Copley Controls
Serial Encoder Guide

P/N 16-01055
Revision 01
May 2013

This page for notes

TABLE OF CONTENTS

ABOUT THIS MANUAL

1.1.1: Overview and Scope

This book describes the use and configuration of encoders on Copley Controls' Plus drives.

1.1.2: Related Documentation

- *CME 2 User Guide*
- *ASCII Programmers Guide*
- *Copley Amplifier Parameter Dictionary*
- *Copley Motion Objects Programmer's Guide*

Links to these publications can be found under Software Documents and Communications Protocols at: http://www.copleycontrols.com/Motion/Downloads/index.html.

Also of interest:

- *CANopen Programmer's Manual*
- *CML Reference Manual*
- *Copley Programming Language (CPL) User Guide*

1.1.3: Comments

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

1.1.4: Copyrights

1.1.5: Document Validity

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

1.1.6: Product Warnings

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

DANGER

Hazardous voltages.

Exercise caution when installing and adjusting Copley drives.

Risk of electric shock.

On some Copley Controls drives, high-voltage circuits are connected to mains power. Refer to hardware documentation.

Risk of unexpected motion with non-latched faults.

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

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

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

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

When operating the drive as a CAN or DeviceNet node, the use of CME 2 or ASCII serial commands may affect operations in progress. Using such commands to initiate motion may cause network operations to suspend.

Operation may restart unexpectedly when the commanded motion is stopped.

Use equipment as described.

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

FAILURE TO HEED THESE WARNINGS CAN CAUSE EQUIPMENT DAMAGE, INJURY, OR DEATH.

Revision History

Revision	Date	Comments
00	September 2012	Initial publication
01	May 2013	Updated bit info.

.

CHAPTER 1: INTRODUCTION

Absolute encoders provide position data without referencing to a mechanical home. For every position a unique binary word is generated.

In single-turn absolute encoders, words are repeated for every revolution of the encoder's shaft. In multi-turn absolute encoders, words are unique for every position, through multiple rotations of the shaft.

For both single and multi-turn encoders, the resolution of a single-turn, and the number of turns, are programmable. The total number of bits is limited to 32.

The encoder selection and configuration examples in this guide use Copley Controls' CME 2 commissioning software. For more detailed information see data sheets for encoders and Copley drives.

CHAPTER 2: ABSOLUTE ENCODERS

This chapter describes the configuration of absolute encoders using Copley's CME 2 commissioning software. Note that configuration steps may be continued on subsequent pages.

2.1: Supported Absolute Encoders

The table below is a list of absolute encoder types that are supported by the following Copley Controls drives: Accelnet Plus, Xenus Plus, and Accelnet MACRO. These Copley drives may also be used with Incremental encoders.

Encoder Type	Universal and Manufacturer Standards
SSI (p. 17)	Binary, Gray Code, Renishaw RLS, Baumer SSI, Zettlex Standard
EnDat (p. 20)	Heidenhain 2.1, Heidenhain 2.2, Kollmorgen EnDat
Absolute A (p. 26)	A format, Sanyo Denki Absolute, Panasonic Absolute A, Tamagawa Absolute A
Incremental A (p. 29)	Panasonic Incremental A
BiSS (p. 32)	Hengstler BiSS, Kollmorgen BiSS, Renishaw BiSS C
Hiperface (p.)	

2.1.1: Firmware Requirements

Updated firmware downloads are available here:
http://www.copleycontrols.com/Motion/Downloads/firmware.html

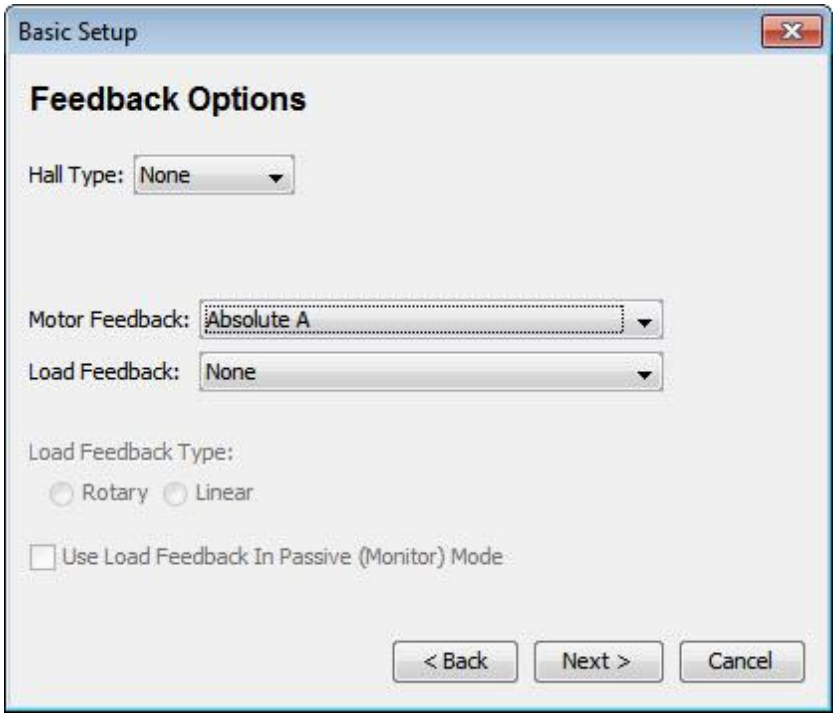
2.2: Emulated Encoder Output

Copley Controls drives can receive position feedback from sensors on the motor, through the Primary Feedback channel, and produce emulated digital encoder output using the drive's multimode port.

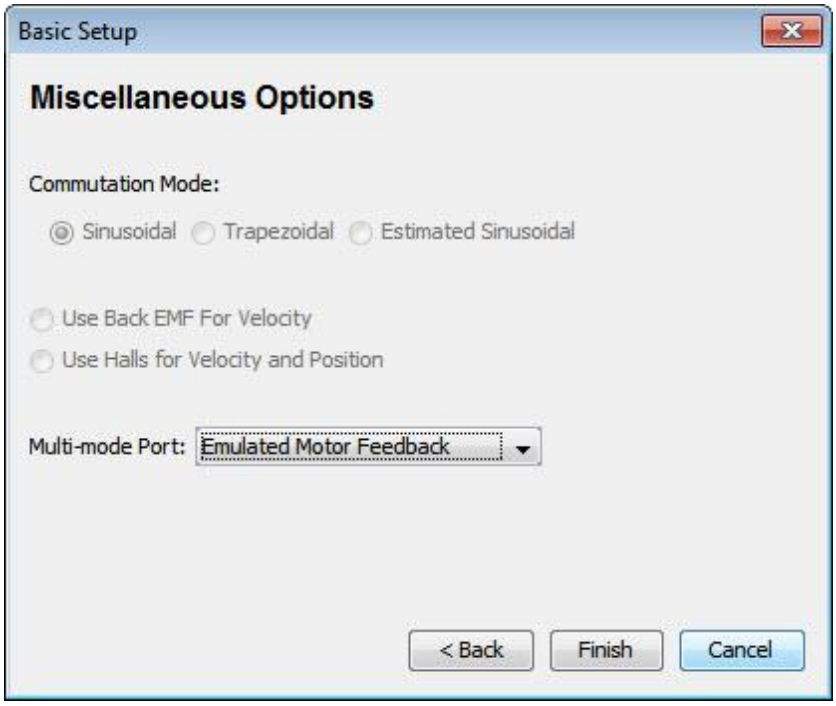
Below is an example of configuring the drive with an absolute encoder on the motor and producing an emulated output from the multi-mode port.

1 In CME2, navigate to the *Feedback Options* screen: **Amplifier→Basic Setup→Change Settings→Next**.

On the *Feedback Options* screen, set **Motor Feedback** to Absolute A, for example.



2 From the **Multi-mode Port** drop down, select Emulated Motor Feedback.

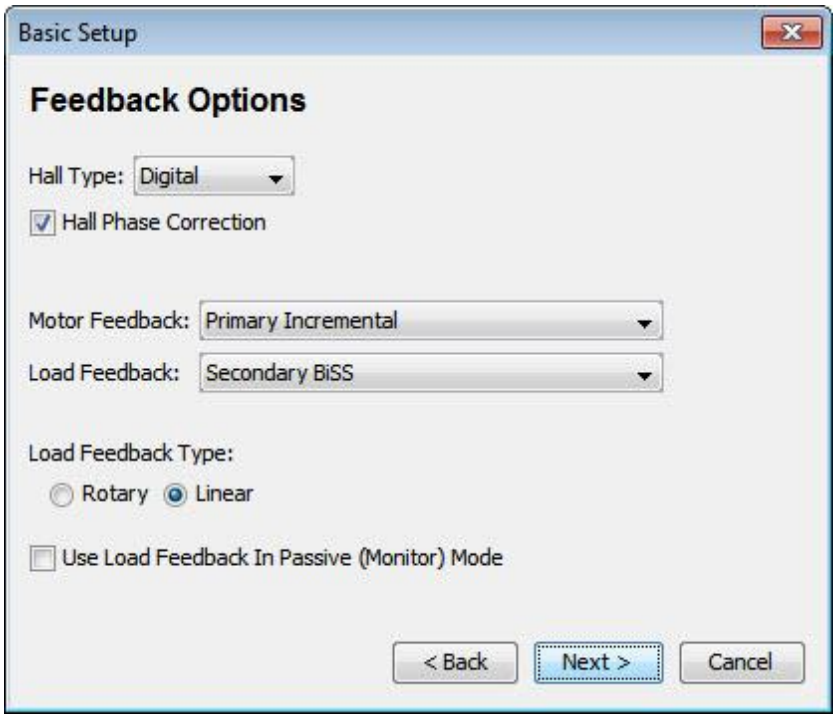


2.3: Dual Feedback

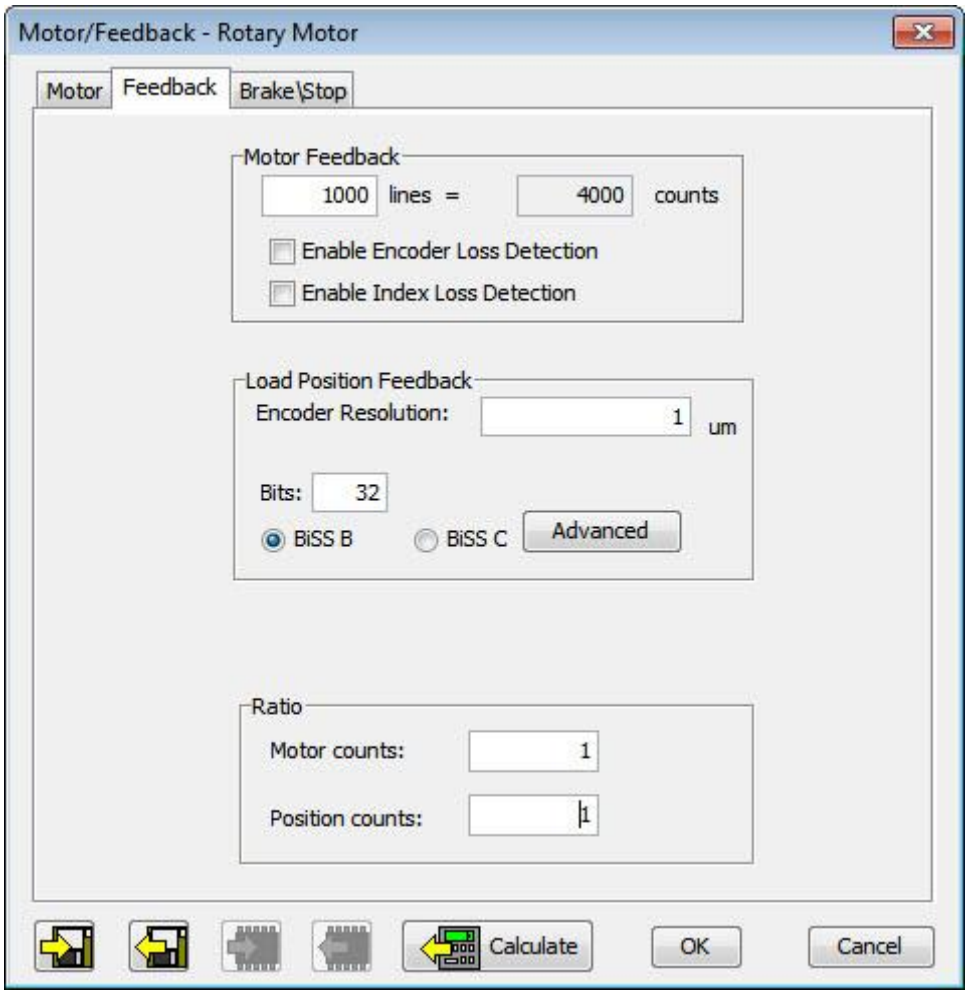
Copley Controls drives can receive position feedback from sensors on the motor and the load through the Primary and Secondary feedback channels. In dual loop control mode, an absolute encoder attached to the load provides position loop feedback, and the motor encoder or resolver provides velocity loop feedback.

