

# Copley DeviceNet Programmer's Guide



P/N 95-00658-000

Revision 2  
June 2008

# Copley DeviceNet Programmer's Guide

**TABLE OF CONTENTS**

**About This Manual ..... iii**

**1: Requirements and Operational Overview ..... 1**

    1.1: Requirements ..... 2

    1.2: DeviceNet and Copley Controls Amplifiers ..... 3

**2: Amplifier Configuration Using CME 2 ..... 5**

    2.1: Start CME 2 ..... 6

    2.2: Download DeviceNet Firmware ..... 7

    2.3: Configure DeviceNet Address and Baud Rate ..... 9

    2.4: View the Amplifier Hardware Type for EDS Name ..... 10

**3: I/O Messaging ..... 11**

    3.1: Overview ..... 12

    3.2: I/O Command Message (Scanner-to-Amplifier) ..... 13

    3.3: I/O Response Message (Amplifier-to-Scanner) ..... 15

    3.4: Handshaking ..... 16

    3.5: Reading Amplifier Parameters ..... 17

    3.6: The Monitor Fields ..... 17

**4: I/O Messages and Basic Motion Tasks ..... 19**

    4.1: Control the Indexer Program ..... 20

    4.2: Home an Axis ..... 22

    4.3: Start an Absolute Trap Move ..... 23

    4.4: Start a Relative S-Curve Move ..... 24

    4.5: Start a Jog Move ..... 25

    4.6: Control Camming ..... 26

**A: Explicit Messages and the DeviceNet Objects ..... 29**

    A.1: Explicit Messages ..... 30

    A.2: The Copley DeviceNet Objects ..... 30

    A.3: The Error Log Object ..... 31

    A.4: The File System Object ..... 32

    A.5: The CVM Control Object ..... 33

    A.6: The Data Trace Object ..... 34

**B: Homing Variables and Methods ..... 37**

    B.1: Homing Methods Overview ..... 38

    B.2: Legend to Homing Method Descriptions ..... 38

    B.3: Homing Method Descriptions ..... 39

**C: I/O Message Amplifier Parameters ..... 53**

    Parameter Descriptions ..... 54



# ABOUT THIS MANUAL

## Overview and Scope

This manual describes the use of Copley Controls amplifiers on a DeviceNet network.

## Related Documentation

Documents of related interest include:

- *CME 2 User Guide* (describes the use of CME 2 amplifier commissioning software to set up and tune the amplifier)
- *Copley Indexer Program User Guide* (describes use of Indexer Program to create motion control sequences)
- *Copley ASCII Interface Programmer's Guide* (describes how to send ASCII format commands over an RS232 serial bus to set up and control one or more amplifiers).

Links to these publications, along with hardware manuals and data sheets, can be found under the *Documents* heading of

<http://www.copleycontrols.com/Motion/Downloads/index.html>.

Copley Controls software and related information can be found on:

<http://www.copleycontrols.com/Motion/Products/Software/index.html>

For DeviceNet specifications, see <http://www.odva.org>.

## Comments

Copley Controls Corporation welcomes your comments on this manual. See

<http://www.copleycontrols.com> for contact information.

## Copyrights

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Copley Controls Corporation.

Xenus, Accelnet, and Stepnet are registered trademarks of Copley Controls Corporation. CME 2 is a registered trademark of Copley Controls Corporation.

Windows NT, 2000, and XP are trademarks or registered trademarks of the Microsoft Corporation.

## 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 Corporation. Copley Controls Corporation assumes no responsibility for any errors that may appear in this document.

### 1.1.1: 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 Corporation should perform repairs to amplifiers.



**DANGER**

**Hazardous voltages.**

Exercise caution when installing and adjusting.

**Risk of electric shock.**

High-voltage circuits are connected to mains power.

**Risk of unexpected motion with non-latched faults.**

After the cause of a non-latched fault is corrected, the amplifier re-enables the PWM output stage without operator intervention. In this case, motion may re-start unexpectedly. Configure faults as latched unless a specific situation calls for non-latched behavior. When using non-latched faults, be sure to safeguard against unexpected motion.

**Latching an output does not eliminate the risk of unexpected motion with non-latched faults.**

Associating a fault with a latched, custom-configured output does not latch the fault itself. After the cause of a non-latched fault is corrected, the amplifier re-enables without operator intervention. In this case, motion may re-start unexpectedly.

**Using CME 2 or serial commands may affect or suspend DeviceNet operations.**

When operating the amplifier as a DeviceNet node, the use of CME 2 or ASCII serial commands may affect DeviceNet operations in progress. Using such commands to initiate motion may cause DeviceNet operations to stop or start unexpectedly.

**Use equipment as described.**

Operate amplifiers within the specifications provided in the relevant hardware manual or data sheet.

**Failure to heed these warnings can cause equipment damage, injury, or death.**

### 1.1.2: Revision History

Revision	Date	DECO #	Comments
1	October 2006		Initial publication.
2	June 2008	17137	Updated Web page references.

# CHAPTER

## 1: REQUIREMENTS AND OPERATIONAL OVERVIEW

Topics include:

---

- ✓ [Requirements \(p. 2\).](#)
  - ✓ [DeviceNet and Copley Controls Amplifiers \(p. 3\).](#)
-

## 1.1: Requirements

### 1.1.1: CME 2 Requirements

Procedures in this manual require Copley Controls CME 2 software, Version 4.2 or higher.

Minimal hardware requirements for CME 2:

- **CPU:** 400 MHZ.
- **RAM:** 128 MB.

Operating Systems Supported: Windows NT, 2000, XP.

Also, for each PC-to-amplifier connection via serial port: One serial communication cable. See the amplifier documentation for part numbers.

### 1.1.2: Firmware

Features described in this manual require Copley Controls amplifier firmware version 5.06 or higher.

Note: Standard Copley Controls amplifiers are not shipped with the DeviceNet firmware. The DeviceNet firmware can be downloaded from:

<http://www.copleycontrols.com/Motion/Downloads/firmware.html>.

For instructions on downloading firmware to the amplifier, see [Download DeviceNet Firmware \(p. 7\)](#).

### 1.1.3: EDS File

A DeviceNet Electronic Data Sheet (EDS) file is available for each Copley Controls amplifier that supports DeviceNet. The EDS file name is based on the amplifier's hardware type. See [View the Amplifier Hardware Type for EDS Name \(p. 10\)](#).

The EDS can be downloaded from:

<http://www.copleycontrols.com/Motion/Downloads/firmware.html>.

### 1.1.4: DeviceNet Communications

Verify that the DeviceNet bus has been cabled and terminated. Copley DeviceNet cabling and connection is identical to Copley CAN cabling and connection. See the amplifier's CAN cabling and connection documentation.

## 1.2: DeviceNet and Copley Controls Amplifiers

Copley Controls Xenus, AccelNet, and StepNet amplifiers can be operated over a DeviceNet network by PLCs and other controllers.

DeviceNet messages can be used to issue basic motion commands, such as start move and start homing sequence, and to read and write internal amplifier parameters.

DeviceNet messages can also control the amplifier's Copley Virtual Machine (CVM) Indexer Program. The Indexer Program provides fully programmable on-board control. Low-bandwidth DeviceNet I/O messages can select and initiate Indexer Program motion sequences and then pass data to and from the Indexer Program, distributing the bulk of motion control overhead to the amplifier.

Copley's DeviceNet implementation supports two explicit message objects and one polled I/O message.

One of the explicit message objects is allocated as part of the predefined master/slave connection set as defined in the DeviceNet specification. The other may be allocated using the Unconnected Message Manager (UCMM) protocol. These objects can be used to access any of the supported DeviceNet objects described in [Explicit Messages and the DeviceNet Objects \(p. 29\)](#). One of these, the Parameter object, provides access to all amplifier parameters.

Copley's I/O messaging is described in [I/O Messaging \(p. 11\)](#). For sample applications of the I/O message protocol, see [I/O Messages and Basic Motion Tasks \(p.19\)](#).

For more information about DeviceNet, including the DeviceNet specifications, see <http://www.odva.org>.



# CHAPTER

## 2: AMPLIFIER CONFIGURATION USING CME 2

Copley Controls CME 2 software allows fast and easy commissioning of Copley Controls amplifiers. CME 2 runs on a PC and connects over the amplifier's RS-232 serial bus.

CME 2 provides access to all amplifier configuration controls. In a typical application, basic amplifier setup and configuration is performed with CME 2, and DeviceNet is used only to modify those parameters which change during amplifier operation.

This chapter shows how to use CME 2 to:

---

- 1 [Start CME 2 \(p. 6\).](#)
  - 2 [Download DeviceNet Firmware \(p. 7\).](#)
  - 3 [Configure DeviceNet Address and Baud Rate \(p. 9\).](#)
  - 4 [View the Amplifier Hardware Type for EDS Name \(p. 10\).](#)
- 

For more information, see the *CME 2 User Guide*.

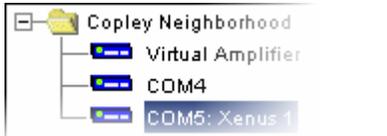
## 2.1: Start CME 2

1

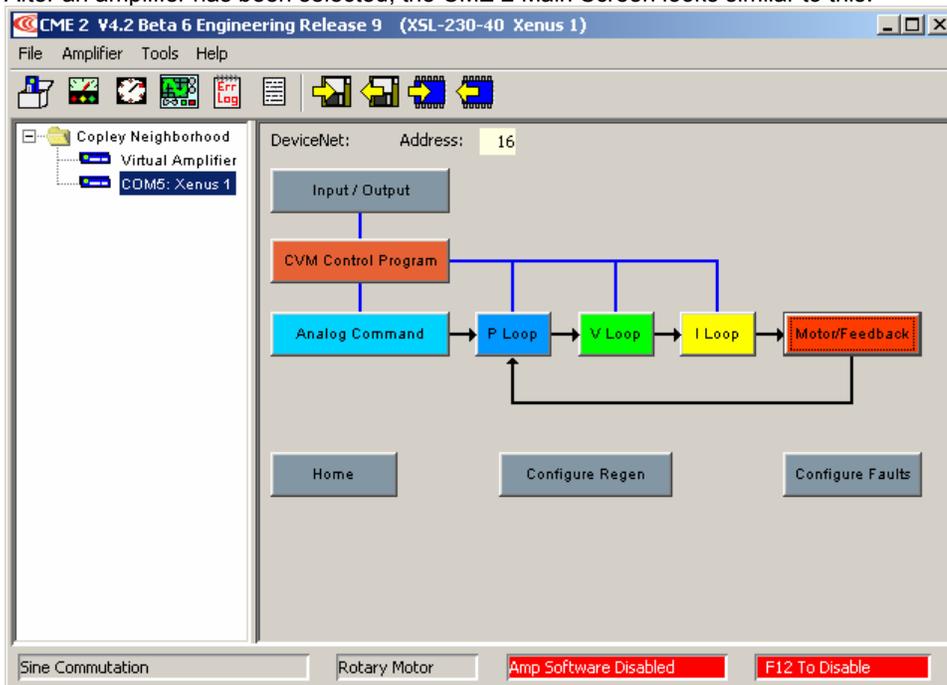


Double-click the CME 2 shortcut icon on the Windows desktop to start CME 2.

2

If communications were set up already...	If communications were not set up...
<p>...the CME 2 Main Screen opens. If there are multiple ports, the Copley Neighborhood root will be selected as shown below:</p>  <p>Select the desired amplifier. For instance:</p> 	<p>...see the screen below:</p>  <p>In this case, select <b>Serial Ports</b> and see the Copley Controls <i>CME 2 User Guide</i> for more set up instructions..</p>

After an amplifier has been selected, the CME 2 Main Screen looks similar to this:



(Note that the DeviceNet Address will not appear until the DeviceNet firmware has been downloaded.)  
 If the Basic Setup options have not been chosen, the *Basic Setup* screen opens.

## 2.2: Download DeviceNet Firmware

The amplifier's flash memory holds the amplifier's firmware. As needed, perform the following steps to download the appropriate DeviceNet firmware version to the amplifier's flash memory.

The DeviceNet firmware can be obtained from:

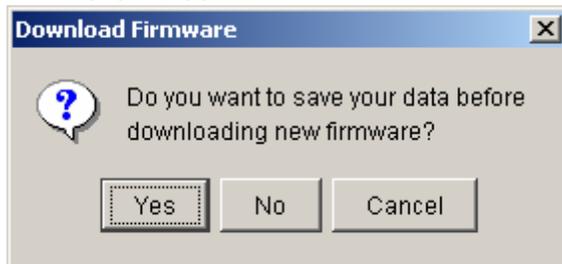
<http://www.copleycontrols.com/Motion/Downloads/firmware.html>.

NOTE: To check the firmware version currently loaded, click the Amplifier Properties button or choose **Help→About**.

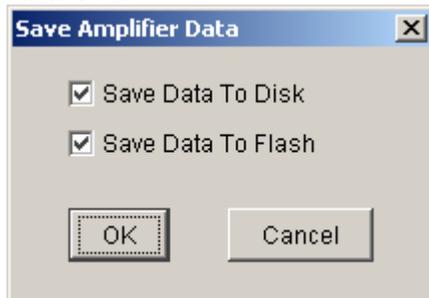
---

### Download Firmware to the Amplifier

- 1 On the *Main* screen choose **Tools→Download Firmware** to open the *Download Firmware* window.



- 2 To download new firmware without saving amplifier and motor data, click **No** and then proceed to Step 4.
- 3 To save amplifier and motor data for backup purposes before downloading firmware, click **Yes**.

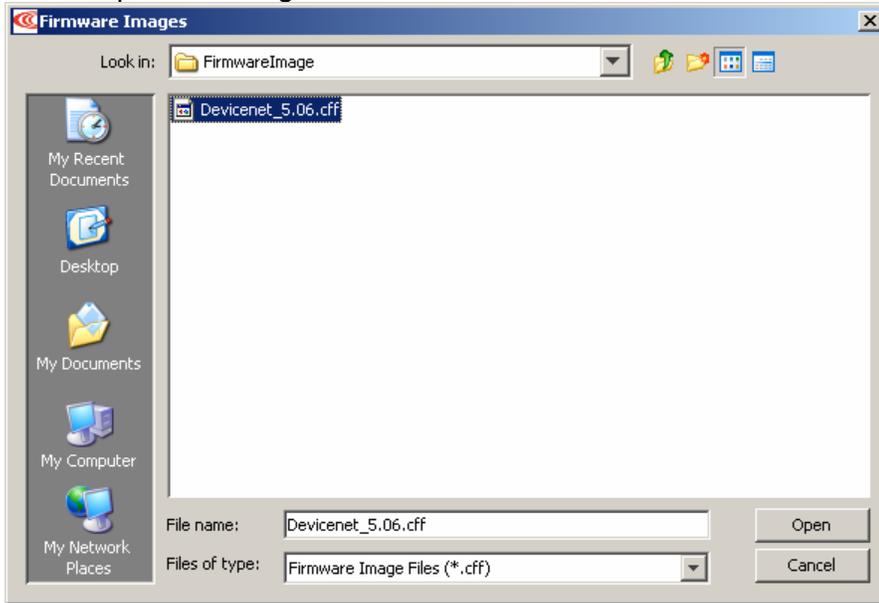


- Choose whether to save to disk, flash, both, or neither.
- Click **OK** to save data and continue to select a firmware image, or click **Cancel** to continue without saving data.
- If *Save Data to Disk* was selected, use the *Save Amplifier Data to Disk* screen to browse to the folder where you want to save the .ccx file. Then enter a name in the *Name* field. Then click **Save**.  
When the Firmware Images window appears, proceed to Step 4.

