C H A P T E R ⑥

Software Reference

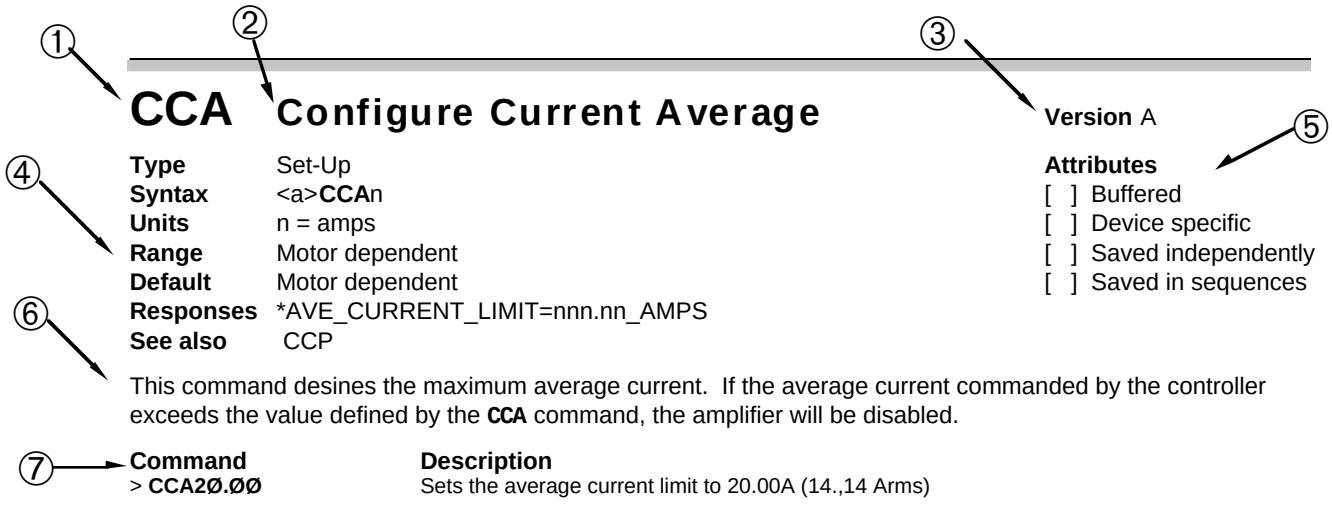
The information in this chapter will enable you to:

- ❏ Understand the Z Drive's command language
- ❏ Use this chapter as a reference for the function, range, default, and sample use of each command

The Z Drive has more than 40 software commands that you can use to tune the drive, define various parameters, and display drive variables. All commands are sent to the drive over the RS-232C serial interface. Any terminal or computer that has an RS-232C port can communicate with the Z Drive. *Chapter ③ Installation* explains the configuration of the RS-232C port. *Chapter ⑧ Maintenance & Troubleshooting*, offers guidelines to help you resolve communications problems.

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.



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 Z Drive 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 Z Drive 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. Z Drive 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 Z Drive 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 Z Drive 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 Z Drive. The right column contains descriptions of what the commands do in the program.

Command List

Command	*Tuning Commands*
CPD	Configure Position Derivative
CPDM	Configure Position Derivative Maximum
CPI	Configure Position Integral
CPIM	Configure Position Integral Maximum
CPP	Configure Position Proportional
CPPM	Configure Position Proportional Maximum
CTC	Configure Time Constant
CTG	Configure Tach Gain
CTGM	Configure Tach Gain Maximum
CVF	Configure Velocity Feed-Forward
CVFM	Configure Velocity Feed-Forward Maximum
CVI	Configure Velocity Integral
CVIM	Configure Velocity Integral Maximum
CVP	Configure Velocity Proportional
CVPM	Configure Velocity Proportional Maximum

Command	*Drive Commands*
CCA	Configure Current Average
CCP	Configure Current Peak
CDB	Configure Deadband
CIP	Configure In Position
CMR	Configure Motor Resolution
CMV	Configure Max. Velocity (Torque Mode)
CMTR	Configure Motor Type
CPE	Configure Position Error
CRR	Configure Resolver Resolution
CVE	Configure Velocity Error

Command	*Display Commands*
DCA	Display Current Average
DCI	Display Current Instantaneous
DCP	Display Current Peak
DDI	Display Drive Information
DPA	Display Position Actual
DPE	Display Position Error
DPR	Display Position Resolver
DPS	Display Position Setpoint
DSP	Display Servo Picture
DVA	Display Velocity Actual
DVE	Display Velocity Error
DVS	Display Velocity Setpoint

Command	*Miscellaneous Commands*
CPB	Configure Pushbuttons
CZM	Configure Z Drive Mode
DISP'	Front Panel Display
E	Enable Communications Interface
F	Disable Communications Interface
FMCA	Find Motor Commutation Angle
^H	Backspace
HELP	Help
OFF	Amplifier Off
ON	Amplifier On
OUT	Real Time Output Control
RFS	Return to Factory Settings
RSE	Report Servo Errors
RV	Revision Level
SS	Set-up Status Report
SSA	RS-232C Echo Control
SSI	Interactive Mode Control
SSM	Direction Input Polarity Control
SSR	Analog Output Control
SSU	Set Warning Display Mode
SV	Save
Z	Reset

Command	*Analog Voltage Commands*
ANDB	Analog Voltage Deadband
ANG	Analog Voltage Gain
ANV	Analog Voltage Range
ANZ	Analog Voltage Zero

ANDB Analog Voltage Deadband

Version A

Type Set-Up
Syntax <a>**ANDB**n
Units n = volts
Range 0.00 - 1.00
Default 0.00
Response *ANALOG_DEADBAND=nnn.nn_VOLTS
See also ANV, ANZ

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

ANDB defines the maximum analog voltage deadband that the Z Drive will allow without commanding torque to the motor. This voltage should normally be zero to allow for the greatest command accuracy. If your application requires zero velocity drift, a deadband may be required.