In the dual-feedback setup shown below, the drive receives feedback from an incremental motor encoder through the Primary feedback channel. Position (load) encoder feedback is received through the multi-mode port. There is also an option to use the feedbacks as simple (passive) monitoring.

1 In CME2, navigate to the *Feedback Options* screen: **Amplifier→Basic Setup→Change Settings→Next**. On the CME 2 *Basic Setup Feedback Options* screen, set **Motor Feedback** to Primary Incremental and **Load Feedback** to Secondary BiSS, for example.



2 Open the *Motor/Feedback* screen *Feedback* tab.



View or change the settings described below.

Field	Description
lines	The number of encoder lines. Quadrature counts/rev will be 4 times the number of lines.
Enable Encoder Loss Detection	An encoder fault will occur when a differential encoder signal, A /A or B /B, is lost.
Enable Index Loss Detection	An encoder fault will occur when a differential index signal, X /X, is lost.
Encoder Resolution	The number of rotary counts/rev or linear distance/count.
Bits	The number of bits used for calculating position. Sign extended to 32 bits.
BiSS B	Mode choice at each start cycle.
BiSS C	Continuous mode.
Motor counts	The position from motor feedback.
Position counts	The position from load feedback.

2.4: SSI Encoders (Type 12)

This section describes the use of Synchronous Serial Interface (SSI) absolute encoders with Copley Controls drives.

2.4.1: SSI Encoder and Drive Communication

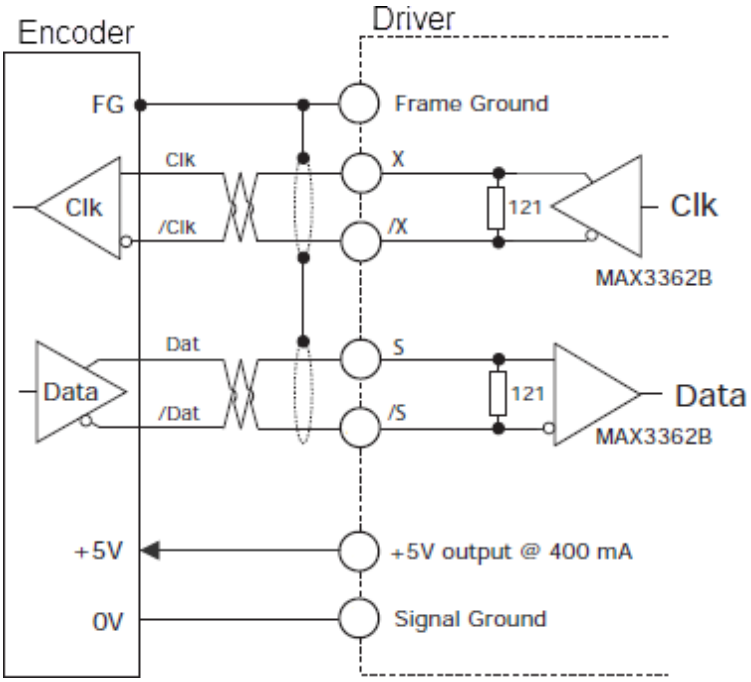
SSI encoders provide all digital one way communications to the selected Copley drive.

A train of clock pulses from the drive initiates the transmission of position data by subsequent clock pulses. The clock and data are wired as differential pairs.

An SSI encoder should have a clock frequency in the range of 1 to 4 MHz. This allows all the data to be transferred at a speed that is sufficient for the position loop update rate.

2.4.2: SSI Encoder Wiring Example

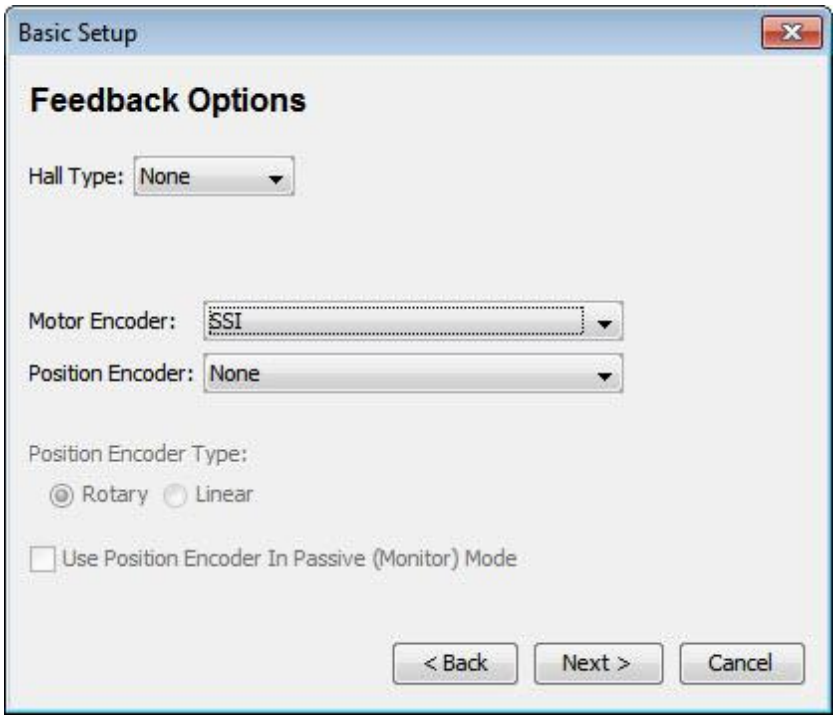
Below is a typical SSI encoder wiring diagram. See data sheets for detailed wiring specifications.



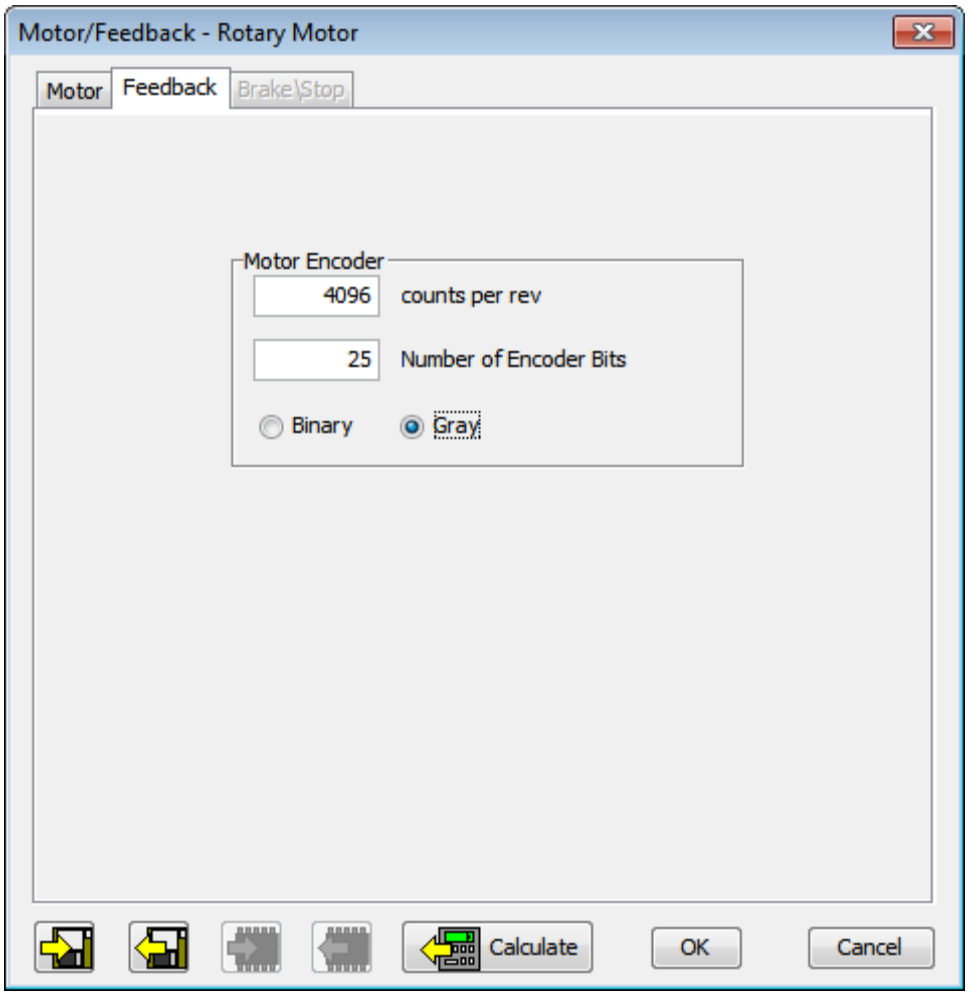
2.4.3: SSI Encoder Configuration Example

1 In CME2, navigate to the Feedback Options screen: **Amplifier→Basic Setup→Change Settings→Next**.

On the *Feedback Options* screen, set **Motor Encoder** to SSI.



2 Open the *Motor/Feedback* screen *Feedback* tab.



View or change the settings described below.

Field	Description
counts per rev	Counts per single revolution of the encoder shaft.
Number of Encoder Bits	Resolution of the encoder output.
Binary	Parallel output.
Gray	Parallel output with single bit change.

2.5: EnDat Encoders (Type 11)

This section describes the use of EnDat absolute encoders with Copley Controls drives.

2.5.1: EnDat Encoder and Drive Communication

EnDat encoders can be single or multi-turn. They provide all digital bidirectional communication to the Copley drive.

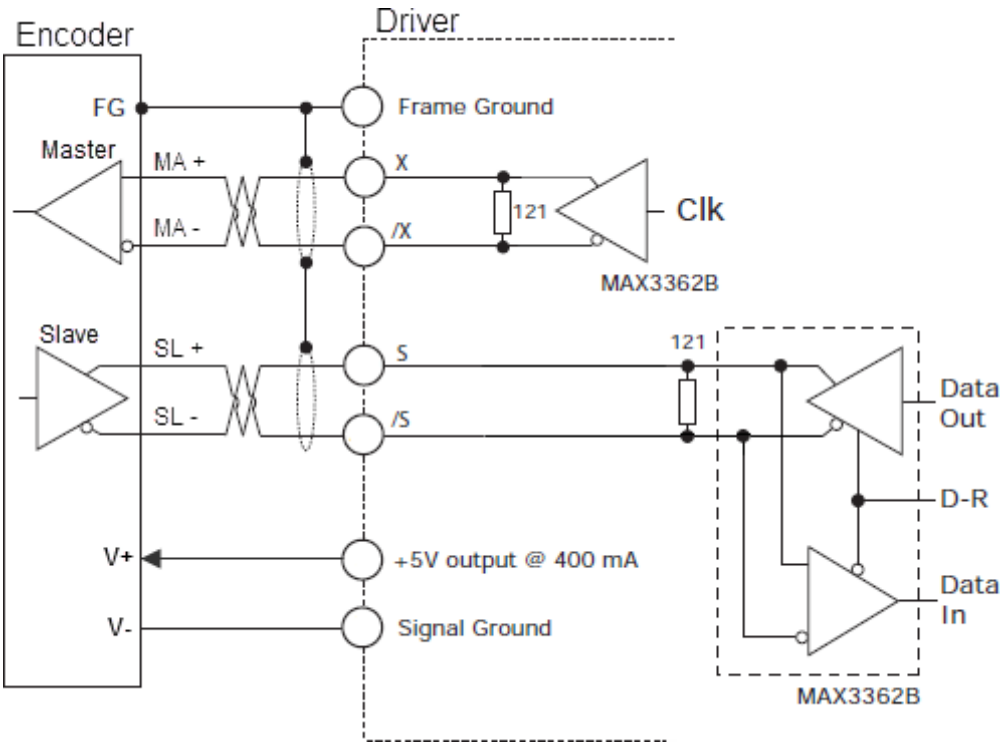
The type of transmission (position values, parameters, diagnostics, etc.) is selected by mode commands that the drive sends to the encoder. The drive can read and update information stored in the encoder, or save new information. The encoder's zero reference point may be set by the drive. The clock and data are wired as differential pairs.

