Copley Indexer 2 Program User Guide

P/N 95-00744-000
Revision 2
June 2008

TABLE OF CONTENTS

ABOUT THIS MANUAL

Overview and Scope

This manual describes the operation of the Copley Virtual Machine Indexer 2 Program developed by Copley Controls Corporation. The manual was written for the reader who has a basic knowledge of motion control theory and operation, Copley Controls amplifiers, and Copley Controls CME 2 software.

Related Documentation

See the user guides and data sheets for the amplifiers you will operate with CME 2 software. These documents can be found at http://www.copleycontrols.com/Motion/Downloads/index.html

Also see the *CME 2 User Guide.*

Users who intend to access Indexer 2 registers via CANopen, DeviceNet, or the Copley ASCII interface should consult, respectively, the *CANopen Programmer's Manual,* the *Copley DeviceNet Programmer's Guide,* and the *Copley ASCII Interface Programmer's Guide.*

DANGER

DANGER: Hazardous voltages.

Exercise caution when installing and adjusting Copley Controls amplifiers.

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

DANGER

Risk of electric shock.

High-voltage circuits are connected to DC power on certain Copley Controls amplifiers.

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

DANGER

Using CME 2 can affect or suspend Indexer 2 Program operations.

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

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

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

DANGER

Moves in progress will run to completion after Indexer 2 Program is stopped.

Stopping the Indexer 2 Program does not stop any move in progress.

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

Amplifier faults and sequence errors are not equivalent.

Amplifier faults and sequence errors are not equivalent.

WARNING

A sequence error does not necessarily result in an amplifier fault condition. Each Indexer step function can result in a particular set of sequence errors, as described in Functions (p. 37).

An amplifier fault condition does not necessarily result in a sequence error. Thus it is possible that amplifier faults can occur without triggering any error handling response in the Indexer 2 Program.

Be aware of the differences between amplifier faults and sequence errors and program accordingly.

Revision History

Revision	Date	DECO #	Comments
1	June 2007	15515	Adapted from *Copley Indexer Program User Guide* Version 2.
2	June 2008	16708	Various updates.

CHAPTER 1: INTRODUCTION

This chapter provides an overview of the Copley Controls Indexer 2 Program. Contents include:

1.1: Copley Controls Indexer 2 Program

The Copley Controls Indexer 2 Program, combined with a Copley Controls amplifier, creates a powerful single axis indexer that can be programmed by anyone with basic computer and motion control knowledge. Working with a PC and the tools built into Copley Controls CME 2 software, the user configures and programs the Indexer 2 Program and then downloads it to the amplifier. On the amplifier, the Indexer 2 Program is run on the Copley Virtual Machine (CVM), an embedded virtual programmable controller.

The user can create up to 32 sequences. A sequence contains one or more steps that can combine homing instructions, moves, gains adjustments, and timed delays. Sequence steps can also apply conditional logic and jumping, control the amplifier's digital outputs, and monitor the digital inputs.

In the simplest applications, a PLC or switch activates the amplifier's digital inputs to select and execute sequences. The amplifier's digital outputs can be used to control machine processes or provide status feedback to the PLC.

For more complex applications, the Indexer 2 Program has 32 registers that control programs can access by issuing ASCII commands over the amplifier's RS-232 serial port. These registers can be used to select and execute sequences and pass numeric parameters to the Indexer 2 Program. The registers may also be accessed by the amplifier's other supported control networks such as CANopen or DeviceNet.

Multi-axis applications are supported by the Copley Controls multi-drop feature. In a multi-drop configuration, an amplifier with a serial connection to an external controller serves as a gateway to multiple amplifiers linked to it by CAN bus connections. This makes it possible for a PLC to control up to 128 axes of motion through one serial port.

1.2: Indexer Features

Features of the Indexer 2 Program include:

- Simple, intuitive programming tools
- 32 programmable multi-step sequences
- Priority sequence, selected and executed with a single digital input
- 32 registers accessible via an RS-232 serial port and other interfaces
- Sequence selection and execution by digital input or register
- Program can start automatically on power up
- Programmable response to sequence errors
- Standard functions include:
  - Move (position, velocity, current, camming, home)
  - Wait (for input, delay time, position, move done, parameter, or event)
  - Conditional jumping
  - Set Gains, limits, and tracking windows
  - Set output, position triggered output
  - Disable amplifier
  - Math
  - Set, get parameters
  - Velocity single pole out filter
  - Set operating mode

1.3: Amplifier Support

The Indexer 2 Program is bundled with Copley Controls amplifier commissioning software, CME 2, version 5.1 or higher. Firmware requirements are described below:

Amplifier Model		Recommended Minimum Firmware Version
Accelnet Series	ACK / R23	5.46*
	ACM / R22	
	ACJ(S) R21(S)	
	ACP	
	AMP	1.36
	AEP	
	ADP / R20	
Xenus Series	XSL(R)	5.46*
	XTL(R / S) R11(R / S)	1.36
	XSJ(R / S)	
Stepnet Series	STL	5.46*
	STM	
	STP	

Notes:

* Accelnet_X.XX.cff or Devicenet_X.XX.cff firmware files are common and can be used with any drive in this family.

.

CHAPTER 2: DESCRIPTION AND OPERATION

This chapter provides an overview of how the Indexer 2 Program operates. Contents include:

2.1: Operational Overview

On startup, the Indexer 2 Program zeros its program registers and enters its main loop, continually polling first the user defined Priority input (if configured) and then the user-defined Go input or register. Optionally, one sequence can be selected to execute on power up or amplifier reset.

If a Priority input or Go command is active, the Indexer 2 Program executes the selected sequence. The program puts the amplifier in the correct mode of operation to execute each step of the sequence as it progresses. During sequence execution, the Priority, Go, and sequence selection inputs are ignored. After successful completion of a sequence, the Indexer 2 Program returns to the main loop.

If an error occurs during a sequence, one of two responses can be programmed for each sequence:

- The current sequence is aborted and the Indexer 2 Program reverts back to its main loop, or
- The Indexer 2 Program starts immediate execution of another programmed sequence. At the end of this sequence, the program returns to its main loop.

2.2: Sequence Selection

The Indexer 2 Program can store up to 32 sequences (0 to 31). Upon receiving a Go command, the program executes the sequence that has been selected by a register or by digital inputs.

2.2.1: Use a Register to Select a Sequence

When the Indexer 2 Program receives a Go command and **Use register to select sequence** is the programmed sequence selection method, the program reads the first 5 bits (0-4) of the specified register to select a sequence. To select a sequence, write to the register a hex value (0x00 - 0x1f) or a decimal value (0 - 31). To write to a register using ASCII commands, see Reading and Writing Registers (p. 77).

2.2.2: Use Digital Inputs to Select a Sequence

DANGER

Conflicting use of inputs is possible.

It is possible to assign functions to the inputs though the CME 2 Input/Output configuration screen that would conflict with the use of the BCD sequence selection, Priority and Go inputs assigned by Indexer 2 Program. Both functions of the input would be in effect simultaneously and might cause unexpected results. Exercise care to avoid such conflicts when assigning input functions.

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

When the Indexer 2 Program receives a Go command and **Use digital input** is the programmed sequence selection method, the amplifier's digital inputs select the sequence to execute. Up to five inputs can be used to represent a binary coded number with a decimal value between 0 and 31.

The table below shows the number of input-selectable sequences per number of inputs used.

# of BCD Inputs	# of Sequences	Sequence # Range
5	32	0-31
4	16	0-15
3	8	0-7
2	4	0-3
1	2	0-1
0	1	0

For instance, suppose the Indexer 2 Program has been configured to read three sequence selection inputs starting at IN2. This table shows the input states that would select each:

	Inputs		
Input Number	IN4	IN3	IN2
Decimal Equivalent	4	2	1
Sequence			
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0

2.3: Go Command

A Go command causes Indexer 2 to execute the selected sequence. Programmable Go command trigger options include registers, digital inputs, and immediate single Go on startup or reset.

2.3.1: Use a Register to Initiate Go

When programmed to **Use Register to Initiate Go**, the Indexer 2 Program monitors bit 15 of the programmed Go register. If Bit 15 is set to 1 the program executes the selected sequence.

To initiate a register-driven Go when the sequence will be selected by digital inputs, set Bit 15 of the Go register by writing to the Go register a hex value 0x8000, or the decimal equivalent, 32768. (All other bits will be ignored.) To write to a register using ASCII commands, see Reading and Writing Registers (p. 77).

When Go command and sequence selection are both programmed to **Use Register**, the same register is used for both purposes. In this case, use the same write operation to write the sequence to the register and set Bit 15 at the same time. For instance, to execute Sequence 12, write to the register the hex value 0x800c or the decimal equivalent, 32780.

Note that the Indexer 2 Program clears Bit 15 before executing the sequence so that a register or digital input cannot trigger another Go until Bit 15 is reset.
(The state of Bit 15 does not affect the operation of Immediate Single Go.)

2.3.2: Use a Digital Input to Initiate Go

When programmed to **Use Input to Initiate Go**, the Indexer 2 Program monitors the state of the programmed Go input. The Go can be programmed to happen whenever the input is at the specified level (low or high), or only on the rising or falling edge of an input transition.

2.3.3: Use Immediate Single Go on Startup or Reset

This option programs the Indexer to trigger a Go command when the amplifier is powered up or reset or the Run button is pushed. A sequence is selected according to the programmed sequence selection choices.

If the Indexer is programmed to **Use register to select sequence**, then the sequence executed will be number 0, because all registers are cleared when the program starts.

If sequence selection is programmed to **Use digital input,** then the states of the programmed selection inputs determine which sequence is executed.

After immediate single Go, further Go commands are executed as programmed.

2.4: Summary: Sequence Selection and Go Choices

Either of the sequence selection methods (register or input) can be used with either of the Go initiation methods (register or input) in any combination, as shown below.

		Initiate Go command with:	
		Inputs	**Register**
Select sequence with:	**Inputs**	• Set the inputs to select the sequence. • Activate Go input	• Set the inputs to select the sequence. • Set Bit 15 of Go register
	Register	• Write sequence number to the sequence register. • Activate Go input	• Write the sequence number and set Bit 15 of the Sequence/Go register.

2.5: Priority Input Command

In addition to the 32 normal sequences, the Indexer 2 Program supports a Priority sequence. Unlike other sequences, the Priority sequence:

- is selected and initiated with a single digital input
- receives process priority over other sequence selection methods

Note that a Priority sequence does not stop or override any sequence in progress.

2.6: Indexer 2 Program Registers

The Indexer 2 Program has 32 registers that can be used to select sequences, initiate Go commands, and pass numeric parameters to functions in sequence steps. Each register is 32 bits long.

Register references can be used to pass numeric parameters, such as gains and trajectory settings, to most Indexer 2 Program functions. A register reference takes the form R*n*, where *n* is a register number (0-31). For more details see Using Registers to Pass Values to Functions (p. 38).

Control applications (HMI, PLC, or PC-based programs) can use any of the supported protocols to read and write the Indexer 2 Program registers. Supported protocols include the Copley ASCII Interface, CANopen, and DeviceNet.

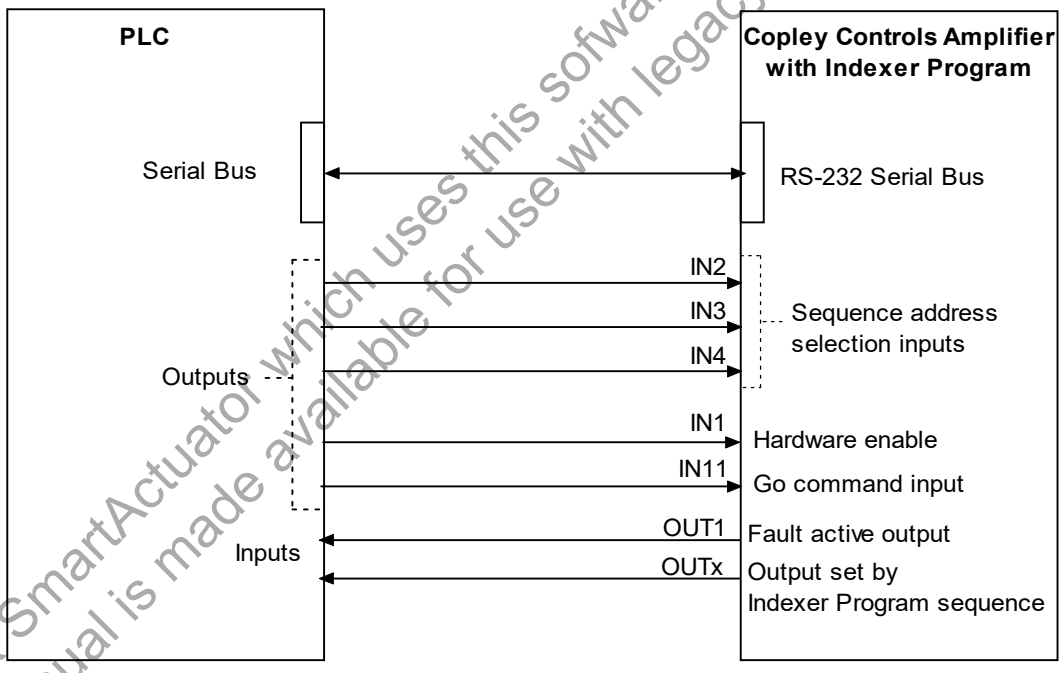
For experimentation and simple setup and control, the CME 2 **Tools→ASCII Command Line** feature can also be used, as can a telnet device such as the standard Microsoft Windows HyperTerminal. See ASCII Commands Over Serial (p. 75).

Current register values can also be viewed while the Indexer 2 Program is running by using the CME 2 CVM Control Program **View→Register Values** command.

NOTE: Starting the Indexer 2 Program always sets all register values to zero.

2.7: Diagram of Typical Connections

In a typical Indexer 2 Program application, the connections between a Copley Controls amplifier and a PLC resemble those shown below.



Here, inputs IN2-IN4 are used for sequence selection. Input IN11 is programmed to initiate the Indexer 2 Program **Go** command. The amplifier's hardware enable input, IN1, is also controlled by the PLC. OUT1 is configured to go active when an amplifier fault occurs, and any of the amplifier's digital outputs can be set by a step in an Indexer 2 Program sequence. The serial connection can be used to set register values.

CHAPTER 3: PROGRAMMING

This chapter describes how to configure the Indexer 2 Program. Contents include:

3.1: Overview of Procedures

The following is a typical series of basic steps for creating and executing a sequence.

Basic Step	Description/More Details
Set the amplifier to take input from the CVM Control program.	See Basic Amplifier Setup (p. 21).
Set up, tune and test the amplifier.	The system should be able to safely perform the types of moves that the Indexer 2 Program will command.
Save amplifier settings to flash	Many amplifier flash parameters are used by Indexer 2 Program for default values.
Access the Indexer 2 Program.	See Accessing the Indexer 2 Program (p. 23).
Configure sequence selection, priority, and Go command.	See Setting Up Sequence Selection and Go Command (p. 27).
Program the sequences.	See Programming (p. 19) and Functions (p. 37).
Save the Indexer 2 Program to the amplifier.	See Save Control Program to amplifier flash memory (p. 26).
Save the Indexer 2 Program to disk.	 Click the **Save Control Program to disk** tool and use the navigation screen to save a copy of the program.
Make sure the amplifier is hardware enabled and free of active faults.	See the *CME 2 User Guide*.
Run the Indexer 2 Program	See Toolbar Functions (p. 26).
Select the appropriate sequence.	Set the inputs or write to the Indexer 2 Program register.
Activate the **Go** command.	Set the input or write to the Indexer 2 Program register.
Debug the Indexer 2 Program as required.	See Using Single-Step/Debug Mode (p. 34).
Stop the Indexer 2 Program.	If it becomes necessary to stop the Indexer 2 Program, use the **Stop Control Program** command. See Toolbar Functions (p. 26). CAUTION: Programmed moves in progress will continue until finished.
Configure the Indexer 2 Program for auto start if required	See Enable Control Program on Startup (p. 25).
Save the final Indexer 2 Program to the amplifier.	See Save Control Program to amplifier flash memory (p. 26).
Re-save to disk if the program changed.	 Click the **Save Control Program to disk** tool and use the navigation screen to save a copy of the program.

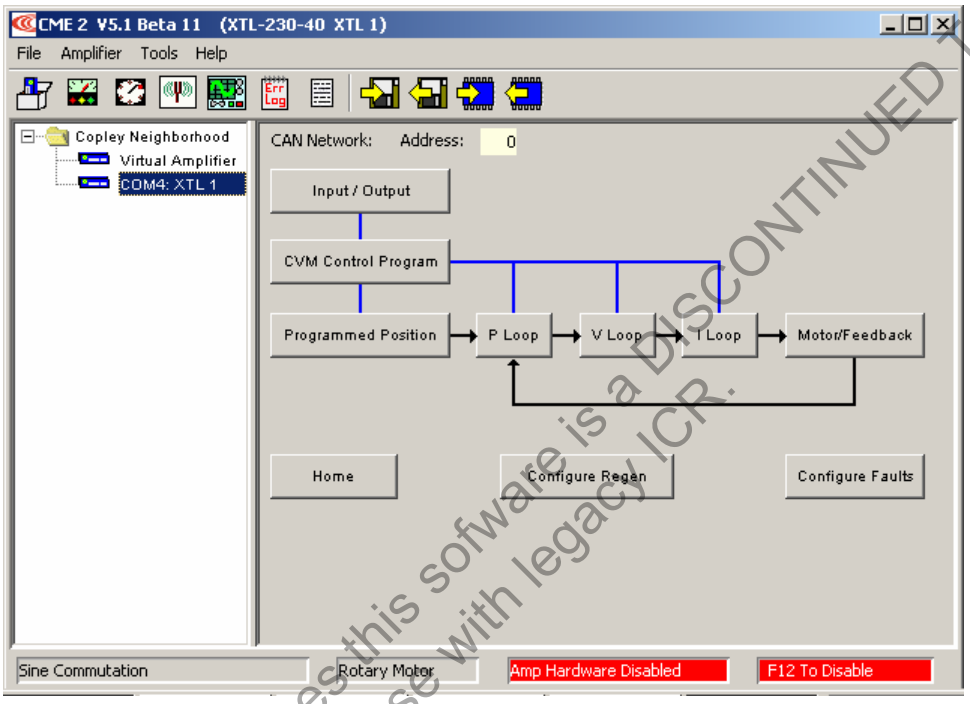
3.2: Basic Amplifier Setup

3.2.1: Typical Basic Amplifier Setup

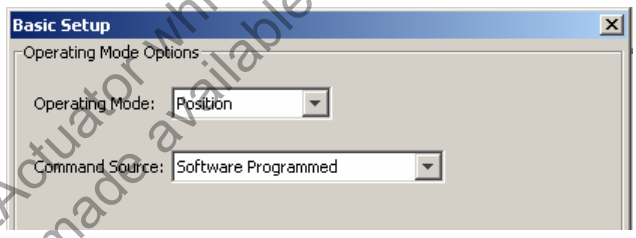
In a typical Indexer 2 Program application, the amplifier is programmed to operate in position mode with a software programmed command source, as described below:

- Access CME 2 by clicking on the desktop icon 

or using the desktop Start menu command **Copley Motion→CME 2→CME 2**. When the *F12 Disables Amplifier* prompt appears, click OK. If necessary, select an amplifier in the *Copley Neighborhood* tree.



