Copley ASCII Interface Programmer's Guide

PN/ CC95-00404-000
Revision A
June 2009

TABLE OF CONTENTS

About This Manual

Overview and Scope

This manual describes Copley ASCII Interface developed by Copley Controls. This manual was written for the reader who has a basic knowledge of motion control theory and operation, Copley Controls amplifiers, and Copley Controls CME 2 software.

Related Documentation

See the user guides and data sheets for the amplifiers that will be programmed using the ASCII Interface. Also, the *Copley Controls Amplifier Parameter Dictionary* contains the complete list of amplifier variables. These documents can be found under *Documents* at:
http://www.copleycontrols.com/Motion/Downloads/index.html

Information on Copley Controls Software can be found at:
http://www.copleycontrols.com/Motion/Products/Software/index.html

Comments

Copley Controls welcomes your comments on this manual. See http://www.copleycontrols.com for contact information.

Copyrights

Document Validity

We reserve the right to modify our products. The information in this document is subject to change without notice and does not represent a commitment by Copley Controls.
Copley Controls assumes no responsibility for any errors that may appear in this document.

Conventions Used When Describing Amplifier Variables

As in the example shown below, this manual contains many descriptions of amplifier variables.

The Bank column indicates whether a variable can be found in RAM (R), flash (F), or both (R F).

Velocity Loop Limits Variables

Variable ID	Bank	Description
0x3a	R F	Velocity loop velocity limit. Units: 0.01 counts/second.
0x36	R F	Velocity loop acceleration limit. Units: 1000 counts/second^2.
0x37	R F	Velocity loop deceleration Limit. Units: 1000 counts/second^2.
0xcf	R F	Fast Stop Ramp. Units: 10 counts/second^2.

Product Warnings

Observe all relevant state, regional, and local safety regulations when installing and using Copley Controls amplifiers. For safety and to assure compliance with documented system data, only Copley Controls should perform repairs to amplifiers.

DANGER

DANGER: Hazardous voltages.

Exercise caution when installing and adjusting.

Failure to heed this warning can cause equipment damage, injury, or death.

DANGER

Risk of electric shock.

Amplifier high-voltage circuits are connected to DC or AC power.

Failure to heed this warning can cause equipment damage, injury, or death.

DANGER

Using CME 2 can affect or suspend externally controlled operations.

When operating the amplifier under control of the ASCII Interface, use of CME 2 to change amplifier parameters can affect operations in progress.

Using CME 2 to initiate motion can cause external program operations to suspend. The operations may restart unexpectedly when the CME 2 move is stopped.

Failure to heed this warning can cause equipment damage, injury, or death.

Revision History

Revision	ECO #	Date	Applies to	Comments
1		August 2005	Latest CME 2 version and latest firmware version.	First general release.
2	14899	December 2006		Updates and corrections.
3	15299	March 2007		Deleted obsolete material.
4	17137	June 2008		Minor updates, including updates to Web page references.
A	33392	June 2009		Support of multi-axis amplifiers and other updates.

CHAPTER
1: INTRODUCTION

This chapter provides an overview of the Copley ASCII Interface, a set of commands that can be sent by an external controller to operate and monitor Copley Controls amplifiers.

Contents include:

1.1: The Copley ASCII Interface

The Copley ASCII Interface is a set of ASCII format commands that can be used to operate and monitor Copley Controls Accelnet, Stepnet, and Xenus series amplifiers over an RS-232 serial connection.

For instance, after basic amplifier configuration values have been programmed using CME 2, a control program can use the ASCII Interface to:

- Enable the amplifier in Programmed Position mode.
- Home the axis.
- Issue a series of move commands while monitoring position, velocity, and other run-time variables.

Commands and Variables

Some of the ASCII Interface commands read, write, and copy the values of a set of amplifier variables.

Some of these variables are used to control and monitor the amplifier's operating modes and states. For instance, variable 0x24, the "amplifier desired state" variable, is used to enable the amplifier in one of a variety of operating modes. Other variables are used to monitor run-time information.

Not all commands affect amplifier variables directly. For instance, the reset (r) command causes the amplifier to reset immediately, and the trajectory (t) command initiates and aborts moves.

RAM and Flash Memory Banks

Variables are maintained in the amplifier's RAM memory, flash memory or in both RAM and flash memory. When using commands to read, write, or copy variables, it is necessary to specify a memory bank.

Unlike flash values, RAM values are cleared with each amplifier reset. Any RAM variable with a flash counterpart is then written with the flash value. Thus, if the desired startup value is stored in flash, the external program need not write the RAM value on startup. For instance, if the desired initial Position Loop gains and limits were saved to flash using CME 2, these values would automatically be loaded into RAM on startup.

Any RAM value with no flash counterpart remains clear until updated by an external program or by a change in the value watched by the variable. For instance, the actual current variable (0x0c) updates to match the actual current value.

1.2: Communication Protocol

The protocol used is of the "speak when spoken to" variety. The amplifier will never initiate communications, but will always respond to commands with an acknowledgment, a returned value, or an error code.

The baud rate of the amplifier will always be set to 9600 on power up or after a reset. Also, Copley amplifiers are designed to identify a break command on the serial port. A serial break command is normally an illegal condition in which the system initiating the break command holds its transmitting serial line low for longer then a single byte of serial data. If such a condition is detected by the amplifier, it will flush any pending input data, reset the amplifier's baud rate to 9600, and wait for a new command to be received on its serial port. The amplifier will never initiate a break command itself.

Use the following protocol for Copley ASCII Interface communications:

Baud Rate	9,600 to 115,200 (Defaults to 9,600 on power up or reset.) See *Setting the Baud Rate* (p 42).
Data Format	N, 8, 1
Flow Control	None

CHAPTER
2: COMMAND SET

This chapter describes the Copley Controls ASCII programming interface command set.

Contents include:

2.1: Copley ASCII Message Format

The syntax of a Copley ASCII message is:

[<optional node ID>] <command code> [<command specific parameters>...] <CR>

where:

- **[<optional node ID>]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. The node ID is followed by a single space.
- **<command code>** is the single-letter code for one of the Copley ASCII commands described below. It is followed by a single space, or, in the case of the Reset command, a carriage return character.

Command Code	Description
s	Set the value of a variable in ram or flash.
g	Get the value of a variable in ram or flash.
c	Copy the value of a variable from ram to flash or flash to ram.
r	Reset the amplifier.
t	Trajectory generator command.
i	Read or write the value of a CVM program register.

- **<command specific parameters>** tell the command what to act on and how. If more than one parameter is required, they should be separated by spaces.
- **<cr>** is a carriage return character that ends the command line.

For multi-axis amplifiers, see Copley ASCII Message Format for Multi-Axis Drives (p. 19).

2.2: Set (s) Command

The **s** command is used to set values of writeable amplifier variables.

The syntax of the **s** command is:

[optional node ID] s <memory bank><variable ID> <value>...]<CR>

where:

- **[optional node ID]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. It is followed by a single space.
- **s** is the Set command code. It is followed by a single space.
- **<memory bank>** identifies which memory bank to set the variable in.
  f = flash memory
  r = RAM memory
- **<variable ID>** identifies the variable to set. Variable ID format can be decimal or hexadecimal.
- **<value>** is the new value(s) to be set in the variable. Value can be sent in integer or hexadecimal format. If multiple values are required, they are separated by spaces.

<CR> is a carriage return character which immediately follows the last value.

The amplifier responds to the **s** command with:

- **ok<CR>** if the command is accepted.
- **e <error #><CR>** if the command was not accepted. See *Error Codes* (p. 77).

s Command Examples

Command	Response	Comment
s r0x30 1200	ok	Set variable 0x30 (position loop proportional gain) to 1200 in RAM. The "ok" response indicates that the command executed successfully.
s f0x30 1200	ok	Set variable 0x30 (position loop proportional gain) to 1200 in flash. The "ok" response indicates that the command executed successfully.
s r 0x30 1000	e 33	Attempted to set 0x30 to 1200 in RAM. Error 33 (ASCII command parsing error) was returned. Note the extra space before the variable ID.

2.3: Get (g) Command

The **g** command is used to get the values of amplifier variables.

The syntax of the **g** command is:

[optional node ID] g <memory bank><variable ID><CR>

where:

- **[optional node ID]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. It is followed by a single space.
- **g** is the Get command code. It is followed by a single space.
- **<memory bank>** identifies which memory bank to get the variable from.
  f = flash memory
  r = RAM memory
- **<variable ID>** identifies the variable to get. Variable ID format can be decimal or hexadecimal.
- **<CR>** is a carriage return character which immediately follows the variable ID.

The amplifier responds to the Get command with:

- **v [value]<CR>** where value equals the contents of the variable. If the variable contains multiple values, they will separated by spaces.
- **e <error #>(CR)** if the command was not accepted. See *Error Codes* (p. 77).

g Command Examples

Command	Response	Comment
g r0x30	v 1200	Get the value of variable 0x30 (position loop proportional gain) from RAM. Example shows a value of 1200 returned.
g f0x17	e 15	Attempted to read variable 0x17 (actual motor position) from flash. Error 15 (Variable doesn't exist on requested page) was returned. Note that actual motor position is stored in RAM only.

2.4: Copy (c) Command

The **c** command is used to copy the value of a variable from one memory bank to another (RAM to flash or flash to RAM).

The syntax of the **c** command is:

[optional node ID] c <memory bank><variable ID><CR>

where:

- **[optional node ID]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. It is followed by a single space.
- **c** is the Copy command code. It is followed by a single space.
- **<memory bank>** identifies which memory bank is the source.
  f = flash memory
  r = RAM memory
- **<variable ID>** identifies the variable to copy. Variable ID format can be decimal or hexadecimal.
- **<CR>** is a carriage return character which immediately follows the variable ID.

The amplifier responds to the **c** command with:

- **ok<CR>** if the command is accepted.
- **e <error #>(CR)** if the command was not accepted. See *Error Codes* (p. 77).

c Command Examples

Command	Response	Comment
c r0x30	ok	Copy the value of 0x30 from RAM to flash. The "ok" response indicates that the command executed successfully.
c f0x30	ok	Copy the value of 0x30 from flash to RAM. The "ok" response indicates that the command executed successfully.

2.5: Reset (r) Amplifier Command

The **r** command is used to immediately reset the amplifier. The command requires no additional parameters. The amplifier baud rate is set to 9600 when the amplifier restarts.

The syntax of the Reset command is:

[optional node ID] r<CR>

where:

- **[optional node ID]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. It is followed by a single space.
- **r** is the Reset command code.
- **<CR>** is a carriage return character which immediately follows the command code.

The amplifier does not respond to the **r** command with an ASCII message.

r Command Example

Command	Response	Comment
r	{none}	Amplifier is reset.

NOTE: if a reset command is issued to an amplifier on a multi-drop network, error code 32, "CAN Network communications failure," will be received. This is because the amplifier reset before responding to the gateway amplifier. This error can be safely ignored in this circumstance.

2.6: Trajectory (t) Generator Command

The **t** command controls the trajectory generator. It can initiate a new move, update a move in progress, or start a home sequence. It can also abort a move.

The syntax of the **t** command is:

[optional node ID] t <sub-command><CR>

where:

- **[optional node ID]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. It is followed by a single space.
- **t** is the Trajectory command code. It is followed by a single space.
- **<sub-command>**
  - 0 = Abort move
  - 1 = Initiate/update move
  - 2 = Initiate home sequence
- **<CR>** is a carriage return character which immediately follows the sub-command.

The amplifier responds to the **t** command with:

- **ok<CR>** if the command is accepted. An "ok" response only indicates the command was accepted by the amplifier. Monitor the trajectory status register to verify that motion has actually been initiated.
- **e <error #>(CR)** if the command was not accepted. See *Error Codes* (p. 77).

t Command Examples

Command	Response	Comment
t 1	ok	Initiate a move.
t2	e 33	Attempted to initiate a homing sequence. Error 33 (ASCII command parsing error) was returned. Note there is no space between the command and sub-command.

2.7: Register (i) Read and Write Command

The Register command (**i**) is used to read and write the CVM program's 32 internal registers.

The syntax of the **i** command is:

[optional node ID] i <r#> [<value>]<CR>

where:

- **[optional node ID]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. It is followed by a single space.
- **i** is the Register command code. It is followed by a single space.
- **<r#>** identifies which register is being accessed.
  \# = Equals the number of the register (0 – 31).
- **<value>** is the new value to be written into the register. If <value> is omitted from the command then the contents of the register will be returned. Value can be sent in integer or hexadecimal format.
- **<CR>** is a carriage return character which immediately follows the register number or the value.

The amplifier responds to the **i** command with:

- **ok<CR>** if the command is accepted and the value is written to the register.
- **r [value]<CR>** where value equals the contents of the register.
- **e <error #>(CR)** if the command was not accepted. See *Error Codes* (p. 77).

i Command Examples

Command	Response	
i r0 15	ok	Write the value "15" to the first register (register 0).
i r0	r 15	Read the value of the first register (register 0). Example displays a returned value equal to 15.
8 i r0	r 35	Read the value of register 0 on amplifier with CAN node ID of 8. Example displays a returned value equal to 35.

2.8: Copley ASCII Message Format for Multi-Axis Drives

Some Copley Controls amplifiers can drive up to three axes of motion. When sending an ASCII command to an amplifier that is driving more than one axis, it is necessary to specify the axis as described below:

[<optional node ID>].<axis letter> <command code> [<command specific parameters>...] <CR>

where:

- **[<optional node ID>]** is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. The node ID is followed by a period and the axis letter.
- **.<axis letter>** specifies the axis (a, b, or c). The axis letter is preceded by a period and followed by a space.
- **<command code>** is the single-letter code for one of the Copley ASCII commands described below. It is followed by a single space, or, in the case of the Reset command, a carriage return character.

