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

E Enable Communications Interface

Version Z5

Type	Programming
Syntax	<a>E
Units	None
Range	None
Default	None
Response	None
See also	F

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

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

Command	Description
> F	Disables all revs (axes) on the communications interface
> 1E	Enables device #1
> 4E	Enables device #4
> G	Executes the move (Go — only axes 1 and 4 will move)

ELSE Else

Version Z5

Type	Programming
Syntax	<a>ELSE
Units	None
Range	None
Default	None
Response	None
See also	IF, NIF, IFTR, IFFL, IFER

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

The ELSE command is used in conjunction with the four IF commands and the end of IF...NIF statement.

If the IF condition evaluates **true**, then the commands between the IF and ELSE commands are executed and the commands between the ELSE and the NIF command are skipped.

Conversely, if the IF condition evaluates **false**, then the commands between the IF and the ELSE are ignored and the commands between the ELSE and NIF are executed.

The ELSE command is optional and does not have to be included in the IF statements. IF...ELSE...NIF statements take the general form show below.

IF(condition) commands ELSE commands NIF

Command	Description
>IF(INXXX1_OR_VARI>20)	If input status is XXX1 (input #1 is active) or variable 1 greater than 20 then execute the command following the ELSE command
> GD1	Execute predefined move #1
> ELSE	Else
> GD2	Execute predefined move #2
> NIF	End of it

F Disable Communications Interface

Version Z5

Type	Programming
Syntax	<a>F
Units	None
Range	None
Default	None
Response	None
See also	E

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

The F command causes the Compumotor Plus to ignore all commands over the RS-232C interface except the Enable (E) command. The command is useful when you are programming multiple units on a single RS-232C interface. You send an F command to axes that you want to exclude from global commands. This allows you to program other units on the daisy chain without including a device identifier with every command.

This command is also useful when uploading programs (XU) from a unit on a daisy chain. If you do not disable the other units in a daisy chain, uploading programs may cause other units on the daisy chain to execute uploaded commands.

Command	Description
> 1F	Disables the communications interface on units with device address #1
> 3F	Disables the communications interface on units with device address #3
> G	All of the indexers (not 1 and 3) will execute a move (Go)

FILT Enable Digital Torque Filter

Version Z5

Type	Set-Up
Syntax	<a>FILTn
Units	ms
Range	0.3 to 100
Default	5
Response	None
See also	TUNE, GAINX

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

The FILT command implements a digital filter on the torque command prior to sending this command out to the amplifier. The parameter specifies the speed of the filter in milliseconds. This corresponds to the bandwidth of the filter (e.g., FILT5 implements a 5mS filter which has a bandwidth of 1/5mS or 200Hz). The higher the number the lower the bandwidth of the torque command.

The presence of the filter significantly reduces audible noise during motion and at standstill. This allows you to use much higher gains than previously possible. The slower the filter the quieter the system, but performance becomes more sluggish. You must assess the dynamic needs of your system in order to decide what filter value to use.

A low bandwidth application, such as a continuous move with low acceleration and deceleration rates can use, for instance, a 100Hz filter (FILT10) with high gains for stiff but quiet action. For applications with high accelerations, filters above 200Hz are more appropriate. FILTØ turns off the filter entirely.

G Go

Version Z5

Type	Motion
Syntax	<a>G
Units	None
Range	None
Default	None
Response	None
See also	A, D, MC, MN, S, V

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

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

In Continuous mode (MC) the Go command initiates a move which must be terminated by an explicit stop action such as the Stop command or stop on trigger. In the Continuous mode (MC), you only need to enter the A and V commands prior to the G command. The system ignores the D command in this mode.

In Preset mode (MN) the distance the motor moves is affected by the current positioning mode: incremental or Absolute.

In Incremental mode (MPI), a G command causes the motor to move the distance you specified with the D command.

In Absolute mode (MPA), a G command interprets the distance as absolute with reference to absolute zero. Refer to the Distance (D) command for more information.

If the motor does not move in response to a G command the problem could be:

- ❑ The indexer is in Absolute Positioning mode and the motor is already at the absolute position specified an active end-of-travel limit switch may be on. Check the limit switches.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> A1	Set acceleration to 1 unit/sec^2
> G	Execute the move (Go)

GA Go Home Acceleration

Version Z5

Type	Motion
Syntax	<a>GAn
Units	rps/sec
Range	0.001-2147483.647 rps/sec
Default	99
Response	*nnnnnnnnn
See also	A, GH, OS

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

The Go Home Acceleration(GA) command sets the linear acceleration value that the motor will use during any subsequent Go Home (GH) moves.

Command	Description
> GA5	Set go home acceleration to 5 rps^2
> GH-5	The motor accelerates at 5 rps^2 to 5 rps in the CCW direction and searches for home

GAINX Set Gain Type

Version Z5

Type	Set-Up
Syntax	aGAINX:n
Units	0
Range	0 or 1
Default	None
Response	None
See also	FILT, TUNE

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

The GAINX command enables self-tuning or PIDV tuning.

