

# **SD4840E2 Networked Stepper Indexer/Driver**

**with Integral 2-Port Ethernet Switch and  
Device Level Ring functionality for EtherNet/IP  
Media Redundancy Protocol for PROFINET**

**EtherNet/IP™**

**PROFI**®  
INDUSTRIAL ETHERNET  
**NET**

 **Modbus**

**User Manual**

**AMCI Motion Control Products**

# GENERAL INFORMATION

## **Important User Information**

The products and application data described in this manual are useful in a wide variety of different applications. Therefore, the user and others responsible for applying these products described herein are responsible for determining the acceptability for each application. While efforts have been made to provide accurate information within this manual, AMCI assumes no responsibility for the application or the completeness of the information contained herein.

UNDER NO CIRCUMSTANCES WILL ADVANCED MICRO CONTROLS, INC. BE RESPONSIBLE OR LIABLE FOR ANY DAMAGES OR LOSSES, INCLUDING INDIRECT OR CONSEQUENTIAL DAMAGES OR LOSSES, ARISING FROM THE USE OF ANY INFORMATION CONTAINED WITHIN THIS MANUAL, OR THE USE OF ANY PRODUCTS OR SERVICES REFERENCED HEREIN.

No patent liability is assumed by AMCI, with respect to use of information, circuits, equipment, or software described in this manual.

The information contained within this manual is subject to change without notice.

This manual is copyright 2018 by Advanced Micro Controls Inc. You may reproduce this manual, in whole or in part, for your personal use, provided that this copyright notice is included. You may distribute copies of this complete manual in electronic format provided that they are unaltered from the version posted by Advanced Micro Controls Inc. on our official website: [www.amci.com](http://www.amci.com). You may incorporate portions of this documents in other literature for your own personal use provided that you include the notice "Portions of this document copyright 2018 by Advanced Micro Controls Inc." You may not alter the contents of this document or charge a fee for reproducing or distributing it.

## **Standard Warranty**

ADVANCED MICRO CONTROLS, INC. warrants that all equipment manufactured by it will be free from defects, under normal use, in materials and workmanship for a period of [18] months. Within this warranty period, AMCI shall, at its option, repair or replace, free of charge, any equipment covered by this warranty which is returned, shipping charges prepaid, within eighteen months from date of invoice, and which upon examination proves to be defective in material or workmanship and not caused by accident, misuse, neglect, alteration, improper installation or improper testing.

The provisions of the "STANDARD WARRANTY" are the sole obligations of AMCI and excludes all other warranties expressed or implied. In no event shall AMCI be liable for incidental or consequential damages or for delay in performance of this warranty.

## **Returns Policy**

All equipment being returned to AMCI for repair or replacement, regardless of warranty status, must have a Return Merchandise Authorization number issued by AMCI. Call (860) 585-1254 with the model number and serial number (if applicable) along with a description of the problem during regular business hours, Monday through Friday, 8AM - 5PM Eastern. An "RMA" number will be issued. Equipment must be shipped to AMCI with transportation charges prepaid. Title and risk of loss or damage remains with the customer until shipment is received by AMCI.

## **24 Hour Technical Support Number**

24 Hour technical support is available on this product. If you have internet access, start at [www.amci.com](http://www.amci.com). Product documentation and FAQ's are available on the site that answer most common questions.

If you require additional technical support, call (860) 583-1254. Your call will be answered by the factory during regular business hours, Monday through Friday, 8AM - 5PM Eastern. During non-business hours an automated system will ask you to enter the telephone number you can be reached at. Please remember to include your area code. The system will page an engineer on call. Please have your product model number and a description of the problem ready before you call.

## **Waste Electrical and Electronic Equipment (WEEE)**



At the end of life, this equipment should be collected separately from any unsorted municipal waste.

# TABLE OF CONTENTS

## **General Information**

Important User Information .....	2
Standard Warranty .....	2
Returns Policy .....	2
24 Hour Technical Support Number .....	2
WEEE Statement .....	2

## **About this Manual**

Audience .....	7
Applicable Units .....	7
Navigating this Manual .....	7
Manual Conventions .....	7
Trademark Notices .....	8
Revision Record .....	8
Revision History .....	8
Where To Go From Here .....	8

## **Reference: Introduction to the SD4840E2**

The SD4840E2 .....	9
General Functionality .....	9
Encoder Functionality .....	10
Network Data Format .....	10
Specifications .....	12
Indexer Functionality .....	13
Stall Detection with SD4840E2 .....	14
Driver Functionality .....	14
Idle Current Reduction .....	14
I/O Connector .....	15
Available Discrete Inputs .....	15
Home Input .....	15
CW Limit Switch or CCW Limit Switch .....	15
Start Indexer Move Input .....	15
Emergency Stop Input .....	15
Stop Jog or Registration Move Input .....	16
Capture Encoder Position Input .....	16
General Purpose Input .....	16
Encoder Feedback Inputs .....	16
Available Discrete Output .....	17
Ethernet Ports .....	17
Motor Connector .....	17

## **Reference: Introduction to the SD4840E2 (continued)**

Front Panel .....	18
Status LED .....	18
Module Status (MS) LED .....	18
Network Status (NS) LED .....	18

## **Reference: Move Profiles**

Definitions .....	19
Units of Measure .....	19
Motor Position .....	19
Home Position .....	19
Count Direction .....	19
Starting Speed .....	19
Target Position .....	20
Relative Coordinates .....	20
Absolute Coordinates .....	20
Definition of Acceleration Types .....	20
What is jerk? .....	20
Constant Acceleration .....	21
S-Curve Accelerations .....	21
Trapezoidal S-Curve Acceleration .....	21
Triangular S-Curve Acceleration .....	22
A Simple Move .....	22
Controlled and Immediate Stops .....	23
Host Control .....	23
Hardware Control .....	23
Basic Move Types .....	24
Relative Move .....	24
Controlled Stops .....	24
Immediate Stops .....	24
Absolute Move .....	25
Controlled Stops .....	25
Immediate Stops .....	25
CW/CCW Jog Move .....	26
Controlled Stops .....	26
Immediate Stops .....	27
CW/CCW Registration Move .....	27
Controlled Stops .....	28
Immediate Stops .....	28
Encoder Registration Moves .....	29
Controlled Stops .....	29
Immediate Stops .....	29

**Reference: Move Profiles**

Assembled Moves .....	29
Blend Move .....	30
Controlled Stops .....	31
Immediate Stops .....	31
Dwell Move .....	31
Controlled Stops .....	32
Immediate Stops .....	32
Assembled Move Programming .....	33
Control Bits – Output Data .....	33
Control Bits – Input Data .....	33
Programming Routine .....	33
Saving an Assembled Move in Flash .....	33
Indexed Moves .....	34
Synchrostep (Virtual Axis) Moves .....	35
Controlling Moves In Progress .....	36
Find Home Moves .....	36
Jog Moves .....	36
Registration Moves .....	36
Absolute, Relative and Encoder Registration Moves .....	36
Assembled Moves .....	36
Electronic Gearing .....	37
Motor Steps/Turn .....	37
ELGearing Multiplier & Divisor ...	37
How It Works .....	37
Controlled Stop Conditions .....	37
Immediate Stop Conditions .....	37
Advanced Ratio Control .....	38
Hybrid Control .....	38
Current Loop Gain .....	39

**Reference: Calculating Move Profiles**

Constant Acceleration Equations .....	41
Variable Definitions .....	41
Total Time Equations .....	43
S-Curve Acceleration Equations .....	44
Triangular S-Curve Acceleration .....	44
Trapezoidal S-Curve Acceleration .....	46
Determining Waveforms by Values .....	48

**Reference: Homing the SD4840E2**

Definition of Home Position .....	51
Position Preset .....	51
CW/CCW Find Home Commands .....	51
Homing Inputs .....	51
Physical Inputs .....	51
Backplane Inputs .....	51
Homing Configurations .....	52
Homing Profiles .....	52
Home Input Only Profile .....	52
Profile with Proximity Input .....	53
Profile with Overtravel Limit .....	54

**Reference: Configuration Mode Data Format**

Modes of Operation .....	55
Configuration Mode .....	55
Command Mode .....	55
Power Up Behavior .....	55
Configuration Mode Data Format .....	55
Command Mode Data Formats .....	55
Output Data Format .....	57
Configuration Word 0 Format .....	57
Configuration Word 1 Format .....	59
Notes on Other Configuration Words .....	60
Input Data Format .....	61
Control Word Format (Word 0) .....	61
Invalid Configurations .....	61

**Reference: Command Mode Data Format**

Data Format .....	63
Command Bits Must Transition .....	64
Output Data Format .....	64
Command Word 0 .....	64
Command Word 1 .....	66
Command Blocks .....	68
Absolute Move .....	68
Relative Move .....	68
Hold Move .....	69
Resume Move .....	69
Immediate Stop .....	70
CW Find Home .....	70

**Reference: Command Mode Data  
Format (continued)**

CCW Find Home .....	71
CW Jog .....	71
CW Registration Move .....	72
Synchrostep (Virtual Axis Follower) Move .....	72
CCW Jog .....	73
CCW Registration Move .....	73
Encoder Follower Move .....	74
Preset Position .....	74
Reset Errors .....	75
Run Assembled Move .....	75
Preset Encoder Position .....	76
Programming Blocks .....	77
First Block .....	77
Segment Block .....	77
Input Data Format .....	78
Format of Position Data Values .....	78
Status Word 0 Format .....	79
Status Word 1 Format .....	80
Notes on Clearing a Driver Fault .....	82
Reset Driver Fault .....	82

**Task 1: Motor and Power Supply  
Sizing**

Sizing Your Motor .....	83
Determining Your Motor Current Setting .....	83
A Note on Microstepping .....	83
Torque and Power Curves .....	84
Power Supply Sizing .....	86
Regeneration Effects .....	86

**Task 2: Installing the SD4840E2**

Safe Handling Guidelines .....	87
Prevent Electrostatic Damage .....	87
Prevent Debris From Entering the Module .....	87
Remove Power Before Servicing ..	87
Mounting .....	87
Dimensions .....	87
Minimum Spacing .....	88
Mounting the SD4840E2 Module .....	88
I/O Connector Pin Out .....	89
Power Wiring .....	89
Input Wiring .....	91
Cable Shields .....	91

**Task 2: Installing the SD4840E2  
(continued)**

Output Wiring .....	91
Encoder Wiring .....	92
Differential Wiring .....	92
Single Ended Wiring .....	93
Installing the Stepper Motor .....	93
Outline Drawings .....	93
Mounting the Motor .....	94
Connecting the Load .....	94
Extending the Motor Cable .....	94
Installing the Motor Cable .....	94
Connecting the Motor .....	95
Motor Connector .....	95
Motor Wiring .....	95
Ethernet Connections .....	97
EtherNet/IP Connections .....	97
Non-DLR Applications .....	97
DLR Applications .....	97
PROFINET Connections .....	97
Non-MRP Applications .....	97
MRP Applications .....	97
Modbus TCP Connections .....	97

**Task 3: Set the IP Address and  
Protocol**

Determine the Best Method for Setting the IP Address .....	99
Use Factory Default Settings .....	99
Use the Embedded Web Server .....	100
Use the AMCI NET Configurator Utility .....	102

**Task 4: Implicit Communications  
with an EDS**

Obtain the EDS file .....	107
Install the EDS file .....	107
Start the EDS Hardware Installation Tool .....	107
Install the EDS File .....	108
Host System Configuration .....	110
Add the SD4840E2 to Your Project .....	110
Configure the SD4840E2 Driver .....	111
General Tab .....	111
Connection Tab .....	111
Configuration Tab .....	111
Buffering the I/O Data .....	112

**Task 5: Implicit Communications  
Without an EDS**

Host System Configuration .....	113
Add the SD4840E2 .....	113
Configure the SD4840E2 .....	116
Buffer I/O Data .....	116

**Task 6: EtherNet/IP Explicit  
Messaging**

Required Message Instructions .....	117
Create Four New Data Files .....	117
Add the Message Instructions to your Ladder Logic .....	118
Troubleshooting .....	121

**Task 7: Modbus TCP Configuration**

Enable Modbus TCP Protocol .....	123
Modbus Addressing .....	123
Modbus Table Mapping .....	123
Host Addressing .....	123
AMCI Modbus TCP Memory Layout .....	124
Supported Number of Connections .....	124
Supported Modbus Functions .....	125
Supported Modbus Exceptions .....	125

**Task 8: PROFINET Network  
Configuration**

Basic Steps .....	127
Download the GSDML files .....	127
GSDML File Installation .....	127
Configure the PROFINET Network .....	127
Add the SD4840E2 to the PROFINET Network .....	128
Configure the SD4840E2 .....	130
Set the I/O Configuration .....	131
Verify and Download the New Configuration .....	131
MRP Installations .....	132
Configure the SD4840E2 as an MRC .....	132

**Optional Task: Configure Your  
Network Interfaces**

Firewall Settings .....	133
Disable All Unused Network Interfaces ...	133
Configure Your Network Interface .....	133
Test Your Network Interface .....	134

# ABOUT THIS MANUAL

Read this chapter to learn how to navigate through this manual and familiarize yourself with the conventions used in it.

## Audience

This manual explains the set-up, installation, and operation of the SD4840E2 Stepper Motor Indexer / Driver from AMCI. It is written for the engineer responsible for incorporating these modules into a design, as well as the engineer or technician responsible for its actual installation.

## Applicable Units

The SD4840E2 supports EtherNet/IP, PROFINET, and Modbus TCP protocols. This manual applies to all SD4840E2 units, regardless of the network protocol it is configured for.

## Navigating this Manual

This manual is designed to be used in both printed and on-line formats. Its on-line form is a PDF document, which requires Adobe Acrobat Reader version 7.0+ to open it. The manual is laid out with an even number of pages in each chapter. This makes it easier to print a chapter to a duplex (double sided) printer.

Bookmarks of all the chapter names, section headings, and sub-headings were created in the PDF file to help navigate it. The bookmarks should have appeared when you opened the file. If they didn't, press the F5 key on Windows platforms to bring them up.

The PDF file is password protected to prevent changes to the document. You are allowed to select and copy sections for use in other documents and, if you own Adobe Acrobat version 7.0 or later, you are allowed to add notes and annotations.

## Manual Conventions

Three icons are used to highlight important information in the manual:



**NOTES** highlight important concepts, decisions you must make, or the implications of those decisions.



**CAUTIONS** tell you when equipment may be damaged if the procedure is not followed properly.



**WARNINGS** tell you when people may be hurt or equipment may be damaged if the procedure is not followed properly.

The following table shows the text formatting conventions:

Format	Description
Normal Font	Font used throughout this manual.
<i>Emphasis Font</i>	Font used the first time a new term is introduced.
<i>Cross Reference</i>	When viewing the PDF version of the manual, clicking on the cross reference text jumps you to referenced section.
<i>HTML Reference</i>	When viewing the PDF version of the manual, clicking on the HTML reference text will open your default web browser to the referenced web page.



**Trademark Notices**

The AMCI logo is a trademark of Advanced Micro Controls Inc. “CIP” is a trademark of Open DeviceNet Vendor Association, Inc. “EtherNet/IP” is a trademark of ControlNet International, Ltd. under license by Open DeviceNet Vendor Association, Inc. “Adobe” and “Acrobat” are registered trademarks of Adobe Systems Incorporated.

All other trademarks contained herein are the property of their respective holders.

**Revision Record**

This manual, 940-0S230, is the first revision of this manual. It was first released December 17th, 2018.

**Revision History**

940-0S230 Initial Release.

**Where To Go From Here**

This manual contains information that is of interest to everyone from engineers to operators. The table below gives a brief description of each chapter’s contents to help you find the information you need to do your job.