Command Code	Description
s	Set the value of a variable in ram or flash.
g	Get the value of a variable in ram or flash.
c	Copy the value of a variable from ram to flash or flash to ram.
r	Reset the amplifier.
t	Trajectory generator command.
i	Read or write the value of a CVM program register.

- **<command specific parameters>** tell the command what to act on and how. If more than one parameter is required, they should be separated by spaces.
- **<cr>** is a carriage return character that ends the command line.

Multi-Axis Examples

Command	Response	Comment
.b g r0x32	v1200	Get the value of variable 0x32 (Actual Motor Position) of axis b from RAM. Example shows a value of 1200 returned.
2.c g r0x32	v1200	Get the value of variable 0x32 (Actual Motor Position) of Node 2, axis c from RAM. Example shows a value of 1200 returned.

CHAPTER
3: OPERATING MODES

This chapter describes the variables related to the amplifier's operating modes.
Contents include:

3.1: Desired State Variable

The amplifier desired state variable (0x24) defines the amplifier's operating mode and which input source controls it. Mode-specific values are mentioned in the remaining sections of this chapter. The relevant values are described in the table below:

Value	State
0	Disabled. **NOTE:** If the desired sate is saved to flash as 0, then CME 2 assumes the amplifier has not been programmed, and when CME connects to the amplifier, the Basic Set Up screen opens.
1	The current loop is driven by the programmed current value.
2	The current loop is driven by the analog command input.
3	The current loop is driven by the PWM & direction input pins.
4	The current loop is driven by the internal function generator.
5	The current loop is driven by UV commands via PWM inputs.
11	The velocity loop is driven by the programmed velocity value.
12	The velocity loop is driven by the analog command input.
13	The velocity loop is driven by the PWM & direction input pins.
14	The velocity loop is driven by the internal function generator.
21	In servo mode, the position loop is driven by the trajectory generator.
22	In servo mode, the position loop is driven by the analog command input.
23	In servo mode, the position loop is driven by the digital inputs (pulse & direction, master encoder, etc).
24	In servo mode, the position loop is driven by the internal function generator.
25	In servo mode, the position loop is driven by the camming function.
30	In servo mode, the position loop is driven by the CANopen interface.
31	In microstepping mode, the position loop is driven by the trajectory generator.
33	In microstepping mode, the position loop is driven by the digital inputs (pulse & direction, master encoder, etc).
34	In microstepping mode, the position loop is driven by the internal function generator.
35	In microstepping mode, the position loop is driven by the camming function.
40	In microstepping mode, the amplifier is driven by the CANopen interface.
42	Micro-stepping diagnostic mode. The current loop is driven by the programmed current value, and the phase angle is micro-stepped.

3.2: Current Mode

3.2.1: Programmed Current Mode

The Programmed Current Mode sets the output of the amplifier at a programmed current level. When the amplifier is enabled in this mode, or when the programmed current level is changed, the output current ramps to the new level at the programmed rate.

Programmed Current Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state: 0 = Disabled. 1 = Programmed Current Mode.
0x02	R F	Programmed current value. Units: 0.01 A.
0x6a	R F	Current ramp rate. Units: mA/second. A value of zero in this register results in a step change.

NOTE: When changing both the level and the ramp parameters while the amplifier is enabled, change the ramp rate first.

Programmed Current Mode Example

Enable the amplifier in Programmed Current Mode. Ramp the output current up to 2 A in 0.5 seconds. The controller monitors the output current, and after it reaches 2 A the current is ramped down to 1 A in 2 seconds.

Command	Response	Comment
s r0x6a 4000	ok	Set ramp rate to 4 A/second.
s r0x02 200	ok	Set the output level to 2 A.
s r0x24 1	ok	Enable the amplifier in Programmed Current Mode. Output current will start increasing at a rate of 4 A/second.

The controller uses the following command to monitor the output current.

Command	Response	Comment
g r0x0c	v 150	Reads actual current output from the amplifier. Example displays a returned value equal to 1.50 A.

After the output current reaches 2 A, the controller sends the new ramp and level parameters.

Command	Response	Comment
s r0x6a 4000	ok	Set new ramp rate of 0.5 A/second.
s r0x02 100	ok	Change the output level to 1 A. Output current will start decreasing at a rate of 0.5 A/second.

The controller disables the amplifier.

Command	Response	Comment
s r0x24 0	ok	Disable the amplifier.

3.2.2: Analog Current Mode

In the Analog Current Mode, the current output of the amplifier is proportional to the analog reference input command signal.

Analog Current Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 2 = Analog Current Mode.
0x19	R F	Analog input scaling factor. Amount of current commanded per 10 volts of input. Units: 0.01 A.
0x26	R F	Analog input dead band. Units: mV.
0x1a	R F	Analog input offset. Units: mV.

NOTE: Variables 0x19, 0x26 and 0x1a are used in Analog Current, Velocity and Position modes. Verify that these variables are set correctly before switching between these modes of operation.

Analog Current Mode Example

The controller sets the scaling, enables the amplifier in Analog Current Mode, monitors the current output, and changes the scaling to a new value.

Command	Response	Comment
s r0x19 1000	ok	Set scaling factor to 10V = 10A.
s r0x24 2	ok	Set amplifier to Analog Current Mode.

The controller uses the following command to monitor the output current.

Command	Response	Comment
g r0x0c	v 525	Reads actual current output from the amplifier. Example displays a returned value equal to 5.25 A.

The controller changes the scaling factor

Command	Response	Comment
s r0x19 100	ok	Set scaling factor to 10V = 1A.

The controller disables the amplifier.

Command	Response	Comment
s r0x24 0	ok	Disable the amplifier.

3.2.3: PWM Current Mode

In the PWM Current Mode, the current output of the amplifier is proportional to the duty cycle of the input command signal. In most applications the command signal configuration is set using CME 2 and not changed during operation.

PWM Current Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 3 = PWM Current Mode.
0xa9	R F	Digital input scaling factor. Amount of current commanded at 100 percent duty cycle. Units: 0.01 A.
0xa8	R F	Digital input command configuration normally set using the CME 2 PWM Command screen. See table below for definition of the values.

NOTE: Variables 0xa9 and 0xa8 are used in PWM Current and Velocity modes. Verify that these variables are set correctly before switching between these modes of operation.

PWM Current Mode Example

The controller sets the scaling, enables the amplifier in PWM Current Mode, and monitors commanded and actual current.

Command	Response	Comment
s r0xa9 1000	ok	Set scaling factor to 10A.
s r0x24 3	ok	Enable the amplifier in PWM Current Mode.
The controller uses the following commands to monitor the commanded and output currents .		
g r0x15	v 500	Reads commanded current from the amplifier. Example displays a returned value equal to 5 A.
g r0x0c	v 495	Reads actual current output from the amplifier. Example displays a returned value equal to 4.95 A.
The controller disables the amplifier.		
s r0x24 0	ok	Disable the amplifier.

PWM Current Mode Command Signal Configuration

If required during operation, the PWM command signal configuration can be changed by setting the value of variable 0xa8 as shown below.

PWM Input Type	Invert PWM Input	Invert Sign Input	Allow 100% Output	Value
50%	No	--	No	0x00
50%	No	--	Yes	0x08
50%	Yes	--	No	0x02
50%	Yes	--	Yes	0x0a
100%	No	No	No	0x01
100%	No	No	Yes	0x09
100%	No	Yes	No	0x05
100%	No	Yes	Yes	0x0d
100%	Yes	No	No	0x03
100%	Yes	No	Yes	0x0b
100%	Yes	Yes	No	0x07

3.3: Velocity Mode

3.3.1: Programmed Velocity Mode

The Programmed Velocity Mode sets the output of the amplifier to a programmed motor velocity. When the amplifier is enabled in this mode, or when the programmed velocity is changed, the motor velocity will ramp to the new level at the programmed rate.

Programmed Velocity Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 11 = Programmed Velocity Mode.
0x2f	R F	Programmed velocity command. Units: 0.1 counts/second.
0x36	R F	Velocity acceleration limit. Units: 1000 counts/second^2
0x37	R F	Velocity deceleration limit. Units: 1000 counts/second^2
0x39	R F	Fast stop ramp. Units: 1000 counts/second^2

Programmed Velocity Mode Example

The controller sets the velocity parameters, enables the amplifier in Programmed Velocity Mode, monitors the actual motor velocity, and then changes the velocity.

Command	Response	Comment
s r0x36 2	ok	Set acceleration limit to 2000 counts/second^2.
s r0x37 4	ok	Set deceleration limit to 4000 counts/second^2.
s r0x2f 10000	ok	Set the programmed velocity to 1000 counts/second.
s r0x24 11	ok	Enable the amplifier in Programmed Velocity Mode.
The controller uses the following commands to monitor the motor velocity .		
g r0x18	v 10010	Reads actual velocity from the amplifier. Example displays a returned value equal to 1001 counts/second.
The controller sets a new velocity of 500 counts/second.		
s r0x2f 5000	ok	Set the programmed velocity to 500 counts/second. Motor will decelerate at 4000 counts/second^2 to 500 counts/second.
The controller disables the amplifier.		
s r0x24 0	ok	Disable the amplifier.

3.3.2: Analog Velocity Mode

In the Analog Velocity Mode, the motor velocity is proportional to the analog reference input command signal.

Analog Velocity Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 12 = Analog Velocity Mode.
0x19	R F	Analog input scaling factor. Velocity commanded per 10 volts of input. Units: 0.1 counts/second
0x26	R F	Analog input dead band. Units: mV.
0x1a	R F	Analog input offset. Units: mV.
0x36	R F	Velocity acceleration limit. Units: 1000 counts/second^2
0x37	R F	Velocity deceleration limit. Units: 1000 counts/second^2
0x39	R F	Fast stop ramp. Units: 1000 counts/second^2

NOTE: Variables 0x19, 0x26 and 0x1a are used in Analog Current, Velocity and Position modes. Verify that these variables are set correctly before switching between these modes of operation.

Analog Velocity Mode Example

The controller sets the scaling, enables the amplifier in Analog Velocity Mode, monitors the actual motor velocity, and changes the scaling.

Command	Response	Comment
s r0x19 10000	ok	Set scaling factor to 1000 counts/second.
s r0x24 12	ok	Enable the amplifier in Analog Velocity Mode.

The controller uses the following command to monitor the actual motor velocity.

g r0x18	v 5000	Reads actual velocity from the amplifier. Example displays a returned value equal to 500.0 counts/second.

The controller changes the scaling factor.

s r0x19 7000	ok	Set scaling factor to 700 counts/second.

The controller disables the amplifier.

s r0x24 0	ok	Disable the amplifier.

3.3.3: PWM Velocity Mode

In the PWM Velocity Mode, the motor velocity is proportional to the duty cycle of the input command signal. In most applications the command signal configuration is set using CME 2 and not changed during operation.

PWM Velocity Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 13 = PWM Velocity Mode.
0xa9	R F	Scaling Factor. Velocity command at 100 percent duty cycle. Units: 0.1 counts/second.
0x36	R F	Velocity acceleration limit. Units: 1000 counts/second^2
0x37	R F	Velocity deceleration limit. Units: 1000 counts/second^2
0x39	R F	Fast stop ramp. Units: 1000 counts/second^2
0xa8	R F	Digital input command configuration. This is normally set using the CME 2 PWM Command screen. See table below for the definition of the values.

NOTE: Variables 0xa9 and 0xa8 are used in PWM Current and Velocity modes. Verify that these variables are set correctly before switching between these modes of operation.

PWM Velocity Mode Example

The controller sets the PWM scaling, enables the amplifier in PWM Velocity Mode, and monitors the commanded and actual velocity.

Command	Response	Comment
s r0xa9 800000	ok	Set scaling factor to 80000 counts/second at 100%.
s r0x24 13	ok	Enable the amplifier in PWM Velocity Mode.

The controller uses the following commands to monitor the commanded and actual motor velocities.

Command	Response	Comment
g r0x2c	v 49995	Reads commanded velocity from the amplifier. Example displays a returned value equal to 4999.5 counts/second.
g r0x18	v 49991	Reads actual velocity from the amplifier. Example displays a returned value equal to 4999.1 counts/second.

The controller disables the amplifier.

Command	Response	Comment
s r0x24 0	ok	Disable the amplifier.

PWM Velocity Mode Command Signal Configuration

If required during operation, the PWM command signal configuration can be changed by setting the value of variable 0xa8 as shown below.

PWM Input Type	Invert PWM Input	Invert Sign Input	Allow 100% Output	Value
50%	No	--	No	0x00
50%	No	--	Yes	0x08
50%	Yes	--	No	0x02
50%	Yes	--	Yes	0x0a
100%	No	No	No	0x01
100%	No	No	Yes	0x09
100%	No	Yes	No	0x05
100%	No	Yes	Yes	0x0d
100%	Yes	No	No	0x03
100%	Yes	No	Yes	0x0b
100%	Yes	Yes	No	0x07

3.4: Position Mode

3.4.1: Updating Trajectory Variables in Position Modes

When the amplifier enters a position mode, the trajectory variables (velocity, acceleration and deceleration) are copied into the trajectory generator. To change any of them after the amplifier is in a position mode, send the new value to the appropriate variable and then send a *t 1*command.

3.4.2: Programmed Position Mode

In the Programmed Position Mode, the axis moves to target positions sent to the amplifier over the serial interface. The target positions can be absolute or relative from the current position. The motion profile used can be set to trapezoidal or S-curve.