- Click **Basic Setup** and then click **Change Settings.** Verify each set of Basic Setup options, using the **Next** button until you reach the *Operating Mode* options. Set *Operating Mode* to **Position** and set *Position Loop Input to* **Software Programmed**.

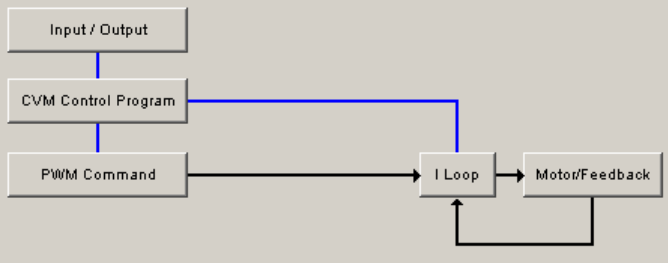


- Click **Next** until the final *Basic Setup* screen (with the **Finish** button) opens.
- Click **Finish**.
- Verify that the amplifier has been configured, tuned and tested. See the *CME 2 User Guide*.

3.2.2: Using Indexer 2 with Other Modes and Command Sources

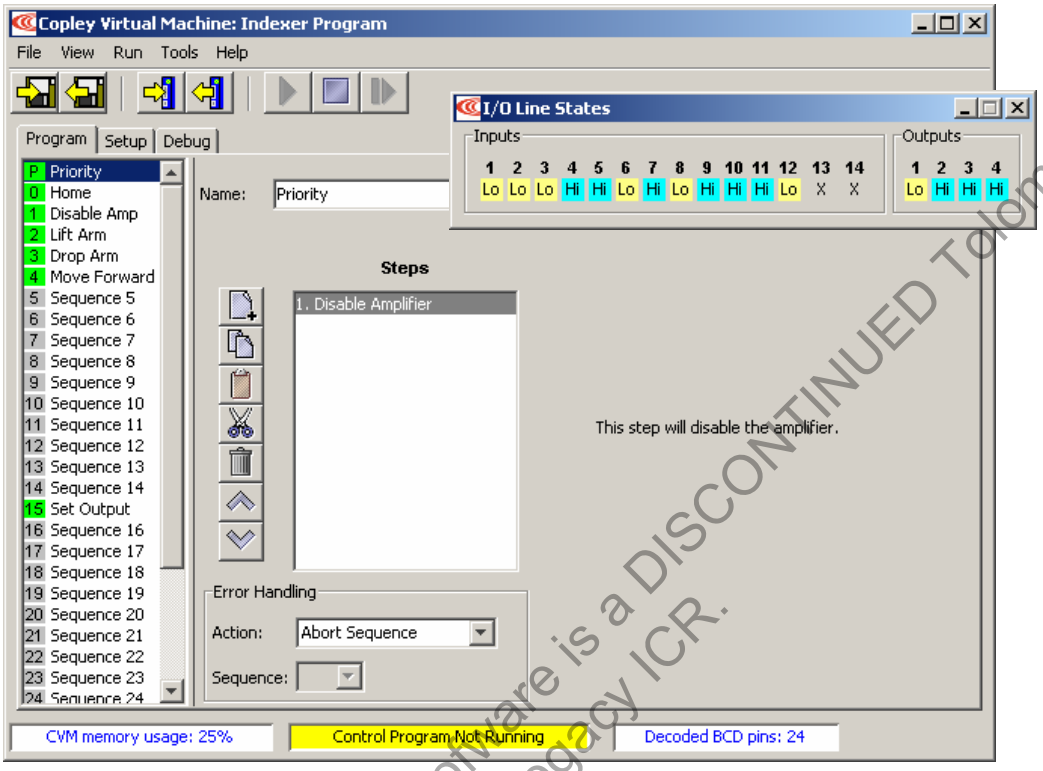
Indexer 2 Program setup controls are available in all operating mode/command source configurations. Likewise, Indexer 2 Program sequences can be launched in any operating mode/command source configuration.

For instance, the amplifier can be configured to run in current mode using a PWM command source. In this case, the Indexer 2 Program controls are accessible through the **CVM Control Program** button on the CME 2 *Main* screen as shown below.



3.3: Accessing the Indexer 2 Program

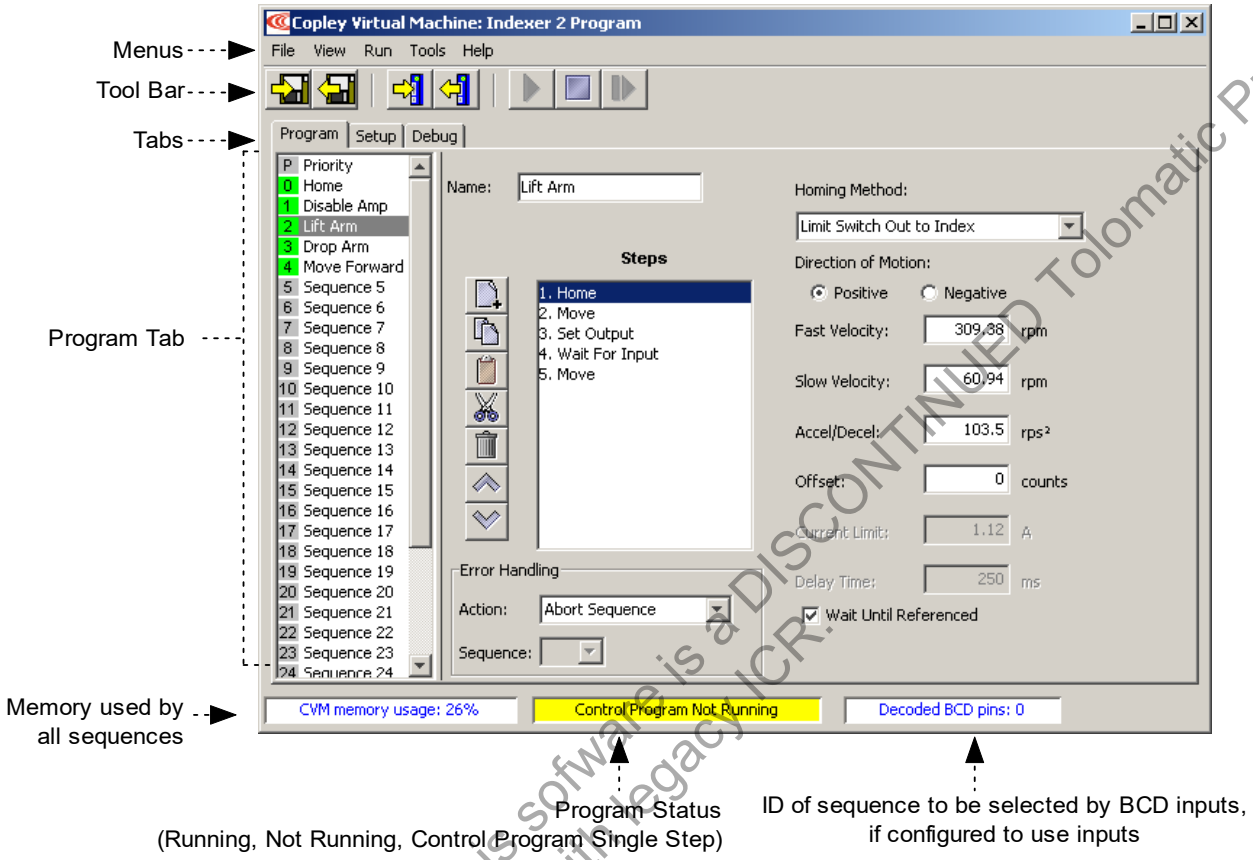
- On the CME 2 *Main* screen, click **CVM Control Program** to open the *Copley Virtual Machine: Indexer 2 Program* and *I/O Line States* screens.



The Indexer 2 Program in the amplifier's flash memory will be displayed. The numbers of sequences which have been programmed with steps will be displayed with a green background, as some of the sequences shown above. The *I/O Line States* screen shows real-time status of the amplifier's digital inputs and outputs.

3.4: Indexer 2 Program Screen Overview

Major elements of the *Indexer 2 Program* screen are highlighted below.



The *Program* tab is used to create and modify sequences.
See Creating and Modifying Sequences (p. 31)
and Adding Steps (Functions) to a Sequence (p. 33).

The *Setup* tab is used to set up sequence selection, Priority sequence, and Go command.
See Setting Up Sequence Selection and Go Command (p. 27).

The *Debug* tab is used to run the Indexer 2 Program in single-step/debug mode.
See Using Single-Step/Debug Mode (p. 34).

3.5: Menus

3.5.1: File Menu

The **File** menu provides options for saving and opening files and deleting the Indexer 2 Program from the PC's memory and the amplifier's flash memory.

Menu Selection	Description
Save Control Program	Saves the current CVM program in the PC's memory to a disk file with a *.ccp* filename extension.
Open Control Program	Opens a saved CVM program. The program contained in this file will replace the current program in the PC's memory.*
New Control Program	Starts a new CVM control program. Note that using this command deletes the current CVM program from the PC's memory.*
Delete Control Program in Amplifier Flash	Removes the CVM control program that is currently stored in the amplifier's flash memory.
*NOTE: When a new CVM control program is started or an existing one is opened from a file (.ccp), Enable Control Program on Startup is automatically selected.	

3.5.2: View Menu

The **View** menu provides options for viewing register values and CVM control program code.

Menu Selection	Description
Assembly Code	Displays the assembly language code of the CVM program in the PC's memory.
CVM Directory	Opens the *CVM Directory* screen, which displays all files stored in the CVM flash memory.
Register Values	Displays the current values of the Indexer 2 Program's 32 registers (R0-R31).
Bean Versions	Displays the ID number and version of all the functions installed in the CME 2 CVM library (typically C:\Program Files\Copley Motion\CME 2\CvmLib).

3.5.3: Run Menu

The **Run** menu provides options for starting and stopping the Indexer 2 Program.

Menu Selection	Description
Run Control Program	Starts Indexer 2 Program execution. CAUTION: Depending on setup configuration and input line state, motion could start immediately. NOTE: The **Run Control Program** menu selection is disabled if the Indexer 2 Program has been changed but not saved to amplifier flash memory.
Stop Control Program	Stops Indexer 2 Program execution. CAUTION: Any programmed moves in progress will continue until finished.
Enable Control Program on Startup*	Configures the Indexer 2 Program to auto start when the amplifier is powered up or reset. This choice is the default setting.*
Disable Control Program on Startup*	Disables auto start of the Indexer 2 Program.*
*NOTE: When a new CVM control program is started or an existing one is opened from a file (.ccp), Enable Control Program on Startup is automatically selected.	

3.5.4: Tools Menu

The **Tools** menu provides access to tools.

Menu Selection	Description
Clear All CVM/Cam Flash	Deletes all CVM control programs, Cam tables and gain scheduling tables from the amplifier CVM flash memory. Available only when Indexer 2 Program is not running. CAUTION: This command will delete all data stored in the amplifier CVM flash memory. It should only be used when it is believed that this memory has been corrupted.

3.6: Toolbar Functions

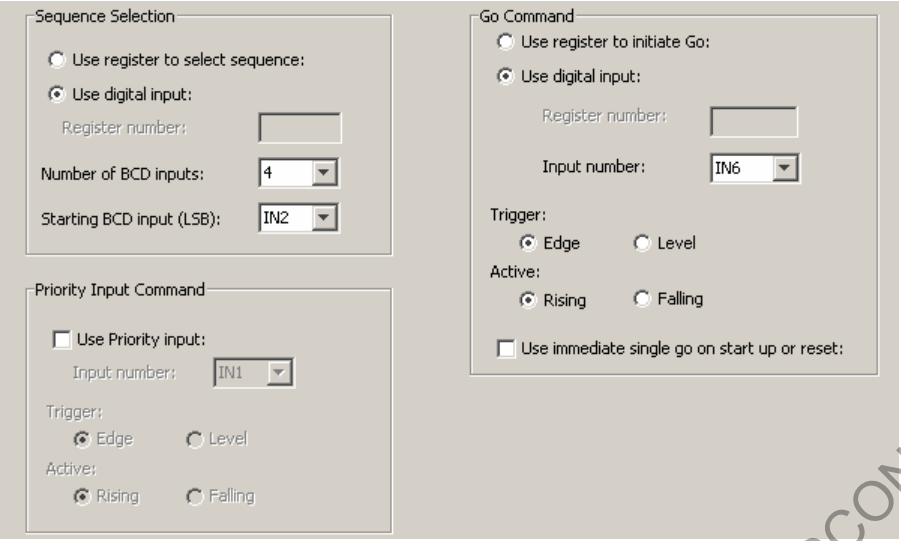
Near the top of the *Indexer 2 Program* screen is a set of tool buttons:

Click on the appropriate tool button to perform the functions described below:

Icon	Tool	Description
	Save Control Program to disk	Saves the current CVM program in PC memory to a disk file with a *.ccp* filename extension.
	Restore Control Program from disk	Opens a saved CVM program. The program contained in this file will replace the current program in the PC memory.*
	Save Control Program to amplifier flash memory	Saves the current CVM program in PC memory to amplifier flash memory. This will overwrite any program currently in the amplifier
	Restore Control Program from amplifier flash memory	Replaces the current CVM program in PC memory with program stored in amplifier flash memory.
	Run Control Program	Starts program execution. CAUTION: Depending on setup configuration and input line state, motion could start immediately. NOTE: The **Run Control Program** menu selection is disabled if the program has been changed but not saved to amplifier flash memory.
	Stop Control Program	Stops program execution. CAUTION: Any programmed moves in progress will continue until finished.
	Single-Step Control Program	Starts the CVM program in single-step mode and opens the *Debug* tab for debugging purposes. See Using Single-Step/Debug Mode (p. 34).
*NOTE: When a control program is restored from disk, Enable Control Program on Startup is automatically selected.		

3.7: Setting Up Sequence Selection and Go Command

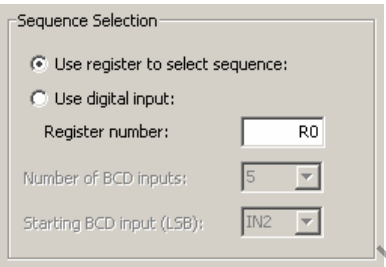
Click *Setup* to access Sequence Selection, Go Command, and Priority Input Command controls.



3.7.1: Sequence Selection

Use Register to Select Sequence

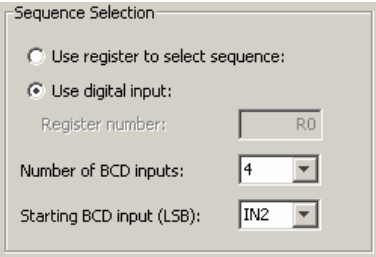
To configure register selection, click **Use register to select sequence** and then enter the **Register number** (range R0-R31) of the register which will contain the sequence number.



For information on writing the sequence number, see Use a Register to Select a Sequence (p. 15).

Use Digital Input to Select Sequence

To configure Indexer 2 Program to use digital inputs for sequence selection, click **Use digital input** in the Sequence Selection box:



Select appropriate values for the following fields.

Field	Description
Number of BCD inputs	Selects the number of inputs to be used for selecting sequences. The number of inputs chosen determines the number of sequences that can be selected using the inputs. For instance, 1 input can be used to select sequence 0 or sequence 1. Five inputs can be used to select any of the 32 sequences. See Use Digital Inputs to Select a Sequence (p. 15).
Starting BCD input (LSB)	Selects which input will be used for the least significant bit of the sequence address. The remaining address lines will be on consecutively higher inputs.

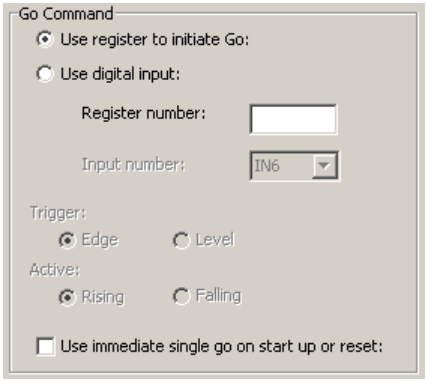
NOTE: The BCD inputs are only read when the Go command is received. The inputs can then be used for other purposes until the next Go command is expected. Exercise caution to avoid conflicting uses of an input.

3.7.2: Go Command

Choose one of the Go commands described below. Clicking **Use immediate single Go on startup or reset** triggers a Go command on start up or reset or when the Run button is pushed. After that, the register or digital input Go commands function as programmed.

Use Register to Initiate Go

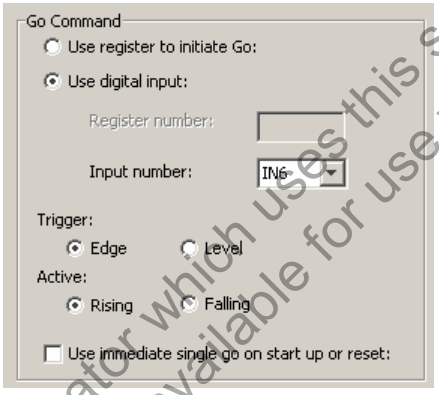
- Click **Use register to initiate go.**
- Enter the **Register number** (range R0-R31) of the Go command register. (If Use register to select sequence was also checked, the sequence selection register number will be changed to match Go command register number.)



For information on setting the Go bit (Bit 15), see Use a Register to Initiate Go (p. 16).

Use Digital Input to Initiate Go

- Click **Use digital input**.

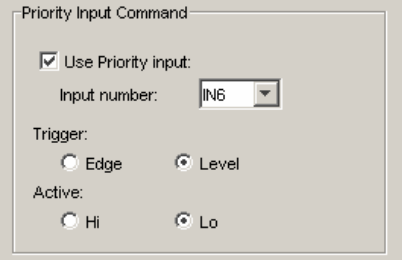


- Select appropriate values for the following fields.

Input number	Selects which input will be used to execute the sequence.
Trigger	• Edge: Input must transition for the sequence to start. • Level: Input must be at the correct level for the sequence to start. Note that a level-triggered Go command will cause the sequence to repeat until the level becomes inactive.
Active	With Edge Trigger: • Rising: Sequence will start on a low to high transition of the Go input. • Falling: Sequence will start on a high to low transition of the Go input. With Level Trigger: • Hi: Sequence will start when the Go input is high. • Lo: Sequence will start when the Go input is low.

3.7.3: Priority Input Command

- Click **Use Priority input**.