<b>Section Title</b>	<b>Intended Audience</b>
<i>Introduction to the SD4840E2</i>	Anyone new to the SD4840E2. This chapter gives a basic overview of the features available on the unit, typical applications, and electrical specifications.
<i>Move Profiles</i>	Anyone that needs detailed information on how the driver can be used to control motion in your application.
<i>Calculating Move Profiles</i>	Anyone that need information on calculating detailed move profiles.
<i>Homing the SD4840E2</i>	Anyone that needs detailed information on the available means to define a known position on the machine, including information on homing profiles available on the SD4840E2.
<i>Configuration Mode Data Format</i>	Anyone that needs detailed information on the format of the network data to and from the SD4840E2 used to configure the indexer/driver.
<i>Command Mode Data Format</i>	Anyone that needs detailed information on the format of the network data to and from the SD4840E2 used to command the indexer/driver.
<i>Motor and Power Supply Sizing</i>	Anyone that need performance data on compatible AMCI stepper motors or guidelines for using a foreign stepper motor.
<i>Installing the SD4840E2</i>	Anyone that must install an SD4840E2 on a machine. Includes information on mounting, grounding, and wiring specific to the units.
<i>Set the IP Address and Protocol</i>	Task instructions that covers the options for setting the IP address on an SD4840E2.
<i>Implicit Communications with an EDS</i>	Task instructions that cover how to add an SD4840E2 to an EtherNet/IP host that supports the use of EDS files.
<i>Implicit Communications Without an EDS</i>	Task instructions for adding an SD4840E2 to a project as a generic device. This configuration is for EtherNet/IP hosts that do not support EDS files while supporting implicit communications.
<i>EtherNet/IP Explicit Messaging</i>	Task instructions for adding message instructions to you host controller program that read and write data to the SD4840E2 through message instructions.
<i>Modbus TCP Configuration</i>	Task instructions for communicating with an SD4840E2 using the Modbus TCP protocol.
<i>PROFINET Network Configuration</i>	Task instructions that covers how to add an SD4840E2 to a PROFINET network.
<i>Optional: Configure Your Network Interfaces</i>	Instructions for the optional task of configuring network interfaces on your computer or laptop.



## INTRODUCTION TO THE SD4840E2

This manual is designed to get you quickly up and running with the SD4840 stepper indexer / driver. As such, it assumes you have some basic knowledge of stepper systems, such as the resolution you want run your motor at, and the reasons why you'd want to use Idle Current Reduction and the reasons why you wouldn't. If these terms or ideas are new to you, we're here to help. AMCI has a great deal of information on our website and we are adding more all the time. If you can't find what you're looking for at <http://www.amci.com>, send us an e-mail or call us. We're here to support you with all of our knowledge and experience.

### **The SD4840E2**

The AMCI SD4840E2 is a DC powered, 4.0 Arms, micro-stepping driver with an internal bus voltage equal to the supply voltage. The drive's built-in indexer accepts configuration and command data from a host system over its internal Ethernet ports. The SD4840E2 driver can be configured to use Ethernet/IP, PROFINET, or Modbus/TCP protocols, which makes these units easy to integrate the drive into a wide variety of controller systems.

Each unit has two Ethernet ports which are internally connected through an onboard, two port, 10/100 Mbps ethernet switch. These ports allow you to wire your network in a "daisy-chain" fashion, which may lower network wiring costs and complexities.

The two ports also allow the units to function as members of a redundant Device Level Ring (DLR) network when using the EtherNet/IP protocol or as clients in a Media Redundancy Protocol (MRP) network when using PROFINET.

In DLR environments, the units act as Beacon-Based Ring Nodes. All units can process beacon packets at the default rate of every 400 microseconds. Beacon-based nodes can respond faster to network changes than nodes that only process Announce packets.

### **General Functionality**

The combination of host and driver gives you several advantages:

- Sophisticated I/O processing can be performed in the host (PLC or other controller) before sending commands to the Networked Indexer/Driver.
- All motion logic is programmed in the host, eliminating the need to learn a motion control language
- The integral two port Ethernet switch simplifies network cabling
- The DLR functionality eliminates single point failures in EtherNet/IP environments
- The MRP functionality eliminates single point failures in PROFINET environments
- Eliminating the separate indexer lowers Total System Cost

The SD4840E2 is powered by a nominal 24 to 48 Vdc power source, and can accept surge voltages of up to 60Vdc without damage. The output motor current is fully programmable from 0.1 Arms to 4.0 Arms which makes the SD4840E2 compatible with the complete line of size 23 and size 34 stepper motors that are available from AMCI. In addition to the Motor Current setting, the Motor Steps per Turn, Idle Current Reduction, and Anti-Resonance Circuit features are also fully programmable. If you have used other stepper indexer products from AMCI you will find programming to be very similar to these products.

The SD4840E2 is a true RMS motor current control driver. This means that you will always receive the motor's rated torque regardless of the *Motor Steps/Turn* setting. (Drivers that control the peak current to the motor experience a 30% decrease in motor torque when microstepping a motor.) The SD4840E2 automatically switches from RMS to peak current control when the motor is idle to prevent overheating the motor.

## The SD4840E2 (continued)

### General Functionality (continued)

In addition to power and motor hookups, the SD4840E2 has three DC inputs and one DC output that are used by the indexer. Configuration data from the host sets the function of these points. The output can be configured to be a Fault Output or a general purpose output. Inputs accept 5 to 24 Vdc signals and they can be individually configured as a:

- CW or CCW Limit Switch
- Home Limit Switch
- Capture Encoder Position Input
- Stop Jog or Registration Move Input
- Start Indexer Move
- Emergency Stop Input
- General Purpose Input

### Encoder Functionality

In addition to the discrete I/O points, the SD4840E2 has three inputs for a 5Vdc differential quadrature encoder. The inputs will also accept 12 to 24Vdc single ended encoder inputs with current limiting resistors.

Using the encoder inputs gives you the ability to:

- Home the machine to the encoder marker pulse
- Detect motor stall conditions

The encoder input also allows you to drive the motor through a feature called *Electronic Gearing*. In this mode, the stepper motor follows the rotation of an external encoder. This encoder is typically attached to another motor. The ratio of encoder pulses to stepper pulses is programmable over a wide range. This mode electronically couples the two motors together through a programmable gear ratio.

Finally, the encoder allows you to use a feature called *Hybrid Control*. When enabled, this feature gives you servo-like performance without the need to tune servo PID loop parameters. When Hybrid Control is enabled, the SD4840E2 uses a motor mounted encoder to feed rotor position information back to the driver. The SD4840E2 uses this information during a move to dynamically increase or decrease the current through the motor's windings to the lowest values needed to achieve motion. This feature eliminates the possibility of resonance related motor stalls and can dramatically lower the motor's operating temperature, which prolongs the life of the motor.

When using this feature, a stepper motor will act in a manner very similar to a servo motor. This includes the possibility of a one count dither in position at zero speed. In order to overcome this condition, the SD4840E2 allows you to enable and disable the Hybrid Control feature on the fly. When the motor is at rest, a single message disables the Hybrid Control feature, and the motor acts as a stepper system, holding its position without dither. You can enable the Hybrid Control feature using the same message that initiates a move, minimizing the amount of motor current needed to complete it.

### Network Data Format


In order to support any host that communicates with the protocols supported by the SD4840E2, the format of the data read from and written to the unit is completely programmable. The format of the network input and output data can be programmed separately.


The smallest data size used by the SD4840E2 is the sixteen bit word, however some parameters and data values can exceed this size. For these thirty-two bit values, the default data format is referred to as the *multi-word* format. The data value is split between the hundreds digit and the thousands digit. For example, a value of 12,345 would have 12 placed in the first (lower addressed) word, and 345 placed in the second (higher addressed) word. This format greatly simplifies setting parameter values when programming command blocks.

## The SD4840E2 (continued)

### Network Data Format (continued)

The other data format is a signed thirty-two bit integer format. When using the thirty-two bit format, there is one additional parameter named *Binary\_Endian*. Its use is best explained with an example. The value of 123,456 equals 0001:E240 in hexadecimal. When storing and transmitting this data, some host controllers will store the least significant word (16#E240) in the lower addressed word in their data tables, while others will store the most significant word (16#0001) stored in the lower addressed word in their data tables. These controllers expect thirty-two bit values to be returned to them using the same format. Least significant word first is called *little endian*, most significant word first is called *big endian*. EtherNet/IP controllers use *little endian* format, while PROFINET and Modbus TCP controllers use *big endian* by default.

**NOTE**  The range of values when using the multi-word format is -32,768,000 to 32,767,999. When used in continuous rotation applications, it is possible to overflow these values. When a position value overflows, the value of the associated data words will become indeterminate. AMCI strongly suggests using the signed 32-bit integer format for continuous rotation applications.

**NOTE**  EtherNet/IP applications that use an EDS file have the parameters and data values defined as double integers. In these applications, the data formats should be programmed to the thirty-two bit integer format and the *Binary\_Endian* parameter should be set to *little endian*.

**Specifications**

**Driver Type**

Two bipolar MOSFET H-bridges with 20KHz PWM current control.

**Physical Dimensions**

Width: 0.9 inches max.  
 Depth: 4.5 inches max.  
 Height: 3.9 inches  
 5.0 inches min. with mating connectors

**Weight**

0.38 lb. (0.17 kg) with mating connectors

**Inputs**

Electrical Characteristics: .....  
 Differential. 560 Vac/dc opto-isolated. Can be wired as single ended inputs.  
 DC Inputs accept 3.5 to 27 Vdc without the need for an external current limiting resistor.  
 Encoder Inputs are designed for 5 Vdc differential (Operational range of 3 to 6 Vdc.) An external current limiting resistor is required for 12 to 24 Vdc operation.

**Output**

Electrical Characteristics:  
 Open Collector/Emitter. 560 Vac/dc opto-isolated. 30 Vdc, 20 mA max.  
 The Output can be programmed to be a general purpose output or a Fault Output.  
 The Fault Output is normally on. Turns off under the following conditions:  
 Reset ..... The driver initialization is not yet complete on power up.  
 Short Circuit ..... Motor Phase to Phase or Phase to Case  
 Over Temp ..... Heat Sink temperature exceeds 90° C (195° F)  
 Faults are reported in the Network Input Data and can be cleared through the Network Output Data.

**Motor Current**

Programmable from 0.1 to 4.0Arms in 0.1 Amp steps.

**+Vdc Auxiliary Current**

70 mA @ 24Vdc, 40mA @48Vdc

**Resolution**

Programmable to any value from 200 to 32,767 steps per revolution.

**Idle Current Reduction**

Programmable from 0% to 100% programmed motor current in 1% increments. Motor current is reduced to selected level if there is no motion for 1.5 seconds. Current is restored to full value when motion is started.

**Internal Power Fuse**

7 Amp Fast Blow. Fuse is not user replaceable.

**Environmental Specifications**

Input Power ..... 24 to 48 Vdc, surge to 60 Vdc without damage to module.  
 Ambient Operating Temperature  
 ..... -4° to 122° F (-20° to 50° C)  
 Storage Temperature  
 ..... -40° to 185° F (-40° to 85° C)  
 Humidity ..... 0 to 95%, non-condensing

**Motor Specifications**

Type ..... 2 phase hybrid. 4, 6, or 8 lead motor  
 Inductance .... 0.3 mH minimum. 2.5 to 45 mH recommended

**Status LED's**

See *Front Panel* on page 18 for a complete description.

**Connectors**

Mating connectors are supplied with the module and are also available separately under the following AMCI part numbers.

Connector	AMCI Part #	Wire	Strip Length	Min. Tightening Torque
I/O	MS-2x11	28 - 16 AWG	0.275 inches	Spring Cage Connector
Motor	MS-4M	28 - 12 AWG	0.394 inches	4.43lb-in (0.5 Nm)

## Indexer Functionality

The table below lists the functionality offered by the indexer built into the SD4840E2.

Feature	Description
Programmable Inputs	Each of the three inputs can be programmed as a Home Limit, Over Travel Limit, Capture Input, Stop Jog or Registration Move, E-Stop Input, or a General Purpose Input.
Programmable Output	The single output on the SD4840E2 can be programmed as a Fault Output or as a general purpose DC output point.
Encoder Inputs	Allows the SD4840E2 to use a quadrature encoder for position verification or Electronic Gearing.
Programmable Parameters	Starting Speed, Running Speed, Acceleration, Deceleration, Distance to Move, and Accel/Decel Types are fully programmable.
Homing	Allows you to set the machine to a known position. The SD4840E2 can home to a discrete input or to an encoder marker pulse.
Jog Move	Allows you to jog the motor in either direction based on an input bit from your host controller.
Synchrostep Move	Allows you to treat the SD4840E2 as a motion axis by constantly updating position and velocity data to the unit.
Registration Move	Allows you to jog the motor in either direction based on an input bit from your host controller. When a controlled stop is received, the move will output a programmable number of steps before coming to a stop.
Relative Move	Allows you to drive the motor a specific number of steps in either direction from the current location.
Absolute Move	Allows you to drive the motor from one known location to another known location.
Blend Move	Allows you to perform a sequence of relative moves without stopping between them.
Dwell Move	Allows you to perform a sequence of relative moves with a stop between each move that has a programmable length of time. Used to create highly accurate move profiles that avoid network latency issues.
Indexer Move	Allows you to program a move that does not start until one of the programmable inputs makes a transition.
Hold Move	Allows you to suspend a move and restart it without losing your position value.
Resume Move	Allows you to restart a previously held move operation.
Immediate Stop	Allows you to immediately stop all motion if an error condition is detected by your host controller.
Stall Detection	When the SD4840E2 uses an encoder, the encoder can be used to verify motion when a move command is issued.
Electronic Gearing	The SD4840E2 can be configured to control the position of a motor based on feedback from an external encoder. The ratio of encoder pulses to motor pulses is fully programmable and can be changed on-the-fly.
Hybrid Control	The SD4840E2 can be configured to use a motor mounted encoder for rotor position feedback and use this data to dynamically alter motor current to the minimum value needed to achieve motion. This operating mode eliminates the possibility of resonance induced motor stalls and can significantly reduce the operating temperature of the motor.

Table R1.1 Indexer Functionality

***Indexer Functionality (continued)***

**Stall Detection with SD4840E2 Units**

When Stall Detection is enabled, the SD4840E2 monitors the encoder for position changes, regardless of whether or not a move is in progress. If the error between the encoder position and the motor position exceeds forty-five degrees, the SD4840E2 responds in the following manner:

- The stall is reported in the network input data.
- The motor position becomes invalid. (The machine must be homed or the motor position preset before Absolute moves can be run again.)
- If a move was in progress, the move is stopped.

Note that a move does not have to be in progress for stall detection to be useful. Power is removed from the motor between moves if the Idle Current Percentage is set to 0%. By enabling stall detection, the unit can notify the system if the motor shaft moves more than forty-five degrees while power is removed from the motor.

***Driver Functionality***

This table summarizes the features of the stepper motor driver portion of the SD4840E2.


Feature	Benefits
RMS Current Control	RMS current control give the SD4840E2 the ability to drive the motor at its fully rated power when microstepping. Peak current controllers typically experience a 30% drop in power when microstepping a motor.
Programmable Motor Current	RMS current supplied to the motor can be programmed from 0.1 to 4.0 amps in 0.1 amp increments. This allows you to use the driver with the full line of AMCI stepper motors.
Programmable Idle Current Reduction	Extends motor life by reducing the motor current when not running. This extends the life of the motor by reducing its operating temperature.
Programmable Current Loop Gain	Allows you to tailor the driver circuitry to the motor’s impedance, thereby maximizing your motor’s performance.
Programmable Motor Steps/Turn	Allows you to scale your motor count to a real world value. (counts per inch, counts per degree, etc.)
Anti-Resonance Circuitry	This circuitry gives the SD4840E2 the ability to modify motor current waveforms to compensate for mechanical resonance in your system. This will give you smooth performance over the entire speed range of the motor.
Wiring Short Detection	Safety feature that removes power from the motor if a short is detected in one of the windings of the motor.
Over Temperature Detection	The SD4840E2 sets a warning bit in the network data when the temperature of the module approaches its safe operating threshold.
Over Temperature Protection	Protects the SD4840E2 from damage by removing power from the motor if the internal temperature of the driver exceed a safe operating threshold.

