the motor. This standby current will reduce holding torque when motor is not moving, however the motor will run cooler and uses less energy.

Example

<u>Command</u> SCA2	<u>Description</u> Every time motor comes to a stop, the current to the motor will be set to $2/8 \times 100\%$ of the maximum current.
-------------------------------	--

SN Set-up	Scan			VALID Software Version E2
SYNTAX <a>SNn	UNITS n = mS	RANGE 1 - 1000	DEFAULT 50	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO None		

Description 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.

Example

<u>Command</u> SN150	<u>Description</u> Sets scan time of sequence select inputs to 150 milliseconds.
--------------------------------	---

SSA Set-up	RS-232C Echo Control			VALID Software Version E2
SYNTAX <a>SSAn	UNITS N/A	RANGE n = 0, 1	DEFAULT 0	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO None		

Description

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

SSA0 = Echo on

SSA1 = Echo off

In the Echo On (**SSA0**) mode, characters that are received by the indexer are echoed automatically. In the Echo Off (**SSA1**) mode, characters are not echoed from the indexer. This command is useful if your computer cannot handle echoes. In a daisy chain, you must have the echo turned on (**SSA0**) to allow indexers further down the chain to receive commands.

Status commands do not echo the command sent, but transmit the requested status report.

Example

Command
SSA1

Description

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

SSD Set-up	Limit Inputs for Stopping			VALID Software Version E2
SYNTAX <a>SSDn	UNITS N/A	RANGE n = 0, 1	DEFAULT 0	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO S, SS		

Description

The hardware limit switches can act as remote stop inputs (can be controlled from PLC rather than via RS-232C link). **SSD0** uses limits as simple limit switch inputs. When this function is enabled (**SSD1**), (and when the CW and CCW limit inputs are both opened), the Stop (**S**) command will be executed, until one or both of the limits are grounded (closed). The **LD** command will not affect the **SSD** command.

Example

Command
SSD1

Description

This command enables CW and CCW limits as stop inputs.

SSG Set-up	Clear/Save the Command Buffer on Limit			VALID Software Version E2
SYNTAX <a>SSGn	UNITS N/A	RANGE n = 0, 1	DEFAULT 0	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO LD		

Description

In most cases, it is desirable that upon activating an end of travel limit input, all motion should cease until the problem causing the over-travel is rectified. This will be assured if all commands pending execution in the command buffer are cleared when hitting a limit. This is the case if **SSG0** is specified. If **SSG1** is specified and a limit is activated, the current move is aborted, but the remaining commands in the buffer continue to be executed.

Example

<u>Command</u>	<u>Description</u>
SSG1	Save buffer on limit
A10	Set acceleration to 10 rev/sec ²
V5	Set velocity to 5 rev/sec
MC	Set to mode continuous
L5	Loop 5 times
G	Execute the move (Go)
H+	Sets direction to CW
N	End loop

The motor will move back and forth between limits.

SSH Set-up	Clear/Save the Command Buffer on Stop			VALID Software Version E2
SYNTAX <a>SSHn	UNITS N/A	RANGE n = 0, 1	DEFAULT 0	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO LD, S, FSF		

Description

SSH0 = Clears command buffer on stop
SSH1 = Saves command buffer on stop

In Normal Operation (**SSH0**) the Stop (**S**) command will cause any commands in the command buffer to be cleared. If you select the Save 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.

Example	Command	Description
	SSH0	Clears buffer on stop
	A10	Sets acceleration to 10 rev/sec ²
	V5	Sets velocity to 5 rev/sec
	MN	Sets indexer to Mode Normal
	D25600	Sets distance to 25,600 steps
	L50	Loops 50 times
	G	Executes the move (Go)
	T.5	Pauses the motor 500 msec
	N	Ends Loop
	S	Stops motion

When you issue the **S** command, the indexer will clear the buffer and stop the move. Stalls, Stop, **FSF**, **SSD1** and **TRIG3** move terminate are treated as stops.

