

ACCU ELECTRIC MOTORS INC

USA: (888) 932-9183

CANADA: (905) 829-2505

- ✓ Over 100 years cumulative experience
- ✓ 24 hour rush turnaround / technical support service
- ✓ Established in 1993



The leading independent repairer of servo motors and drives in North America.

Visit us on the web:

www.servo-repair.com

www.servorepair.ca

www.ferrocontrol.com

www.sandvikrepair.com

www.accuelectric.com

Scroll down to view your document!

For 24/7 repair services :

USA: 1 (888) 932 - 9183

Canada: 1 (905) 829 -2505

Emergency After hours: 1 (416) 624 0386

Servicing USA and Canada

ST Shutdown the Amplifier

Version Z5

Type	Programming
Syntax	<a>STb
Units	Energize/Shutdown
Range	0 or 1
Default	0
Response	None
See also	OFF, ON

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

The Shutdown (ST1) command commands the drive to de-energize. The motor has no torque when the drive is de-energized, allowing it to be moved by hand if not restricted by the mechanism it is mounted to. The Compumotor Plus ignores move commands that you issue while the drive is de-energized

The ST0 command energizes the drive. Once you enable the drive, you can execute moves. The absolute position counter is set to zero when you enter an ST0 command so that an Absolute Position Report (PR) reports 0 position and absolute moves are made with respect to the current position.

This command alleviates motor heating.

Command	Description
> ST1	Shuts off current to the motor

STOP Stop Motion

Version Z5

Type	Motion
Syntax	<a>STOP
Units	None
Range	None
Default	None
Response	None
See also	S, SSH, SSL

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

The STOP command is identical to the S command except that it is buffered and can be put into sequences. Refer to the S command for a description.

SV Save Set-Up and Sequences

Version Z5

Type	Programming
Syntax	<a>SV
Units	None
Range	None
Default	None
Response	None
See also	SAVE

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

The SV command saves any parameters you have defined since the last save (SV). Values you have not changed, are not changed by the Compumotor Plus. When you define a sequence, you must issue an SV command to save the sequence into permanent memory.

The following parameters are restored to their default values by the RFS command:

CCA, CCP, CDB, CDM, CDG, CGS, CIM, CIG, CIL, CPE, CPG, CMR, CPM, CVG, CVM.

Command	Description
> SV	Save all the changes
> *	Save successful

T Time Delay

Version Z5

Type Programming
Syntax <a>T
Units Seconds
Range 0.00 - 53687.09
Default None
Response None
See also None

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

This command performs a timed wait for the number of seconds specified by its parameter. No further buffered commands are executed during this wait. You may specify times as small as 0.01 second.

You may use this command in Continuous Mode (MC) to generate variable move profiles.

Command	Description
> MN	Sets to Normal mode
> A5	Sets acceleration to 5 rps ²
> V5	Sets velocity to 5 rps
> D25000	Sets distance to 25,000 steps
> T10	Pauses for 10 seconds
> G	Executes the move (Go)
> T5	Pauses for 5 seconds after the move ends
> G	Executes the move (Go)

TR Wait for Trigger

Version Z5

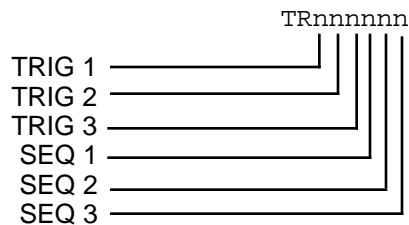
Type Programming
Syntax <a>TRb<bbbb>
Units Off, on, or don't care
Range 0, 1 or X
Default None
Response None
See also IFTR, TS

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

Triggers are used to synchronize indexer operations with external events. They can be used to implement a handshaking function with other devices. Three symbols that are used for b are listed below.

Symbol	State
0	Open
1	Grounded
X	Don't care

You may omit trailing X's if you wish. Parameters not specified with the command are assumed to be X's. For example, TR11 is the same as TR11XXX.



All six triggers are available in all input modes.

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.

If you omit trailing x's in the TR command, those trigger lines are ignored. The command TR100 is equivalent to TR100XXX.