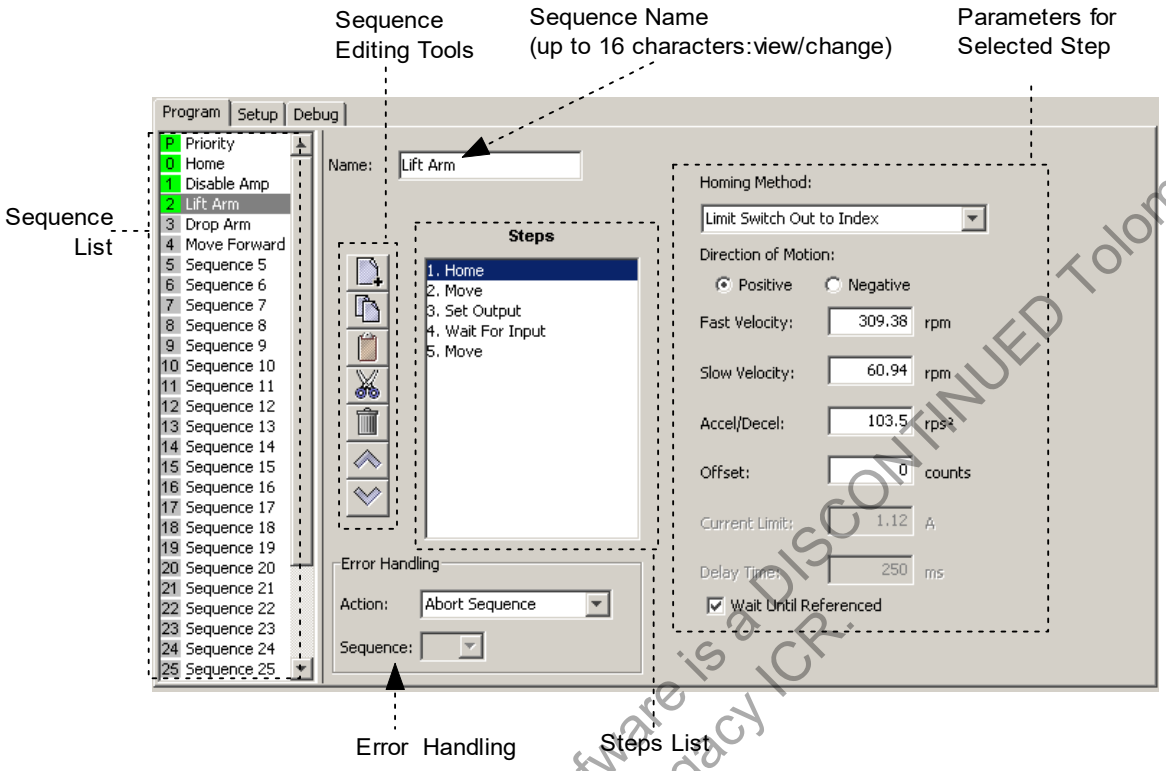


- Select appropriate values for the following fields.

Input number	Selects which input will be used to execute the Priority sequence.
Trigger	• Edge: Input must transition for the Priority sequence to start. • Level: Input must be at the correct level for the Priority sequence to start. Note that a level-triggered Priority Input command will cause the Priority sequence to repeat until the level becomes inactive.
Active	With Edge Trigger: • Rising: Sequence will start on a low to high transition of the Priority input. • Falling: Sequence will start on a high to low transition of the Priority input. With Level Trigger: • Hi: Sequence will start when the Priority input is high. • Lo: Sequence will start when the Priority input is low.

3.8: Creating and Modifying Sequences

The *Program* tab contains tools for creating sequences.



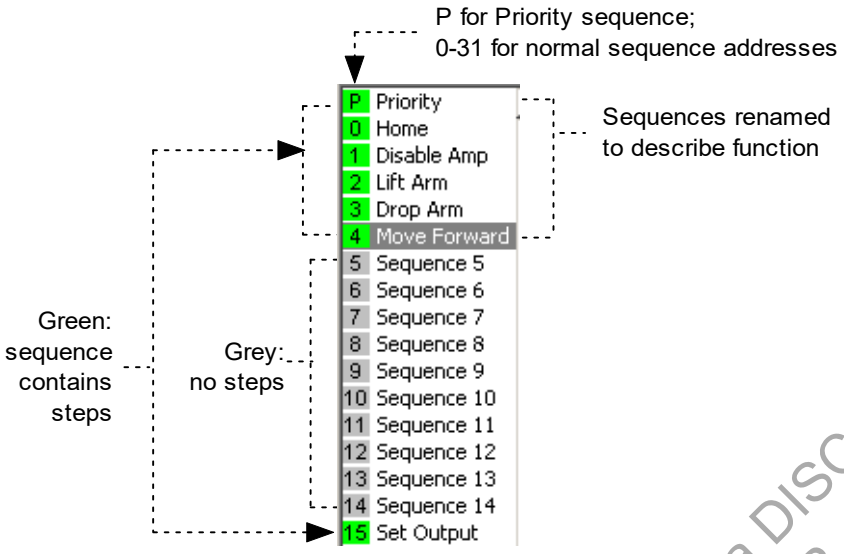
3.8.1: Tools and Controls

Icon	Tool/Field	Description
	Name	Displays the name of the selected sequence. Name can contain up to 16 characters. Enter name changes directly in this field.
	Add Step	Opens the *Indexer Functions* screen. When a Function is selected from this screen and added, the function appears as a step in the sequence.
	Copy Step	Place a copy of the selected step on the Indexer 2 Program's clipboard.
	Paste Step	Paste the contents of the Indexer 2 Program's clipboard in the selected sequence. The step is added after the selected step.
	Cut Step	Remove the selected step from the sequence and place it on the Indexer 2 Program's clipboard.
	Delete Step	Remove the selected step from the sequence. After the step is deleted it cannot be retrieved.
	Step Up	Moves the selected step to the previous position in the sequence.
	Step Down	Moves the selected step to the next position in the sequence.
	Parameters	The right side of the *Program* tab displays the function parameters associated with the selected step, as described in Functions (p. 37).
	Error Handling: Action	Abort Sequence: When a sequence error occurs, the Indexer 2 Program will immediately abort the sequence and wait for a Go command. If a Go command is received, the program will run the currently selected sequence.
	Error Handling: Sequence	Jump to Sequence: On error, the Indexer 2 Program will immediately execute the sequence selected in the Sequence field (no Go command required). NOTE: Sequence errors do not necessarily result in amplifier faults and amplifier faults do not necessarily result in sequence errors.

3.8.2: Sequence List

The sequence list has entries for the priority sequence and 32 sequences.

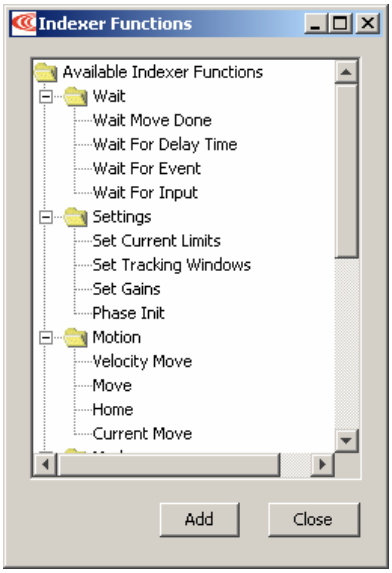
A green sequence number indicates that steps have been added to the sequence.
A grey sequence number indicates that no steps have been added to the sequence.



- To change a sequence name, select the sequence and then enter the new name in the *Name* field on the *Program* tab.
- To program a sequence, select the sequence and proceed to Adding Steps (Functions) to a Sequence (p. 33).

3.9: Adding Steps (Functions) to a Sequence

- Select the sequence.
- Click **Add New Step**  on the *Program* tab to open the *Indexer Functions* screen.



For function descriptions see Functions (p. 37).

- To add a function as a step at the end of the sequence, click the function name and then click **Add**.
- To add a multiple functions, hold the Control key while clicking on the functions you wish to add, in the order in which you wish them to appear. Then click **Add** to add the steps to the sequence.
- Program the functions as instructed in the function descriptions.
- Click **Close** to close the screen.

See Overview of Procedures (p. 20) for a list of steps to perform to run the Indexer 2 Program.

3.10: Using Single-Step/Debug Mode

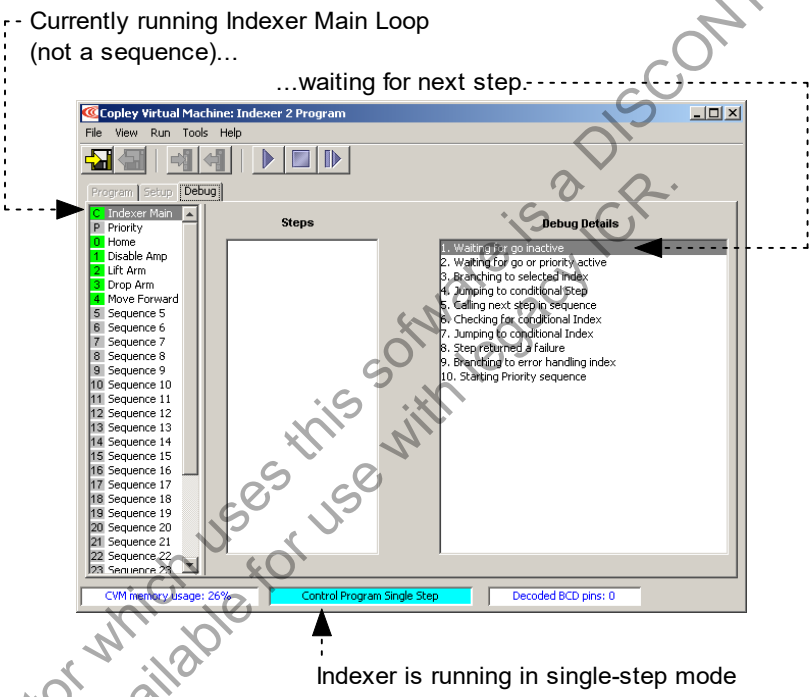
3.10.1: Overview

The *Debug* tab is used to display status information while the Indexer 2 Program is running in single-step/debug mode.

3.10.2: Entering Single-Step/Debug Mode

- To single-step a sequence from the beginning:

1. Click **Run Program**  , then **Stop Program**  and then **Single Step Control Program**  .
   OR
   Click **Save to Flash**  and then Single Step Control Program  .
   The CVM program begins executing at the Indexer 2 Program Main loop level. The *Debug Details* frame contains the list of instructions associated with the current sequence step or Indexer 2 Main program state.



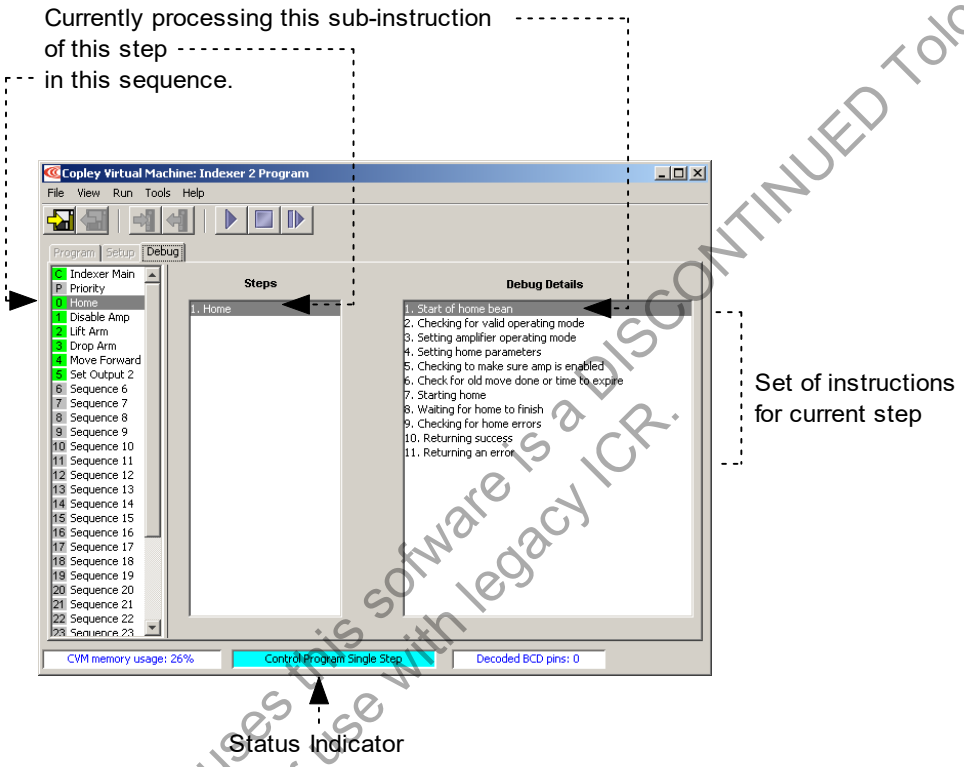
2. Click **Single Step Control Program**  as needed to proceed to the next step.
   If the Indexer 2 Program is at a Wait step, the condition for the wait must be satisfied before the program will proceed.

- To start single-stepping while an Indexer 2 Program is running:

  1 Click **Stop Program** and then **Single Step Control Program** .

  2 Click **Single Step Control Program** as needed to proceed to the next step. If the program is at a Wait step, the condition for the wait must be satisfied before the program will proceed.

  In the example below, a Home step in Sequence 0 (named home) is ready to begin with the next click of the **Single Step Control Program** button ( ).



The set of instructions shown in *Debug Details* is an ordered list with conditional branches. For instance, if instruction 2, *Checking for valid operating mode*, determines that the current operating mode is valid for the move to be executed, then the program skips instruction 3, *Setting amplifier operating mode* and continues with instruction 4. Likewise, if step 8 finds move errors, then the program processes instruction 10, *Returning an error*, instead of 9, *Returning success*.

3.10.3: Exiting Single-Step/Debug Mode

Click **Stop Control Program** to exit single-step mode and stop the Indexer 2 Program.

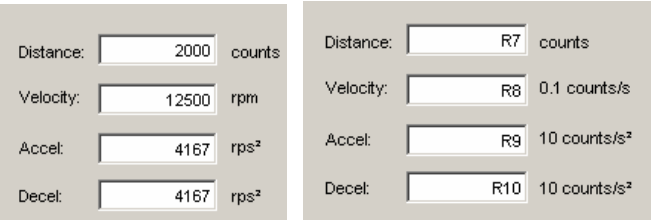
Click **Run Control Program** to exit single-step mode and re-start the Indexer 2 Program.

CHAPTER 4: FUNCTIONS

This chapter describes the functions that can be used in an Indexer 2 Program sequence. Contents include:

4.1: Using Registers to Pass Values to Functions

In many Indexer 2 Program function parameter fields, references to the program's 32 registers can be used in place of hard-coded numbers. For instance, move parameters can be entered as numbers (at left, below) or as register references (at right, below).



Note that in some cases, the units required for entry via the *Indexer 2 Program* screen are different from the units required by a register. Be sure to express all values in the units called for in the field label.

When the move is processed, the Indexer 2 Program performs the register value substitutions. Invalid data in a register will result in a sequence error.

To see the current register contents, use the **Tools→View Register Values** command.

4.2: Wait Move Done

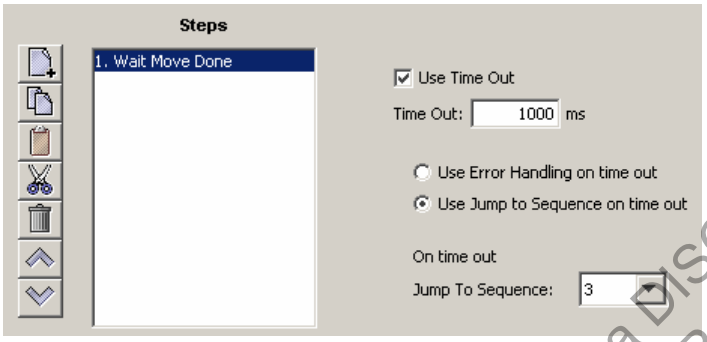
4.2.1: Wait Move Done Function Overview

Wait Move Done pauses execution of the sequence until the move in progress is completed.

4.2.2: Adding a Wait Move Done Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Wait: Wait Move Done** to select the function. Click **Add** to add it to the sequence.

  The *Program tab* displays the Wait Move Done parameters.



- Choose appropriate values for the following parameters. Enter the *Time out* value directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Use time out	If checked and the move is not completed within the *Time out* period, the Indexer 2 Program performs the action selected from the choices below (Use Error Handling or Use Jump to Sequence).
Time out	Time period used by the time out option. Register Units: ms.
Use Error Handling on time out	If Time Out occurs, generate a sequence error.
Use Jump to Sequence on time out	If Time Out occurs, jump to the specified sequence.

4.2.3: Wait Move Done Notes

Errors

A sequence error will occur if during the wait time:

- The amplifier becomes hardware disabled.
- The amplifier faults.
- A software travel limit is reached or a hardware limit switch is activated.
- A register used for a function parameter contains a value that is not valid for the parameter.

4.3: Wait for Delay Time

4.3.1: Wait for Delay Time Overview

Wait for Delay Time pauses execution of the sequence for the specified amount of time.

4.3.2: Adding a Wait for Delay Time Step to a Sequence

- Click **Add New Step**  on the *Program* tab to open *Indexer Functions*.
- Click **Wait for Delay Time** to select the function. Click **Add** to add it to the sequence.
  The *Program tab* displays the Wait for Delay Time parameters.

- Enter the value directly or enter the number (R0-R31) of an Indexer 2 Program register.
  See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Delay Time	The amount of time that the sequence will pause. Register units: ms.

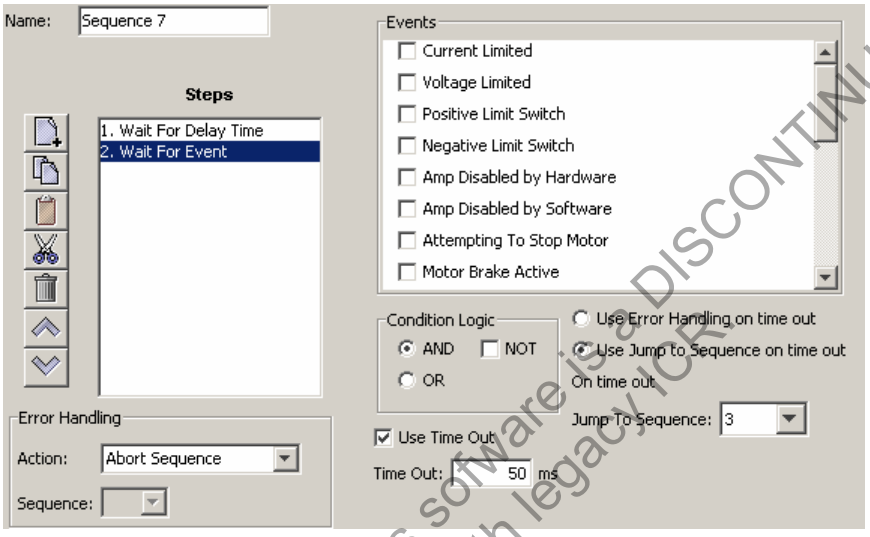
4.4: Wait for Event

4.4.1: Wait for Event Function Overview

Wait for Event pauses execution of the sequence until the specified event conditions are met.

4.4.2: Adding a Wait for Event Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Wait: Wait for Event** to select the function. Click **Add** to add it to the sequence.
  The *Program tab* displays the Wait for Event parameters.



- Select from the Events described on the next page. Set the condition logic and timeout controls described below as needed. Enter *Time out* time directly or enter the number (R0-R31) of an Indexer 2 Program register.
  See Using Registers to Pass Values to Functions (p. 38).