ST		Shutdown			VALID
Programming					Software Version E2
SYNTAX	UNITS	RANGE	DEFAULT	ATTRIBUTES	
<a>STn	N/A	n = 0, 1	0	Buffered Savable in Sequence	
EXECUTION TIME <2mS			SEE ALSO None		

Description

The Shutdown (**ST1**) command rapidly decreases the motor current to zero. The system ignores move commands that you issue after the **ST1** command. Torque on the motor is not maintained after you issue the **ST1** command.

The **ST0** command rapidly increases the motor current to normal. Once you restore the current, you can execute moves.

This command is useful for reducing motor heating and allows you to manually position the load.

Example

Command	Description
ST1	Shuts off current to the motor

SV Programming	Servoing Parameter			VALID Software Version E2
SYNTAX <a>SVn	UNITS N/A	RANGE n = 0 - 3	DEFAULT None	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO FSC, ST		

Description

The Servoing Parameter (**SV**) command provides four different ways of simultaneously changing state of the motor shutdown and position maintenance functions. The four commands are as follows:

- SV0** This command causes the position maintenance function to be turned off, but does not turn off motor power. It is identical in function to the **FSC0** command.
- SV1** This command causes the position maintenance function to be turned off and the motor to be shut down simultaneously.
- SV2** This command causes the position maintenance function to be turned on and turns the motor power back on if it was turned off due to **SV1** or **SV0** command. The encoder position will be read and this newest position will be maintained.
- SV3** This command causes the position maintenance function to be turned on and turns the motor power on if it was turned off. The indexer will servo back to the rest position held before the position maintenance function was disabled.

Example

<u>Command</u>	<u>Description</u>
SV1	Simultaneously turns off Position Maintenance function and shuts down the motor.

T Programming	Time Delay			VALID Software Version E2
SYNTAX <a>Tn	UNITS n = seconds	RANGE 0.01 - 999.99	DEFAULT None	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2ms		SEE ALSO None		

Description The Time (T) command causes the indexer to wait the number of seconds that you specify (n) 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..

Position maintenance is not active during a Time delay.

Example

<u>Command</u>	<u>Description</u>
MN	Set to mode normal
A5	Set acceleration to 5 rev/sec ²
V6	Set velocity to 6 rev/sec
D12800	Set distance to 12,800 steps
G	Execute the move (Go)
T3	Delays 3 seconds between moves
G	Execute the move (Go)

TEST Motion	Test Routine			VALID Software Version E2
SYNTAX <a>TEST	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME		SEE ALSO None		

Description The Test Routine (TEST) command is intended for use as an initial checkout command to verify if the system functions properly. Immediately after you issue the TEST command, the following sequence is executed:

LD3 MN MPI A2 V1 D12800 G T.5 H G

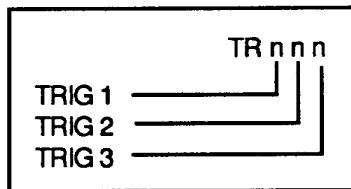
Example	Command	Description
	TEST	Executes a test sequence
	Test Sequence:	
	LD3	Disables CW and CCW limits (all axes)
	MN	Sets indexer to normal mode
	MPI	Sets positioning mode to incremental
	A2	Sets acceleration to 2 rps ²
	V1	Sets velocity to 1 rps
	D12800	Sets distance to 12,800 steps
	G	Executes the move (Go)
	T.5	Waits 0.5 seconds after finishing the move
	H	Changes the direction of the next move
	G	Executes the move (Go)

After issuing the **TEST** command the motor turns one CW revolution, pauses for 0.5 seconds, and turns one CCW revolution.

TR Programming		Wait for Trigger			VALID Software Version E2
SYNTAX <a>TRn	UNITS n = input	RANGE n = 0, 1, x	DEFAULT None	ATTRIBUTES Buffered Savable in Sequence	
EXECUTION TIME <2mS			SEE ALSO TS		

Description This command allows you to specify a trigger configuration to be matched before continuing execution of the move. The command is in the form TRnnn, where nnn corresponds to triggers 1, 2, and 3 respectively. The possible values for n are as follows:

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



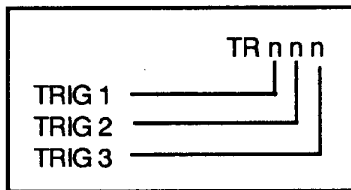
When **TR** command is used in a buffer, the indexer will get to this command and wait until the input pattern is matched before going on to the next command.