The GAINX command is a buffered command which allows you to switch between the pole-placement control scheme resulting from self-tuning and the PIDV gain scheme. This command can be put in a sequence or its result can be saved in non-volatile memory. Since the two gain schemes are not directly compatible, this command can be used to take advantage of both self-tuning and the flexible PIDV control scheme. The format of this command is as follows:

GAINX:0 Enable PIDV tuning
GAINX:1 Enable self-tuning

The Compumotor Plus allows you to select whether you want to set the systems gains yourself using the CIG, CDG, CPG and CVG commands or use the gains the Compumotor Plus has chosen in response to the TUNE command. Refer to the TUNE and FILT commands for details.

Do not use GAINX:1 unless you have gone through the self-tuning procedure. Failure to do so will lead to unpredictable results. It is always a good idea to make sure your maximum position error is set to a relatively low value (e.g., 1 rev) while doing any sort of tuning.

Command	Description
GAINX:1	Select self tuning gains
A99	Set acceleration to 99 rps/s
V5Ø	Set velocity to 50 rps
D1ØØØØØ	Set distance to 100,000 steps
G	Execute the move
GAINX:Ø	Select PIDV gains

GD Go Defined

Version Z5

Type	Motion
Syntax	<a>GDn
Units	Move number
Range	1 to 32
Default	None
Response	None
See also	GDEF, G

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

The Go Defined (GD) command executes a move that has been predefined and precalculated with the move definition (GDEF) command.

The command sequence An Vn Dn G is compressed to GD. You set the acceleration, velocity and distance with the GDEF command. The values specified by the A, V, and D commands are not changed by the GDEF command. The delay from the time the command is processed until motion occurs is approximately 4ms which is significantly faster than executing the series: An, Vn, Dn, G.

Command	Description
> A1 V1 D1000	Set accel, vel and distance
> GDEF10,A25,V20,D500	Define and calculate move #10
> GDEF16,A2,V1,D200	Define and calculate move #16
> GD10	Execute predefined move #10
> GD16	Execute predefined move #16
> G	Use A1, V1, D1000 from above and move

GDEF Move Definition

Version Z3

Type Motion
Syntax <a>GDEFn,Ab,Vc,De
Units b = acceleration, c = velocity, e = distance
Range 1 to 10
Default None
Response None
See also GD

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

The GDEF command defines move number n with parameters A (acceleration), V (velocity), and D (distance). The parameters for the move are calculated when the GDEF command is issued. The move may be executed using the GD command. Because the GDEF command is precalculated there is virtually no delay from the issuance of the GD and motion start.

You may define up to 32 predefined moves. The moves may be used any number of times in any sequence.

Command	Description
> GDEF1,A5,V5,D250000	Define and calculate move #1
> GDEF2,A20,V10,D100000	Define and calculate move #2
> GD1	Execute predefined move #1
> GD2	Execute predefined move #2

GH Go Home

Version Z5

Type Motion
Syntax <a>GHn
Units rps
Range ±0-49.99 rps
Default None
Response *nnnnnnnn
See also RC, OS, GA, GHF

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

The Go Home (GH) command instructs the Compumotor Plus to search for an externally specified reference position. The external position is specified using the home input. Once home, the Compumotor Plus resets the absolute position to zero. If another position is desired, you can use the Buffer Set Absolute Position (BSP) command to specify the position when home.

The Compumotor Plus searches for home as follows:

① Search using the direction and velocity specified in the GH command.
② When the home input changes state, the Compumotor Plus stops the motor using the Go Home Acceleration (GA).
③ If the home input is inactive as defined by the OSC command the motor reverses direction to re-enter the *home region*.
④ If the OSB command is set to 0 the motor stops, and considers itself at home. If the OSB command is set to 1 the motor moves to find the home edge defined by the OSH command. It does this at the Go Home Final velocity (GHF).
⑤ The motor then moves to approach home from the direction defined in the OSG command.
⑥ If the OSA1 command has been issued the motor then moves to the resolver position as defined by the GHP command.

If the system encounters an end-of-travel limit during the homing sequence, the motor reverses direction. If a second limit is encountered during homing, the homing sequence is aborted. You can use the RB command to find out if the home attempt was successful.

Command	Description
> GH-20	The motor moves in the negative direction at 20 rps and looks for the home limit input to go active

GHF Go Home Final Velocity

Version Z5

Type	Motion
Syntax	<a>GHFn
Units	rps
Range	0.00 to 9.99
Default	None
Response	*nn.nn
See also	GH, OS, GA

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

This command sets the velocity for the final approach to home in the go home sequence of the GH command.