An Endat encoder transmits data in synch with the drive's clock signal.

Because EnDat 2.1 encoders have slow update rates, the transmission of position values require sin/cos inputs (1V p-p). EnDat 2.2 encoders have fast update rates and do not require sin/cos inputs.

2.5.2: EnDat 2.2 Encoder Wiring Example

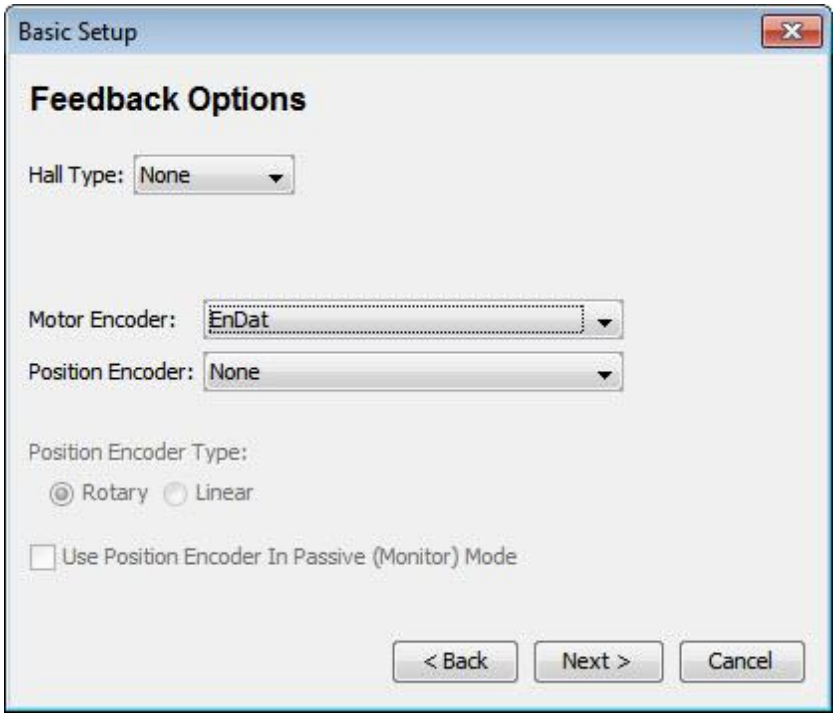
Below is a typical EnDat 2.2 encoder wiring diagram. Use of programmable sin/cos incremental signals is not required. See data sheets for detailed wiring specifications.



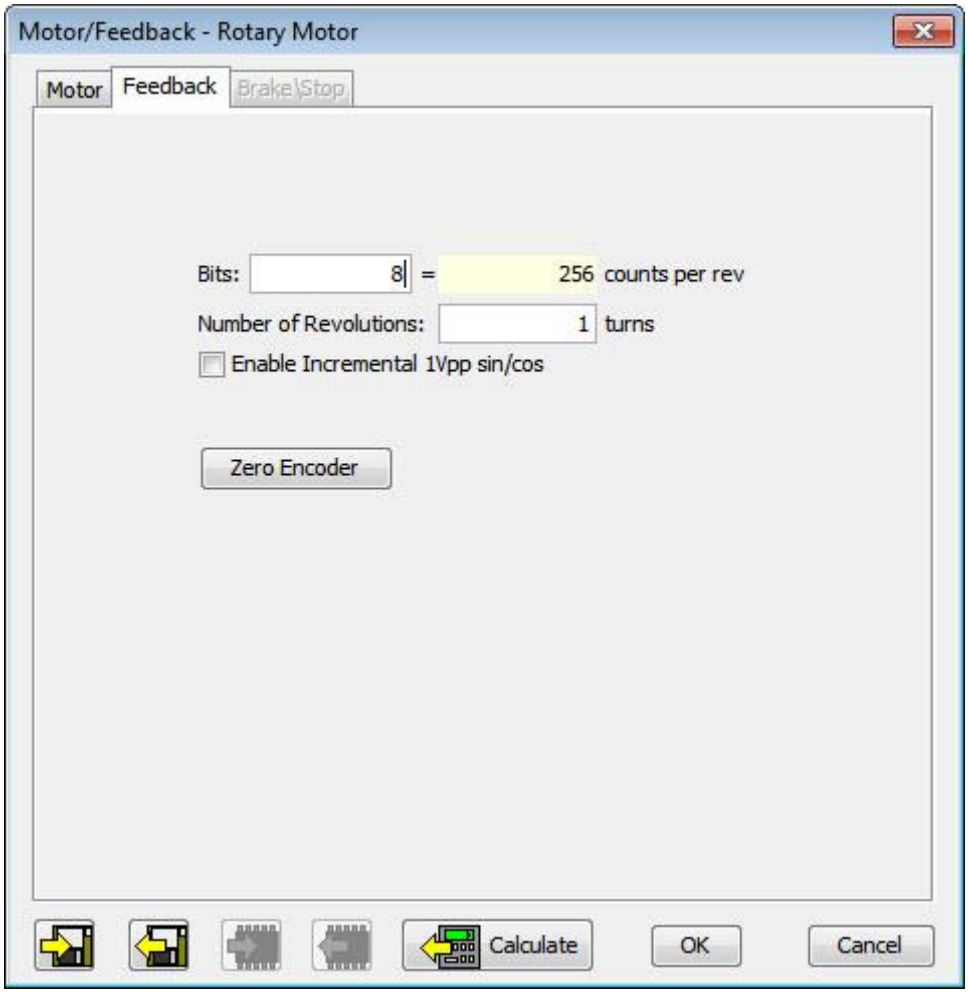
2.5.3: Endat 2.2 Encoder Configuration Example

1 In CME2, navigate to the Feedback Options screen: **Amplifier→Basic Setup→Change Settings→Next**.

On the *Feedback Options* screen, set **Motor Encoder** to EnDat.



2 Open the *Motor/Feedback* screen's *Feedback* tab.

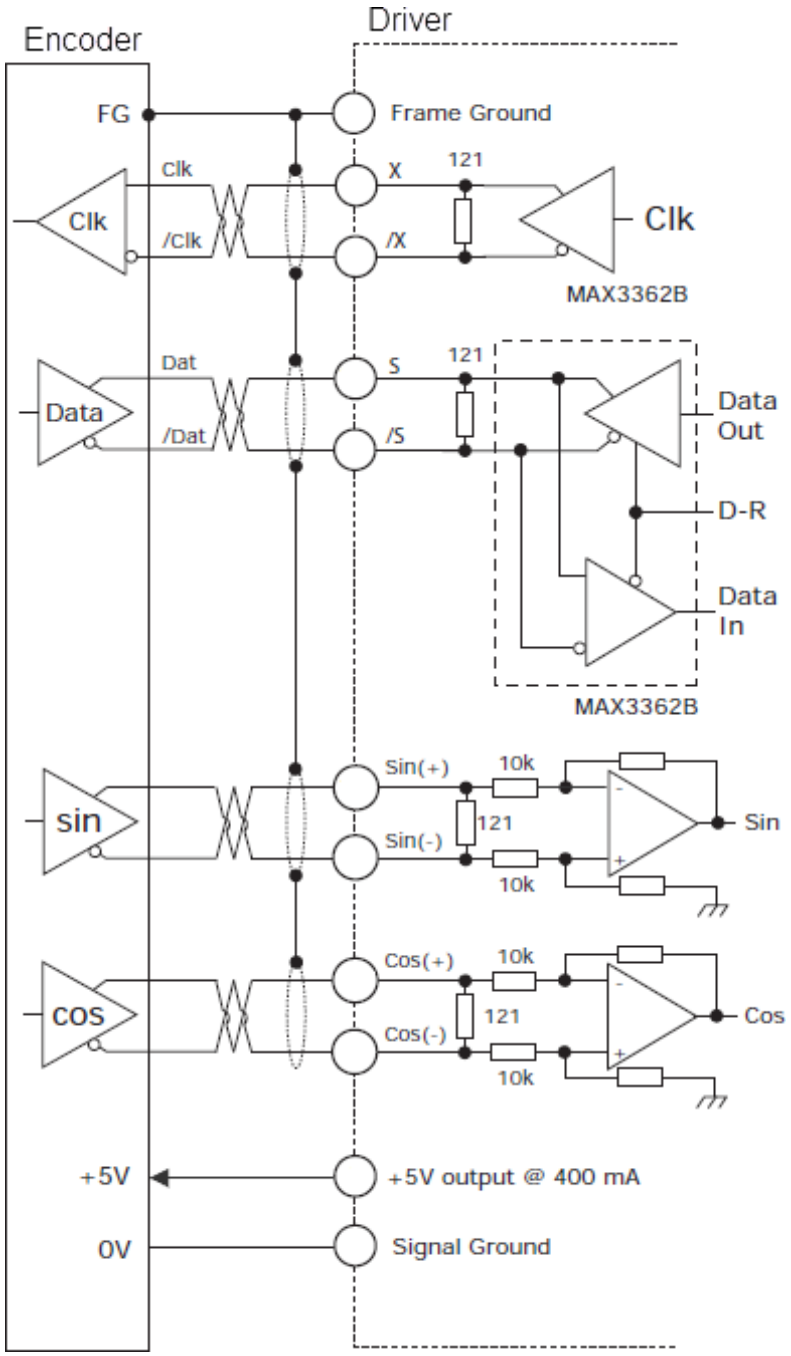


View or change the settings described below.

Field	Description
Bits	Resolution of the encoder output.
Number of revolutions	The number of turns on a multi-turn encoder.
Enable Incremental 1Vpp sin/cos	Not available or needed with EnDat 2.2.
Zero Encoder	Parallel output with single bit change.

2.5.4: EnDat 2.1 Wiring Example

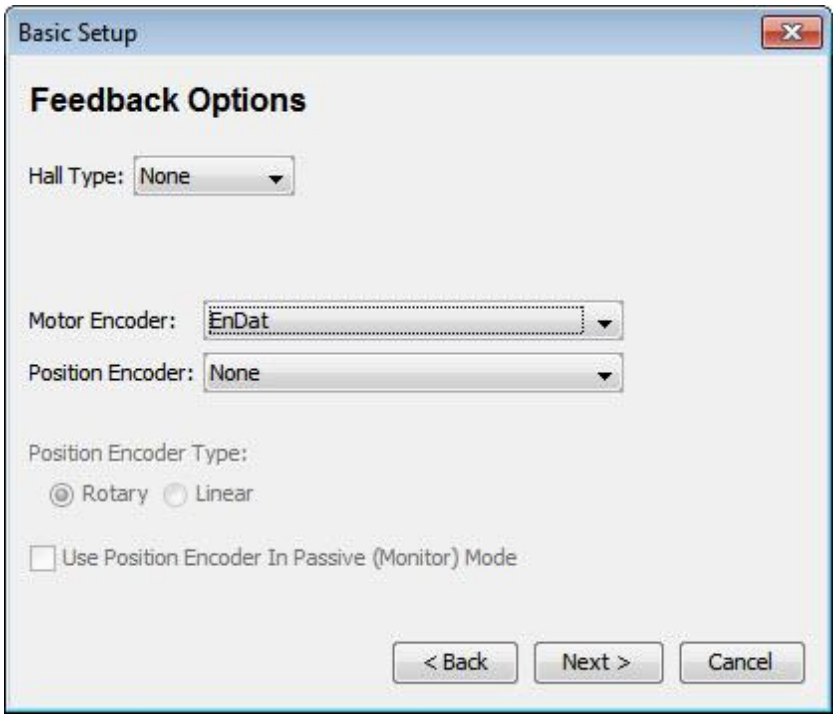
Below is a typical EnDat 2.1 encoder wiring diagram. See data sheets for detailed wiring specifications.