Command	Description
> ANDBØ.1Ø	Sets the analog deadband to 100 mV

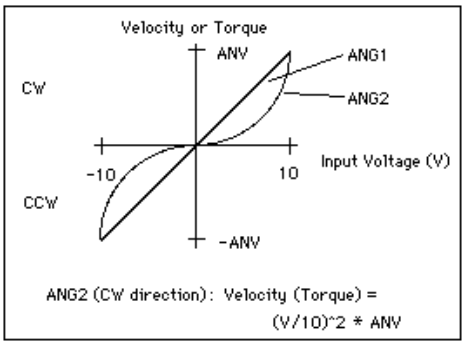
ANG Analog Voltage Gain

Version C

Type Set-Up
Syntax <a>**ANG**n
Units n = mode
Range 1 or 2
Default 1
1
Response *ANALOG_GAIN=1
See also ANV

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

This command offers two different analog input voltage to desired velocity (or torque) relationships. **ANG1** use a linear function (see **ANV** command). **ANG2** uses a parabolic function (refer to the following figure).



ANG Graphical Relationship

ANV Analog Voltage Range

Version A

Type Set-Up
Syntax <a>**ANV**n
Units n = rpm or amp
Range Motor dependent
Default Motor dependent
Response *10V_ANALOG=nnnnn_RPM
*10V_ANALOG=nn.nn_AMPS
See also ANDB, ANZ, CZM, ANG

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

This command defines the ratio between the input voltage range and the commanded velocity or torque. Once the **ANV** value is specified, the desired speed or torque is calculated as follows (assuming **ANG** = 1):

$$\text{Desired speed or torque} = \left(\frac{\text{Input voltage}}{10\text{V}}\right) * \textbf{ANV Setting}$$

First, verify the drive's the operational mode. Use the Configure Z Drive Mode (**CZM**) command before using **ANV**. While in Velocity mode (**CZM3**) the **ANV** default is set so that 10V equals the rated speed of the motor. While in Torque mode (**CZM2**), the **ANV** default is set so that 10V equals the rated continuous torque of the motor. The value can be set as high as the peak allowable torque. If an application requires greater analog input resolution for low speeds or low torques, use **ANV** to set 10V equal to a lower speed or torque value.

Command	Description
> CZM3	Sets the Z Drive to velocity mode
> ANV1ØØØ	Sets 10V analog input equal to 1,000 rpm

ANZ Analog Voltage Zero

Version A

Type Set-Up
Syntax <a>**ANZ**
Units None
Range None
Default None
Response None
See also ANV, ANDB

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

This command executes the Analog Auto Zero feature. Execute this command to set the current voltage level to represent 0V. This command is used to remove the DC offset out of the commanded velocity (or torque). If this command is not used, a small offset in the servo controller will cause the motor to creep.

Command	Description
> ANZ	Defines the analog input value as 0V

CCA Configure Current Average

Version A

Type Set-Up
Syntax <a>**CCA**n
Units n = amperes
Range Motor and Drive dependent
Default Motor dependent
Response *AVE_CURRENT_LIMIT=nnn.nn_AMPS
See also CCP

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

This command defines the maximum average current. If the average current commanded by the controller exceeds the value defined by **CCA** command, the amplifier will be disabled.

Command	Description
> CCA20.00	Sets the average current limit to 20.00A (14.14 Arms)

CCP Configure Current Peak

Version A

Type Set-Up
Syntax <a>**CCP**n
Units n = amperes
Range Motor and Drive dependent
Default Motor dependent
Response *PEAK_CURRENT_LIMIT=nnn.nn_AMPS
See also CCA

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

This command defines the maximum peak current that will be sent to the motor. The commanded current will never exceed the value defined by the **CCP** command.

Command	Description
> CCP40.00	Sets the peak current limit to 40.00A (28.28 Arms)

CDB Configure Deadband

Version A

Type Set-Up
Syntax <a>**CDB**n
Units n = steps
Range 0 - (1048575)
Default 0
Response *SLIP_FAULT_DEADBAND=nnnnnnn_STEPS
See also CIP, CMR, CPE

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

The **CDB** command defines the deadband in steps. The deadband is the range of input signals for which there is no system response. If the absolute value of the position error exceeds the deadband, the in-position output on the Indexer Connector will be active. When the In-Position line is off, the absolute value of the position error is within the deadband region (for the minimum time specified by the Configure In-Position (**CIP**) command. This command is scaled by the Configure Motor Resolution (**CMR**) command (steps are in the user's resolution). Set the desired **CMR** value before setting the **CDB** value.

CDB differs from the Configure Position Error (**CPE**) command. When the deadband is exceeded, the In-Position signal on the indexer connector will be activated, but the drive will continue to strive for zero position error. When the position error limit is exceeded, the drive will fault.

Command	Description
> CDB50	Sets the deadband to 50 steps

CIP Configure In-Position

Version A

Type Set-Up
Syntax <a>**CIP**n
Units n = 5 ms increments
Range 0 - 255
Default 0
Response *IN_POSITION_TIME=nnn_(*5_milliSeconds)
See also CDB

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

The **CIP** command defines the minimum time for the motor to be declared *in-position*. This command is used in conjunction with the Configure Deadband (**CDB**) command. The deadband is the range of input signals for which there is no system response. The **CIP** command specifies the amount of time required within the deadband to activate the in-position output. When the in-position line is off, the absolute value of the position error has been within the deadband region for the minimum time specified by the **CIP** command.

Command	Description
> CIP1Ø	Sets the in-position time to 10*5ms = 50 ms

CMR Configure Motor Resolution

Version A

Type Set-Up
Syntax <a>**CMR**n
Units n = steps
Range 200-32768 (Any #), 32768-65536 (even #'s only)
Default 5000
Response *MOTOR_RESOLUTION_(STEPS)=nnnn
See also OFF, ON

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

The **CMR** command defines the resolution of the motor in steps per revolution. This resolution will be used when interpreting the step input from the indexer.

This command changes the values and responses of the **CPE**, **CDB**, **DPA**, and **DPE** commands. You should configure motor resolution before you use any of these commands. Before you define the resolution, you must issue the **OFF** command to shut down the drive. After the **CMR** command is executed, you must issue the **ON** command to re-enable the drive.

Changing motor resolution does not affect the system's performance. The controller converts the commanded position to the proper absolute resolver position.

