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

C H A P T E R ⑤

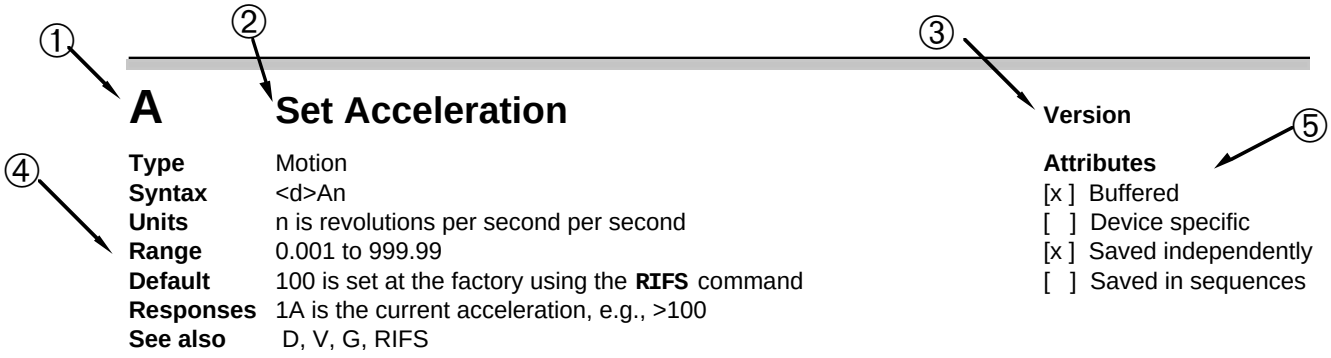
Software Reference

The information in this chapter will enable you to:

- ❑ Identify the four types of commands in Compumotor's X-Series Language
- ❑ Use this chapter as a reference for the function, range, default, and sample use of each command

Command Format Description

The following section describes the format of the command descriptions used in this chapter. The numbered arrows refer to the numbered sections below the drawing.



A **Set Acceleration**

Version

Type	Motion
Syntax	<d>An
Units	n is revolutions per second per second
Range	0.001 to 999.99
Default	100 is set at the factory using the **RIFS** command
Responses	1A is the current acceleration, e.g., >100
See also	D, V, G, RIFS

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

The acceleration command specifies the acceleration rate used for subsequent moves (G command). The acceleration remains set until you change it again. You do not need to reissue this command for subsequent Go (G) commands. Acceleration outside the valid range cause the acceleration to remain in previous valid acceleration setting. The Compumotor Plus uses the same value for deceleration.

The Compumotor Plus imposes limits on the maximum command acceleration for each of the Compumotor forcers.

Though there is no effective limit for the acceleration definition, values above 24,000 rps^2 will all appear to be instantanious.

Example	Description
A1ØØ	Set the acceleration rate
V1Ø	Set the velocity
D1ØØØØ	Set the move distance
G	Start the move

① Command Identifier

The letter or letters used to represent the command.

② Command name

This name used to refer to the command. For example, Acceleration for the A command.

③ Version

The revision of software in the Compumotor Plus when the described command was first introduced or last modified. If the revision level of the software you are using is equal to or greater than the revision level listed here, the command is available in your unit. You can determine the level of software in your Compumotor Plus by issuing the Revision Level (RV) command.

④ Characteristics

The following sections describe the main characteristics of the command.

Type

This portion of the box contains the command's type. The four command types are listed below.

Set-Up: These commands define Set-Up conditions for the application. Set-Up commands include the following types of commands:

- Homing (go home acceleration and velocity, etc.)
- Input/Output (limits, scan time, in-position time, etc.)
- Tuning (servo or position tracking)
- General (set switches, return to factory settings, etc.)

Programming: Programming commands affect programming and program flow. For example, trigger, output, all sequence commands, quote, time delays, pause and continue, enable and front-panel, loop and end loop, line feed, carriage return, and backspace.

Status: Status commands respond (report back) information.

Motion: Motion commands affect motor motion (for example, acceleration, velocity, distance, go home, stop, direction, mode, etc.)

Syntax

This field shows the syntax for the command. Compumotor Plus commands use the following generic syntax: acspd

Variable a This variable is the device address. If the address is optional it is shown in angle brackets: **<d>**. Only commands which require the Compumotor Plus to send a response require a device address. All commands may use a device address to designate which unit on a daisy chain the command is intended for.

Variable c This variable is the command identifier, which is one or more letters.

Variable s This variable represents a sign. A sign is not allowed for all commands. The **s** is not shown in the syntax if not allowed.

Variable p This variable represents the parameters the command requires. There may be zero or more parameters. If the number of parameters is zero **n** is not shown in the syntax

Variable d This variable is the end of command delimiter. This is always required and is not shown in the following descriptions for clarity. The delimiter may be a space character or a carriage return.

Units

This field describes what unit of measurement the parameter in the command syntax represents.

Range

This is the range of valid values that you can specify for **n** (or any other parameter specified).

Default

The default setting for the command is shown in this box. A command will perform its function with the default setting if you do not provide a value.

Response

The response to the command is shown in this box. Status commands report a condition in the indexer. Status commands do not affect the status they read.

Commands that set parameters report the parameters when the command is issued without a parameter. For example, **A1ØØ** sets the acceleration to 100 rps, but **1A** returns the current setting. ***Note****: To receive a response, a device address is required.*

See Also

Commands that are related or similar to the command described are listed here.

⑤ Attributes

Each command has attributes as shown below.

Attributes
[X] Buffered
[ ] Device specific
[X] Independently saved
[ ] Saved in sequences

Buffered

If the Buffered box is checked the command is buffered. If it is not checked the command is acted on immediately. Buffered commands are executed in the order they are received. An internal buffer, or storage area, holds the commands in a queue until the previous command has been executed.