To initiate a move, first set the appropriate variables and then send the trajectory command *t 1* to start the move (see *Trajectory (t) Generator Command*, p. 17). When using the trapezoidal profile, the move parameters can be changed during the move. Again, first set the appropriate variables and then send another *t 1* command. When the *t 1* command is received, the target position, absolute / relative, velocity, acceleration and deceleration rates will be updated. In this manner, the move in progress can be changed. The S-curve profile cannot be updated in this manner.

To abort a move in progress, send a *t 0* command. This will stop the move in progress using the abort deceleration rate. The amplifier will remain enabled.

A special velocity mode can be used to move the axis using the velocity, acceleration and deceleration of the trapezoidal profile but with no specific target position. Direction of motion is set by entering a "1" or "-1" into the position command variable. Once started, the move will continue until the velocity variable is set to zero and a t 1 command is sent or a *t 0* abort command is sent.

Programmed Position Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 21 = Programmed Position Mode, Servo 31 = Programmed Position Mode, Stepper
0xc8	R F	Profile type. 0 = Absolute move, trapezoidal profile. 1 = Absolute move, S-curve profile. 256 = Relative move, trapezoidal profile. 257 = Relative move, S-curve profile. 2 = Velocity move.
0xca	R F	Position command. Units: counts. Relative move = the distance of the move. Absolute move = the target position of the move. Velocity move = 1 for positive direction, -1 for negative direction.
0xcb	R F	Maximum velocity. Units: 0.1 counts/second.
0xcc	R F	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	R F	Maximum deceleration rate. Units: 10 counts/second^2.
0xce	R F	Maximum jerk rate. Units: 100 counts/ second^3.
0Xcf	R F	Abort deceleration rate. Units: 10 counts/second^2.

NOTES: 1) Maximum jerk rate is not used in the trapezoidal profile. 2) In the S-curve profile, the maximum deceleration rate is note used. The maximum acceleration rate is used for both acceleration and deceleration.

Programmed Position Mode Example

The controller sets profile parameters, executes an absolute trapezoidal move to position 40,000 counts, monitors for move completion, and then executes a relative move of 10,000 counts using the same profile parameters.

Command	Response	Comment
s r0xc8 0	ok	Set the trajectory generator to absolute move, trapezoidal profile.
s r0xca 40000	ok	Set the position command to 40000 counts.
s r0xcb 70000	ok	Set maximum velocity to 7000 counts/second.
s r0xcc 200000	ok	Set maximum acceleration to 2000000 counts/second^2.
s r0xcd 200000	ok	Set maximum deceleration to 2000000 counts/second^2.
s r0x24 21	ok	Enable the amplifier in Programmed Position (Trajectory Generator) Mode.
The controller verifies actual axis position before starting move.		
g r0x32	v 0	Read actual position. Example displays an actual position of 0.
t 1	ok	Execute the move.
The controller monitors the event status register to determine when the move has been completed.		
g r0xa0	v 134217728	The controller monitors bit 27 of the event status register to determine when the move is complete. Example displays a status register value of 134217728. When this is decoded, it shows that bit 27 is set meaning the axis is in motion.
After the controller determines that motion has stopped, it checks the trajectory status register to see if the move was aborted for any reason.		
g r0xc9	v 4096	The controller checks bit 14 of the trajectory status register to determine if the move was aborted. Example displays a status register value of 4096. When this is decoded, it shows that bit 14 is not set meaning the move was not aborted.
The controller sets the trajectory configuration and commanded position variables to their new values and executes the new move.		
s r0xc8 256	ok	Set the trajectory generator to relative move, trapezoidal profile.
s r0xca 10000	ok	Set the position command to 10000 counts.
t 1	ok	Execute the move.
The controller aborts the move.		
t 0	ok	Motion stops and the amplifier is left enabled
The controller disables the amplifier.		
s r0x24 0	ok	Disable the amplifier.

3.4.3: Analog Position Mode

In the Analog Position Mode, the axis position is commanded by the analog reference input command signal.

The analog position command operates as a relative motion command. When the amplifier is enabled the voltage on the analog input is read. Then any change in the command voltage will move the axis a relative distance, equal to the change in voltage, from its position when enabled. To use the analog position command as an absolute position command, the amplifier should be homed every time it is enabled.

Analog Position Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 22 = Analog Position Mode.
0x19	R F	Analog input scaling factor. Commanded position per 10 volts of input. Units: counts.
0x26	R F	Dead band. Units: mV.
0x1a	R F	Analog input offset. Set to 0 when in this mode of operation.
0xcb	R F	Maximum velocity. Units: 0.1 counts/second.
0xcc	R F	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	R F	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	R F	Abort deceleration rate. Units: 10 counts/second^2.

NOTES: 1) Variables 0x19, 0x26 and 0x1a are used in Analog Current, Velocity and Position modes. Verify that these variables are set correctly before switching between these modes of operation. 2) To invert the direction of motion with respect to the polarity of the command voltage, set the scaling factor as a negative value.

Analog Position Mode Example

The controller sets the move parameters, homes the axis and then places the amp in the Analog Position Mode. The controller monitors actual position. The controller then changes the maximum velocity and scaling factor.

Command	Response	Comment
s r0x19 4000	ok	Set analog scaling to 4000 counts per 10V.
s r0xcb 70000	ok	Set velocity to 7000 counts/second
s r0xcc 20000	ok	Set acceleration to 200000 counts/second^2
s r0xcd 20000	ok	Set deceleration to 200000 counts/second^2
s r0x24 21	ok	Amplifier set in Programmed Position Mode required for homing.
t 2	ok	Execute homing. Assumes all homing parameters have been previously set.
The controller monitors the trajectory status register to determine when the axis has been homed.		
g r0xc9	v 8192	Controller checks bit 12 of the trajectory status register to determine if the axis was homed successfully. Example displays a status register value of 8192. Decoded, this value shows that bit 12 is not set, meaning the axis has not finished homing.
After a successful homing, the controller changes the amplifier's operating mode.		
s r0x24 22	ok	Amplifier set in Analog Position Mode
t 1	ok	This command will guarantee all new move parameters are in effect.
The controller monitors actual motor position.		
g r0x32	v 2012	Reads actual motor position from the amplifier. Example displays a returned value equal to 2012 counts.
The controller changes velocity and scaling variables		
s r0xcb 20000	ok	Set velocity to 2000 counts/second
s r0x19 1000	ok	Set analog scaling to 1000 counts / 10V input.
t 1	ok	This command required for new velocity to take effect.
The controller disables the amplifier.		
s r0x24 0	ok	Disable the amplifier.

3.4.4: Pulse and Direction Mode

In the Pulse and Direction Position Mode, the axis position is commanded by pulses applied to one of the amplifiers digital inputs. The direction of the commanded move is determined by the logic level of a second digital input.

The scaling factor sets the ratio of position command, in counts, for each input pulse. This ratio is stored in variable 0xa9 as two 16 bit words. The first word stores the numerator or number of position counts. The second stores the denominator or the number of input pulses.
Example: To set a ratio of 10 counts of position change for every input pulse.
The ration would be 10/1. To make sending the data easier, it should be converted to hex word format so the ratio would now be 0x000a / 0x0001. The two words can now simply be combined and sent to the amplifier by sending the command *s r0xa9 0x000a0001*.

To invert the direction, the numerator should be set to a negative value.
Example: Changing direction of the previous example would require a ratio of -10/1. Using the 2's complement method, -10 is represented as 0xfff6 hex. The ratio in hex would now be 0xfff6 / 0x001. Combining these words, the command to be sent would be *s r0xa9 0xfff60001*.

Pulse and Direction Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state. 0 = Disabled 23 = Digital Input Position Mode, Servo 33 = Digital Input Position Mode, Stepper
0xa8	R F	Digital Command Configuration Pulse and Direction 0 = Increment position on rising edge 4096 = Increment position on falling edge
0xa9	R F	Scaling factor. Output counts/Input pulses
0xcb	R F	Maximum velocity. Units: 0.1 counts/second.
0xcc	R F	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	R F	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	R F	Abort deceleration rate. Units: 10 counts/second^2.

Pulse and Direction Mode Example

The controller sets the move parameters, places the amp in the Pulse and Direction Position Mode, monitors commanded and actual position, and then changes the scaling factor.

Command	Response	Comment
s r0xa8 0	ok	Configure the digital inputs to pulse and direction with the position incrementing on the rising edge of the input pulse.
s r0xa9 0x00020001	ok	Set scaling factor to 2 output counts per input pulse.
s r0xcb 70000	ok	Set velocity to 7000 counts/second.
s r0xcc 20000	ok	Set acceleration to 200000 counts/second^2.
s r0xcd 20000	ok	Set deceleration to 200000 counts/second^2.
s r0x24 23	ok	Enable the amplifier in Digital Input Position Mode.
The controller monitors commanded and actual motor position.		
g r0x3d	v 4000	Reads commanded position from the amplifier. Example displays a returned value equal to 4000 counts.
g r0x32	v 2012	Reads actual motor position from the amplifier. Example displays a returned value equal to 2012 counts.
The controller changes the scaling variable.		
s r0xa9 0x00010001	ok	Set scaling factor to 1 output count per input pulse.
The controller disables the amplifier.		
s r0x24 0	ok	Disable the amplifier.

3.4.5: Pulse Up/Down Mode

In the Pulse Up/Down Position Mode, the axis position is commanded by pulses applied to the amplifiers digital inputs. The direction of the commanded move is determined by which of the digital inputs the pulses are applied to.

The scaling factor sets the ratio of position command, in counts, for each input pulse. It is stored in variable 0xa9 as two 16 bit words. The first word stores the numerator or number of position counts. The second stores the denominator or the number of input pulses.
Example: To set a ratio of 10 counts of position change for every input pulse.
The ration would be 10/1. To make sending the data easier, it should be converted to hex word format so the ratio would now be 0x000a / 0x0001. The two words can now simply be combined and sent to the amplifier by sending the command *s r0xa9 0x000a0001*.

To invert the direction, the numerator should be set to a negative value.
Example: Changing direction of the previous example would require a ratio of -10/1. Using the 2's complement method, -10 is represented as 0xfff6 in hex format. The ratio in hex format would now be 0xfff6 / 0x001. Combining these words, the command to be sent would be
s r0xa9 0xfff60001.

Pulse Up/Down Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state: 0 = Disabled. 23 = Digital Input Position Mode, Servo. 33 = Digital Input Position Mode, Stepper.
0xa8	R F	Digital Command Configuration: Pulse Up/Down Mode. 256 = Increment position on rising edge. 4352 = Increment position on falling edge.
0xa9	R F	Input / Output Ratio.
0xcb	R F	Maximum velocity. Units: 0.1 counts/second.
0xcc	R F	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	R F	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	R F	Abort deceleration rate. Units: 10 counts/second^2.

Pulse Up/Down Mode Example

The controller sets the move parameters and then places the amp in the Pulse Up/Down Position Mode, monitors commanded and actual position, and then changes the scaling factor.

Command	Response	Comment
s r0xa8 256	ok	Configure the digital inputs to pulse up/down with the position incrementing on the rising edge of the input pulse.
s r0xa9 0x000f0001	ok	Set scaling factor to 15 output counts per input pulse.
s r0xcb 70000	ok	Set velocity to 7000 counts/second.
s r0xcc 20000	ok	Set acceleration to 200000 counts/second^2.
s r0xcd 20000	ok	Set deceleration to 200000 counts/second^2.
s r0x24 23	ok	Enable the amplifier in Digital Input Position Mode.
The controller monitors commanded and actual motor position.		
g r0x3d	v 4000	Reads commanded position from the amplifier. Example displays a returned value equal to 4000 counts.
g r0x32	v 2012	Reads actual motor position from the amplifier. Example displays a returned value equal to 2012 counts.
The controller changes the scaling variable.		
s r0xa9 0x00010001	ok	Set scaling factor to 1 output count per input pulse.
The controller disables the amplifier.		
s r0x24 0	ok	Disable the amplifier.

3.4.6: Quadrature Mode

In the Quadrature Position Mode, the axis position is commanded by a master encoder with its A and B channels applied to the amplifier's digital inputs.

The scaling factor sets the ratio of position command, in counts, for each count of the master encoder. The scaling factor is stored in 0xa9 as two 16 bit words. Word 1 stores the numerator or number of position counts. Word 2 stores the denominator or the number of input counts. Example: To set a ratio of 10 counts of position change for every input count, the ratio would be 10/1. To make sending the data easier, the ratio should be converted to its hex equivalent (0x000a/0x0001). The two words can now be combined and sent to the amplifier by sending the command *s r0xa9 0x000a0001*.

To invert the direction, the numerator should be set to a negative value.
Example: Changing direction of the previous example would require a ratio of -10/1. Using the 2's complement method, -10 is represented as 0xfff6 in hex format. The ratio in hex format would now be 0xfff6/0x001. Combining these words, the command to be sent would be
s r0xa9 0xfff60001.

Quadrature Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state: 0 = Disabled. 23 = Digital Input Position Mode, Servo. 33 = Digital Input Position Mode, Stepper.
0xa8	R F	Digital Command Configuration: 512 = Quadrature Mode.
0xa9	R F	Input / Output Ratio.
0xcb	R F	Maximum velocity. Units: 0.1 counts/second.
0xcc	R F	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	R F	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	R F	Abort deceleration rate. Units: 10 counts/second^2.

Quadrature Mode Example

The controller sets the move parameters, enables the amplifier in the Quadrature Position Mode, and monitors commanded and actual position.