Command	Description
> OFF	Turns amplifier off
> CMR4Ø96	Defines motor resolution of 4,096 steps/rev
> ON	Turns amplifier on

CMTR Configure Motor

Version A

Type Set-Up
Syntax <a>**CMTR**n
Units n = motor type
Range 605, 606, 610, 620, 630, 635, 640, 805, 806, 810, 820, 830, 840,910, 920, 930, 940
Default 605
Response *MOTOR_TYPE=Znnn
See also None

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

Use this command to configure the drive to the motor size that is being used. The drive is configured at the factory, but if you change motor size, you must reconfigure the motor type.

This command selects the proper current values and factory defaults for the various motor sizes that may be use with the drive. Issuing the **CMTR** command will return the following set-up parameters to their default values: **ANV**, **CCA**, **CCP**, **CMV**, **CPD**, **CPDM**, **CPI**, **CPIM**, **CPP**, **CPPM**, **CRR**, **CTC**, **CTG**, **CTGM**, **CVE**, **CVF**, **CVFM**, **CVI**, **CVIM**, **CVP**, **CVPM**, and motor commutation data. **CMTR** automatically saves these parameters.

Command	Description
> CMTR62Ø	Configures drive for default settings for the Z620 motor

CMV Configure Maximum Velocity

Version A

Type Set-Up
Syntax <a>**CMV**n
Units n = rpm
Range 0 - 7000
Default Motor dependent
Response *MAX_VELOCITY_(TORQUE_MODE_ONLY)=nnnn_RPM
See also CZM

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

Use **CMV** only when operating in Torque mode (**CZM2**). **CMV** limits maximum motor speed in Torque mode.

Command	Description
> CMV32	Configures drive for maximum velocity of 32 rpm when in Torque mode

CPB Configure Pushbuttons

Version A

Type Set-=Up
Syntax <a>**CPB**n
Units n = pushbuttons
Range 0 = disabled, 1 = enabled, 2 = chgs disabled
Default 1
Response *CPB=2...BUTTON_CHANGES_DISABLED
*CPB=1...BUTTONS_ENABLED
*CPB=0...BUTTONS_DISABLED
See also None

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

This command controls user access to the front panel pushbuttons. Since many drive parameters can be modified via the front panel, it may be desirable to disable this interface.

- ❑ **CPBØ** = All pushbutton functionality is disabled.
- ❑ **CPB1** = Pushbuttons are enabled.
- ❑ **CPB2** = Pushbuttons are partially enabled—display features are enabled. Parameter/drive modifications are disabled.

Command	Description
> CPBØ	Pushbuttons are disabled

CPD Configure Position Derivative

Version A

Type Set-Up
Syntax <a>**CPD**n
Units n = % of maximum derivative gain
Range 0 - 99
Default Motor dependent
Response *POSITION_DERIVATIVE_PERCENT=nn
See also CPDM (Refer to *Chapter ④ Application Design* for tuning information.)

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

CPD defines the derivative gain for the position control loop. When you tune the drive using **CPD**, a percentage of the Configure Position Derivative Maximum (**CPDM**) is set. **CPD** overrides the Configure Velocity Feed-Forward (**CVF**) and Configure Tach Gain (**CTG**) commands.

Command	Description
> CPD15	Sets position derivative gain to 15% of the maximum value

CPDM Configure Position Derivative Maximum

Version A

Type Set-Up
Syntax <a>**CPDM**n
Units n = maximum derivative gain
Range 0 - 32767
Default Motor dependent
Response *POSITION_DERIVATIVE_MAXIMUM=nnnn
See also CPD

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

CPDM defines the maximum derivative gain for the position control loop. When tuning the drive, set **CPDM** and adjust the overall derivative gain parameter by increasing or decreasing the **CPD** value. **CPDM** is configured at the factory and typically does not need to be adjusted. (Refer to Display Drive Information [**DDI**] example.)

Command	Description
> CPDM1ØØØ	Sets maximum derivative gain to 1,000
> CPD15	Sets position derivative gain to 15% of 1,000

CPE Configure Position Error

Version A

Type Set-Up
Syntax <a>**CPE**n
Units n = steps
Range 0 - 1048575
Default 5000
Response *POSITION_ERROR_LIMIT=nnnnnn_STEPS
See also CMR

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

This command specifies a value for the maximum position error. If the position error exceeds the **CPE** value, the controller will disable the amplifier. **CPE** is scaled by the Configure Motor Resolution (**CMR**) command (steps are in the user's resolution). Set the desired **CMR** value before setting the **CPE** value. **CPEØ** will disable the position error fault.

CPE differs from the Configure Deadband (**CDB**) command. When the deadband is exceeded, the in-position signal on the indexer connector will be activated, but the drive will continue to strive for zero position error. When the position error limit is exceeded, the drive will fault.

Command	Description
> CPE1000	Sets the maximum position error to 1,000 motor steps

CPI Configure Position Integral

Version A

Type Set-Up
Syntax <a>**CPI**n
Units n = % of maximum integral gain
Range 0 - 99
Default Motor dependent
Response *POSITION_INTEGRAL_PERCENT=nn
See also CPIM

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

This command defines the integral gain for the position control loop. When you tune the drive using the **CPI** command, you are setting a percentage of the Configure Position Integral Maximum (**CPIM**).

Command	Description
> CPI5	Sets position integral gain to 5% of the maximum value

CPIM Configure Position Integral Maximum

Version A

Type Set-Up
Syntax <a>**CPIM**n
Units n = maximum integral gain
Range 0 - 32767
Default Motor dependent
Response *POSITION_INTEGRAL_PERCENT=nn
See also CPI

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

This command defines the maximum integral gain for the position control loop. When tuning the drive, first set **CPIM** and then adjust the overall derivative gain parameter by increasing or decreasing **CPI**. **CPIM** is configured at the factory and typically does not need to be adjusted. Refer to the Display Drive Information (**DDI**) command example.

Command	Description
> CPIM500	Sets maximum integral gain to 500
> CPI5	Sets position derivative gain to 5% of 500

CPP Configure Position Proportional

Version A

Type Set-Up
Syntax <a>**CPP**n
Units n = % of maximum proportional gain
Range 0 - 99
Default Motor dependent
Response *POSITION_PROPORTIONAL_PERCENT=nn
See also CPPM

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

This command defines the proportional gain for the position control loop. When you tune the drive using the **CPP** command, you are setting a percentage of the Configure Position Proportional Maximum (**CPPM**).