Example	Command	Description
	TR1ØX	Wait for input 1 to be opened and input 2 to be grounded before going on to the next command. Input 3 will be ignored.
	A1Ø	Sets acceleration to 10 rev/sec ²
	V5	Sets velocity to 5 rev/sec
	D256ØØ	Sets distance to 25,600 steps
	G	Executes the move (Go)

TS Status	Trigger Input Status			VALID Software Version E2
SYNTAX aTS	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO TR		
RESPONSE TO aTS IS nnn				

Description This command retrieves the state of the trigger inputs. The response is in the form nnn, where nnn reports the status of triggers 1, 2, and 3 respectively. The possible values for n are as follows:

- n = 1** Input is high (opened)
- n = 0** Input is low (closed)



TS command is useful for checking the status of the trigger inputs when it appears as though execution is being halted by a **TR** command. To make sure that your trigger pattern is met, you can check with **TS** command.

Example	Command	Response
	1TS	101

Trigger bits 1 and 3 are high (opened) and Trigger bit 2 is low (closed).

U Programming	Pause and Wait for Continue			VALID Software Version E2
SYNTAX <a>U	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO PS, C		

Description This command causes the indexer to complete the move in progress, then wait until it receives a Continue (C) to resume processing. Since the buffer is saved, the indexer continues to execute the program (at the point where it was interrupted). The indexer continues processing when it receives the C command. This command is typically used to stop a machine while it is unattended.

Example	<u>Command</u> MN A 5 V 5 L D25600 G T 10 N U	<u>Description</u> Sets move to Normal mode Sets acceleration to 5 rev/sec ² Sets velocity to 5 rev/sec Loops indefinitely Sets distance to 25,600 steps Executes the move (G) Waits 10 seconds after the move Ends loop Halts execution until the indexer receives the Continue command.
----------------	--	---

This command string pauses at the point where the U command is entered. A Continue (C) command causes execution to resume at the point where it was paused. In this example, the loop stops at the end of a move, and resumes when the indexer receives the C command. There may be a 10-second delay before motion resumes after the C command is executed, depending on when the Pause and Wait for Continue (U) command is completed.

V Motion	Velocity			VALID Software Version E2
SYNTAX <a>Vn	UNITS n = revs/sec ²	RANGE 0.001-50.000	DEFAULT 0	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO A, D, G		

Description The Velocity (**V**) command defines the maximum speed at which the motor will run when given the Go (**G**) command.

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

In preset mode, Mode Normal (**MN**) the maximum velocity may also be limited when the resulting move profile is triangular. In Mode Continuous (**MC**), when a Go (**G**) command is completed, the indexer moves on to the next command in the buffer once the specified velocity is reached.

When the Go Home (**GH**) command is executed, the velocity is changed accordingly; subsequent moves should include a new **V** command unless the Go Home velocity is applicable.

Once you define the velocity, that velocity will be valid until you define another velocity, cycle AC power or issue a **Z** command.

*NOTE: If the value specified for the **V** command is not valid, the AX ignores that value and defaults to the value specified in the last **V** command.*

VC Motion	Change Velocity in Continuous Mode			VALID Software Version E2
SYNTAX <a>VCn	UNITS n = rev/sec	RANGE 0.001 - 50.000	DEFAULT None	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2ms		SEE ALSO V, AC		

Description This command changes the velocity while moving in continuous mode. Valid velocities for this command range from 0.001 to 50.000 revolutions per second. This command is effective only when running at constant velocity. Velocity change occurs immediately following the issuance of the **VC** command. Not more than 2 digits before the decimal point are allowed. This command does not change the velocity value set by the **V** command.

Example	<u>Command</u>	<u>Description</u>
	MC	Do moves in continuous mode
	A 10	Sets acceleration to 10 rev/sec ²
	V 5	Sets velocity to 5 rev/sec
	G	Executes the move (Go)
	VC 7	Change velocity to 7 rev/sec ² at acceleration of 10 rev/sec ²

The motor will ramp up to constant velocity at 5 rps after the first **G** command is entered. The motor will ramp up to a constant velocity of 7 rps after the **VC** command is entered.