Command	Response	Comment
s r0xa8 512	ok	Configure the digital inputs to quadrature position mode.
s r0xa9 0x00010001	ok	Set scaling factor to 1 output counts per input pulse.
s r0xcb 70000	ok	Set velocity to 7000 counts/second
s r0xcc 20000	ok	Set acceleration to 200000 counts/second^2
s r0xcd 20000	ok	Set deceleration to 200000 counts/second^2
s r0x24 23	ok	Enable the amplifier in Digital Input Position Mode.
The controller monitors commanded and actual motor position.		
g r0x3d	v 4000	Reads commanded position from the amplifier. Example displays a returned value equal to 4000 counts.
g r0x32	v 2012	Reads actual motor position from the amplifier. Example displays a returned value equal to 2012 counts.
The controller disables the amplifier.		
s r0x24 0	ok	Disable the amplifier.

3.4.7: Homing Mode

Homing sequences can be performed using the *t 2* command when the amplifier is in Programmed Position Mode (servo or stepper). In most applications the homing sequence is configured using CME 2 and not changed during operation.

Homing Mode Variables

Variable ID	Bank	Description
0x24	R F	Desired state: 0 = Disabled. 21 = Programmed Position Mode, Servo. 31 = Programmed Position Mode, Stepper. This is the required mode for homing.
0xc2	R F	Homing Method. See table below for values.
0xc3	R F	Fast Velocity Units: 0.1 counts/second.
0xc4	R F	Slow Velocity Units: 0.1 counts/second.
0xc5	R F	Acceleration / Deceleration Units: 10 counts/second^2.
0xc6	R F	Home Offset Units: counts.
0xc7	R F	Current Limit Units: 0.01 Amps.
0xbf	R F	Current Delay Time Units: milliseconds.
0xb8	R F	Positive Software Limit Units: counts.
0xb9	R F	Negative Software Limit Units: counts.

Homing Example

The controller modifies the homing parameters, enables the amplifier in the Programmed Position Mode, initiates a homing sequence and then monitors homing status.

Command	Response	Comment
Setting the homing parameters is only required if the home sequence stored in flash memory is not satisfactory.		
s r0xc2 544	ok	Sets the homing method to use the next index pulse as home.
s r0xc4 40000	ok	Sets the slow velocity to 4000 counts/second.
s r0xc6 1000	ok	Sets the home offset to 1000 counts.
s r0x24 21	ok	Enables the amplifier in programmed position mode.
t 2	ok	Starts the homing sequence.
The controller monitors the trajectory status register to determine when the homing sequence is complete.		
g r0xc9	v 20480	Controller checks bit 12 of the trajectory status register to determine if the axis was homed successfully. Example displays a status register value of 20480. Decoded, this value shows that bit 12 is set meaning the axis is referenced.

Homing Methods (0xc2)

For a full description of the methods listed below, see *Homing Method Descriptions* (p. 59).

Method	Start Direction	Value
Set Current Position as Home	-----	512
Next Index	Positive	544
	Negative	560
Limit Switch	Positive	513
	Negative	529
Limit Switch Out to Index	Positive	545
	Negative	561
Home Switch	Positive	514
	Negative	530
Home Switch Out to Index	Positive	546
	Negative	562
Home Switch In to Index	Positive	610
	Negative	626
Hard Stop	Positive	516
	Negative	532
Hard Stop Out to Index	Positive	548
	Negative	564
Lower Home	Positive	771
	Negative	787
Upper Home	Positive	515
	Negative	531
Lower Home Outside Index	Positive	803
	Negative	819
Lower Home Inside Index	Positive	867
	Negative	883
Upper Home Outside Index	Positive	547
	Negative	563
Upper Home Inside Index	Positive	611
	Negative	627

CHAPTER
4: OPERATION

This chapter describes the variables involved in basic operation of the amplifier.
Contents include:

4.1: Setting the Baud Rate

Variable 0x90 (R) controls the amplifier's serial port baud rate. To change the baud rate, write the new value to 0x90. For instance, to change the value to 19200 send: s r0x90 19200. The amplifier will respond with an "ok" if the command is successful but it will be sent at the new baud rate.

After the carriage return of the s command, no other characters should be sent at 9600 (by default, some programs automatically append a line feed). If more characters are sent at 9600, they may be misinterpreted as a break command and cause the amplifier to change back to 9600 baud. There should also be a delay of 100 mSec minimum before characters at the new baud rate are sent to the amplifier.

When reading variable 0x90, note that the value received may not be the exact value set. This is because the amplifier sets the baud rate as close to the requested baud rate as possible given the internal clock frequencies of the amplifier's microprocessor.

4.2: Setting Limits and Gains

This section describes the variables used to set control loop limits and gains.

4.2.1: Current Loop Limits and Gains

Current Loop Limits Variables

Variable ID	Bank	Description
0x21	R F	Peak current limit. Units: 0.01 A.
0x23	R F	I^2T time limit. Units: mS.
0x22	R F	Continuous current limit. Units: 0.01 A.
0xae	R F	Current loop offset. Units: 0.01 A.

Current Loop Gains Variables

0x00	R F	Current loop proportional gain (Cp).
0x01	R F	Current loop integral gain (Ci).

4.2.2: Velocity Loop Limits and Gains

Velocity Loop Limits Variables

Variable ID	Bank	Description
0x3a	R F	Velocity loop velocity limit. Units: 0.01 counts/second.
0x36	R F	Velocity loop acceleration limit. Units: 1000 counts/second^2.
0x37	R F	Velocity loop deceleration Limit. Units: 1000 counts/second^2.
0xcf	R F	Fast Stop Ramp. Units: 10 counts/second^2.

Velocity Loop Gains Variables

Variable ID	Bank	Description
0x27	R F	Velocity loop proportional gain (Vp).
0x28	R F	Velocity loop integral gain (Vi).

4.2.3: Position Loop Gains

Position loop limits are described in *Position Mode* (p. 29).

Position Loop Gains Variables

Variable ID	Bank	Description
0x30	R F	Pp - Position loop proportional gain.
0x33	R F	Vff - Velocity feed forward.
0x34	R F	Aff - Acceleration feed forward.
0xe3	R F	Position loop gain multiplier. 100 equals a factor of 1.

4.2.4: Filters

Velocity Loop Filters Variables

Variable ID	Bank	Description
0x6b	R F	Velocity loop command filter co-efficients.
0x5f	R F	Velocity loop output filter co-efficients.

Velocity Loops Filters Usage Notes

The velocity loop command and output filters should be set up using CME 2. If it is required that the filters be changed during operation, the following procedure should be used to determine the new filter co-efficients.

1 Set the filter up using CME 2.
   1 On the CME 2 Main screen, click **V Loop**.
   2 On the Velocity Loop screen, click **Command Filter** or **Output Filter** as desired.
   3 On the Filter screen, choose the filter type, set the parameters, click **Apply** and then click **Close**.

2 Use the CME 2 ASCII command line tool (**Tools->ASCII Command Line**) to read the updated variable. For instance, to read the command filter variable:

Command	g r0x6B
Response	v -7936 200 0 775 1550 775 -12774 32763 5813

3 Write program instructions to update the appropriate variable with those values. For instance, to write the command filter variable:

Command	s r0x6B -7936 200 0 775 1550 775 -12774 32763 5813
Response	ok

4.3: Monitoring Status

Status Register Variable (0xa0)

The status register variable (0xa0) provides amplifier status information.
0xa0 is read-only, and available in RAM only (not Flash). Bit mapped values described below:

Bits	Description
0	Short circuit detected.
1	Amplifier over temperature.
2	Over voltage.
3	Under voltage.
4	Motor temperature sensor active.
5	Feedback error.
6	Motor phasing error.
7	Current output limited.
8	Voltage output limited.
9	Positive limit switch active.
10	Negative limit switch active.
11	Enable input not active.
12	Amplifier is disabled by software.
13	Trying to stop motor.
14	Motor brake activated.
15	PWM outputs disabled.
16	Positive software limit condition.
17	Negative software limit condition.
18	Tracking error.
19	Tracking warning.
20	Amplifier has been reset.
21	Position has wrapped. The Position variable cannot increase indefinitely. After reaching a certain value the variable rolls back. This type of counting is called position wrapping or modulo count. Note that this bit is only active as the position wraps.
22	Amplifier fault. An amplifier fault that was configured as latching has occurred. For information on latching faults, see the CME 2 User Guide.
23	Velocity limit has been reached.
24	Acceleration limit has been reached.
25	Position outside of tracking window.
26	Home switch is active.
27	Set if trajectory is running or motor has not yet settled into position (within Position Tracking Error Limit) at the end of the move. Once the position has settled, the in motion bit won't be set until the next move starts.
28	Velocity window. Set if the absolute velocity error exceeds the velocity window value.
29	Phase not yet initialized. If the amplifier is phasing with no Halls, this bit is set until the amplifier has initialized its phase.
30	Command fault. PWM or other command signal not present.
31	Not defined.

Trajectory Register Variable (0xc9)

The trajectory register variable (0xc9) provides trajectory generator status information.
0xc9 is read-only, and available in RAM only (not Flash). Bit mapped values described below:

Bit	Description
0-8, 10	Reserved for future use.
9	Cam table underflow.
11	Homing error. If set, an error occurred in the last home attempt. Cleared by a home command.
12	Referenced. Set when a homing command has been successfully executed. Cleared by a home command.
13	Homing. If set, the amplifier is running a home command.
14	Set when a move is aborted. Cleared at the start of the next move.
15	In-Motion Bit. If set, the trajectory generator is presently generating a profile.

Fault Register Variable (0xa4)

The fault register variable (0xa4) shows latching faults that have occurred.
0xa4 is available in RAM only (not Flash).

Bit mapped values described below:

Bit	Fault Description
0	Data flash CRC failure. This fault is considered fatal and cannot be cleared.
1	Amplifier internal error. This fault is considered fatal and cannot be cleared.
2	Short circuit.
3	Amplifier over temperature.
4	Motor over temperature.
5	Over voltage.
6	Under voltage.
7	Feedback fault.
8	Phasing error.
9	Following error.
10	Over Current (Latched),
11	FPGA failure. This fault is considered fatal and cannot usually be cleared. If this fault occurred after a firmware download, repeating the download may clear this fault.
12	Command input lost.
13-31	Reserved.

Note that when a latching fault has occurred, bit 22 of the status register (0xa0) is set.

To clear a fault condition, write a 1 to the associated bit of the fault register (0xa4).

4.4: Reading Run Time Variables

This section describes the variables used to monitor run time conditions.

Current Loop Run Time Variables

Variable ID	Bank	Description
0x15	R	Commanded current. Units: 0.01 A.
0x0c	R	Actual current. Units: 0.01 A.
0x25	R	Limited current. Units: 0.01 A.

Velocity Loop Run Time Variables

Variable ID	Bank	Description
0x2c	R	Commanded velocity. Units: 0.1 counts/second.
0x29	R	Limited velocity. Units: 0.1 counts/second.
0x18	R	Actual motor velocity. Units: 0.1 counts/second.
0x5e	R	Actual load velocity. Units: 0.1 counts/second.
0x2a	R	Velocity loop error. Units: 0.1 counts/second.

Position Loop Run Time Variables

Variable ID	Bank	Description
0x32	R	Motor position. Units: counts.
0x17	R	Load position. Units: counts.
0x35	R	Following Error. Units: counts.

Position Loop Inputs from the Trajectory Generator (Variables)

Variable ID	Bank	Description
0x3d	R	Commanded position. Units: counts.
0x2d	R	Limited position. Units: counts.
0x3B	R	Profile velocity. Units: 0.1 counts/second.
0x3C	R	Profile acceleration. Units: 10 counts/second^2.

Miscellaneous System Variables

Variable ID	Bank	Description
0x1d	R	Analog input voltage. Units: mV.
0x1b	R	Sin input voltage. Units: mV.
0x1c	R	Cos input voltage. Units: mV.
0x1e	R	Bus voltage. Units: 100 mV.
0x20	R	Amplifier temperature. Units: degrees C.
0xb0	R	Phase angle. Units: degrees.

4.5: Reading Digital Inputs

Input States Variable (0xa6)

The high/low states of the amplifier's programmable digital inputs can be read using variable 0xa6. Each bit represents an input number as shown below. If an input is high, the corresponding bit is set to 1. If the input is low, the corresponding bit is set to 0.

For instance, if the value of 0xa6 is 33, the binary equivalent is 100001, showing that IN1 and IN6 are high and the other inputs are low.

0xa6 is read-only, and available in RAM only (not Flash). Bit mapped values described below.

NOTE: The number of programmable digital inputs varies depending on amplifier model. See the amplifier documentation.

Bit	Input
0	Digital Input 1
1	Digital Input 2
2	Digital Input 3
3	Digital Input 4
4	Digital Input 5
5	Digital Input 6
6	Digital Input 7
7	Digital Input 8
8	Digital Input 8
9	Digital Input 10
10	Digital Input 11
11	Digital Input 12
12	Digital Input 13
13	Digital Input 14
14	Digital Input 15
15	Digital Input 16

4.6: Reading/Setting Digital Outputs

The amplifiers digital outputs can be programmed by CME 2 to reflect the state of any one or more of the amplifier's event status register bits. The outputs can also be configured so their state can be set by the controller program.

The external controller, through the Output State variable, can set an output inactive or active. The actual level of the output pin however is determined by the Output Configuration variable. This variable sets the actual output pin to be high or low when active. When the amplifier powers up or is reset, all outputs are initially inactive. To ensure that outputs are high, or off, after power up or reset, they should be configured as active low.

Configuring Outputs (0x70 – 0x77)

Before a controller program can set an output pin's active/inactive state, the output must be configured for program control. This is done by setting the appropriate bits in the output's configuration variable.