Command	Description
> CPP35	Sets position proportional gain to 35% of the maximum value

CPPM Configure Position Proportional Maximum

Version A

Type Set-Up
Syntax <a>**CPPM**n
Units n = maximum proportional gain
Range 0 - 32767
Default Motor dependent
Response *POSITION_PROPORTIONAL MAXIMUM=nnnn
See also CPP

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

This command defines the maximum proportional gain for the position control loop. When you tune the drive, you must first set **CPPM** and then adjust the overall derivative gain parameter by increasing or decreasing the **CPP**. **CPPM** is configured at the factory and typically does not need to be adjusted. Refer to the Display Drive Information (**DDI**) command example.

Command	Description
> CPPM25ØØ	Sets maximum to 2,500
> CPP35	Sets position proportional gain to 35% of 2,500

CRR Configure Resolver Resolution

Version A

Type Set-Up
Syntax <a>**CRR**n
Units n = bits or auto (A)
Range 12, 14, 16, A
Default A for position or torque mode, n for Velocity mode
Response *RESOLVER_RESOLUTION=AUTO
*RESOLVER_RESOLUTION=nn_BITS...[nnnn_RPM_MAX]
See also DPR

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

The **CRR** command defines the resolution of the resolver. The resolver provides information required by the controller to commutate the motor and perform its servo loop. This number is expressed in bits. Higher resolutions are desirable, but limit the speed at which the resolver to digital converter can operate. The following is the maximum speed for each resolution setting:

- 12-bits = 4,096 counts/rev 7000 rpm max
- 14-bits = 16,384 counts/rev 3800 rpm max
- 16-bits = 65,536 counts/rev 900 rpm max

If the velocity commanded by the indexer exceeds these maximum values, the amplifier will be disabled.

The **CRRA** command selects automatic resolution switching. In this mode, the Z Drive chooses the highest resolution, based on the present motor speed. This is desirable for most applications. Do not use the **CRRA** command if you are using the pseudo-quadrature outputs (refer to *Chapter ④ Application Design*).

Command	Description
> CRR12	Sets resolver resolution to 4,096 counts/rev

CTC Configure Time Constant

Version A

Type Set-Up
Syntax <a>**CTC**n
Units n = µSec
Range 0 - 30000
Default Position mode = 5, Velocity mode = 5 Torque mode = 20
Response *TIME_CONSTANT_=nnnn_(*100_microSeconds)
See also None

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

This command defines the filter time constant of the servo loop in increments of 100 µs periods, where 100 µs is the servo update rate. The **CTC** command controls the -3dB frequency of the torque command's low-pass filter.

Command	Description
> CTC2	Sets time constant to 200 µs (0.2 ms).

CTG Configure Tach Gain

Version A

Type Set-Up
Syntax <a>**CTG**n
Units % of torque gain
Range 0 - 99
Default Motor dependent
Response *TACH_GAIN_PERCENT=nn
See also CTGM

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

This command defines the tach gain of the servo loop. When you tune the drive using the **CTG** command, you are setting a percentage of the Configure Tach Gain Maximum (**CTGM**).

Command	Description
> CTG33	Sets tach gain to 33% of the maximum value

CTGM Configure Tach Gain Maximum

Version A

Type Set-Up
Syntax <a>**CTGM**n
Units n = maximum torque gain
Range 0 - 32767
Default Motor dependent
Response *TACH_GAIN_MAXIMUM=nnnnn
See also CTG

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

This command defines the maximum tach gain for the servo loop. When you tune the drive, you must first set **CTGM** and then adjust the overall derivative gain parameter by increasing or decreasing the **CTG** value. **CTGM** is configured at the factory and typically does not need to be adjusted. Refer to Display Drive Information (**DDI**) command example.

Command	Description
> CTGM2000	Sets maximum tach gain to 2,000
> CTG33	Sets tach gain to 33% of 2,000

CVE Configure Velocity Error

Version A

Type Set-Up
Syntax <a>**CVE**n
Units n = rpm
Range 0 - 7000
Default Motor dependent
Response *VELOCITY_ERROR_LIMIT=nnnn_RPM
See also None

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

This command specifies a value for the maximum velocity error. If the absolute value of velocity error exceeds the **CVE** value, the amplifier will be disabled. This value represents the absolute value of velocity error. **CVE0** will disable the velocity error fault.

Command	Description
> CVE100	Sets the maximum velocity error to +/- 100 rpm

CVF Configure Velocity Feed-Forward

Version A

Type Set-Up
Syntax <a>**CVF**n
Units % of maximum velocity feed-forward
Range 0 - 99
Default Motor dependent
Response *VELOCITY_FEED-FORWARD_PERCENT=nn
See also CVFM

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

This command defines the velocity gain for the velocity control loop. When you tune the drive using the **CVF** command, you are setting a percentage of the Configure Velocity Feed-Forward Maximum (**CVFM**).

Command	Description
> CVF15	Sets velocity feed-forward gain to 15% of the maximum value

CVFM Configure Velocity Feed-Forward Maximum

Version A

Type	Set-Up
Syntax	<a>**CVFM**n
Units	n = maximum velocity feed-forward
Range	0 - 32767
Default	Motor dependent
Response	*VELOCITY_FEED-FORWARD_MAXIMUM=nnnn
See also	CVF

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

This command defines the maximum velocity gain for the velocity control loop. When you tune the drive, you must first set **CVFM** and then adjust the overall derivative gain parameter by increasing or decreasing the **CVF**. **CVFM** is configured at the factory and typically does not need to be adjusted. Refer to the Display Drive Information (**DDI**) command example.

Command	Description
> CVFM1000	Sets the maximum velocity feed-forward gain to 1,000
> CVF15	Sets velocity feed-forward gain to 15% of 1,000

CVI Configure Velocity Integral

Version A

Type	Set-Up
Syntax	<a>**CVI**n
Units	% of maximum velocity integral gain
Range	0 - 99
Default	Motor dependent
Response	*POSITION_INTEGRAL_PERCENT=nn
See also	CVIM

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

This command defines the integral gain for the velocity control loop. When you tune the drive using the **CVI** command, you are setting a percentage of the Configure Velocity Integral Maximum (**CVIM**).