Command	Description
> TR100XXX	Wait for input #1 to be active and input #2 to be opened before going on to the next command. Inputs 3, 4, 5, & 6 are ignored.
> A10	Sets acceleration to 10 rps ²
> V5	Sets velocity to 5 rps
> D25000	Sets distance to 25,000 steps
> G	Executes the move (Go)

TS Trigger Input Status

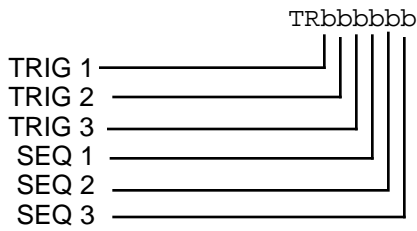
Version Z5

Type Status
Syntax <a>TS
Units None
Range None
Default None
Response <a>TS = *bbbbbb
See also IN, TR

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

This command retrieves the state of the trigger inputs. Response is in the form bbbbbbb where b can be any of the following.

Symbol	State
0	Open
1	Grounded
X	Don't care



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 the TS command.

Command	Description
> 1TS	*1010000[cr] Trigger bits #1 and #3 are closed and Bits #2, #4 and #6 are opened

TUNE Self Tune

Version Z5

Type	Set-Up
Syntax	aTUNE:m,o
Units	m = move type, o = options
Range	m = 1 to 3, o = 1 to 4
Default	m = 1, o = 1
Response	1TUNE = *TUNING_BEGUN/*TUNING_COMPLETE
See also	GAINX, FILT

Attributes
 Buffered
 Device specific
 Saved independently
 Saved in sequences

The TUNE command initiates the self-tuning sequence in the Compumotor Plus. For additional information on self-tuning, consult *Chapter 4, Application Design*.

Note: Before using the TUNE command you should set the maximum position error to a relatively low value (for example, 1 revolution). Also, ensure the motor is firmly mounted, and the motor shaft is coupled to the load.

The TUNE command initiates a sequence of small fast moves and collects information on the system performance. The amplifier then shuts down, and the Compumotor Plus analyzes the data. The analysis takes approximately two minutes.

When the analysis is complete the system computes appropriate gains based on the desired performance based on your setting of parameter b (explained below). Once the gains are computed, the amplifier is re-energized with these new gains in place. The system waits for about five seconds to make sure that the resulting gains are stable. If you move the shaft during this time, an error is reported. If the resulting gains are for some reason unstable, the Compumotor Plus reports an error. When the Compumotor Plus determines that the gains are suitable, it reports the following.

```
>*TUNING_COMPLETE
```

The gains resulting from self-tuning form a control scheme called *pole-placement* which is distinct from PIDV control. At present there is no direct translation between the pole-placed gains of self-tuning and the user accessible PIDV gains. To switch between these control schemes, use the GAINX command. The gains resulting from self-tuning are typically much higher than the default PIDV gains. If you find that the resulting gains produce excessive audible noise in the motor, use the FILT command to remove this noise.

The command format of of this self-tuning command is:

```
TUNE:m,o
```

Parameter m determines the type of initial tuning move to use. If your system cannot move in both directions or there is known backlash in the system, use move 2 or move 3.

m=1	512 step (at 12800 steps/rev) back and forth move
m=2	512 step hops in CW direction
m=3	512 step hops in the CCW direction

Parameter o determines the desired closed-loop performance:

o=1	low overshoot response, low in-position stiffness
o=2	fast response, some overshoot, higher stiffness
o=3	faster response, more overshoot, very stiff
o=4	moderate response for large loads

The results of the self-tuning procedure can be saved to non-volatile memory with the SV command.

U Pause & Wait for Continue

Version Z5

Type	Programming
Syntax	<a>U
Units	None
Range	None
Default	None
Response	None
See also	C, PS, SSL

Attributes
 Buffered
 Device specific
 Saved independently
 Saved in sequences

This command causes the Compumotor Plus to complete the command 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.

This following program fragment 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 Compumotor Plus 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.

Command	Description
> MN	Sets move to Normal mode
> A5	Sets acceleration to 5 rps ²
> V5	Sets velocity to 5 rps
> L0	Loops indefinitely
> D25600	Sets distance to 25,600 steps
> G	Executes the move (G)
> T10	Waits 10 seconds after the move
> N	Ends loop
> U	Halts execution until the Compumotor Plus receives the Continue command

V Velocity

Version Z5

Type	Motion
Syntax	<a>Vn
Units	rps
Range	0.00 to 60.00
Default	1
Response	*nnnnnnnn (where n is a number from 0 to 9)
See also	A, C, D

Attributes	
[X]	Buffered
[]	Device specific
[]	Saved independently
[X]	Saved in sequences

The Velocity (v) command defines the maximum speed at which the motion will occur when given the Go (G) command.