W1 Status	Signed Binary Position Report			VALID Software Version E2
SYNTAX aW1	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME		SEE ALSO W3, PR		
RESPONSE TO aW1 IS *nnnn				

Description

Report back gives immediate binary representation of position relative to start of the current move. The format of the response is a four character response (*nnnn) that is interpreted as a 32-bit binary number. The number must then be interpreted by the computer to give a numerical position in steps. The format is in 2's complement notation. Moves in the negative direction (CCW) will report back negative numbers (bit 31 is set to 1).

If you are using a terminal to communicate with the indexer, the response may not be a printable character. The response must be decoded using a computer.

This command is useful if you want to receive a position report while the motor is moving.

Example

None

W3 Status	Hexadecimal Position Report			VALID Software Version E2
SYNTAX aW3	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO W1, PR		
RESPONSE TO aW3 IS *FFFFnnnn				

Description

This command will respond with an immediate hexadecimal character position report back in 2's complement format. The position response indicates the motor position relative to the current move. The format of the response is an eight digit ASCII hexadecimal number. The **PZ** command does not affect this value.

Assume the response was *0000024E. The decimal value would be 590 (pulses).

If the first digit of the response is an *F* (e.g., *Fnnnnnnnn), then the response represents a *2s complement* negative number.

Use the following steps to interpret a negative number (starting with *F*)

The binary approach:

1. Convert the hexadecimal response to binary form.
2. Complement the binary number
3. Add 1 to the binary result
4. Convert the binary result to decimal value with a minus sign placed ahead of the decimal value.

The computer approach:

Subtract the hexadecimal number from 168 (232) (4,294,967,296).

The easy way:

1. Leave off all the leading *F*s, and convert to decimal
2. Convert and subtract the next largest power of 16.

Example: *FFFF9E58

1. Leave off the <i>F</i> s:	9E58 hex	=	40,536
2. Subtract from 164	10000 hex	=	65,536
	Results	=	-25,000

Example

Command
1W3

Response

***FFFA19C** In the current move, you are at 24,163 steps from initiation of move.

WV Programming	Select Waveform			VALID Software Version E2
SYNTAX <a>WVn	UNITS N/A	RANGE n = 1 - 9	DEFAULT 5	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO None		

Description

The **WV** command will select 1 of 9 stored waveforms. The waveforms differ in the amount of 3rd harmonic that each has. Generally, the higher the current in the motor, the greater the percentage of out-of-phase 3rd harmonic you would want. Empirical determination of the best sine wave for a given motor and load is the best means for selecting a waveform.

Waveform	% of 3 rd Harmonic
#1	-8
#2	-6
#3	-4
#4	-2
#5	0
#6	+2
#7	+4
#8	+6
#9	+8

Example

Command
1WV1

Description
Waveform #1 is selected

XC Status	Sequence Checksum			VALID Software Version E2
SYNTAX aXC	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO XD, XE		
RESPONSE TO aXC IS nnn				

Description This command computes the EEPROM checksum. After the indexer has been programmed, the response can be used for system error checking. The response is in the form nnn, where the range for nnn is 000 - 255. The number reported does not indicate the number of bytes programmed. This response is designed to be used for comparison. As long as the indexer is not reprogrammed, the checksum response should always be the same.

Example

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

XD Programming	Sequence Definition			VALID Software Version E2
SYNTAX <a>XDn	UNITS n = sequences	RANGE 1 - 7	DEFAULT None	ATTRIBUTES Buffered Never Saved
EXECUTION TIME <2mS		SEE ALSO XE, XR, XRP, XSD, XT		

Description This command begins sequence definition for a specific sequence. All the commands between the **XD** command and the Sequence Termination (**XT**) command will be defined as a sequence. The sequences will automatically be saved when the **XT** command is issued. If a sequence you are trying to define already exists, you must erase that sequence (**XE** command) before defining it. Each sequence cannot be longer than 256 characters

Immediate commands cannot be entered into a sequence.