Table R1.2 Driver Functionality

**Idle Current Reduction**

Idle Current Reduction allows you to prolong the life of your motor by reducing its idling temperature. Values for this parameter range from 0% (no holding torque when idle) to 100%.

Idle current reduction should be used whenever possible. By reducing the current, you are reducing the I<sup>2</sup>R losses in the motor. Therefore, the temperature drop in the motor is exponential, not linear. This means that even a small reduction in the idle current can have a large effect on the temperature of the motor.

**NOTE**  Note that the reduction values are “to” values, not “by” values. Setting a motor current to 4 Arms and the current reduction to 25% will result in an idle current of 1Apk. (The SD4840E2 always switch from RMS to peak current control when the motor is idle.)



## I/O Connector

As shown in figure R1.1, the I/O connector is located on the top of the module. All digital I/O connections are made at this connector as well as the power supply connections. The mating connector is supplied with the SD4840E2 and is also available from AMCI under the part number MS-2X11. It is also available from Phoenix Contact under their part number 173 88 98.

## Available Discrete Inputs

The SD4840E2 has three discrete DC inputs that accept 3.5 to 27 Vdc signals. (5 to 24 Vdc nominal) They can be wired as differential, sinking, or sourcing inputs. How the SD4840E2 uses these inputs is fully programmable as is their active states. (Inputs can be programmed as Normally Open (NO) or Normally Closed (NC) inputs.)

### Home Input

Many applications require that the machine be brought to a known position before normal operation can begin. This is commonly called “homing” the machine or bringing the machine to its “home” position. The SD4840E2 allows you to define this starting position in three ways. The first is with a Position Preset Command. The second is with a sensor mounted on the machine. When you define one of the inputs as the Home Input, you can issue commands to the SD4840E2 that will cause the unit to seek this sensor. The third option is homing to the Z pulse of a quadrature encoder. When using the Z pulse, you can use one of the inputs or a network data bit as a home proximity sensor. How the SD4840E2 actually finds the Home sensor is described in chapter 4, [Homing the SD4840E2](#), starting on page 51.

### CW Limit Switch or CCW Limit Switch

Each input can be defined as a CW or CCW Limit Switch. When configured this way, the inputs are used to define the limits of mechanical travel. For example, if you are moving in a clockwise direction and the CW Limit Switch activates, all motion will immediately stop. At this point, you will only be able to move in the counter-clockwise direction.

### Start Indexer Move Input

Indexer Moves are programmed through the Network Data like every other move. The only difference is that Indexer Moves are not run until an input that is configured as a Start Indexer Move Input makes an inactive-to-active state transition. This allows the SD4840E2 to run critically timed moves that cannot be reliably started from the network due to data transfer lags.

If the quadrature encoder is enabled and one of the discrete DC inputs is programmed as a Start Indexer Move Input, then the quadrature encoder position data will be captured whenever the DC input makes a transition. An inactive-to-active state transition on the DC input will also trigger an Indexer Move if one is pending.

### Emergency Stop Input

When an input is defined as an Emergency Stop, or E-Stop Input, motion will immediately stop when this input becomes active. The driver remains enabled and power is supplied to the motor. No move can begin while this input is active.

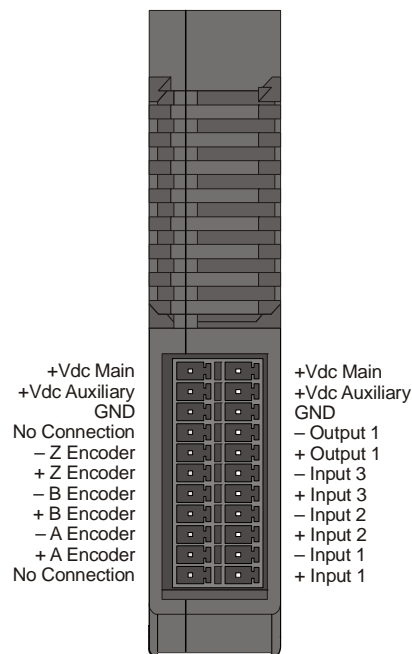


Figure R1.1 I/O Connector

## Available Discrete Inputs (continued)

### Stop Jog or Registration Move Input

When an input is configured as a Stop Jog or Registration Move Input, triggering this input during a Jog Move or Registration Move will bring the move to a controlled stop. The controlled stop is triggered on an inactive-to-active state change on the input. Only Jog Moves and Registration Moves can be stopped this way, all other moves ignore this input.

If the quadrature encoder is enabled, the quadrature encoder position data will be captured when the DC input makes an inactive-to-active transition if it is configured as a Stop Jog or Registration Move Input. The encoder position data is not captured if a Jog or Registration Move is not in progress. If you want to capture encoder position data on every transition of a DC input, configure it as a Start Indexer Move Input.

### Capture Encoder Position Input

As described in the *Start Indexer Move Input* and *Stop Jog or Registration Move Input* sections above, the SD4840E2 can be configured to capture the encoder position value on a transition of a discrete DC input.

### General Purpose Input

If your application does not require all three inputs, you can configure the unused inputs as General Purpose Inputs. The inputs are not used by the SD4840E2, but the input state is reported in the network data.

## Encoder Feedback Inputs

The SD4840E2 has three 5 Vdc differential inputs that accept quadrature encoder signals. These inputs can accept single ended signals of 5 to 24 Vdc with the addition of external current limiting resistors.

An encoder is used by the SD4840E2 in multiple ways.

- When the encoder is mounted on the back of the motor controlled the SD4840E2, the encoder can be used for position feedback or stall detection. The position data of the encoder can be preset to any value within its range, it is reported in the network data, and it can be captured during a move.
- The Z pulse can be used to home the machine as described in chapter 4, *Homing the SD4840E2*, starting on page 51. A home proximity sensor can be wired into the SD4840E2 or the Network Home Proximity Bit in the network data can be used to determine which occurrence of the Z-pulse defines the home of the machine.
- The addition of an encoder also allows you to perform an *Encoder Registration Move*, which is described in the *Encoder Registration Moves* section, starting on page 29.
- The SD4840E2 must use an encoder with a feature called *Hybrid Control*. When using this feature, the SD4840E2 uses feedback from the motor mounted encoder to dynamically adjust motor current to the minimum value needed to achieve motion. Hybrid Control will also eliminate resonance induced stalls during the move.
- Finally, it is also possible to use an encoder that is not mechanically coupled to the motor controlled by the SD4840E2. This configuration allows you to monitor the encoder data, or use a feature called *Electronic Gearing*. When this feature is active, the SD4840E2 will change the position of the motor in response to a change in encoder position. The ratio of encoder turns to motor turns is fully programmable.

### Available Discrete Output

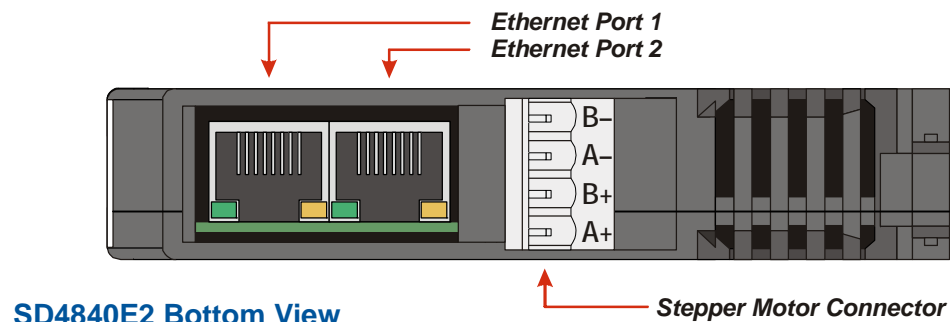
The SD4840E2 has a single DC output that has a maximum rating of 30 Vdc at 20 mA. The output can be configured to be a general purpose output or a Fault Output. When configured as a Fault Output, the output will conduct under normal conditions and will switch off when a fault occurs. The following faults affect the Fault Output:

- Reset ..... The driver initialization is not yet complete on power up.
- Short Circuit ... Motor Phase to Phase or Phase to Earth Ground
- Over Temp ..... Heat Sink temperature exceeds 90° C (195° F)

Faults are reported in the Network Input Data and can be cleared through the Network Output Data.

### Ethernet Ports

The two Ethernet ports are located on the bottom of the SD4840E2. When using the drive in non-redundant networks, either port can be used to attach to your core network. The second port can be used to daisy-chain an additional device onto the network. When installed in a DLR or MRP network, both ports are used.



SD4840E2 Bottom View

Figure R1.2 Location of Ethernet Ports and Motor Connector

### Motor Connector

Figure R1.2 also shows the location of the Stepper Motor Connector. The mate to this connector is included with the SD4840E2 and is also available from AMCI under the part number MS-4M. It is also available from Phoenix Contact under their part number 187 80 37.

**Front Panel**

The front panel of the SD4840E2 contains the drive status LED as well as the two network status LED's.

**Status LED**

The Status LED is a bi-color red/green LED shows the general status of the module.

- **Steady Green:** Module OK
- **Steady Red:** An Overtemperature Fault or Motor Short Circuit Fault exists. Note that the SD4840E2 will only detect short circuit faults when the motor current is enabled.
- **Blinking Green:** Successful write to flash memory. Power must be cycled to the module before additional commands can be written to it.
- **Blinking Red:** Failed write to flash memory. You must cycle power to the module to clear this fault.
- **Alternating Red/Green:** There is a communications error between the main processor and the ethernet co-processor within the drive. You must cycle power to the module to attempt to clear this fault.

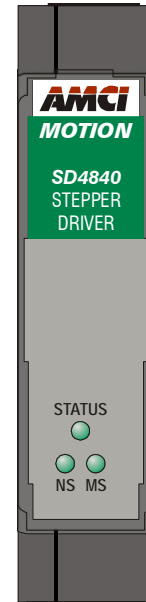


Figure R1.3 SD4840E2 Front Panel

**Module Status (MS) LED**

The Module LED is a bi-color red/green LED that indicates the health of the Ethernet co-processor on the module. This LED will briefly flash red once on power up and then flash green while the unit is initializing. It will be on steady green when the Ethernet co-processor is functioning correctly. It will turn red on an error. If this occurs, cycle power to the module.

**Network Status (NS) LED**

The Network Status LED is a bi-color red/green LED. The state of the LED depends on the protocol the SD4840E2 is configured to for.

LED State	EtherNet/IP Definition	Modbus TCP Definition	PROFINET Definition
Off	No Power	No power or no TCP connections	No power, duplicate IP address on the network, or no connection to IO Controller.
Alternating Red/Green	Power up Self-Test	Power up Self-Test	Power up Self-Test
Flashing Green	Ethernet connection, but no CIP connections	Indicates number of connections with 2 second delay between group. The SD4804E2 supports up to 7 concurrent connections.	On-line, Stop state. A connection with the IO Controller is established and it is in its STOP state.
Steady Green	Valid Ethernet network and CIP connections	Should not occur. LED should always flash when network is connected.	On-line, Run state. A connection with the IO Controller is established and it is in its RUN state.
Flashing Red	Network Connection Timeout	Not implemented in Modbus TCP.	Not Implemented
Steady Red	Duplicate IP address on network.		Not Implemented

Table R1.3 Network Status LED States

# REFERENCE 2

## MOVE PROFILES

When a move command is sent to the SD4840E2, the module calculates the entire profile before starting the move or issuing an error message. This chapter explains how the profiles are calculated and the different available moves.

### Definitions

#### Units of Measure

**Distance:** Every distance is measured in steps. When you configure the unit, you will specify the number of steps you want to complete one rotation of the motor shaft. It is up to you to determine how many steps are required to travel the appropriate distance in your application.

**Speed:** All speeds are measured in steps/second. Since the number of steps needed to complete one shaft rotation is determined by your programming, it is up to you to determine how many steps per second is required to rotate the motor shaft at your desired speed.

**Acceleration:** The typical unit of measure for acceleration and deceleration is steps/second/second, or steps/second<sup>2</sup>. However, when programming an SD4840E2, all acceleration and deceleration values must be programmed in the unit of measure of steps/second/millisecond.

- To convert from steps/second<sup>2</sup> to steps/second/millisecond, divide the value by 1000. This must be done when converting from a value used in the equations to a value programmed into the unit.
- To convert from steps/second/millisecond to steps/second<sup>2</sup>, multiply the value by 1000. This must be done when converting from the value programmed into a unit to the value used in the equations.

#### Motor Position

Motor Position is defined in counts, and its limits are based on the data format you choose when configuring the unit. The default multi-word format limits the Motor Position range from -32,768,000 to +32,767,999. If you choose the thirty-two bit double integer format, the range is -2,147,483,648 to +2,147,483,647. In continuous rotation applications, you should choose the double integer format.

#### Home Position

The Home Position is any position on your machine that you can sense and stop at. There are two ways to defining the Home Position. The first is using the Preset Position command to set the Motor Position register to a known value. The second method is using one of the *Find Home* commands. If you use the unit's *Find Home* commands, the motor position and encoder position registers will automatically be set to zero once the home position is reached. Defining a Home Position is completely optional. Some applications, such as those that use the SD4840E2 for speed control, don't require position data at all.

#### Count Direction

Relative moves with a positive offset will increase the motor position register reported back to the host. Some of the moves, such as the Jog Move, have a positive and negative command. A positive command, such as the CW Jog Move command, will result in an increase in the motor position value. If the motor is wired as shown in this manual, these moves will result in clockwise rotation when looking at the motor shaft.

#### Starting Speed

The Starting Speed is the speed that most moves will begin and end at. This value is set while configuring the unit and it has a valid range of 1 to 1,999,999 steps/second. This value is typically used to start the move above the motor's low frequency resonances and, in micro-stepping applications, to limit the amount of time needed for acceleration and deceleration. AMCI does not specify a default value in this manual because it is very dependent on motor size and attached load. With that said, a starting speed between 0.25 and 0.5 RPS is usually a good starting point.

## Definitions (continued)

### Target Position

The Target Position is the position that you want the move to end at. There are two ways to define the Target Position, with relative coordinates or absolute coordinates.

#### Relative Coordinates

Relative coordinates define the Target Position as an offset from the present position of the motor. Most SD4840E2 moves use relative coordinates.

- ▶ The range of values for the Target Position when it is treated as an offset is  $\pm 8,388,607$  counts. Assuming that the motor is wired to the SD4840E2 as shown in this manual, positive offsets will result in clockwise moves, while negative offsets result in counter-clockwise moves.
- ▶ The Motor Position value reported back to the host can exceed  $\pm 8,388,607$  counts. The only way to move beyond  $\pm 8,388,607$  counts is with multiple relative moves or jog commands.

#### Absolute Coordinates

Absolute coordinates treat the Target Position as an actual position on the machine. Note that you must set the Home Position on the machine before you can run an Absolute Move. (See [Home Position](#) on the previous page.)

- ▶ The range of values for the Target Position when it is treated as an actual position on the machine is  $\pm 8,388,607$  counts. The move will be positive if the Target Position is greater than the Current Position and negative if the Target Position is less than the Current Position.
- ▶ The Motor Position value reported back to the host can exceed  $\pm 8,388,607$  counts. However, you cannot move beyond  $\pm 8,388,607$  counts with an Absolute Move. The only way to move beyond  $\pm 8,388,607$  counts is with multiple relative moves or jog commands.

## Definition of Acceleration Types

Most of the move commands allow you to define the acceleration type used during the move. The SD4840E2 supports three types of accelerations and decelerations. The type of acceleration used is controlled by the *Acceleration Jerk* parameter.

### What is jerk?

Just as speed is a measurement of change in position per unit time and acceleration is a measurement of change in speed per unit time, jerk is a measurement of change in acceleration per unit time. Likewise, just as a change in position equals speed \* time,  $\Delta p = s(t)$ , and a change in speed equals acceleration \* time,  $\Delta s = a(t)$ , a change in acceleration equals jerk \* time,  $\Delta a = j(t)$ . Jerk has units of steps/sec<sup>3</sup>.