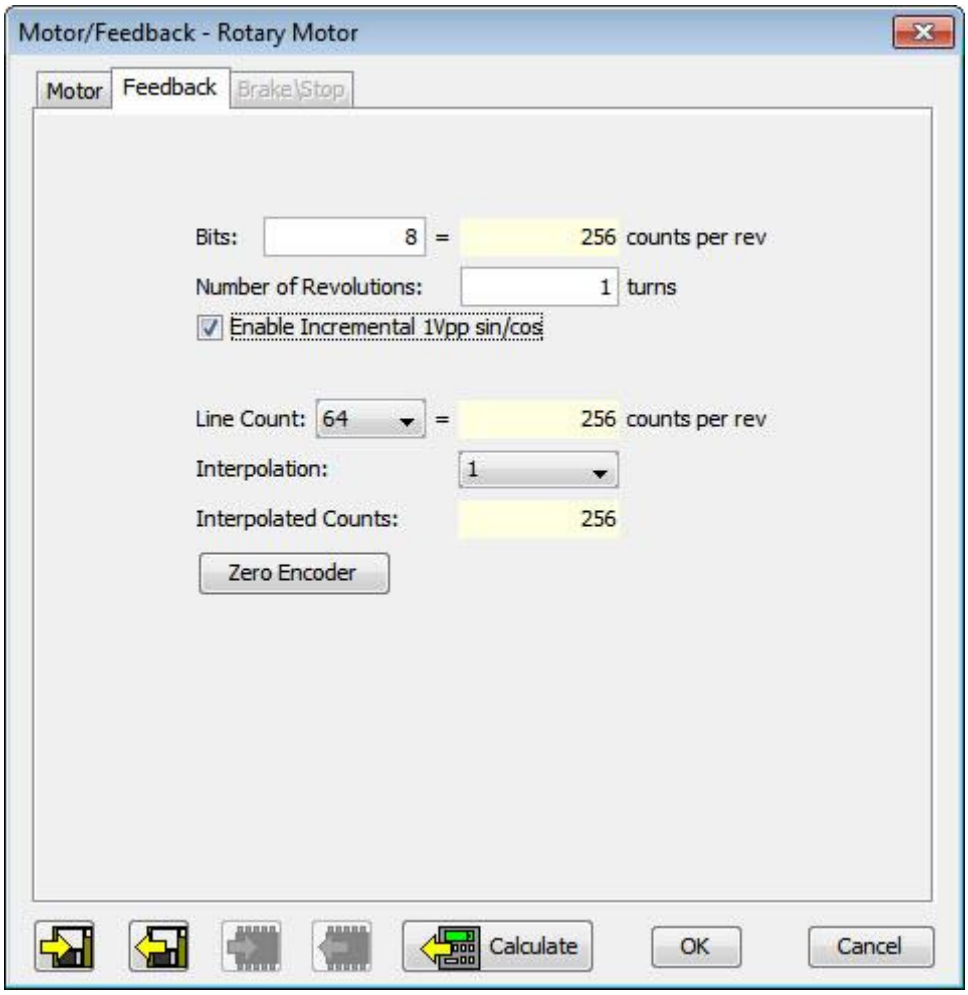
2.5.5: EnDat 2.1 Encoder Configuration Example

1 In CME2, navigate to the *Feedback Options* screen: **Amplifier→Basic Setup→Change Settings→Next**.

On the *Feedback Options* screen, set **Motor Encoder** to EnDat.



2 Open the *Motor/Feedback* screen's *Feedback* tab. Note that *Enable Incremental 1Vpp sin/cos* has been checked.



View or change the settings described below.

Field	Description
Bits	Resolution of the encoder output.
Number of Revolutions	The number of turns on a multi-turn encoder.
Enable Incremental 1Vpp sin/cos	Required for EnDat 2.1 to close the position loop. Lines interpolated.
Line Count	Fundamental lines that are photo etched on an encoder disk for quadrature count and interpolation.
Interpolation	The number of counts/post quadrature fundamental lines.
Zero Encoder	Automatically zeros the encoder.

2.6: Absolute A Encoders (Type 14)

This section describes the use of Absolute A encoders in Copley Controls drives.

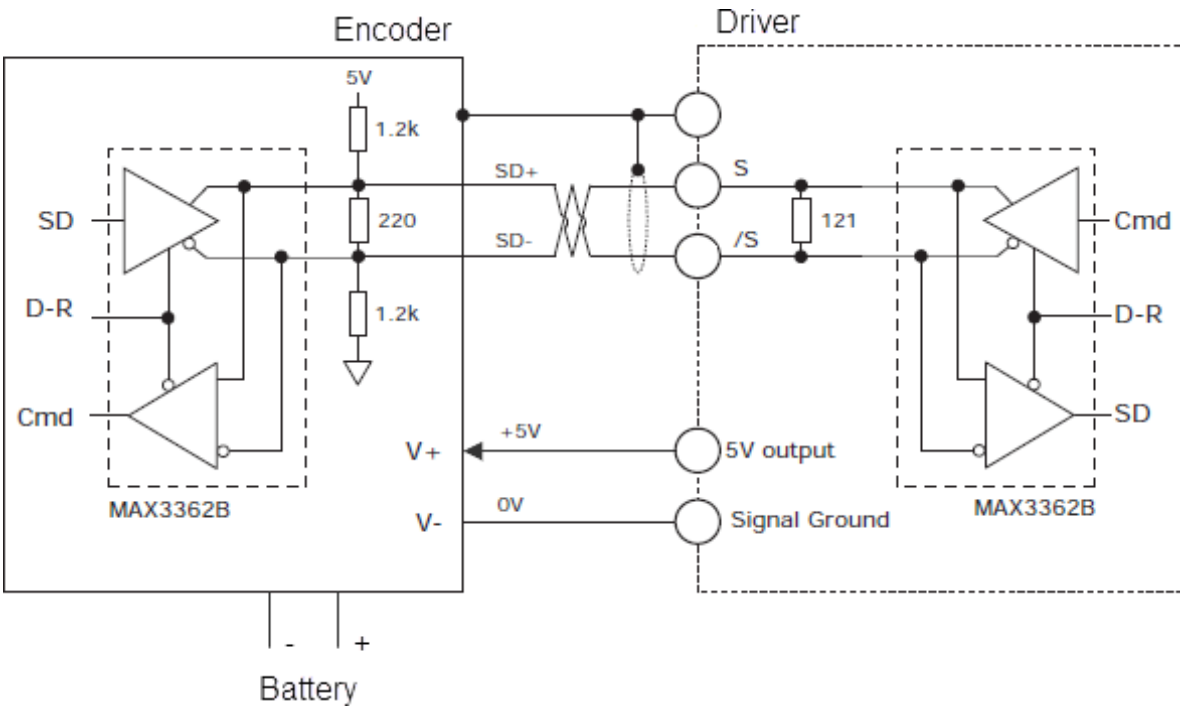
2.6.1: Absolute A Encoder and Drive Communication

Absolute A encoders can be single or multi-turn. They provide all digital one way communications to the selected Copley drive with a bi-directional option.

The Absolute A interface uses 2-wire, half-duplex communication to handle position values and diagnostics.

2.6.2: Absolute A Encoder Wiring Example

Below is a typical Absolute A encoder wiring diagram. Absolute A encoders typically use batteries to save multi-turn position data in case of power failure. See data sheets for detailed wiring specifications.

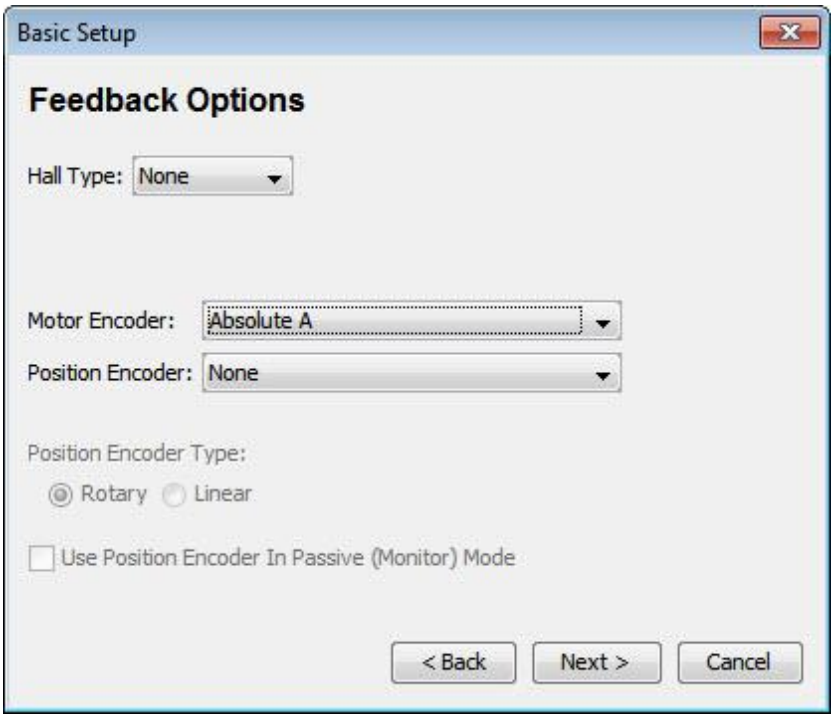


2.6.3: Absolute A Configuration Example

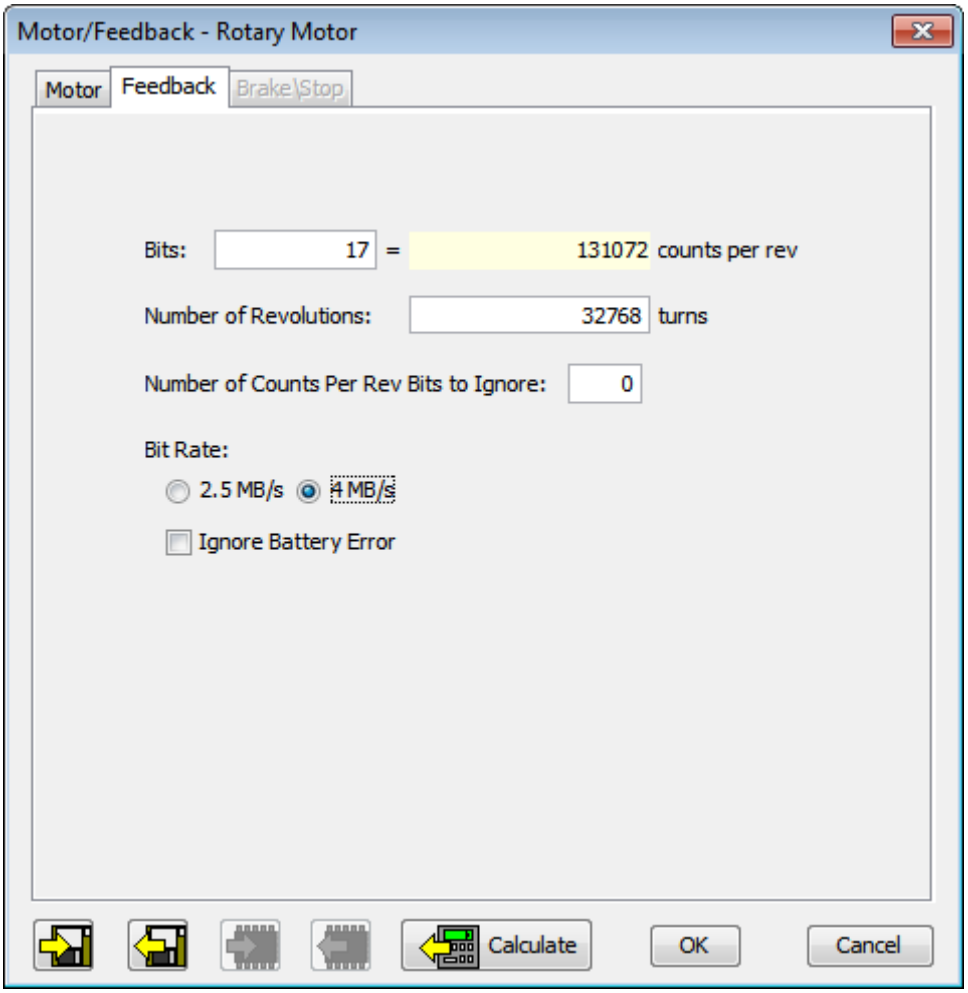
In the example below, Tamagawa, Panasonic, and Sanyo Denki Absolute A encoders may be substituted for *Absolute A* in the Feedback Options screen's Motor Encoder entry drop down menu.