Control	Description
AND	Wait until all selected events are true.
OR	Wait until any of the selected events happen.
NOT	Wait until the selected event conditions are not present.
Use time out	If checked and the wait condition is not met within the *Time out* period, the Indexer 2 Program performs the action selected from the choices below (Use Error Handling or Use Jump to Sequence).
Time out	Time period used by the time out option. Register Units: ms.
Use Error Handling on time out	If Time Out occurs, generate a sequence error.
Use Jump to Sequence on time out	If Time Out occurs, jump to the specified sequence.

Wait for Event: Events	
Event	**Description**
Current Limited	The current output is being limited by the I^2T algorithm or a latched current fault has occurred.
Voltage Limited	Current loop is commanding the full bus voltage in an attempt to control current. Often occurs when motor is running as fast as the available bus voltage allows.
Positive Limit Switch	Axis has contacted positive limit switch.
Negative Limit Switch	Axis has contacted negative limit switch.
Positive Software Limit	Actual position has exceeded the positive software limit setting.
Negative Software Limit	Actual position has exceeded the negative software limit setting.
Following Error	Following error has occurred.
Following Warning	Following error has reached the programmed warning limit.
Velocity Limited	The velocity command (from analog input, PWM input, or position loop) has exceeded the programmed velocity limit.
Acceleration Limited	In velocity mode, motor has reached a programmed acceleration or deceleration limit.
Velocity Outside of Tracking Window	Difference between target and actual velocity has exceeded the window.
Position Outside of Tracking Window	The following error has exceeded the programmed value.
Amp Disabled by Hardware	Amplifier enable input(s) is not active.
Amp Disabled by Software	Amplifier is disabled by a software command.
Attempting to Stop Motor	The amplifier, while in velocity or position mode, has been disabled. In velocity mode, amplifier is using the programmed Fast Stop Ramp. In position mode, the amplifier is using the programmed Abort Deceleration. The output remains active until the amplifier is re-enabled.
Motor Brake Active	Motor brake activated.
PWM Outputs Disabled	The amplifier's PWM outputs are disabled.
Home Switch Active	Axis has contacted the home limit switch.
In Motion	Motor is moving, or has not settled after a move. Settled when it comes within the position tracking window and stays there for the tracking time at the end of a move. Once settled, it remains settled until a new move is started.
Phase not Initialized	Set when the amplifier has not yet initialized its phase when using phase initialization (wake and wiggle).

4.4.3: Wait for Event Function Notes

Errors

A sequence error will occur if this function is executed when a register used for a function parameter contains a value that is not valid for the parameter.

4.5: Wait for Input Mask

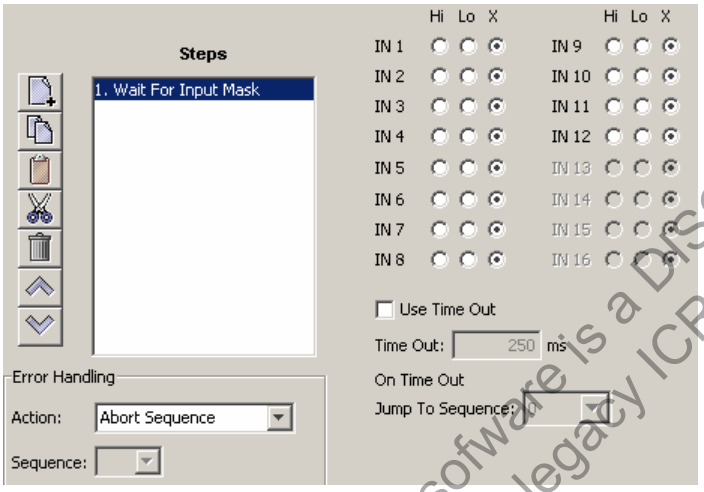
4.5.1: Wait for Input Mask Function Overview

This function configures an Indexer 2 Program step to wait for the states of the amplifier's digital inputs to match the specified mask before continuing to the next step.

4.5.2: Adding a Wait for Input Mask Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Wait: Wait for Input Mask** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Wait for Input Mask parameters.



- Choose appropriate values for the following parameters. Enter the *Time out* value directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Hi Lo X (IN 1 – IN 16)	Defines the mask: Hi: Mask condition is met if the input is high. Lo: Mask condition is met if the input is low. X: Don't care.
Use Time Out	If checked and the mask condition is not met within the *Time out* period, the Indexer 2 Program will jump to the specified sequence.
Time out	Time period used by the time out option. Register Units: ms
On Time Out Jump to Sequence	If Time Out occurs, jump to the specified sequence.

4.5.3: Wait for Input Mask Notes

Errors

A sequence error will occur if a register used for a function parameter contains a value that is not valid for the parameter.

4.6: Wait for Position

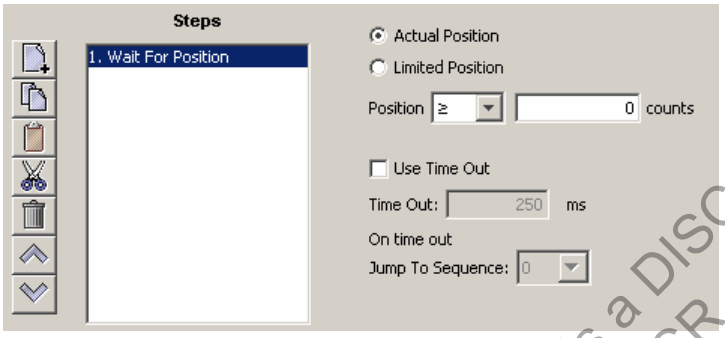
4.6.1: Wait for Position Overview

Wait for Position pauses execution of the sequence until the axis position meets the programmed criteria.

4.6.2: Adding a Wait for Position Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Wait: Wait for Position** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Wait for Position parameters.



- Choose appropriate values for the following parameters.

Parameter	Description
Position	Wait until the axis position is, as specified: • greater than or equal to the specified position or • less than or equal to the specified position. Register Units: counts.
Actual Position	Use the actual position for the comparison position.
Limited Position	Use the limited position for the comparison position. Typically used for stepper motors operated in open-loop stepper mode.
Use Time Out	If checked and the position condition is not met within the *Time out* period, the Indexer 2 Program will jump to the specified sequence.
Time Out	Time period used by the time out option. Register Units: ms
On Time Out Jump to Sequence	If Time Out occurs, jump to the specified sequence.

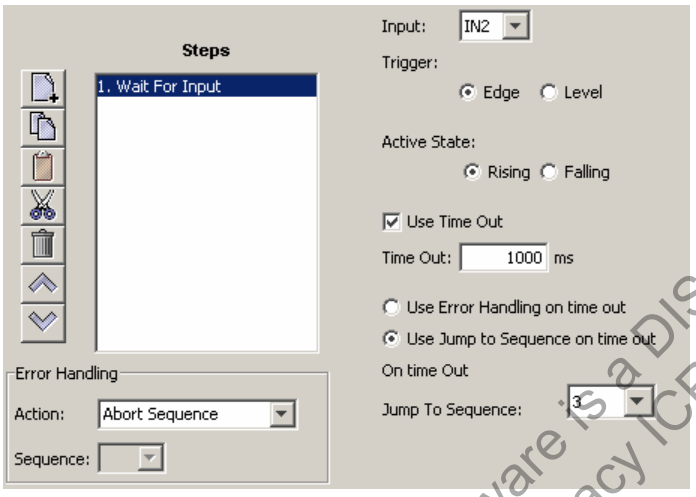
4.7: Wait for Input

4.7.1: Wait for Input Overview

Wait for Input pauses execution of the sequence until the specified input condition is met.

4.7.2: Adding a Wait for Input Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Wait for Input** to select the function. Click **Add** to add it to the sequence.
  The *Program* tab displays the Wait for Input parameters.



- Choose appropriate values for the following parameters.

Parameter	Description
Input	Selects which input will be monitored.
Trigger	• Edge: Will wait for an edge on the selected input. • Level: Will wait for a level condition on the selected input. If the selected level is present on the input at the start of the wait, the sequence will continue to the next step.
Active State	With Edge Trigger: • Rising: Sequence will continue on a low to high transition of the selected input. • Falling: Sequence will continue on a high to low transition of the selected input. With Level Trigger: • Hi: Sequence will continue when the selected input is high. • Lo: Sequence will continue when the selected input is low.
Use time out	If checked and the input condition is not met within the *Time out* period, the Indexer 2 Program performs the action selected from the choices below (Use Error Handling or Use Jump to Sequence).
Time out	Time period used by the time out option. Register Units: ms.
Use Error Handling on time out	If Time Out occurs, generate a sequence error.
Use Jump to Sequence on time out	If Time Out occurs, jump to the specified sequence.

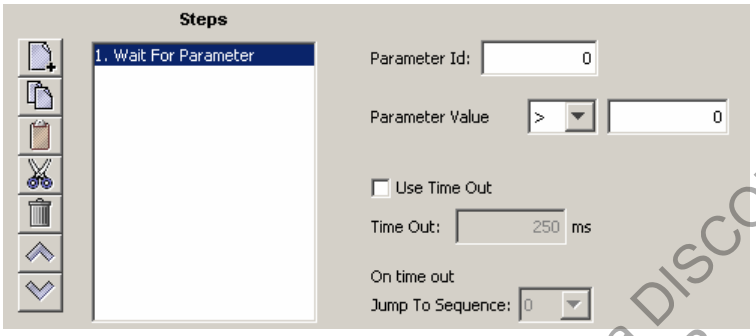
4.8: Wait for Parameter

4.8.1: Wait for Parameter Overview

Wait for Parameter pauses execution of the sequence until the specified parameter condition is met.

4.8.2: Adding a Wait for Parameter Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Wait for Parameter** to select the function. Click **Add** to add it to the sequence.
  The *Program* tab displays the Wait for Parameter parameters.



- Choose appropriate values for the following parameters.

Parameter	Description
Parameter ID	The ID of the amplifier parameter which will be monitored.
Parameter Value	The value of the monitored parameter is compared to the value in the field, using the chosen comparison operator.
Use time out	If checked and the parameter condition is not met within the *Time out* period, the Indexer 2 Program jumps to the specified sequence.
Time out	Time period used by the time out option. Register Units: ms.
On time out Jump to Sequence	If Time Out occurs, jump to the specified sequence.

4.8.3: Wait for Parameter Function Notes

For a list of amplifier parameter IDs, see the *Copley Parameter Dictionary.* Use the ASCII ID value.

Parameter ID's can be entered as decimal or hexadecimal numbers, using the 0x notation (for instance, 0x00c1). When the Indexer 2 Program is reloaded, all values will be displayed in decimal form.

Due to the finite scan time of this function, the "equal to operator" should not be used with rapidly changing parameters such as actual velocity or actual current. Instead, chose "equal to or greater than" or "equal to or less than".

Errors

A sequence error will occur if:

- The specified parameter ID does not exist.
- The specified parameter ID is for a flash only parameter.

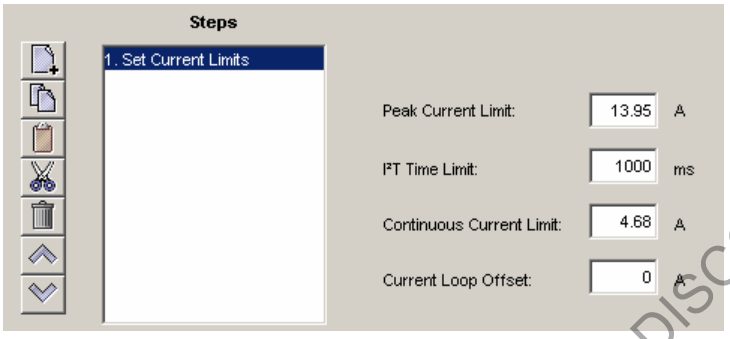
4.9: Set Current Limits

4.9.1: Set Current Limits Function Overview

The Set Current Limits function changes the current limits.

4.9.2: Adding an Set Current Limits Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Settings:Set Current Limits** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Set Current Limits parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Peak Current Limit	Maximum current that can be generated by the amplifier for a short duration of time. This value cannot exceed the peak current rating of the amplifier. Register Units: 0.01 Amps
I2T Time Limit	Maximum amount of time that the peak current can be applied to the motor before it must be reduced to the continuous limit or generate a fault. Register Units: ms
Continuous Current Limit	Maximum current that can be constantly generated by the amplifier. Register Units: 0.01 Amps
Current Loop Offset	Sets a current offset. Register Units: 0.01 Amps

4.9.3: Set Current Limits Function Notes

Errors

A sequence error will occur if this function is executed when a register used for a function parameter contains a value that is not valid for the parameter.

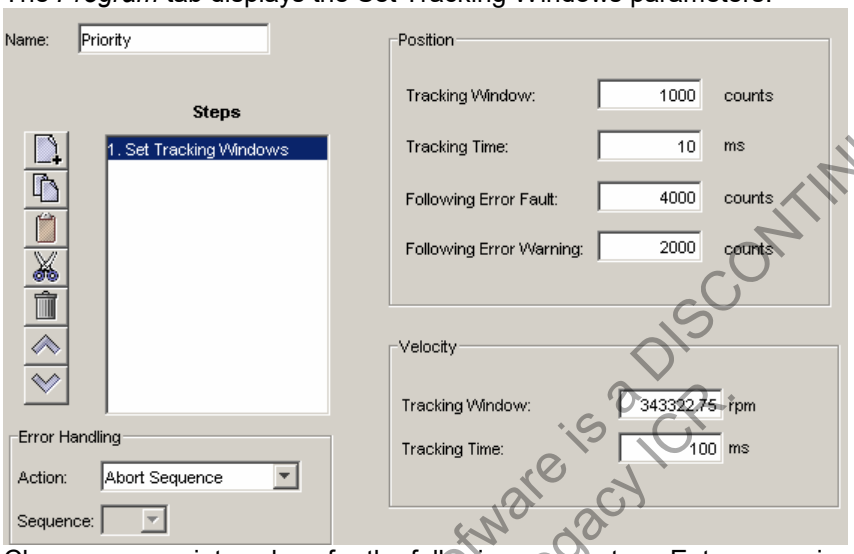
4.10: Set Tracking Windows

4.10.1: Set Tracking Windows Function Overview

This function modifies the velocity and position tracking windows.

4.10.2: Adding a Set Tracking Windows Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Settings:Set Tracking Windows** to select the function. Click **Add** to add it to the sequence.
  The *Program* tab displays the Set Tracking Windows parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Position Tracking	
Parameter	**Description**
Tracking Window	Width of the tracking window. Register units: counts.
Tracking Time	Position must remain in the tracking window for this amount of time to be considered tracking. Register Units: ms
Following Error Fault	The level at which the following error produces a fault. Register units: counts.
Following Error Warning	The level at which the following error produces a warning. Register units: counts.
Velocity Tracking	
Parameter	**Description**
Tracking Window	Width of the tracking window. Register units: 0.1 counts/s.
Tracking Time	Velocity must remain in the tracking window for this amount of time to be considered tracking. Register Units: ms

4.10.3: Set Tracking Windows Function Notes

Errors

A sequence error will occur if this function is executed when a register used for a function parameter contains a value that is not valid for the parameter.

4.11: Set Gains

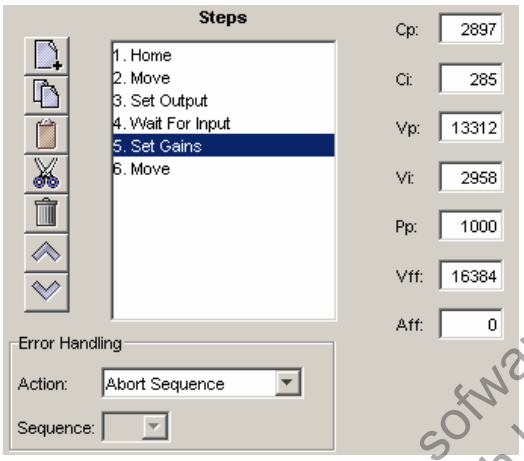
4.11.1: Set Gains Function Overview

This function sets new gain values for the current, velocity and position loops during a sequence. This can be used, for instance, when the load on the axis has been changed.

The new values will stay in effect for all sequences until changed again by another Set Gains function or by an amplifier reset.

4.11.2: Adding a Set Gains Step to a Sequence

- Click Add New Step on the *Program* tab to open the *Indexer Functions* screen.
- Click **Set Gains** to select the function. Click **Add** to add it to the sequence.
  The *Program* tab displays the Set Gains parameters.



- Choose appropriate values for the following parameters. Enter number values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Cp	Current loop proportional gain.
Ci	Current loop integral gain.
Vp	Velocity loop proportional gain.
Vi	Velocity loop integral gain.
Pp	Position loop proportional gain.
Vff	Velocity feed forward gain.
Aff	Acceleration feed forward gain.

4.12: Velocity Move Position Mode

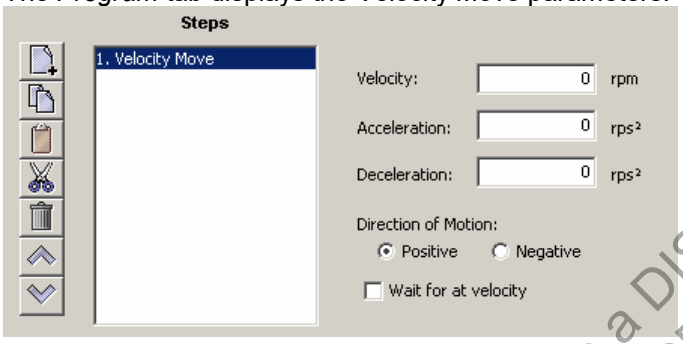
4.12.1: Velocity Move Position Mode Function Overview

This function is used to change the operating mode of the amplifier to programmed position mode and configures a constant velocity trajectory.

4.12.2: Adding a Velocity Move Position Mode Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Motion:Velocity Move Position Mode** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Velocity Move parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Velocity	Commanded velocity. Register units: 0.1 counts/s. (Positive values only)
Acceleration	Acceleration rate. Register units: 10 counts/s^2.
Deceleration	Deceleration rate. Register units: 10 counts/s^2.
Direction of Motion	Positive or negative.
Wait for at velocity	If checked, sequence execution will wait at this step until the commanded velocity has reached the new value.

4.12.3: Velocity Move Position Mode Notes

Errors

A sequence error will occur if this function is executed when:

- The amplifier is hardware disabled.
- The amplifier is faulted.
- Motor phasing is not initialized.
- A register used for a function parameter contains a value that is not valid for the parameter.

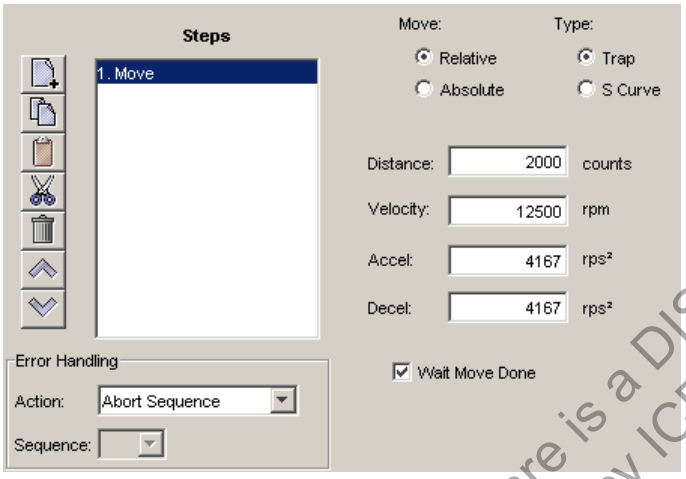
4.13: Move

4.13.1: Move Function Overview

The Move function executes a profile move using the specified parameters.