*Continued...*

...Continued:

- 4 Use the *Firmware Images* window to locate and select the firmware image file. The firmware file name shown below is an example. Contact Customer Support if you need help determining which firmware version to load:



- 5 Click **Open** to begin the download. A message window displays a series of progress messages:



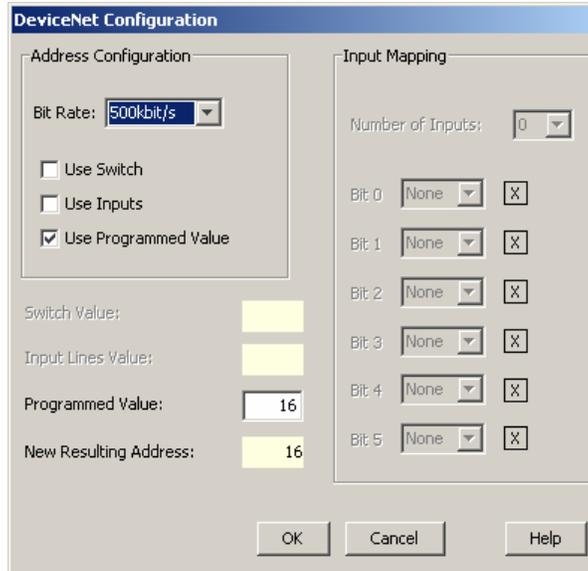
When the message window closes, the firmware download is complete.

---

## 2.3: Configure DeviceNet Address and Baud Rate

A DeviceNet network can support up to 64 nodes. Each node must have a unique and valid seven-bit address (Node ID) in the range of 0-63.

- 1 Verify that the DeviceNet bus has been cabled and terminated. Copley DeviceNet cabling and connection is identical to Copley CAN cabling and connection. See the amplifier's CAN cabling and connection documentation.
- 2 On the CME 2 *Main* screen, choose Amplifier→CAN/DeviceNet Configuration.



(Note that options may vary based on amplifier model and configuration.)

- 3 Choose a Bit Rate and choose any combination of address sources (Switch, Inputs, and Programmed Value). The address is the sum of the values from these sources.
- 4 For each source selected, perform the additional steps described below.

Source	Additional Steps
Use Switch	Verify the S1 switch setting. (Assigns values for Bit 0 – Bit 3 of address.)
Use Inputs	Enter the <b>Number of inputs</b> . Choose the input that will represent each address bit.
Use Programmed Value	Enter the <b>Programmed value</b> .

- 5 Click **OK** to close the screen and save the changes to flash.  
NOTE: DeviceNet address and bit rate changes take effect only after power-up or reset.

## 2.4: View the Amplifier Hardware Type for EDS Name

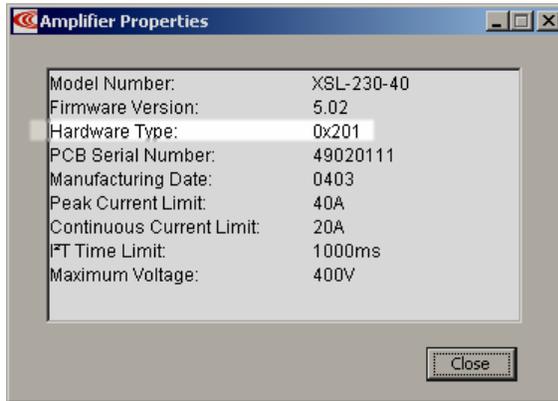
The EDS file for an amplifier will have a file name based on the amplifier's hardware type. Note the Hardware Type before downloading the EDS file at:

<http://www.copleycontrols.com/Motion/Downloads/firmware.html>.

---



On the *Main* screen click the Amplifier Properties button or choose **Amplifier→Properties**. View the hardware type.



# CHAPTER

## 3: I/O MESSAGING

This chapter describes Copley's support of DeviceNet I/O messaging. Contents include:

---

- 1 [Overview \(p. 12\).](#)
  - 2 [I/O Command Message \(Scanner-to-Amplifier\) \(p. 13\).](#)
  - 3 [I/O Response Message \(Amplifier-to-Scanner\) \(p. 15\).](#)
  - 4 [Handshaking \(p. 16\).](#)
  - 5 [Reading Amplifier Parameters \(p. 17\).](#)
  - 6 [The Monitor Fields \(p. 17\).](#)
-

## 3.1: Overview

Copley Controls DeviceNet firmware supports one polled I/O message format. This message can be used to enable the amplifier, clear faults and errors, start and halt moves, and home the amplifier. It also provides access to the amplifier parameters described in [I/O Message Amplifier Parameters \(p. 53\)](#) and access to the amplifier's Indexer Program registers.

The I/O message is configured as part of the *predefined master/slave connection set*, a DeviceNet standard method for allocating I/O messages through the DeviceNet object. The amplifier is the slave, and the DeviceNet scanner is the master.

## 3.2: I/O Command Message (Scanner-to-Amplifier)

### 3.2.1: Command Message Format

The command message consists of 8 bytes of data formatted as follows:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Byte 1</b>	Reserved			Clear Error	Handshake	Halt move	Clear Faults	Enable
<b>Byte 2</b>	Command Code							
<b>Byte 3</b>	Parameter Identifier							
<b>Byte 4</b>	Reserved					Monitor ID		
<b>Byte 5</b>	Data Value bits 0-7							
<b>Byte 6</b>	Data Value bits 8-15							
<b>Byte 7</b>	Data Value bits 16-23							
<b>Byte 8</b>	Data Value bits 24-31							

See [Command Message Field Details \(p. 14\)](#).

### 3.2.2: Command Message Field Details

Field	Description	
Enable	Enable Bit clear: amplifier is disabled. Enable Bit set: amplifier is enabled unless disabled for some other reason (such as an inactive enable input pin).	
Clear Faults	A 0-1 transition of the Clear Faults Bit clears all latched amplifier faults.	
Halt Move	Halt Move Bit set: 1) any move in progress is halted and the amplifier holds position; 2) no new moves may be started.	
Handshake	Helps to properly time the transfer of I/O messages between the DeviceNet scanner and the amplifier. See <a href="#">Handshaking (p. 16)</a> .	
Clear Error	A 0-1 transition of the Clear Error Bit clears all latched command errors.	
Command Code	This 8-bit code is used to initiate amplifier actions. Note that command codes are only processed when the Handshake Bit changes state. The codes are:	
	Code	Description
	0	No action. Can be used in a command message whose only purpose is to get the value of a parameter.
	1	Start a new move. First, the 32-bit value in the data bytes is set as the move destination position, and then the move is started.
	2	Start a homing sequence. Before starting a homing sequence, make sure the homing parameters have been configured. See <a href="#">Home an Axis (p. 22)</a> .
	3	Reset the amplifier.
	4	Write to an Indexer Program register. The 32-bit value passed in the data bytes is written to one of the 32 Indexer Program registers. The register number (0-31) is passed in the Parameter ID field. See <a href="#">Control the Indexer Program (p. 20)</a> .
	5	Read an Indexer Program register. A register number (0-31) is passed in the Parameter ID field. The value of the register is returned in the data bytes of the response message. See <a href="#">Control the Indexer Program (p. 20)</a> .
	6	Write an amplifier parameter. The value passed in the data bytes is written to the parameter identified in Parameter ID. This command can be used to write to the list of parameters described in <a href="#">I/O Message Amplifier Parameters (p. 53)</a> . NOTES: 1. This command writes the parameter value to RAM only (not flash). 2. There is no "read amplifier parameter" command code. See <a href="#">Reading Amplifier Parameters (p. 17)</a> .
7	Update a move in progress, or start a move without setting the destination position. Any changes that have been made to move parameters (velocity, acceleration, etc.) by other command messages during a move will not take effect until this command is entered. This allows multiple move parameters to be changed at the same time. Note: S-Curve moves in progress cannot be updated.	
Parameter ID	The function of this field depends on the command code, as described below:	
	Code	Parameter ID Function
	4	Identifies the Indexer Program register to write to.
	5	Identifies the Indexer Program register to read from.
	6	Identifies the amplifier parameter to write to.
	0-3,7	Identifies the amplifier parameter to read.
	Note that the amplifier does not recognize a change in the value of Parameter ID until the Handshake Bit has changed state.	
Monitor ID	Identifies a source for the Monitor Data in the response message. See <a href="#">The Monitor Fields (p. 17)</a> . Note that the amplifier does not recognize a change in the value of Monitor ID until the Handshake Bit has changed state.	
Data Value	The Data Value bytes hold the 32-bit data value used with command codes 1, 4, and 6.	

## 3.3: I/O Response Message (Amplifier-to-Scanner)

### 3.3.1: Response Message Format

The amplifier sends a response message to every command message received. The response message consists of 8 bytes of data formatted as follows:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Error	Pos Limit	Neg Limit	Warning	Handshake	Move	Fault	Enabled
Byte 2	Reserved					Tracking	Trajectory	Homed
Byte 3	Monitor Value bits 0-7							
Byte 4	Monitor Value bits 8-15							
Byte 5	Data Value bits 0-7							
Byte 6	Data Value bits 8-15							
Byte 7	Data Value bits 16-23							
Byte 8	Data Value bits 24-31							

### 3.3.2: Response Message Field Details

Field	Description								
Enabled	The Enabled Bit is set when the amplifier is enabled and cleared when the amplifier is disabled for any reason.								
Fault	The Fault Bit is set when the amplifier is in a faulted state.								
Move	The Move Bit is set when the motor is in motion and cleared when the axis settles into position at the end of the move.								
Handshake	The response message Handshake Bit echoes the value of the Handshake Bit in the previous command message. Helps to properly time the transfer of I/O messages between the DeviceNet scanner and the amplifier. See <a href="#">Handshaking (p. 16)</a> .								
Warning	The Warning Bit is set if a warning is present in the amplifier. Warning conditions include: current limiting, voltage limiting, velocity or acceleration limiting, position tracking window exceeded, velocity window exceeded, and phase initialization in progress.								
Neg Limit	The Neg Limit Bit is set when a negative limit switch is active or a negative software limit has been violated.								
Pos Limit	The Pos Limit Bit is set when a positive limit switch is active or a positive software limit is violated.								
Error	The Error Bit is set when a command sent to the amplifier causes an error. It remains set until the error is cleared by a 0-1 transition of the command message Clear Error Bit. The error code can be read in parameter 37, <a href="#">Latched I/O Command Error Code (p. 59)</a> .								
Homed	The Homed Bit is set if the amplifier has been successfully homed.								
Trajectory	The Trajectory Bit indicates when the amplifier's trajectory generator is running. It is set during a move and cleared as soon as the trajectory generator reaches the end point of the move. The Move Bit (Bit 2) will usually be cleared shortly after the Trajectory Bit because the Move Bit indicates the axis has settled into position at the end of the move.								
Tracking	The Tracking Bit is set if the position error exceeds the tracking/settling position window.								
Monitor Value	The Monitor ID value in a previous command message selects one of several 16-bit values to display in this field. See <a href="#">The Monitor Fields (p. 17)</a> .								
Data Value	The Data Value depends on the previous command code: <table border="1" data-bbox="500 1730 1446 1885"> <thead> <tr> <th>Code</th> <th>Response Message Data Value</th> </tr> </thead> <tbody> <tr> <td>0-3, 6, 7</td> <td>Value of the parameter named in the command message Parameter ID.</td> </tr> <tr> <td>5</td> <td>Value of the Register named in the command message Parameter ID.</td> </tr> <tr> <td>4</td> <td>Not defined.</td> </tr> </tbody> </table>	Code	Response Message Data Value	0-3, 6, 7	Value of the parameter named in the command message Parameter ID.	5	Value of the Register named in the command message Parameter ID.	4	Not defined.
Code	Response Message Data Value								
0-3, 6, 7	Value of the parameter named in the command message Parameter ID.								
5	Value of the Register named in the command message Parameter ID.								
4	Not defined.								

## 3.4: Handshaking

The I/O message handshake protocol supports the proper timing of I/O messages between the DeviceNet scanner and the amplifier.

The DeviceNet polled I/O messaging protocol is asynchronous. The PLC puts an I/O message into an output register on the scanner. The scanner sends the register contents at the polling frequency. There is no guarantee that the contents of an output register will be updated between any two scanning intervals. Thus, the same command message could be sent multiple times.

The handshake protocol ensures that the amplifier responds only to new commands. The protocol employs a single Handshake Bit (Bit 3 of the I/O message Control Byte).

### Amplifier Handshaking Features

- The amplifier always checks the Control Byte before processing the rest of the message.
- The amplifier executes the command code only if the Handshake Bit has changed state since the previous command. Note that the other Control Byte functions (Clear Faults, Halt Move, Clear Error, and Enable) are processed regardless of the Handshake Bit state.
- The Handshake Bit should be changed AFTER all other bytes in the command message have been set. This ensures that the message processed by the amplifier will include all of the updated information.
- Parameter and Monitor Data values are updated in the response message regardless of the Handshake Bit state. The amplifier does not recognize a change in the value of Parameter ID or Monitor ID until the Handshake Bit has changed state.
- The amplifier's response message always echoes the Handshake Bit from the previous command message.

### Handshake Programming on the Control Side

- When sending an I/O command message, always change the value of the Handshake Bit. Otherwise, the amplifier will not execute the command. (Note the amplifier will process the other Control Byte functions (Clear Faults, Halt Move, Clear Error, and Enable) and update Parameter and Monitor values regardless of the Handshake Bit state.)
- If acknowledgement is required, program the controller to send the command message and then wait for a response message with the same Handshake Bit setting.

## 3.5: Reading Amplifier Parameters

In most cases the response message Data Value bytes return the value of the amplifier parameter identified in the command message Parameter ID field.

For instance, to start a move and retrieve the Actual Motor Position, set the command code to 1 [Start a Move], set Parameter ID to 11 [Actual Motor Position], and change the Handshake Bit. The response message Data Value bytes will contain the Actual Motor Position value.

Note that this method of reading parameter values can NOT be used with command codes 4 [Write Indexer Program Register] or 5 [Read Indexer Program Register].

To retrieve a parameter value without executing a command, send a message with the desired Parameter ID and with command code set to 0 [No Action command], and change the Handshake Bit.

## 3.6: The Monitor Fields

The response message's 16-bit Monitor Data field is used to monitor the amplifier data specified in the Monitor ID field of the previous command message.

ID	Data Returned
0	Bits 0-7: The currently selected parameter ID as of the last Handshake Bit change. Bits 8-16: The most recently accepted command code as of the last Handshake Bit change.
1	The state of all input pins.
2	The voltage on the analog command input (millivolts).