Command	Description
> GHF.1	The velocity of the final approach of the next go home move will be 0.1 rps
> GH2	Execute go home at 2 rps^2 in CW direction

GHP Define Resolver Home Position

Version Z5

Type	Motion
Syntax	<a>GHPn
Units	motor steps
Range	0 to 255
Default	128
Response	None
See also	GH, GA, GHF, OSA, OSB, OSC, OSG, OSH

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

This command defines the resolver position the motor moves to at the end of a Go Home sequence (GH). The Compumotor Plus does not move to a resolver position unless the OSA command is set to 1 and the home limit is active as defined by the OSC command.

Compumotor Plus motors have 50 poles and a built in resolver which can resolve 256 steps within each pole. Therefore, a given resolver position is only unique with 7.2° of shaft rotation. This command provides an accurate and repeatable home position at the end of a Go Home sequence defined by OSB1 and the OSG and OSH commands. It is necessary that the GHF definition be such that the home limit switch causes the motor to stop repeatably within ±3.6° for the Go Home to Resolver position to be repeatable.

Command	Description
> GHP15	Defines the home resolver position to be 15

^H Backspace (Control-H)

Version Z5

Type	Programming
Syntax	<a>^H
Units	None
Range	None
Default	None
Response	None
See also	None

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

The Backspace command (Control-H) allows you to delete the last character that you issued to the Compumotor Plus. (^H indicates that the Ctrl key is held down as the H key is pressed.)

Note: Once a command has begun execution the Backspace command has no effect.

The backspace (^H) command does not prevent execution of immediate commands after you send a delimiter (i.e., space or carriage return). This command tells the indexer to backup one character in its input buffer, regardless of what may be showing on a terminal. On some terminals, the Ctrl and the left arrow (←) key issues a backspace character.

H Set Direction

Version Z5

Type	Motion
Syntax	<a>H<n>
Units	Direction
Range	Nothing, + or -
Default	None
Response	None
See also	D

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

The Set Direction (H) command changes or defines the direction of the next move that the system will execute.

This command does not effect moves already in progress.

H+ = Set move to CW direction
H– = Set move to CCW direction
H = Reverses direction from the previous setting

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

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D25000	Set distance to 25,000 steps
> G	Execute the move (Go) in CW direction
> H	Reverse direction
> G	Execute the move (Go) in CCW direction
> MC	Set to Continuous mode
> H+	Set direction to CW
> G	Move continuously in CW direction

IFER If Error Condition

Version Z5

Type Status
Syntax <a>IFER
Units None
Range None
Default None
Response None
See also IFER, IFIN, IFTR, NIF, RSE, XFR

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

This command checks to see if any fault condition or limit condition exists. If one does, the Compumotor Plus executes a conditional command. This command is very similar to the IF_THEN statement in basic programming.

If any hard or soft limit has been hit or if an excessive position error, excessive average current error, drive enable disconnect error, or EEPROM error is detected, the commands between IFER and NIF will be executed. At present this command does not distinguish between error conditions in the Compumotor Plus.

An IFxx command (IFTR, IFER, IFFL) cannot be used within another IF command.

Command	Description
> L	Loop forever
> IFER	If error condition do the following commands
>"ERROR!	Send the string "ERROR!" over RS-232C
> ST1	Turn off amplifier
> NIF	End of IF statement
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D1000	Set distance to 1000 steps
> G	Execute the move (Go)
> N	End of loop

IFFL Compare User Flag

Version Z5

Type Programming
Syntax <a>IFFLnnnnnnnn
Units None
Range None
Default None
Response None
See also NIF, SFL, IFER, IFIN

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

The IFFL command uses the pattern set by the Set User Flag (SFL) command to determine if it should execute commands between the IFFL command the the next NIF command. This command is similar to the IF...THEN compound statement found in Basic programming.

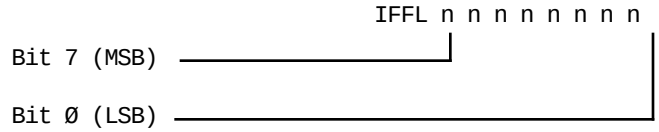
The 8 bits set by the SFL command are compared to the 8 flags following the IFFL command. If the patterns match, the commands following the IFFL command up to the next NIF are executed. If the bit patterns do not match the program skips to the command immediately following the next NIF command. An IFxx command (IFTR, IFER, IFFL) cannot be used within another IF command.

Each flag may have a one of three values shown below.

0 = not set,
1 = set

X = don't care (i.e., don't look at this flag)

This command is useful if you wish to make a decision based on previous sequence executions than will set or clear user flag bits. Also you can use a host computer to generate the SFL (User Flag) pattern depending on the condition, then you can use IFFL to perform branching type of operations.



Command	Description
> SFL1Ø1	Set user flag bits 7 and 5 and clear bit 6, the remaining bits are left unaltered
> IFFL1Ø1Ø	If user flag bits 5 and 7 are set and bits 6 and 4 clear, do the following commands
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> NIF	End of statement
	If the IFFL pattern matches the SFL setting the motor moves 25,000 steps.