The SD4840E2 uses the jerk property to smoothly change the acceleration applied during the move. In this case, the speed of the move does not increase linearly, but exponentially, resulting in an “S” shaped curve. This limits mechanical shocks to the system as the load accelerates.

In order to keep the Acceleration Jerk parameter value that is programmed into the SD4840E2 below sixteen bits, the SD4840E2's Acceleration Jerk parameter does not have units of steps/sec<sup>3</sup>. The Acceleration Jerk parameter equals  $(\{100 * \text{jerk in steps/sec}^3\} / \text{acceleration in steps/sec}^2)$ . This translates to the jerk property in steps/sec<sup>3</sup> equalling  $(\{\text{Acceleration Jerk parameter}/100\} * \text{acceleration in steps/sec}^2)$ . With the range of values for the Acceleration Jerk parameter being 1 to 5,000, the jerk value ranges from  $0.01a$  to  $50a$  where “ $a$ ” is the acceleration value in steps/sec<sup>2</sup>. For example, if the acceleration is programmed to 20,000 steps/sec<sup>2</sup>, then the value of the jerk property used by the module can be programmed to be between 200 steps/sec<sup>3</sup> ( $0.01 * 20,000$ ) and 1,000,000 steps/sec<sup>3</sup> ( $50 * 20,000$ ).



## Definition of Acceleration Types (continued)

### Constant Acceleration

When the Acceleration Jerk parameter equals zero, the axis accelerates (or decelerates) at a constant rate until the programmed speed is reached. This offers the fastest acceleration, but consideration must be given to insure the smoothest transition from rest to the acceleration phase of the move. The smoothest transition occurs when the configured Starting Speed is equal to the square root of the programmed Acceleration value. Note that other values will work correctly, but you may notice a quick change in velocity at the beginning or end of the acceleration phase.

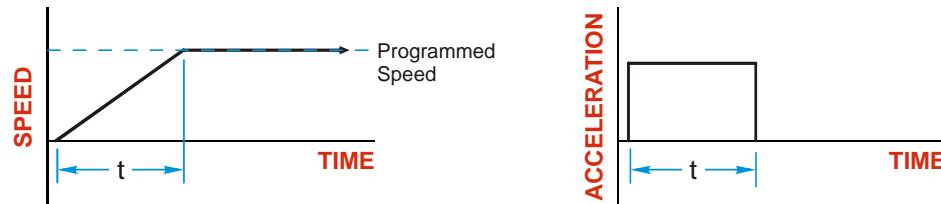


Figure R2.1 Constant Acceleration

Additional information, including example move calculations, can be found in section 3, *Calculating Move Profiles* starting on page 41.

### S-Curve Accelerations

When the Acceleration Jerk parameter value is in the range of 1 to 5,000, the SD4840E2 uses this value to accelerate and decelerate the rate of acceleration. This is known as S-Curve acceleration because of the shape of the speed curve that results from the variable acceleration.

When using S-Curve accelerations, the starting speed does not have to be equal to the square root of the programmed acceleration value. The S-Curve acceleration will provide smooth transitions at the beginning and end of the acceleration phase.

#### Trapezoidal S-Curve Acceleration

When the Acceleration Jerk parameter is set high, Trapezoidal S-Curve acceleration usually results. The acceleration value quickly increases (accelerates) until it reaches the value of the Acceleration Parameter. At this point, the acceleration remains constant until the SD4840E2 begins to apply the jerk property value to decrease the acceleration value until it equals zero when the programmed maximum speed is reached. Figure R2.2 shows a trapezoidal curve when the Acceleration Jerk setting results in the acceleration being constant for half of the acceleration time. With this setting, the Trapezoidal S-Curve acceleration only requires 33% more time to achieve the same velocity as a Constant Acceleration move.

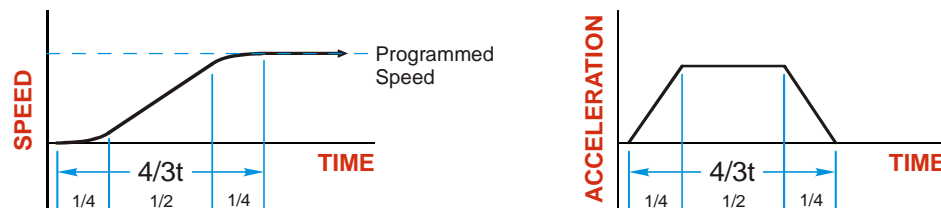


Figure R2.2 Trapezoidal S-Curve Acceleration

**Definition of Acceleration Types (continued)**

**S-Curve Accelerations (continued)**

**Triangular S-Curve Acceleration**

When the Acceleration Jerk parameter is set low, Triangular S-Curve acceleration usually results. This occurs because the programmed maximum acceleration value is not reached before the SD4840E2 must start decreasing the acceleration value as the move’s speed approaches its programmed maximum value. Triangular S-Curve is the smoothest form of acceleration, but the time needed to reach the move’s programmed speed is increased. An example is shown in figure R2.3 where the acceleration and jerk settings results in a move that takes twice as long as a Constant Acceleration move to achieve the same velocity.

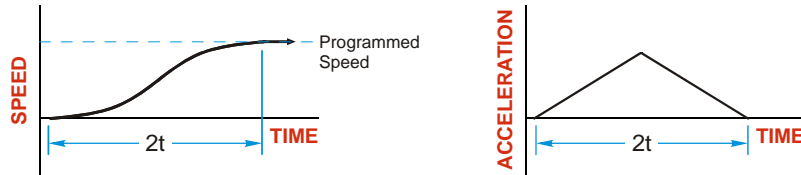


Figure R2.3 Triangular S-Curve Acceleration

Additional information, including example move calculations, can be found in section 3, *Calculating Move Profiles* starting on page 41.

**A Simple Move**

As shown in the figure below, a move from A (Current Position) to B (Target Position) consists of several parts.

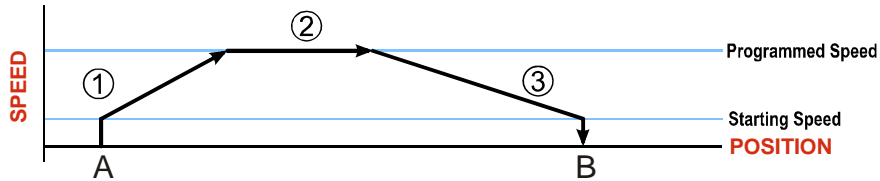


Figure R2.4 A Trapezoidal Profile

- 1) The move begins at point A, where the motor jumps from rest to the configured *Starting Speed*. The motor then accelerates at the programmed *Acceleration Value* until the speed of the motor reaches the *Programmed Speed*. Both the *Acceleration Value* and the *Programmed Speed* are programmed when the move command is sent to the SD4840E2.
- 2) The motor continues to run at the *Programmed Speed* until it reaches the point where it must decelerate before reaching point B.
- 3) The motor decelerates at the *Deceleration Value*, which is also programmed by the move command, until the speed reaches the *Starting Speed*, which occurs at the Target Position (B). The motor stops at this point. Note that the acceleration and deceleration values can be different in the move.

Figure R2.4 above shows a Trapezoidal Profile. A Trapezoidal Profile occurs when the *Programmed Speed* is reached during the move. This occurs when the number of steps needed to accelerate and decelerate are less than the total number of steps in the move. Figure R2.5 below shows a Triangular Profile. A Triangular Profile occurs when the number of steps needed to accelerate to the *Programmed Speed* and decelerate from the *Programmed Speed* are greater than the total number of steps in the move. In this case, the profile will accelerate as far as it can before decelerating and the *Programmed Speed* is never reached.

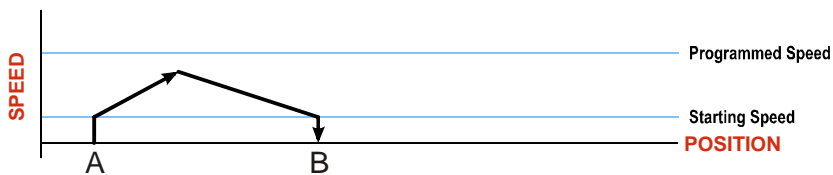


Figure R2.5 A Triangular Profile

## Controlled and Immediate Stops

Once a move is started, there are several ways to stop the move before it comes to an end. These stops are broken down into two types:

- ▶ **Controlled Stop:** The axis immediately begins decelerating at the move's programmed deceleration value until it reaches the configured Starting Speed. The axis stops at this point. The motor position value is still considered valid after a Controlled Stop and the machine does not need to be homed again before Absolute Moves can be run.
- ▶ **Immediate Stop:** The axis immediately stops outputting pulses regardless of the speed the motor is running at. Because it is possible for the inertia of the load attached to the motor to pull the motor beyond the stopping point, the motor position value is considered invalid after an Immediate Stop and the machine must be homed again before Absolute Moves can be run.

## Host Control

**Hold Move Command:** This command can be used with some moves to bring the axis to a Controlled Stop. The move can be resumed and finished, or it can be aborted. Not all moves are affected by this command. The section *Basic Move Types*, starting on page 24, describes each move type in detail, including if the move is affected by this command.

**Immediate Stop Command:** When this command is issued from the host, the axis will come to an Immediate Stop. The move cannot be restarted and the machine must be homed again before Absolute Moves can be run.

## Hardware Control

**Stop Jog or Registration Move Input:** Triggering this input type during a Jog Move or Registration Move will bring the move to a controlled stop. The controlled stop is triggered on an inactive-to-active state change on the input. Only Jog Moves and Registration Moves can be stopped this way, all other moves ignore this input.

**CW Limit and CCW Limit Inputs:** In most cases, activating these inputs during a move will bring the axis to an Immediate Stop. The exceptions are the *CW/CCW Find Home* commands, the *CW/CCW Jog Move* commands, and the *CW/CCW Registration Move* commands. The *CW/CCW Find Home* commands are explained in chapter 3, *Homing the SD4840E2*, which starts on page 51. The *CW/CCW Jog Move* commands are fully explained on page 26, and the *CW/CCW Registration Move* commands are fully explained on page 27.

**Emergency Stop Input:** It is possible to configure an input as an Emergency Stop Input. When an Emergency Stop Input is activated, the axis will come to an Immediate Stop, regardless of the direction of travel.

## Basic Move Types

### Relative Move

Relative Moves move an offset number of steps ( $n$ ) from the Current Position ( $A$ ). A trapezoidal profile is shown to the right, but Relative Moves can also generate triangular profiles. The command's Target Position is the move's offset. The offset can be in the range of  $\pm 8,388,607$  counts. Positive offsets will result in clockwise moves, while negative offsets result in counter-clockwise moves.

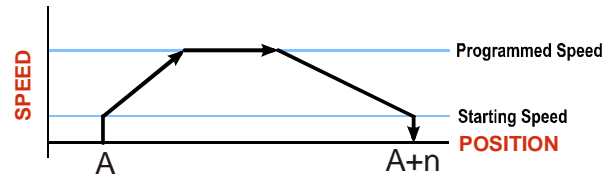


Figure R2.6 Relative Move

#### NOTE

- 1) You do not have to preset the position or home the machine before you can use a Relative Moves. That is, the Position Invalid status bit can be set.
- 2) Relative Moves allow you to move your machine without having to calculate absolute positions. If you are indexing a rotary table, you can preform a relative move of  $30^\circ$  multiple times without recalculating new positions in your controller. If you perform the same action with Absolute Moves, you would have to calculate your  $30^\circ$  position followed by your  $60^\circ$  position, followed by your  $90^\circ$  position, etc.

Relative Moves can be brought to a Controlled Stop by using the Hold Move Command from the network data. When the command is accepted, the axis will immediately decelerate at the programmed rate and stop. When stopped successfully, the SD4840E2 will set a *Hold State* bit in the input data table. The Relative Move can be restarted with the Resume Move command from the network data or the move can be aborted. The Resume Move command allows you to change the move's Programmed Speed, Acceleration Value and Type, and the Deceleration Value and Type. The Target Position cannot be changed with the Resume Move Command.

### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Relative Move by using the Resume Move command or abandon the held move by starting a new one. The use of the Hold Move and Resume Move bits is further explained in the *Controlling Moves In Progress* section starting on page 36.

### Immediate Stop Conditions

- The Immediate Stop bit makes a  $0 \rightarrow 1$  transition in the Network Input Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached. If the limit that is reached is the same as the direction of travel, for example, hitting the CW limit while running a CW move, a *Reset Errors* command must be issued before moves are allowed in that direction again. If the limit that is reached is opposite the direction of travel, a *Reset Errors* command does not have to be issued.

## Basic Move Types (continued)

### Absolute Move

Absolute Moves move from the Current Position (A) to a given position (B). (The SD4840E2 calculates the number of steps needed to move to the given position and moves that number of steps.) A trapezoidal profile is shown to the right, but Absolute Moves can also generate triangular profiles. The command's Target Position can be in the range of  $\pm 8,388,607$  counts. The move will be clockwise if the Target Position is greater than the Current Position and counter-clockwise if the Target Position is less than the Current Position.

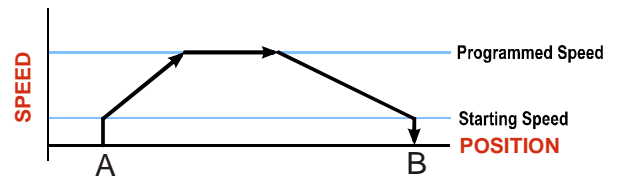


Figure R2.7 Absolute Move

#### NOTE

- 1) The *Home Position* of the machine must be set before running an Absolute Move. See chapter 4, *Homing the SD4840E2*, which starts on page 51, for information on homing the machine.
- 2) The Motor Position must be valid before you can use an Absolute Move. The Motor Position becomes valid when you preset the position or home the machine.
- 3) Absolute Moves allow you to move your machine without having to calculate relative positions. If you are controlling a rotary table, you can drive the table to any angle without having to calculate the distance to travel. For example an Absolute Move to  $180^\circ$  will move the table to the correct position regardless of where the move starts from.

#### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Absolute Move by using the Resume Move bit or abandon the held move by starting a new one. The use of the Hold Move and Resume Move bits is explained in the *Controlling Moves In Progress* section starting on page 36.

#### Immediate Stop Conditions

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Data.
- A inactive-to-active transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached. If the limit that is reached is the same as the direction of travel, for example, hitting the CW limit while running a CW move, a *Reset Errors* command must be issued before moves are allowed in that direction again. If the limit that is reached is opposite the direction of travel, a *Reset Errors* command does not have to be issued.

## Basic Move Types (continued)

### CW/CCW Jog Move

Jog Moves move in the programmed direction as long as the command is active. Two commands are available, the CW Jog Move will cause clockwise motion when the motor is wired as shown in this manual, while the CCW Jog Move will cause counter-clockwise motion. These commands are often used to give the operator manual control over the axis.

Jog Moves are typically used to drive the machine under direct operator control, but they can also be used when you are interested in controlling the speed of the shaft instead of its position. One such application is driving a conveyor belt. To accommodate these applications, the running speed, acceleration, and deceleration of the Jog Move can be changed *while the move is in progress*.

The CW Limit and CCW Limit inputs behave differently for CW/CCW Jog Moves and CW/CCW Registration Moves than all other move types. Like all moves, activating a limit will bring the move to an Immediate Stop. Unlike other moves, a Jog or Registration move can be started when an end limit switch is active provided that the commanded direction is opposite that of the activated switch. For example, a CW Jog Move can be issued while the CCW limit switch is active. This allows you to move off of an activated end limit switch.

As shown below, a Jog Move begins at the programmed Starting Speed, accelerates at the programmed rate to the Programmed Speed and continues until a stop condition occurs. If it is a *Controlled Stop Condition*, the SD4840E2 will decelerate the motor to the starting speed and stop without losing position. If it is an *Immediate Stop Condition*, the motion stops immediately and the position becomes invalid.