The velocity specified may not be reached if the move is too short. In this case the move will be triangular instead of trapezoidal. In Continuous mode (MC) the indexer executes the next command in the buffer only after the motor has reached the set speed. In Normal mode (MN) the indexer waits for the move to be complete before executing the next command in the buffer.

Command	Description
> MC	Set to Continuous mode
> A5	Set acceleration to 5 rps ²
> V5	Set velocity to 5 rps
> G	Execute the move (Go)

W1 Signed Binary Position Report

Version Z5

Type	Status
Syntax	aW1
Units	None
Range	None
Default	None
Response	cccc (where c is a single character (8 bits))
See also	PR, W2, W3

Attributes	
[]	Buffered
[]	Device specific
[]	Saved independently
[]	Saved in sequences

The W1 command returns the distance in steps the motor has moved since it last started motion. The reportback is a signed binary number with a range of ±2,147,483,648.

If the motor is moving, the reportback is the distance the motor has moved since starting motion. If the motor is not moving, it reports the distance of the previous move. The format of the response is a four byte response (cccc) that is interpreted as a 32-bit binary number. The information is transmitted in the order of high-byte to low-byte. The number must then be interpreted by the host device to give a decimal position in steps. The response is in 2's complement notation. Moves in the negative direction will report back negative number (bit 31 is set to 1).

The transmission is not preceded with an asterisk (*) like most numeric reportback commands in order to give the fastest possible response time. *Do not use the W1 command when multiple Compumotor Plus units are daisy-chained. Doing so will have unpredictable results.*

Interpreting Binary Position Reports

This form of position report (cccc), consisting of four bytes. The four bytes must be linked together to form a 32-bit binary number. Most terminals and communications programs for PCs expect to handle characters

rather than binary numbers and will not correctly display this kind of response. For example, a binary 7 will not display on a terminal, but will cause the terminal to beep.

The following table gives the weight of each byte:

Byte	Weight (decimal)	Weight (power of 2)
First	16,777,216	2 ²⁴
Second	65,536	2 ¹⁶
Third	256	2 ⁸
Fourth	1	2 ⁰

To convert the reportback into a decimal, multiply each byte by the appropriate value in the table (i.e., the first byte by 16,777,216 etc...) and then add the decimal values for each to get the total.

Suppose you tell the motor to move 25,000 steps. You then issue the W1 command. The response from the Compumotor Plus is interpreted as shown below. Note: only one of the bytes is displayable on most terminals.

Byte	Binary Value	Decimal Value		Decimal Weight	Decimal Weighted Value
First	00000000	0	x	16,777,216	0
Second	00000000	0	x	65,536	0
Third	01100001	97	x	256	24,832
Fourth	10101000	168	x	1	168
				Total	25,000

W2

Hexadecimal Unsigned Position Report

Version Z5

Type Status
Syntax <a>W2
Units None
Range None
Default None
Response hhhhhhhh (where h is a hexadecimal number from 0 to F)
See also PR, W1, W3

Attributes
 [] Buffered
 [] Device specific
 [] Saved independently
 [] Saved in sequences

The immediate hexadecimal character position report back (normally used during motor motion) indicates the position in relation to the start of the current move. The format of the response is 8 hexadecimal characters. You must convert these characters into a usable format. This command does not report direction (i.e., it reports an unsigned number).

Digit	Base 16 Weight	Decimal Weight
h (MSD)	16^7	268,435,456
h	16^6	16,777,216
h	16^5	1,048,576
h	16^4	65,536
h	16^3	4,096
h	16^2	256
h	16^1	16
h (LSD)	16^0	1

The hexadecimal number h, can be one of the following values:

Decimal Value	Hexadecimal Value (h)	Decimal Value	Hexadecimal Value (h)
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

To convert a hexadecimal number to a decimal number, multiply each digit by its decimal weight and add the value of each digit. For example, if you tell the motor to move 25,000 steps and issue the W2 command the Compumotor Plus responds with 000061A8. To convert this hexadecimal number to decimal:

Digit		Decimal Weight	Decimal Weighted Value
0	x	268,435,456	0
0	x	16,777,216	0
0	x	1,048,576	0
0	x	65,536	0
6	x	4,096	24,576
1	x	256	256
A	x	16	10
8	x	1	8
Total			25,000

Command > 1W2 *0000FA04
Description A distance of FA04 (64,004 steps) has occurred since the start of the last move

W3 Hexadecimal Signed Position Report

Version Z5

Type	Status
Syntax	aW3
Units	None
Range	None
Default	None
Response	*hhhhhhh (where h is a hexadecimal number from 0 to F)
See also	W1, W2, PR

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

This command provides a position report in signed hexadecimal format while the motor is moving. The report indicates position in steps relative to the start of the current move.

This form of position report (*nnnnnnnn) is generated by the W3 command. It consists of an asterisk followed by eight hexadecimal characters; 0 through 9 and A through F. The position report is followed by a carriage return.

If the response is positive use the conversion process explained above for the W2 command. You can tell if the response is positive by looking at the first digit of the response after the asterisk. If it is 8 or greater (8, 9, A, B, C, D, E, F) the response is negative.

If the response is negative you must be careful when converting it to decimal.

The Standard Approach

- ① Convert the number to decimal using the method described for the W2 command.
- ② Subtract 4,294,967,296 from the number.

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 16^8 (2^{32} or 4,294,967,296)

The Easy Way

- ① Ignore all the leading F's, then convert the hexadecimal number to decimal.
- ② Subtract the next largest power of 16.

Example

The indexer responds to W3 as follows: *FFFF9E58

- ① Chop off the F's: 9E58 hex = 40,536
- ② Subtract from 16^4 10000 hex = 65,536
- ③ Subtraction Result = -25,000

XC Sequence Checksum Report

Version Z5

Type	Status
Syntax	aXC
Units	None
Range	None
Default	None
Response	*nnnnn (where n is a number from 0 to 9)
See also	XD, XE

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

This command reports the nonvolatile memory checksum. After the unit has been programmed, the response can be used for system error checking. The number reported does not indicate the number of bytes programmed. This response is designed to be used for comparison. As long as the sequences are not reprogrammed, the checksum response should always be the same.

Command	Description
> 1XC	*00149

XD Sequence Definition

Version Z5

Type Programming
Syntax <a>XDn
Units Sequence number
Range 1 to 40
Default None
Response None
See also SAVE, XE, XR, XT, XU

Attributes
[x] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

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. If a sequence you are trying to define already exists, you must erase that sequence before defining it. **Immediate commands cannot be entered into a sequence.** Be sure to save your sequence with the SAVE command.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> MN	Sets to Normal mode
> A10	Sets acceleration to 10 rps ²
> V5	Sets velocity to 5 rps
> D10000	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XT	End defining sequence #1
> XR1	Execute sequence #1

XDIR Sequence Directory Report

Version Z5

Type Status
Syntax aXDIR
Units None
Range None
Default None
Response None
See also XBS, XT, XE, XD

Attributes
[x] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

The Compumotor Plus lists the sequence number and amount of memory used for all sequences defined when this command is issued. The response is in the following format:

If no sequences are defined:

```
> 1XDIR *NO_SEQUENCES_DEFINED
>
```

If sequences are defined:

```
> 1XDIR
*SEQUENCE_A_USES_X_BYTES
*SEQUENCE_B_USES_Y_BYTES
>
```

Command	Description
> 1XDIR	
*SEQUENCE_5_USES_251_BYTES	
*SEQUENCE_39_USES_45_BYTES	Reports the number of bytes used by sequences 5 and 39.
>	

XE Erase Sequence

Version Z5

Type Programming
Syntax <a>XEn
Units Sequence number
Range 1 to 40
Default None
Response None
See also XD, XR, XT

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

This command allows you to delete a sequence. The sequence that you specify (n) will be deleted when you issue this command. Caution should be used when executing this command as the sequence is irretrievable.

Command	Description
> XE1	Deletes sequence #1

XFK Set Fault or Kill Sequence

Version Z5

Type Motion
Syntax <a>XFKn
Units Sequence number
Range 0 to 39
Default 0
Response *n (where n is a number from 0 to 39)
See also RSE, XR, K

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

This command selects the sequence that will be executed if a fault condition occurs. No sequence is executed if you select sequence zero. This command is useful if you want to reset an output in case a fault condition exists. You can also use this command to send a message through the RS-232C interface when a fault or kill condition exists. The fault conditions are listed below.