Immediate commands are executed as they are received. Immediate commands are executed even if the command buffer has commands in it. For example, the Stop (**S**) command is immediate. When a Stop command is received the motor is stopped as soon as the command is received. The Compumotor Plus does not process the commands in its command buffer before stopping the motor.

Device specific

If the Device specific box is checked the command requires a device identifier. If it is not checked the command may be used with or without a device identifier. Commands which are device specific are normally Status commands. Device specific commands have a syntax description with a d by itself before the command. If it is not device specific the command syntax description has a **<d>** in angle brackets before the command.

Saved always

If the Independently saved box is checked the parameter controlled by the command is always saved. This differs from commands which may only be saved in sequences and those which are never saved. If neither the Saved always nor the Saved in sequences box is checked the command is never saved.

Saved in sequences

If the Saved in sequences box is checked the command will be saved only if it is in a sequence and you issue the Save command (**SV**). If neither the Saved always nor the Saved in sequences box is checked the command is never saved.

⑥ Description

A description of the command appears in this area along with any special considerations you should know about.

⑦ Example

An example of how to use the command appears in this area. The left column contains the commands you would issue to the Compumotor Plus. The right column contains descriptions of what the commands do in the program.

Alphabetical Command List

" Quote

Version Z5

Type Programming
Syntax a"x
Units Any printable character
Range Any printable ASCII character (maximum of 11 characters)
Default None
Response SAMPLE
See also None

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

Any characters entered after the quotation mark (") are 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 transmitted after the last character in the string. This command is used during buffered moves or sequences, or to command other Compumotor devices to move.

Command	Description
> MN	Set to Normal mode
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D12500	Set distance to 12,500 steps
> G	Execute the move (Go)
> "MOVE_DONE	After motor finished the move, the Compumotor Plus will send the message MOVE_DONE out from the RS-232C port

Command	Description
> MN	Set to Normal mode
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D12500	Set distance to 12,500 steps
> G	Execute the move (Go)
> "2XR1	Once the move is done, run sequence #1 is commanded on a unit with device address #2

Single Step

Version Z5

Type Programming
Syntax <a>#n
Units None
Range None
Default None
Response None
See also XST, XTR

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

This command controls the execution of a sequence when the Single Step Mode (XST) is enabled. Each time you enter the " command followed by a delimiter (carriage return or space), one command in the sequence buffer will be executed. You can run in Single Step mode only if you have RS-232C interface connected to a host. If you issue a Kill (K) command, while you are in Single Step mode, the sequence execution will be aborted, but the Single Step mode is retained. When you cycle power, the indexer will no longer be in Single Step mode.

Command	Description
> XE1	Erases sequence #1
> XD1	Defines sequence #1
> A5	Set acceleration to 5 rps^2
> V2	Set velocity to 2 rps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)
> XT	Ends sequence definition
> XST1	Enables Single Step mode
> XR1	Executes sequence #1
> XTR1	Enables trace mode
#	Execute a command
*SEQUENCE_001_COMMAND_A5	Reports executing the A5 command in sequence #1
#	Execute the next command
*SEQUENCE_001_COMMAND_V2	Reports executing the V2 command in sequence #1
#	Execute the next command
*SEQUENCE_001_COMMAND_D10000	Reports executing the D1000 command in sequence #1
#	Execute the next command
*SEQUENCE_001_COMMAND_G	Reports executing the G command in sequence #1 The motor moves 10,000 steps
#	Execute the next command
*SEQUENCE_001_COMMAND_XT	Reports executing the XT command in sequence 1

A Set Acceleration

Version Z5

Type Motion
Syntax <a>An
Units rps
Range 0.001 to 2147483.687
Default 10
Response current acceleration, e.g., 10
See also D, V, G

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

The acceleration command specifies the acceleration rate to be used upon executing the next Go (G) command. The acceleration remains set until you change it. You do not need to reissue this command for subsequent Go (G) commands. Accelerations outside the valid range cause the acceleration to remain in previous valid acceleration setting. The Compumotor Plus uses the same value for deceleration.

Though there is no effective limit for the acceleration definition, values above 24,000 rps^2 will all appear instantaneous.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D10000	Set distance to 10,000 steps
> G	Execute the move (Go)

B Buffer Status

Version Z5

Type Status
Syntax aB
Units None
Range None
Default None
Response *R or *B
See also BS

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

The buffer status command will report the status of the command buffer. If the command buffer is empty or less than 90% full, the controller will respond with a ***R**[cr].

The command buffer is 2,000 bytes long. A *B[cr] response will be issued if less than 10% of the command buffer is free.

*R = More than 10% of the buffer is free

*B = Less than 10% of the buffer is free

You may want to use this command when you load a long series of commands remotely. If the buffer size is exceeded, the extra commands will not be received by the system.

Command	Description
> 1B	*R (more than 10% of the Buffer is free)

BCCA Buffered Configure Current Average

Version Z5

Type Set-Up
Syntax aBCCA
Units Amps
Range 0 to 7.5 amps
Default None
Response *AVERAGE_CURRENT_LIMIT=7.5_AMPERES
See also BCCP, RSE, CCA

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

This command defines a new maximum average current. If the average current commanded exceeds the value defined by the BCCA command (while the motor is turned on), the controller disables the amplifier and indicates an error. This command is identical to the CCA command, except that this command is buffered and may be executed within a sequence, where the CCA command is an immediate command. It is useful when the average current must be changed within a sequence.

Command	Description
> XD1	Define sequence #1
> BCCA2.Ø	Set average current limit to 2.0 amps
> A1Ø	Set acceleration to 10 units/sec^2
> V1Ø	Set velocity to 10 units/sec
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> BCCA5.Ø	Set average current limit to 5.0 amps
> D1ØØØØ	Set distance to 10,000 steps
> G	Execute the move (Go)
> XT	End sequence definition

The average current limit is set to 2 amps before the first 5,000 step move and to 5.0 amps before the second 5,000 stop move.