Command	Description
> SFLX1X1	Set bits 4 and 6
> IFFL1Ø11XX1Ø	If bits 1, 4, 5, and 7 are set and bits Ø and 6 are not set do the following commands
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D5ØØØ	Set distance to 5,000 steps
> G	Execute the move (Go)
> NIF	End of statement
> IFFLXXX1	If bit 4 on SFL is set, do the following commands
> A1Ø	Set acceleration to 10 rps^2
> V2	Set velocity to 2 rps
> D-25ØØØ	Set distance to 25,000 steps in the CCW direction
> G	Execute the move (Go)
> NIF	End of statement

There are 2 IF_THEN statements in this example. The first one does not match the user flag, and skips to the command after NIF. The second IFFL pattern does match the SFL (User flag) pattern, therefore the motor moves -25,000 steps.

IFIN If Input

Version Z5

Type Programming
Syntax <a>IFINnnnnnnnnn
Units None
Range None
Default None
Response None
See also IFER, IFTR, IFFL

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

The IFIN command checks to see if the inputs match the parameters listed after the command. If they match, sequence execution continues with the command immediately after the IFIN command. If they do not match, sequence execution contues after the next ELSE or NIF statement.

This command differs from the IFTR command in that this command looks at all the inputs (including CW, CCW, and Home) and the IFTR command looks only at the trigger inputs as defined with the IM command.

IFTR If Trigger

Version Z5

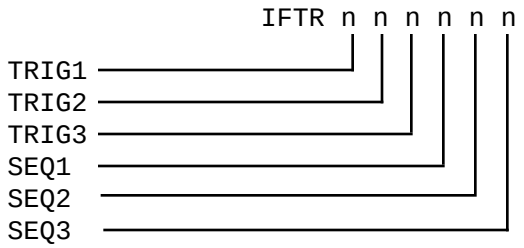
Type Status
Syntax <a>IFTRnnnnnn
Units trigger condition
Range 0, 1, or X
Default None
Response None
See also IFTR, IFFL, IFER, NIF, TR

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

This command uses the input pattern (6 inputs) of the Compumotor Plus Controller to execute the conditional commands. This command is very similar to the IF_THEN statement in basic programming. The difference is, the 6 input bits will determine the next command to be executed. If the hardware input patterns match the IFTR input pattern, then the commands between IFIN and NIF will be executed. If patterns do not match, the command following the NIF will be executed. For this command, 1 means active and 0 means not active. An X means the Compumotor Plus doesn't care whether that particular input is active or not. An IFxx command

(IFTR, IFER, IFFL) cannot be used within another IF command. *Note: IF commands may not be nested.*

This command is useful for branching and performing conditional moves. For example, depending on where you are, you may want to perform different moves.



Command	Description
> IFTR1ØX	If TRIG1 is active and TRIG2 is not active, issue the following commands
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> G	Execute the move (Go)
> NIF	End of IF statement
> IFTRXXX1ØX	If SEQ1 is active and SEQ2 is not active, issue the following commands
> A1Ø	Set acceleration to 10 rps^2
> V1	Set velocity to 1 rps
> G	Execute the move (Go)
> NIF	End of IF statement

IM Input Mode Configure

Version Z5

Type Set-Up
Syntax <a>IMn
Units Input number
Range 1 to 5
Default 1
Response n
See also SN, SS

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

This command configures the auxiliary inputs for different functions. By reconfiguring the inputs, the same input may be used for different functions such as trigger inputs, sequence select inputs, jog inputs, registration input, or stop input.

This command specifies how you may define the three sequences and three trigger inputs as follows.

Mode 1 The inputs are used to select sequences 1 through 7. (Note the BCD weighting of the sequence lines listed below.)

Mode 2 The inputs are used to select sequences 1 through 39, using BCD weighting for the input lines.

Mode 3 Dedicates the triggers for use as Stop, Jog+, and Jog-.

The Trigger Pause (TR) command may still be used to halt sequence execution on any input line even though it is defined as a sequence select by this mode. Because of this, if you are not using all of your input lines for sequence selection you may use the unused inputs as trigger inputs even though they have been defined as sequence selection inputs.

Sequence 40 may not be selected via the sequence select inputs. Sequence 40 is always run on power up and may be run via the XR4Ø command over the RS-232C interface.

In input modes 1 through 3, the Continuous Sequence Scan (SSJ1) command allows sequences to be selected by grounding the appropriate sequence select inputs to obtain a desired BCD value. If the Sequence Hold (XQ1) command is issued, the input lines must be released (ungrounded) before another sequence may be run. Grounding all sequence select lines causes no sequence to be selected.