4.13.2: Adding a Move Step to a Sequence

- Click **Add New Step** to open the *Indexer Functions* screen.
- Click **Move** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Move parameters.



- Choose appropriate values for the following parameters. Enter number values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Move	Type of move: • Relative: The axis will move the specified distance from the current position. • Absolute: The axis will move to the specified absolute position.
Type	Type of profile: • Trap: The move will use a trapezoidal profile. • S Curve: The move will use an S curve profile.
Distance	Distance of a relative move. Register units: counts.
Position	Destination position of an absolute move. Register units: counts.
Velocity	Velocity during the constant velocity portion of the move. Register units: 0.1 counts/s
Accel	Acceleration rate of trapezoidal profile or maximum acceleration/deceleration rate of S curve profile. Register units: 10 counts/s^2
Decel (Trap move only)	Deceleration rate of trapezoidal profile. Register units: 10 counts/s^2
Jerk (S-Curve only)	Rate of change of acceleration and deceleration of the S curve profile. Register units: 100 counts/s^3
Wait Move Done	If checked, sequence execution will wait at this step for the commanded move to finish. If an error occurs during the move, the sequence will exit as programmed.

4.13.3: Move Function Notes

Wait Move Done

If **Wait Move Done** is not checked and a second move is commanded, the second move executes immediately with the following results:

- If the second move is a relative, trapezoidal move, the axis moves the relative distance from where the axis was when the second move was executed.
- If the second move is an absolute, trapezoidal move, the axis moves to the position specified by the second move.
- If an S curve profile move is executed while another move is still in progress, an error occurs.

Errors

A sequence error will occur if:

- The amplifier is hardware disabled when the Move begins, or becomes hardware disabled during the Move.
- The amplifier has a fault when the move begins or a fault occurs during the move.
- A software travel limit is reached or a hardware limit switch is activated during the move.

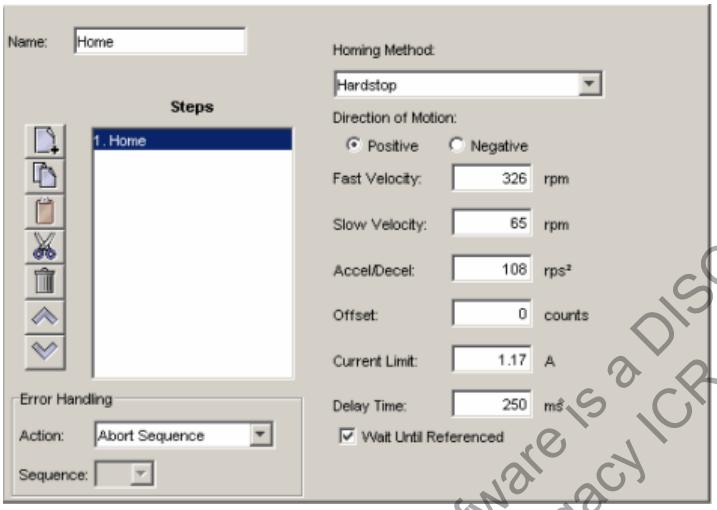
4.14: Home

4.14.1: Home Function Overview

The Home function executes a home command using the method and parameters specified.

4.14.2: Adding a Home Function Step to a Sequence

- Click **Add New Step** to open the *Indexer Functions* screen.
- Click **Home** to select the function. Click **Add** to add it to the sequence.
  The *Program* tab displays the Home function parameters.



- Choose appropriate values for the following parameters. Enter number values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Homing Method	Selects which homing method will be used. See Homing Method Descriptions (p. 79).
Direction of Motion	Sets the initial direction of motion. Positive or Negative.
Fast Velocity	The velocity used to find a limit or home switch. Also used when moving to an offset position, or a resolver or Servo Tube index position. Register units: 0.1 counts/s.
Slow Velocity	The velocity used to find a switch edge, index pulse, or hard stop. Register units: 0.1 counts/s.
Accel/Decel	The acceleration and deceleration rate used during homing. Register units: 10 counts/s^2.
Offset	The axis will execute a move of this distance after the reference is found. The actual position will be reset to 0 at this new position and will now be considered home. Register units: counts.
Current Limit	The current used during the home to hard stop method. Register units: 0.01 A.
Delay Time	The amount time the current limit is applied during the home to hard stop. Register units: ms.
Wait until referenced	If checked, sequence execution will wait at this step for the home move to finish. If a homing error occurs during homing, the Indexer 2 Program will use the Error Handling action selected from the choices below: • Abort Sequence. • Jump to the specified sequence.

4.14.3: Home Function Notes

Do not execute a move function until the homing function is complete.

Errors

A sequence error will occur if:

- The amplifier is, or becomes, hardware disable during the Homing function.
- The amplifier has a fault at the beginning of the Home function or a fault occurs during the function.

Other sequence errors may occur that are specific to the homing method selected.
See Homing Method Descriptions (p. 79).

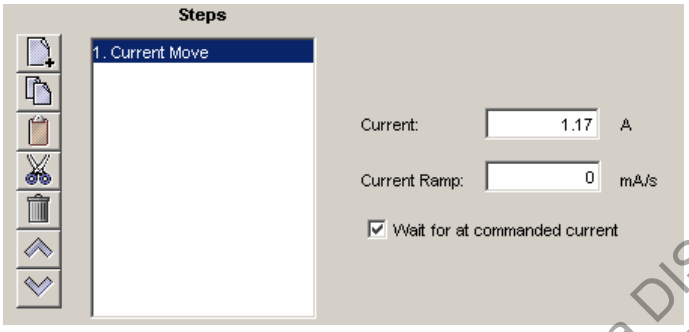
4.15: Current Move

4.15.1: Current Move Function Overview

This function is used to change the operating mode of the amplifier to programmed current mode and configure the mode.

4.15.2: Adding a Current Move Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Motion:Current Move** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Current Move parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Current	Commanded current. Register units: 0.01 A.
Current Ramp	Rate at which the current will change to its commanded value. Register Units: mA/s
Wait for at commanded current	If checked, sequence execution will wait at this step until the commanded current has ramped to the new value.

4.15.3: Current Move Notes

Errors

A sequence error will occur if this function is executed when:

- The amplifier is hardware disabled.
- The amplifier is faulted.
- Motor phasing is not initialized.
- A register used for a function parameter contains a value that is not valid for the parameter.

4.16: Velocity Move Velocity Mode

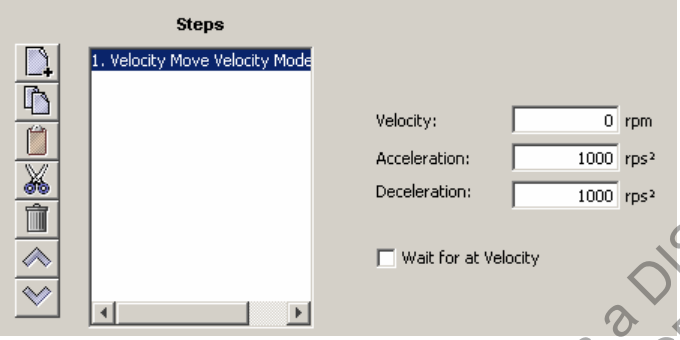
4.16.1: Velocity Move Velocity Mode Function Overview

This function is used to change the operating mode of the amplifier to programmed velocity mode and configures a constant velocity trajectory.

4.16.2: Adding a Velocity Move Velocity Mode Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Motion:Velocity Move Velocity Mode** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Velocity Move parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Velocity	Commanded velocity. Register units: 0.1 counts/s. (Positive values only)
Acceleration	Acceleration rate. Register units: 10 counts/s^2.
Deceleration	Deceleration rate. Register units: 10 counts/s^2.
Direction of Motion	Positive or negative.
Wait for at velocity	If checked, sequence execution will wait at this step until the commanded velocity has reached the new value.

4.16.3: Velocity Move Velocity Mode Notes

Errors

A sequence error will occur if this function is executed when:

- The amplifier is hardware disabled.
- The amplifier is faulted.
- Motor phasing is not initialized.
- A register used for a function parameter contains a value that is not valid for the parameter.

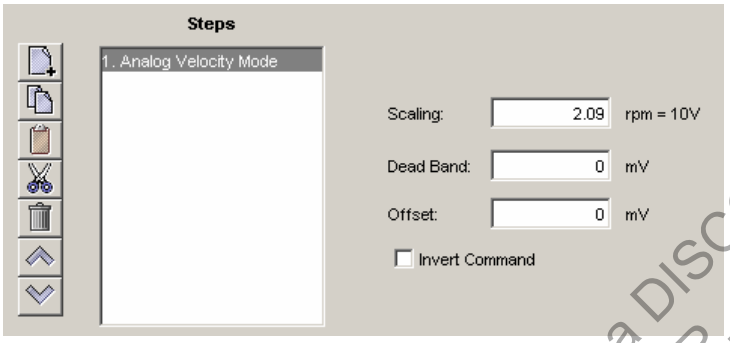
4.17: Analog Velocity Mode

4.17.1: Analog Velocity Mode Function Overview

This function is used to change the operating mode of the amplifier to analog velocity mode and configure the mode.

4.17.2: Adding an Analog Velocity Mode Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Mode:Analog Velocity Mode** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Analog Velocity Mode parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Scaling	Sets the input-voltage to-velocity command ratio. Register units: Counts / second = 10 V.
Dead Band	Sets dead band. The amplifier treats anything within the dead band ranges as zero, and subtracts the dead band value from all other values. Register units: mV .
Offset	Offset applied to the input voltage. Register units: mV.
Invert Command	Inverts polarity of amplifier output with respect to input signal.

4.17.3: Analog Velocity Mode Notes

Errors

A sequence error will occur if this function is executed when:

- The amplifier is hardware disabled.
- The amplifier is faulted.
- Motor phasing is not initialized.
- A register used for a function parameter contains a value that is not valid for the parameter.

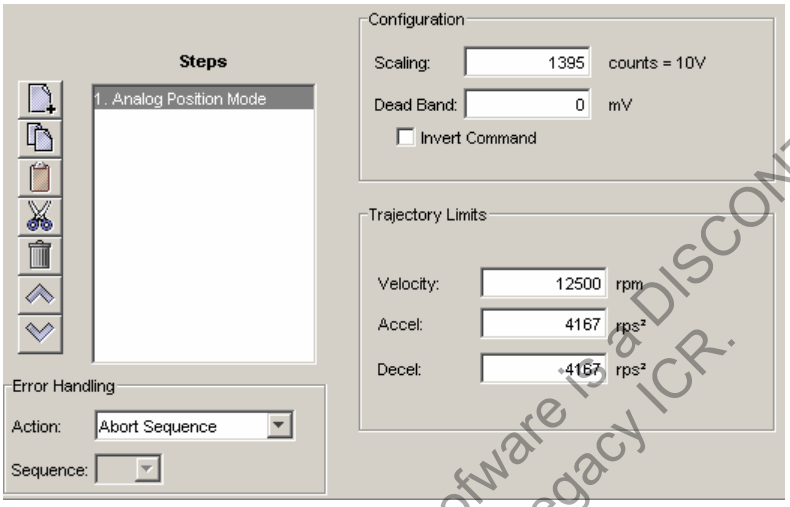
4.18: Analog Position Mode

4.18.1: Analog Position Mode Function Overview

This function is used to change the operating mode of the amplifier to Analog Position mode and configure the mode.

4.18.2: Adding an Analog Position Mode Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Mode: Analog Position Mode** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Analog Position Mode parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Scaling	Controls the input-to-command ratio. Position change produced by +10 Vdc of input. Register units: Counts = 10 V.
Dead Band	Sets dead band. The amplifier treats anything within the dead band ranges as zero, and subtracts the dead band value from all other values. Register units: mV .
Invert Command	Inverts polarity of amplifier output with respect to input signal.
Velocity	Maximum velocity that will be commanded during a move. Register units: 0.1 counts/s.
Accel	Maximum acceleration that will be used during a move. Register units: 10 counts/s^2.
Decel	Maximum deceleration that will be used during a move. Register units: 10 counts/s^2.

4.18.3: Analog Position Mode Notes

The analog position command operates as a relative motion command. When the amplifier is enabled the voltage on the analog input is read. Then, any change in the command voltage will move the axis a relative distance, proportional as programmed to the change in voltage, from its position when enabled.

To use the analog position command as an absolute position command, the amplifier should be homed every time it is enabled. A simple way to accomplish this is to program the priority sequence to use the same input as the enable input. The priority sequence would be programmed to home the axis. In this way, the axis will be automatically homed every time the amplifier is enabled.

Errors

A sequence error will occur if this function is executed when:

- The amplifier is hardware disabled.
- The amplifier is faulted.
- Motor phasing is not initialized.
- A register used for a function parameter contains a value that is not valid for the parameter.

4.19: Disable Amplifier

4.19.1: Disable Amplifier Function Overview

This function is used to software disable the amplifier.

4.19.2: Adding an Disable Amplifier Step to a Sequence

- Click **Add New Step**  on the *Program* tab to open *Indexer Functions*.
- Click **Mode: Disable Amplifier** to select the function. Click **Add** to add it to the sequence.

4.19.3: Disable Amplifier Notes

The amplifier will automatically re-enable with the execution of any function, such as Home or Move, which requires axis motion.

Errors

This function does not generate an error.

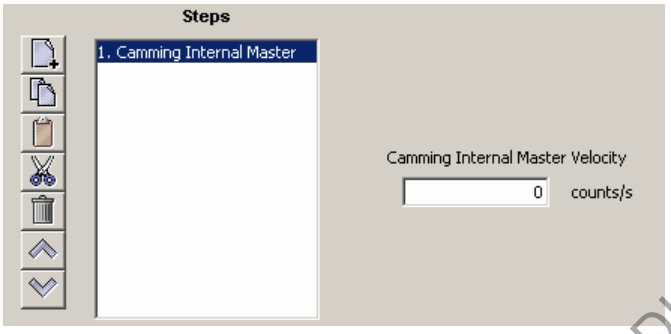
4.20: Camming Internal Master

4.20.1: Camming Internal Master Function Overview

This function is used to change the velocity of the Camming Internal Master.

4.20.2: Adding a Camming Internal Master Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Mode: Camming Internal Master** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Camming Internal Master settings.



- Choose appropriate values for the following parameter. Enter the velocity directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Velocity	Velocity of the Camming Internal Master. Register units: 0.1 counts/s.

4.20.3: Camming Internal Master Function Notes

Errors

A sequence error will occur if this function is executed when a register used for a function parameter contains a value that is not valid for the parameter.

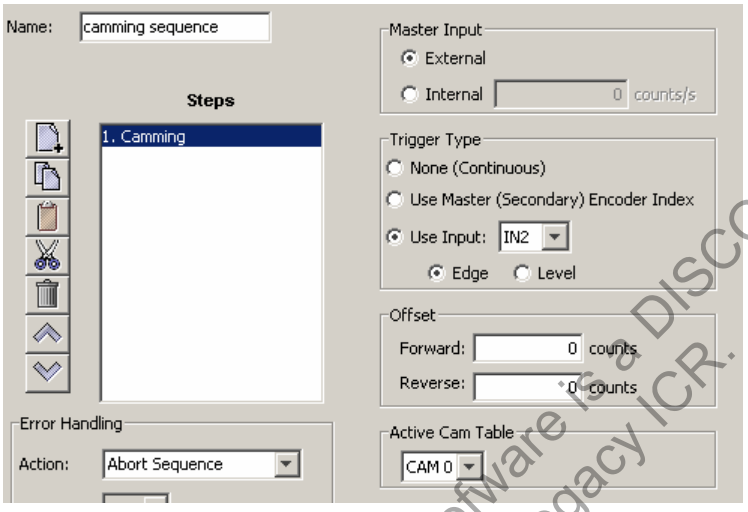
4.21: Camming

4.21.1: Camming Function Overview

This function is used to change the operating mode of the amplifier to Camming and to configure the Camming mode.

4.21.2: Adding a Camming Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Mode: Camming** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Camming mode settings.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Master Input	**External:** Source of camming master pulse is external. **Internal:** Source of camming master pulse is internal generator. Velocity of internal master pulse is set using the **counts/s** field. It can also be reset using the Camming Internal Master function.
Trigger Type	Controls how execution of a set of Camming moves is triggered: **None (Continuous):** The active Cam Table is executed continuously. **Use Master (Secondary) Encoder Index:** The active Cam Table is executed when the amplifier receives an index pulse from the camming master encoder. Index pulses received during execution are ignored. **Use Input, Edge:** The active Cam Table begins executing on the rising edge of the designated input pin. **Use Input, Level:** The active Cam Table repeats continuously as long as the designated input is active.
Offset	**Forward:** A delay (in counts) applied before the active Cam Table is executed when the Master is moving forward. **Reverse:** A delay (in counts) applied before the active Cam Table is executed when the Master is moving in reverse.
Active Cam Table	The table containing the move profile that the amplifier will execute when the camming trigger is activated.

4.21.3: Camming Function Notes

- The velocity of internal master pulse can be changed using the Camming Internal Master function.
- The amplifier will stay in the camming mode of operation until another Indexer 2 Program function changes it or the amplifier is reset or power cycled.

Errors

A sequence error will occur if this function is executed when:

- The amplifier is hardware disabled.
- The amplifier is faulted.
- Motor phasing is not initialized.
- A register used for a function parameter contains a value that is not valid for the parameter.

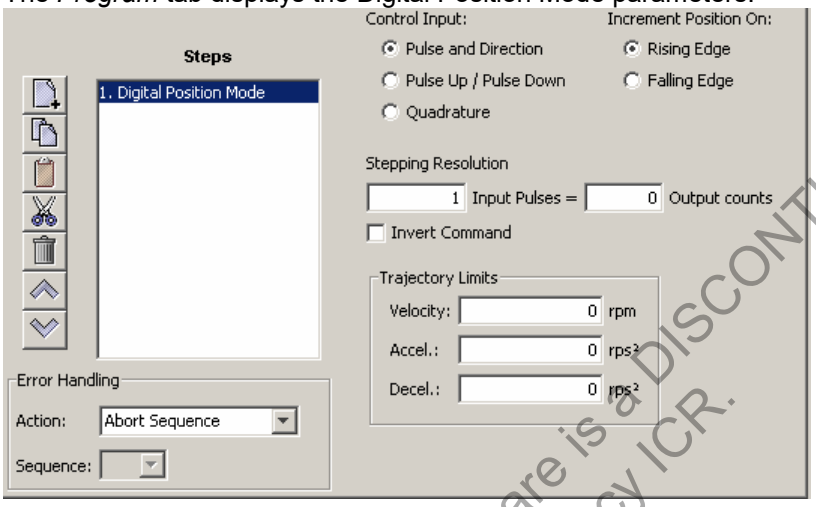
4.22: Digital Position Mode

4.22.1: Digital Position Mode Function Overview

Change the operating mode of the amplifier to Digital Position mode and configure the mode.