BCCP Buffered Configure Current Peak

Version Z5

Type Set-Up
Syntax aBCCP
Units Amps
Range 0 to 8.5 amps
Default 8.5 amps
Response *PEAK_CURRENT_LIMIT=8.5_AMPERES
See also BCCA, CCA, RSE

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

This command defines the maximum peak current that can be sent to the motor. The value entered becomes the new peak current limit. If the current going to the motor exceeds the value defined by the BCCP command, the controller disables the amplifier and indicates an error. This command is identical to CCP except that it is buffered. It is a useful command when you want to change the peak current in a sequence.

Command	Description
> XD1	Define sequence #1
> A1Ø	Set acceleration to 10 rps^2
> BCCA5.Ø	Set average current limit to 5.0 amps
> V1Ø	Set velocity to 10 rps
> D25ØØØ	Set distance to 25,000 steps
> G	Execute the move (Go)
> 1BCCP7.Ø	Set peak current limit to 7.0 amps
> D1ØØØØ	Set distance to 10,000 steps
> G	Execute the move (Go)
> XT	End sequence definition

The peak current limit is set to five (5) amps before the first 10,000 step move and to eight (8) amps before the second 10,000 step move.

BCDB Buffered Configure Dead Band

Version Z5

Type	Set-Up
Syntax	aBCDB
Units	Steps
Range	0 to 167772160
Default	0
Response	Current setting, e.g., 0
See also	CDB, SSCI

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

The buffered configure dead band command defines the dead band value. If a parameter follows the command, the value will become the new dead band value in motor steps. Output #2 may be used to indicate when the absolute value of the following error is outside of the dead band region. When the following error exceeds the dead band, the in-position output will turn ON. When the following error is within the dead band region set with this command, the in-position output will turn OFF.

Command	Description
> A10	Set acceleration to 10 rps^2
> V5	Set velocity to 5 rps
> D50000	Set distance to 10,000 steps
> SSCI	Turn on enable output #2 when *In Position*
> BCDB50	Configure dead band to 50 steps

BCDG Buffered Configure Derivative Gain

Version Z5

Type	Set-Up
Syntax	aBCDG
Units	Percent of maximum value
Range	0 to 99
Default	30
Response	*DIFFERENTIAL_GAIN=30_PERCENT
See also	CDB, BCDB, BCDM, CDG

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

This command defines the derivative gain to be used for tuning. When you tune the drive using the derivative gain (BCDG) command, you are actually setting a percentage of the derivative gain maximum (CDM).

This command is identical to the CDG command except that it is buffered. It is useful when you must change the derivative gain in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D32000	Set distance to 32,000 steps
> BCDG44	Set the differential gain to 44% of the maximum value
> G	Execute the move (Go)
> BCDG20	Set the differential gain to 20% of the maximum value
> G	Execute the move (Go)

BCDM Buffered Configure Derivative Maximum

Version Z5

Type	Set-Up
Syntax	aBCDMn
Units	None
Range	0 to 32767
Default	400
Response	*DERIVATIVE_GAIN_MAXIMUM=400
See also	CDM, BCDG, BCIM, BCPM

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

This command defines the maximum derivative gain you may use for tuning purposes. If a valid number is entered, it will become the new maximum derivative gain. When you tune the drive using the Configure Derivative Gain (CDG), you will actually set a percentage of the maximum derivative gain. This command is identical to the Configure derivative Maximum (CDM) command except that it is buffered.

Command	Description
> A5	Set acceleration to 5 rps^2
> V10	Set velocity to 10 rps
> D32000	Set distance to 32,000 steps
> BCDM15000	Set derivative maximum gain to 15,000
> G	Execute the move (Go)
> BCDM5000	Set derivative maximum gain to 5,000
> G	Execute the move (Go)

BCIG Buffered Configure Integral Gain

Version Z5

Type Set-Up
Syntax aBCIGn
Units Percent
Range 0 to 99
Default 50
Response *INTEGRAL_GAIN=50_PERCENT
See also CIG, BCDG, BCIM

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

If a value is supplied, the specified value will become the new value for the integral gain. A new value for the integral gain is calculated using the entered value as a percent of maximum I value. The BCIG command differs from the Configure Integral Gain (CIG) command only in that it can be stored in a sequence.

This command defines the new integral gain to be used by the drive. If a valid number is entered the new integral gain is calculated using the following equation:

CIG * CIM ÷ 100 = Integral gain

A new value for the integral gain will be calculated using the entered value as a percent of maximum I value. The BCDM command differs from the CDM command only in that it is a buffered command and can be executed in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps^2
> D32ØØØ	Set distance to 32,000 steps
> G	Execute the move (Go)
> BCIG5Ø	Set integral gain to 50% of maximum value
> G	Execute the move (Go)

BCIL Buffered Configure Maximum Integral Sum Limit

Version Z5

Type Set-Up
Syntax aBCILn
Units None
Range 0 to 214748347
Default 65535
Response *INTEGRATOR_LIMIT=65536
See also CIL

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

This command defines the maximum integral sum limit. The value for the sum of the integral value cannot exceed the value set by the BCIL command. This command is useful if you want to move into a final position quickly. If you have a small BCIL value, the system will not allow the sum of the errors to exceed the n value resulting in the motor moving into position faster. If you make the BCIL value very large, the response of the final move will be slow. The Save (SV) command must be used to retain the value in nonvolatile memory.