It is possible to change the speed of a Jog Move without stopping the motion. The Programmed Speed, Acceleration, and Deceleration parameters can be changed during a Jog Move. When the Programmed Speed is changed, the motor will accelerate or decelerate to the new Programmed Speed using the new accelerate/decelerate parameter values. If you write a Programmed Speed to the SD4840E2 that is less than the configured Starting Speed, the Jog Move will continue at the present speed and the *Invalid\_Jog\_Change* status bit will be set to “1”.

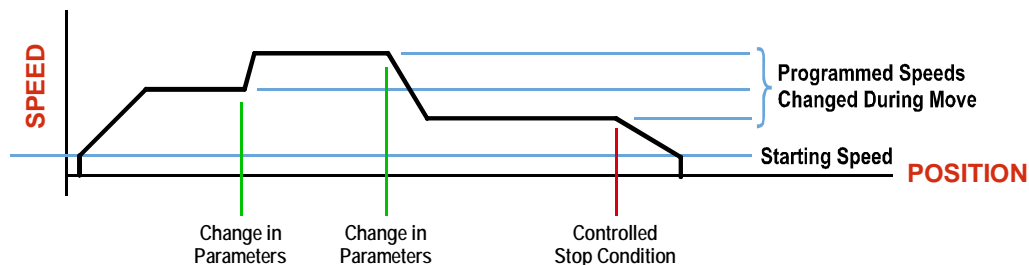


Figure R2.8 Jog Move

#### Controlled Stop Conditions

- The Jog Move Command bit is reset to “0”.
- An inactive-to-active transition on an input configured as a *Stop Jog or Registration Move* Input.
- You toggle the Hold Move control bit in the Network Output Data. The use of the Hold Move and Resume Move bits is explained in the *Controlling Moves In Progress* section starting on page 36.



**Basic Move Types (continued)****CW/CCW Jog Move (continued)****Immediate Stop Conditions**

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Data.
- A inactive-to-active transition on an input configured as an E-Stop Input.
- A CW/CCW Limit Switch is reached. If the limit that is reached is the same as the direction of travel, for example, hitting the CW limit while running a CW move, a *Reset Errors* command must be issued before moves are allowed in that direction again. If the limit that is reached is opposite the direction of travel, a *Reset Errors* command does not have to be issued.



**NOTE** Note that it is possible to *start* a move while a CW or CCW Limit Switch is active as long as the direction of travel is *opposite* that of the activated Limit Switch. For example, it is possible to start a clockwise Jog Move while the CCW Limit Switch is active.

**CW/CCW Registration Move**

Similar to a Jog Move, a Registration Move will travel in the programmed direction as long as the command is active. CW Registration Moves will cause clockwise motion when the motor is wired as shown in this manual, while CCW Registration Moves will cause counter-clockwise motion. When the command terminates under Controlled Stop conditions, the SD4840E2 will output a programmed number of steps as part of bringing the move to a stop. Note that all position values programmed with a Registration Move are relative values, not absolute machine positions.

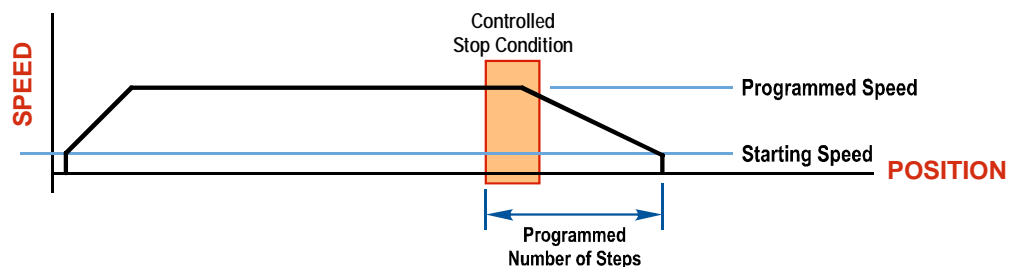


Figure R2.9 Registration Move

If the Programmed Number of Steps are less than the number of steps needed to bring the axis to a stop based on the Programmed Speed and Deceleration values set with the command, the SD4840E2 will decelerate at the programmed Deceleration value until it has output the Programmed Number of Steps and then stop the move without further deceleration.

**Basic Move Types (continued)****CW/CCW Registration Move (continued)**

An additional feature of the SD4840E2 is the ability to program the driver to ignore the Controlled Stop conditions until a minimum number of steps have occurred. This value is programmed through the Minimum Registration Move Distance parameter, which is set when you command the Registration Move. The figure below shows how the Minimum Registration Move Distance parameter affects when the Stop Condition is applied to the move. As shown in the second diagram, Controlled Stop conditions are level triggered, not edge triggered. If a Controlled Stop Condition occurs before the Minimum Registration Move Distance is reached and stays active, the move will begin its controlled stop once the Minimum Registration Move Distance is reached.

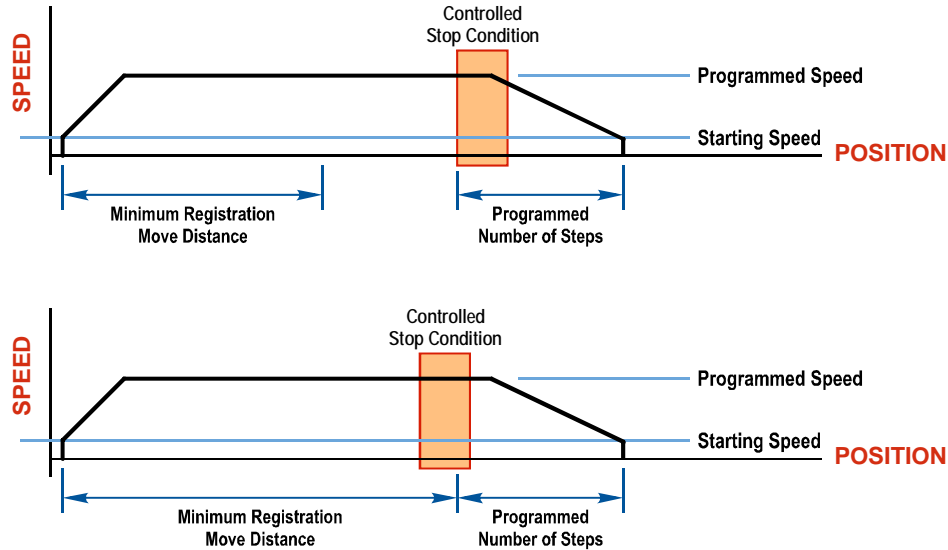


Figure R2.10 Min. Registration Move Distance

**Controlled Stop Conditions**

- The Registration Move Command bit is reset to “0”.
- A positive transition on an input configured as a *Stop Jog or Registration Move Input*.

**NOTE** Starting a Registration Move with a *Stop Jog or Registration Move Input* in its active state will result in a move of (*Minimum Registration Distance + Programmed Number of Steps*).

- You toggle the Hold Move control bit in the Network Output Data. The SD4840E2 responds by using the programmed Deceleration value to bring the move to a stop, without using the programmed Programmed Number of Steps. A Registration Move does not go into the Hold State if the Hold Move control bit is used to stop the move and it cannot be restarted.

**Immediate Stop Conditions**

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW/CW Limit Switch is reached. If the limit that is reached is the same as the direction of travel, for example, hitting the CW limit while running a CW move, a *Reset Errors* command must be issued before moves are allowed in that direction again. If the limit that is reached is opposite the direction of travel, a *Reset Errors* command does not have to be issued.

**NOTE** Note that it is possible to *start* a move while a CW or CCW Limit Switch is active as long as the direction of travel is *opposite* that of the activated Limit Switch. For example, it is possible to start a clockwise Registration Move while the CCW Limit Switch is active.

## Encoder Registration Moves

When the SD4840E2 is configured to use a quadrature encoder, the position value from the encoder can be used as a registration mark to control when the move begins to decelerate. Absolute and relative type moves are both supported.

**NOTE** You do not have to preset the position or home the machine before you can use a relative Encoder Registration Move.

The figure below represents either a relative Encoder Registration Move of 11,000 counts or an absolute Encoder Registration Move to position 16,000. The figure shows that the encoder position you program in the move defines the point at which the motor begins to decelerate and stop. *It does not define the stopping position as it does in other move types.* The endpoint of the move depends on the speed of the motor when the programmed encoder position is reached and the deceleration values. This behavior is different from Absolute and Relative Moves where the position you program into the move is the end point of the move.

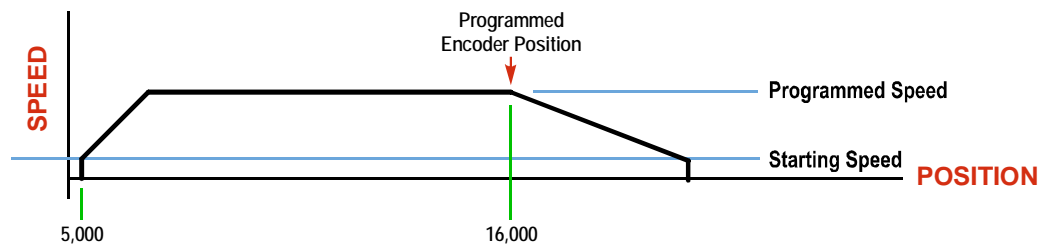


Figure R2.11 Encoder Registration Move

### Controlled Stop Conditions

- The move completes without error
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Encoder Registration Move by using the Resume Move bit or abandon the held move by starting a new one. The use of the Hold Move and Resume Move bits is explained in the *Controlling Moves In Progress* section starting on page 36.

### Immediate Stop Conditions

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached. If the limit that is reached is the same as the direction of travel, for example, hitting the CW limit while running a CW move, a *Reset Errors* command must be issued before moves are allowed in that direction again. If the limit that is reached is opposite the direction of travel, a *Reset Errors* command does not have to be issued.

## Assembled Moves

All of the moves explained so far must be run individually to their completion or be stopped before another move can begin. The SD4840E2 gives you the ability to assemble more complex profiles from a series of relative moves that are stored in memory and then run with a single command. Each Assembled Move can consist of 2 to 16 segments. Two types of Assembled Moves exist in the SD4840E2:

- **Blend Move** - A Blend Move gives you the ability to string multiple relative moves together and run all of them sequentially without stopping the shaft between moves. All of the moves are in the same direction.
- **Dwell Move** - A Dwell Move gives you the ability to string multiple relative moves together, and the SD4840E2 will stop between each move for a programmed *Dwell Time*. Because motion stops between each segment, a Dwell Move allows you to reverse direction during the move.

**Assembled Moves (continued)**

**Blend Move**

Each Relative Move defines a *segment* of the Blend Move. The following restrictions apply when programming Blend Moves.

- 1) Each segment of the Blend Move must be written to the SD4840E2 before the move can be initiated.
  - ▶ The SD4840E2 supports Blend Moves with up to sixteen segments.
- 2) Each segment is programmed as a relative move. Blend Moves cannot be programmed with absolute coordinates.
- 3) All segments run in the same direction. The sign of the target position is ignored and only the magnitude of the target position is used. The move's direction is controlled by the bit pattern used to start the move. If you want to reverse direction during your move, consider using the *Dwell Move* which is explained starting on page 31.
- 4) The Programmed Speed of each segment must be greater than or equal to the Starting Speed.
- 5) The Programmed Speed can be the same between segments. This allows you to chain two segments together.
- 6) For all segments except for the last one, the programmed position defines the end of the segment. For the last segment, the programmed position defines the end of the move.
- 7) Once you enter a segment, that segment's programmed acceleration and deceleration values are used to change the speed of the motor.
- 8) The blend segment must be long enough for the acceleration or deceleration portions of the segment to occur. If the segment is not long enough, the motor speed will jump to the speed of the next segment without acceleration or deceleration.

The figure below shows a three segment Blend Move that is run twice. It is first run in the clockwise direction, and then in the counter-clockwise direction.

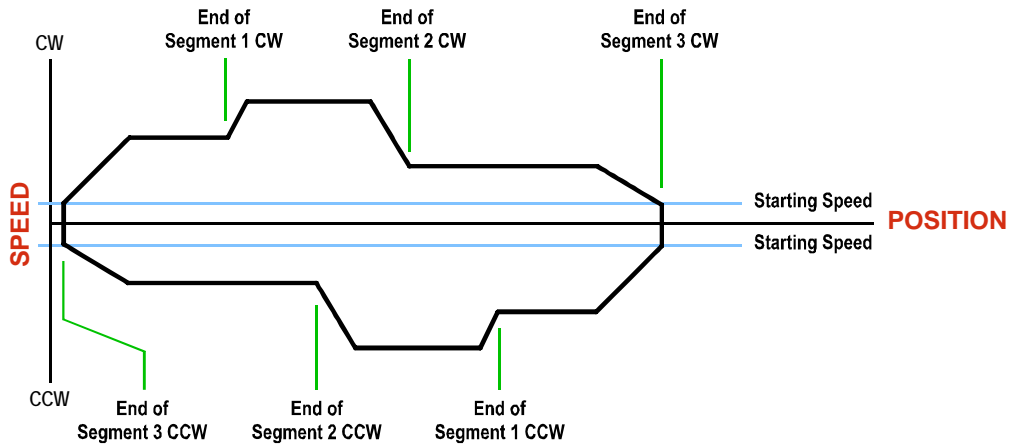


Figure R2.12 Blend Move

**NOTE**

- 1) You do not have to preset the position or home the machine before you can use a Blend Move. Because the Blend Move is based on Relative Moves, it can be run from any location.
- 2) The Blend Move is stored in the internal memory of the SD4840E2 and can be run multiple times once it is written to the unit. The Blend Move data stays in memory until power is removed, the unit is sent new Configuration Data, or a new Blend or Dwell Move is written to the SD4840E2. As described in *Assembled Move Programming* on page 33, it is also possible to save a Blend Move to flash memory. This move is restored on power up and can be run as soon as you configure the SD4840E2 and enter Command Mode.
- 3) There are two control bits used to specify which direction the Blend Move is run in. This gives you the ability to run the Blend Move in either direction.

## Assembled Moves (continued)

### Blend Moves (continued)

#### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. When this occurs, the SD4840E2 decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. The use of the Hold Move bit is explained in the *Controlling Moves In Progress* section starting on page 36.

#### Immediate Stop Conditions

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached. If the limit that is reached is the same as the direction of travel, for example, hitting the CW limit while running a CW move, a *Reset Errors* command must be issued before moves are allowed in that direction again. If the limit that is reached is opposite the direction of travel, a *Reset Errors* command does not have to be issued.

### Dwell Move

A Dwell Move gives you the ability to string multiple relative moves together and run all of them sequentially with a single start condition. Like a Blend Move, a Dwell Move is programmed into the SD4840E2 as a series of relative moves before the move is started.

Unlike a Blend Move, the motor is stopped between each segment of the Dwell Move for a programmed *Dwell Time*. The Dwell Time is programmed with the command that starts the Dwell Time. The same Dwell Time is used between segments. Because the motor is stopped between segments, the motor direction can be reversed during the move. The sign of the target position for the segment determines the direction of motion for that segment. Positive segments will result in clockwise moves while a negative segment will result in a counter-clockwise move. The following figure shows a drilling profile that enters the part in stages and reverses direction during the drilling operation so chips can be relieved from the bit.

You can accomplish this Dwell Move with a series of six relative moves that are sent down to the SD4840E2 sequentially. The two advantages of a Dwell Move in this case is that the SD4840E2 will be more accurate with the Dwell Time than you could accomplish in the PLC, and Dwell Moves simplifies your program's logic.