4.22.2: Adding an Digital Position Mode Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Mode:Digital Position Mode** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Digital Position Mode parameters.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Control Input	**Pulse and Direction:** One input takes a series of pulses as motion step commands, and another input takes a high or low signal as a direction command. **Pulse Up / Pulse Down:** One input takes each pulse as a positive step command, and another takes each pulse as a negative step command. **Quadrature:** A/B quadrature commands from a master encoder (via two inputs) provide velocity and direction commands.
Increment position on	**Rising Edge**: Increment position on the rising edge of the input pulse. **Falling Edge:** Increment position on the falling edge of the input pulse.
Stepping Resolution	**Input Pulses:** Number of Input Pulses/Counts required to produce given number of output counts. **Output Counts:** Number of Output Counts per given number of input pulses. Range: 1 to 32,767. Register Units: Pulses/Counts.
Invert Command	When checked, inverts commanded direction.
Velocity	Velocity during the constant velocity portion of the move. Register units: 0.1 counts/s.
Accel	Acceleration rate of trapezoidal profile. Register units: 10 counts/s^2.
Decel	Deceleration rate of trapezoidal profile. Register units: 10 counts/s^2.

4.22.3: Digital Position Mode Notes

Errors

A sequence error will occur if this function is executed when:

- The amplifier is hardware disabled.
- The amplifier is faulted.
- Motor phasing is not initialized.
- A register used for a function parameter contains a value that is not valid for the parameter.

4.23: Velocity Loop Single Pole Output Filter

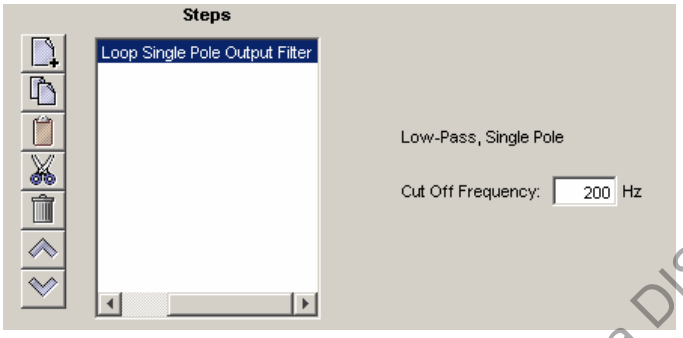
4.23.1: Velocity Loop Single Pole Output Filter Function Overview

This function modifies the Velocity Loop Output Filter.

4.23.2: Adding a Velocity Loop Single Pole Output Filter Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Filters:Velocity Loop Single Pole Output Filter** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Filter parameters.



- Enter a Cut Off Frequency value of between 1 and 1500.

4.23.3: Velocity Loop Single Pole Output Filter Notes

Errors

This function does not generate sequence errors.

4.24: Set Output

4.24.1: Set Output Overview

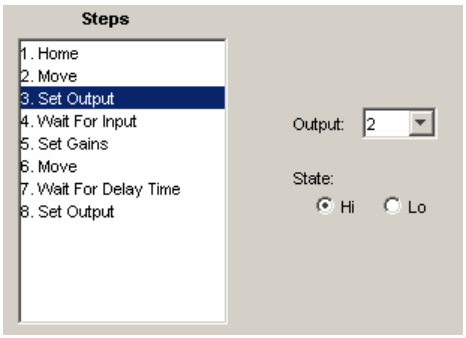
Set Output sets an output to the specified state.

NOTE: When this function is executed, it changes the configuration of the selected output to "Program Control, Active Low." The "Program Control, Active Low" configuration will stay in effect until the amplifier is reset. Consider this effect when programming digital outputs.

4.24.2: Adding a Set Output Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Input/Output: Set Output** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Set Output parameters.



- Choose appropriate values for the following parameters.

Parameter	Description
Output	Selects which output will be set.
State	• Hi: Sets the selected output high or off. • Lo: Sets the selected output low or on.

4.25: Position Triggered Output

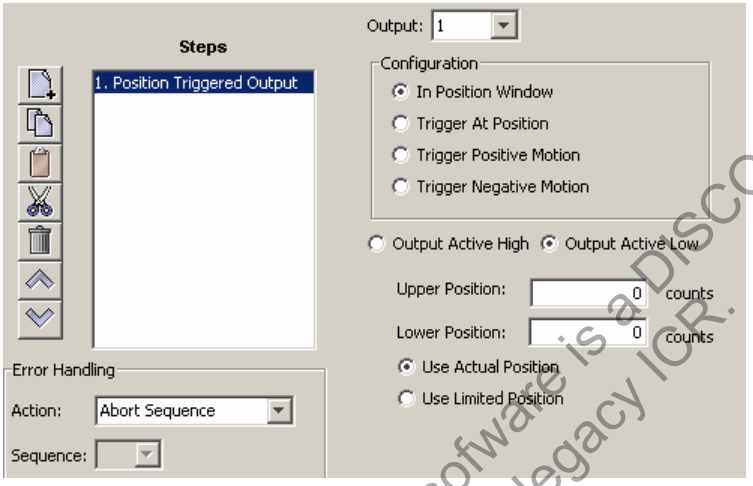
4.25.1: Position Triggered Output Function Overview

This function programs the amplifier to activate a digital output when the specified position criteria are met.

4.25.2: Adding a Position Triggered Output Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Input/Output: Position Triggered Output** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Position Triggered Output settings.



- Choose appropriate values for the basic settings and configurations described below. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Basic Setting	Description
Output	The digital output that is being programmed.
Output Active High	Output goes high when activated.
Output Active Low	Output goes low when activated.
Use Actual Position	Use the actual position for the target position.
Use Limited Position	Use the limited position for the target position. Typically used for stepper motors operated in open-loop stepper mode.

Configuration	Description and Parameters
In Position Window	Activates the output while the axis is in the window between the programmed **Upper Position** and **Lower Position**.
Trigger at Position	Activates the output for the programmed **Time** when the axis travels through the programmed **Position**.
Trigger Positive Motion	Activates the output for the programmed **Time** when the axis travels in the positive direction through the programmed **Position**.
Trigger Negative Motion	Activates the output for the programmed **Time** when the axis travels in the negative direction through the programmed **Position**.

4.25.3: Position Triggered Output Notes

When this function is executed, it changes the configuration of the selected output to the chosen Position Triggered Output configuration. This configuration of the output will stay in effect until the amplifier is reset or the output configuration is changed by another function.

Errors

A sequence error will occur if this function is executed when a register used for a function parameter contains a value that is not valid for the parameter.

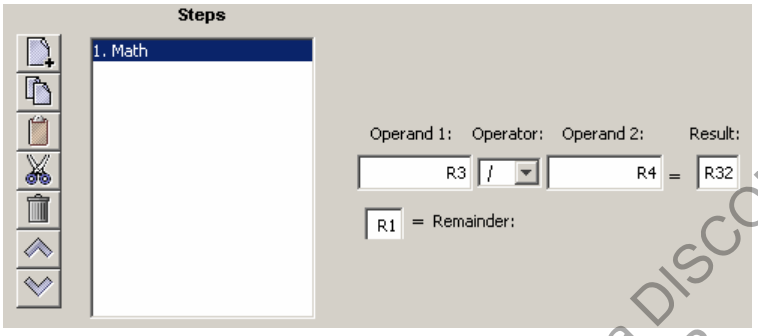
4.26: Math

4.26.1: Math Function Overview

This function is used to perform basic integer math and assign the result to an Indexer 2 Program register.

4.26.2: Adding a Math Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Tools: Math** to select the function. Click **Add** to add it to the sequence.
  The *Program* tab displays the Math settings.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Operand 1	An operand. Value can be an integer constant or a reference to an Indexer 2 Program register. *
Operator	Addition, Subtraction, Multiplication, or Division.
Operand 2	An operand. Value can be an integer constant or a reference to an Indexer 2 Program register.*
Result	The register to which the math results will be written. For division, the result value is the integer portion of the quotient.
Remainder	The register to which the remainder portion of the quotient will be written.
For multiplication operations, the Math function imposes a value range of -32768 through 32767. To multiply numbers outside of this range, use the function Extended Math (p. 74).	

4.26.3: Math Function Notes

The Math function does not support carry on addition or borrow on subtraction. Thus, if a Math operation results in a number that exceeds the capacity of the register (between 2^{31}-1 and -2^{31}) the results will be incorrect (but no error will be generated).

Errors

A sequence error will occur if:

- During multiplication, one of the operands is a register with a value less than -32768 or greater than 32767.
- During division, Operand 2 is a register with a value equal to 0.

4.27: If Register Set

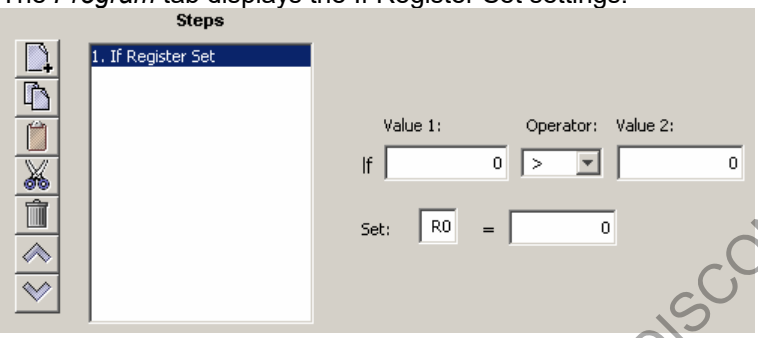
4.27.1: If Register Set Function Overview

This function is used to conditionally assign a value to an Indexer 2 Program register.

4.27.2: Adding an If Register Set Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Tools: If Register Set** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the If Register Set settings.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Value 1	One of the two values being compared. Value can be an integer constant or a reference to an Indexer 2 Program register.
Operator	• Greater than • Less than • Equal to • Not equal to • Greater than or equal • Less than or equal.
Value 2	One of the two values being compared. Value can be an integer constant or a reference to an Indexer 2 Program register.
Set	On the left side of the equation, enter the register to be set. On the right side of the equation, enter an integer constant or a reference to an Indexer 2 Program register.

4.27.3: If Register Set Notes

When the Indexer 2 Program has been set up to use a register to select a sequence, the If Register Set function can determine which sequence will execute after the current sequence has finished.

Errors

This function does not generate an error.

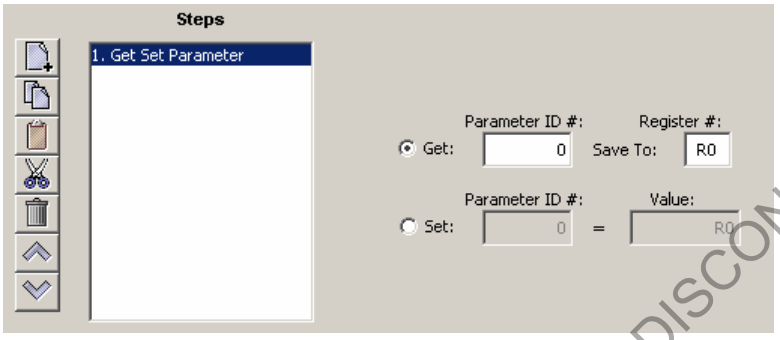
4.28: Get Set Parameter

Get Set Parameter Function Overview

This function is used to get or set the value of one the amplifier's internal parameters.

4.28.1: Adding a Get Set Parameter Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Tools: Get Set Parameter** to select the function. Click **Add** to add it to the sequence. The *Program* tab displays the Get Set Parameter settings.



- Choose appropriate values for the following settings. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Setting	Description
Get	Copies the current value of the specified amplifier parameter into the register.
Parameter ID #	The ID of the amplifier parameter that the function will read. For a list of amplifier parameter IDs, see the *Copley Parameter Dictionary.* Use the ASCII ID value given in that book.
Register #	The value of the amplifier parameter will be copied into this register.
Set	Copies a value into the specified amplifier parameter.
Parameter ID #	The ID of the amplifier parameter that the function will write.
Value	The value in this field or the value in the specified register will be copied to the amplifier parameter.

4.28.2: Get Set Parameter Notes

Parameter ID's can be entered as decimal or hexadecimal numbers. When entering hexadecimal numbers, use the 0x notation (for instance, 0x00c1). The Indexer 2 Program will convert and display all values in hexadecimal form.

Errors

A sequence error will occur if:

- The specified parameter ID does not exist.
- The specified parameter ID is for a flash only parameter.
- A set is attempted on a read only parameter.
- A set is attempted and the value is illegal for the parameter being set.

4.29: Conditional Jump

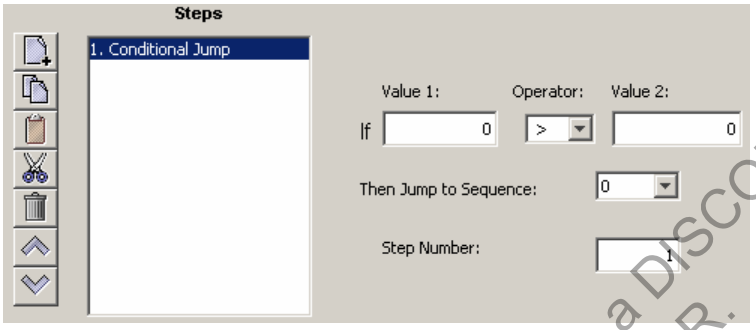
4.29.1: Conditional Jump Overview

This function is used to conditionally jump to a specified step within a specified sequence. If the conditional statement is not true, the Indexer 2 Program continues to the next step in the original sequence.

4.29.2: Adding a Conditional Jump Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Tools: Conditional Jump** to select the function. Click **Add** to add it to the sequence.

  The *Program* tab displays the Conditional Jump settings.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Setting	Description
Value 1	One of the two values being compared. Value can be an integer constant or a reference to an Indexer 2 Program register.
Operator	• Greater than • Less than • Equal to • Not equal to • Greater than or equal • Less than or equal.
Value 2	One of the two values being compared. Value can be an integer constant or a reference to an Indexer 2 Program register.
Then Jump to Sequence	Sequence to jump to when condition is met. Note that this can be the same sequence that initiates the Conditional Jump.
Step Number	Step within sequence to jump to when condition is met.

4.29.3: If Register Then Notes

If a jump is made to a sequence that has not been programmed or to a step in a sequence that does not exist, an error is not generated. Instead, the Indexer 2 Program will return to an idle state waiting for the next Go command.

Errors

This function does not generate sequence errors.

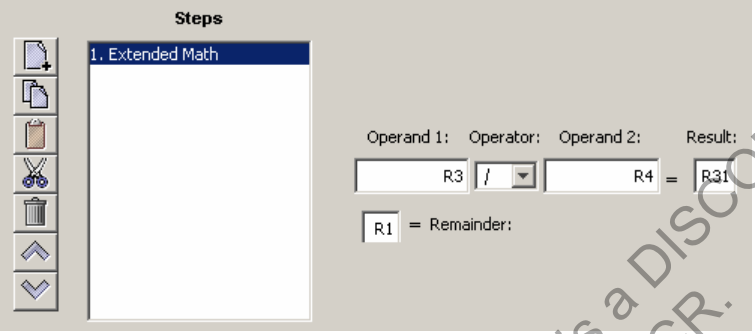
4.30: Extended Math

4.30.1: Extended Math Function Overview

This function is used to perform basic integer math and assign the result to an Indexer 2 Program register. Extended Math is identical to Math (p. 70) except that Extended Math does not impose a size limit on multiplication operands.

4.30.2: Adding an Extended Math Step to a Sequence

- Click **Add New Step** on the *Program* tab to open *Indexer Functions*.
- Click **Tools: Extended Math** to select the function. Click **Add** to add it to the sequence.
  The *Program* tab displays the Extended Math settings.



- Choose appropriate values for the following parameters. Enter numeric values directly or enter the number (R0-R31) of an Indexer 2 Program register. See Using Registers to Pass Values to Functions (p. 38).

Parameter	Description
Operand 1	An operand. Value can be an integer constant or a reference to an Indexer 2 Program register.
Operator	Addition, Subtraction, Multiplication, or Division
Operand 2	An operand. Value can be an integer constant or a reference to an Indexer 2 Program register.
Result	The register to which the math results will be written. For division, the result value is the integer portion of the quotient.
Remainder	The register to which the remainder portion of the quotient will be written.

4.30.3: Extended Math Function Notes

The Math function does not support carry on addition or borrow on subtraction. Thus, if a Math operation results in a number that exceeds the capacity of the register (between 2^{31}-1 and -2^{31}) the results will be incorrect (but no error will be generated).

Errors

A sequence error will occur if:

- During division, Operand 2 is a register with a value equal to 0.

.

APPENDIX

A: ASCII COMMANDS OVER SERIAL

The Indexer 2 Program provides 32 four-byte registers. They can be used to:

- Select a sequence. See Use a Register to Select a Sequence (p. 15).
- Initiate a Go command. See Use a Register to Initiate Go (p. 16).
- Pass values to the Indexer 2 Program. See Using Registers to Pass Values to Functions (p. 38).

This chapter describes how to set up connections for serial control and issue ASCII commands to access Indexer 2 Program registers.

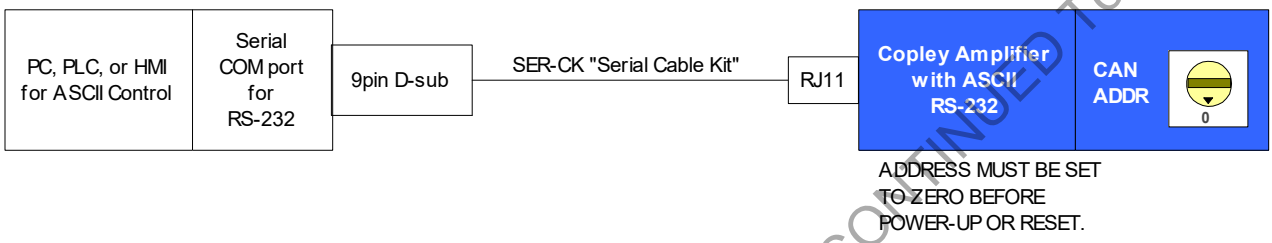
Contents include:

A.1: Connecting

This section describes how to connect an amplifier for control via the RS-232 bus. The serially connected amplifier can also be used as a multi-drop gateway for access to other amplifiers linked in a series of CAN bus connections. Instructions for hooking up a single-axis connection and a multi-drop network appear below.