# CHAPTER

## 4: I/O MESSAGES AND BASIC MOTION TASKS

Through examples, this chapter describes how to use I/O messages for some basic motion tasks.

Contents include:

---

- ✓ [Control the Indexer Program \(p. 20\).](#)
  - ✓ [Home an Axis \(p. 22\).](#)
  - ✓ [Start an Absolute Trap Move \(p. 23\).](#)
  - ✓ [Start a Relative S-Curve Move \(p. 24\).](#)
  - ✓ [Start a Jog Move \(p. 25\).](#)
  - ✓ [Control Camming \(p. 26\).](#)
- 



### **DANGER**

#### **Enable Amplifiers with Caution.**

Some of the examples in this chapter include instructions to enable the amplifier. In the examples, it is assumed that the instructions will be performed in a context in which enabling the amplifier is safe.

Before enabling an amplifier make sure that the action will not cause unexpected motion.

**Failure to heed these warnings can cause equipment damage, injury, or death.**

## 4.1: Control the Indexer Program

The Copley Virtual Machine (CVM) Indexer Program is an embedded controller that allows users to create up to 32 motion sequences. A sequence has one or more steps that can combine homing instructions, moves, gains adjustments, and timed delays. Sequence steps can also control the amplifier's digital outputs and monitor the digital inputs.

The Indexer also has 32 registers that can be programmed to select and execute sequences and pass numeric parameters to the program. These registers can be read and written by DeviceNet I/O messages, as shown in the examples below.

### 4.1.1: Write and Read Indexer Registers

- ✓ Send a command message to write the value 1400 to Indexer Program register 15.

Byte		Value	Comments
0	Control	0x09	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x04	Set Command to 4 [Write an Indexer Program Register]. 4 = 0x04.
2	Parameter ID	0x0F	ID of the Register to write to. In this example, 15. 15 = 0x0F.
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0x78	Set the register value. Example shows 1400. 1400 = 0x 00 00 05 78.
5		0x05	
6		0x00	
7		0x00	

- ✓ Send an I/O command message to read the value of Indexer Program register 16.

Byte		Value	Comments
0	Control	0x01	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 1, so set the Handshake Bit to 0.* Binary 00000001 = 0x01.
1	Command	0x05	Set Command to 5 [Read an Indexer Program Register]. 5 = 0x05.
2	Parameter ID	0x10	ID of the Register to write to. In this example, 16. 16 = 0x10.
3	Monitor ID	0x--	Specifies values to monitor in the response message.**
4	Data	0x00	Data bytes are ignored.
5		0x00	
6		0x00	
7		0x00	

\* See [Handshaking \(p. 16\)](#). \*\*\* See [The Monitor Fields \(p.17\)](#).

### 4.1.2: Select a Sequence and Initiate the Go Command

Any Indexer Program register can be used to select a motion sequence, trigger the Go command that executes the selected sequence, or both. When a register is used for both functions, the sequence is identified in Register bits 0-4 and the Go command is initiated by setting Register Bit 15.

The following example assumes that Indexer Program Register 11 was programmed to select a sequence and initiate the Go command. The I/O message is used to select and run sequence number 5.

- ✓ Send a command message to select sequence 5 and trigger the Go command.

Byte		Value	Comments
0	Control	0x09	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x04	Set Command to 4 [Write an Indexer Program Register]. 4 = 0x04.
2	Parameter ID	0x0B	ID of the Register to write to. In this example, 11. 11 = 0x0B.
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0x05	Bits 0-4 [Select Indexer Sequence]: 5. Bits 5-7: Reserved. Binary 00000101 = 0x05.
5		0x80	Bits 8-14: Reserved. Bit 15 [Go]: 1. Binary 10000000 = 0x80.
6		0x00	
7		0x00	

\* See [Handshaking \(p. 16\)](#). \*\*\* See [The Monitor Fields \(p.17\)](#).

Note that the Indexer program clears the Go Register's Bit 15 before executing the sequence, requiring the control application to reset the bit before the program will execute another sequence.

For more information, see the *Copley Indexer Program User Guide*, which includes sections titled *Use a Register to Select a Sequence* and *Use a Register to Initiate Go*.

## 4.2: Home an Axis

The following example uses CME 2 to configure homing parameters and then uses an I/O command message to start the homing sequence.

- ✓ Configure homing parameters using CME 2. See *Homing* in the *CME 2 User Guide*.
- ✓ Send an I/O command message to enable the amplifier and command the home action.

Byte		Value	Comments
0	Control	0x09	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x02	Set Command Code to 2 [Start Homing Sequence]. 2 = 0x02.
2	Parameter ID	0x--	Specifies the parameter whose value is returned in the response message Data Value bytes.**
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0x--	No data required for the Home command. Data bytes are ignored.
5		0x--	
6		0x--	
7		0x--	

\* See [Handshaking \(p. 16\)](#). \*\* See [Reading Amplifier Parameters \(p. 17\)](#). \*\*\*See [The Monitor Fields \(p.17\)](#).

NOTE: If the application requires further homing configuration after the initial CME 2 amplifier setup, I/O messages can access homing configuration parameters. These are I/O message parameters 18-24, as described in [I/O Message Amplifier Parameters \(p. 53\)](#).)

## 4.3: Start an Absolute Trap Move

- ✓ Set Trajectory Profile Type to absolute move, Trapezoidal Profile. (Not necessary if set in previous move.)

Byte	Value	Comments
0	Control	0x09 Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x06 Set Command Code to 6 [Write Parameter]. 6 = 0x06.
2	Parameter ID	0x19 Set Parameter ID to 25 [Trajectory Profile Type]. 25 = 0x19.
3	Monitor ID	0x-- Specifies values to monitor in the response message.***
4	Data	0x00 Set the Trajectory Profile Type value to 0 [Absolute Move, Trap].
5		0x00
6		0x00
7		0x00

- ✓ As necessary, use a series of messages to set Profile Velocity, Acceleration, and Deceleration. In this example, set Profile Velocity 1400 (Units: 0.1 counts/sec).

Byte	Value	Comments
0	Control	0x01 Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 1, so clear the Handshake Bit (0).* Binary 00000001 = 0x01.
1	Command	0x06 Set Command Code to 6 [Write Parameter]. 6 = 0x06.
2	Parameter ID	0x1C Set Parameter ID to 28 [Maximum Velocity]. 28 = 0x1C.
3	Monitor ID	0x-- Specifies values to monitor in the response message.***
4	Data	0x78 Set the velocity. Units: 0.1 counts/sec.
5		0x05 Example shows 1400. 1400 = 0x 00 00 05 78.
6		0x00
7		0x00

- ✓ Send the Move Destination and start the move:

Byte	Value	Comments
0	Control	0x09 Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x01 Set Command Code to 1 [Start a New Move]. 1 = 0x01.
2	Parameter ID	0x-- Specifies the parameter whose value is returned in the response message Data Value bytes.**
3	Monitor ID	0x-- Specifies values to monitor in the response message.***
4	Data	0x98 Set the Move Destination. Units: counts.
5		0x3A Example shows 15000. 15000 = 0x 00 00 3A 98.
6		0x00
7		0x00

\* See [Handshaking \(p. 16\)](#). \*\* See [Reading Amplifier Parameters \(p. 17\)](#). \*\*\*See [The Monitor Fields \(p.17\)](#).

## 4.4: Start a Relative S-Curve Move

- ✓ Set Trajectory Profile Type to relative move, S-Curve Profile. (Not necessary if set in previous move.)

Byte		Value	Comments
0	Control	0x09	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x06	Set Command Code to 6 [Write Parameter]. 6 = 0x06.
2	Parameter ID	0x19	Set Parameter ID to 25 [Trajectory Profile Type]. 25 = 0x19.
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0x01	Set the Trajectory Profile Type value to 257 [Relative Move, S-Curve]. 257 = 0x 00 00 01 01.
5		0x01	
6		0x00	
7		0x00	

- ✓ As necessary, use a series of messages to set Velocity, Accel, Decel, and Jerk. In this example, set Maximum Jerk Rate 750 (Units: 100 counts/ sec<sup>3</sup>).

Byte		Value	Comments
0	Control	0x01	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 1, so clear the Handshake Bit (0).* Binary 00000001 = 0x01.
1	Command	0x06	Set Command Code to 6 [Write Parameter]. 6 = 0x06.
2	Parameter ID	0xCF	Set Parameter ID to 31 [Maximum Jerk Rate]. 31 = 0xCF.
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0xEE	Set the Maximum Jerk Rate. Units: 100 counts/ sec <sup>3</sup> . Example shows 750. 750 = 0x 00 00 02 EE.
5		0x02	
6		0x00	
7		0x00	

- ✓ Set the Move Distance and start the move:

Byte		Value	Comments
0	Control	0x09	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.*
1	Command	0x01	Set Command Code to 1 [Start a New Move]. 1 = 0x01.
2	Parameter ID	0x--	Specifies the parameter whose value is returned in the response message Data Value bytes.**
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0x98	Set the Move Distance. Units: counts. Example shows 15000. 15000 = 0x 00 00 3A 98.
5		0x3A	
6		0x00	
7		0x00	

\* See [Handshaking \(p. 16\)](#). \*\* See [Reading Amplifier Parameters \(p. 17\)](#). \*\*\*See [The Monitor Fields \(p.17\)](#).

## 4.5: Start a Jog Move

This jog move uses the Velocity Profile mode. The Velocity profile resembles the trapezoidal profile, but with no target position. The move obeys acceleration, deceleration, and velocity limits, but continues as though the target position were infinite.

- ✓ Set Trajectory Profile Type to Velocity. (Not necessary if set in previous move.)

Byte	Value	Comments
0	Control	0x09 Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x06 Set Command Code to 6 [Write Parameter]. 6 = 0x06.
2	Parameter ID	0x19 Set Parameter ID to 25 [Trajectory Profile Type]. 25 = 0x19.
3	Monitor ID	0x-- Specifies values to monitor in the response message.***
4	Data	0x02 Set the Trajectory Profile Type value to 2 [Velocity Profile Move]. 2 = 0x02.
5		0x00
6		0x00
7		0x00

- ✓ As necessary, use a series of I/O messages to set Velocity, Accel, and Decel. In this example, set Profile Acceleration to 750 (Units: 10 counts/second<sup>2</sup>).

Byte	Value	Comments
0	Control	0x01 Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 1, so clear the Handshake Bit (0).* Binary 00000001 = 0x01.
1	Command	0x06 Set Command Code to 6 [Write Parameter].
2	Parameter ID	0x1D Set Parameter ID to 29 [Maximum Acceleration Rate]. 29 = 0x1D.
3	Monitor ID	0x-- Specifies values to monitor in the response message.***
4	Data	0xEE Set the Maximum Acceleration Rate. Units: 10 counts/sec <sup>2</sup> .
5		0x02 Example shows 750. 750 = 0x 00 00 02 EE.
6		0x00
7		0x00

- ✓ Set a positive move direction and start the move:

Byte	Value	Comments
0	Control	0x09 Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.*
1	Command	0x01 Set Command Code to 1 [Start a New Move]. 1 = 0x01.
2	Parameter ID	0x-- Specifies the parameter whose value is returned in the response message Data Value bytes.**
3	Monitor ID	0x-- Specifies values to monitor in the response message.***
4	Data	0x01 Set the direction of movement by entering a target position value. Use any positive value for movement in the positive direction, and any negative value for movement in the negative direction. Example uses a value of 1 for positive movement.
5		0x00
6		0x00
7		0x00

\* See [Handshaking \(p. 16\)](#). \*\* See [Reading Amplifier Parameters \(p. 17\)](#). \*\*\*See [The Monitor Fields \(p.17\)](#).

## 4.6: Control Camming

Camming allows Copley amplifiers to synchronize the motion of an axis (the Cam Slave) to the motion or commands of an external device (the Cam Master). A Cam Master can be an encoder, a PLC, or any device that generates electronic pulses to indicate the position of the Master axis. A Cam Slave is an axis controlled by a Copley amplifier. A user defines camming moves by populating Cam Tables. Each Cam Table line contains a Cam Master Position and a corresponding Slave Position. The user also configures the Cam Trigger and the source of the Master input. For more information, see the *Copley Camming User's Guide*.

DeviceNet I/O messages can access the Camming control parameters described as I/O message parameters 42-45 in [I/O Message Amplifier Parameters \(p. 53\)](#).

### 4.6.1: Select CAM Table and Trigger

The following example uses CME 2 for basic configuration and then uses I/O messages at run time to select the active Cam Table and set up the Cam Trigger.

- ✓ Use the CME 2 *Basic Setup* screen to set Operating Mode to Position and Position Loop Input to Camming Single-ended or Camming Differential. See the *Copley Camming User Guide*.
- ✓ Use the CME 2 *Camming* screen to:
  - Set the *Master Control Input* type as appropriate for the application.
  - Set or clear the *Invert Command* field as appropriate.
  - Set *Increment Position On* to Rising or Falling edge as appropriate.
  - Set *Offset Forward* and *Reverse* as appropriate.
  - Set *Startup* to **Camming takes control on startup**.
  - In this example, ignore the CME 2 Cam Trigger settings. They will be configured using an I/O message in the next step.
- ✓ Send an I/O command message to write the active Cam Table number in the Camming Configuration parameter. This parameter configures the Cam Trigger, so we will use this I/O message to set up the Trigger as well. This message also enables the amplifier.

Byte		Value	Comments
0	Control	0x09	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x06	Set Command Code to 6 [Write Parameter]. 6 = 0x06.
2	Parameter ID	0x2A	Set Parameter ID to 42 [Camming Configuration]. 42 = 0x2A.
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0x08	The Camming Configuration parameter is bit-mapped: Bits 0-3: 8 to choose Cam Table 8. Bits 4, 5: 0 [Reserved]. Bit 6: 0 to choose external Cam Master. Bit 7: 0 [Reserved]. Binary 00001000 = 0x08.
5		0x1B	Bits 8-11: 11. Use programmable input 11 (IN11) as trigger. Bits 12-13: 1 to choose trigger type Use Input, Edge. Bits 14, 15:0 [Reserved]. Binary 00011011 = 0x1B.
6		0x00	Ignored.
7		0x00	Ignored.

\* See [Handshaking \(p. 16\)](#). \*\*\*See [The Monitor Fields \(p.17\)](#).

### 4.6.2: Adjust Camming Delay

This example shows how to use an I/O message to set the Camming Delay Forward parameter value to 750 (Units: Master Command Counts). Assume here that Camming has already been configured.

- ✓ Send the I/O message:

Byte		Value	Comments
0	Control	0x09	Bit 0 [Enable] Set to 1 to enable the amplifier. Bit 3 [Handshake] Assume here that the previous handshake value was 0, so set the Handshake Bit to 1.* Binary 00001001 = 0x09.
1	Command	0x06	Set Command Code to 6 [Write Parameter]. 6 = 0x06.
2	Parameter ID	0x2B	Set Parameter ID to 43 [Camming Delay Forward]. 43 = 0x2B.
3	Monitor ID	0x--	Specifies values to monitor in the response message.***
4	Data	0xEE	Set Camming Delay Forward. Units: Master Command Counts. Example shows 750. 750 = 0x 00 00 02 EE.
5		0x02	
6		0x00	
7		0x00	

\* See [Handshaking \(p. 16\)](#). \*\*\*See [The Monitor Fields \(p.17\)](#).