Changing the BCD values of sequence input lines results in a new sequence being run that corresponds to the new value after the current sequence is completed. The BCD sum of the values of the grounded lines determines which sequence is run. For example, grounding SEQ1 and SEQ2 results in sequence 6 being run (refer to the table below).

The SN command is used to determine how long the input must be stable before running a new sequence.

The following table shows the different input modes available.

Input Mode 1	Front Panel Designation	IM1	BCD
	TRIG 1	Trigger 1	Ø
	TRIG 2	Trigger 2	Ø
	TRIG 3	Trigger 3	Ø
	SEQ 1	Sequence Select	4
	SEQ 2	Sequence Select	2
	SEQ 3	Sequence Select	1

Input Mode 2	Front Panel Designation	IM2	BCD
	TRIG 1	Sequence Select	2Ø
	TRIG 2	Sequence Select	1Ø
	TRIG 3	Sequence Select	8
	SEQ 1	Sequence Select	4
	SEQ 2	Sequence Select	2
	SEQ 3	Sequence Select	1

Input Mode 3	Front Panel Designation	IM3	BCD
	TRIG 1	STOP	Ø
	TRIG 2	JOG +	Ø
	TRIG 3	JOG -	Ø
	SEQ 1	Sequence Select	4
	SEQ 2	Sequence Select	2
	SEQ 3	Sequence Select	1

Input Mode 4	Front Panel Designation	IM4	BCD
	TRIG 1	Registration	0
	TRIG 2	JOG +	0
	TRIG 3	JOG -	0
	SEQ 1	Sequence select	4
	SEQ 2	Sequence select	2
	SEQ 3	Sequence select	1

Input Mode 5	Front Panel Designation	IM5	BCD
	TRIG 1	Registration	0
	TRIG 2	Trigger	0
	TRIG 3	Trigger	0
	SEQ 1	Sequence select	4
	SEQ 2	Sequence select	2
	SEQ 3	Sequence select	1

Command	Description
> XE4Ø	Erases sequence #40
> XD4Ø	Defines sequence #40
> IM1	Changes to input mode #1
> XR1Ø	Runs sequence #10
> XT	Ends sequence definition
	Indexer will run sequence #10 upon power up.

IO Immediate Output

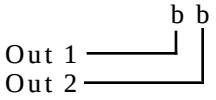
Version Z5

Type Set-Up
Syntax <a>IObb
Units State of an output
Range 0 or 1
Default None
Response None
See also O

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

This command immediately sets the output bits as specified in the pattern. Set the output mode (O) command for other uses of the outputs.

This command allows output to be used as general-purpose outputs, which can be controlled through RS-232C (independent of motor movement).



Command	Description
> MN	Set to Normal mode
> A1Ø	Set acceleration to 10 rps^2
> V1	Set velocity to 1 rps
> D9999	Set distance to 9,999 steps
> IO11	Turn outputs 1 and 2 on before beginning of move
> G	Execute the move (Go)
> IOØØ	Turn outputs 1 and 2 off after the move

IS Input Status

Version Z5

Type	Status
Syntax	<a>IS
Units	None
Range	None
Default	None
Response	*bbb_bbb_bbb_b
See also	None

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

Reports the status of all hardware inputs. Including trigger, sequence, select lines, and limits.

This is not a software status. It will report the actual hardware status of the inputs. This command is useful for troubleshooting an application, to verify that Limit switches, Trigger inputs and home switches work.

Command	Description
> 1IS	Input status of device 1 is reported

JA Jog Acceleration

Version Z5

Type	Motion
Syntax	<a>JAn
Units	rps/sec
Range	0.001 to 99.999 rps/sec
Default	99
Response	None
See also	JV, IM, OS

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

The jog acceleration command allows you to specify the linear acceleration rate that will be applied to the motor until it reaches constant jog velocity. The jog acceleration remains set until you change it.

Command	Description
> JA3Ø	Set jog acceleration to 30 rps^2
> JV2	Set jog velocity to 1 unit/sec

JV Jog Velocity

Version Z5

Type	Motion
Syntax	<a>JVn
Units	rps
Range	0.00 to 50.00
Default	1
Response	None
See also	JA, IM, OS

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

This command allows you to set the maximum jog velocity. Jogging allows you to manually position the motor. When you jog the motor, it will accelerate to the velocity specified and remain at this speed until you turn off the jog input.

Command	Description
> JA1Ø	Set jog acceleration to 10 rps^2
> JV5	Set jog velocity to 5 rps

K Kill Motion

Version Z5

Type	Motion
Syntax	<a>K
Units	None
Range	None
Default	None
Response	None
See also	S

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

The Kill (K) command is an emergency stop command. This command causes indexing to cease immediately. The Compumotor Plus attempts to stop the motor instantaneously.

CAUTION

Because of the high deceleration rate commanded by the K command, the motor may lose torque, particularly when driving large inertial loads being moved at high speeds. The load could drive itself past limit switches and cause damage to the mechanism and or personnel.