The output configuration variables start with 0x70 for Output 1 and run to 0x77 for Output 8, as described below. These variables require two values be sent with Set (s) command.

NOTE: The number of programmable digital inputs varies depending on amplifier mode. See the amplifier documentation.

Variable ID	Memory Bank	Description
0x70	RF	Output 1 Configuration. 258 0 = Program Control, Active High 2 0 = Program Control, Active Low
0x71	RF	Output 2 Configuration. Same as Output 1
0x72 – 0x77	RF	Output 3 though 8 Configuration. Same as Output 1

Setting Output States (0xab)

Writing the variable 0xab sets the active/inactive states of digital outputs that have been configured for program control. Each bit represents an output number as shown below. A bit value of 1 corresponds to an active output. A bit value of 0 corresponds to an inactive output.

Writing a 1 or 0 to an output that has not been configured for program control will have no effect on the output.

NOTE: The number of programmable digital outputs varies depending on amplifier model. See the amplifier documentation.

Bit	Output
0	Digital Output 1
1	Digital Output 2
…	
7	Digital Output 8

Reading Output States (0xab)

Reading 0xab gets the active/inactive states of all the amplifier's digital outputs, including those which are not set to program control.

Reading/Setting Output Example

The controller configures 2 outputs for program control, reads the state of the outputs, and then sets an output low.

Command	Response	Comment
s r0x72 258 0	ok	Configures output 3 to program control, active low.
s r0x73 258 0	ok	Configures output 4 to program control, active low.
g r0xab	v 10	Reads the state of the outputs. Example returns a value of 10. Converting this value to binary equals 1010 which indicates outputs 2 and 4 are active.
s r0xab 4	ok	4 converted to binary equals 0100. This value will set Output 4 inactive and Output 3 active. Outputs 4 and 3 have been programmed active low so Output 4 will be high and 3 will be low. Since Outputs 1 and 2 are not under program control, they will not change state.

APPENDIX

A: QUICK REFERENCE TO THE VARIABLES

This chapter provides quick reference to the variables described in this manual.

Contents include:

A.1: Variables by Function

Programmed Current Mode Variables	
0x02	Programmed current value. Units: 0.0.
0x6a	Current ramp rate. Units: mA/second.
Analog Current Mode Variables	
0x19	Analog input scaling factor. Units: 0.01 A.
0x26	Analog input dead band. Units: mV.
0x1a	Analog input offset. Units: mV.
PWM Current Mode Variables	
0xa9	Digital input scaling factor. Units: 0.01 A.
0xa8	Digital input command configuration.
Programmed Velocity Mode Variables	
0x2f	Programmed velocity command. Units: 0.1 counts/second.
0x36	Velocity acceleration limit. Units: 1000 counts/second^2
0x37	Velocity deceleration limit. Units: 1000 counts/second^2
0x39	Fast stop ramp. Units: 1000 counts/second^2
Analog Velocity Mode Variables	
0x19	Analog input scaling factor. Units: 0.1 counts/second
0x26	Analog input dead band. Units: mV.
0x1a	Analog input offset. Units: mV.
0x36	Velocity acceleration limit. Units: 1000 counts/second^2
0x37	Velocity deceleration limit. Units: 1000 counts/second^2
0x39	Fast stop ramp. Units: 1000 counts/second^2
PWM Velocity Mode Variables	
0xa9	Scaling Factor. Units: 0.1 counts/second.
0x36	Velocity acceleration limit. Units: 1000 counts/second^2
0x37	Velocity deceleration limit. Units: 1000 counts/second^2
0x39	Fast stop ramp. Units: 1000 counts/second^2
0xa8	Digital input command configuration.

Programmed Position Mode Variables	
0xc8	Profile type: 0 = Absolute move, trapezoidal profile. 1 = Absolute move, S-curve profile. 256 = Relative move, trapezoidal profile. 257 = Relative move, S-curve profile. 2 = Velocity profile.
0xca	Position command: Relative move = the distance of the move. Absolute move = the target position of the move. Velocity move = 1 for positive direction, -1 for negative direction. Units: counts.
0xcb	Maximum velocity. Units: 0.1 counts/second.
0xcc	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	Maximum deceleration rate. Units: 10 counts/second^2.
0xce	Maximum jerk rate. Units: 100 counts/ second^3.
0xcf	Abort deceleration rate. Units: 10 counts/second^2.
Analog Position Mode Variables	
0x19	Analog input scaling factor. Units: counts.
0x26	Dead band. Units: mV.
0xcb	Maximum velocity. Units: 0.1 counts/second.
0xcc	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	Abort deceleration rate. Units: 10 counts/second^2.
Pulse and Direction Mode Variables	
0xa8	Digital Command Configuration. Pulse and Direction: 4096 = Increment position on rising edge. 0 = Increment position on falling edge.
0xa9	Scaling factor. Output counts/Input pulses.
0xcb	Maximum velocity. Units: 0.1 counts/second.
0xcc	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	Abort deceleration rate. Units: 10 counts/second^2.
Pulse Up/Down Mode Variables	
0xa8	Digital Command Configuration. Pulse Up/Down Mode: 4352 = Increment position on rising edge. 256 = Increment position on falling edge.
0xa9	Input / Output Ratio.
0xcb	Maximum velocity. Units: 0.1 counts/second.
0xcc	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	Abort deceleration rate. Units: 10 counts/second^2.

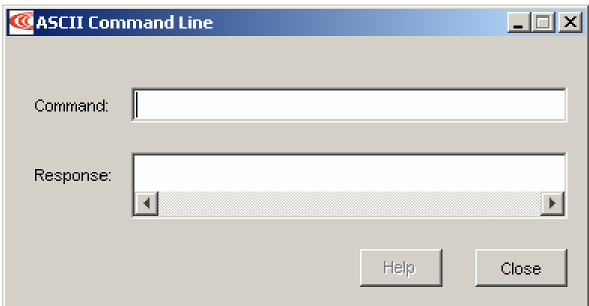
Quadrature Mode Variables	
0xa8	Digital Command Configuration. 512 = Quadrature Mode.
0xa9	Input / Output Ratio.
0xcb	Maximum velocity. Units: 0.1 counts/second.
0xcc	Maximum acceleration rate. Units: 10 counts/second^2.
0xcd	Maximum deceleration rate. Units: 10 counts/second^2.
0xcf	Abort deceleration rate. Units: 10 counts/second^2.
Homing Mode Variables	
0xc2	Homing Method. See table below for values.
0xc3	Fast Velocity. Units: counts/second
0xc4	Slow Velocity. Units: counts/second
0xc5	Acceleration / Deceleration. Units: 10 counts/second^2.
0xc6	Home Offset. Units: counts.
0xc7	Current Limit. Units: 0.01 A.
0xbf	Current Delay Time. Units: milliseconds.
0xb8	Positive Software Limit. Units: counts.
0xb9	Negative Software Limit. Units: counts.
Current Loop Limits Variables	
0x21	Peak current limit. Units: 0.01 A.
0x23	I^2T time limit. Units: mS.
0x22	Continuous current limit. Units: 0.01 A.
0xae	Current loop offset. Units: 0.01 A.
Current Loop Gains Variables	
0x00	Current loop proportional gain (Cp).
0x01	Current loop integral gain (Ci).
Velocity Loop Limits Variables	
0x3a	Velocity loop velocity limit. Units: 0.01 counts/second.
0x36	Velocity loop acceleration limit. Units: 1000 counts/second^2.
0x37	Velocity loop deceleration Limit. Units: 1000 counts/second^2.
0xcf	Fast Stop Ramp. Units: 10 counts/second^2.
Velocity Loop Gains Variables	
0x27	Velocity loop proportional gain (Vp).
0x28	Velocity loop integral gain (Vi).
Velocity Loop Filters Variables	
0x6b	Velocity loop command filter co-efficients.
0x5f	Velocity loop output filter co-efficients.
Position Loop Gains Variables	
0x30	Pp - Position loop proportional gain.
0x33	Vff - Velocity feed forward.
0x34	Aff - Acceleration feed forward.
0xe3	Position loop gain multiplier.

Current Loop Run Time Variables	
0x15	Commanded current. Units: 0.01 A.
0x0c	Actual current. Units: 0.01 A.
0x25	Limited current. Units: 0.01 A.
Velocity Loop Run Time Variables	
0x2c	Commanded velocity. Units: 0.1 counts/second.
0xcb	Profile velocity. Units: 0.1 counts/second.
0x29	Limited velocity. Units: 0.1 counts/second.
0x18	Motor velocity. Units: 0.1 counts/second.
0x5e	Load velocity. Units: 0.1 counts/second.
0x2a	Velocity loop error.
Position Loop Run Time Variables	
0x3d	Commanded position. Units: counts.
0x2d	Limited position. Units: counts.
0x32	Motor position. Units: counts.
0x17	Load position. Units: counts.
0x35	Following Error. Units: counts.
Position Loop Inputs from the Trajectory Generator (Variables)	
0x3b	Profile velocity. Units: 0.1 counts/second.
0x3c	Profile acceleration. Units: 10 counts/second^2.
0x2d	Limited position. Units: counts.
Miscellaneous System Variables	
0x1d	Analog input voltage. Units: mV.
0x1b	Sin input voltage.
0x1c	Cos input voltage.
0x1E	Bus voltage. High voltage A/D reading. Units: 100 mV.
0x20	Amplifier temperature. Units: degrees C.
0xb0	Phase angle. Units: degrees.
0x90	Baud rate
Inputs and Outputs	
0xa6	Read input states
0xab	Read/Write output states.
0x70 thru 0x77	Configure outputs.
Status and State Variables	
0xa0	Status Register.
0xc9	Trajectory Register.
0xa4	Fault Register.
0x24	Amplifier desired state.

APPENDIX B: CME 2 ASCII COMMAND LINE TOOL

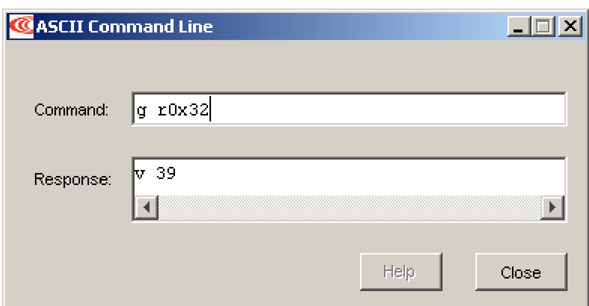
The CME 2 ASCII Command Line tool lets users send individual ASCII commands to amplifiers.

- From the CME 2 Main screen, choose **Tools->ASCII Command Line** to open the ASCII Command Line tool:

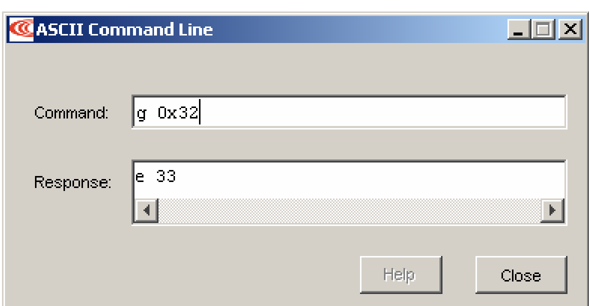


- Enter an ASCII command in the *Command* field.
- Press the Enter key to send the command to the amplifier. Observe the response in the Response field.

  If a value is returned, it is preceded by the letter "v." In the following example, the get command was used to retrieve the RAM value of variable 0x32 (actual position).



  An error code is preceded by the letter "e." In the following example, the get command was entered without the required memory bank designation, resulting in an ASCII command parsing error (error code 33). See *Error Codes* (p. 77).



TIP: To view an error definition, hold the mouse pointer over the error number.

APPENDIX

C: HOMING METHOD DESCRIPTIONS

This appendix describes the homing methods that can be chosen using the homing method variable (0xc2) as described in *Homing Mode*, p. 38.

Contents include:

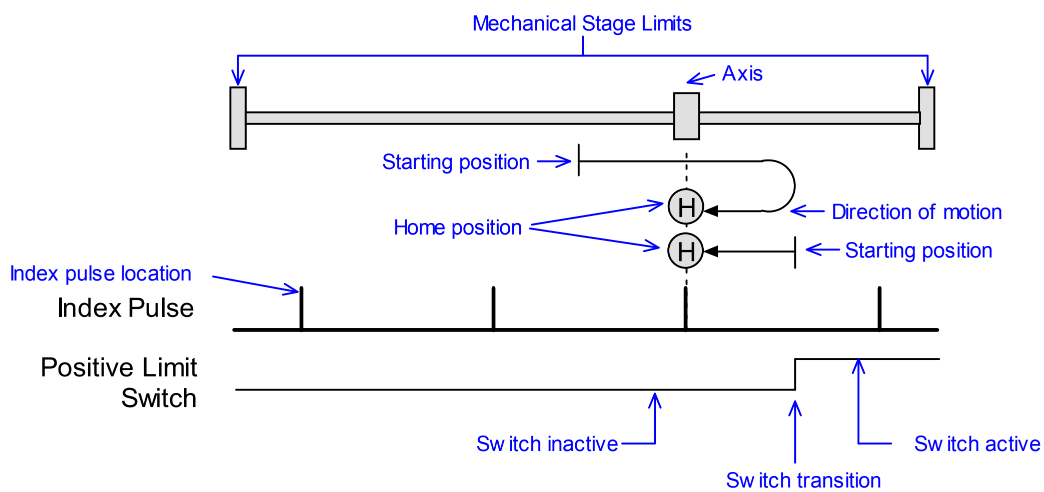
C.1: Homing Methods Overview

There are several homing methods. Each method establishes the:

- Home reference (limit or home switch transition or encoder index pulse)
- Direction of motion and, where appropriate, the relationship of the index pulse to limit or home switches.