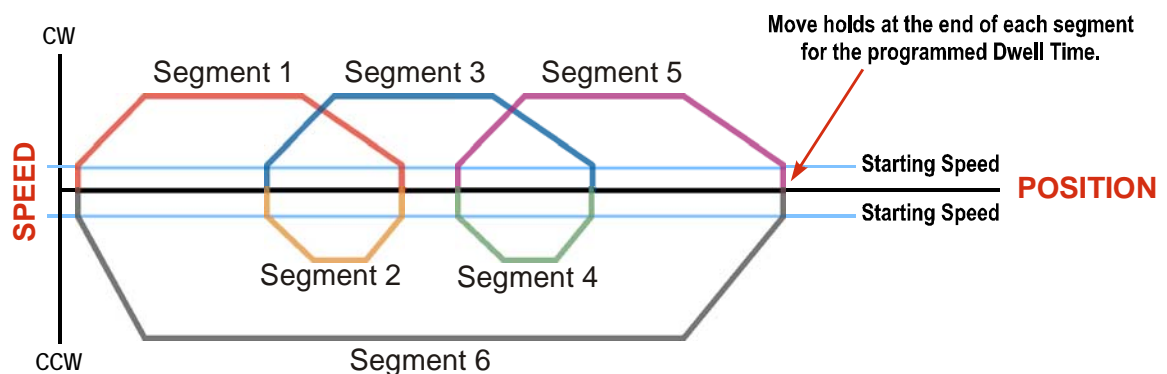


Figure R2.13 Dwell Move

**Assembled Moves (continued)****Dwell Move (continued)****NOTE** 

- 1) You do not have to preset the position or home the machine before you can use a Dwell Move. Because the Dwell Move is based on Relative Moves, it can be run from any location.
- 2) The Dwell Move is stored in the internal memory of the SD4840E2 and can be run multiple times once it is written to the unit. The Dwell Move data stays in memory until power is removed, the unit is sent new Configuration Data, or a new Blend or Dwell Move is written to the SD4840E2. As described in *Assembled Move Programming* on page 33, it is also possible to save a Dwell Move to flash memory. This move is restored on power up and can be run as soon as you configure the SD4840E2 and enter Command Mode.

**Controlled Stop Conditions**

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. When this occurs, the SD4840E2 decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. The use of the Hold Move bit is explained in the *Controlling Moves In Progress* section starting on page 36.

**Immediate Stop Conditions**

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW/CWW Limit Switch is reached. If the limit that is reached is the same as the direction of travel, for example, hitting the CW limit while running a CW move, a *Reset Errors* command must be issued before moves are allowed in that direction again. If the limit that is reached is opposite the direction of travel, a *Reset Errors* command does not have to be issued.



## Assembled Move Programming

All of the segments in a Blend or Dwell Move must be written to the SD4840E2 before the move can be run. Segment programming is controlled with two bits in the Network Output Data and two bits in the Network Input Data. Blend and Dwell Moves are programmed in exactly the same way. When you start the move, a bit in the command data determines which type of Assembled Move is run. In the case of a Blend Move, the sign of each segment's Target Position is ignored and all segments are run in the same direction. In the case of a Dwell Move, the sign of each segment's Target Position determines the direction of the segment. For Dwell Moves, the Dwell Time is sent to the SD4840E2 as part of the command.

### Control Bits – Output Data

- **Program\_Assembled bit** – A 0→1 transition on this bit tells the SD4840E2 that you want to program a Blend or Dwell Move Profile. The SD4840E2 will respond by setting the *In\_Assembled\_Mode* bit in the Network Input Data. At the beginning of the programming cycle, the SD4840E2 will also set the *Waiting\_For\_Assembled\_Segment* bit to signify that it is ready for the first segment.
- **Read\_Assembled\_Data bit** – A 0→1 transition on this bit tells the SD4840E2 that the data for the next segment is available in the remaining data words.

### Control Bits – Input Data

- **In\_Assembled\_Mode bit** – The SD4840E2 sets this bit to tell you that it is ready to accept segment programming data in the remaining output data words. The actual transfer of segment data is controlled by the *Waiting\_For\_Assembled\_Segment* and *Read\_Assembled\_Data* bits.
- **Waiting\_For\_Assembled\_Segment bit** – A 0→1 transition on this bit from the SD4840E2 is the signal to the host that the SD4840E2 is ready to accept the data for the next segment.

### Programming Routine

- 1) The host sets the *Program\_Assembled* bit in the Network Output Data.
- 2) The SD4840E2 responds by setting both the *In\_Assembled\_Mode* and *Waiting\_For\_Assembled\_Segment* bits in the Network Input Data.
- 3) When the host detects that the *Waiting\_For\_Assembled\_Segment* bit is set, it writes the data for the first segment in the Network Output Data and sets the *Read\_Assembled\_Data* bit.
- 4) The SD4840E2 checks the data, and when finished, resets the *Waiting\_For\_Assembled\_Segment* bit. If an error is detected, it also sets the *Command\_Error* bit.
- 5) When the host detects that the *Waiting\_For\_Assembled\_Segment* bit is reset, it resets the *Read\_Assembled\_Data* bit.
- 6) The SD4840E2 detects that the *Read\_Assembled\_Data* bit is reset, and sets the *Waiting\_For\_Assembled\_Segment* bit to signal that it is ready to accept data for the next segment.
- 7) Steps 3 to 6 are repeated for the remaining segments until the entire move profile has been entered. The maximum number of segments per profile is sixteen.
- 8) After the last segment has been transferred, the host exits Assembled Move Programming Mode by resetting the *Program\_Assembled* bit.
- 9) The SD4840E2 resets the *In\_Assembled\_Mode* and the *Waiting\_For\_Assembled\_Segment* bits.

### Saving an Assembled Move in Flash

The SD4840E2 also contains the *Save\_Assembled\_to\_Flash* bit that allows you to store the Assembled Move in flash memory. This allows you to run the Assembled Move right after power up, without having to go through a programming sequence first. To use this bit, you follow the above programming routine with the *Save\_Assembled\_to\_Flash* bit set. When you reach step 9 in the sequence, the SD4840E2 responds by resetting the *In\_Assembled\_Mode* and *Transmit Blend Move Segments* bits as usual and then it will flash the Status LED. If the LED is flashing green, the write to flash memory was successful. If it flashes red, then there was an error in writing the data. In either case, power must be cycled to the SD4840E2 before you can continue. With a limit of 10,000 write cycles, the design decision that requires you to cycle power to the SD4840E2 was made to prevent an application from damaging the module by continuously writing to it.

### ***Indexed Moves***

All of the moves that have been explained in the chapter up to this point can be started by a transition on one of the three inputs instead of a command from the network. If the *Indexed Move* bit is set when the command is issued, the SD4840E2 will not run the move until the configured input makes an inactive-to-active transition. This allows you to run time critical moves that cannot be reliably started from the network because of messaging time delays.

- ▶ The input must be configured as a *Start Indexed Move Input*.
- ▶ The move begins with an inactive-to-active transition on the input. Note that an active-to-inactive transition on the input will not stop the move.
- ▶ The move command must stay in the Network Output Data while performing an Indexed Move. The move will not occur if you reset the command word before the input triggers the move.
- ▶ The move can be run multiple times as long as the move command data remains unchanged in the Network Output Data. The move will run on every inactive-to-active transition on the physical input if a move is not currently in progress. Once a move is triggered, the Start Indexed Move Input is ignored by the SD4840E2 until the triggered move is finished.
- ▶ As started above, a move can be run multiple times as long as the move command data remains unchanged. If you wish to program a second move and run it as an Indexed Move type, then you must have a 0→1 transition on the move command bit before the new parameters are accepted. The easiest way to accomplish this is by writing a value of 0x0000 to the command word between issuing move commands.
- ▶ A Jog Move that is started as an Indexed Move will come to a controlled stop when the command bit in the Network Output Data is reset to zero.
- ▶ It is possible to perform an Indexed Registration Move by configuring two inputs for their respective functions. The first input, configured as a *Start Indexed Move Input*, starts the move and the second, configured as a *Stop Jog or Registration Move Input* causes the registration function to occur.
- ▶ You cannot issue a Hold Command with the Indexed Bit set and have the Hold Command trigger on the inactive-to-active transition of a physical input. Hold Commands are always acted upon as soon as they are accepted from the Network Output Data.
- ▶ You cannot issue an Immediate Stop Command with the Indexed Bit set and have the Immediate Stop Command trigger on the inactive-to-active transition of a physical input. Immediate Stop Commands are always acted upon as soon as they are accepted from the Network Output Data. If you need this functionality, consider programming the physical input as an E-Stop Input.
- ▶ You cannot issue a Clear Error Command with the Indexed Bit set and have the Clear Error Command trigger on the inactive-to-active transition of a physical input. Clear Error Commands are always acted upon as soon as they are accepted from the Network Output Data.


## Synchrostep (Virtual Axis) Moves


On controllers that support motion axis programming, such as the Rockwell Automation ControlLogix platforms, an SD4840E2 can be tied to the motion axis, with the host controller periodically sending position and velocity data to the unit as part of the axis update. The loop is closed by the SD4840E2 by controlling the velocity of the motor. Both linear and circular axes are supported.


Linear and Circular Synchrostep Moves have the following parameters and characteristics:


- Position and Velocity are programmed as 32 bit double integer values.
- The loop can be closed with respect to the internal motor position.
- The loop can be closed with respect to the encoder position on units that use an encoder.
- A proportional constant is available to control the sharpness of the control.  
AMCI suggests a value of 1 or 2 when using the motor position to control the loop.  
AMCI suggests a value between 10 and 50 when using the encoder position to control the loop.
- A network delay value that can be used to offset some of the network delays incurred when communicating with the SD4840E2. Programmed in milliseconds, using this value is optional and defaults to zero. If used, a good starting point is the update time of the connection in milliseconds. The range of this parameter is 0 to 20.

Circular Synchrostep Moves have one additional parameter. This parameter is often called the “Position Unwind” value, and it defines the point at which the position rolls over and returns to zero. On the SD4840E2 units, this parameter has a range of 21 to 65,535.

**NOTE**  When using the SD4840E2 as an axis follower, it is best to run the virtual axis as a high priority event driven periodic task.

**NOTE**  When using the SD4840E2 as an axis follower, it is best to configure the unit to have a starting speed of 1. This will reduce jitter in the motor position at slow speeds or when the position is near its target.

**NOTE**  When using the SD4840E2 as an axis follower, the programmed acceleration and deceleration values determine the response time of the internal closed loop algorithm.  
A suggested starting value for the Acceleration and Deceleration Parameters is Motor Resolution / 10.

**NOTE**  When using the SD4840E2 as a linear axis follower, use caution in systems that always rotate in the same direction. If the position value overflows, it will switch from the maximum positive value to the maximum negative value and unpredictable motion may occur.

Assembled Moves offer you similar functionality to linear motion axes when the move profile is well defined before the move begins. Consider using this functionality for simple linear profiles. An Assembled Move is programmed into an SD4840E2 before it is run and the unit precisely controls position and velocity throughout the entire move. Additional information on *Assembled Moves* functionality can be found starting on page 29.

### Controlling Moves In Progress

The SD4840E2 has the ability to place a running move on hold and later resume the move if an error did not occur while the move was in its Hold state. One potential application for this feature is bringing a move to a controlled stop when your controller senses an end-of-stock condition. The move can be put in its Hold state until the stock is replenished and then the move can be resumed.

Note that you do not have to resume a move once it has been placed in its Hold state. You can place a move in its Hold state to prematurely end the move with a controlled stop and issue any type of new move from the stopped position.

The figure below shows a profile of a move that is placed in its Hold state and later resumed.

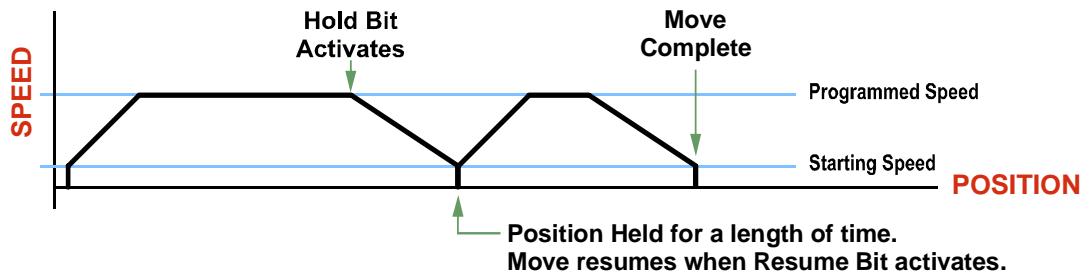


Figure R2.14 Hold/Resume a Move Profile

### Find Home Moves

A Find Home command can be placed in a Hold state but cannot be resumed. This gives you the ability to bring a Find Home command to a controlled stop if an error condition occurs.

### Jog Moves

Jog Moves can be placed in a Hold state and resumed if error conditions, such as programming errors, have not occurred. New Acceleration, Deceleration, and Programmed Speed parameters can be written to the SD4840E2 while a Jog Move is in its hold state. If these parameters are accepted without error, the move can be resumed and it will use the new parameter values.

### Registration Moves

Registration Moves can be brought to a controlled stop with the Hold bit, but they cannot be restarted.

### Absolute, Relative and Encoder Registration Moves

Absolute, Relative and Encoder Registration Moves can be placed in a Hold state and resumed if error conditions, such as programming errors, have not occurred. New Acceleration, Deceleration, and Programmed Speed parameters can be written to the SD4840E2 while these moves are in their hold states. If the parameters are accepted without error, the move can be resumed and it will use the new parameter values. Note that a change to the Target Position is ignored.

### Assembled Moves

A Blend or Dwell Move can be placed in a Hold state but cannot be resumed. This gives you the ability to prematurely end an Assembled Move with a controlled stop. The Assembled Move is not erased from memory and can be run again without having to reprogram it.

## Electronic Gearing

The final form of motion control available with the SD4840E2 is Electronic Gearing. A quadrature encoder is required but it is not mounted on the motor controlled by the SD4840E2. The encoder is typically mounted on a second motor, but it can be mounted anywhere, including on something as simple as a hand crank.

This mode is sometimes referred to as *encoder following*, because the motor will change position in response to a change in position of the encoder. AMCI refers to it as Electronic Gearing because the SD4840E2 has three parameters that allow you to set any turns ratio you want between the encoder and the motor.

### Motor Steps/Turn

This is the same parameter explained at the beginning of this chapter. In Electronic Gearing mode, this parameter sets the number of encoder counts required to complete one rotation of the shaft of the motor driven by the SD4840E2. It has a range of 200 to 32,767. This parameter is programmed when you configure the module and cannot be adjusted while a move is in progress.

### ELGearing Multiplier and Divisor

The ratio of these two parameters is applied to the number of encoder pulses read by the SD4840E2 before it determines the number of motor steps to move. Each parameter has a range of 1 to 255. These two parameters can be adjusted while a move is in progress which allows you to adjust the tracking speed and position of the motor.

### How It Works

The SD4840E2 always uses 4X decoding when counting pulses from the encoder. If you set both of your ELGearing Multiplier and Divisors to 1 and set the Motor Steps/Turn to four times the number of encoder lines, then the motor will complete one rotation for every rotation of the encoder's shaft.

Once placed in Electronic Gearing mode, the SD4840E2 monitors the Jog Move command bits in the Network Output Registers. When either of these bits is set, the encoder inputs are monitored for a change in position. When a change is sensed, the SD4840E2 will begin to turn the motor within 50 microseconds. An increase in encoder counts will result in clockwise rotation. A decrease in encoder counts will result in counter-clockwise rotation.

The values of the ELGearing Multiplier and Divisor can be changed while electronic gearing motion is occurring. The SD4840E2 will accelerate or decelerate the motor to match the new ratio.

Encoder position data can be trapped while in Electronic Gearing mode by configuring one of the discrete DC input as a Capture Encoder Position input.

### Controlled Stop Conditions

- The encoder stops moving.
- Both of the Jog Move command bits equal zero.
- Electronic Gearing moves cannot be brought to a controlled stop by using the Hold Move control bit in the Network Output Registers.