1 In CME2, navigate to the Feedback Options screen: **Amplifier→Basic Setup→Change Settings→Next**.

On the *Feedback Options* screen, set **Motor Encoder** to Absolute A.



2 Open the Motor/Feedback screen's Feedback tab.



View or change the settings described below.

Field	Description
Bits	Resolution of the encoder output.
Number of Revolutions	The number of turns on a multi-turn encoder.
Number of Counts Per Rev Bits to Ignore	Bits (which contain no useful information) to ignore, after the position data has been sent.
Bit Rate	Used for communication. 2.5MB/s or 4 MB/s

2.7: Incremental A Encoders (Type 14)

This section describes the use of Incremental A encoders with Copley Controls drives.

The incremental A encoder is considered a subset of Absolute encoders because it provides absolute Hall information for commutation.

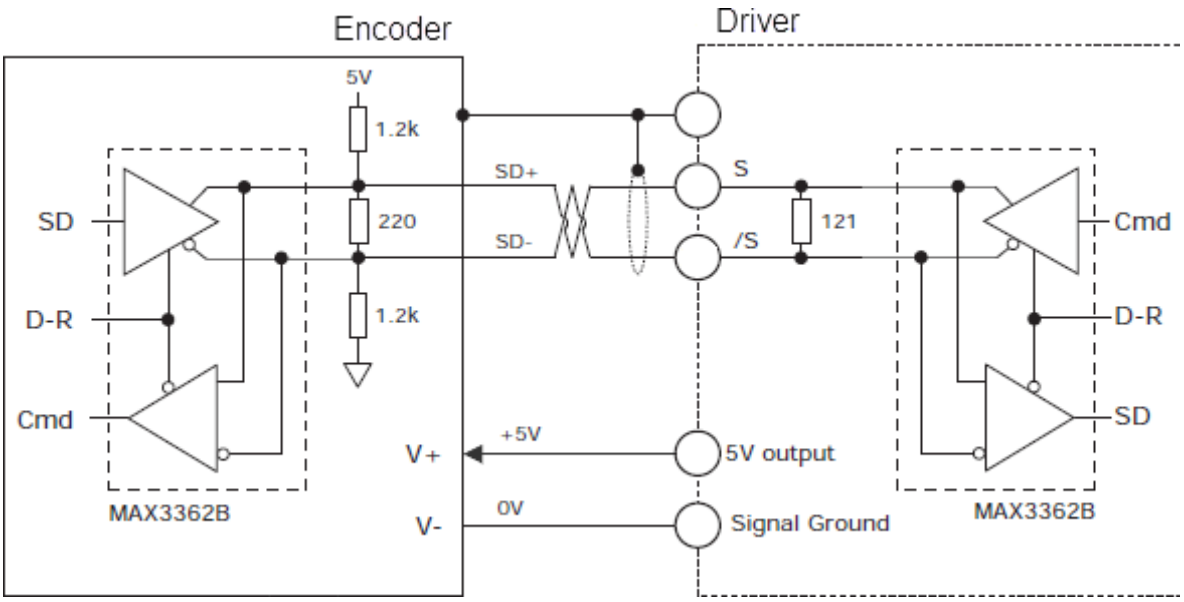
2.7.1: Incremental A Encoder and Drive Communication

Incremental A encoders provide incremental position and Hall data to the selected Copley drive.

The incremental A interface uses 2-wire, half-duplex communication to handle position values and diagnostics. The data is wired as a differential pair.

2.7.2: Incremental A Encoder Wiring Example

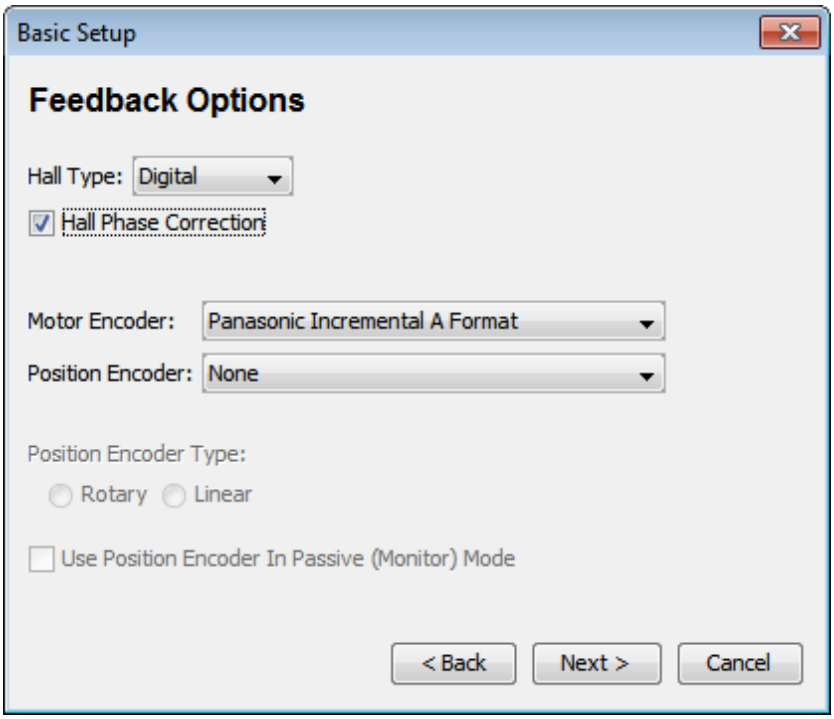
Below is a typical Incremental A encoder wiring diagram. Because there is no absolute position, a battery is not required. See data sheets for detailed wiring specifications.



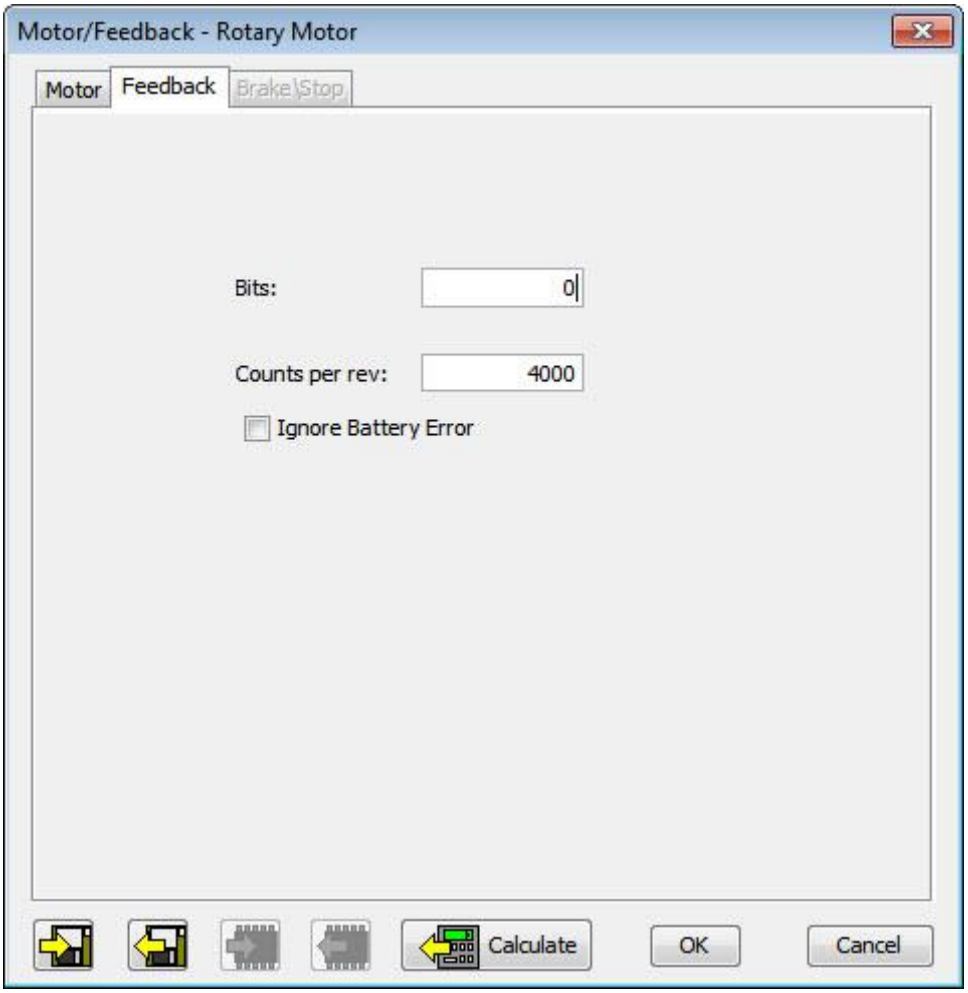
2.7.3: Incremental A Configuration Example

1 In CME2, navigate to the Feedback Options screen: **Amplifier→Basic Setup→Change Settings→Next**.

On the *Feedback Options* screen, set **Motor Encoder** to *Incremental A Format*, for example.



2 Open the *Motor/Feedback* screen's *Feedback* tab.



View or change the settings described below.

Field	Description
Bits	Resolution of the encoder output.
Counts per rev	The number of counts/rev post quadrature.
Ignore Battery Error	Ignores battery error status from the encoder.

2.8: BiSS Encoders (Type 13)

This section describes the use of BiSS encoders in Copley Controls drives.

2.8.1: BiSS Encoder and Drive Communication

BiSS encoders can be single or multi-turn. They provide all digital one way communications to the selected Copley drive with a bi-directional option.

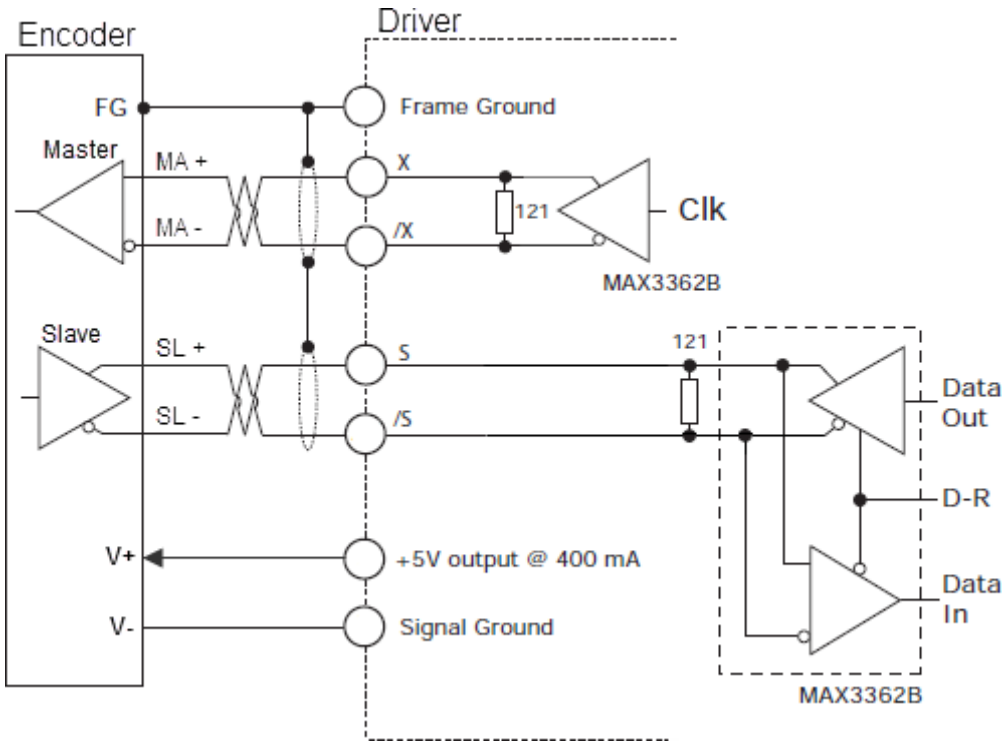
A train of clock pulses from the drive initiates the transmission of position data by subsequent clock pulses. The clock and data are wired as differential pairs.