C.2: Legend to Homing Method Descriptions

As highlighted in the example below, each homing method diagram shows the starting position on a mechanical stage. The arrow line indicates direction of motion, and the circled H indicates the home position. Solid line stems on the index pulse line indicate index pulse locations. Longer dashed lines overlay these stems as a visual aid. Finally, the relevant limit switch is represented, showing the active and inactive zones and transition.



Note that in the homing method descriptions, negative motion is leftward and positive motion is rightward.

C.3: Homing Method Descriptions

C.3.1: Set current position as home

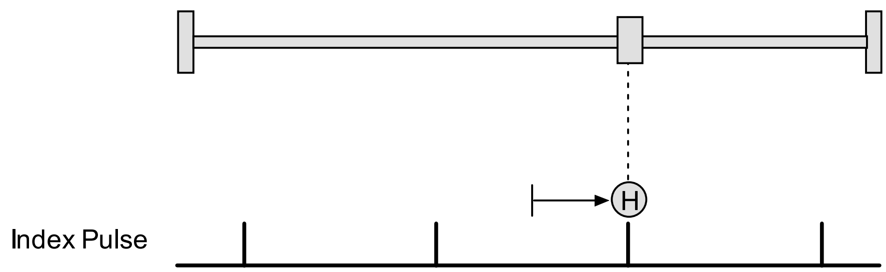
Direction of Motion: N/A (0xc2 = 512)

The current position is the home position.

C.3.2: Next Index

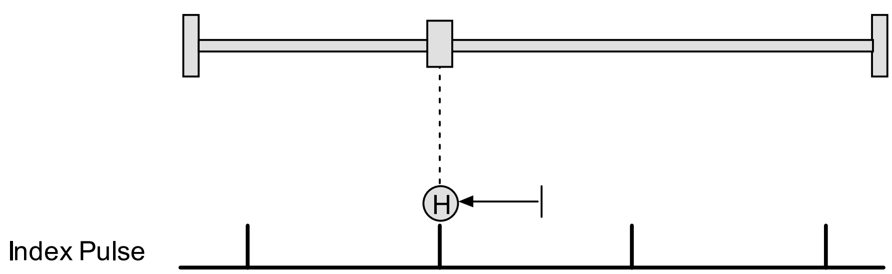
Direction of Motion: Positive (0xc2 =544)

Home is the first index pulse found in the positive direction. Direction of motion is positive. If a positive limit switch is activated before the index pulse, an error is generated.



Direction of Motion: Negative (0xc2 =560)

Home is the first index pulse found in negative direction. Direction of motion is negative. If a negative limit switch is activated before the index pulse, an error is generated.

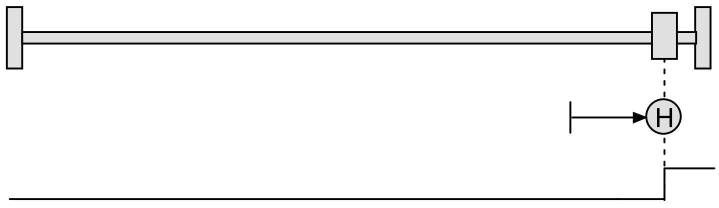


C.3.3: Limit Switch

Direction of Motion: Positive (0xc2 =513)

Home is the transition of the positive limit switch. Initial direction of motion is positive if the positive limit switch is inactive.

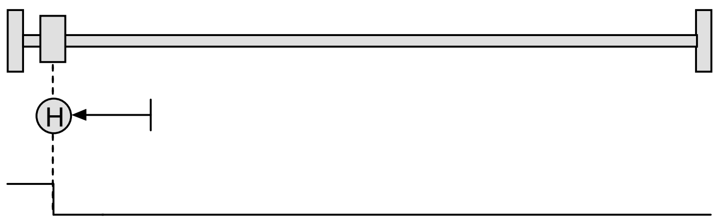
Positive Limit Switch



Direction of Motion: Negative (0xc2 =529)

Home is the transition of negative limit switch. Initial direction of motion is negative if the negative limit switch is inactive.

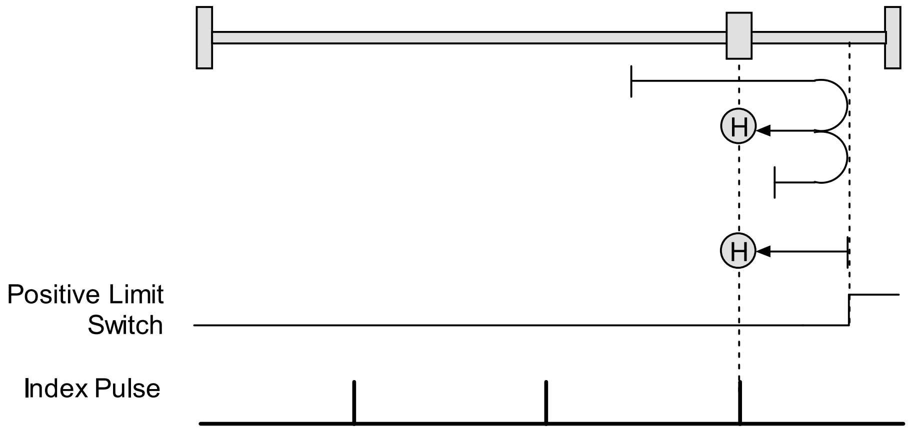
Negative Limit Switch



C.3.4: Limit Switch Out to Index

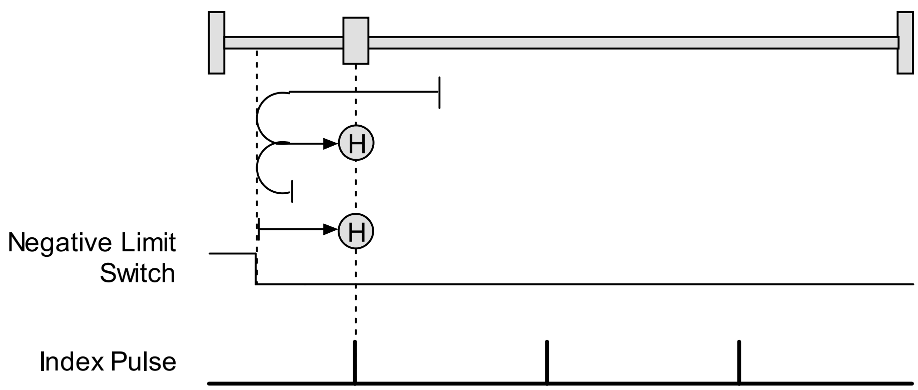
Direction of Motion: Positive (0xc2 =545)

Home is the first index pulse to the negative side of the positive limit switch transition. Initial direction of motion is positive if the positive limit switch is inactive (shown here as low).



Direction of Motion: Negative (0xc2 =561)

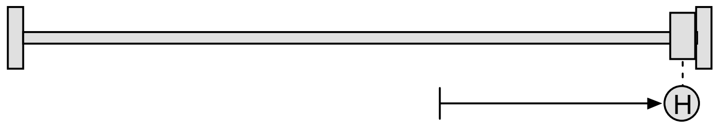
Home is the first index pulse to the positive side of the negative limit switch transition. Initial direction of motion is negative if the negative limit switch is inactive (shown here as low).



C.3.5: Hardstop

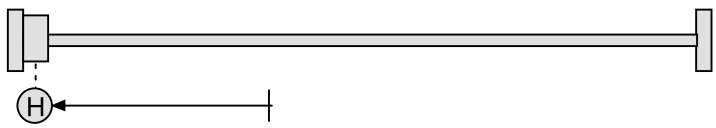
Direction of Motion: Positive (0xc2 =516)

Home is the positive hard stop. Direction of motion is positive. The hard stop is reached when the amplifier outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a positive limit switch is activated before the hard stop, an error is generated.



Direction of Motion: Negative (0xc2 =532)

Home is the negative hard stop. Direction of motion is negative. The hard stop is reached when the amplifier outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a negative limit switch is activated before the hard stop, an error is generated.



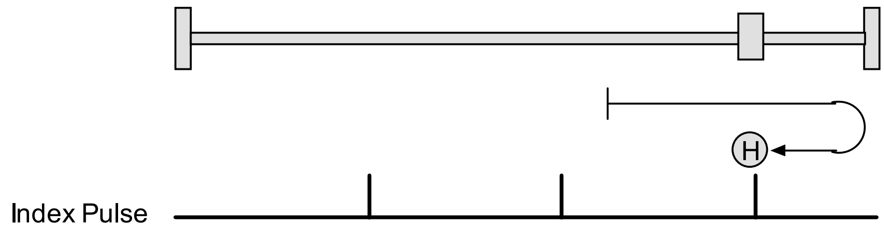
Hardstop Methods in Stepper Mode

In Stepnet amplifiers operating in stepper mode with an encoder, the hard stop is reached when the following error is exceeded. When using hardstop methods in stepper mode, do not disable following error.

C.3.6: Hardstop Out to Index

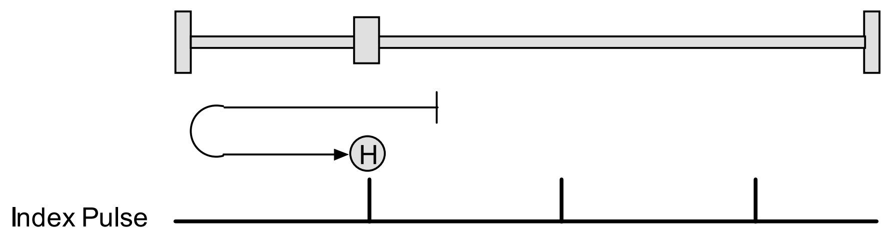
Direction of Motion: Positive (0xc2 =548)

Home is the first index pulse on the negative side of the positive hard stop. Initial direction of motion is positive. The hard stop is reached when the amplifier outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a positive limit switch is activated before the hard stop, an error is generated.



Direction of Motion: Negative (0xc2 =564)

Home is the first index pulse on the positive side of the negative hard stop. Initial direction of motion is negative. The hard stop is reached when the amplifier outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a negative limit switch is activated before the hard stop, an error is generated.



Hardstop Out to Index Methods in Stepper Mode

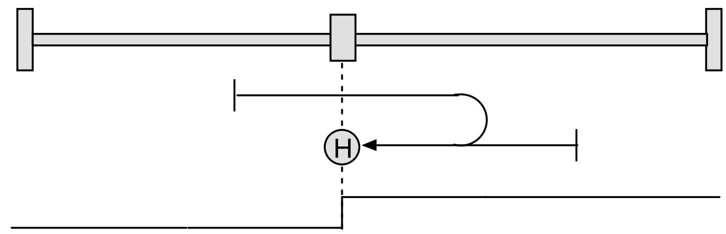
In Stepnet amplifiers operating in stepper mode with an encoder, the hard stop is reached when the following error is exceeded. When using hardstop methods in stepper mode, do not disable following error.

C.3.7: Home Switch

Direction of Motion: Positive (0xc2 =514)

Home is the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.

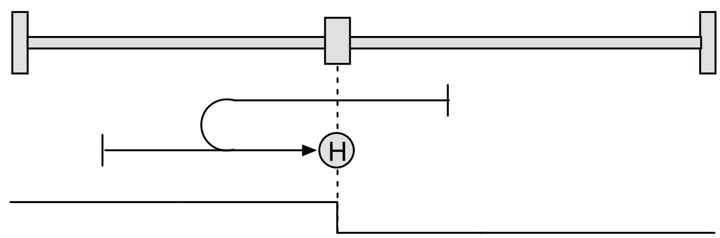
Home Switch



Direction of Motion: Negative (0xc2 =530)

Home is the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.