# CHAPTER

## A: EXPLICIT MESSAGES AND THE DEVICENET OBJECTS

Contents include:

---

- ✓ [Explicit Messages \(p. 30\).](#)
  - ✓ [The Copley DeviceNet Objects \(p. 30\).](#)
-

## A.1: Explicit Messages

One of the explicit message objects is allocated as part of the predefined master/slave connection set as defined in the DeviceNet specification.

The other may be allocated using the Unconnected Message Manager (UCMM) protocol. These objects can be used to access any amplifier parameter.

## A.2: The Copley DeviceNet Objects

Through the use of the explicit message protocol, the following standard DeviceNet objects can be accessed.

Object Name	ID	Description
Identity	1	Identifies the amplifier as a Copley Controls servo amplifier.
DeviceNet	3	The DeviceNet object allows the baud rate and MAC ID of the amplifier to be read but not written. The baud rate and MAC ID can be changed through Copley's CME 2 software, or by accessing the appropriate amplifier parameters using the parameter object. After changing these parameters, reset or power-cycle the amplifier.
Assembly	4	The Copley Controls DeviceNet firmware currently supports one Assembly object, which the amplifier employs to support the use of I/O messages.
Connection	5	The Copley Controls DeviceNet firmware supports two connection objects. One which is used for explicit messaging and the second is used for I/O messaging.
Parameter	15	The Copley Controls DeviceNet firmware allows all amplifier parameters to be accessed through the DeviceNet standard parameter object class. There is one Parameter object instance for each amplifier variable.  For the complete list of amplifier parameters, see the <i>Copley Controls Amplifier Parameter Dictionary</i> .  When parameters are set through the DeviceNet parameter objects, their values are only set in RAM. To store all parameter settings in non-volatile flash memory, send the SAVE service code (0x16) to the class instance of the Parameter object. (Note that I/O messages only write to RAM.)
Error Log	100	This object class is used to read from the amplifier's error log. See <a href="#">The Error Log Object (p. 31)</a> .
File System	101	This object class allows access to the file system present in the amplifier. This file system is the location where CVM programs and cam tables are stored. See <a href="#">The File System Object (p. 32)</a> .
CVM Control	102	This object class allows various aspects of the Copley Virtual Machine (CVM) to be controlled. See <a href="#">The CVM Control Object (p. 33)</a> .
Data Trace	103	This object class allows access to the internal data tracing feature of the amplifier. The trace may be used to capture internal amplifier parameters in real time and download the captured data. See <a href="#">The Data Trace Object (p. 34)</a> .

## A.3: The Error Log Object

The Error Log object class (class ID 100) may be used to read from the amplifier's internal error log. It can also be used to clear the error log.

### Class Instance Attributes

Attribute	Description
1	Object revision. This always reads the value 1.
2	Read-only value giving the maximum number of instances of this object. Always reads the value 1.

### Object Instance Attributes

Attribute	Description
1	Error counter totals. This read-only attribute returns an array of 32- byte values. Each value gives the number of errors (up to 255) of the given error type since the drive was manufactured.
2	Error counter totals. This read-only attribute returns an array of 32-byte values. Each value gives the number of errors (up to 255) of the given error type since the last time the error log was cleared.
3	Total number of system resets. This read-only 16-bit attribute gives the number of power/reset cycles since drive was manufactured.
4	Clear error log. Writing any value to this attribute causes the error log to be cleared. Note that the totals accessed by attributes 1 [Errors since amplifier manufacture date] and 3 [Resets since amplifier manufacture date] are not cleared.
5	Error log index. This 16-bit read/write attribute is used to access an index value used to read the log (see attribute 6).
6	Error log entry. This read-only attribute is used to read a single entry in the error log. The entry read is identified by a zero-based index which may be set using attribute 5. The index is automatically incremented each time attribute 6 is read. The data returned is an 8-byte error log entry.

## A.4: The File System Object

The amplifier flash memory houses a small file system which may be used to store Copley Virtual Machine (CVM) programs and cam tables. The File System object class (class ID 101) may be used to access files in this file system.

Each file stored in the amplifier file system is allocated two blocks of memory. One, the main data area, is used by the amplifier when running the CVM code. The other, a header area, is typically used to store an ASCII file name.

### Class Instance Attributes

Attr	Description
1	Object revision. A 16-bit read-only value. Always set at 1.
2	Number of instances. This 16-bit read-only attribute gives the maximum number of files that can be stored in the file system. In the present version of firmware this is 32.
8	File system information. This read-only attribute gives some basic information about the file system. The returned data is an array of 4 16-bit values. These values contain the: <ul style="list-style-type: none"> <li>• Maximum number of files that the file system can hold.</li> <li>• Size of each block of memory used to implement the file system.</li> <li>• Total number of data blocks available in the amplifier.</li> <li>• Number of free blocks available for new files.</li> </ul>
9	Maximum transmit words. This read/write 16-bit attribute allows the maximum number of data words to be controlled when reading from the file system. The value programmed here gives the maximum number of 16-bit words that will be sent in a single transmission by the amplifier. Its default value is to send the maximum amount possible (200 words). There is no need to change this unless the host system is unable to handle a transmission of this length.

### Object Instance Attributes

The file system has a limited number of directory entries used to store files of variable length. Each of these entries has an object of this type associated with it. The attributes of these objects are used to allocate file space, read and write data, and delete files. Internal to the amplifier, the directory entries are numbered from 0 to 31. The object instance numbers used to access these directory entries range from 1 to 32. Object instance 1 accesses directory entry 0, object instance 2 accesses directory entry 1, etc.

Attr	Description
1	File size. This read/write attribute accesses the current size of a file. Reading this attribute returns two 16-bit values: The first value gives the size of the data block reserved for the file header. The file header can store file attributes such as the file name. The second value gives the size of the file's data area. Writing to this object allocates space for the file. The amplifier first deletes the file and then reallocates the file space using the two 16-bit values passed. The first value gives the size of file header. This is limited to 255 words max. The second gives the size (in words) of the file data area.
2	Access file header. This attribute allows the programmer to read and write file header information. When writing, an even number of bytes must be passed in each access. When reading data from the file, the number of 16-bit words of data sent from the drive will be limited by the value of class attribute 9, <a href="#">Maximum transmit words (p. 32)</a> . This value may be adjusted to reduce the size of any individual transfer. On both reads and writes, the first word of data sent will be identified by the current value of the file position pointer. See instance attribute 4 <a href="#">File position pointer (p. 32)</a> for details.
3	Access file data. This attribute is used to read and write file data. It works in the same way as attribute 2 except that it accesses the data area of the file rather than the header area.
4	File position pointer. The file system maintains a 16-bit file position pointer shared among all the files in the file system. On the first access of a particular file, the position will be set to zero. On subsequent accesses, the position will be adjusted so consecutive words will be read or written. Instance attribute 4 allows the position pointer to be read. It is also possible to write to the position pointer, but only to adjust the initial position for the file's data area (not the header area). Each time a different file or different area of the same file is accessed, the position pointer is reset to zero. On reads of file header or data, position pointer's initial value may be passed as extra data along with the 'get attribute single' service code. Send the position pointer as a 16-bit word value.

## A.5: The CVM Control Object

The CVM Control Object (object class 102) may be used to access the Copley Virtual Machine (CVM).

### Object Class Attributes

Attribute	Description
1	Object revision. A 16-bit read-only value. Always set at 1.
2	Number of instances. This 16-bit read-only attribute gives the number of object instances of the CVM Control object class. This will always return zero since the CVM Control object class does not have any object instances.
8	Status & control. This 16-bit read/write attribute can be used to query and control CVM status. The status value returned on a read will have its least significant bit set if the CVM is currently executing a program, and cleared if it is not. Other bits in this status word are reserved for internal use by the amplifier (which may set and clear the bits for internal purposes).  Writing a zero to this attribute will stop the CVM. Writing a one will start it again. Do not write any other values to this attribute.
9	Get / run current program. Reading this 16-bit attribute returns the current CVM program number. Writing to it causes the specified program to be loaded into the CVM and run. The CVM program numbers are the same as their file system file numbers. Note that CVM program numbers range from 0 to 31.
10	Get / load current program. Like attribute 9 above, but when a value is written to this attribute the program will be loaded into CVM memory space, but not started. The program may be later started by writing a 1 to attribute 8.
11-13	Reserved for use by Copley Controls.
14	CVM startup program number. This 16-bit read/write attribute identifies the CVM program number (0-31) that will be loaded and run on power-up.  Writing the value 0xFFFF to this attribute prevents any CVM program from running automatically on amplifier startup.

## A.6: The Data Trace Object

The Data Trace object (object class 103) is used to access the amplifier's data trace mechanism. (This is the mechanism that is used to implement the oscilloscope display in the CME 2 user interface program.)

The data trace mechanism allows the amplifier to record some internal parameters in real time to local memory. This data may later be downloaded over the network. All the following attributes are 16-bit values unless specified otherwise.

### Object Class Attributes

Attribute	Description
1	Object revision. A 16-bit read-only value. Always set at 1.
2	Number of instances. This 16-bit read-only attribute gives the number of object instances of the Data Trace object class. This will always return zero since the Data Trace object class does not have any object instances.
8	Trace variable 1. This attribute is used to program the first trace variable to collect. Trace variable codes are listed in <a href="#">Trace Variables (p. 35)</a> .
9	Trace variable 2. Second variable to collect (set to zero for none).
10	Trace variable 3.
11	Trace variable 4.
12	Trace variable 5.
13	Trace variable 6.
14, 15	Reserved.
16	This attribute is used to read or write the trace status. Write 1 to start a trace, write 0 to stop it.
17	This 32-bit read-only attribute gives the fundamental trace period in units of nanoseconds. The fundamental period defines the maximum frequency that the trace system can sample data. The actual trace period is set in integer multiples of this value by programming attribute 20.
18	Samples available. This read-only attribute gives the number of trace samples collected so far.
19	Max samples. This read-only attribute gives the maximum number of samples that the internal trace memory buffer can hold. Note that the maximum number of samples is dependent on the number and type of active trace variables. For an accurate value, the trace variables should be set first, and then the maximum number of samples available may be requested.
20	Sample period. This attribute sets the trace sample period between trace samples. The sample period is set in units of fundamental trace periods (see attribute 17).
21	Trigger delay. This attribute gives the delay between the trigger occurring and the start of captured data. The delay is given in units of trace sample periods (see attribute 20).  Note that the delay may be either positive or negative. A negative delay means that the data captured will precede the trigger event by the specified number of cycles. Although any input value is accepted, the number of samples preceding the trigger is limited to the length of the trace buffer and the number (and size) of channels being captured.
22	Trigger type. This attribute defines the type of trigger to be used with the trace. The data written to this attribute is 48 bits long. It consists of a 16-bit configuration value followed by a 32-bit level. See <a href="#">Trace Trigger Types (p. 36)</a> .
23	Read the collected trace data. This read-only attribute is used to read the array of collected trace data from the amplifier.  Two additional words of data must be passed, the (zero based) index of the first sample, and the number of samples requested. The requested samples will be returned limited to the number that will fit in the internal communications buffer.  Each sample consists of N variables that are each 32-bits long where N is the number of active channels.

### Trace Variables

Each of Data Trace Class Attributes 8-13 can be configured to monitor one of the trace variables described below by programming the attribute with the code.

Trace Code	Trace Variable
0	No data. Setting a channel to this value disables it. Disabling unused channels saves space in the trace buffer.
3	Current reading winding A (0.01 A).
4	Current reading winding B (0.01 A).
5	Reference A/D reading (millivolts).
6	High voltage reference (0.1 volts).
7	Commanded torque.
8	Limited torque.
9	Commanded current (D rotor axis) (0.01 A).
10	Commanded current (Q rotor axis) (0.01 A).
11	Actual current (X stator axis) (0.01 A).
12	Actual current (Y stator axis) (0.01 A).
13	Actual current (D rotor axis) (0.01 A).
14	Actual current (Q rotor axis) (0.01 A).
15	Current Error (D rotor axis) (0.01 A).
16	Current Error (Q rotor axis) (0.01 A).
17	Current Integral (D rotor axis).
18	Current Integral (Q rotor axis).
19	Current loop output (D rotor axis).
20	Current loop output (Q rotor axis).
21	Current loop output (X stator axis).
22	Current loop output (Y stator axis).
23	Actual motor velocity (0.1 counts/sec or 0.01 RPM if using back EMF velocity estimate).
24	Commanded motor velocity.
25	Limited motor velocity command.
26	Velocity loop error.
27	Velocity loop integral.
28	Actual load position (encoder counts).
29	Commanded position.
30	Position loop error.
31	Motor encoder position (encoder counts).
32	Position loop output velocity.
33	Raw input pin readings (no debounce).
34	Reserved.
35	Reserved.
<i>Continued...</i>	

...Continued:	
<b>Trace Code</b>	<b>Trace Variable</b>
36	Motor phase angle (1 degree units).
37	Amplifier temperature (degrees C).
38	Amplifier Status Register.
39	Amplifier event latch word.
40	Hall sensor state.
41	Position capture status register
42	Index capture register.
43	Load encoder velocity (0.1 counts / second).
44	Velocity command from trajectory generator (0.1 counts/sec).
45	Acceleration command from trajectory generator (10 counts/sec <sup>2</sup> ).
46	The analog encoder sine input. Only valid for amplifiers with analog encoder support.
47	The analog encoder cosine input. Only valid for amplifiers with analog encoder support.
48	The value of the digital inputs (after debounce).
49	The destination position input to the trajectory generator.
50	Actual motor velocity as seen by velocity loop. This is an unfiltered version of trace variable.

**Trace Trigger Types**

Data Trace Class Attribute 22 is programmed with one of the codes below to select a trace trigger type.

<b>Type</b>	<b>Description</b>
0	No trigger in use.
1	Trigger as soon as the selected channel's input is greater then or equal to the trigger level.
2	Trigger as soon as the selected channel's input is less then or equal to the trigger level.
3	Trigger when the selected channel's input changes from below to above the trigger level.
4	Trigger when the selected channel's input changes from above to below the trigger level.
5	Trigger when any selected bits in the channel value are set. The bits are selected using the trigger level value as a mask.
6	Trigger when any selected bits in the channel value are clear. The bits are selected using the trigger level value as a mask.
7	Trigger any time the selected channel value changes.
8	The trigger level mask selects one or more bits in the Status Register. The trigger occurs when any of these bits change from to 1. In this mode, the channel number selected by the trigger is not used.
9	Like type 8, but the trigger occurs when the bit(s) change from 1 to 0.
10	Trigger on the start of the next function generator cycle. This trigger type is only useful when running in function generator mode. The trigger channel number is not used.