Response	Description	Shown On Front Panel
00	No Errors	Yes
16	Non-commanded drive off	Yes
20	Maximun position error exceeded	Yes
22	Maximun average current exceeded	Yes
23	Drive enable plug not inserted	Yes
30	Battery Backed RAM Checksum error	Yes
60	RS-232C commanded shutdown (Using the ST1 or OFF command)	No
61	Idexer incoming pulses (non-indexer version only)	Yes
66	User defined fault	Yes
88	Microprocessor fault	Yes

Command	Description
> XFK5	Execute sequence #5 when fault or kill condition exists
> XE5	Erase sequence #5
> XD5	Define sequence #5
1 "FAULT_OR_KILL	Send the message
> XT	End sequence definition

XG Goto Sequence

Version Z5

Type Programming
Syntax <a>XGn
Units Sequence number
Range 1 to 39
Default None
Response None
See also XR

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

This command causes your program to jump to a designated sequence for execution. Once you jump to a sequence using the XG command, you cannot return to the sequence from which the XG originated (unless another XG command is executed). To jump to a sequence and return (GOSUB operation), you must use the XR command. There are no limitations on the number of XG commands as there is no nesting involved.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A2	Sets acceleration to 2 rps ²
> V5	Sets velocity to 5 rps
> D10000	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XG5	Go to sequence #5
> XT	End defining sequence #1
> XE5	Erase sequence #5
> XD5	Define sequence #5
> 1PR	Absolute position report
> XT	End sequence #5 definition
> XR1	Execute sequence #1

XQ Set Sequence Interrupted-Run Mode

Version Z5

Type	Set-Up
Syntax	<a>XQ
Units	Mode
Range	0 or 1
Default	0
Response	None
See also	TR, V, IM, SSJ

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

The XQ command allows you to specify that the programmable inputs must all be taken to their inactive state before another sequence may be run. When the unit is in the continuous run (SSJ) mode.

Set interrupted run mode (1)
Clear interrupted run mode (0)

This command can only be used in conjunction with continuous mode (SSJ), and external sequence select lines. If XQ1 is executed, the Compumotor Plus will not accept a sequence selected from the inputs, until all sequence select lines have been brought inactive. After all lines have simultaneously been brought inactive, the Compumotor Plus will then read the sequence select lines and execute the sequence whose number appears there. This interrupted mode will continue until an XQ0 command is executed. You may use S or K commands to stop sequence execution.

Command	Description
> XE40	Erase sequence #40
> XD40	Define sequence #40
> LD3	Disable CW & CCW limits
> SSJ1	Set to Continuous mode
> XQ1	Set to Interrupted mode
> XT	End sequence #100

XR Run a Sequence

Version Z5

Type	Programming
Syntax	<a>XRn
Units	Sequence number
Range	1 to 39
Default	None
Response	None
See also	XD, XE, XT, XRP

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

This command is used to execute sequence n. Once sequence n finishes, the program resumes execution with the command following the XR command. This command is similar to a subroutine or GOSUB statement in Basic. The maximum number of nested XR commands is 16.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A10	Sets acceleration to 10 rps ²
> V5	Sets velocity to 5 rps
> D10000	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XT	End defining sequence #1
> XR1	Execute sequence #1

XRP Sequence Run With Pause

Version Z5

Type Programming
Syntax <a>XRPn
Units Sequence number
Range 1 to 40
Default None
Response None
See also C, XD, XE, XR, XT

Attributes
 Buffered
 Device specific
 Saved independently
 Saved in sequences

The Pause and Run Sequence (XRP) command pauses execution of the current command buffer or sequence and waits for a Continue (C) command. It then executes the sequence specified.

The pause condition is asserted only if the sequence is valid. This command is normally used interactively because it allows you to execute a sequence without the delay of buffering that sequence. An XRP command can be used within one sequence to start execution of another sequence (in this respect an XRP acts like a GOSUB). The maximum number of nested XRP commands is 16.