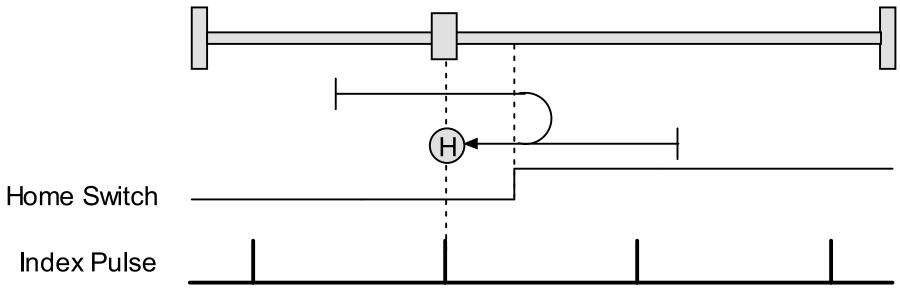
Home Switch



C.3.8: Home Switch Out to Index

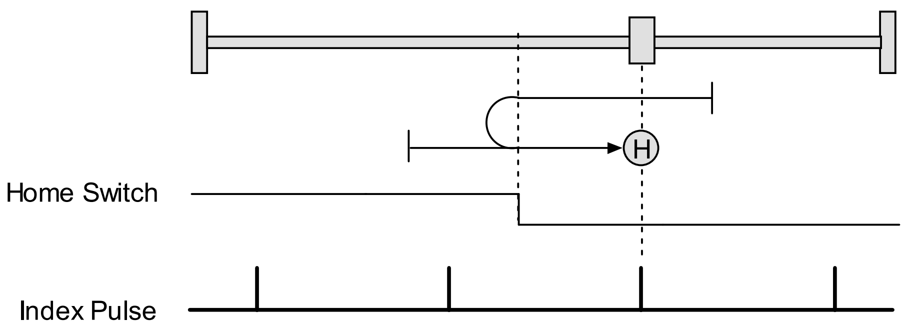
Direction of Motion: Positive (0xc2 =546)

Home is the first index pulse to the negative side of the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



Direction of Motion: Negative (0xc2 =562)

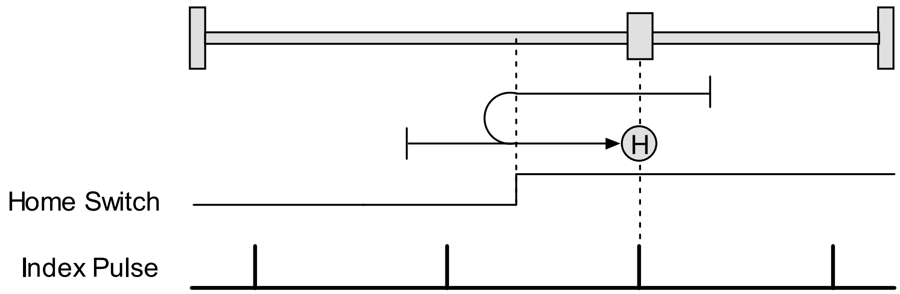
Home is the first index pulse to the positive side of the home switch transition.
Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



C.3.9: Home Switch In to Index

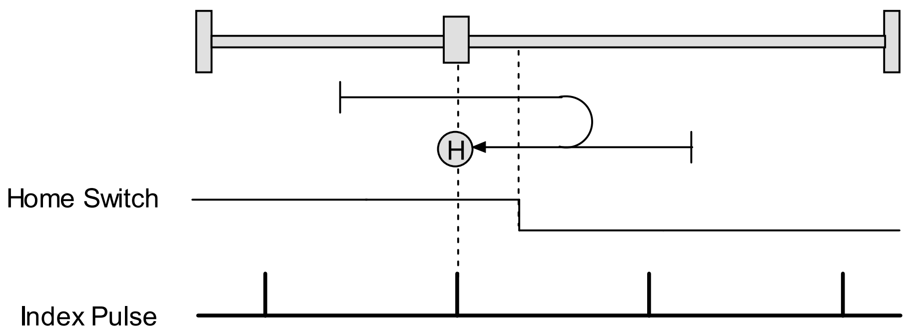
Direction of Motion: Positive (0xc2 =610)

Home is the first index pulse to the positive side of the home switch transition. Initial direction of motion is positive if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



Direction of Motion: Negative (0xc2 =626)

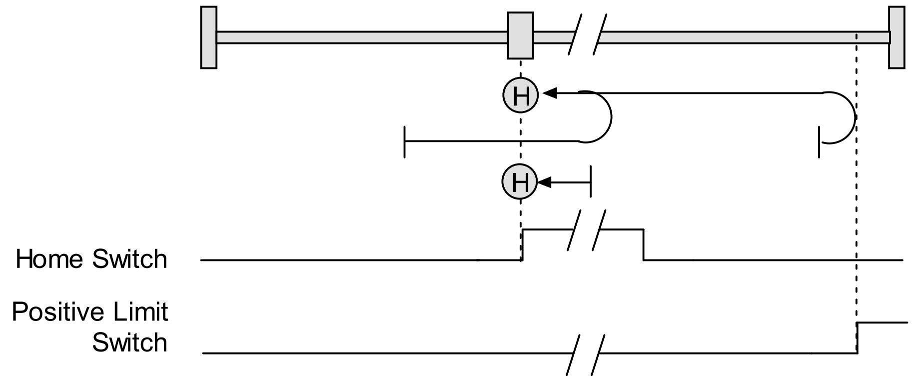
Home is the first index pulse to the negative side of the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



C.3.10: Lower Home

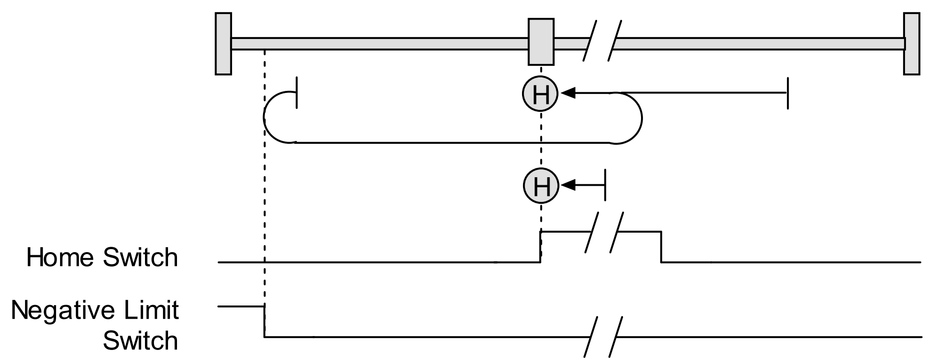
Direction of Motion: Positive (0xc2 =771)

Home is the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. Motion will reverse if a positive limit switch is activated before the home switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Direction of Motion: Negative (0xc2 =787)

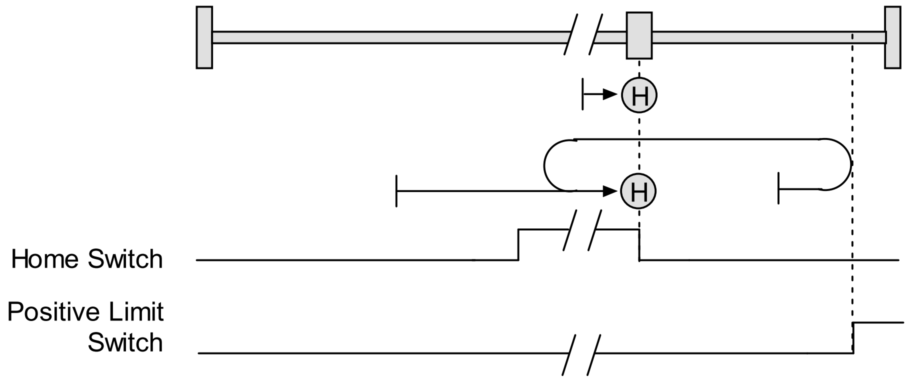
Home is the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home switch, an error is generated.



C.3.11: Upper Home

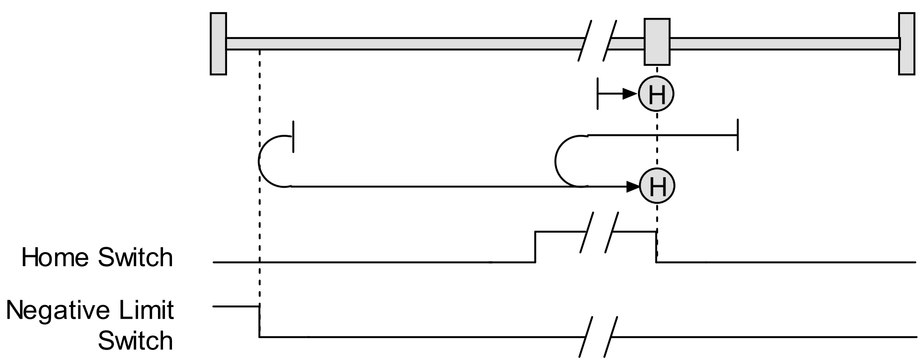
Direction of Motion: Positive (0xc2 =515)

Home is the positive edge of a momentary home switch. Initial direction of motion is positive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Direction of Motion: Negative (0xc2 =531)

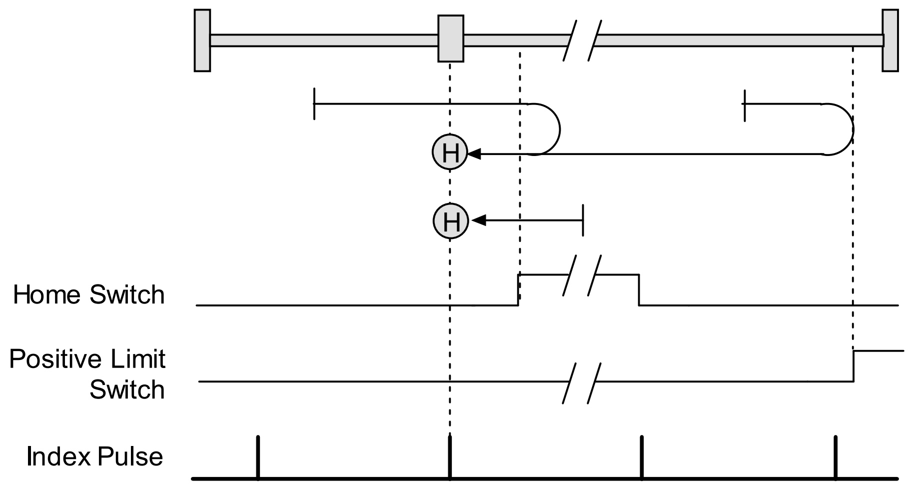
Home is the positive edge of momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home switch, an error is generated.



C.3.12: Lower Home Outside Index

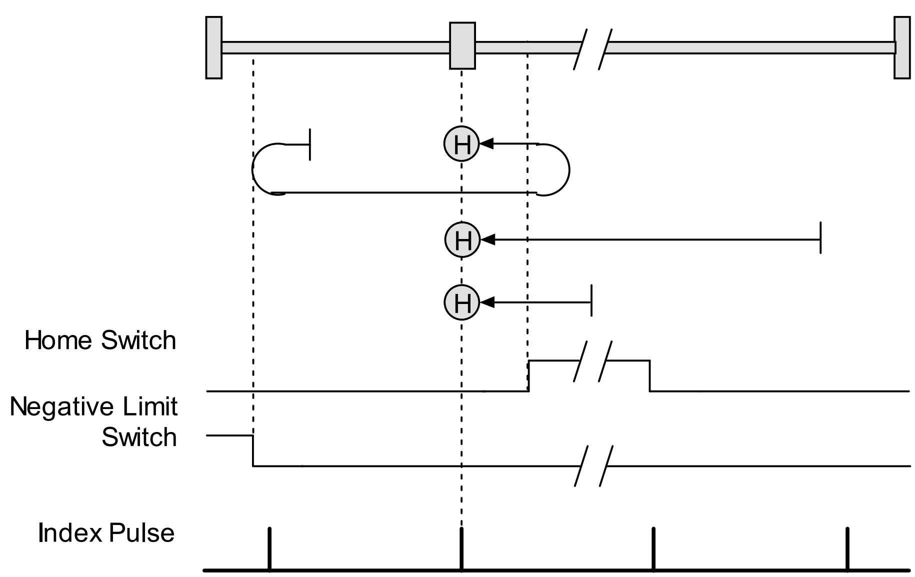
Direction of Motion: Positive (0xc2 =803)

Home is the first index pulse on the negative side of the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Direction of Motion: Negative (0xc2 =819)

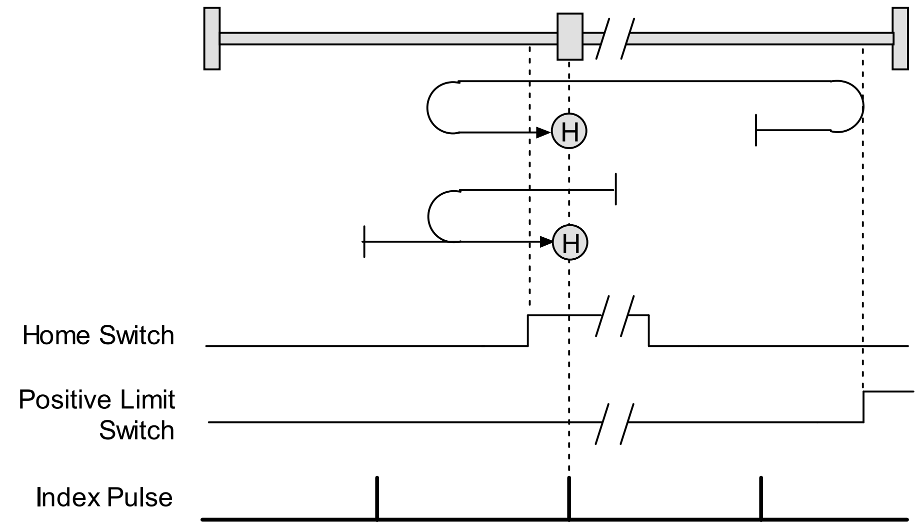
Home is the first index pulse on the negative side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



C.3.13: Lower Home Inside Index

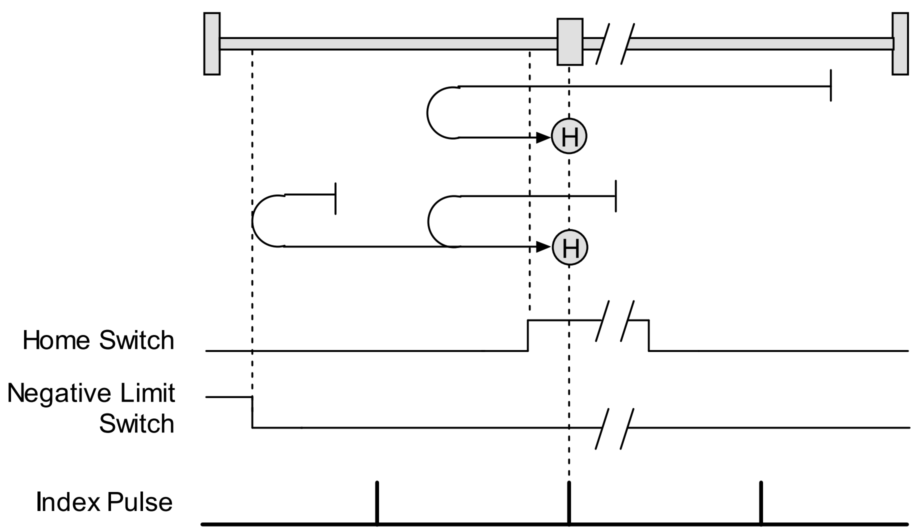
Direction of Motion: Positive (0xc2 =867)