Example	Command	Description
	XE 1	Erase sequence #1
	XD 1	Define sequence #1
	MN	Set to mode normal
	A 1 0	Sets acceleration to 10 rev/sec ²
	V 5	Sets acceleration to 5 rev/sec
	D 256 0 0	Sets distance to 25,600 steps
	G	Executes the move (Go)
	XT	End defining sequence #1
	XR 1	Execute sequence #1

The commands in sequence 1 are defined and executed.

XE Programming		Sequence Erase			VALID Software Version E2
SYNTAX <a> XE n	UNITS n = sequences	RANGE 1 - 7	DEFAULT None	ATTRIBUTES Buffered Never Saved	
EXECUTION TIME <2mS			SEE ALSO XD, XT, XR, XRP		

Description This command allows you to delete a sequence. The sequence that you specify (n) will be deleted when you issue the command.

As a good practice, you should delete a sequence before defining it.

Example See example for **XD** command.

XP Set-up	Set Power-up Sequence Mode			VALID Software Version E2
SYNTAX <a>XPn	UNITS N/A	RANGE n = 0 - 9	DEFAULT 0	ATTRIBUTES Buffered Automatically Saved
EXECUTION TIME <2mS		SEE ALSO K, S, XR, XQ, XSP, XSSR, XZ		

Description

Set power-on sequence mode. This command will cause a single sequence or multiple sequences to be executed on power-up. The possible settings for *n* are as follows:

- n = 0** No sequence is executed
- n = 1-7** Sequence 1 - 7 is executed on power-up
- n = 8** Sequence Select inputs are read (single run)
- n = 9** Sequence Select inputs are read (continuous run)

A value of 1-7 for *n* will result in the sequence whose value = *n* being executed on power-up. Control will then be passed to the RS-232C interface.

A value of 8 for *n* will result in the sequence whose number appears on the sequence select inputs to be executed on power-up. Control will then be passed to the RS-232C interface.

A value of 9 for *n* will cause the sequence whose number appears on the Sequence Select inputs to be executed on power-up. When the first sequence is finished in XP9 mode, the AX will scan the Sequence Select inputs again and execute the next sequence whose number appears on the inputs. This cycle will continue until a Stop (**S**) or Kill (**K**) command is issued, a limit is encountered, or the unit is powered down.

Example

<u>Command</u>	<u>Description</u>
XP1	Execute Sequence #1 on power-up
XE1	Erase sequence #1
XD1	Define sequence #1
A1Ø	Sets acceleration to 10 rev/sec ²
D256ØØ	Sets distance to 25,600 steps
SN1Ø	Set sequence debounce time to 10mS
GH5	Go Home at 5 rev/sec
XT	End of sequence #1
Z	Reset the indexer

The motor will move 25,600 steps every time you power-up or reset using **Z** command.

XQ Set-up	Sequence Interrupted Run Mode			VALID Software Version E2
SYNTAX <a> XQ n	UNITS N/A	RANGE 0, 1	DEFAULT 0	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO XD, XE, XT, XZ, XP		

Description

XQ1 = Set interrupted run mode (on)
XQ0 = Clear interrupted run mode (off)

This command can be used only when stand-alone power-up sequencing in XP9 mode. If **XQ1** is executed, the indexer will ignore sequence select inputs, until all sequence select lines have been brought to a high state. After all lines have simultaneously been brought to a high state, the indexer will then read the sequence select lines and execute the sequence whose number appears there. This paused mode will continue until an **XQ0** command is executed. You may use **S** or **K** command to stop sequence execution.

The interrupted run mode is cleared at the start of execution of a sequence.

Example

<u>Command</u>	<u>Description</u>
XE 1	Erase sequence #1
XD 1	Define sequence #1
LD 3	Disable CW & CCW limits
XQ 1	Sets interrupted mode on
XT	End Sequence #1
XP 9	Sets power-up sequences as sequence select inputs
Z	Resets the AX to start sequence scanning

If you execute Sequence 1 upon power-up by setting **SEQ1-SEQ3** inputs properly, the interrupted run mode will be set. Sequence select input lines all need to go high (open) before selecting any other sequences.