Command	Description
> CVI5	Set velocity integral gain to 5% of the maximum value

CIM Configure Velocity Integral Maximum

Version A

Type	Set-Up
Syntax	<a>**CVIM**n
Units	n = maximum integral gain
Range	0 - 32767
Default	Motor Dependent
Response	*POSITION_INTEGRAL_MAXIMUM=nnnn
See also	CVI

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

This command defines the maximum integral gain for the velocity control loop. When you tune the drive, you must first set **CVIM** and then adjust the overall derivative gain parameter by increasing or decreasing the **CVI**. **CVIM** is configured at the factory and typically does not need to be adjusted. Refer to the Display Drive Information (**DDI**) command example.

Command	Description
> CVIM500	Sets maximum integral gain to 500
> CVI5	Sets velocity integral gain to 5% of 500

CVP Configure Velocity Proportional

Version A

Type	Set-Up
Syntax	<a>**CVP**n
Units	n = % of maximum velocity proportional gain
Range	0 - 99
Default	Motor dependent
Response	*POSITION_PROPORTIONAL_PERCENT=nn
See also	CVPM

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

This command defines the proportional gain for the velocity control loop. When you tune the drive using the **CVP** command, you are setting a percentage of the Configure Velocity Proportional Maximum (**CVPM**).

Command	Description
> CVP35	Sets velocity proportional gain to 35% of the maximum value

CVPM Configure Velocity Proportional Maximum

Version A

Type	Set-Up
Syntax	<a>**CVPM**n
Units	n = maximum velocity proportional gain
Range	0 - 32767
Default	Motor dependent
Response	*POSITION_PROPORTIONAL_MAXIMUM=nnnnn
See also	CVP

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

This command defines the maximum proportional gain for the velocity control loop. When you tune the drive, you must first set **CVPM** and then adjust the overall derivative gain parameter by increasing or decreasing the **CVP**. **CVPM** is configured at the factory and typically does not need to be adjusted. Refer to the Display Drive Information (**DDI**) command example.

Command	Description
> CVPM25ØØ	Sets maximum to 2,500
> CVP35	Sets velocity proportional gain to 35% of 2,500

CZM Configure Z Drive Mode

Version A

Type	Set-Up
Syntax	<a>**CZM**n
Units	n = mode
Range	1 - 3
Default	1
Response	*Z_MODE=1...POSITION *Z_MODE=2...TORQUE *Z_MODE=3...VELOCITY
See also	None

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

The **CZM** command defines the mode of the Z Drive controller:

1 = Position servo mode

2 = Torque mode (from analog input)

3 = Velocity mode (from analog input)

Changing the operating mode of the Z Drive with the **CZM** command will reload all of the default tuning parameters for that particular mode. These new parameters are automatically saved. Refer to ***Modes of Operation*** in *Chapter ④ Application Design* for a detailed description of these parameters.

Command	Description
> CZM1	Defines the controller as a position servo

DCA Display Current Average

Version A

Type	Status
Syntax	a**DCA**<n>
Units	n = amps
Range	Motor Dependent
Default	None
Response	*AVERAGE_CURRENT_(AMPS)=nnn.nn
See also	DCP, DCI

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

This command reports the average current commanded to the motor in amperes. This number is calculated by averaging 256 samples of instantaneous current.

- a**DCA1** Reports the average current only once.
- a**DCA** Reports the average current continuously until a character is sent.

Command	Response
> 1DCA	*AVERAGE_CURRENT_(AMPS)=012.45

DCI Display Current Instantaneous

Version A

Type Status
Syntax a**DCI**<a>
Units n = Amps
Range Motor Dependent
Default None
Response *INSTANTANEOUS_CURRENT_(AMPS)=nnn.nn
See also DCA, DCP

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

This command reports the instantaneous current commanded to the motor in amperes.

- ❑ a**DCI1** Reports the instantaneous current only once.
- ❑ a**DCI** Reports the instantaneous current continuously until a character is sent.

Command	Response
> 1DCI	*INSTANTANEOUS_CURRENT_(AMPS)=032.88

DCP Display Current Peak

Version A

Type Status
Syntax a**DCP**<a>
Units n = Amps
Range Motor Dependent
Default None
Response *PEAK_CURRENT_(AMPS)=nnn.nn
See also DCA, DCI

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

This command reports the largest instantaneous current commanded by the controller since the **DCP** command was issued.

- ❑ a**DCP1** Reports the peak current only once.
- ❑ a**DCP** Reports the peak current continuously until a character is sent.
- ❑ a**DCPØ** Resets peak current hold then, reports the peak current continuously until a character is sent.

Command	Response
> 1DCP	*PEAK_CURRENT_(AMPS)=36.42

DDI Display Drive Information

Version A

Type Status
Syntax a**DDI**
Units None
Range None
Default None
Response See Below
See also None

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

This command displays all tuning parameters and other drive parameters at one time. A sample response is provided below.

```
            PP    PI    PD    VP    VI    VF    TG
*PERCENT    50    10    00    20    00    50    60
*MAXIMUM 10000   400  32000  5000   00  32000 32000

*TIME_CONSTANT=00005(*100_microSeconds)
*AVE_CURRENT_LIMIT=020.00_AMPS
*PEAK_CURRENT_LIMIT=040.00_AMPS
*MOTOR_RESOLUTION=05000_STEPS
*RESOLVER_RESOLUTION=AUTO
*MOTOR_TYPE=Z640
*Z_MODE=1...[POSITION]
```

DISP' Front Panel Display

Version A

Type Status
Syntax <a>**DISP'**x
Units x = characters
Range up to 126 characters
Default OK
Response None
See also None

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

The **DISP'** command is used to change the *home* front-panel display message. The default message is **OK** (fixed, non-scrolling). The display message can either be a fixed message of 4 characters or less, or a scrolling message of up to 126 characters.

A four character display message can be entered by typing **> DISP'xxxx** where **xxxx** will be the message. If a blank space is desired in the message, an underscore must be entered (a space is interpreted as an end of command delimiter).

A scrolling display message can be entered by typing **> DISP'xxx....xxx** where **xxx....xxx** will be the scrolling message. Four blank spaces will be appended to the message. The message will repeatedly scrolled when no error messages are present.