### Immediate Stop Conditions

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Registers.
- A positive transition on an input configured as an E-Stop Input.
- A CW or CWW Limit Switch is reached.

## Electronic Gearing (continued)

### Advanced Ratio Control

The ELGearing Multiplier and Divisor values give you a great deal of control over the ratio of motor turns per encoder turn, but you can achieve even finer control by adjusting the Motor Steps/Turn parameter.

The Z pulse is not used to correct the encoder position once per turn, so you can actually program the Motor Steps/Turn to any value you want within its valid range. For example, if your encoder outputs 4,096 pulse per turn (a 1,024 line encoder) and you set the Motor Steps/Turn parameter to 8,192, you will have built a 2:1 gear down into your system before applying the ELGearing Multiplier and Divisors. (Two rotations of the encoder = 8,192 counts = 1 motor rotation.)

This technique allows you to set a median gear ratio in your system that you can adjust on-the-fly by using the ELGearing Multiplier and Divisor parameters.

### Hybrid Control

Hybrid Control gives you servo like performance from your stepper motor without the need to tune the servo PID loop parameters. This feature is only available when using an encoder that is attached to the motor.

In order for the Hybrid Control feature to work correctly:

- The *Encoder Pulses/Turn* parameter must be set to four times the number of encoder lines. (When using a 1,024 line encoder, the *Encoder Pulses/Turn* parameter must equal 4,096.)
- The *Motor Steps/Turn* parameter must also be set equal to the *Encoder Pulses/Turn* value.

The SD4840E2 uses the encoder feedback during a move to dynamically increase or decrease the current through the motor's windings to the lowest values needed to achieve motion. This feature eliminates the possibility of resonance induced motor stalls and can dramatically lower the motor's operating temperature.

Like a servo system, there is the possibility of a one count dither in position at zero speed and a load induced error in the final position if this feature is always enabled. The "stiffness" of the shaft, (how quickly the system compensates for shaft position error), can be adjusted with a single gain setting. The SD4840E2 also allows you to enable and disable the Hybrid Control feature on the fly. When the motor is at rest, a single command will disable the Hybrid Control feature, and the motor will again act as a stepper system, holding its position without dither. You can then re-enable the Hybrid Control feature using the same command that initiates the next move, minimizing the amount of motor current needed to complete the move.

When using Hybrid Control, the SD4840E2 has an additional status bit, the *Motion Lag* bit, that turns on if the position error becomes larger than 360°. This status bit indicates that the speed of the move is not the one that is expected, but the move is still in progress. This can be an indication that something has changed mechanically in the system that has placed an additional load on the motor. Using this bit is the preferred method of performing stall detection when using Hybrid Control. When this bit turns on, monitor the encoder position value for change. If the encoder value is changing, then the motor is heavily loaded but still moving. If the encoder value is not changing, then the motor cannot rotate under load and the move can be stopped by your program.

When using Hybrid Control, you should always check the value of the encoder position at the end of every move and make accommodations for any load induced error in the final position.

The stall detection feature that is built into the SD4840E2 can be used with the Hybrid Control feature. If the rotor position becomes more than 45° behind the commanded position, the motor will be considered stalled and the move will be terminated. Error bits will be set in the network data to inform you of the stalled condition. Note that the *Motion Lag* bit will never be set when using the stall detection feature of the SD4840E2 because it will trip before the 360° position error that will set the *Motion Lag* bit.



**Current Loop Gain**

This feature gives you the ability to adjust the gain of the power amplifiers in the Networked Driver to match the electrical characteristics of your motor. The value of this parameter can range from 1 to 40 with 40 representing the largest gain increase. In general, using a larger gain will increase high speed torque but the motor will run louder. A lower gain will offer quieter low speed operation at the cost of some high speed torque.

The use of this feature is completely optional and you can leave the Current Loop Gain at its default setting of "1" for standard motor performance. AMCI motors can use a setting of "5" for normal operation.

***Notes***

## REFERENCE 3

# CALCULATING MOVE PROFILES

This reference was added for customers that must program very precise profiles. Understanding this section is not necessary before programming the SD4840E2 and therefore can be considered optional. Two different approaches are presented here. The constant acceleration example takes given parameters and calculates the resulting profile. The variable acceleration example starts with a desired speed profile and calculates the required parameters.

The equations in this reference use a unit of measure of steps/second/second ( $\text{steps/second}^2$ ) for acceleration and deceleration. However, when programming the SD4840E2, all acceleration and deceleration values must be programmed in the unit of measure of steps/second/millisecond.

- To convert from  $\text{steps/second}^2$  to  $\text{steps/second/millisecond}$ , divide the value by 1000. This must be done when converting from a value used in the equations to a value programmed into the SD4840E2.
- To convert from  $\text{steps/second/millisecond}$  to  $\text{steps/second}^2$ , multiply the value by 1000. This must be done when converting from the value programmed into the SD4840E2 to the value used in the equations.

### Constant Acceleration Equations

When you choose to use constant accelerations, the speed of the move will increase linearly towards the Programmed Speed. This is the fastest form of acceleration, resulting in the fastest move between two points at its programmed speed. For the smoothest transition from the starting speed, the starting speed should be equal to the square root of the acceleration in  $\text{steps/sec}^2$ . For example, if the choose acceleration is  $20,000 \text{ steps/sec}^2$ , the smoothest transition occurs when the starting speed is 141. ( $141^2 \approx 20,000$ )

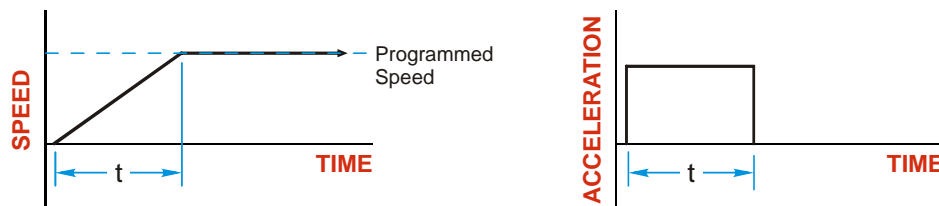
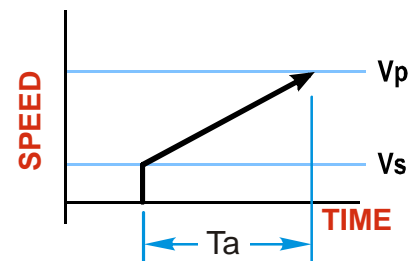


Figure R3.1 Constant Acceleration Curves

### Variable Definitions

The following variables are used in these equations:

- $V_S$  = Configured Starting Speed of the move
- $V_P$  = Programmed Speed of the move
- $a$  = Acceleration value. Must be in the units of  $\text{steps/second}^2$
- $d$  = Deceleration value. Must be in the units of  $\text{steps/second}^2$
- $T_A$  or  $T_D$  = Time needed to complete the acceleration or deceleration phase of the move
- $D_A$  or  $D_D$  = Number of Steps needed to complete the acceleration or deceleration phase of the move



**Constant Acceleration Equations (continued)**

Figure R3.1 gives the equations to calculate Time, Distance, and Acceleration values for a constant acceleration move.

Acceleration Type	T <sub>A</sub> or T <sub>D</sub> (Time to Accelerate or Decelerate)	D <sub>A</sub> or D <sub>D</sub> (Distance to Accelerate or Decelerate)	a (Average Acceleration)
Linear	$T_A = (V_P - V_S)/a$	$D_A = T_A * (V_P + V_S)/2$	$a = (V_P^2 - V_S^2)/2D_A$

Table R3.1 Acceleration Equations

If the sum of the D<sub>A</sub> and D<sub>D</sub> values of the move is *less than* the total number of steps in the move, your move will have a Trapezoidal profile.

If the sum of the D<sub>A</sub> and D<sub>D</sub> values of the move is *equal to* the total number of steps in the move, your move will have a Triangular profile and your move will reach the Programmed Speed before it begins to decelerate.

If the sum of the D<sub>A</sub> and D<sub>D</sub> values of the move is *greater than* the total number of steps in the move, your move will have a Triangular profile and it *will not* reach the Programmed Speed before it begins to decelerate.

As an example, lets assume the values in table R3.2 for a move profile.

Name	Value	SD4840E2 Parameter Values
Acceleration (a)	20,000 steps/sec <sup>2</sup>	20
Deceleration (d)	25,000 steps/sec <sup>2</sup>	25
Starting Speed (V <sub>S</sub> )	141 steps/sec	141
Programmed Speed (V <sub>P</sub> )	100,000 steps/sec	100,000

Table R3.2 Sample Values

From figure R3.1:

Time to accelerate:  $T_A = (V_P - V_S)/a = (100,000 - 141)/20,000 = 4.993$  seconds  
 Time to decelerate:  $T_D = (V_P - V_S)/d = (100,000 - 141)/25,000 = 3.994$  seconds  
 Distance to Accelerate:  $D_A = T_A * (V_P + V_S)/2 = 4.993 * (100,000 + 141)/2 = 250,002$  steps  
 Distance to Decelerate:  $D_D = T_D * (V_P + V_S)/2 = 3.994 * (100,000 + 141)/2 = 199,982$  steps

Total Distance needed to accelerate and decelerate:  $250,002 + 199,982 = 449,984$  steps

If a move with the above acceleration, deceleration, starting speed, and programmed speed has a length greater than 449,984 steps, the SD4840E2 will generate a Trapezoidal profile. If the move is equal to 449,984 steps, the unit will generate a Triangular profile and the it will output one pulse at the programmed speed. If the move is less than 449,984 steps, the unit will generate a Triangular profile and the programmed speed will not be reached.

In the case of a Triangular profile where the programmed speed is not reached, it is fairly easy to calculate the maximum speed (V<sub>M</sub>) attained during the move. Because the move is always accelerating or decelerating, the total distance traveled is equal to the sum of D<sub>A</sub> and D<sub>D</sub>.

$D_A = T_A * (V_M + V_S)/2$  and  $T_A = (V_M - V_S)/a$ . By substitution:  
 $D_A = (V_M - V_S)/a * (V_M + V_S)/2 = (V_M^2 - V_S^2)/2a$ . By the same method,  
 $D_D = (V_M^2 - V_S^2)/2d$ .

Therefore, total distance traveled =

$D_A + D_D = (V_M^2 - V_S^2)/2a + (V_M^2 - V_S^2)/2d$ .

In the case where the acceleration and deceleration values are equal, this formula reduces to:

$D_A + D_D = (V_M^2 - V_S^2)/a$

**Constant Acceleration Equations (continued)**

Continuing the example from table R3.2, assume a total travel distance of 300,000 steps.

$$D_A + D_D = \frac{V_M^2 - V_S^2}{2a} + \frac{V_M^2 - V_S^2}{2d}$$

$$300,000 \text{ steps} = \frac{V_M^2 - 141^2}{2(20,000)} + \frac{V_M^2 - 141^2}{2(25,000)}$$

$$300,000 \text{ steps} = \frac{V_M^2 - 20,000}{40,000} + \frac{V_M^2 - 20,000}{50,000}$$

$$300,000 \text{ steps} = \frac{5(V_M^2 - 20,000)}{5(40,000)} + \frac{4(V_M^2 - 20,000)}{4(50,000)}$$

$$300,000 \text{ steps} = \frac{5V_M^2 - 100,000}{200,000} + \frac{4V_M^2 - 80,000}{200,000}$$

$$300,000(200,000) = 9V_M^2 - 180,000$$

$$\frac{60,000.18 \times 10^6}{9} = V_M^2$$

$$V_M = 81,650 \text{ steps/sec}$$

Once you have calculated the maximum speed, you can substitute this value into the time and distance formulas in table R3.1 to calculate time spent and distance traveled while accelerating and decelerating.

**Total Time Equations**

For Trapezoidal Profiles you must first determine the number of counts that you are running at the Programmed Speed. This value, ( $D_P$  below), is equal to your  $D_A$  and  $D_D$  values subtracted from your total travel. You can then calculate your total profile time, ( $T_P$  below), from the second equation.

$$D_P = (\text{Total Number of Steps}) - (D_A + D_D)$$

$$T_P = T_A + T_D + D_P/V_P$$

For Triangular Profiles, the total time of travel is simply:

$$T_P = T_A + T_D$$

**S-Curve Acceleration Equations**

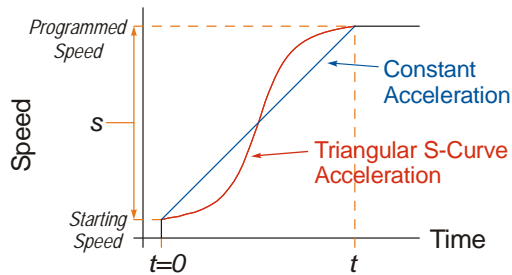
When the Acceleration Jerk parameter value is in the range of 1 to 5,000, the SD4840E2 uses this value to smoothly change the acceleration value applied during the move. In this case, the speed of the move does not increase linearly, but exponentially, resulting in an “S” shaped curve. This limits mechanical shocks to the system as the load accelerates. Just as constant acceleration will result in a trapezoidal or triangular speed profile, the Acceleration Jerk parameter will result in a trapezoidal or triangular acceleration phase.

In order to keep the Acceleration Jerk parameter value that is programmed into the SD4840E2 below sixteen bits, the Acceleration Jerk parameter programmed into the driver does not have units of steps/sec<sup>3</sup>. The Acceleration Jerk parameter equals  $(\{100 * \text{jerk in steps/sec}^3\} / \text{acceleration in steps/sec}^2)$ . This translates to the jerk property in steps/sec<sup>3</sup> equalling  $(\{\text{Acceleration Jerk parameter}/100\} * \text{acceleration in steps/sec}^2)$ . With the range of values for the Acceleration Jerk parameter being 1 to 5,000, the jerk value ranges from 0.01a to 50a where “a” is the acceleration value in steps/sec<sup>2</sup>. For example, if the acceleration is programmed to 20,000 steps/sec<sup>2</sup>, then the value of the jerk property used by the unit can be programmed to be between 200 steps/sec<sup>3</sup> (0.01\*20,000) and 1,000,000 steps/sec<sup>3</sup> (50\*20,000). This statement applies to the Deceleration Parameter as well. If the Acceleration and Deceleration parameters are different, the calculated jerk values will also differ.

When using variable accelerations, the starting speed does not have to be equal to the square root of the programmed acceleration value. Variable acceleration provides smooth transitions at the beginning and end of the acceleration phase.

**Triangular S-Curve Acceleration**

Figure R3.2 shows the speed profile of a move during its acceleration phase. The figure shows the desired triangular S-curve acceleration in red along with the equivalent constant acceleration in blue. The equivalent constant acceleration is equal to the change in speed divided by the time it takes to achieve this change in speed. This is the value that would have to be used if the Jerk parameter was left at zero and we will use this information to calculate the S-curve acceleration and the value of the Jerk Parameter.



$$s = \text{Programmed Speed} - \text{Starting Speed}$$

$$\text{Acceleration} = \frac{\text{speed}}{\text{time}}$$

$$a = \frac{s}{t}$$

$$at = s$$

$$\text{jerk} = \frac{\text{acceleration}}{\text{time}}$$

$$j = \frac{a}{t}$$

$$jt = a$$

SD4840E2's  
Acceleration  
Jerk Parameter(J) =  $\frac{100j}{a}$

$$j = \frac{Ja}{100}$$

Figure R3.2 Move Profile Example

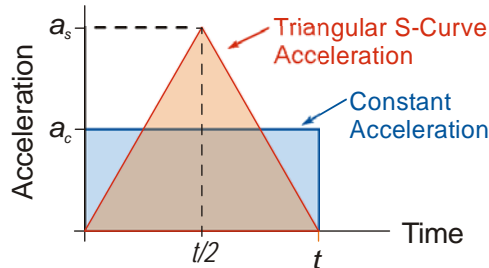


Figure R3.3 Triangular Acceleration

Speed is equal to acceleration multiplied by the time it is applied. This is shown graphically in figure R3.3 as the area of the blue rectangle. In order for the Triangular S-curve acceleration to reach the same speed in the same amount of time, the area of the triangle must equal the area of the square. Area of a triangle is one half of the base length multiplied by the height. Therefore:

$$a_c t = \frac{a_s t}{2} \text{ Area of rectangle} = \text{Area of triangle}$$

$$a_s = 2a_c$$

This means that a triangular S-curve acceleration profile requires twice the programmed maximum acceleration as a constant acceleration profile to achieve the same speed in the same amount of time.