BiSS is an open source digital interface for sensors and actuators. It refers to principles of well known industrial standards for Serial Synchronous Interfaces like SSI, with additional options:

- BiSS B-protocol: Mode choice at each cycle start
- BiSS C-protocol: Continuous mode
- Synchronous high speed communication
  Line delay compensation for high speed data transfer
- Request for data generation at slaves
- Safety capable: CRC, Errors, Warnings

2.8.2: BiSS Encoder Wiring Example

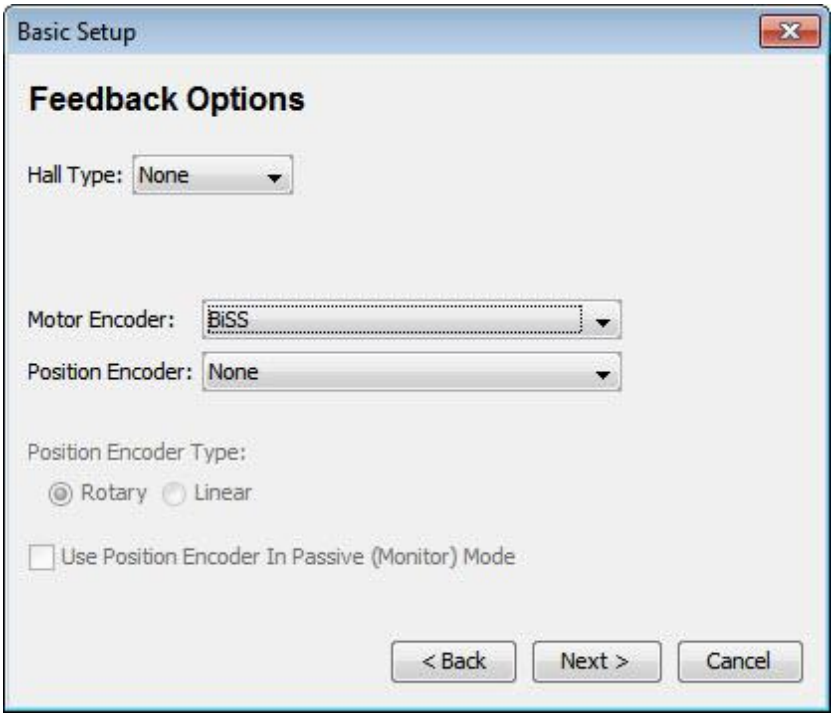
Below is a typical BiSS encoder wiring diagram. See data sheets for detailed wiring specifications.



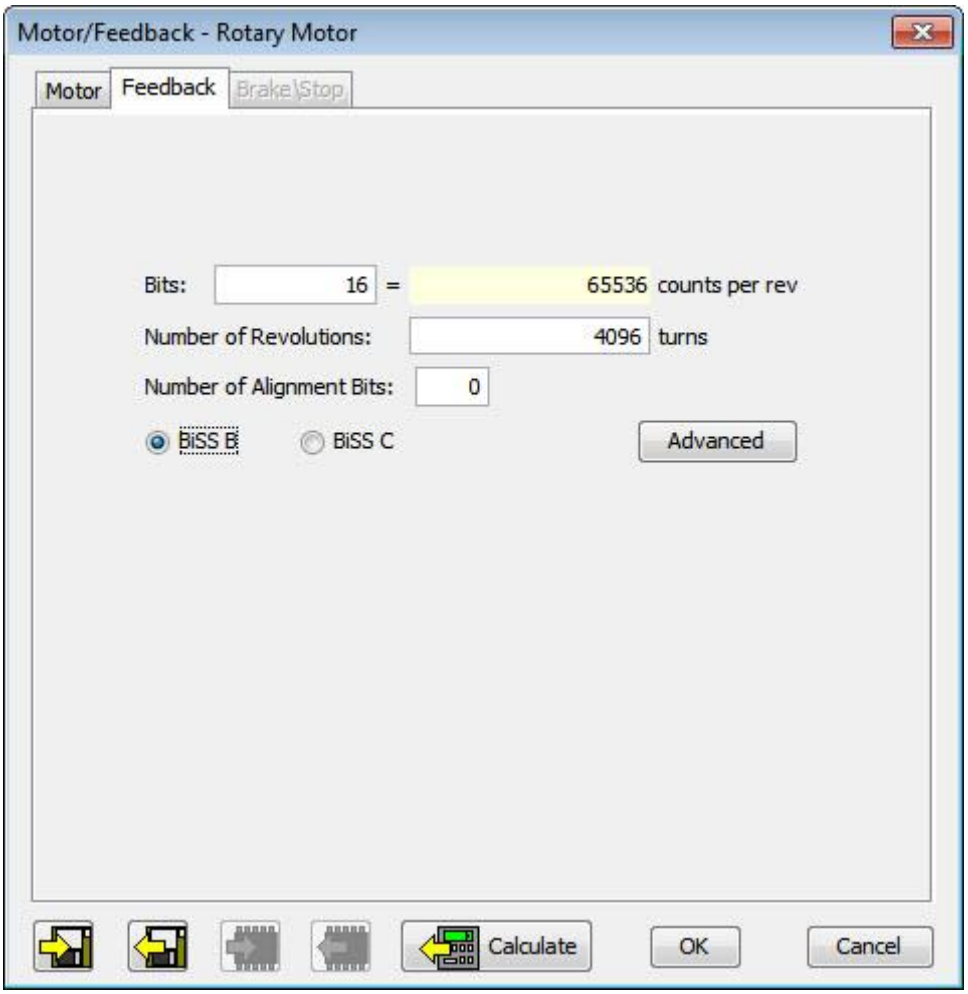
2.8.3: BiSS Encoder Configuration Example

1 In CME2, navigate to the Feedback Options screen: **Amplifier→Basic Setup→Change Settings→Next**.

On the *Feedback Options* screen, set **Motor Encoder** to BiSS.



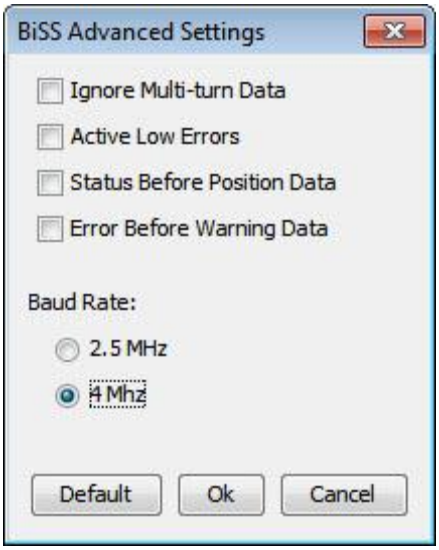
2 Open the *Motor/Feedback* screen's *Feedback* tab.



View or change the settings described below.

Field	Description
Bits	Resolution of the encoder output.
Number of Revolutions	The number of turns on a multi-turn encoder.
Number of alignment Bits	Bits (which contain no useful information), that may be ignored, after position data has been sent.
BiSS B	Mode choice at each start cycle.
BiSS C	Continuous mode.

3 On the *Feedback* screen click the **Advanced** button to open the *BiSS Advanced Settings* window.



View or change the settings described below.

Field	Description
Ignore Multi-turn Data	Use only single turn data.
Active Low Errors	Active low option. Active high is the default.
Status Before Position Data	Status before position data option. Position data before status is the default.
Error Before Warning Data	Error before warning option. Warning before error is the default.
Baud Rate	A baud rate option of 2.5 MHz . 4 MHz is the default.

CHAPTER 3: ENCODER MANAGEMENT

This chapter discusses the management of encoder errors, ASCII programming using CME 2, clearing faults, and absolute position.

CME 2 uses ASCII variables to read status information and to configure absolute encoders. The mapping of bits to function depends on the encoder type. For details pertaining to specific encoder types see Encoder Status Words, p. 40 and Encoder Option Words, p. 42.

For a full list of parameters see the Copley Amplifier Parameter Dictionary. For detailed ASCII programming (which includes error codes), see the Copley ASCII Interface Programmer's Guide. Both guides can be found at: http://www.copleycontrols.com/Motion/Downloads/protocols.html.

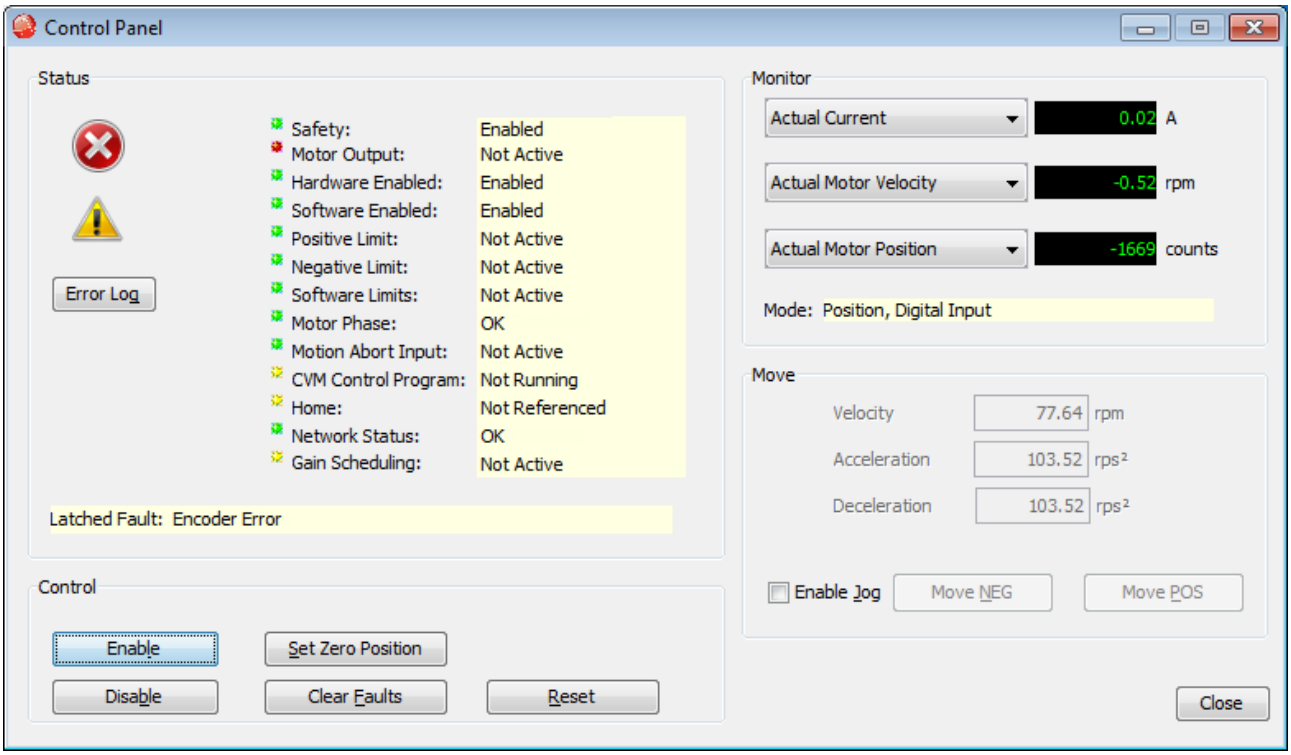
3.1: CME 2 Encoder Errors

CME 2 displays fault information via the Control Panel and Error Log windows.

3.1.1: Control Panel

The CME2 Control Panel will indicate if there is an encoder error. Any red lights in the Status section indicate that motion will be inhibited. After problems have been fixed, fault indicators can be cleared by clicking the **Clear Faults** or the **Reset** button.