Command	Description
> DISP'_#1	Home panel display is set to **#1**
> DISP'TOOL_SLIDE	Display will scroll **TOOL SLIDE**

DPA Display Position Actual

Version A

Type Status
Syntax a**DPA**<n>
Units n = steps
Range +2,147,483,647 -2,147,483,689
Default None
Response *ACTUAL_POSITION_(STEPS)=±nnnnnnnnn
See also DPE, DPS, CMR

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

This command displays the actual cumulative resolver position in motor steps. The value is a cumulative count based on the shaft position when the drive was enabled. This command is scaled by the Configure Motor Resolution (**CMR**) command (steps are in the user's resolution).

- ❑ a**DPA1** Reports the actual position only once.
- ❑ a**DPA** Reports the actual position continuously until a character is sent.

Command	Response
> 1DPA	*ACTUAL_POSITION_(STEPS)=+0000345092

DPE Display Position Error

Version A

Type Status
Syntax a**DPE**<n>
Units n = steps
Range +2,147,483,647 -2,147,483,689
Default None
Response *POSITION_ERROR_(STEPS)=±nnnnnnnnn
See also DPA, DPS, CMR

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

This command reports the difference between the position setpoint and the actual position. This command is scaled by the Configure Motor Resolution (**CMR**) command (steps are in the user's resolution).

- ❑ a**DPE1** Reports the actual position once without descriptive text.
- ❑ a**DPE** Reports the actual position continuously until a character is sent to the Z Drive.

Command	Response
> 1DPE	*POSITION_ERROR_(STEPS)=+0000000012

DPR Display Position Resolver

Version A

Type Status
Syntax a**DPR**<n>
Units n = resolver steps
Range 0 - 65,535
Default None
Response *RESOLVER_POSITION=nnnnn
See also CMR

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

DPR reports the resolver position. The resolver is a single-speed resolver; for one complete CW rotation of the motor shaft the resolver position will go from 0 to 65535 counts. CCW rotation will cause the resolver position to count backwards. This information is not scaled by the **CMR** command. Use this command to verify proper resolver connection.

- ❑ a**DPR** Continually reports the resolver position value until any key is pressed
- ❑ a**DPR1** Report resolver position value 1 time only without descriptive text.

Command	Response
> 1DPR	*RESOLVER_POSITION=63281

DPS

Version A

Type Status
Syntax a**DPS**<n>
Units n = steps
Range +2,147,483,647 -2,147,483,689
Default None
Response *POSITION_SETPOINT_(STEPS)=±nnnnnnnnn
See also DPA, DPE

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

This command reports the absolute position setpoint. The position setpoint indicates the Z Drive's commanded position. This position setpoint is the total number of steps that the drive has received since the amplifier was enabled.

- ❑ a**DPS** Continually reports the setpoint position value until any key is pressed
- ❑ a**DPS1** Reports position setpoint value 1 time only without descriptive text.

Command	Response
> 1DPS	*POSITION_SETPOINT_(STEPS)=+0000538112

DSP Display Servo Picture

Version A

Type Status
Syntax a**DSP**
Units None
Range None
Default None
Response None
See also SPA, DPE, DPS, DPR, DVA, DVE, DVS

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

This command draws a simple picture of the Z Drive servo loop, labeling various nodes, and then repeatedly displays most servo variables. This command can be used to display **DPA**, **DPE**, **DPS**, **DPR**, **DVA**, **DVE**, and **DVS** information. The servo variables are sampled at different points in time, the numbers shown will not necessarily correlate exactly.

DVA Display Velocity Actual

Version A

Type Status
Syntax a**DVA**<n>
Units n = rpm
Range 0 - 8,000
Default None
Response *ACTUAL_VELOCITY_(rpm)-±nnnnn
See also DVS, DVE

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

This command reports the shaft velocity in rpm (revolutions per minute).

- ❑ a**DVA** Continuously reports the actual velocity until any key is pressed
- ❑ a**DVA1** Report the actual velocity once, without descriptive text.

Command	Response
> 1DVA	*ACTUAL_VELOCITY_(rpm)=+05732

DVE

Version A

Type	Status
Syntax	a**DVE**<n>
Units	n = rpm
Range	0 - 8,000
Default	None
Response	*VELOCITY_ERROR_(rpm)-±nnnnn
See also	DVS, DVA

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

This command reports the difference between the velocity setpoint and the actual velocity.

- ❑ a**DVE1** Reports the velocity error only once.
- ❑ a**DVE** Reports the velocity error continuously until a character is sent.

Command	Response
> 1DVE	*VELOCITY_ERROR_(rpm)=+00003

DVS Display Velocity Setpoint

Version A

Type	Status
Syntax	a**DVS**<n>
Units	n = rpm
Range	0 - 8,000
Default	None
Response	*VELOCITY_SETPOINT_(rpm)-±nnnnn
See also	DVA, DVE

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

This command continuously reports the commanded shaft velocity setpoint in rpm (revolutions per minute).

- ❑ a**DVS** Continuously reports the velocity setpoint until any key is pressed.
- ❑ a**DVS1** Report the velocity setpoint once, without descriptive text.

Command	Response
> 1DVS	*VELOCITY_SETPOINT_(rpm)=+03712

E Enable Communications Interface

Version A

Type	Communication
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 drive to accept commands over the RS-232C interface. You can re-enable the communications interface with this command if you had previously disabled the interface with the **F** command. If several units are using the same communications interface, the **E** and **F** commands can help streamline programming.

Command	Description
> F	Disables all units on the RS-232C interface
> 1E	Enables Device 1
> 4E	Enables Device 4
> CPP35	Sets CPP value to 35% for device 1 and 4 only

F Disable Communications Interface

Version A

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

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

Use the Disable Communications Interface (**F**) command to program multiple units on a single interface. Units that are not intended to process global commands should receive device specific **F** commands. This allows you to program other units without specifying a device identifier on every command.

Command	Description
> 1F	Disables communications on Device 1
> 3F	Disables communications on Device 3
> CCA15	Sets CCA value to 15 on all units except devices #1 and #3

FMCA Find Motor Commutation Angle

Version A

Type Set-Up
Syntax a**FMCA**
Units None
Range None
Default 0
Response See Below
See also CMTR

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

All Z Drive motors have aligned resolvers so no calculation of the commutation angle is required. However, the **FMCA** command will recalculate this commutation angle if necessary. **FMCA** may be used to guarantee that the stored commutation angle is correct.