# APPENDIX

## B: HOMING VARIABLES AND METHODS

Contents include:

Section	Page
B.1: Homing Methods Overview .....	38
B.2: Legend to Homing Method Descriptions .....	38
B.3: Homing Method Descriptions .....	39
B.3.1: Set Current Position as Home .....	39
B.3.2: Next Index .....	39
B.3.3: Limit Switch .....	40
B.3.4: Limit Switch Out to Index .....	41
B.3.5: Hardstop .....	42
B.3.6: Hardstop Out to Index .....	43
B.3.7: Home Switch .....	44
B.3.8: Home Switch Out to Index .....	45
B.3.9: Home Switch In to Index .....	46
B.3.10: Lower Home .....	47
B.3.11: Upper Home .....	48
B.3.12: Lower Home Outside Index .....	49
B.3.13: Lower Home Inside Index .....	50
B.3.14: Upper Home Outside Index .....	51
B.3.15: Upper Home Inside Index .....	52

## B.1: Homing Methods Overview

There are several homing methods. Each method establishes the:

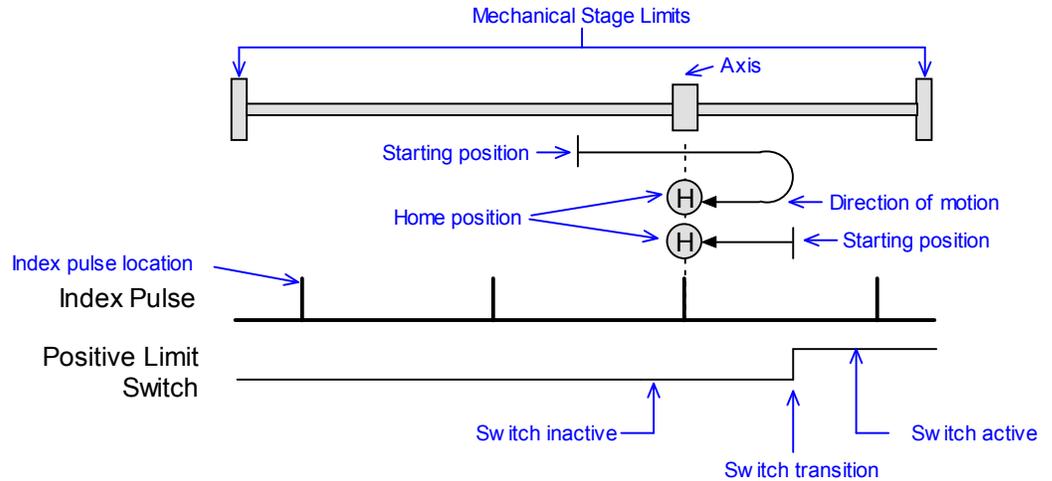
- Home reference (limit or home switch transition or encoder index pulse)
- Direction of motion and, where appropriate, the relationship of the index pulse to limit or home switches.

The Homing Method can be selected using CME 2 or by using DeviceNet explicit or I/O messages to write the parameter [Homing Method \(p. 57\)](#).

As illustrated in the example [Home an Axis \(p. 22\)](#), the preferred method is to use CME 2 for this and other configuration tasks.

## B.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.

## B.3: Homing Method Descriptions

### B.3.1: Set Current Position as Home

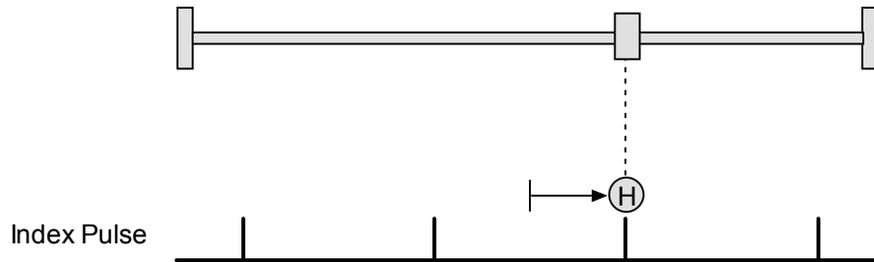
(Homing Method Value: 512)

The current position is the home position.

### B.3.2: Next Index

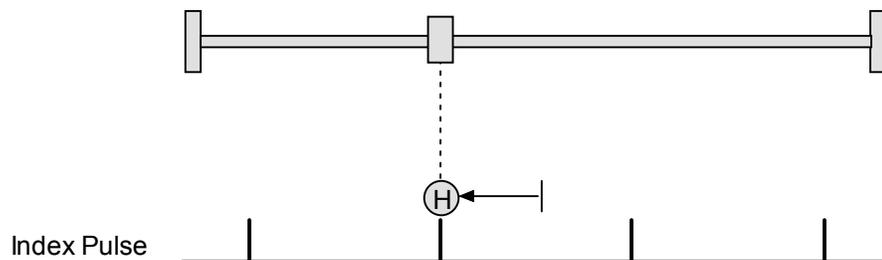
**Direction of Motion: Positive (Homing Method Value: 544)**

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 (Homing Method Value: 560)**

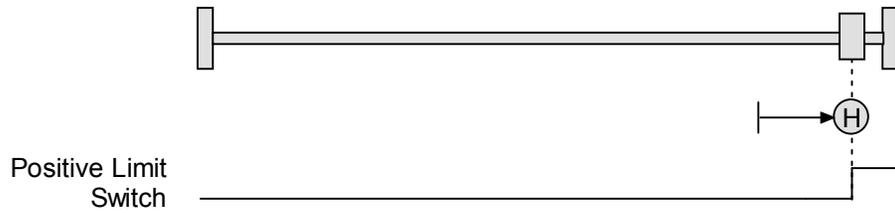
Home is the first index pulse found in negative direction. Direction of motion is negative. If a negative limit switch is activated before the index pulse, an error is generated.



### B.3.3: Limit Switch

**Direction of Motion: Positive (Homing Method Value: 513)**

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



**Direction of Motion: Negative (Homing Method Value: 529)**

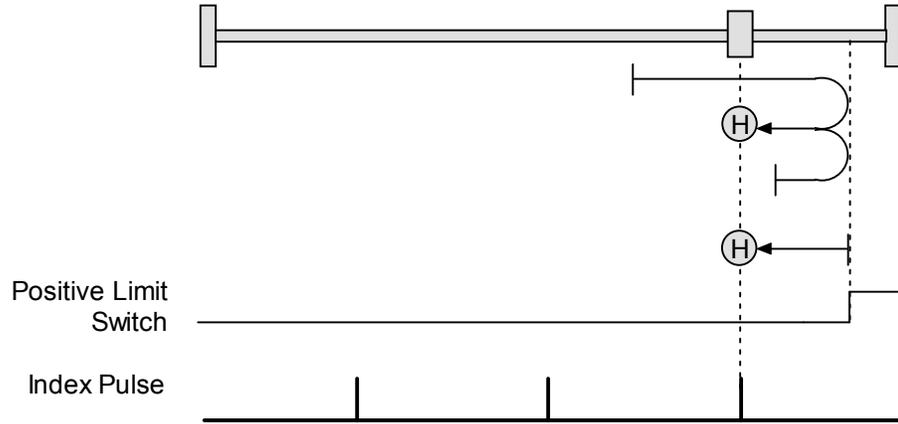
Home is the transition of negative limit switch. Initial direction of motion is negative if the negative limit switch is inactive.



### B.3.4: Limit Switch Out to Index

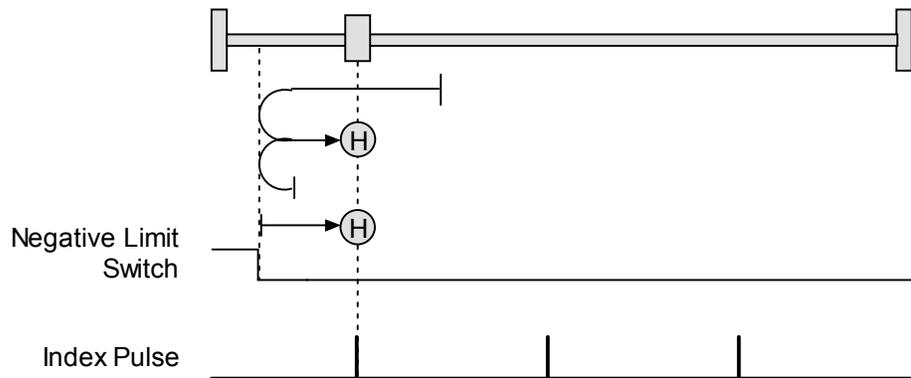
#### Direction of Motion: Positive (Homing Method Value: 545)

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 (Homing Method Value: 561)

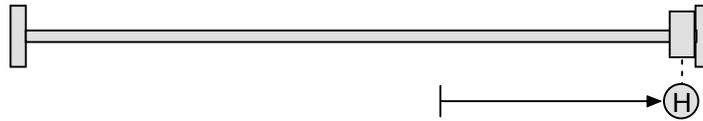
Home is the first index pulse to the positive side of the negative limit switch transition. Initial direction of motion is negative if the negative limit switch is inactive (shown here as low).



### B.3.5: Hardstop

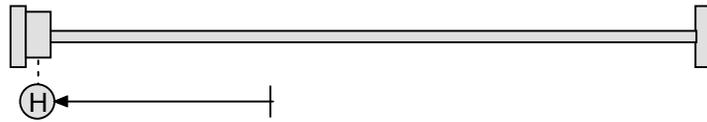
**Direction of Motion: Positive (Homing Method Value: 514)**

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 (Homing Method Value: 530)**

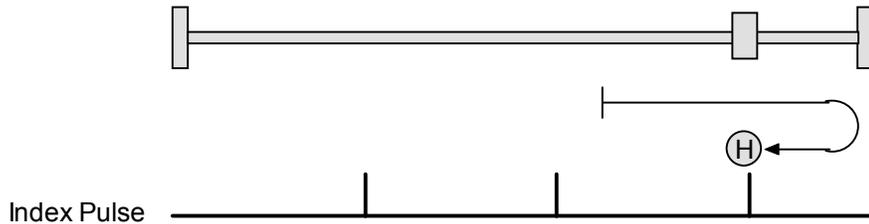
Home is the negative hard stop. Direction of motion is negative. The hard stop is reached when the amplifier outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a negative limit switch is activated before the hard stop, an error is generated.



### B.3.6: Hardstop Out to Index

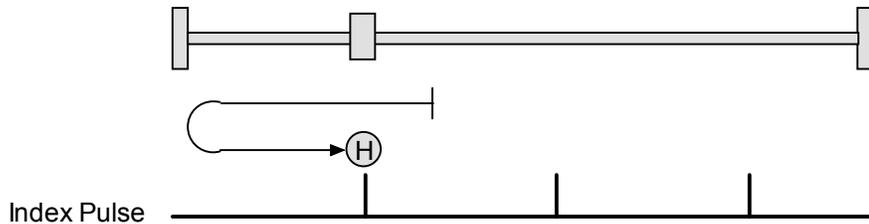
#### Direction of Motion: Positive (Homing Method Value: 546)

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 (Homing Method Value: 562)

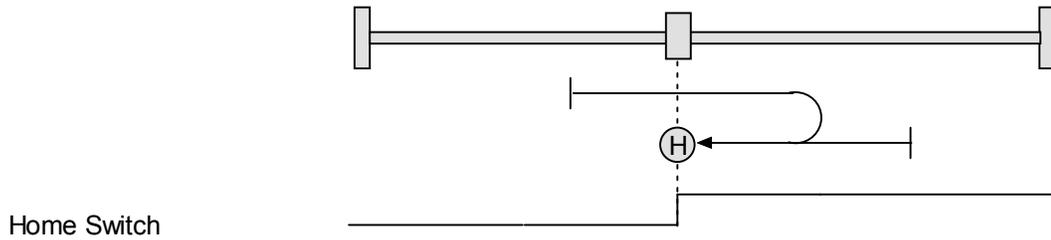
Home is the first index pulse on the positive side of the negative hard stop. Initial direction of motion is negative. The hard stop is reached when the amplifier outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a negative limit switch is activated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



### B.3.7: Home Switch

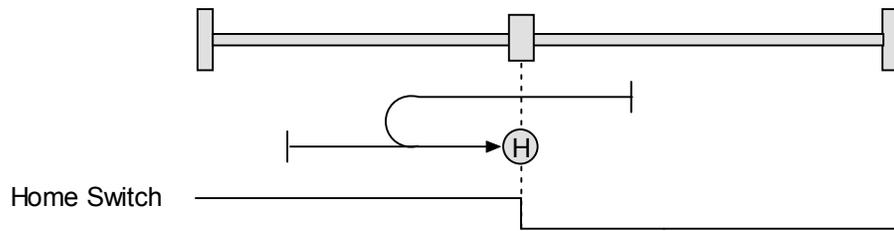
**Direction of Motion: Positive (Homing Method Value: 610)**

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.



**Direction of Motion: Negative (Homing Method Value: 626)**

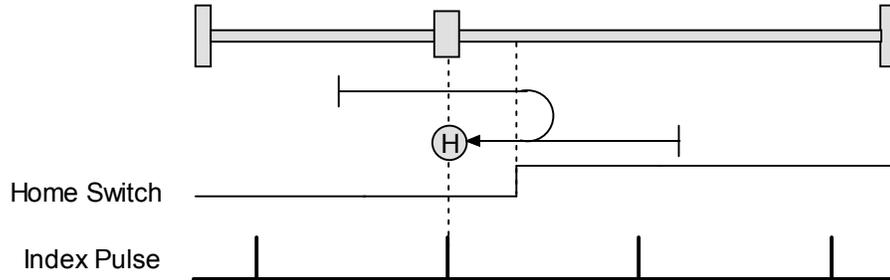
Home is the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



### B.3.8: Home Switch Out to Index

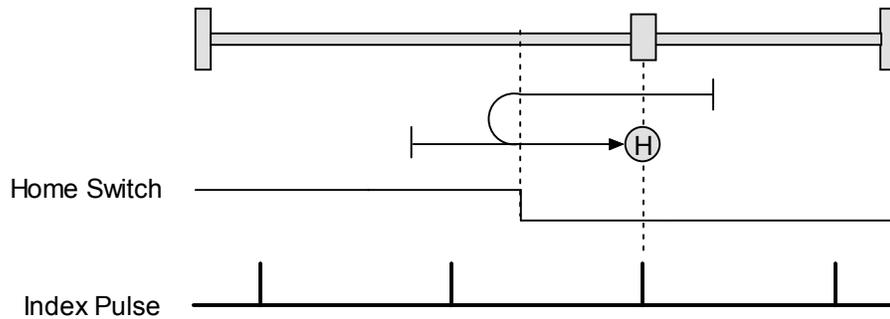
#### Direction of Motion: Positive (Homing Method Value: 516)

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 (Homing Method Value: 532)

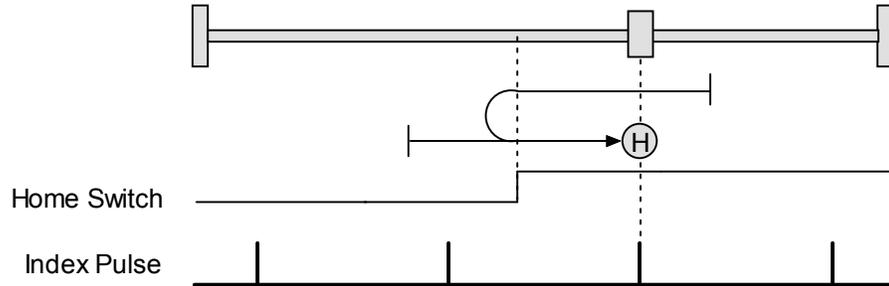
Home is the first index pulse to the positive side of the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