Command	Description
> BCIL7ØØØØ	Set the maximum sum of the integral error to 70,000

BCIM Buffered Configure Integral Maximum

Version Z5

Type Set-Up
Syntax aBCIM
Units None
Range 0 to 32767
Default 9000
Response *INTEGRAL_GAIN_MAXIMUM=9000
See also CIM, CIG

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

This command is identical to the Configure Integral Maximum (CIM) command except that it is buffered.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps^2
> D32ØØØ	Set distance to 32,000 steps
> BCIM2ØØØ	Set maximum integral gain to 2000
> G	Execute the move (Go)
> BCIM5ØØØ	Set maximum integral gain to 5000
> G	Execute the move (Go)

BCPE Buffered Configure Position Error

Version Z5

Type	Set-Up
Syntax	aBCPEn
Units	Steps
Range	0 to 336544320
Default	5000
Response	*MAXIMUM_POSITION_ERROR=5000_STEPS
See also	CDE, CDB, CPE

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

The BCPE command specifies the value for the maximum position (or following) error. The BCPE command differs from the CPE command only in that it can be executed sequentially. If no parameter is specified, the BCPE command reports the current value for CPE.

This command is identical to Configure Position Error (CPE) command except that it is buffered. It is useful when the position error must be changed in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> BCPE4ØØØ	Set maximum following error to 4,000
> D32ØØØ	Set distance to 32,000
> G	Execute the move (Go)
> BCPE6ØØØ	Set maximum following error to 6,000
> G	Execute the move (Go)

BCPG Buffered Configure Position Gain

Version Z5

Type	Set-Up
Syntax	aBCPGn
Units	Percent
Range	0 to 99 percent
Default	15
Response	*PROPORTIONAL_GAIN=15_PERCENT
See also	CPG, CPM

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

This command is identical to the Configure Positional Gain (CPG) command except that it is buffered. It is useful when the proportional gain needs to be changed in a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D32ØØØ	Set distance to 32,000
> BCPG2Ø	Set proportional gain to 20
> G	Execute the move (Go)
> BCPG5Ø	Set proportional gain to 50
> G	Execute the move (Go)

BCPM Buffered Configure Positional Maximum

Version Z5

Type	Set-Up
Syntax	aBCPMn
Units	Steps
Range	0 to 32767
Default	100
Response	*PROPORTIONAL_GAIN_MAXIMUM=100
See also	CPG, CPM

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

This command causes the Compumotor Plus to calculate a new value for integral gain. It will be calculated using the entered value as a percent of maximum value. The BCPM command is a buffered version of the CPM command.

You may want to use this command when you must change the positional maximum within a sequence.

Command	Description
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D25ØØØ	Set distance to 25,000 steps
> BCPM5ØØ	Set maximum proportional gain to 500
> G	Execute the move (Go)
> BCPM6ØØ	Set maximum proportional gain to 600
> G	Execute the move (Go)

BCVG Buffered Configure Velocity Gain

Version Z5

Type	Set-Up
Syntax	aBCVG
Units	Percent of maximum
Range	0 to 99
Default	25
Response	*VELOCITY_GAIN=25_PERCENT
See also	None

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

The Buffered Configure Velocity Gain (BCVG) command is identical to the Configure Velocity Gain (CVG) command except that it is buffered. BCVG is useful when you need to change the velocity gain in a sequence.

The velocity gain is related to the error in the motor speed with respect to the velocity commanded by the PID control loop. If a valid parameter is entered, the velocity gain will be recalculated using the new percentage of the maximum.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D5ØØØ	Set distance to 5,000 steps
> BCVG5Ø	Set velocity gain to 50
> G	Execute the move (Go)
> BCVG1Ø	Set velocity gain to 10
> G	Execute the move (Go)

BCVM Buffered Configure Velocity Maximum

Version Z5

Type	Set-Up
Syntax	<a>BCVMn
Units	Velocity
Range	0 to 32767
Default	2000
Response	VELOCITY_GAIN_MAXIMUM=n
See also	CVG, CVM

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

This command defines the maximum velocity gain to be used for tuning. When you tune the drive using the Velocity Gain (CVG) command, you are actually setting a percentage of the Velocity Gain Maximum (CVM) command. This command is identical as Configure Velocity Maximum (CVM) except that it is buffered.

Changes made with this command will not be permanent (saved in non-volatile memory) until a Save (SV) command is issued.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 10 rps^2
> V1Ø	Set velocity to 10 rps
> D5ØØØ	Set distance to 5,000 steps
> BCVM5Ø	Set velocity maximum to 50
> G	Execute the move (Go)
> BCVM1Ø	Set velocity maximum to 10
> G	Execute the move (Go)

BS Buffer Space

Version Z5

Type	Status
Syntax	aBS
Units	Bytes (characters)
Range	None
Default	None
Response	*100
See also	B

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

The Buffer Space command reports the amount of space available in the command buffer. The report is in bytes (characters). When entering long string commands, check the buffer status to be sure it will not be overflowed by subsequent commands. All characters sent to the Compumotor Plus use space in the buffer including delimiters such as carriage returns and spaces.

Command	Description
> 1BS	Request buffer space available
> *1ØØ	Space for 100 characters is remaining in the command buffer

BSP Buffer Space

Version Z5

Type	Set-Up
Syntax	aBSP
Units	Steps
Range	±838860800
Default	0
Response	None
See also	SP, PI, MPA, PR, PZ, D

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

This command is a buffered version of the **SP** command, it allows you to set the buffered absolute counter (value n). The **BSP** command is useful for labeling a position value at a certain point. For example, you can set the zero reference point (home) to some location other than that of the physical hardware home.

C Continue

Version Z5

Type	Programming
Syntax	aC
Units	None
Range	None
Default	None
Response	None
See also	PS, U

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

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

Command	Description
> PS	Pause execution until the indexer receives a C command
> MC	Set to Continuous mode
> A5	Set acceleration to 5 rps^2
> V5	Set velocity to 5 rps
> G	Execute the move (Go)
> T1Ø	Wait 10 seconds after the move
> VØ	Set velocity to zero
> G	Decelerates the motor to zero velocity
> C	Start executing commands in buffer

CCA Configure Current Average

Version Z5

Type	Set-Up
Syntax	aCCAn
Units	Amperes
Range	0 to 7.5
Default	7.5
Response	*AVERAGE_CURRENT_LIMIT=n_AMPERES
See also	BCCA, CCP, RSE

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

The CCA command defines a new maximum average current. If the average current commanded exceeds the value defined by CCA command, while the motor is turned on, the controller will disable the amplifier and indicate an error.