XR Programming	Run A Sequence			VALID Software Version E2
SYNTAX <a>XRn	UNITS n = sequences	RANGE 1 - 7	DEFAULT N/A	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO XE, XD, XT, XRP, SN		

Description This command loads a predefined sequence (identified by n) into the command buffer (clears the buffer first) and executes these commands as a normal set of commands. This command automatically recalls the sequence from EEPROM.

An **XR** command can be used within one sequence to start execution of another sequence; however, all commands in the first sequence following the **XR** will be ignored (in this respect an **XR** acts like a GOTO not a GOSUB). If using continuous mode, the velocity must be 0 rps when calling another sequence. An **XR** command placed within a loop will be ignored.

Example

<u>Command</u>	<u>Description</u>
XE1	Erase sequence #1
XD1	Define sequence #1
A10	Sets acceleration to 10 rev/sec ²
V5	Sets acceleration to 5 rev/sec
D25600	Sets distance to 25,600 steps
G	Executes the move (Go)
XT	End defining sequence #1
XR1	Execute sequence #1

Sequence 1 is defined and executed using **XD1** and **XR1** commands respectively

XRP Programming	Sequence Run With Pause			VALID Software Version E2
SYNTAX <a>XRPn	UNITS n = sequences	RANGE 1 - 7	DEFAULT None	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO XR, XD, XT, XE, C, SN		

Description

This command is identical to the Sequence Run (**XR**) command, except that it automatically generates a pause condition. You must clear this condition with the Continue (**C**) command before the indexer executes the command buffer. The pause condition is asserted only if the sequence is valid. This allows you to execute a sequence without the delay of buffering that sequence.

Example

<u>Command</u>	<u>Description</u>
XE5	Erases Sequence #5
XD5	Defines Sequence #5
A10	Sets acceleration to 10 rev/sec ²
V5	Sets velocity to 5 rev/sec
D25600	Sets distance to 25,600 units
G	Executes the move (Go)
XT	Ends defining Sequence #5
XRP5	Runs Sequence #5 with a pause
C	Indexer executes Sequence #5

Upon issuing **XRP5**, Sequence #5 is entered into the command buffer, but is not executed. You must issue a Continue (**C**) command to execute Sequence #5.

XSD Status	Sequence Download Status			VALID Software Version E2
SYNTAX aXSD	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Never Saved
EXECUTION TIME <2mS		SEE ALSO XD, XE, XT		
RESPONSE TO aXSD IS n				

Description

This command reports back the status of the previous sequence definition (**XD...XT**). The response is 0 - 3. The **VALID** values and descriptions of possible responses are shown below:

- 0=** Download O.K.
- 1=** A sequence already exists with the number you have specified.
- 2=** Out of memory. The sequence buffer is full.
- 3=** EEPROM write error.

The **XSD** command is useful for verifying that the last sequence definition attempt was successful.

Example

Command
1XSD

Response
1 (This response indicates that you need to erase the existing sequence to define that specific sequence.)

XSP Status	Power-up Mode Sequence Status			VALID Software Version E2
SYNTAX aXSP	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Savable in Sequence
EXECUTION TIME <2mS		SEE ALSO XP, XSR, XQ, XZ		
RESPONSE TO aXSP IS n				

Description 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 using the **XSP** command. The command reports back the sequence that the system will execute during power-up. The range of the response is 0 - 9.

Example

<u>Command</u>	<u>Response</u>
1XSP	3 (indicates that sequence number #3, if it exists, will be executed upon power-up or reset)

XSR Status	Sequence Run Status			VALID Software Version E2
SYNTAX aXSR	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Never Saved
EXECUTION TIME <2mS		SEE ALSO XR, XRP		
RESPONSE TO aXSR IS n				

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

0 = Last attempt to run the sequence was successful
Non-zero = Not running:
1 = In a loop
2 = NON-VALID sequence was requested
3 = Erased sequence
4 = Bad checksum

Example

<u>Command</u>	<u>Response</u>
XR2	Runs Sequence 2
1XSR	0 (Sequence ran OK)

XSS Status	Sequence Status			VALID Software Version E2
SYNTAX aXSSn	UNITS n = sequences	RANGE 1 - 7	DEFAULT None	ATTRIBUTES Buffered Never Saved
EXECUTION TIME <2mS		SEE ALSO XD, XE, XT		
RESPONSE TO aXSSn IS n				

Description This command reports whether the sequence specified by n (representing one of seven sequences) is empty, has a bad checksum, or is OK. The possible responses are as follows:

- 0 = Empty
- 1 = Bad Checksum
- 3 = O.K.