Command	Description
> XE5	Erases sequence #5
> XD5	Defines sequence #5
> A10	Sets acceleration to 10 rps ²
> V5	Sets velocity to 5 rps
> D10000	Sets distance to 10,000 steps
> G	Execute the move (Go)
> XT	Ends defining sequence #5
> XRP5	Runs sequence #5 with a pause
> C	Compumotor Plus executes sequence #5

XSD Sequence Status Definition Report

Version Z5

Type Status
Syntax aXSD
Units None
Range None
Default None
Response aXSD = *n (where n is a number from 0 to 3)
See also None

Attributes
 Buffered
 Device specific
 Saved independently
 Saved in sequences

This command reports the status of previous sequence definition (refer to the XD and XT commands). The response is in the form *n[cr]. The valid values and descriptions of n are shown below:

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

The XSD command is useful for verifying that the last sequence definition attempt was successful. To retain the sequence, you must save it with the Save (SV) command.

Command	Response
> 1XSD *1	1

XSR Sequence Run Status

Version Z5

Type Status
Syntax aXSR
Units None
Range None
Default None
Response aXSR = *n (where n is a number from 0 to 2)
See also XR, XRP

Attributes
 Buffered
 Device specific
 Saved independently
 Saved in sequences

The XSR command allows you to check whether the last sequence issued was started successfully. The valid values and descriptions for n are shown below.

0 = The last attempt to run the sequence was successful
1 = The last attempt to run the sequence was unsuccessful
2 = An invalid sequence number was requested

Command	Description
> XR2	Runs sequence #2
> 1XSR	Request sequence status
*Ø[cr]	The sequence started OK

XSS Sequence Status

Version Z5

Type	Status
Syntax	aXSSn
Units	Sequence number
Range	None
Default	None
Response	*n (where n is a number from 0 to 2 as listed below)
See also	XD, XE, XT

Attributes	
[x]	Buffered
[]	Device specific
[]	Saved independently
[x]	Saved in sequences

This command reports whether the sequence is empty or programmed. The valid values and descriptions of x are shown below:

- Ø = Empty
- 1 = Bad Checksum
- 2 = O.K.

This command is useful to see if the particular sequence exist and if that portion of memory has been corrupted or not.

Command	Description
> 1XSS1	*Ø [cr] Nothing programmed in that sequence

XST Set Sequence Single-Step Mode

Version Z5

Type	Set-Up
Syntax	aXSTn
Units	Mode
Range	0 or 1
Default	0
Response	*STEP_MODE_ACTIVE or *STEP_MODE_INACTIVE
See also	#, XR, XTR

Attributes	
[x]	Buffered
[]	Device specific
[]	Saved independently
[x]	Saved in sequences

This command puts the Compumotor Plus into Sequence Step mode. This command can only be used with the Step (#) command. When you run a sequence with the Sequence Step mode active, every time you issue a Step (#) command, the controller executes n commands in the sequence buffer.

- XST1: Sequence Step mode active
- XSTØ: Sequence Step mode inactive

Since you need to send a # command over the RS-232C interface, this command cannot be run in stand alone mode. You must be executing the sequence in RS-232C mode. You must enter a delimiter after the Step (#) command to execute the command. If you are in the Trace (XTR) mode, the controller will display n commands every time you enter the #n command. This command is useful for troubleshooting your program to see where you are in the program and what takes place with each command. You can use the Kill (K) command to abort the sequence execution.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A5	Sets acceleration to 5 rps ²
> V2	Sets velocity to 2 rps
> D1ØØØØ	Sets distance to 10,000 steps
> G	Executes the move (Go)
> XT	End defining Sequence #1
> XST1	Enable Single Step mode
> 1XTR1	Enable Trace mode
> XR1	Execute sequence #1
> #	Execute the 1st command
*SEQUENCE_ØØ1_COMMAND_A5	Displays the 1st command executed
> #	Execute the 2nd command
*SEQUENCE_ØØ1_COMMAND_V2	Displays the 2nd command executed
> #	Execute the 3rd command
*SEQUENCE_ØØ1_COMMAND_D1ØØØØ	Displays the 3rd command executed
> #	Execute the 4th command
*SEQUENCE_ØØ1_COMMAND_G	Displays the 4th command executed motor should have moved 10,000 steps
> #	Execute the 5th command
SEQUENCE_ØØ1_COMMAND_XT	Displays the last command executed

XT Terminate (End) Sequence

Version Z5

Type	Programming
Syntax	<a>XT
Units	None
Range	None
Default	None
Response	None
See also	XD, XE, XR

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

The XT command 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.