Command	Description
> CCA5.0	Set average current to 5 amps

CCP Configure Maximum Current Peak

Version Z5

Type	Set-Up
Syntax	<a>CCPn
Units	Amperes
Range	0 to 8.5
Default	8.5
Response	*PEAK_CURRENT_LIMIT=n_AMPERES
See also	BCCP, CCA

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

This command defines the maximum peak current that will be sent to the motor. The value entered will become the new peak current limit. If the current going to the motor ever exceeds the value defined by CCP command, the controller will disable the amplifier and indicate an error.

Command	Description
> CCP4	Set peak current limit to 4 amps

CDB Configure Dead Band

Version Z5

Type	Set-Up
Syntax	<a>CDBn
Units	Steps
Range	0 to 167772160
Default	0
Response	*SLIP_FAULT_DEADBAND=n_STEPS
See also	BCDB, OM

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

The Configure Dead Band (CDB) command defines the new dead band value in motor steps. If dead band is exceeded the slip fault output to the indexer connector will be active. When the slip fault line is off it indicates that the absolute value of the following error is within the dead band region. This is useful when you need to know if the motor rotor is within a certain tolerance range with respect to the commanded position.

Command	Description
> 1CDB5Ø	Set the dead band to 50 steps
> 1CDB	Response *SLIP_FAULT_DEADBAND=50 STEPS

CDG Configure Differential Gain

Version Z5

Type	Set-Up
Syntax	<a>CDGn
Units	Percent of maximum (CDM)
Range	0 to 99
Default	30
Response	*DIFFERENTIAL_GAIN_MAXIMUM=n_PERCENT
See also	BCDG, CDM

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

This command defines the differential gain be used for tuning. When you tune the drive using the Differential Gain (CVG) command, you are setting a percentage of the Differential Gain Maximum (CDM) command.

Command	Description
> 1CDG3Ø	Set the differential gain to 30% of the maximum value

CDM Configure Differential Maximum

Version Z5

Type	Set-Up
Syntax	<a>CDMn
Units	None
Range	0 to 32,767
Default	400
Response	*DIFFERENTIAL_GAIN_MAXIMUM=n
See also	BCDM, CDG

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

This command defines the maximum differential gain you may use for tuning purposes. If you enter a valid number, it will become the new maximum differential gain. When you tune the drive using the Configure Derivative Gain (CDG) command, you will actually set a percentage of the maximum differential gain.

Command	Description
> CDM5ØØ	Set the maximum differential gain to 500
> CDG35	Set the differential gain to 35% of the maximum value

CGS Configure Gain Switching

Version Z5

Type Set-Up
Syntax <a>CGSn
Units Mode
Range 0 to 4
Default 1 and 4
Response *DERIVATIVE_SWITCHING_xx_TORQUE_SWITCHING_yy
See also CDG, CDM

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

This command defines and reports the current gain switching mode. Torque and derivative gain switching are affected by this command.

Derivative gain switching linearly scales the derivative gain value (defined by the CDM and CDG commands) down to zero as the motor speed drops from 2.0 rps to 0.0 rps.

Torque gain switching provides a high-resolution torque command to the amplifier when the torque command computed by the PIDV software falls below a certain threshold. The purpose of this mode is to provide high in-position stability.

CGS1	Derivative switching off
CGS2	Derivative switching on
CGS3	Torque switching off
CGS4	Torque switching on

The default setting is derivative (switching off) and torque (switching on).

CIG Configure Integral Gain

Version Z5

Type Set-Up
Syntax <a>CIGn
Units Percent of maximum (CIM)
Range 0 to 99
Default 50
Response *INTEGRAL_GAIN=n_PERCENT
See also BCIG, CIM

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

This command configures the integral gain to be used for tuning. When you tune the drive using Configure Integral Gain (CIG) command, you are actually setting a percentage of the Maximum Integral Gain (CIM) command.

Command	Description
> CIG70	Set the integral gain to 70% of the maximum integral gain

CIL Configure Integral Sum Maximum

Version Z5

Type Set-Up
Syntax <a>CIMn
Units Sum limit
Range 0 to 2147483647
Default 65535
Response *INTEGRATOR_LIMIT=n
See also BCIG, BCIL, BCIM, CIG, CIM, SV

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

This command defines the maximum integral sum limit. The value for the sum of the integral value is set by the CIL command. This command is useful if you want to move into a final position quickly. If you have a small CIL value, the system will not allow the sum of the errors to exceed the n value. This will result in the motor moving into position faster. If you make the CIL value very large, the response of final move will be slow. The Save (S) command must be used to retain the value in non-volatile memory.

Command	Description
> CIL70000	Set the maximum sum of the integral error to 70,000

CIM Configure Integral Maximum

Version Z5

Type	Set-Up
Syntax	<a>CIMn
Units	Max integral gain
Range	0 to 32,767
Default	9000
Response	*INTEGRAL_GAIN_MAXIMUM=n
See also	BCIM, CIG

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

This command defines the maximum integral gain you may use for tuning purposes. If you enter a valid number, it will become the new maximum integral gain. When you tune the drive using the Configure Integral Gain (CIG) command, you are actually setting the percentage of the maximum integral gain.

Command	Description
> CIM1000	Set the maximum integral gain to 1,000 steps
> CIG45	Set the integral gain to 45% of the maximum integral gain

CIP Configure In-Position Time

Version Z5

Type	Set-Up
Syntax	<a>CIPn
Units	ms
Range	1 to 32767
Default	2
Response	*IN_POSITION_WAIT=n
See also	CDB, IM

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

The CIP command specifies the time period that the motor must be within the dead band region before the in-position output is turned on. The n value is the number of 10ms periods to be used as the testing time frame.