This command is useful to see if the particular sequence exists and if that portion of memory has been corrupted.

Example

<u>Command</u>	<u>Response</u>
1XSS1	0 (Nothing programmed in sequence #1)

XT Programming	Sequence Termination			VALID Software Version E2
SYNTAX <a>XT	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Buffered Never Saved
EXECUTION TIME		SEE ALSO XD, XE, XR		

Description The **XT** command is a sequence terminator. This command flags the end of the sequence currently being defined. Sequence definition is not complete (saved) until this command is issued.

Example

<u>Command</u>	<u>Description</u>
XD1	Define sequence #1
MN	Sets move to continuous
A1Ø	Sets acceleration to 10 rev/sec ²
V5	Sets velocity to 5 rev/sec
D256ØØ	Sets distance to 25,600 steps
G	Executes the move (Go)
XT	End sequence definition

XU Status	Upload Sequence			VALID Software Version E2
SYNTAX aXUn	UNITS n = sequences	RANGE 1 - 7	DEFAULT N/A	ATTRIBUTES Buffered Never Saved
EXECUTION TIME		SEE ALSO XD, XE, XT, F		

Description

This command sends the contents of sequence *n* to the host computer via RS-232C interface. All commands in sequence *n* are displayed on the CRT. All command delimiters in the sequence will be sent out as spaces (20H). Any device identifiers that were included in the original sequence will also be eliminated (they are not stored in the sequence).

NOTE: When using a daisy-chain, this command must be used cautiously as the contents of the sequence will go to all controllers in the loop between the indexer that is uploading and the host. The F command may be useful in this context to turn off communication on units you are not uploading from.

Example

<u>Command</u>	<u>Description</u>
F	Turn off communication to all units
1 E	Turn on communication to unit #1
1XU7	Upload sequence #7 from unit #1
E	Enable other units

XZ Programming	Set power-up Sequence to Zero			VALID Software Version E2
SYNTAX <a> XZ	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME		SEE ALSO Z, XSP, XSR, XP, XQ		

Description

This command sets the power-up sequence number to zero (thereby disabling sequence activation on power-up). The purpose of this command is to handle the situation in which the set power-up sequence has a checksum error, causing control to pass to the error routine (flashing LED); only immediate commands are active in this error state. In testing externally controlled sequences, it can also be used to disable continuous run sequencing (followed by a **Z** command to reset the indexer).

The main difference between the **XZ** command and the **XP0** command is that **XZ** is executed immediately and **XP0** is entered into a buffer. If you are already running a sequence, use this command.

Example

None

Y Programming	Stop Loop			VALID Software Version E2
SYNTAX <a>Y	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME <2mS		SEE ALSO L, N, CL, CN		

Description The Stop Loop (Y) command takes you out of a loop when the loop completes its current pass. This command does not halt processing of the commands in the loop until the indexer processes reach the last command of the current loop. At that time, the indexer executes the command that follows the End Loop (N) command. You cannot restart the command loop unless you enter the entire command structure, including the Loop (L) and End Loop (N) commands.

Example	<u>Command</u>	<u>Description</u>
	L	Loops indefinitely
	A 1 0	Sets acceleration to 10 rev/sec ²
	V 5	Sets velocity to 5 rev/sec
	D 256 0 0	Sets distance to 25,600 steps
	T 2	Waits 2 seconds
	G	Executes the move (Go)
	N	Ends loop
	Y	Stops loop

The loop requires the motor to move 25,600 steps CW and then wait for 2 seconds. The loop terminates at the end of the loop cycle it is executing when it receives the Y command.

Z Programming	Reset			VALID Software Version E2
SYNTAX <a>Z	UNITS N/A	RANGE N/A	DEFAULT N/A	ATTRIBUTES Immediate Never Saved
EXECUTION TIME 800mS		SEE ALSO S, K		

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

Any commands entered while resetting are ignored.

This command sets all position counters to zero.

Example None