The K command clears all indexer flags (e.g., Display Flags of Indexer (DFX) command will report all zeros.)

In addition to stopping the motor, the K command terminates any loops, time delays or down-loads in process. It also clears the command buffer.

Command	Description
> MC	Set to Continuous mode
> A5	Set acceleration to 5 rps^2
> V2	Set velocity to 2 rps
> G	Execute the move (Go)
•	
•	
•	
> K	Commands the motor to stop instantly

KILL Kill Motion

Version Z5

Type	Motion
Syntax	KILL
Units	None
Range	None
Default	None
Response	None
See also	None

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

The KILL command is identical to the K command. See the K command for a complete description.

L Begin Loop

Version Z5

Type	Programming
Syntax	<a>Ln
Units	number of loops
Range	0 to 65,535
Default	None
Response	None
See also	N, Y

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

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 the L command without a value specified for n, or with a 0, subsequent commands will be repeated continuously.

The End-of-Loop command prompts the indexer to proceed with further commands after the designated number of loops have been executed. The Stop Loop (Y) command indicates where execution will stop. The Immediate Pause (U) command allows you to temporarily halt loop execution. You can use the Continue (C) command to resume loop execution.

Loops may be nested up to 16 levels deep. Possible structures are shown below:

```
L N
L L N N
L L N L N N
L L L L N N N N
L L L N  L N N L N N
```

Command	Description
> L5	Loops 5 times
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D1ØØØØ	Set distance to 10,000 steps
> G	Execute the move (Go)
> N	Specifie the above 10,000-step move to be repeated five times

LA Acceleration Limit

Version Z5

Type Set-Up
Syntax <a>LAn
Units rps/sec
Range 0.001 to 2147483.647
Default 900
Response None
See also A, LD

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

The Limit Acceleration (LA) command allows you to define the deceleration rate that should be used when an end-of-travel limit is encountered. This command is useful if you do not want an abrupt stop upon encountering a limit. However, you should be careful to specify a deceleration rate that will stop the load before it can do any damage. Normally, limit switches are placed so that the motor has room to safely decelerate the load.

Command	Description
> LA5Ø	The motor decelerates at 50 rps^2 when it encounters an end-of-travel limit.

LD Limit Disable

Version Z5

Type Set-Up
Syntax <a>LDn
Units State of limit enable
Range 0 to 3
Default 0
Response None
See also None

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

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 the LD3 command.

When you disable limit inputs, they are reconfigured as trigger inputs.

Enable CCW and CW limits	0 (Default)
Disable CW limits	1
Disable CCW limits	2
Disable CCW and CW limits	3

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

LED Show Number on LED Display

Version Z5

Type	Programming
Syntax	<a>LDn
Units	Number to display
Range	0 to 99
Default	None
Response	None
See also	None

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

This command allows the user to set a number to the 2 digit LED display on the Compumotor Plus. This command is useful when you want the Compumotor Plus to interact with the operator. For example, you might define number 90 as the moving status, or number 95 as stopped status.

Command	Description
> PS	Pause execution of following commands until the Continue (C) command is issued
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25000	Set move distance to 25,000 steps
> LED90	Turn on 2 digit LEDs to 90 (Moving)
> G	Execute the move (Go)
> LED95	Turn on 2 digit LEDs to 95 (stopped)
> C	Continue execution

LF Line Feed

Version Z5

Type	Programming
Syntax	aLF
Units	None
Range	None
Default	None
Response	[lf] where [lf] is a line feed character (ASCII 10)
See also	CR, "

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

The LF command causes the Compumotor Plus to send a line feed character (ASCII 10) when the command is processed. Since the command is buffered it may be executed interactively or in a sequence.

If you place the LF command after a Go (G) command, the line feed is sent at the completion of the move. This is because both the LF and G commands are buffered. Placing a LF command after a Trigger (TR) command, causes the line feed to be sent when the trigger condition is met.

Command	Description
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D15000	Set distance to 15,000 revs
> G	Execute the move (Go)
> LF	Transmits a line feed character over the communications interface after the move is completed

MC Mode Continuous

Version Z5

Type	Motion
Syntax	<a>MC
Units	None
Range	None
Default	None
Response	None
See also	A, D, MN, MPA, MPI, V

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

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 MC command will only change the mode when the motor is not moving.

The Indexer uses the 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.

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)

MN Mode Normal

Version Z5

Type Motion
Syntax <a>MA
Units None
Range None
Default None
Response None
See also MC, D, A, V, MPA, MPI

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

The Mode Normal (MN) command sets the positioning mode to preset. In Normal mode, the motor will move the distance specified with the 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. The MN command can only have an effect when the motor is not moving.

Command	Description
> MN	Set to Preset Positioning mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D1000	Set distance to 1,000 steps
> G	Execute the move (Go)

MPA Mode Position Absolute

Version Z5

Type Set-Up
Syntax <a>MPA
Units None
Range None
Default None
Response None
See also D, MC, MN, MPI, PZ

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

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

Position Absolute mode (MPA) is most useful in applications that require moves to specific locations.