The least amount of time the motor can be stopped and within the dead-band region before the motor is considered in-position is one 10ms time period. If at any point during that 10 ms the motor is out of the dead-band region, the in-position output will not turn on (the factory default setting is two 10ms periods, or 20 ms).

Note: Be sure to divide the number of milliseconds you want by 10.

The In-Position output provides a signal that can be used by a PLC to indicate when the motor has completed its move and settled into the user-specified position tolerance. The position tolerance is specified the the Configure Deadband (CDB) command.

Command	Description
> CIP5	In-position will be on if the motor positions its shaft within a dead-band region and stays in that position for at least 50 ms

CMR Configure Motor Resolution

Version Z5

Type	Set-Up
Syntax	<a>CMRn
Units	Steps per revolution
Range	200 to 25,600
Default	5000
Response	*n
See also	OFF, ON, ST1, ST0, SV

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

The Compumotor Plus motor actually functions at 12,800 steps per revolution (resolver resolution12,800). If you choose a multiple or submultiple of 12,800 as your motor resolution, you will not get any truncation error. Compumotor recommends that you use a multiple or submultiple of 12,800 as the motor resolution.

If you enter a CMR value of 5,000 steps per revolution, you would compute the scale factor as:

(12,800 • 65,536)/5,000 = 167,772.16

Since there is 16-bit precision, the 0.16 is truncated.

If you set move distance with D5ØØØØØ, the conversion from user-defined revs back to resolver revs is done as follows:

(500,000 • 167,772)÷65,536 = 1,279,998.779

Since you can't move to a fractional position, the motor actually moves to 1,279,998 counts of the resolver. You might expect the motor to go exactly 100 revolutions (500,000/5,000). For this move, the motor will actually go 1,279,998/12800 = 99.99984375 revolutions. This error does not accumulate, because if you give a second move of same distance, the calculation will use the absolute distance requested by the indexer to calculate the next move.

If truncation error is a problem you can choose a resolution that divides evenly into (12,800 • 65,536) or 2,147,483,648. For example, 4,096; 8,1912; 16,384; etc. Remember to save (SV) any changes you wish to retain before cycling power to the drive.

Note; The CMR command can only change motor resolution when the amplifier if off.

Command	Description
> OFF	Turn amplifier off
> CMR4Ø96	Define a motor resolution of 4,096 steps per revolution
> ON	Turn amplifier on
> SV	Save the new resolution to EEPROM

CMTR Configure Motor

Version Z5

Type Set-Up
Syntax <a>CMTR
Units None
Range 1, 2L, 2H or 3
Default None
Response None
See also None

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

The CMTR command sets a number of internal parameters appropriate for the motor connected to the drive. There are two types of CPX drives. The low-power CPX drive can run CP57-120 and CP83-150 motor sizes. The high-power CPX can drive CP83-150 and CP106-210 motors. The CP83-150 motor is the only motor that runs on both high- and low-power drives.

Note: Be sure to use the appropriate command parameter for the type of drive you have. Refer to the following table.

Low Power Drive (CPL)	High Power (CPH)
CMTR1 for the CP57-120 motor	CMTR2H for the CP83-150 motor
CMTR2L for the CP83-150 motor	CMTR3 for the CP106-220 motor

Command	Description
> 1CMTR1	Set the CPX Drive for the CP57-120 motor

CPB Configure Pushbutton

Version Z5

Type Set-Up
Syntax <a>CPBn
Units None
Range 0 or 1
Default 1
Response None
See also None

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

This command allows you to control user access to the front-panel pushbuttons. Since the PID & V tuning parameters can be modified via the front control panel, and changing these values can effect move time and final position accuracy, it may be desirable to disable this interface. The CPB1 command enables the front-panel pushbuttons. CPBØ disables the front panel.

Command	Description
> L	Loop infinite times
> A1Ø	Set acceleration to 10 rps^2
> V1	Set velocity to 1 unit/sec
> D5ØØØ	Set distance to 5,000 steps
> G	Execute the move (Go)
> CPB1	Enable pushbuttons
> TR1XX	Wait for Trigger input 1
> CPBØ	Disable pushbuttons
> N	End loop

CPE Configure Position Error

Version Z5

Type	Set-Up
Syntax	<a>CPEn
Units	Steps
Range	0 to 336544320
Default	5000
Response	None
See also	BCPE, RSE

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

The response to this command defines or reports the maximum following error. If the absolute position error is greater than this number, the amplifier will shut itself off and generate an error code #20 on the LED display. If a valid number in steps is entered, it will become the new maximum following error. Otherwise, the current setting is reported. Exceeding the maximum following error is an error condition that will cause the amplifier to be shutdown. If the maximum following error is defined as zero, the shutdown motor on following error exceeded function is disabled and no amount of following error will generate an error condition or shutdown the motor. The factory default setting is one revolution of the motor.

The value of the following error is only calculated when the CPE command is given. The stored number is in terms of motor revolution. Changes of the motor resolution will leave the following error actual distance unchanged unless a new CPE command is issued.

This command differs from the Configure Dead Band (CDB) command, since being outside of deadband region only affects the slip fault output. The CPE settings will shut off the drive. Save your settings to the EEPROM if you wish them to be permanent.

Command	Description
> 1CMR5000	Set motor resolution to 5,000 steps/rev
> 1CPE5000	Set position error to 1 revolution

CPG Configure Proportional Gain

Version Z5

Type	Set-Up
Syntax	<a>CPGn
Units	Percent
Range	0 to 99
Default	15
Response	*PROPORTIONAL_GAIN=n_PERCENT
See also	BCPG, CPM

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

This command defines the proportional gain to be used for tuning. When you tune the drive using the Configure Proportional Gain (CPG) command, you are actually setting a percentage of the Proportional Gain Maximum (CPM). This command is useful in tuning out the positional error during moves.