Home is the first index pulse on the positive side of the negative edge of a momentary home switch. Initial direction of motion is positive if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Direction of Motion: Negative (0xc2 =883)

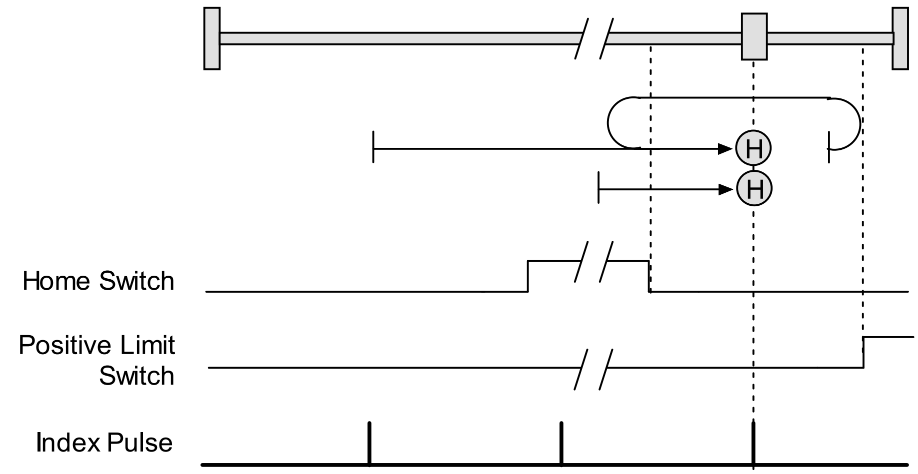
Home is the first index pulse on the positive side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



C.3.14: Upper Home Outside Index

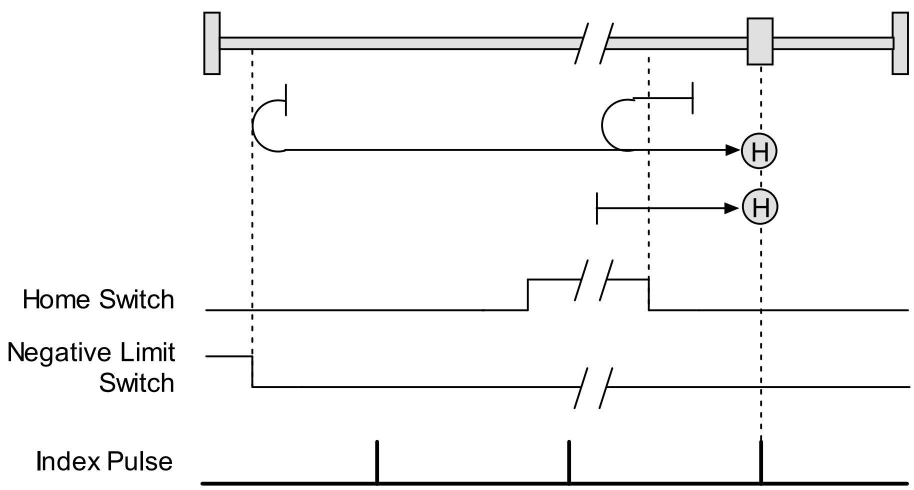
Direction of Motion: Positive (0xc2 = 547)

Home is the first index pulse on the positive side of the positive edge of a momentary home switch. Initial direction of motion is positive. If the initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Direction of Motion: Negative (0xc2 =563)

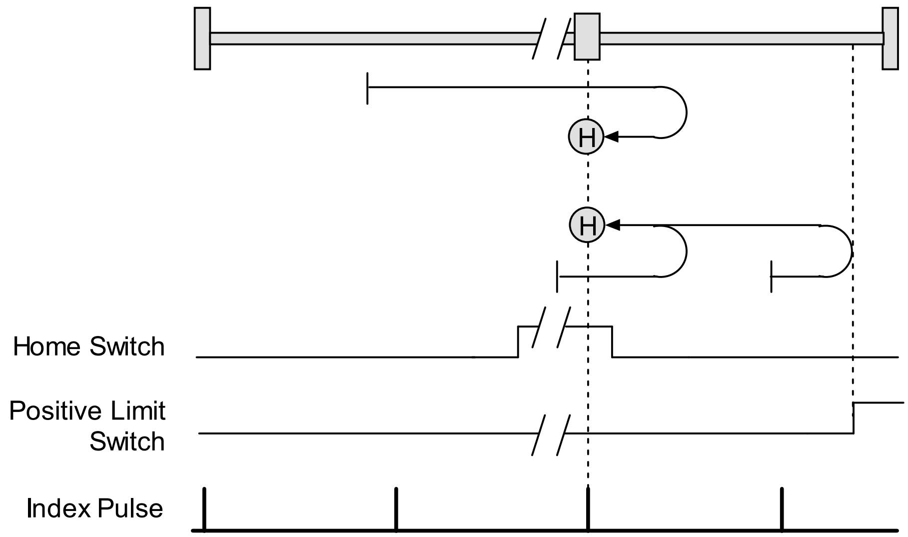
Home is the first index pulse on the positive side of the positive edge of a momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial position is right of the home position, the axis reverses on encountering the home switch.



C.3.15: Upper Home Inside Index

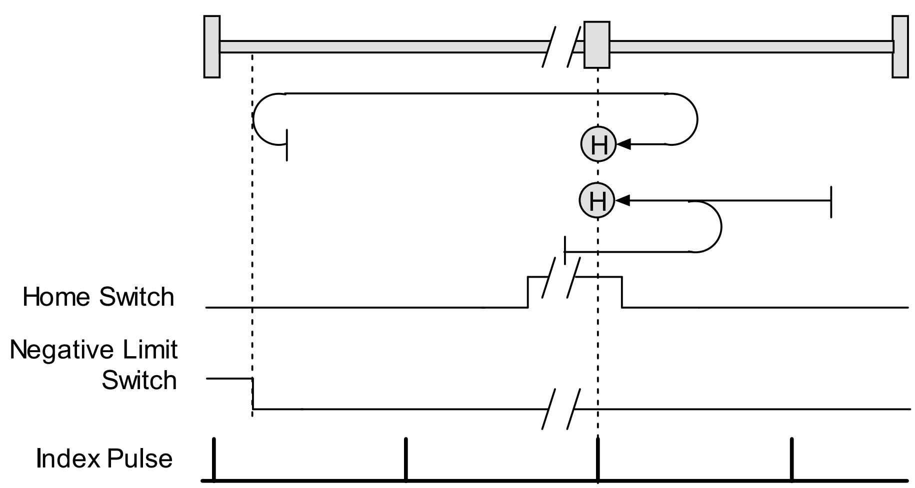
Direction of Motion: Positive (0xc2 =611)

Home is the first index pulse on the negative side of the positive edge of momentary home switch. Initial direction of motion is positive. If initial motion leads away from the home switch, the axis reverses on encountering the positive limit switch; then, if a negative limit switch is activated before the home switch, an error is generated.



Direction of Motion: Negative (0xc2 =627)

Home is the first index pulse on the negative side of the positive edge of a momentary home switch. Initial direction of motion is negative if the home switch is inactive. If initial motion leads away from the home switch, the axis reverses on encountering the negative limit; then, if a negative limit switch is activated before the home switch, an error is generated.



APPENDIX

D: SERIAL AND MULTI-DROP CONNECTION

This appendix describes how to connect an amplifier for control via the RS-232 serial port.

The serially connected amplifier can also be used as a multi-drop gateway to control other amplifiers linked in a series of CAN bus connections.

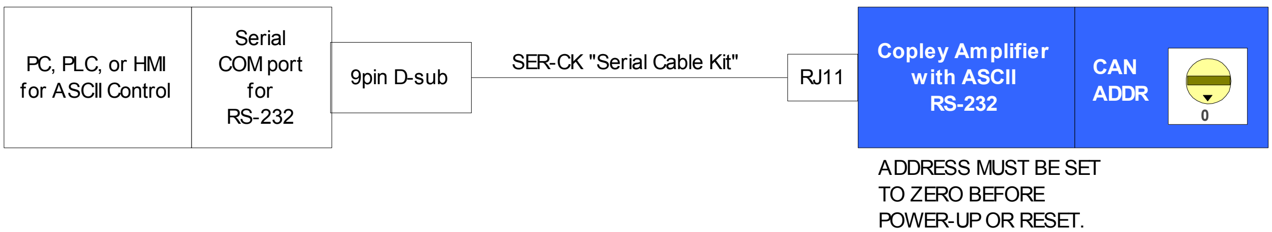
Contents include:

D.1: Connecting

Instructions for hooking up a single-axis connection and a multi-drop network appear below.

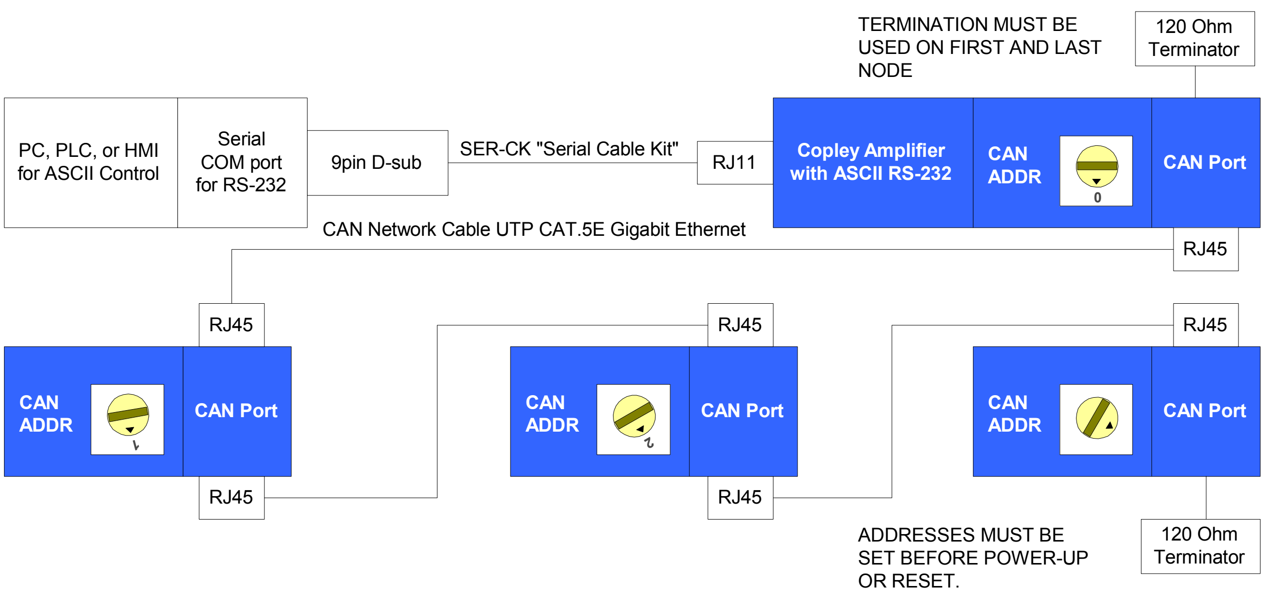
D.1.1: Single-Axis Connections

For RS-232 serial bus control of a single axis, set the CAN node address of that axis to zero (0). Note that if the CAN node address is switched to zero after power-up, the amplifier must be reset or power cycled to make the new address setting take effect.



D.1.2: Multi-Drop Network Connections

For RS-232 serial bus control of multiple axes, set the CAN node address of the serially connected amplifier (gateway) to zero (0). Assign each additional amplifier in the chain a unique CAN node address value between 1 and 127. For information on CAN node address, see the amplifier user guide or data sheet. Verify that all amplifiers are set to the same CAN bit rate. Use 120 Ω termination on the first and last amplifier.



The CAN Status Light and Multi-Drop Connections

When starting amplifiers on a multi-drop CAN loop, it is common to see a green-green-red flash sequence on the CAN Status Indicator LED of the first amplifier to start up. This sequence indicates that the amplifier has not found any other active nodes on the CAN loop. Under normal circumstances, this flash sequence does not indicate a problem, and it will clear after the first few commands are sent to the amplifier.

To avoid seeing this flash sequence, assure that the gateway amplifier starts up first. The CAN status indicator will always be off on node 0.

APPENDIX
E: ERROR CODES

Most ASCII Interface commands can return an error message in the format “e <code>” where code is one of the error code numbers described below.

Code	Meaning
1	Too much data passed with command
3	Unknown command code.
4	Not enough data was supplied with the command.
5	Too much data was supplied with the command.
9	Unknown variable ID.
10	Data value out of range.
11	Attempt to modify read-only variable
14	Unknown axis state.
15	Variable doesn’t exist on requested page.
18	Illegal attempt to start a move while currently moving
19	Illegal velocity limit for move.
20	Illegal acceleration limit for move.
21	Illegal deceleration limit for move.
22	Illegal jerk limit for move.
25	Invalid trajectory mode.
27	Command is not allowed while CVM is running
31	Invalid node ID for serial port forwarding.
32	CAN Network communications failure.
33	ASCII command parsing error.

Copley ASCII Interface Programmer's Guide
P/N CC95-00404-000

Revision A

June 2009