You can set the absolute counter to zero by cycling power or issuing a Position Zero (PZ) command.

Command	Description
> MN	Set to Normal mode
> MPA	Set to Position Absolute mode
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D25000	Set distance to 25,000 steps
> G	Motor will move to absolute position 25,000
> D12500	Set absolute position to +12,500 steps
> G	Motor will move to absolute position +12,500 steps

MPI Mode Position Incremental

Version Z5

Type Set-Up
Syntax <a>MPI
Units None
Range None
Default None
Response None
See also MPA, D, MN

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

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. The MPI command is most useful in applications that require repetitive movements, such as feed to length applications.

Command	Description
> MN	Set to Normal mode
> MPI	Set to Positioning Incremental mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D1ØØØØ	Set distance of move to 10,000 steps
> G	Move 10,000 steps CW
> G	Move another 10,000 steps CW

N End of Loop

Version Z5

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

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

This command marks the end of loop. You can use this command in conjunction with the Loop (L) command. All buffered commands that you enter between the L and N commands are executed as many times as the number that you enter following the L command.

You may nest loops 16 levels deep.

Command	Description
> PS	Pauses execution of buffered commands until the indexer receives a Continue (C) command
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> D1ØØØØ	Set move distance to 10,000 steps
> L5	Loops the above series of commands five times
> G	Execute the move (Go)
> N	Ends the loop
> C	Clears pause and executes all the buffered commands

NIF End of IFxx Command

Version Z5

Type Programming
Syntax <a>NIF
Units None
Range None
Default None
Response None
See also IFFL, IFER, IFIN, IFTR

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

This command marks the end of IF compound command. The Compumotor Plus has four IF commands.

IFFL If the User Flag pattern matches, execute the following commands
IFER If certain error conditions exist, execute the following commands
IFIN If the input pattern matches, execute the following command
IFTR If the trigger input pattern matches, execute the following command

When you use any of the four commands, listed above, you must complete the command by placing the NIF command after the commands you want to execute should the IFxx command test true.

In case the IF condition does not match, the commands between IF and NIF are not executed, and the program will skip to the command following the NIF command.

Command	Description
> IFTRXXX1Ø	If SEQ 1 is active and SEQ 2 is not active issue the following commands and go to command following the NIF command
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> NIF	End of statement
> IFTRXXXØ1	If SEQ 1 is not active and SEQ 2 is active, issue the following commands, skip to the command following NIF
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D5ØØØ	Set distance to 5,000 steps in the opposite direction
> G	Execute the move (Go)
> NIF	End of statement
> IFTRXXX1	If SEQ 1 is active, issue the following command, then skip to the command following the **NIF** command
> 1"DONE	Transmit message saying done
> NIF	End of statement

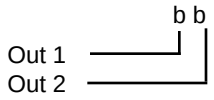
O Output

Version Z5

Type	Programming
Syntax	<a>Obb
Units	Output—on or off
Range	0, 1 or X
Default	None
Response	None
See also	IO, OM

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

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



Command	Description
> A1Ø	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D2ØØØØ	Set move distance to 20,000 steps
> OØ1	Set programmable output 1 off and output 2 on
> G	Execute the move (Go)
> OØØ	After the move ends, turn off output 2

OFF Amplifier Off

Version Z5

Type	Programming
Syntax	<a>OFF
Units	None
Range	None
Default	None
Response	None
See also	ON, STØ, ST1

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

This command allows you to shut the amplifier off and remove current from the motor. You must issue an ON command to re-energize the motor. The contents of the buffer will be cleared when you execute this command.

The OFF command removes motor torque, allowing you to move the motor manually. This command is similar to the Shutdown (ST1) command. However, this command is immediate, and ST1 is buffered.

Command	Description
> OFF	Powers down the motor (no torque)

OM Output Mode

Version Z5

Type	Set-Up
Syntax	<a>OMn
Units	Mode
Range	1 to 6
Default	1
Response	None
See also	IO, O, CDB

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

This command allows you to set different output modes.

Mode	Output 1	Output 2
OM1	Programmable	Programmable
OM2	Programmable	Slip Fault (In Position)
OM3	Motor Moving	Programmable
OM4	Motor Moving	Slip Fault (In Position)
OM5	Motor Moving/Not Moving	Registration Occurred in the current or previous move
OM6	Motor Moving/Not Moving	Registration Failed. The previous mode was not terminated by a registration move

Programmable means the output may be turned on and off with the Output (O) and Immediate Output (IO) commands.

Motor Moving means the output is on when the internal indexer is commanding the motor to move. It is off when the internal indexer is commanding the motor to hold position. Note: The servo loop can cause motion

which will not turn on the motor moving output. For example, if the motor is stationary and you move the motor shaft with your hand and the motor makes a move to correct its position, the motor moving output will not turn on, because the indexer did not command the motion.