Command	Description
> 1CPG4	Configure the proportional gain to 4% of the maximum value

CPM Configure Proportional Maximum

Version Z5

Type	Set-Up
Syntax	<a>CPMn
Units	None
Range	0 to 32,767
Default	100
Response	*PROPORTIONAL_GAIN_MAXIMUM=n
See also	CPG

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

This command defines the maximum of the term that amplifies the position error. If a valid number is entered, it will become the new proportional gain maximum. Otherwise, the current setting is reported.

Command	Response
>1CPM	*PROPORTIONAL_GAIN_MAXIMUM=n

CR Carriage Return

Version Z5

Type	Programming
Syntax	<a>CR
Units	None
Range	None
Default	None
Response	[cr]
See also	LF, " (Quote)

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

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

Command	Description
> MPA	Set to Absolute Position mode
> A5Ø	Set acceleration to 50 rps^2
> V5	Set velocity to 5 rps
> D5ØØØ	Set distance to 5,000 steps
> G	Execute the move (Go)
> CR	Sends a carriage return

CVG Configure Velocity Gain

Version Z5

Type	Set-Up
Syntax	<a>CVGn
Units	Percent
Range	0 to 99
Default	25
Response	None
See also	BCVG

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

This command defines the velocity gain to be used for tuning. When you tune the drive using the Configure Velocity Gain (CVG) command, you are actually setting a percentage of the Velocity Gain Maximum (CVM) command. This command is useful in determining how well the motor/drive servos or maintains a commanded velocity during moves.

Command	Description
> 1CVG6Ø	Set the velocity gain to 60% of the maximum velocity gain

CVM Configure Velocity Maximum

Version Z5

Type	Set-Up
Syntax	<a>CVMn
Units	Steps
Range	0 to 32,767
Default	3000
Response	aCVM IS *VELOCITY_GAIN_MAXIMUM=n
See also	CVG, BCVM

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

This command defines the maximum velocity gain to be used for tuning. When you tune the drive using the Velocity Gain (CVG) command, you are actually setting a percentage of the Velocity Gain Maximum (CVM) command.

Command	Description
> 1CVM3ØØØ	Set velocity gain maximum to 3,000.steps
> 1CVG6Ø	Set velocity gain to 60% of the maximum value

D Distance

Version Z5

Type	Motion
Syntax	<a>Dn
Units	Steps
Range	-838860800 to +838860800
Default	5000
Response	None
See also	A, G, MN, MPA, MPI, V

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

The Distance (D) command specifies the end position for subsequent moves. The position is interpreted as incremental or absolute depending on the current positioning mode as set by the Motor Position Absolute (MPA) and Motor Position Incremental (MPI) commands.

In Incremental mode (MPI), the motor moves the distance specified by this command. The absolute position is changed by the same amount.

In Absolute mode (MPA), the motor moves to the absolute position specified by this command. If the current absolute position is the same as the specified position the motor does not move. It is possible for the motor to move in the negative direction for a positive move and vice versa. This happens when the absolute position specified is less than the current absolute position even though both positions may be positive.

The absolute position can be set to zero with a PZ or a Z (Reset) command.

Command	Description
> MN	Set to Normal mode
> A5	Set acceleration to 5 rps^2
> V1Ø	Set velocity to 10 rps
> D5ØØØØ	Set distance to 50,000 steps
> G	Execute the move (Go)

DCA Display Current Average

Version Z5

Type	Status
Syntax	<a>DCA
Units	None
Range	None
Default	None
Response	*AVERAGE_CURRENT=n_AMPERES
See also	DCI, DCP

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

The DCA command periodically reports the average current flowing through the motor in amperes. This information is reported and repeatedly updated until any character is sent to the Compumotor Plus. On a terminal, the character can be sent by pressing any key, such as the spacebar.

The Compumotor Plus calculates the average current by recording the instantaneous current several thousand times each second. It always keeps the previous 10 seconds of readings (about 60,000 readings). It then averages the readings every 10ms, and reports the average to the user upon receipt of the CDA command.

Command	Response
> 1DCA	*AVERAGE-CURRENT=n_AMPERES

DCI Display Current Instantaneous

Version Z5

Type	Status
Syntax	<a>DCI
Units	None
Range	None
Default	None
Response	aDCI = *CURRENT=n_AMPERES
See also	DCA, DCP

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

The DCI command periodically reports motor current on an instantaneous basis. This number is reported in amperes and is repeatedly updated until a key is pressed. This number is a single sampling of the current.

Command	Response
> 1DCI	*CURRENT=+2.83_AMPERES

DCP Display Current Peak

Version Z5

Type Status
Syntax <a>DCP
Units None
Range None
Default None
Response *CURRENT=n_AMPERES
See also DCI, DCA

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

The Display Current Peak command will periodically display/report the largest instantaneous current value commanded to the motor since the command was issued. This value is reported in amperes and is repeatedly updated until any character is sent to the controller. The controller samples the instantaneous current at 300 microsecond intervals. Each reading is compared to the largest previous reading. If the new value is larger, it will become the new value. This reading accumulates from the time the command was sent, so that the highest instantaneous current ever seen by the motor over long periods of time may be captured.

DFS Display Flags for Servo Parameters

Version Z5

Type Status
Syntax <a>DFS
Units None
Range None
Default None
Response *bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb
See also DFX

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

This command returns all the drives status flag as a 32 bit response. Each bit refers to a status flag as shown on the chart below. The associated number for each flag corresponds to each binary (1 or 0) response is in the following order: bit 31, bit 30...bit 0.