Command	Description
> XD1	Define sequence #1
> MN	Set to Normal mode
> A10	Sets acceleration to 10 rps ²
> V5	Sets velocity to 5 rps
> D25000	Sets distance to 25,000 steps
> G	Executes the move (Go)
> XT	End sequence definition

XTR Set Trace Mode

Version Z5

Type	Set-Up
Syntax	aXTRn
Units	Mode
Range	0 or 1
Default	0
Response	*TRACE_MODE_ACTIVE or *TRACE_MODE_INACTIVE
See also	XR, XST

Attributes
[X] Buffered
[] Device specific
[] Saved independently
[X] Saved in sequences

This command transmits the command that is being executed from the Compumotor Plus to the host via RS-232C interface. This command only works if you are running a sequence.

XTR1: Enables Trace mode
XTR0: Disables Trace mode

Enabling trace mode transmits the commands and the sequence number being executed. If you have a Loop (L), REPEAT, or WHILE command in a sequence, it will also display the iteration count.

This command is useful if the user wishes to see where you are in the program as the program is being executed.

Command	Description
> XE1	Erase sequence #1
> XD1	Define sequence #1
> A10	Sets acceleration to 10 rps ²
> V5	Sets velocity to 5 rps
> D25000	Sets distance to 25,000 steps
> L2	Loop 2 times
> G	Executes the move (Go)
> N	End Loop
> XT	End defining sequence #1
> 1XTR1	Enable Trace mode
> XR1	Execute sequence #1

After turning on the Trace mode, as you run the sequence (XR1), the controller will display the current command being executed. A sample of Trace mode output is shown below:

```
*SEQUENCE_001_COMMAND_A10
*SEQUENCE_001_COMMAND_V5
*SEQUENCE_001_COMMAND_D25000
*SEQUENCE_001_COMMAND_L2
*SEQUENCE_001_COMMAND_G_LOOP_COUNT_001
*SEQUENCE_001_COMMAND_N_LOOP_COUNT_001
*SEQUENCE_001_COMMAND_G_LOOP_COUNT_002
*SEQUENCE_001_COMMAND_N_LOOP_COUNT_002
*SEQUENCE_001_COMMAND_XT
```

XU Upload Sequence

Version Z5

Type	Programming
Syntax	aXUn
Units	Sequence number
Range	1 to 40
Default	None
Response	*x (where x is the contents of sequence n)
See also	XD, XE, XT

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

The XU command sends all the characters in the requested sequence out the RS-232C port. The response is preceded with an asterisk (*) and terminated by an extra carriage return. All command delimiters in the sequence (spaces and carriage returns) are converted to underscores. If the sequence is empty, *_ is sent.

Command	Description
> 1XD1 A10 V10 D5000 G XT	Define sequence #1
> 1XU1	Upload sequence #1
1XD1_A10_V10_D5000_G	Response from Compumotor Plus
>	

Y Stop Loop

Version Z5

Type	Programming
Syntax	<a>Y
Units	None
Range	None
Default	None
Response	None
See also	L, LRD, N

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

The Stop Loop (Y) command exits a loop when the loop completes its current pass. This command does not halt processing of the commands in the loop until the Compumotor Plus reaches the last command of the current loop. At that time, the Compumotor Plus executes the command following 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, or restart the sequence.

Command	Description
> L	Begin infinite loop
> A10	Set acceleration to 10 rps ²
> V5	Set velocity to 5 rps
> D25000	Set distance to 25,000 steps
> T2	Wait 2 seconds
> G	Execute the move (Go)
> N	End loop
> Y	Stop loop upon receipt of the command

Z Reset

Version Z5

Type	Programming
Syntax	<a>Z
Units	None
Range	None
Default	None
Response	None
See also	K, S

Attributes
[] Buffered
[] Device specific
[] Saved independently
[] Saved in sequences

The Reset (Z) command is equivalent to cycling power to the Compumotor Plus. This command returns all internal settings to their power-up values. It clears the command buffer and sets all position counters to zero.

When you use the Reset command, the Compumotor Plus is busy for 2.5 seconds and ignores all commands during that time.