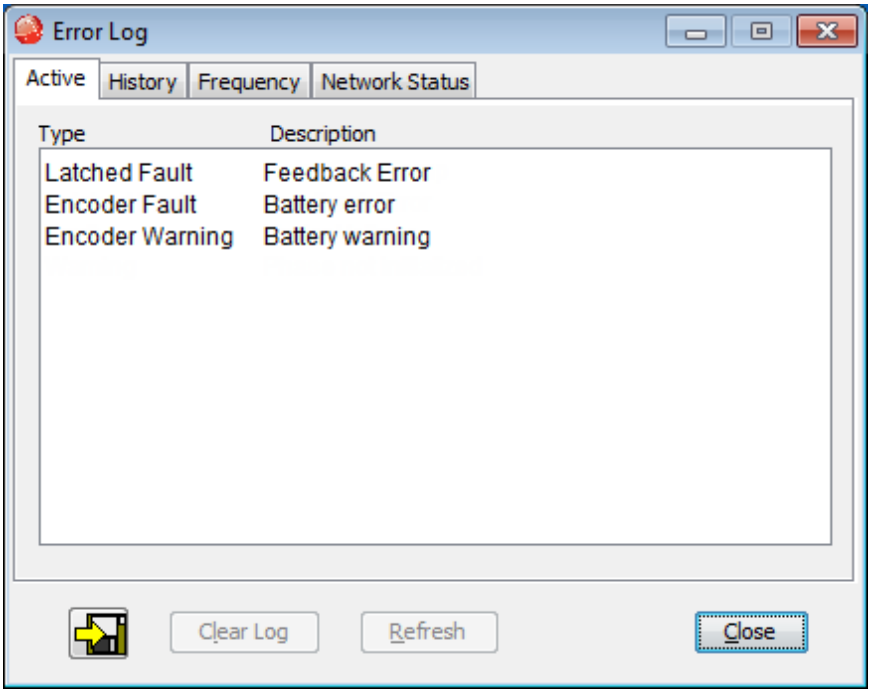
To open the *Control Panel*, click on the icon  from the Toolbar, or, from the Main Menu choose **Amplifier→Control Panel**.



3.1.2: Error Log

The Error Log window will display the type of encoder error and its description. From here, troubleshooting may begin. The window below shows examples of errors.

To open the *Error Log*, click on the icon from the Toolbar, or, from the Main Menu, choose **Amplifier→Error Log**.



Troubleshooting Table

The table below shows examples of errors described in the Error Log window and their likely solutions.

Description	Problem	Solution
Feedback Error	This could indicate an open wire.	Fix and verify good connections.
CRC Error	CRC (Cyclical Redundancy Check). Check on feedback data. This is an indication of poor cabling, grounding or shielding.	Check cabling system lengths, shield path to Earth, and drive grounding.
Battery Warning	Battery is not in the circuit or dead, but encoder still has power.	Replace battery.
Battery Error	Battery was not in the circuit or dead while the power was off.	Replace battery and manage possible lost multi-turn position.

After faults have been managed, clear them using the Control Panel **Clear Faults** button.

3.2: Encoder Status Words

There are two encoder status words available, one for motor encoder status: ASCII **0x12E**, and one for load encoder status: ASCII **0x12F**.

3.2.1: 0x12E Motor Encoder Status

This status word returns status information for the motor encoder. Bits set in the status word are latched and cleared when the status value is read (See Encoder Status Bits below). The format of this status word is dependent on the encoder type. Many error bits are taken directly from encoder data stream. For a full description of what these error bits mean, consult the encoder manufacturer.

The table below includes the equivalent MACRO, CANopen, and EtherCAT (use the CAN ID) status words.

From the Copley Amplifier Parameter Dictionary					
ASCII	**DvcNet**	**Macro**	**CAN ID:sub**	**Bank**	**Type**
0x12E	0x12F	0x52E	0x2224	R*	U32

* Read only

3.2.2: 0x12F Load Encoder Status

This status word is the same as 0x12E, but for the load encoder.

The table below includes the equivalent MACRO, CANopen, and EtherCAT (use the CAN ID) status words.

From the Copley Amplifier Parameter Dictionary					
ASCII	**DvcNet**	**Macro**	**CAN ID:sub**	**Bank**	**Type**
0x12F	0x12G	0x52F	0x2225	R*	INT32

* Read only

For more details see the Copley Amplifier Parameter Dictionary: (http://www.copleycontrols.com/Motion/Downloads/protocols.html).

3.2.3: Encoder Status Bits

Below are status bit tables for commonly used encoder types.

SSI Encoders (Type 12)

Bits	Description
0 - 6	Fault flags returned from the encoder.
15	Encoder data invalid bit set.

EnDat Encoders (Type 11)

Bits	Description
0	CRC error on data received from encoder.
1	Failed to detect encoder connected to amplifier.
2	Error bit on encoder stream is active.
3	Encoder failed to respond to request for position.

Absolute A Encoders (Type 14)

Bits	Description
0	Battery warning reported by encoder.
1	Battery error reported by encoder.
3	Over speed reported by encoder.
4	Memory error reported by encoder.
5	STERR reported by encoder.
6	PSERR reported by encoder.
7	Busy error reported by encoder.
8	Memory busy reported by encoder.
9	Over temperature reported by encoder.
15	CRC error on data received from encoder.

Incremental A Encoders (Type 14)

Bits	Description
0	Battery warning reported by encoder.
1	Battery error reported by encoder.
3	Over speed reported by encoder.
4	Memory error reported by encoder.
5	STERR reported by encoder.
6	PSERR reported by encoder.
7	Busy error reported by encoder.
8	Memory busy reported by encoder.
9	Over temperature reported by encoder.
15	CRC error on data received from encoder.

BiSS Encoders (Type 13)

Bits	Description
0	CRC error on data received from encoder.
1	Encoder failed to transmit data to amp.
2	Error bit on encoder stream is active.
3	Warning bit on encoder stream is active.
4	Encoder transmission delay is too long.

3.3: Encoder Option Words

Encoder option words are used to configure motor or load encoders. There are two encoder options words available, one for motor encoders: ASCII **0x12A**, and one for load or position encoders: ASCII **0x12B**. The mapping of option bits to function depends on the encoder type, see Encoder Option Bits below.

3.3.1: 0x12A Motor Encoder Options

0x12A specifies various configuration options for the motor encoder.

The table below includes the equivalent MACRO, CANopen, and EtherCAT (use the CAN ID) status words.

From the Copley Amplifier Parameter Dictionary					
ASCII	**DvcNet**	**Macro**	**CAN ID:sub**	**Bank**	**Type**
0x12A	0x12B	0x52A	0x2222	F	INT32

3.3.2: 0x12B Load Encoder Options

This word is the same as 0x12A, but for the load or position encoder. The table below includes the equivalent MACRO, CANopen, and EtherCAT (use the CAN ID) status words.

From the Copley Amplifier Parameter Dictionary					
ASCII	**DvcNet**	**Macro**	**CAN ID:sub**	**Bank**	**Type**
0x12B	0x12C	0x52B	0x2223	F	INT32

For more details see the Copley Amplifier Parameter Dictionary (http://www.copleycontrols.com/Motion/Downloads/protocols.html).

3.3.3: Encoder Option Bits

Below are option bit tables for configuring commonly used encoder types.

SSI Encoders (Type 12)

Bits	Description
0-5	Number of bits of position data available.
8-10	Number of extra status bits sent after position data.
12	If set, ignore the first bit of data sent by the encoder.
13	If set, encoder outputs position data using Gray code.
14	If set, pull clock low briefly after data (custom for Codechamp encoder).
15	If set, data is sent LSB first.
16-21	Encoder bit rate in 100 kHz units. If zero, default to 1MHz.
22	If set, use setting of encoder counts/rev to determine how many data bits to use.
24	If set, first bit is 'data valid' bit.

EnDat Encoders (Type 11)

Bits	Description
0-4	Number of bits of single turn data available from encoder.
8-12	Number of bits of multi-turn data available from encoder.
16	Set if analog inputs are supplied by encoder.
17	If set, use multi-mode port.
18	If set, read position using EnDat 2.2 style commands rather than the default 2.1 style.
20-23	Number of least significant bits of the encoder reading to discard.

Absolute A Encoders (Type 14)

Bits	Description
0-5	Number of bits of single turn data.
8-12	Number of bits of multi-turn data.
16-19	Number of LSB to discard from reading.
20-22	Number of consecutive CRC errors to ignore before generating an error.
24-27	Encoder sub-type (0=Tamagawa, 1=Panasonic absolute, 2=HD systems, 3=Panasonic Incremental, 4=Sanyo Denki).
28	Bit rate (set for 4 Mbit, clear for 2.5 Mbit).
30	If set, treat encoder battery errors as warnings.

Incremental A Encoders (Type 14)

Bits	Description
0-5	Number of bits of single turn data.
8-12	Number of bits of multi-turn data.
16-19	Number of LSB to discard from reading.
20-22	Number of consecutive CRC errors to ignore before generating an error.
24-27	Encoder sub-type (0=Tamagawa, 1=Panasonic absolute, 2=HD systems, 3=Panasonic Incremental, 4=Sanyo Denki).
28	Bit rate (set for 4 Mbit, clear for 2.5 Mbit).
30	If set, treat encoder battery errors as warnings.

BiSS Encoders (Type 13)

Bits	Description
0-5	Number of bits of single turn data.
8-12	Number of bits of multi-turn data.
15	If set, assume the encoder position data wraps after the number of encoder counts programmed in parameter 0x62.
16	Set for mode-C encoder format.
20	Set if encoder error and warning bits are active low.
21	Set if encoder status bits are sent before position data, clear if status bits are sent after position data.
22	Set if encoder error bit is transmitted before warning bit. Clear for warning bit sent first.
24-26	Number of alignment bits (reserved bits sent before position info).
28	Use multi-mode port if set. If clear use primary encoder interface.
30	Set for 2.5MHz baud rate, clear for 4MHz baud rate.

3.4: Clearing Faults

In the Encoder: Before an encoder's latched *faults* can be cleared, any faulty mechanical or electrical conditions in the encoder must be corrected.

In the Drive: Before a drive's latched *encoder faults* can be cleared, any faulty mechanical or electrical conditions in the encoder, drive or mechanical system must be corrected.

3.4.1: Clearing Faults in an Absolute Encoder

After faulty conditions have been corrected, any internal latched faults in an encoder must be cleared to resume normal encoder use.

Latched faults in an encoder may be cleared through Copley drives or, with some smart encoders, through the encoders themselves.

The following interfaces can be used to clear latched faults in an encoder:

- ASCII Interface (p. 45)
- Binary Serial Interface (p. 46)
- CANopen and EtherCAT (p. 47)
- MACRO (p. 47)

Note: power cycling an encoder may not clear an *internal* encoder fault condition.

ASCII Interface

Copley drives support a command on CME 2's ASCII interface that can be used to clear encoder faults.

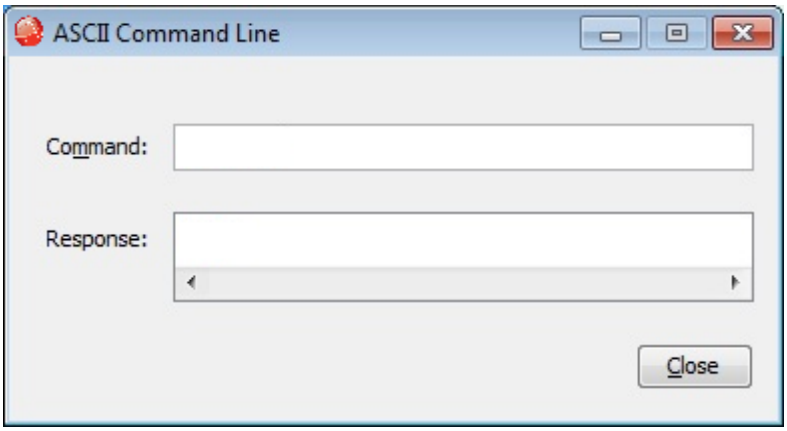
To clear a fault on a motor encoder, use the CME ASCII command **enc clear.**

To clear a fault on a load encoder, use the CME 2 ASCII command **ldenc clear.**

Upon success, these commands will return '**ok**'. If an error occurs or the drive's firmware does not support clearing latched faults for the specific type of encoder, an error response will be returned.