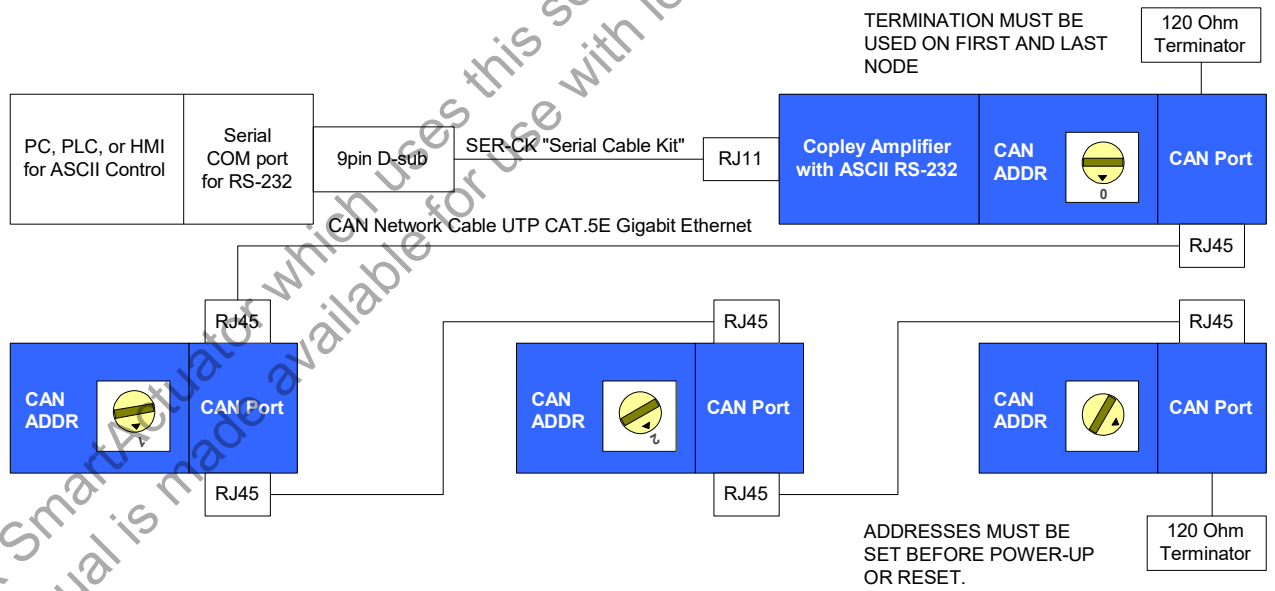
A.1.1: Single-Axis Connections

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



A.1.2: Multi-Drop Network Connections

A serially connected amplifier can be used as a multi-drop gateway for access to other amplifiers linked in a series of CAN bus connections. Set the CAN node address of the serially connected amplifier (gateway) to zero (0). Assign each additional amplifier in the chain a unique CAN node address value between 1 and 127. For more information on CAN node address assignment, see the *CAN Network Configuration* chapter in the *CME 2 User Guide*. Use 120 Ohms termination on the first and last amplifier.



A.2: Communications Protocol

Use the following protocol for reading and writing Indexer 2 Program registers:

Baud Rate	9,600
Data Format	N, 8, 1
Flow Control	None

A.3: Reading and Writing Registers

To read or write the Indexer 2 Program's 32 registers with Copley ASCII commands, use the "i" (for Indexer) command with the "r" (for register) argument as shown here:

[optional node ID] **i r***n* [optional write value]<cr>

where:

- [optional node ID] is the CAN node address of an amplifier in a multi-drop network. Range is 0-127. The node ID should be followed by a single space. If omitted, the serially connected amplifier is addressed by default.
- **i r**n is the command code for accessing the Indexer 2 Program registers. Lowercase ASCII character i followed by a single space followed by r followed by a number n, where n is the number of the register to read or write. Range 0-31.
- [optional write value] is the value to write to the register. A space is required between the register number and this value. If no value is entered, the command returns the value in the specified register. The value can be entered as a positive or negative integer or in hex format.
- <cr> is a carriage return charter that ends the command line.

If a write command was accepted by the Indexer 2 Program, it returns an "ok" followed by a carriage return character. If a read command was accepted by the amplifier, it returns an "r " followed by the decimal equivalent of the registers value followed by a carriage return character. Examples:

Operation	Command	Response
Read R31 via direct serial connection	i r31<cr>	r [register contents]<cr>
Write value 500 to R0 via direct serial connection	i r0 500<cr>	ok<cr>
Write value -500 to R31 via direct serial connection	i r31 -500<cr>	ok<cr>
Read R0 on CAN node amplifier #15 on a multi-drop network	15 i r0<cr>	r [register contents]<cr>
Write hex value 800d to R31 on CAN node amplifier #15 on a multi-drop network	15 i r31 0x800d<cr>	ok<cr>

If there was a problem with a command, the amplifier returns an "e" followed by an error code followed by a carriage return character. Error codes

Code	Meaning
10	The passed data value is out of range.
31	Invalid node ID.
32	CAN network communications failure.
33	ASCII command parse error.
34	Internal amplifier error.

Note that all register values are cleared when the Indexer 2 Program starts.

For experimentation and simple setup and control, the CME 2 **Tools→ASCII Command Line** tool can be used. In addition, a telnet device such as the standard Microsoft Windows HyperTerminal can be used to send commands in ASCII format. Use the protocol described in Communications Protocol (p. 76).

APPENDIX

B: HOMING METHOD DESCRIPTIONS

This appendix describes the homing methods that can be chosen using the Indexer 2 Program Home function (see Home, p. 53).

Contents include:

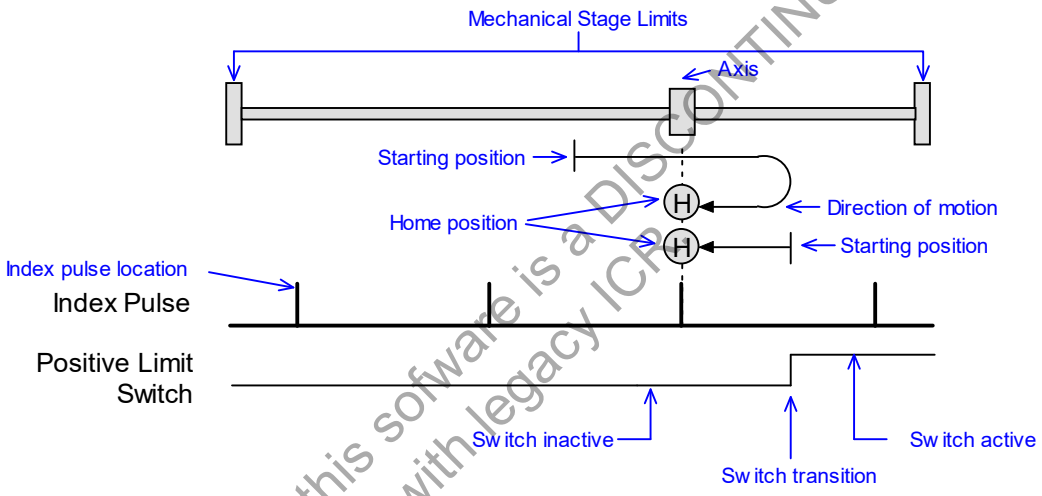
B.1: Homing Methods Overview

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.

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

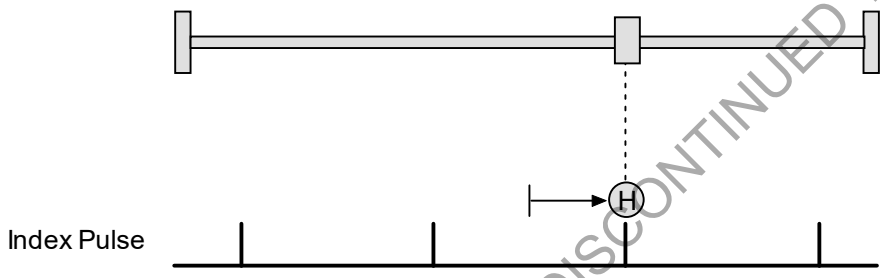
B.3.1: Set current position as home

The current position is the home position.

B.3.2: Next Index

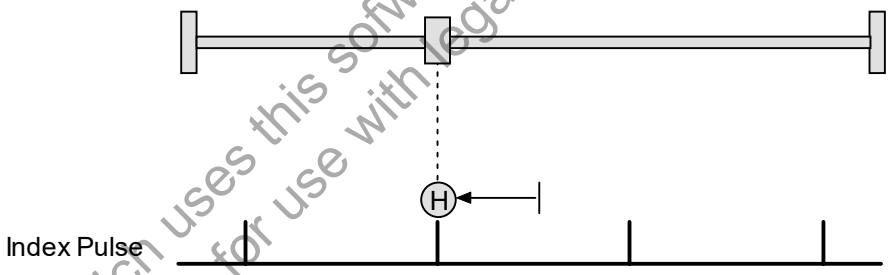
Direction of Motion: Positive

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

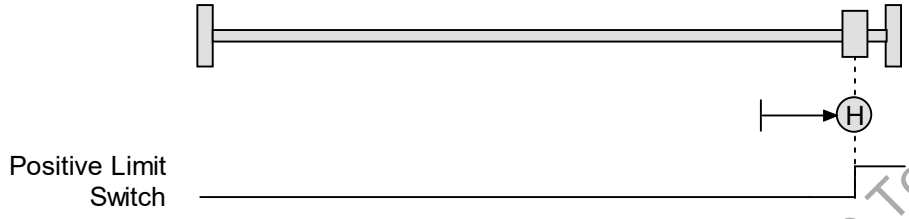
Home is the first index pulse found in negative direction. Direction of motion is negative. If a negative limit switch is activated before the index pulse, an error is generated.



B.3.3: Limit Switch

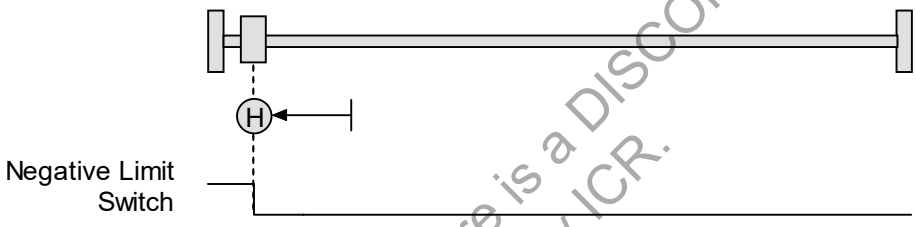
Direction of Motion: Positive

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

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



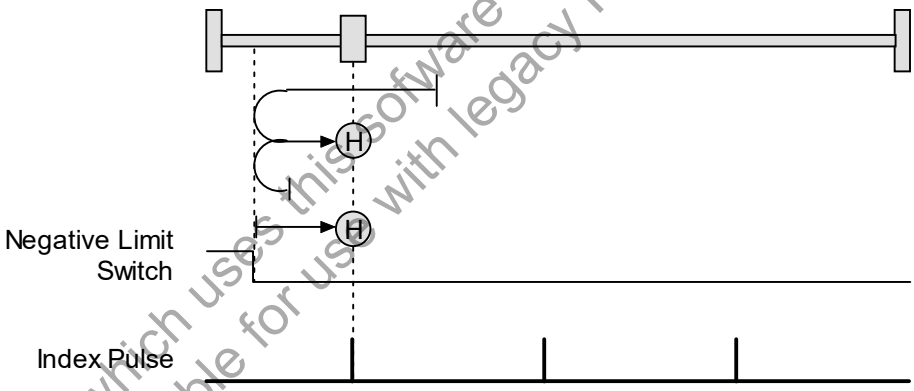
B.3.4: Limit Switch Out to Index

Direction of Motion: Positive

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

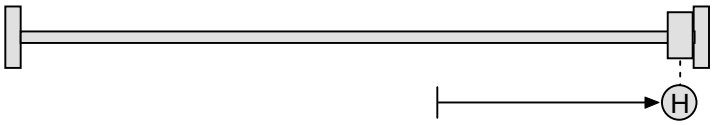
Home is the first index pulse to the positive side of the negative limit switch transition. Initial direction of motion is negative if the negative limit switch is inactive (shown here as low).



B.3.5: Hardstop

Direction of Motion: Positive

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.



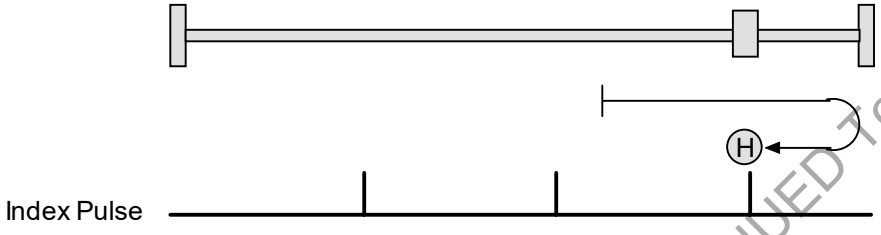
Direction of Motion: Negative

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

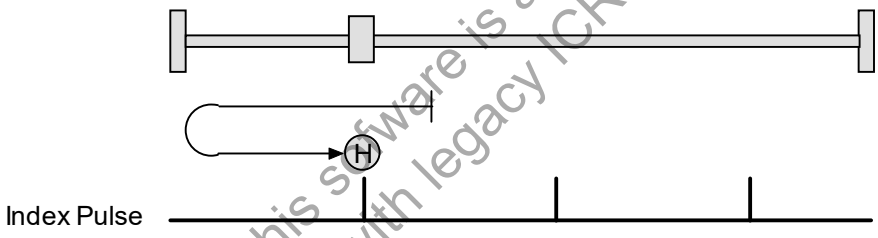
Direction of Motion: Positive

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

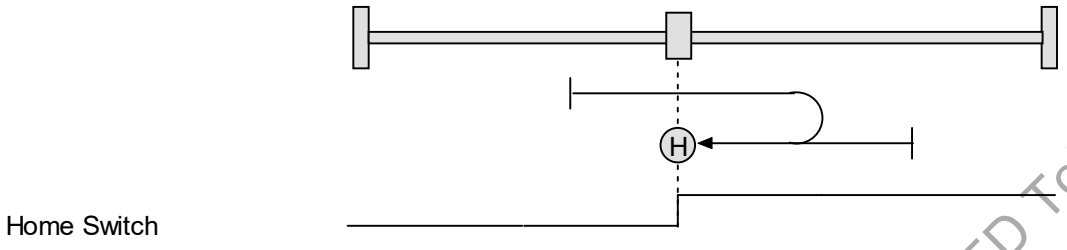
Home is the first index pulse on the positive side of the negative hard stop. Initial direction of motion is negative. The hard stop is reached when the amplifier outputs the homing Current Limit continuously for the amount of time specified in the Delay Time. If a negative limit switch is activated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



B.3.7: Home Switch

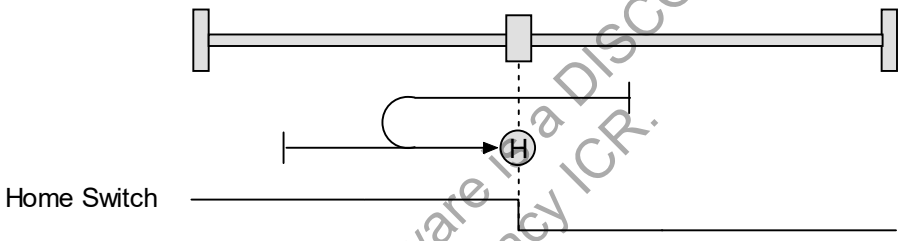
Direction of Motion: Positive

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

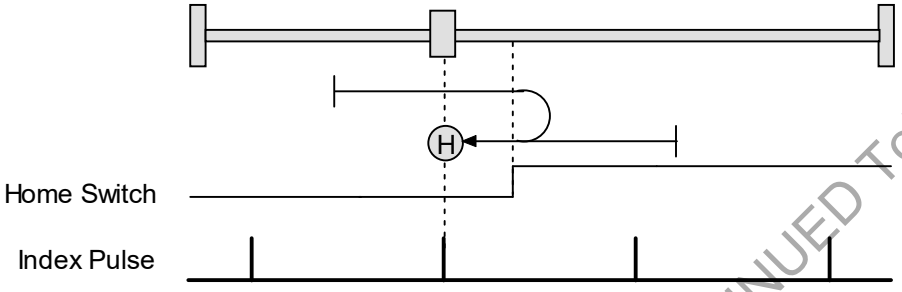
Home is the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



B.3.8: Home Switch Out to Index

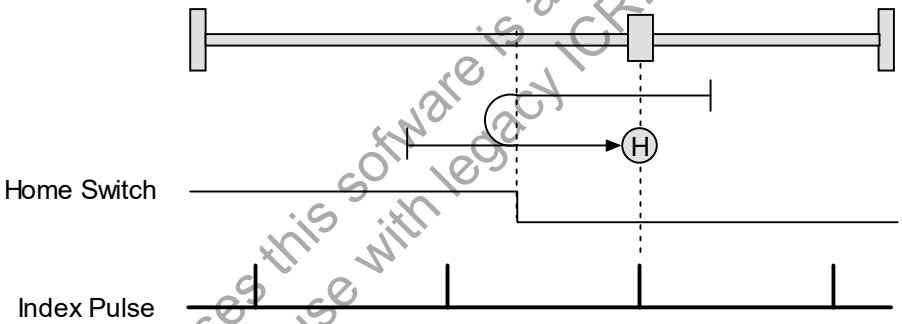
Direction of Motion: Positive

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

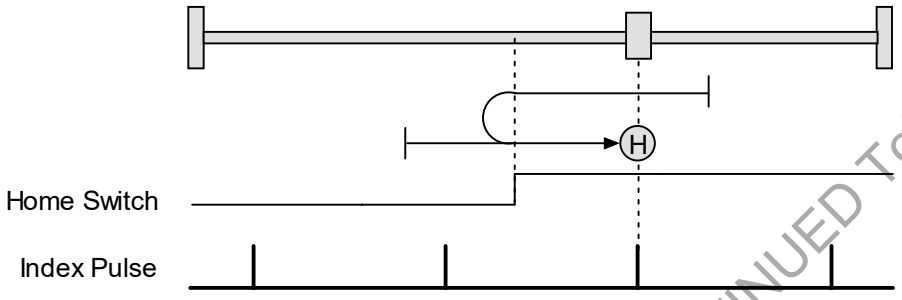
Home is the first index pulse to the positive side of the home switch transition.
Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



B.3.9: Home Switch In to Index

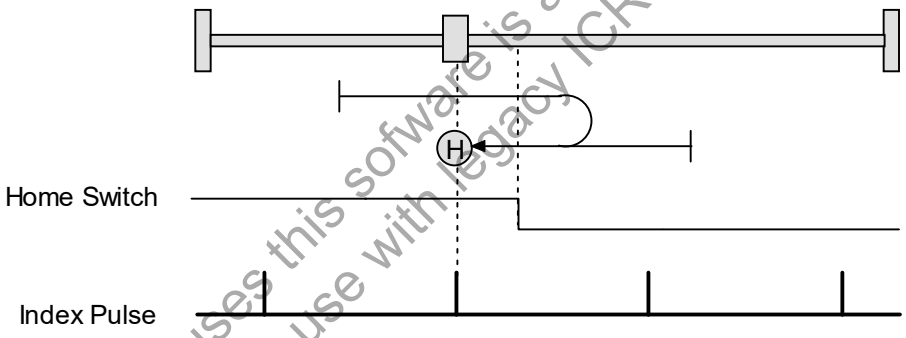
Direction of Motion: Positive

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

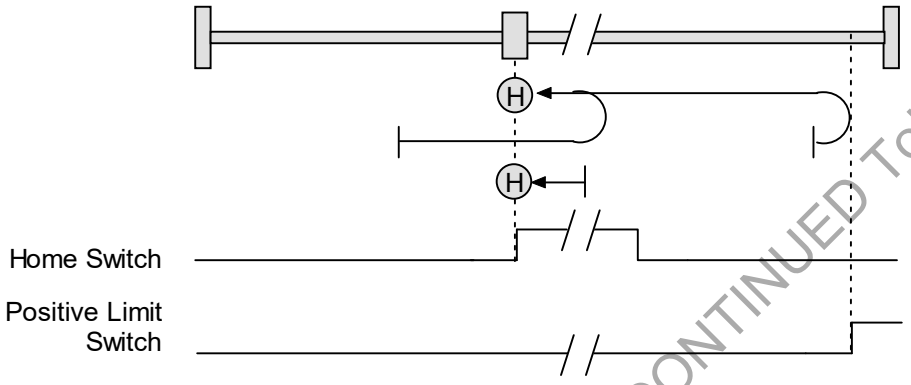
Home is the first index pulse to the negative side of the home switch transition. Initial direction of motion is negative if the home switch is inactive. If a limit switch is activated before the home switch transition, an error is generated.