### B.3.9: Home Switch In to Index

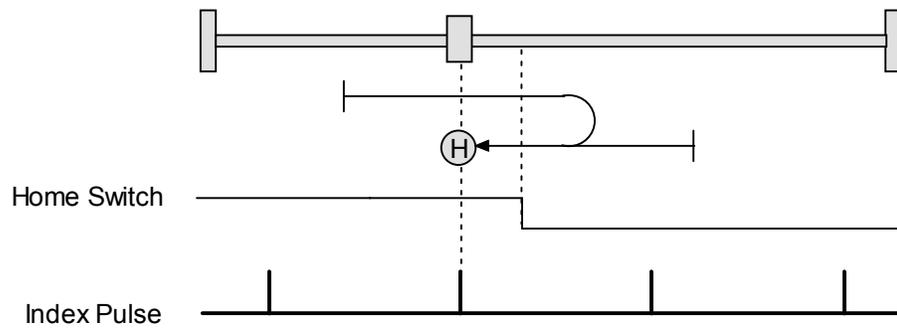
**Direction of Motion: Positive (Homing Method Value: 548)**

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 (Homing Method Value: 564)**

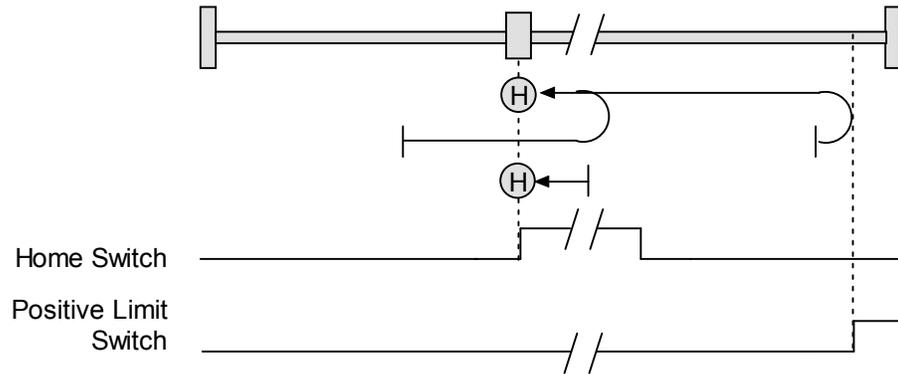
Home is the first index pulse to the negative side of the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



### B.3.10: Lower Home

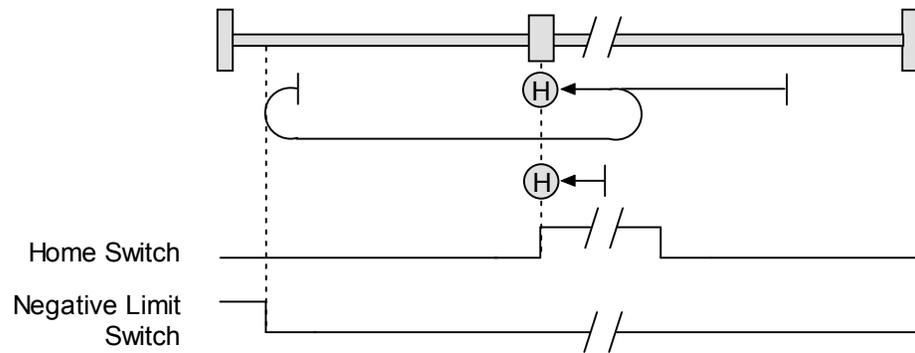
#### Direction of Motion: Positive (Homing Method Value: 771)

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 (Homing Method Value: 787)

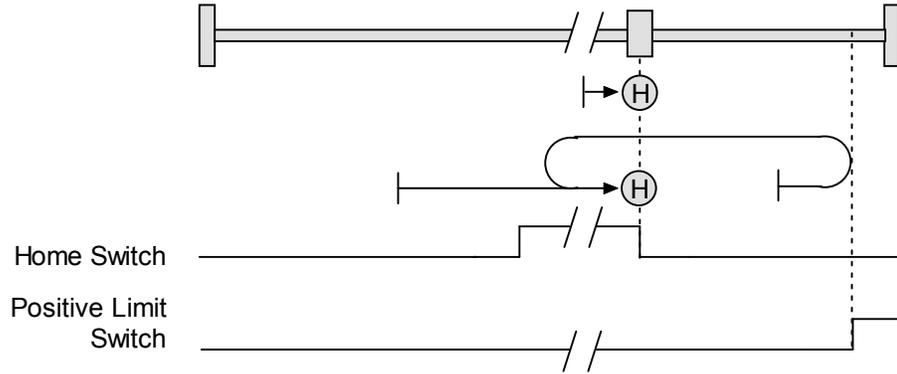
Home is the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home switch, an error is generated.



### B.3.11: Upper Home

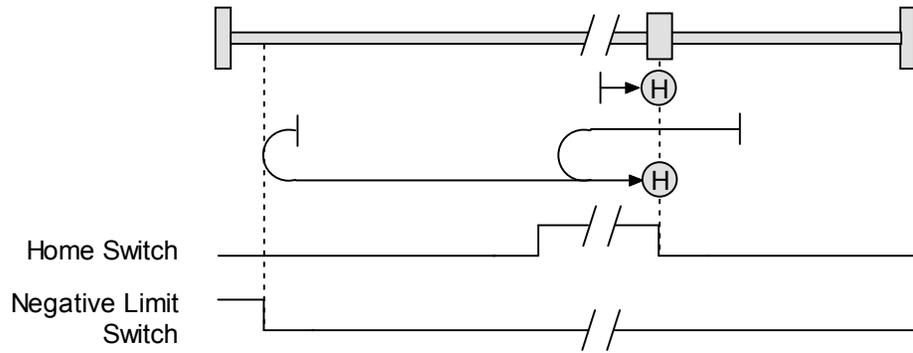
#### Direction of Motion: Positive (Homing Method Value: 515)

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 (Homing Method Value: 531)

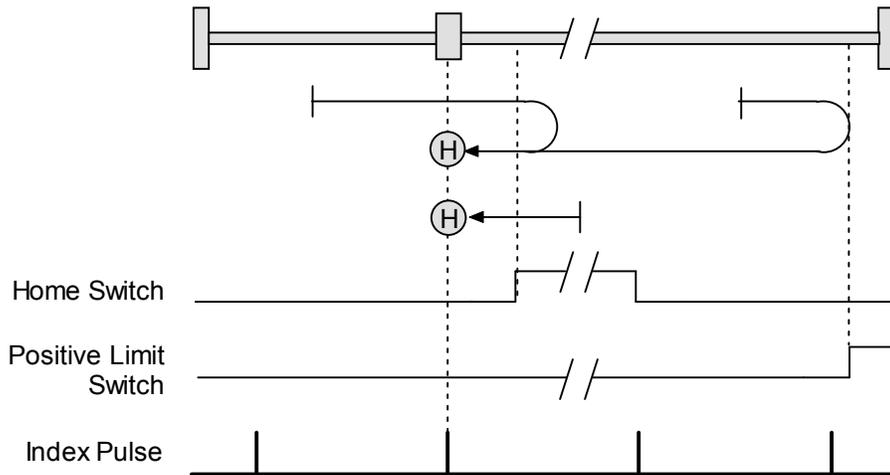
Home is the positive edge of momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home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



### B.3.12: Lower Home Outside Index

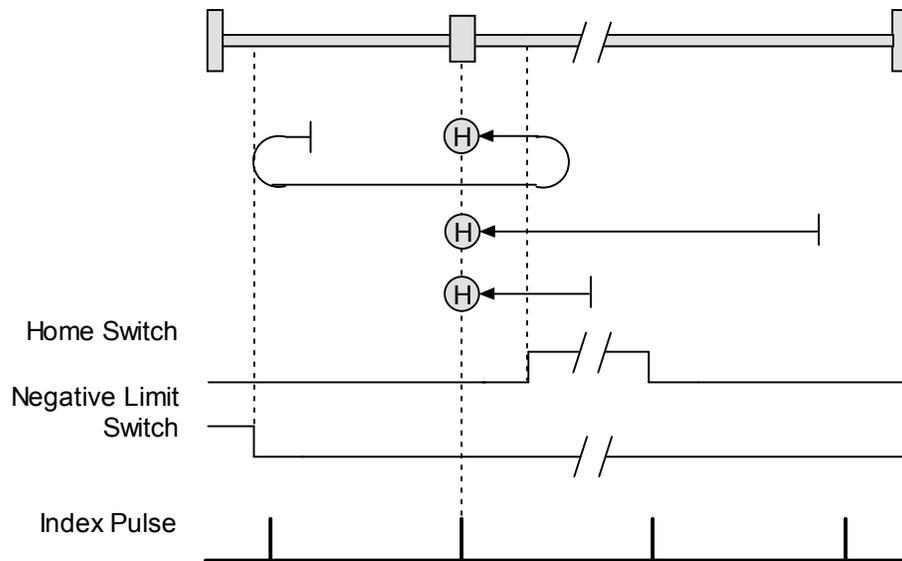
#### Direction of Motion: Positive (Homing Method Value: 803)

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 (Homing Method Value: 819)

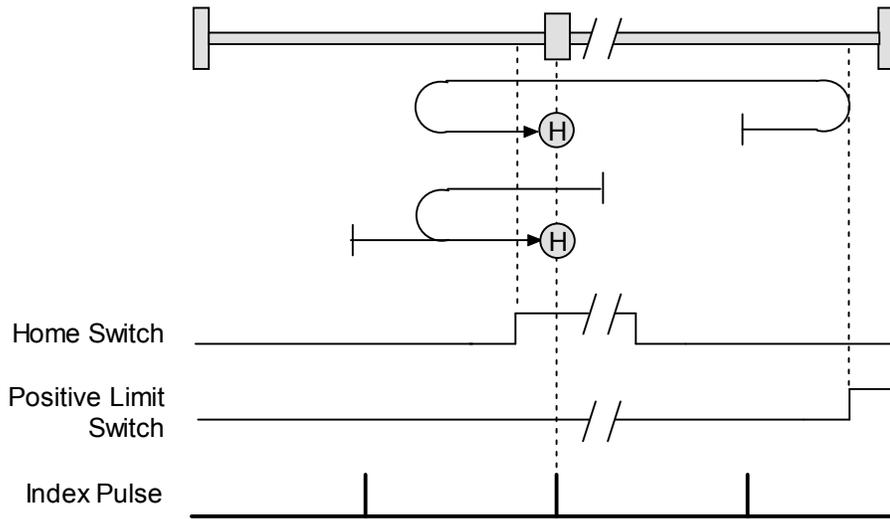
Home is the first index pulse on the negative side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated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



### B.3.13: Lower Home Inside Index

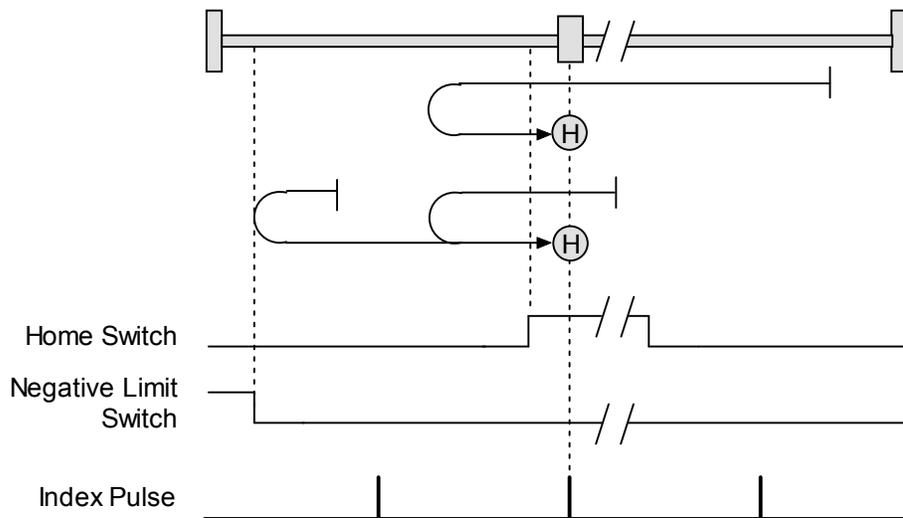
#### Direction of Motion: Positive (Homing Method Value: 867)

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 (Homing Method Value: 883)

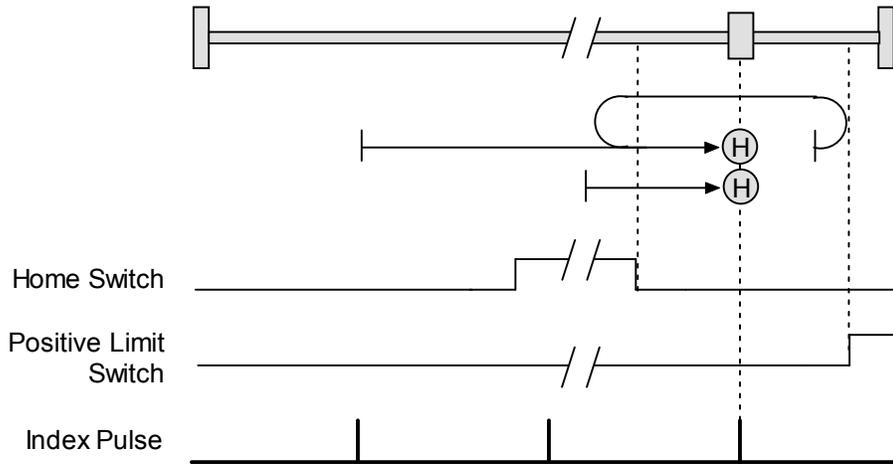
Home is the first index pulse on the positive side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated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



### B.3.14: Upper Home Outside Index

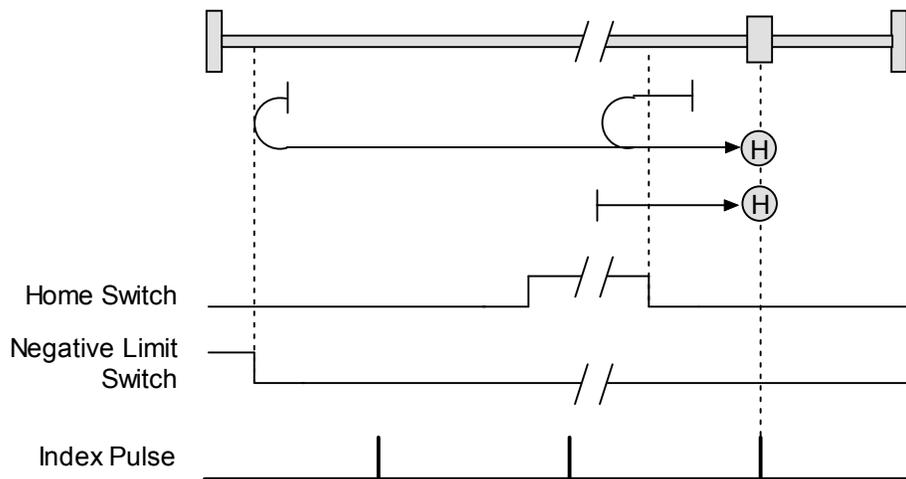
**Direction of Motion: Positive (Homing Method Value: 547)**

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 (Homing Method Value: 563)**

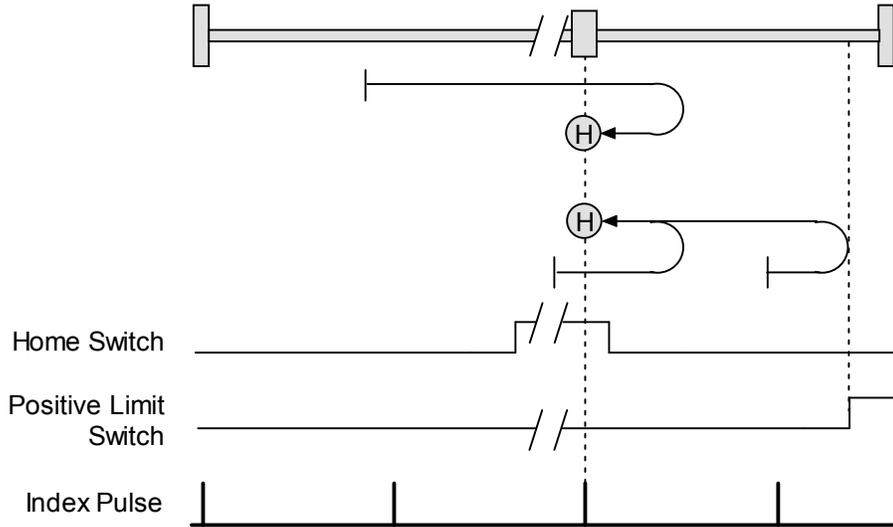
Home is the first index pulse on the positive side of the positive edge of a momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial position is right of the home position, the axis reverses on encountering the home switch.