FMCA rotates the motor under program control in the drive. *While it is rotating, the user has no control over the motor. Once you start this sequence of events, you will have to cut off AC power to stop the motor before the commutation procedure is completed.* The motor must be disconnected from the load when **FMCA** is executed to ensure proper commutation angle.

Once commutation is completed, this angle is automatically saved in non-volatile memory. For any given motor, this commutation procedure does not have to be repeated.

Command	Description / Response
> 1OFF	Turns amplifier off
> 1FMCA	*COMMUTATION_OFFSET=00006...[000_DEGRESS]

^H Backspace

Version A

Type Communication
Syntax **^H**
Units None
Range None
Default None
Response None
See also None

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

^H allows you to delete the last character that you entered. (**^H** indicates that the Ctrl key is held down when the **H** key is pressed.) **^H** will not prevent execution of a command after the command delimiter has been entered. The drive will backspace one character in the command buffer, regardless of what appears on the terminal. On some terminals, the Ctrl and the left arrow keys produce the same character.

Command	Description
> CCX^HP35	CCP setting is 35A

HELP Help

Version A

Type Status
Syntax a**HELP**
Units None
Range None
Default None
Response None
See also None

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

This command lists all of the Z Drive commands and their descriptions.

OFF Off

Version A

Type Programming
Syntax <a>**OFF**
Units None
Range None
Default None
Response None
See also ON

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

This command disables the amplifier, no current is commanded to the motor. When you issue an **OFF** command, the fault LED will be on to indicate that the drive is shut down. You must issue an **ON** command to re-energize the motor and clear the fault. **OFF** removes motor torque, allowing you to move the motor manually. *If **OFF** is issued during motor motion, the motor will freewheel.*

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

ON On

Version A

Type Programming
Syntax <a>**ON**
Units None
Range None
Default None
Response None
See also OFF

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.

Command	Description
> ON	Turns the amplifier on

OUT Real Time Output Control

Version A

Type Set-Up
Syntax <a>**OUT**n
Units None
Range 0 - 3
Default 0
Response *OUT=Ø...RTO_LOW
*OUT=1...RTO_HIGH
*OUT=2...TEMPERATURE_PRE-FAULT
See also None

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

This command can activate (**OUT1**) or negate (**OUTØ**) the differential RTO (real time output) line or indicate a temperature fault (**OUT2**).

- **OUTØ** = RTO Low, never saved
- **OUT1** = RTO High, never saved
- **OUT2** = Temperature pre-fault, automatically saved
- **OUT3** = Over-voltage warning, automatically saved

*The **OUTØ**, **OUT1** commands are not savable.* The RTO line can be used to get an advance notice, or a pre-fault, of a temperature fault by entering **OUT2**. The RTO line will be activated one minute (fixed) prior to the amplifier shutdown when a PCB or motor temperature fault occurs. The system does not provide a pre-fault indication for a heatsink temperature fault. OUT3 will activate the RTO line whenever the over-voltage warning is active.

Command	Description
> OUT1	Activates the RTO line

RFS Return to Factory Settings

Version A

Type Communication
Syntax <a>**RFS**
Units None
Range None
Default None
Response None
See also SV

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

RFS returns the following setup parameters to their default values: **ANDB**, **ANV**, **CCA**, **CCP**, **CDB**, **CIP**, **CMV**, **CPD**, **CPDM**, **CPE**, **CPI**, **CPIM**, **CPP**, **CPPM**, **CRR**, **CTC**, **CTG**, **CTGM**, **CVE**, **CVF**, **CVFM**, **CVI**, **CVIM**, **CVP**, **CVPM**, **DISP'**, and motor commutation information**.**

You cannot execute the **RFS** command if the drive is enabled. Use the **OFF** command to turn the drive off. The tuning parameters that are returned to factory defaults *are not* automatically saved. Issue a Save (**SV**) command to save the tuning parameters in non-volatile memory. Non-tuning parameters *are* automatically saved.

Command	Description
> OFF	Turns amplifier off
> RFS	Set-up parameters are returned to default values
> ON	Turns amplifier on

RSE Report Servo Errors

Version A

Type Status
Syntax a**RSE**<n>
Units None
Range None
Default None
Response See Below
See also None

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

In addition to the front panel display, you can report error or warning conditions over RS-232C by using this command. Use **RSE** to troubleshoot the unit when a fault occurs. *Chapter ⑧ Troubleshooting & Maintenance* discusses all of these errors and offers troubleshooting help. The possible messages are:

```
*NO_ERRORS/WARNINGS
*DRIVE_WARNINGS:
       *Ø3_OVER_VOLTAGE
*DRIVE_ERRORS:
       *Ø4_OVER_VOLTAGE
       *17_INDEXER_SHUTDOWN
       *18_LOW_VOLTAGE
       *19_SHORT_CIRCUIT:_CYCLE_POWER
       *2Ø_POSITION_ERROR_EXCEEDED
       *21_ROLLING_AVE_CURRENT_FAULT
       *22_AVERAGE_CURRENT_LIMIT_EXCEEDED
       *23_DRIVE_ENABLE_NOT_ACTIVE
       *27_VELOCITY_ERROR_EXCEEDED
       *3Ø_ZRAM_FAILURE
       *55_POWER_SUPPLY_FAILURE
       *6Ø_COMMANDED_SHUTDOWN
       *61_INCOMING_INDEXER_PULSES
       *7Ø_RESOLVER_DISCONNECTED
       *77_RESOLVER_RESOLUTION_FAULT
       *91_HEATSINK_OVER_TEMPERATURE
       *92_MOTOR_OVER_TEMPERATURE
       *93_PCB_OVER_TEMPERATURE
```

RSEØ displays old errors; errors that have been cleared by turning the drive on (**ON** command or a reset). The microprocessor saves errors that disable the amplifier. Additional errors that occur after the amplifier is off will not be saved. Errors are saved prior to the microprocessor canceling the drive fault output on the Indexer Connector. If you want to remove power from the Z Drive when a drive fault occurs, use the drive fault output to ensure that the microprocessor has enough time to save the errors before power is lost. The fault output on the screw terminal I/O[2] connector is a hardware fault signal (hardware generated fault) that will be activated before the microprocessor saves errors.