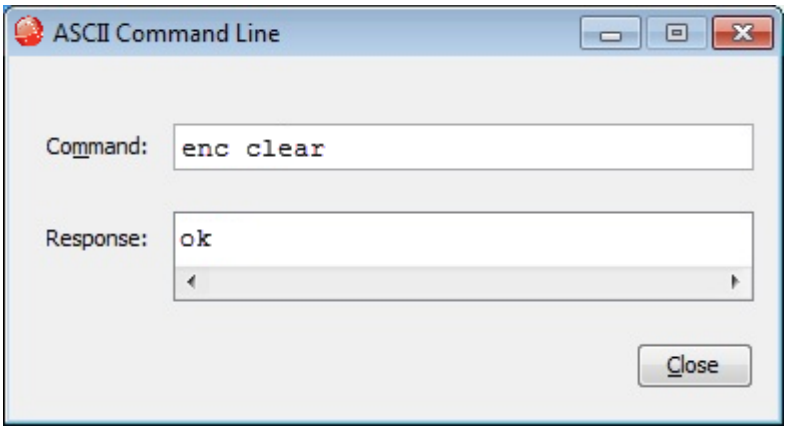
To enter a command using CME 2's ASCII interface:

1. From the Main screen, choose **Tools→ASCII Command Line** to open the tool.



2. Enter the ASCII Command **enc clear** or **ldenc clear** in the *Command* field.

3. Press the **Enter** key to send the command to the drive. Observe the *Response* field. The response should read *ok*.



Binary Serial Interface

The binary serial interface of a Copley drive supports clearing latched encoder faults through op-code 0x1B. This op-code uses a single word of data to identify the type of command to send to the encoder. The data word is formatted as follows:

Bits	Description
0	Identify which encoder to clear; 0 for motor encoder, 1 for load encoder.
1-3	Reserved.
4-7	Sub-command code (Should be 1 to clear encoder errors).
8-11	Reserved.
12	Axis number; 0 for axis 1, 1 for axis 2.
13-15	Reserved.

The following encoder sub-commands are currently supported:

Code	Description
0	Read a register from within the encoder. The register number is passed as a second word of data to the encoder command.
1	Reset any errors currently latched on the encoder
2	Zero the encoder's internal position.

For example:

To clear latched faults on the motor encoder of axis 1, the data word would be **0x0010**.

To clear latched faults on the load encoder of axis 2, the data word would be **0x1011**.

The full binary command consists of a 4 byte header followed by the two byte data word. To clear latched faults on the motor encoder of axis 1, the full command sent to the drive would be:

0x00 0x50 0x01 0x1B 0x00 0x10

For more details on the Binary Serial Protocol, please contact Copley Controls.

CANopen and EtherCAT

To clear latched encoder faults over an EtherCAT or CANopen interface, object 0x2000 is used.

This object allows binary serial commands to be transmitted over the CANopen/EtherCAT interface. Serial port commands can be sent using object 0x2000 by first writing the command to the object using an SDO. The data written to 0x2000 consists of the op-code code 0x1B (for encoder commands), followed by any data words. Note that data words are sent LSB first when communicating over the CANopen/EtherCAT interface, rather then MSB first when communicating over the normal binary serial interface.

For example:

> To clear latched encoder faults on the motor encoder of axis 1, use op-code **0x1B** and a single data word **0x0010**. To write this command to object **0x2000**, pass the following three bytes of data:
>
> **0x1B 0x10 0x00**

To read back the result of the command, read object 0x2000. It should contain a single byte of data which will be 0x00 for success, or an error code for failure.

MACRO

To clear latched encoder faults over the MACRO interface, the MACRO node I-variable 1018 should be used. This I-variable allows binary serial commands to be sent over the MACRO interface.

The 48-bit value written to 1018 consists of an operation code in the upper 16-bits, and data in the lower 32-bits. A more detailed description of this I-variable is available in Copley's MACRO interface documentation.

As described above, the serial port command code used to access absolute encoders is 0x1B. To clear latched encoder faults, a single additional word of data needs to be passed. To clear faults on the motor encoder of axis 1, the passed data word is 0x0010.

Sending this command over the MACRO interface involves two writes to I-1018:

1. Save the data word to the internal buffer associated with 1018:

   MS0,i1018 = 0x0002**0010**0000

2. Execute serial port command code 0x1B:

   MS0,i1018 = 0x000700**1B**0000

3.4.2: Clearing Faults in the Drive

This section describes five ways of clearing faults in the drive:

- CME2 Control Panel
- CANopen or EtherCAT
- Power Cycling
- ASCII Command using an External Application
- ASCII clear command via CME 2

After all faulty conditions have been corrected, the following actions may be taken.

CME2 Control Panel

Open the CME 2 Control Panel and click on the **Clear Faults** or **Reset** button. See the CME 2 User Guide for more details.

CANopen or EtherCAT

To clear a driver's *Encoder Faults* over a CANopen or EtherCAT network, use control word 0x6040, bit 7.

See Copley's CANopen Programmer's Manual for more information.

Power Cycling

Power cycle the drive to clear a driver's encoder faults.

See Related Documentation, p. 5, for manuals mentioned in this section.

ASCII Command using an External Application

A drive's RS-232 serial bus can be used by an external control application (HMI, PLC, PC, etc.) for setup and direct serial control of the drive. The control application can issue drive commands from the set of ASCII format commands that make up the Copley Controls ASCII Interface.

For example:

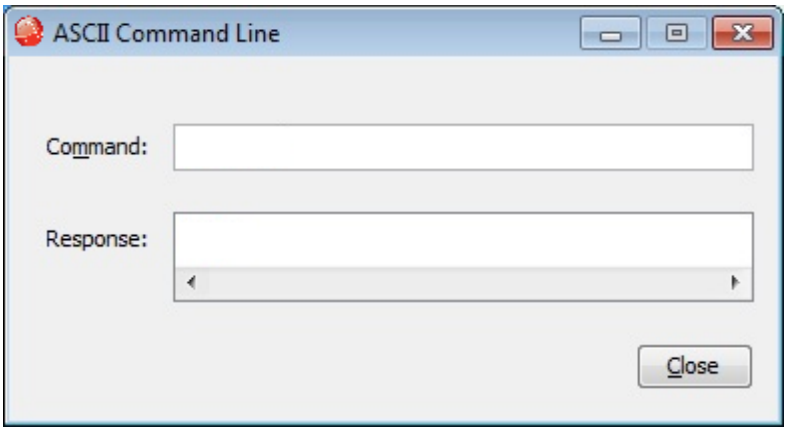
> To clear a driver's encoder faults using an external control application, write a **1** to bit **7** of the fault register **0xa4**.

See Copley's ASCII Programmers Guide for more information.

CME 2 may also be used to send an ASCII command to clear latched *Encoder Faults* on the drive as in the following procedure.

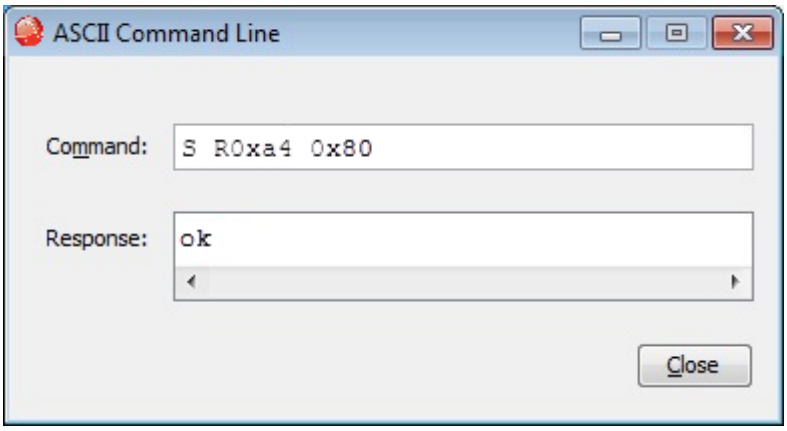
ASCII clear command via CME 2

1 From the Main screen, choose **Tools→ASCII Command Line** to open the tool.



2 Type **S R0xa4 0x80** in the *Command* field.

3 Press the **Enter** key to send the command to the amplifier. Observe the *Response* field. The response should read *ok*.



3.5: Managing Absolute Position

3.5.1: Lost Battery

Do not ignore lost battery faults and warnings if using multi-turn absolute encoders. If a battery is removed and the system is powered off for an extended length of time, the absolute position may need to be re-calibrated.

Some multi-turn encoders use mechanical linkage and do not require a battery to hold absolute multi-turn information.

Single-turn encoders do not usually require a battery.

3.5.2: Mechanical Slipping

Absolute position may be lost due to mechanical slipping between the absolute encoder and the mechanical system to the load. Always verify that there is reliable mechanical linkage with no slipping in the mechanical system.

WARNING

If multi-turn absolute position is lost, damage to equipment or life may be at risk.

3.5.3: Calibration

All absolute encoders need to be calibrated to zero or home. This section describes three choices: Manual Calibration, the Homing Method, and Writing to an Encoder (using CME 2 or through CANopen/EtherCAT).

Manual Calibration

This procedure begins with the encoder not mounted to the mechanical system.

1. Rotate the encoder shaft until the zero position is displayed. CME 2 may also be used with the drive to move the encoder to the zero position.
2. Move the mechanical system to a zero reference.
3. Keeping the encoder in the zero position, mount the encoder to the mechanical system.

Homing Method

This procedure begins with the encoder mounted to the mechanical system. The absolute encoder will power up with an arbitrary position value, and the drive will add an offset to reference the encoder to zero.

1. Rotate the motor to a good Home position.
2. In CME 2, run the Homing Method *Absolute Position Home*. For information on running the Homing Method, see the CME 2 User Guide.

Writing to an Encoder

Some encoders allow the zero calibration position to be written directly to the encoder. In this case the following applies.

Lost Commutation

Writing zero to the encoder will cause loss of motor manufacturer's standard calibration of phase relationship to position and cause lost commutation. Motor manufactures phase calibration will need to be performed again.

WARNING **Failure to heed this warning can cause equipment damage.**

Using CME 2

From the CME 2 main screen, choose **Tools→ASCII Command Line**. In the command field enter ***enc zero***.

OR

From CME 2 navigate to the Manual Phase screen and press the zero encoder button.

Through CANopen/EtherCAT

To zero an encoder over an EtherCAT or CANopen interface, a command is sent to object 0x2000 (via the binary serial interface) using an SDO. The data written to 0x2000 consists of the op-code 0x1B, followed by any data words. Note that data words are sent least significant word first when communicating over the CANopen/EtherCAT interface, rather then most significant word first when communicating over the normal binary serial interface.

For example:

To zero the motor encoder of axis 1, use op-code 0x1B and a single data word 0x0020. To write this command to object 0x2000, pass the following three bytes of data:

0x1B 0x20 0x00

To read back the result of the command, read object 0x2000. This object should contain a single byte of data which will be 0x00 for success, or an error code for failure.

Op-code 0x1B uses a single word of data to identify the type of command to send to the encoder. The data word is formatted as follows:

Bits	Description
0	Identify which encoder to clear; 0 for motor encoder, 1 for load encoder.
1-3	Reserved.
4-7	Sub-command code; 2, to zero the encoder's internal position.
8-11	Reserved.
12	Axis number; 0 for axis 1, 1 for axis 2.
13-15	Reserved.

Sub-commands are shown on the next page.

The following encoder sub-commands are currently supported:

Code	Description
0	Read a register from within the encoder. The register number is passed as a second word of data to the encoder command.
1	Reset any errors currently latched on the encoder
2	Zero the encoder's internal position.

Copley Controls Serial Encoder Guide

P/N 16-01055

Revision 01

May 2013

Copley Controls
20 Dan Road
Canton, MA 02021 USA