Bit	Function	Bit	Function
31	Reserved	15	Reserved
30	Reserved	14	Reserved
29	Reserved	13	Reserved
28	Reserved	12	Reserved
27	Reserved	11	enable circuit (enabled=0, disabled=1)
26	Reserved	10	Reserved
25	Reserved	9	Reserved
24	Reserved	8	Failed crc check (no = 0, yes = 1)
23	Reserved	7	Reserved
22	Reserved	6	Average current exceeded (no = 0, yes = 1)
21	Reserved	5	Max position error exceeded (no = 0, yes = 1)
20	Reserved	4	Reserved
19	Reserved	3	Driver internal error (no error=0, pwm hardware shutdown=1)
18	Reserved	2	Reserved
17	Reserved	1	Overcurrent (no=0, yes (shutdown error)=1)
16	Reserved	0	RS-232C CMD (on (ST0)=0, off (ST1)=1)

DFX Display Indexer Flags

Version Z5

Type Status
Syntax <a>DFX
Units None
Range None
Default None
Response *bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb_bbbb
See also DFS

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

The display Indexer Flags command returns all the indexers status flags as a 32 bit response. Each bit refers to a status flag as shown on the chart below. The associated number for each flag corresponds to each binary (1 or 0) response is in the following order: bit 31, bit 30...bit 0

Bit	Function	Bit	Function
31	Reserved	15	U command waiting for continue (no=0, yes=1)
30	Reserved	14	Waiting for a trigger (no=0, yes=1)
29	Reserved	13	Home to resolver position (no=0, yes=1)
28	Reserved	12	Back up to home limit (no=1, yes = 1)
27	Reserved	11	High speed portion of home move (not in progress=0, in progress=1)
26	Reserved	10	Execute a sequence (no=0, yes=1)
25	Reserved	9	Waiting on a timer (no=0, yes=1)
24	Reserved	8	Hit a CCW limit (no=0, yes=1)
23	Homing to resolver position (no=0, yes=1)	7	Hit a CW limit (no=0, yes=1)
22	Hit a soft CW limit (no=0, yes=1)	6	PS command waiting for continue (no=0, yes=1)
21	Hit a soft CCW limit (no=0, yes=1)	5	Absolute move direction (positive=0, negative=1)
20	Registration move in progress (no=0, yes=1)	4	Positioning mode (incremental/absolute) (MPI=0, MPA=1)
19	Home limit (not found=0,found=1)	3	Mode (preset/continuous) (MN=0, MC=1)
18	Jog (disable=0, enable=1)	2	Commanded move direction (positive=0, negative=1)
17	Queued for RM mode (no = 0, yes = 1)	1	Preset move in progress (not moving=0, moving=0)
16	Run sequence on power up (no = 0, yes=1)	0	Continuous move in progress (not moving=0, moving=0)

DPA Display Position Actual

Version Z5

Type Status
Syntax <a>DPA
Units None
Range None
Default None
Response ±nnnnnnnn where n is a digit from zero to nine.
See also DPE

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

The Display Position Actual (DPA) command periodically displays the actual absolute position in motor steps. The function is canceled by sending any character to the controller. The value displayed is a cumulative count based on the shaft position when the drive was enabled and is scaled by the Configure Motor Resolution (CMR) command.

Resulting decimals are truncated. This can cause a small non-cumulative error in position if the CMR resolution does not evenly divide into 12,800

DPE Display/Report Position Error

Version Z5

Type Status
Syntax <a>DPE
Units None
Range None
Default None
Response ±nnnnnnnn where n is a digit from zero to nine
See also None

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

This command reports the difference between set point and actual position in steps. The position control algorithm uses this number to determine what sort of current should be sent to the motor. The difference between the command set point and the actual position is also used to determine if the motor is within the dead band specified in the Configure Dead Band (CDB) command. This number is reported in motor steps and is repeatedly updated until a key is pressed.

DPR Display/Report Position Resolver

Version Z5

Type Status
Syntax <a>DPR
Units None
Range None
Default None
Response None
See also None

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

This command continuously reports the resolver position in motor steps within a single pole of the motor shaft. This position is reported in steps and is repeatedly updated until a key is pressed. The Compumotor Plus motor has 50 poles, and the resolver built onto the motor is absolute within one motor pole. The resolver divides each pole into 256 steps.

The drive has a natural resolution of 12,800 steps/rev (256 steps/pole * 50 poles/rev). The report from the DPR command can be useful in diagnosing resolver cable problems.

DPS Display/Report Position Set Point

Version Z5

Type Status
Syntax <a>DPS
Units None
Range None
Default None
Response None
See also None

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

This command reports the absolute number of pulses sent to the drive from the indexer since the drive was enabled (or reset). This number is reported in motor steps and is repeatedly updated until a key is pressed.

DTP Display Tuning Parameters

Version Z5

Type Status
Syntax <a>DTP
Units None
Range None
Default None
Response
```
*---------P-----I-----D-----V
*PERCENT__20____50____50____60
*MAXIMUM__100___9000__400___2000
*INTEGRATOR_LIMIT=10000
*MINIMUM_POSITION_ERROR=0_STEP
*MOTOR_TYPE=CP1
*MOTOR_RESOLUTION=5000_STEP/REV
*LIMIT_DISABLED
```
See also None

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

This command displays all pertinent tuning parameters at once. This is useful when tuning several parameters at once.

DVA Display/Report Actual Velocity

Version Z5

Type Status
Syntax <a>DVAn
Units None
Range None
Default None
Response None
See also None

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

This command reports the velocity of the motor in rpm. The instantaneous velocity is stored every millisecond. The average of the readings is reported every 128ms. Velocity is reported until a key is pressed.

DVS Display/Report Velocity Setpoint

Version Z5

Type Status
Syntax <a>DVSn
Units None
Range None
Default None
Response *VELOCITY_SETPOINT=n
See also None

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

The DVS command continuously reports the current target velocity. The velocity is reported in steps per second and is repeatedly updated until a key is pressed. The target velocity is the velocity used in the velocity portion of the servo loop by the PID algorithm. For more information on PID refer *Chapter 4, Application Design* section.

Command	Description
> 1DVS	Velocity setpoint is reported repeatedly until a number key is pressed