Command	Response
> 1RSE	*DRIVE_ERRORS: *2Ø_POSITION_ERROR_EXCEEDED
> ON	
> 1RSE	*NO_ERRORS/WARNINGS
> 1RSEØ	*OLD_ERRORS: *2Ø_POSITION_ERROR_EXCEEDED

RV Revision Level

Version A

Type Status
Syntax a**RV**
Units None
Range None
Default None
Response *92-010672-01xx *92-010799-01xx
See also None

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

The Revision (**RV**) command reports the software part numbers of both Z Drive processors and their revision levels. The part numbers (92-010672-01 and 92-010799-01) identify the installed software. The suffixes (xx) identify the revision levels. Record this information, it is required if you consult Compumotor's Applications Department.

Command	Response
> 1RV	*92-010672-01A *92-010799-01A

The software is identified by 92-010672-01 revision level A, and 92-010799-01 revision level A.

SS Set-Up Status Report

Version A

Type Status
Syntax a**SS**
Units None
Range None
Default None
Response See Below
See also SSA, SSI, SSM, SSR, SST, SSU

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

Reports the status of the **SS** commands (**SSA**, **SSI**, **SSM**, **SSR**, and **SST**) settings.

Command	Response
> 1SS	*SSA=Ø...RS232_ECHO=ON *SSI=Ø...INTERACTIVE_MODE=ON *SSM=Ø...ACTIVE_DIR_INPUT=CW *SSR=Ø...ANALOG_OUT=VELOCITY *SST=Ø...ENABLE_INPUT=NORMAL *SSU=1...WARNING_DISPLAY=OFF

SSA RS-232C Echo Control

Version A

Type Set-Up
Syntax <a>**SSA**n
Units n = mode
Range 0, 1
Default 0
Response None
See also None

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

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

- **SSAØ** = Echo on
- **SSA1** = Echo off

In the Echo On (**SSAØ**) mode, characters that are received by the drive are echoed automatically. In the Echo Off (**SSA1**) mode, characters are not echoed from the drive. Use **SSA1** if you do not want the drive to echo. In a daisy chain, you must have the echo turned on to allow drives further down the chain to receive commands.

Command	Description
> SSA1	Turns echo off

SSI Interactive Mode Control

Version A

Type Set-Up
Syntax <a>**SSI**n
Units n = mode
Range 0, 1
Default 0
Response None
See also None

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

This command turns the RS-232C interactive mode on and off.

- **SSIØ** = Interactive mode on
- **SSI1** = Interactive mode off

If the interactive mode is on (**SSIØ**), the drive transmits a command prompt after each RS-232C command has been executed. The command prompt consists of a carriage return, line feed, a '**>**', and a space. If the last command was rejected, then a '**?**' character would be sent (instead of a '**>**'). '***READY**' will be sent each time the Z Drive is reset.

If the interactive mode is off (**SSI1**), then no prompt is sent. Use **SSI1** if you do not want to transmit a carriage return or line feed. Only device address 1 is capable of sending command prompts.

Command	Description
> SSI1	Turns the command prompt off

SSM Direction Input Polarity Control

Version A

Type Set-Up
Syntax <a>**SSM**n
Units n = mode
Range 0, 1
Default 0
Response None
See also None

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

This command can be used to reverse the polarity of the direction input.

- ❑ **SSMØ** = Active Direction input is CW
- ❑ **SSM1** = Active Direction input is CCW

Compumotor's standard interpretation is an active DIR input (DIR+ = 5V, DIR- = 0V) equals CW. With **SSM1**, you can reverse the polarity if desired.

Command	Description
> SSMØ	An active DIR input will cause incoming steps to be interpreted as a CW commanded position

SSR Analog Output Control

Version A

Type Set-Up
Syntax <a>**SSR**n
Units n = mode
Range 0, 1
Default 0
Response None
See also None

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

You can use **SSR** to monitor either velocity or torque at the monitor output on the Z Drive.

- ❑ **SSRØ =** Analog monitor output is velocity
- ❑ **SSR1** = Analog monitor output is the absolute torque value

Command	Description
> SSRØ	The motor shaft velocity will be in an analog format at the monitor output

SST Enable Input Control

Version A

Type Set-Up
Syntax <a>**SST**n
Units n = mode
Range 0, 1
Default 0
Response None
See also None

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

- ❑ **SSTØ** Enable input is in *normal* mode, once disabled from this input the drive remains disabled until an **ON** command or a drive reset.
- ❑ **SST1** Enable input is in *Shutdown* mode, once disabled from this input the drive can be re-enabled by replacing the jumper (see *Chapter ③ Installation* for detailed description of the Enable input). In this *shutdown* mode the Enable input behaves just like the Remote Power Shutdown input.

Command	Description
> SST1	Enable input is in Shutdown mode

SSU Set Warning Display Mode

Version A

Type Set-Up
Syntax <a>**SSU**n
Units n = mode
Range 0, 1
Default 0
Response None
See also None

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

SSUØ = Enables Warning Display mode
SSU1 = Disables Warning Display mode

When **SSU1** is entered the Z will not display warning messages. When **SSUØ** is entered, the Z will display all pending warning messages.

SV Save Tuning Parameters

Version A

Type	Set-Up
Syntax	<a>**SV**
Units	None
Range	None
Default	None
Response	None
See also	None

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

SV will save all current tuning parameters that have been defined (into battery backed RAM). The following parameters are saved in response to a **SV** command. Commands not listed here may be automatically saved when that command is entered. See the description of those individual commands: **CPD**, **CPDM**, **CPI**, **CPIM**, **CPP**, **CPPM**, **CTC**, **CVD**, **CVDM**, **CVI**, **CVIM**, **CVP**, **CVPM**.

Command	Description
> SV	Response after successful save is:*****

Z Reset

Version A

Type	Programming
Syntax	<a>**Z**
Units	None
Range	None
Default	None
Response	None
See also	None

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

The Reset (**Z**) command is equivalent to cycling AC power to the Z Drive. This command returns all internal settings to their power-up values. *When the Z Drive is reset, the drive is busy for approximately 25 ms for system initialization (all commands are ignored).*

Command	Description
> Z	Software reset of drive