### B.3.15: Upper Home Inside Index

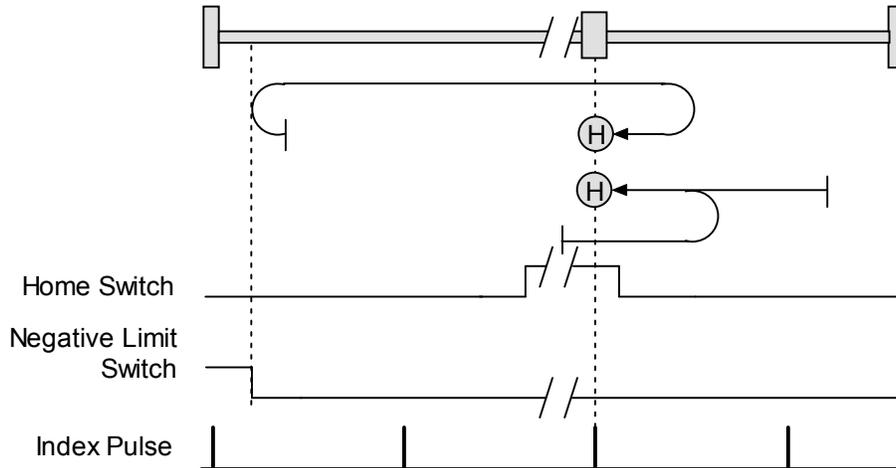
**Direction of Motion: Positive (Homing Method Value: 611)**

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 (Homing Method Value: 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.



# CHAPTER

## C: I/O MESSAGE AMPLIFIER PARAMETERS

This chapter describes the amplifier parameters that can be accessed by DeviceNet I/O messages. I/O messages access only a subset of the complete set of amplifier parameters. This subset consists largely of those variables most likely to be changed during amplifier use, as opposed to configuration parameters.

(Note that the explicit message provides access to all amplifier parameters. For the complete list of amplifier parameters, see the *Copley Controls Amplifier Parameter Dictionary*.)

## Parameter Descriptions

### Legend

**ID:** The ID to use in the Parameter ID fields of I/O messages.

**Explicit ID:** The ID to use in Explicit messages.

**R/W:** Indicates whether the parameter is read only (RO) or read/write (RW).

ID	Explicit ID	RW	Description	
0	n/a	RW	Dummy parameter. Reads return zero, writes have no effect.	
1	0xA1	RO	Amplifier Status Register. Bit-mapped as described below:	
			Bits	Description
			0	Short circuit detected.
			1	Amplifier over temperature.
			2	Over voltage.
			3	Under voltage.
			4	Motor temperature sensor active.
			5	Encoder power 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	Amp 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 is currently in a reset condition.
			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.
			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 error is outside of tracking window.
			26	Home switch is active.
			27	In motion. This bit is set if the trajectory generator is running a profile OR the position error is outside the tracking window. The bit is clear when the amplifier is settled in position.
			28	Velocity window. Set if the absolute velocity error exceeds the velocity window value.
29	Phase not yet initialized. Set when the amplifier has not yet initialized its phase while phasing with no halls.			
30, 31	Not defined.			

Continued...

...Continued:

ID	Explicit ID	RW	Description	
2	0x25	RW	Amplifier Desired State. This parameter selects between position, velocity and current modes. For most applications it should be left at the default value configured using CME 2.	
			<b>Value</b>   <b>Description</b>	
			0	Disabled.
			1	The current loop is driven by the programmed current value.
			2	The current loop is driven by the analog reference input.
			3	The current loop is driven by the PWM & direction input pins.
			4	The current loop is driven by the internal function generator.
			11	The velocity loop is driven by the programmed velocity value.
			12	The velocity loop is driven by the analog reference 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 reference 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.
			32	In microstepping mode, the position loop is driven by the analog reference input.
			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.			

Continued...

...Continued:

<b>ID</b>	<b>Explicit ID</b>	<b>RW</b>	<b>Description</b>
3	0xA7	RO	Input States. The high/low states of the amplifier's programmable digital inputs. Each bit represents an input number. Bit 0 = Digital Input 1 (IN1), Bit 1 = IN2, etc. up to IN $n$ , the number of digital inputs on the amplifier. Additional bits are ignored.  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.
4	0xAC	RW	Output States. Reads the high/low state of all output pins. Each bit represents an input number. Bit 0 = digital output 1 (OUT1), Bit 1 = OUT2, etc., up to OUT $n$ , the number of digital outputs on the amplifier. Additional bits are ignored.  Also, outputs which have been configured for program control can be set by writing to this parameter. Set a bit to activate the output. It will be activated high or low according to how it was programmed. If an output was not configured for program control it will not be affected.
5	0x39	RO	Actual Motor Current. Units: 0.01 A.
6	0x26	RO	Current Loop Limited Current. Units: 0.01 A.
7	0x03	RO	Programmed Current Value. Used only in Programmed Current mode (Amplifier Desired State = 1). Units: 0.01 A.
8	0x19	RO	Actual Motor Velocity. Units: 0.1 counts/sec.
9	0x2A	RO	Limited Velocity (output of position loop). Units: 0.1 counts/sec.
10	0x30	RO	Programmed Velocity Command. Only used in Programmed Velocity Mode (Amplifier Desired State = 11). Units: 0.1 counts/sec.
11	0x18	RO	Actual Motor Position. Units: counts.
12	0x2E	RO	Limited Position. A trajectory generator output. Units: counts.
13	0x3C	RO	Profile Velocity. A trajectory generator output. Units: 0.1 counts/sec.
14	0x3D	RO	Profile Acceleration. A trajectory generator output. Units: counts/sec <sup>2</sup> .
15	0x1E	RO	Analog Input Voltage. Some amplifier models include a +/- 10V analog input which can be read through this parameter. Units: millivolts.
16	0x1F	RO	Bus Voltage. Units: 0.1 V.
17	0x21	RO	Amplifier Temperature. Units: degrees C.

Continued...

...Continued:

ID	Explicit ID	RW	Description																																																																												
18	0xC3	RW	Homing Method. Select a homing method and direction using the values below. For full description of homing methods, see <a href="#">Homing Method Descriptions (p. 39)</a> .																																																																												
			<table border="1"> <thead> <tr> <th>Method</th> <th>Start Direction</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Set Current Position as Home</td> <td>----</td> <td>512</td> </tr> <tr> <td rowspan="2">Next Index</td> <td>Positive</td> <td>544</td> </tr> <tr> <td>Negative</td> <td>560</td> </tr> <tr> <td rowspan="2">Limit Switch</td> <td>Positive</td> <td>513</td> </tr> <tr> <td>Negative</td> <td>529</td> </tr> <tr> <td rowspan="2">Limit Switch Out to Index</td> <td>Positive</td> <td>545</td> </tr> <tr> <td>Negative</td> <td>561</td> </tr> <tr> <td rowspan="2">Home Switch</td> <td>Positive</td> <td>514</td> </tr> <tr> <td>Negative</td> <td>530</td> </tr> <tr> <td rowspan="2">Home Switch Out to Index</td> <td>Positive</td> <td>546</td> </tr> <tr> <td>Negative</td> <td>562</td> </tr> <tr> <td rowspan="2">Home Switch In to Index</td> <td>Positive</td> <td>610</td> </tr> <tr> <td>Negative</td> <td>626</td> </tr> <tr> <td rowspan="2">Hard Stop</td> <td>Positive</td> <td>516</td> </tr> <tr> <td>Negative</td> <td>532</td> </tr> <tr> <td rowspan="2">Hard Stop Out to Index</td> <td>Positive</td> <td>548</td> </tr> <tr> <td>Negative</td> <td>564</td> </tr> <tr> <td rowspan="2">Lower Home</td> <td>Positive</td> <td>771</td> </tr> <tr> <td>Negative</td> <td>787</td> </tr> <tr> <td rowspan="2">Upper Home</td> <td>Positive</td> <td>515</td> </tr> <tr> <td>Negative</td> <td>531</td> </tr> <tr> <td rowspan="2">Lower Home Outside Index</td> <td>Positive</td> <td>803</td> </tr> <tr> <td>Negative</td> <td>819</td> </tr> <tr> <td rowspan="2">Lower Home Inside Index</td> <td>Positive</td> <td>867</td> </tr> <tr> <td>Negative</td> <td>883</td> </tr> <tr> <td rowspan="2">Upper Home Outside Index</td> <td>Positive</td> <td>547</td> </tr> <tr> <td>Negative</td> <td>563</td> </tr> <tr> <td rowspan="2">Upper Home Inside Index</td> <td>Positive</td> <td>611</td> </tr> <tr> <td>Negative</td> <td>627</td> </tr> </tbody> </table>	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
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																																																																													
	0xC4	RW	Homing Velocity (fast moves). Units: 0.1 counts/sec.																																																																												
20	0xC5	RW	Homing Velocity (slow moves). Units: 0.1 counts/sec.																																																																												
21	0xC6	RW	Homing Acceleration/Deceleration. Units: 10 counts/second <sup>2</sup> .																																																																												
22	0xC7	RW	Home Offset. Units: counts.																																																																												
23	0xC8	RW	Homing Current Limit (used with home to hard stop mode only). Units: 0.01 A																																																																												
24	0xC0	RW	Homing Current Delay Time (used with home to hard stop mode only). Units: milliseconds.																																																																												
25	0xC9	RW	Trajectory 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.																																																																												

Continued...

...Continued:

<b>ID</b>	<b>Explicit ID</b>	<b>RW</b>	<b>Description</b>														
26	0xCA	RO	Trajectory Status Register. Bit-mapped as shown here: <table border="1" data-bbox="472 296 1349 653"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-10</td> <td>Reserved for future use.</td> </tr> <tr> <td>11</td> <td>Homing error. If set an error occurred in the last home attempt. Cleared by a home command.</td> </tr> <tr> <td>12</td> <td>Referenced. Set if a homing command has been successfully executed. Cleared by a home command.</td> </tr> <tr> <td>13</td> <td>Homing. Set when the amplifier is running a home command.</td> </tr> <tr> <td>14</td> <td>Set when a move is aborted. Cleared at the start of the next move.</td> </tr> <tr> <td>15</td> <td>In-Motion Bit. If set, the trajectory generator is presently generating a profile.</td> </tr> </tbody> </table>	Bit	Description	0-10	Reserved for future use.	11	Homing error. If set an error occurred in the last home attempt. Cleared by a home command.	12	Referenced. Set if a homing command has been successfully executed. Cleared by a home command.	13	Homing. Set when 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.
Bit	Description																
0-10	Reserved for future use.																
11	Homing error. If set an error occurred in the last home attempt. Cleared by a home command.																
12	Referenced. Set if a homing command has been successfully executed. Cleared by a home command.																
13	Homing. Set when 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.																
27	0xCB	RW	Position Command. Units: Counts. Meaning depends on move type: 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.														
28	0xCC	RW	Maximum Velocity. Units: 0.1 counts/sec.														
29	0xCD	RW	Maximum Acceleration Rate. Units: 10 counts/second <sup>2</sup> .														
30	0xCE	RW	Maximum Deceleration Rate. Units: 10 counts/second <sup>2</sup> .														
31	0xCF	RW	Maximum Jerk Rate. Units: 100 counts/ second <sup>3</sup> .														
32	0x3E	RO	Commanded Position. Units: Counts.														
33	0xB9	RW	Positive Software Limit. Units: Counts.														
34	0xBA	RW	Negative Software Limit. Units: Counts.														
35	0xBF	RW	Software Limit Acceleration. Units: 10 counts/second <sup>2</sup> .														

Continued...