Slip Fault is turned on when the deadband specified is exceeded (refer to the CDB command for more information). It is turned off when a slip fault is detected. The mode you use will depend upon your application. If you need to control other devices using programmable outputs, run your application in OM2 mode. If you are only interested in detecting a slip fault or monitoring the motor movement, use the OM4 mode.

Registration Occured turns on Out 2, which becomes active as soon as registration is detected and stays active until the next move is begun. Thus it stays active **after** the registration move is over. This feature was added to allow signalling of an error condition. If the output never comes on, registration has not occured during the previous move. This gives you a way to detect if the registration sensor is functioning properly, and that your mechanical system is aligned properly.

Registration Failed turns on Out 2, which is used to signal an error condition. This output comes on if, during the previous move, a registration signal was not detected. (Registration must be enabled to detect the registration sensor.) This allows external controllers to let the registration moves occur continuously in a sequence while an external controller monitors Out 2. If the output ever comes on the controller knows that a move has occured in which no registration mark was detected. This may be an error condition in some systems, warranting stopping the system in order to find the cause of the error.

Command	Description
> OM1	Both outputs may be programmed to turn on and off using the O and IO commands.
> O1X	Turns on Output #1 and leaves Output #2 at itsprevious setting.

ON Amplifier On

Version Z5

Type Programming
Syntax <a>ON
Units None
Range None
Default None
Response None
See also OFF, ST1, STØ

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

This command turns the amplifier back on from the off state. If you issue the OFF command to shut the drive down, issuing the ON command re-enables the current to the motor, restoring motor torque. The ON command stops existing motion. This command is similar to the Shutdown (STØ) command. The ON command, however, is immediate and STØ is buffered.

Command	Description
> ON	Turns the motor on

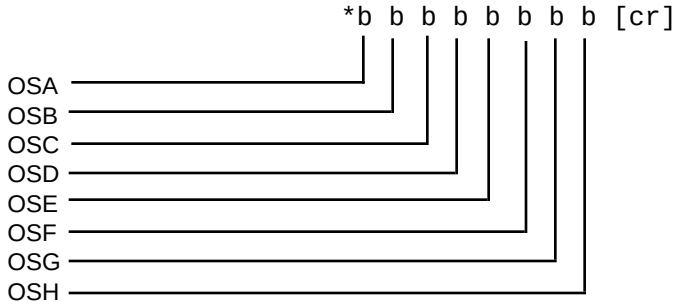
OS Function Setup Report

Version Z5

Type Status
Syntax aOS
Units None
Range None
Default None
Response *bbbb_bbbb
See also OSA, OSB, OSC, OSC, OSE, OSG, OSH

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

The OS command reports the status of the commands OSA, OSB, OSC, OSC, OSE, OSG, and OSH.



The following table briefly describes each of the OS commands.

Command	Description
OSA	Go home to resolver position
OSB	Back-up to home switch
OSC	Define active edge of home switch 1 = Active high signal
OSD	Not defined
OSE	Enable Jogging
OSF	Not defined
OSG	Define final home approach direction 1=CCW
OSH	Define active edge of home switch to stop on 1 = CCW
OSI	Save Sequence Scan mode on stop

OSA Go Home to Resolver Position

Version Z5

Type Set-Up
Syntax <a>OSAb
Units None
Range 0 or 1
Default 0
Response None
See also GH, GHP, OSA, OSB, OSC, OSE, OSG, OSH

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

The OSA command enables positioning to a resolver position at the end of a go home sequence.

OSAØ Do not go home to resolver position

OSA1 Go home to resolver position

The OSA1 command causes the motor to creep to the resolver position specified by the GHP command at the end of a go home move (GH). This command is used to produce a highly accurate, repeatable go home move. See the GHP command for more details. The OSA0 command causes, the motor to go home in accordance with settings of the OSB, OSG, and OSH commands.

OSB Backup to Home Switch

Version Z5

Type Set-Up
Syntax <a>OSBb
Units Mode
Range 0 or 1
Default 0
Response None
See also OSG, OSH

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

The OSB command affects the way the motor goes home in a Go Home sequence (GH) . It causes the motor to reverse direction and find the edge of the home region if set to 1, or stop as soon as the motor enters the home region if set to 0.

OSBØ Do not back up to home switch
OSB1 Back up to home switch

If this command is set to 0, the Compumotor Plus considers the motor at home if the home input is active, after encountering the active edge of home region.

If this command is set to 1, the Compumotor Plus decelerates the motor to a stop after encountering the active edge of the home region. The motor moves in the opposite direction of the initial go home move at the velocity set by the GF command until the active edge of the home region is encountered. The Compumotor Plus then considers the motor to be at home.

Refer to the GH command for more detailed information.