B.3.10: Lower Home

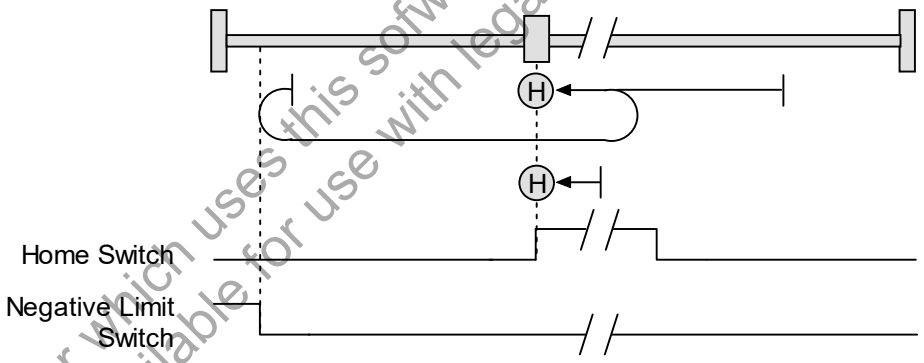
Direction of Motion: Positive

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

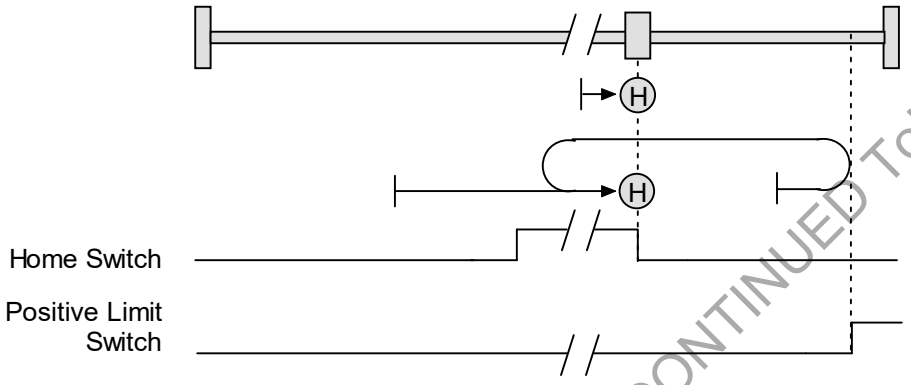
Home is the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home switch, an error is generated.



B.3.11: Upper Home

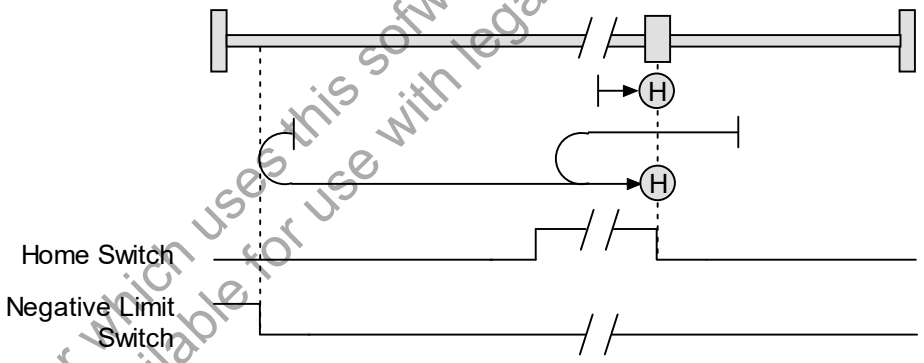
Direction of Motion: Positive

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

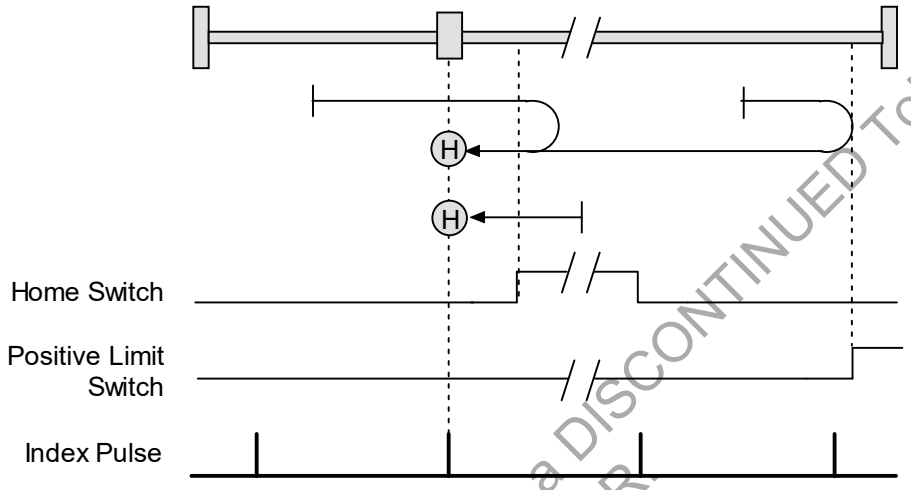
Home is the positive edge of momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a positive limit switch is activated before the home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



B.3.12: Lower Home Outside Index

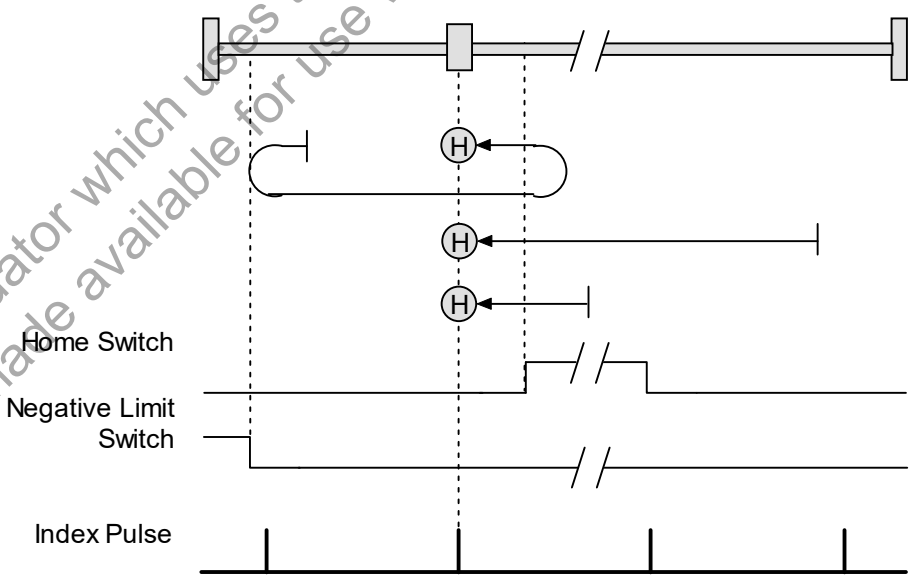
Direction of Motion: Positive

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

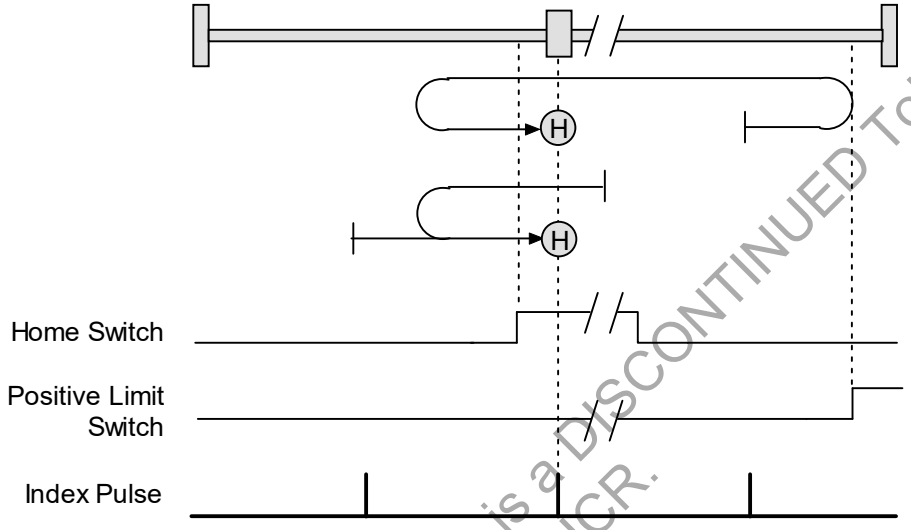
Home is the first index pulse on the negative side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated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



B.3.13: Lower Home Inside Index

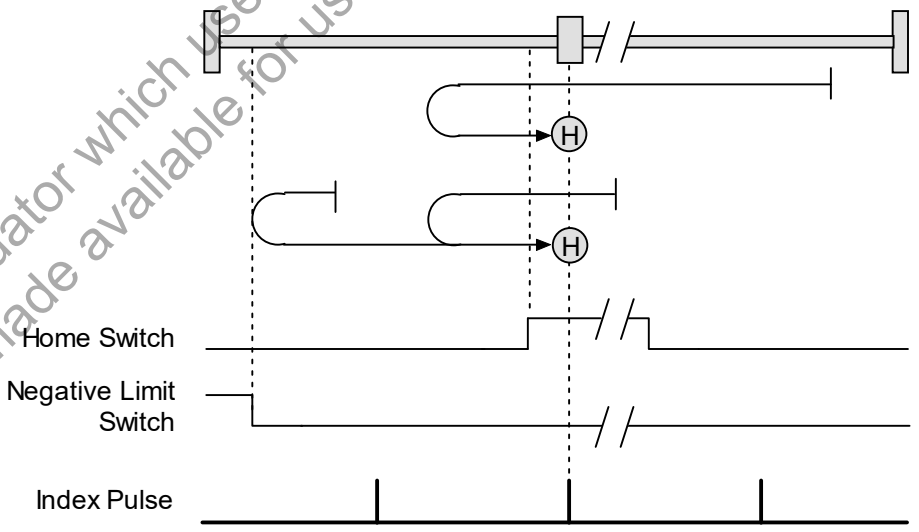
Direction of Motion: Positive

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

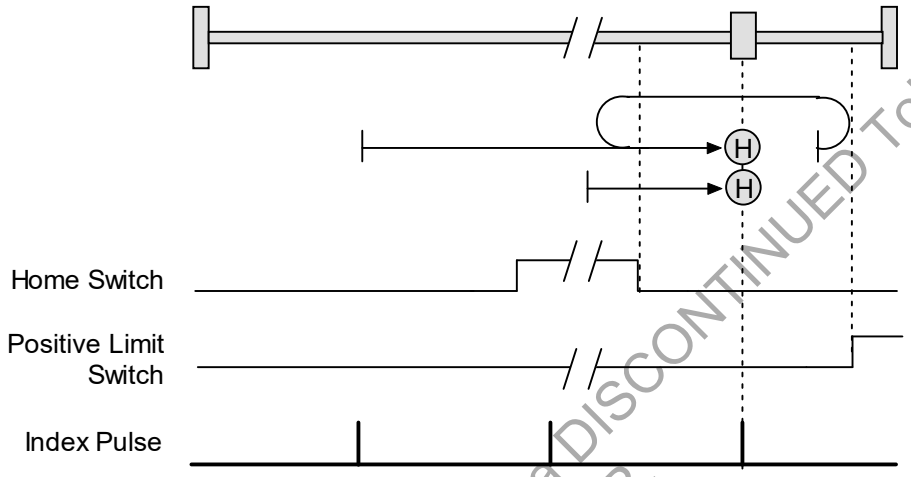
Home is the first index pulse on the positive side of the negative edge of a momentary home switch. Initial direction of motion is negative. If the initial motion leads away from the home switch, the axis reverses on encountering the negative limit switch; then, if a negative limit switch is activated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



B.3.14: Upper Home Outside Index

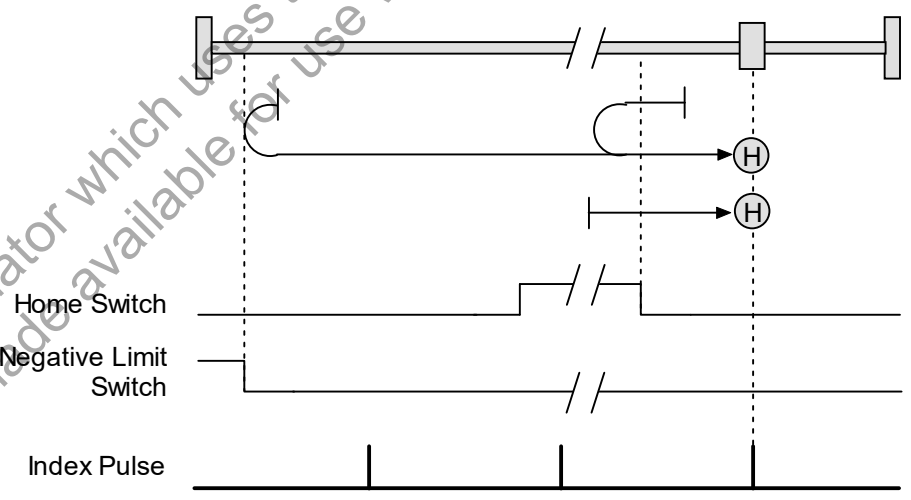
Direction of Motion: Positive

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

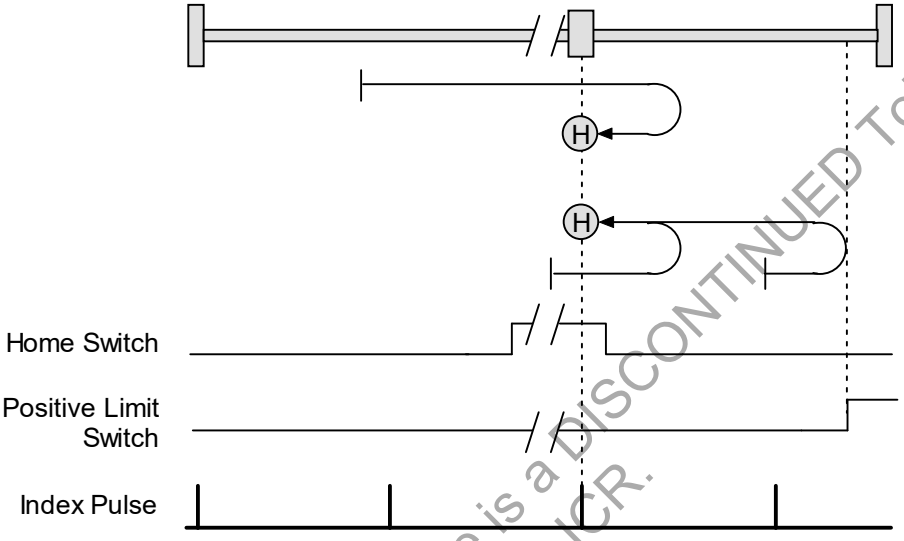
Home is the first index pulse on the positive side of the positive edge of a momentary home switch. Initial direction of motion is negative if the home switch is inactive. If the initial position is right of the home position, the axis reverses on encountering the home switch.



B.3.15: Upper Home Inside Index

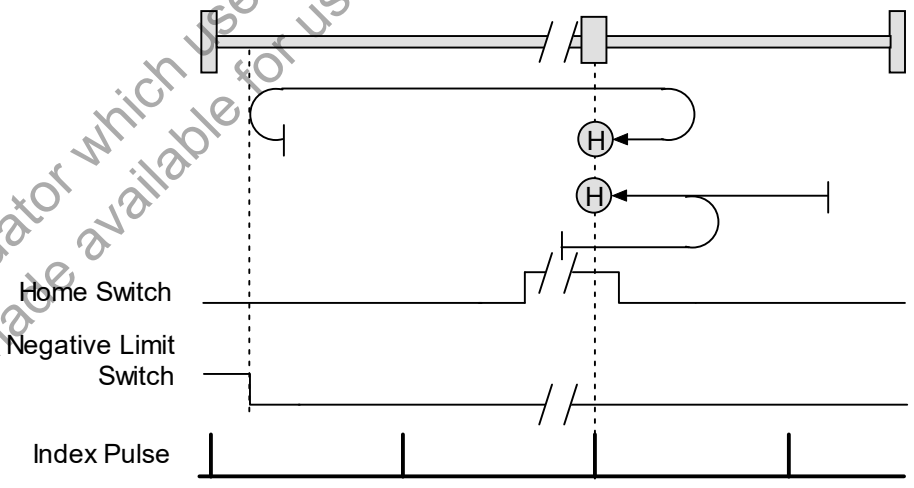
Direction of Motion: Positive

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

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



APPENDIX

C: LOADING CONFIGURATIONS FROM FILES

This appendix describes how to load amplifier and Indexer 2 Program configurations from files.

C.1: Amplifier Quick Copy Setup

Use this procedure to configure an amplifier/motor pair by copying a *.ccx* file that was prepared for the amplifier/motor combination.

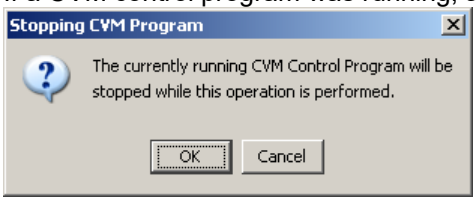
- Make sure the amplifier has either a direct serial bus connection to the PC running CME 2, or a CAN connection to the gateway amplifier that has the serial bus connection.
- 

Start CME 2 by clicking the CME 2 shortcut icon on the Windows desktop.
- In the CME 2 *Copley Neighborhood*, select the amplifier that you wish to set up.
-  On the CME 2 *Main* screen, click **Restore amplifier data from disk**.
- When the *Restore Amplifier Data from Disk* screen opens, navigate to the folder containing the appropriate *.ccx* file. Highlight the file name and then click **Open** to load the file data into volatile memory.
-  On the CME 2 *Main* screen, click **Save amplifier working memory to flash** to commit the new settings to flash memory.

C.2: CVM Program Quick Copy Setup

Use this procedure to load a CVM Program into an amplifier.

- Make sure the amplifier has either a direct serial bus connection to the PC running CME 2, or a CAN connection to the gateway amplifier that has the serial bus connection.
- 

Start CME 2 by clicking the CME 2 shortcut icon on the Windows desktop.
- In the CME 2 *Copley Neighborhood*, select the amplifier that you wish to set up.
- From the **File** menu choose **Restore CVM Control Program**.
If a CVM control program was running, a prompt appears:


- If the prompt appears, click **OK**.
- When the *Restore Control Program from Disk* screen opens, navigate to the folder containing the appropriate *.ccp* file. Highlight the file name and then click **Open** to load the file data into amplifier flash.

This procedure also results in the setting of the Indexer 2 Program option Enable Control Program on Startup (p. 25). This configures the program to auto start when the amplifier is powered up or reset.

Copley Indexer 2 Program User Guide
P/N 95-00744-000

Revision 2

June 2008

Copley Controls Corporation
20 Dan Road
Canton, MA 02021 USA