...Continued:

<b>ID</b>	<b>Explicit ID</b>	<b>RW</b>	<b>Description</b>																																																																				
36	n/a	RO	Latched I/O Command Error Code. When an I/O command causes one of the following amplifier errors, the respective code is returned in this parameter. The I/O response message Error Bit remains set until cleared by a 0-1 transition of the command message Clear Error Bit.																																																																				
			<table border="1"> <thead> <tr> <th><b>Code</b></th> <th><b>Description</b></th> </tr> </thead> <tbody> <tr><td>0</td><td>No Error. The command was performed successfully.</td></tr> <tr><td>1</td><td>Too much data passed with command.</td></tr> <tr><td>2</td><td>Bad checksum on input command.</td></tr> <tr><td>3</td><td>Unknown command code.</td></tr> <tr><td>4</td><td>Not enough data was supplied with the command.</td></tr> <tr><td>5</td><td>Too much data was supplied with the command.</td></tr> <tr><td>6</td><td>Flash erase failed.</td></tr> <tr><td>7</td><td>Flash write failed.</td></tr> <tr><td>8</td><td>Attempt to access a bad memory group.</td></tr> <tr><td>9</td><td>Unknown variable ID.</td></tr> <tr><td>10</td><td>Data value out of range.</td></tr> <tr><td>11</td><td>Attempt to modify read-only variable</td></tr> <tr><td>14</td><td>Unknown axis state.</td></tr> <tr><td>15</td><td>Variable doesn't exist on requested page.</td></tr> <tr><td>16</td><td>Reserved.</td></tr> <tr><td>17</td><td>Data flash failed CRC check.</td></tr> <tr><td>18</td><td>Illegal attempt to start a move while currently moving</td></tr> <tr><td>19</td><td>Illegal velocity limit for move.</td></tr> <tr><td>20</td><td>Illegal acceleration limit for move.</td></tr> <tr><td>21</td><td>Illegal deceleration limit for move.</td></tr> <tr><td>22</td><td>Illegal jerk limit for move.</td></tr> <tr><td>23</td><td>Trajectory buffer under flowed during move.</td></tr> <tr><td>24</td><td>Trajectory buffer overflowed when adding data.</td></tr> <tr><td>25</td><td>Invalid trajectory mode.</td></tr> <tr><td>26</td><td>CVM Program location already in use.</td></tr> <tr><td>27</td><td>Command is not allowed while CVM is running</td></tr> <tr><td>28</td><td>CVM Program too big to upload.</td></tr> <tr><td>29</td><td>CVM file system error.</td></tr> <tr><td>30</td><td>Specified CVM Program does not exist.</td></tr> <tr><td>31</td><td>Invalid node ID for serial port forwarding.</td></tr> <tr><td>32</td><td>CAN Network communications failure.</td></tr> <tr><td>33</td><td>ASCII command parsing error.</td></tr> <tr><td>34</td><td>Internal amplifier error.</td></tr> </tbody> </table>	<b>Code</b>	<b>Description</b>	0	No Error. The command was performed successfully.	1	Too much data passed with command.	2	Bad checksum on input command.	3	Unknown command code.	4	Not enough data was supplied with the command.	5	Too much data was supplied with the command.	6	Flash erase failed.	7	Flash write failed.	8	Attempt to access a bad memory group.	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.	16	Reserved.	17	Data flash failed CRC check.	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.	23	Trajectory buffer under flowed during move.	24	Trajectory buffer overflowed when adding data.	25	Invalid trajectory mode.	26	CVM Program location already in use.	27	Command is not allowed while CVM is running	28	CVM Program too big to upload.	29	CVM file system error.	30	Specified CVM Program does not exist.	31	Invalid node ID for serial port forwarding.	32	CAN Network communications failure.	33	ASCII command parsing error.	34	Internal amplifier error.
<b>Code</b>	<b>Description</b>																																																																						
0	No Error. The command was performed successfully.																																																																						
1	Too much data passed with command.																																																																						
2	Bad checksum on input command.																																																																						
3	Unknown command code.																																																																						
4	Not enough data was supplied with the command.																																																																						
5	Too much data was supplied with the command.																																																																						
6	Flash erase failed.																																																																						
7	Flash write failed.																																																																						
8	Attempt to access a bad memory group.																																																																						
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.																																																																						
16	Reserved.																																																																						
17	Data flash failed CRC check.																																																																						
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.																																																																						
23	Trajectory buffer under flowed during move.																																																																						
24	Trajectory buffer overflowed when adding data.																																																																						
25	Invalid trajectory mode.																																																																						
26	CVM Program location already in use.																																																																						
27	Command is not allowed while CVM is running																																																																						
28	CVM Program too big to upload.																																																																						
29	CVM file system error.																																																																						
30	Specified CVM Program does not exist.																																																																						
31	Invalid node ID for serial port forwarding.																																																																						
32	CAN Network communications failure.																																																																						
33	ASCII command parsing error.																																																																						
34	Internal amplifier error.																																																																						

Continued...

...Continued:

ID	Explicit ID	RW	Description																								
37	0xA5	RO	<p>Fault Register. Bit-mapped to show latching faults that have occurred in the amplifier. When a latching fault is present, the I/O response message Fault Bit remains set. The faults can be cleared by a 0-1 transition of the command message Clear Faults Bit.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Fault Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Fatal hardware error: the flash data is corrupt (CRC error).</td> </tr> <tr> <td>1</td> <td>Fatal hardware error: A/D offset is out of range.</td> </tr> <tr> <td>2</td> <td>Short Circuit.</td> </tr> <tr> <td>3</td> <td>Amp Over Temperature.</td> </tr> <tr> <td>4</td> <td>Motor Over Temperature.</td> </tr> <tr> <td>5</td> <td>Over Voltage.</td> </tr> <tr> <td>6</td> <td>Under Voltage.</td> </tr> <tr> <td>7</td> <td>Feedback Error.</td> </tr> <tr> <td>8</td> <td>Motor Phasing Error.</td> </tr> <tr> <td>9</td> <td>Following Error.</td> </tr> <tr> <td>10</td> <td>Over Current (Latched).</td> </tr> </tbody> </table>	Bit	Fault Description	0	Fatal hardware error: the flash data is corrupt (CRC error).	1	Fatal hardware error: A/D offset is out of range.	2	Short Circuit.	3	Amp Over Temperature.	4	Motor Over Temperature.	5	Over Voltage.	6	Under Voltage.	7	Feedback Error.	8	Motor Phasing Error.	9	Following Error.	10	Over Current (Latched).
Bit	Fault Description																										
0	Fatal hardware error: the flash data is corrupt (CRC error).																										
1	Fatal hardware error: A/D offset is out of range.																										
2	Short Circuit.																										
3	Amp Over Temperature.																										
4	Motor Over Temperature.																										
5	Over Voltage.																										
6	Under Voltage.																										
7	Feedback Error.																										
8	Motor Phasing Error.																										
9	Following Error.																										
10	Over Current (Latched).																										
38	0x6D	RW	<p>Position Capture Control Register. Sets up position capture based on the index or home switch input. Bit-mapped as follows:</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>If set, the index position is captured on the falling edge of the index.</td> </tr> <tr> <td>1</td> <td>If set, the index position is captured on the rising edge of the index.</td> </tr> <tr> <td>2</td> <td>If set, a captured index position will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.</td> </tr> <tr> <td>3,4</td> <td>Reserved.</td> </tr> <tr> <td>5</td> <td>If set, capture falling edges of the home switch input transition; if clear, capture rising edges.</td> </tr> <tr> <td>6</td> <td>If set, a captured home switch input position will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.</td> </tr> </tbody> </table>	Bit	Description	0	If set, the index position is captured on the falling edge of the index.	1	If set, the index position is captured on the rising edge of the index.	2	If set, a captured index position will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.	3,4	Reserved.	5	If set, capture falling edges of the home switch input transition; if clear, capture rising edges.	6	If set, a captured home switch input position will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.										
Bit	Description																										
0	If set, the index position is captured on the falling edge of the index.																										
1	If set, the index position is captured on the rising edge of the index.																										
2	If set, a captured index position will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.																										
3,4	Reserved.																										
5	If set, capture falling edges of the home switch input transition; if clear, capture rising edges.																										
6	If set, a captured home switch input position will not be overwritten by a new position until it has been read. If clear, new positions will overwrite old positions.																										
39	0x6E	RO	<p>Position Capture Status Register. Shows the current status of the index or home switch capture mechanism. Bit-mapped as follows:</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>If set, an index position has been captured. Cleared when the captured position is read.</td> </tr> <tr> <td>1,2</td> <td>Reserved.</td> </tr> <tr> <td>3</td> <td>If set, a new index transition occurred when a captured position was already stored. Depending on the mode programmed in Bit 2 of the Position Capture Control Register, the new position may have been discarded, or overwritten the previously stored position.</td> </tr> <tr> <td>4</td> <td>If set, new home switch transition data has been captured.</td> </tr> <tr> <td>5,6</td> <td>Reserved.</td> </tr> <tr> <td>7</td> <td>If set, a new home switch input transition occurred when a captured position was already stored. Depending on the mode programmed in Bit 6 of the Position Capture Control Register, the new position may have been discarded, or overwritten the previously stored position.</td> </tr> </tbody> </table>	Bit	Description	0	If set, an index position has been captured. Cleared when the captured position is read.	1,2	Reserved.	3	If set, a new index transition occurred when a captured position was already stored. Depending on the mode programmed in Bit 2 of the Position Capture Control Register, the new position may have been discarded, or overwritten the previously stored position.	4	If set, new home switch transition data has been captured.	5,6	Reserved.	7	If set, a new home switch input transition occurred when a captured position was already stored. Depending on the mode programmed in Bit 6 of the Position Capture Control Register, the new position may have been discarded, or overwritten the previously stored position.										
Bit	Description																										
0	If set, an index position has been captured. Cleared when the captured position is read.																										
1,2	Reserved.																										
3	If set, a new index transition occurred when a captured position was already stored. Depending on the mode programmed in Bit 2 of the Position Capture Control Register, the new position may have been discarded, or overwritten the previously stored position.																										
4	If set, new home switch transition data has been captured.																										
5,6	Reserved.																										
7	If set, a new home switch input transition occurred when a captured position was already stored. Depending on the mode programmed in Bit 6 of the Position Capture Control Register, the new position may have been discarded, or overwritten the previously stored position.																										
40	0x69	RO	<p>Captured Index Position. Units: counts. Provides the position that the axis was in when an index pulse was captured. Can be configured by setting bits in the Position Capture Control Register, and the status of the captured data can be checked in the Position Capture Status Register. Reading this variable resets bits 0 &amp; 3 of the Position Capture Status Register.</p>																								
41	0x10B	RO	<p>Captured Home Position. Units: counts. Provides the position that the axis was in when an input pin configured as a home switch input became active. Can be configured by setting bits in the Position Capture Control Register, and the status of the captured data can be checked in the Position Capture Status Register. Reading this variable resets bits 4 &amp; 7 of the Position Capture Status Register.</p>																								

Continued...

...Continued:

ID	Explicit ID	RW	Description																										
42	0x106	RW	<p>Camming Configuration. Configures camming feature. Bit-mapped as follows:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0-3</td> <td>Number of the Cam Table to use (0-9).</td> </tr> <tr> <td>4,5</td> <td>Reserved</td> </tr> <tr> <td>6</td> <td>If set, use the internal generator. If clear, use digital command input as configured in using CME 2 Camming controls or explicit message variable 0xA7. The internal generator runs at constant velocity programmed in variable 0x109.</td> </tr> <tr> <td>7</td> <td>Reserved.</td> </tr> <tr> <td>8-11</td> <td>Input number to use as trigger. Note: a value of 0 selects input 1, value of 1 selects input 2, etc.</td> </tr> <tr> <td>12-13</td> <td>Camming trigger type:</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Code</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>None (Continuous): The active Cam Table is repeated continuously.</td> </tr> <tr> <td>1</td> <td>Use Input, Edge: The active Cam Table begins executing on the rising edge of the input pin selected by bits 8-11.</td> </tr> <tr> <td>2</td> <td>Use Input, Level: The active Cam Table will run as long as the input selected by bits 8-11 is high.</td> </tr> <tr> <td>3</td> <td>Use Master (Secondary) Encoder Index: The active Cam Table is executed when the amplifier receives an index pulse from the Master encoder. Index pulses received during execution are ignored.</td> </tr> </tbody> </table> </td> </tr> </tbody> </table> <p>For an example, see <a href="#">Select CAM Table and Trigger (p. 26)</a>. For more information, see the <i>Copley Camming User Guide</i>.</p>	Bits	Description	0-3	Number of the Cam Table to use (0-9).	4,5	Reserved	6	If set, use the internal generator. If clear, use digital command input as configured in using CME 2 Camming controls or explicit message variable 0xA7. The internal generator runs at constant velocity programmed in variable 0x109.	7	Reserved.	8-11	Input number to use as trigger. Note: a value of 0 selects input 1, value of 1 selects input 2, etc.	12-13	Camming trigger type:		<table border="1"> <thead> <tr> <th>Code</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>None (Continuous): The active Cam Table is repeated continuously.</td> </tr> <tr> <td>1</td> <td>Use Input, Edge: The active Cam Table begins executing on the rising edge of the input pin selected by bits 8-11.</td> </tr> <tr> <td>2</td> <td>Use Input, Level: The active Cam Table will run as long as the input selected by bits 8-11 is high.</td> </tr> <tr> <td>3</td> <td>Use Master (Secondary) Encoder Index: The active Cam Table is executed when the amplifier receives an index pulse from the Master encoder. Index pulses received during execution are ignored.</td> </tr> </tbody> </table>	Code	Type	0	None (Continuous): The active Cam Table is repeated continuously.	1	Use Input, Edge: The active Cam Table begins executing on the rising edge of the input pin selected by bits 8-11.	2	Use Input, Level: The active Cam Table will run as long as the input selected by bits 8-11 is high.	3	Use Master (Secondary) Encoder Index: The active Cam Table is executed when the amplifier receives an index pulse from the Master encoder. Index pulses received during execution are ignored.
Bits	Description																												
0-3	Number of the Cam Table to use (0-9).																												
4,5	Reserved																												
6	If set, use the internal generator. If clear, use digital command input as configured in using CME 2 Camming controls or explicit message variable 0xA7. The internal generator runs at constant velocity programmed in variable 0x109.																												
7	Reserved.																												
8-11	Input number to use as trigger. Note: a value of 0 selects input 1, value of 1 selects input 2, etc.																												
12-13	Camming trigger type:																												
	<table border="1"> <thead> <tr> <th>Code</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>None (Continuous): The active Cam Table is repeated continuously.</td> </tr> <tr> <td>1</td> <td>Use Input, Edge: The active Cam Table begins executing on the rising edge of the input pin selected by bits 8-11.</td> </tr> <tr> <td>2</td> <td>Use Input, Level: The active Cam Table will run as long as the input selected by bits 8-11 is high.</td> </tr> <tr> <td>3</td> <td>Use Master (Secondary) Encoder Index: The active Cam Table is executed when the amplifier receives an index pulse from the Master encoder. Index pulses received during execution are ignored.</td> </tr> </tbody> </table>	Code	Type	0	None (Continuous): The active Cam Table is repeated continuously.	1	Use Input, Edge: The active Cam Table begins executing on the rising edge of the input pin selected by bits 8-11.	2	Use Input, Level: The active Cam Table will run as long as the input selected by bits 8-11 is high.	3	Use Master (Secondary) Encoder Index: The active Cam Table is executed when the amplifier receives an index pulse from the Master encoder. Index pulses received during execution are ignored.																		
Code	Type																												
0	None (Continuous): The active Cam Table is repeated continuously.																												
1	Use Input, Edge: The active Cam Table begins executing on the rising edge of the input pin selected by bits 8-11.																												
2	Use Input, Level: The active Cam Table will run as long as the input selected by bits 8-11 is high.																												
3	Use Master (Secondary) Encoder Index: The active Cam Table is executed when the amplifier receives an index pulse from the Master encoder. Index pulses received during execution are ignored.																												
43	0x107	RW	Camming Delay Forward. Units: master command counts. For an example, see <a href="#">Adjust Camming Delay (p. 27)</a> .																										
44	0x108	RW	Camming Delay Reverse. Units: master command counts.																										
45	0x109	RW	Camming Master Velocity. Units: 0.1 counts/sec.																										

Copley DeviceNet Programmer's Guide  
P/N 95-00658-000

Revision 2

June 2008

© 2006, 2007, 2008  
Copley Controls Corporation  
20 Dan Road  
Canton, MA 02021 USA  
All rights reserved