**S-Curve Acceleration Equations (continued)****Triangular S-Curve Acceleration (continued)**

The value of the Acceleration Jerk parameter can now be easily calculated.

$$j = \frac{a_s}{t/2} \quad (j = a/t)$$

$$j = \frac{2a_s}{t}$$

$$\frac{Ja_s}{100} = \frac{2a_s}{t} \quad \left( j = \frac{Ja}{100} \right)$$

$$Ja_s t = 200a_s$$

$$J = \frac{200}{t} \quad \text{Acceleration Jerk parameter} = 200 / \text{acceleration time}$$

This value represents the ideal Acceleration Jerk parameter value for a triangular S-curve acceleration. Setting the value lower than this will result in a longer acceleration period, while setting the value above this will result in a trapezoidal S-curve acceleration.

*When  $a_s = a_c$*

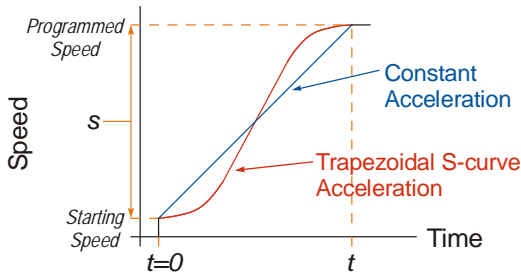
The above examples assume that you can increase the programmed acceleration value to keep the acceleration time the same. If your constant acceleration value is the maximum your system will allow, then using S-curve accelerations will lengthen the time needed to accelerate to your desired speed.

In the case of Triangular S-curve accelerations where the Acceleration Jerk parameter is optimized at  $200/t$ , the value of "t" must be twice that of the acceleration period when constant acceleration is used. For example, assume a equivalent constant acceleration of 20,000 steps/sec<sup>2</sup> that is applied for 2.0 seconds. If the acceleration value must remain at 20,000 steps/sec<sup>2</sup>, then the acceleration phase will take 4.0 seconds and the Acceleration Jerk parameter should be set to 50 (200/4.0)

**S-Curve Acceleration Equations (continued)**

**Trapezoidal S-Curve Acceleration**

Figure R3.4 shows the speed profile of a move during its acceleration phase. The figure shows the desired trapezoidal S-curve acceleration in red along with the equivalent constant acceleration in blue. The equivalent constant acceleration is equal to the change in speed divided by the time it takes to achieve the change in speed. This is the value that would have to be used if the Acceleration Jerk parameter was left at zero and we will use this information to calculate the S-curve acceleration and the value of the Acceleration Jerk Parameter.



$$S = \text{Programmed Speed} - \text{Starting Speed}$$

$$\text{Acceleration} = \frac{\text{speed}}{\text{time}}$$

$$\text{jerk} = \frac{\text{acceleration}}{\text{time}}$$

$$\text{SD4840E2's Acceleration Jerk Parameter (J)} = \frac{100j}{a}$$

$$a = \frac{S}{t}$$

$$at = S$$

$$j = \frac{a}{t}$$

$$jt = a$$

$$\Rightarrow j = \frac{Ja}{100}$$

Figure R3.4 Move Profile Example

In this example, the period of constant acceleration is 50% of the acceleration phase.

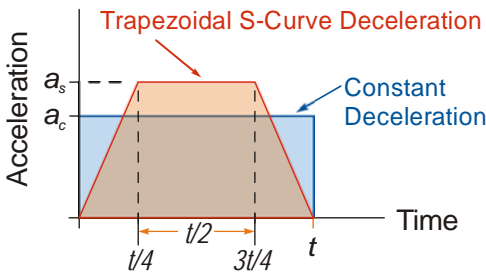


Figure R3.5 Trapezoidal Acceleration

Speed is equal to acceleration multiplied by the time it is applied. This is shown graphically in figure R3.5 as the area of the blue rectangle. In order for the Trapezoidal S-curve acceleration to reach the same speed in the same amount of time, the area of the polygon must equal the area of the rectangle.

$$\frac{a_s t}{2} + \frac{a_s t}{4} = a_c t \quad \text{Area of polygon} = \text{Area of rectangle}$$

$$\frac{2a_s t}{4} + \frac{a_s t}{4} = a_c t$$

$$\frac{3a_s t}{4} = a_c t$$

$$a_s = \frac{4}{3}a_c$$

This means that a trapezoidal S-curve acceleration profile that has a period of constant acceleration equal to half of the total phase time, requires its programmed acceleration value to be 4/3 that of the constant acceleration value used to achieve the same speed in the same amount of time.

**S-Curve Acceleration Equations (continued)****Trapezoidal S-Curve Acceleration (continued)**

The value of the Acceleration Jerk parameter can now be easily calculated.

$$j = \frac{a_s}{t/4} \quad (j = a/t)$$

$$j = \frac{4a_s}{t}$$

$$\frac{Ja_s}{100} = \frac{4a_s}{t} \quad \left( j = \frac{Ja}{100} \right)$$

$$Ja_s t = 400a_s$$

$$J = \frac{400}{t} \quad \text{Acceleration Jerk Parameter} = 400 / \text{acceleration time}$$

This value represents the ideal Acceleration Jerk parameter value for a trapezoidal S-curve acceleration with a constant acceleration for half of the phase. Setting the value lower than this will result in a shorter constant period, while setting the value greater than this will result in a longer constant period.

Another example of a trapezoidal S-curve acceleration is when the linear acceleration occurs for one third of the time. In this case, the programmed acceleration must be the constant acceleration value multiplied by 3/2 and the Acceleration Jerk parameter must be set to 300/t.

*When  $a_s = a_c$*

The above examples assume that you can increase the programmed acceleration value to keep the time of the acceleration phase the same. If your constant acceleration value is the maximum your system will allow, then using S-curve accelerations will lengthen the time needed to accelerate to your desired speed.

In the case of trapezoidal S-curve accelerations, calculating the percentage increase in time is shown in figure R3.6. The time added to the acceleration phase is equal to the time spent increasing the acceleration during the phase. As shown in the figure, when the Trapezoidal S-curve is programmed to spend 50% of its time at the programmed acceleration value, the time spent in the acceleration phase will be 133.33% of the time spent if a constant acceleration were used.

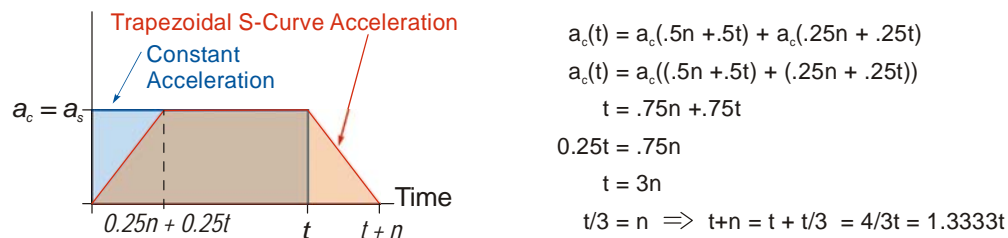


Figure R3.6 Trapezoidal S-curve Time Increase Example

In this case the value of the Acceleration Jerk parameter should be based on the new, longer time. For example, assume an equivalent constant acceleration of 15,000 steps/sec<sup>2</sup> that is applied for 2.0 seconds. If the acceleration value must remain at 15,000 steps/sec<sup>2</sup>, then the acceleration phase will take 2.667 seconds (2.0×1.333) and the Acceleration Jerk parameter should be set to 150 (400/2.667)

Similarly, if the Trapezoidal S-curve acceleration is to spend 33.3% of its time at constant acceleration, and the programmed acceleration value cannot be increased, the time spent accelerating will increase by 50% and the Acceleration Jerk parameter should be adjusted accordingly.

**S-Curve Acceleration Equations (continued)**

**Determining Waveforms by Values**

If your programmed acceleration and deceleration values are the same, then your move's acceleration and decelerations will be identical. If these two programmed values are different, use the above methods to determine the Acceleration Jerk parameter for either the move's acceleration or deceleration phases and use the following calculations to determine the shape of the other phase.

Two examples are given below. Both assume a change in speed between the Starting Speed and Programmed Speed of 30,000 steps/sec and an acceleration of 58,000 steps/sec<sup>2</sup>. The first example uses an Acceleration Jerk parameter value of 20 and the second a value of 400.

Triangular or Trapezoidal S-curve accelerations are always symmetrical. We'll use this fact to calculate the profile up to one-half of the change in speed. At that point, doubling the time and distance will yield the total time and distance traveled.

*Example 1, Jerk = 20*

$$S_m = \frac{30,000 \text{ steps/sec}}{2} = 15,000 \text{ steps/sec} \quad S_m = \text{midpoint of change in speed}$$

$$J = \frac{100j}{a} \Rightarrow j = \frac{Ja}{100} \quad J = \text{Acceleration Jerk parameter}$$

$j$  = physical jerk property

$a_f$  = calculated final acceleration

$$j = \frac{20(58,000 \text{ steps/sec}^2)}{100}$$

$$j = 11,600 \text{ steps/sec}^3$$

**Just as displacement =  $\frac{1}{2}at^2$ , Speed =  $\frac{1}{2}jt^2$**

$$15,000 \text{ steps/sec} = \frac{11,600 \text{ steps/sec}^3(t^2)}{2}$$

$$t^2 = \frac{15,000 \text{ steps/sec}}{5,800 \text{ steps/sec}^3}$$

$$t = 1.608 \text{ seconds}$$

**Just as speed =  $at$ , acceleration =  $jt$**

$$a_f = 11,600 \text{ steps/sec}^3(1.608 \text{ sec})$$

$$a_f = 18,655 \text{ steps/sec}^2$$

Because  $a_f$  is less than or equal to the programmed acceleration of 58,000 steps/sec<sup>2</sup>, the resulting acceleration is a Triangular S-curve. Total time to accelerate is twice the value calculated above, or 3.216 seconds.

**S-Curve Acceleration Equations (continued)****Determining Waveforms by Values (continued)**

Example 2, Jerk = 400

$$S_m = \frac{30,000 \text{ steps/sec}}{2} = 15,000 \text{ steps/sec}$$

$$J = \frac{100j}{a} \Rightarrow j = \frac{Ja}{100}$$

$$j = \frac{400(58,000 \text{ steps/sec}^2)}{100}$$

$$j = 232,000 \text{ steps/sec}^3$$

$$\text{Just as displacement} = \frac{1}{2}at^2, \text{ speed} = \frac{1}{2}jt^2$$

$$15,000 \text{ steps/sec} = \frac{232,000 \text{ steps/sec}^3(t^2)}{2}$$

$$t^2 = \frac{15,000 \text{ steps/sec}}{116,000 \text{ steps/sec}^3}$$

$$t = 0.3596 \text{ seconds}$$

Just as speed = at, acceleration = jt

$$a_f = 232,000 \text{ steps/sec}^3(0.3596 \text{ sec})$$

$$a_f = 83,427 \text{ steps/sec}^2$$

Because  $a_f$  is greater than the programmed acceleration of 58,000 steps/sec<sup>2</sup>, the resulting acceleration is a trapezoidal S-curve. As shown in figure R3.7, two additional calculations must be made. The first is the time ( $t_1$ ) it takes to jerk to the programmed acceleration value. The second is the time ( $t_2$ ) it takes to accelerate to half of the required change in speed ( $S_m$ ).

$$232,000 \text{ steps/sec}^3(t_1) = 58,000 \text{ steps/sec}^2 \quad jt = a$$

$$t_1 = 0.25 \text{ seconds}$$

Determine speed at  $t_1$ : Speed =  $\frac{1}{2}jt^2$

$$S_1 = \frac{232,000 \text{ steps/sec}^3(0.25)^2}{2}$$

$$S_1 = 7,250 \text{ steps/sec}$$

Determine remaining change in speed and required time based on programmed acceleration

$$S_2 = S_m - S_1 = (15,000 - 7,250) \text{ steps/sec}$$

$$S_2 = 7,750 \text{ steps/sec}$$

$$S_2 = a_c(t_2) \Rightarrow t_2 = S_2/a_c$$

$$t_2 = \frac{7,750 \text{ steps/sec}}{58,000 \text{ steps/sec}^2}$$

$$t_2 = 0.1336 \text{ seconds}$$

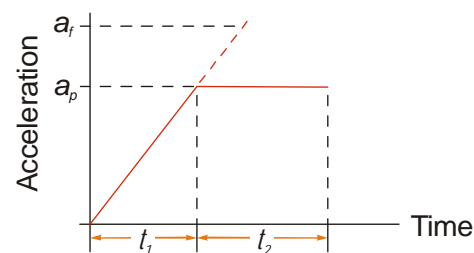


Figure R3.7 Calculating Trapezoidal S-Curve

The time for this acceleration phase is  $2(t_1 + t_2)$ , which equals  $2(0.2500 \text{ sec} + 0.1336 \text{ sec})$  or 0.7672 seconds. Time spent in the constant acceleration period is  $(2(0.1336))/0.7672$  or 34.8% of the entire phase.

***Notes***



## HOMING THE SD4840E2

This chapter explains the various ways of homing an axis on the SD4840E2. Inputs used to home the module are introduced and diagrams that show how the module responds to a homing command are given.

### Definition of Home Position

The Home Position is any position on your machine that you can sense and stop at. Once at the Home Position, the motor position register of the SD4840E2 must be set to an appropriate value. If you use the module's *CW/CCW Find Home* commands, the motor position register will be set to zero once the home position is reached. The Encoder Position register will also be reset to zero if the quadrature encoder is enabled for the axis.



**NOTE** Defining a Home Position is completely optional. Some applications, such as those that use a stepper for speed control, don't require position data at all.

With the exception of Absolute Moves, the SD4840E2 can still perform all of its move commands if the Home Position is not defined.

### Position Preset

One of the ways to define the Home Position is to issue the Preset Position command to the SD4840E2. Before doing this, you will need a way of sensing position outside the SD4840E2 module. The machine position data must be brought into the host, the correct preset value calculated, and this value written to the SD4840E2 axis with the Position Preset command. The motor and encoder position values can be preset anywhere in the range of -8,388,608 to +8,388,607.

### CW/CCW Find Home Commands

The other choice is to use the module's Find Home commands to order the SD4840E2 to find the Home Position based on sensors brought into the unit. The CW Find Home command begins searching by rotating the motor shaft in the clockwise direction, (when the motor is wired as shown in this manual), and ends when the home sensor triggers while the SD4840E2 is rotating the motor in a clockwise direction at a low rate. The CCW Find Home command operates in the same way but starts and ends with CCW rotation.

### Homing Inputs

Five inputs can be used when homing the module. These inputs are either physical inputs attached to the module or bits in the PLC output data words.

#### Physical Inputs

- **Home Input:** This input is used in one of two ways: 1) This input is used to define the actual home position of the machine. 2) The input is used as a home proximity input when using the encoder marker pulse to home the machine.
- **Encoder Marker (Z) Pulse:** If you configure the SD4840E2 to use an encoder, you have the option of using the encoder's marker pulse to home the machine.
- **CW Limit Switch Input:** This input is used to prevent overtravel in the clockwise direction.
- **CCW Limit Switch Input:** This input is used to prevent overtravel in the counter-clockwise direction.


#### Backplane Inputs

- **Network Home Proximity Bit:** The SD4840E2 can be configured to ignore changes on the physical homing input until the Network Home Proximity Bit makes a 0→1 transition. The SD4840E2 will home on the next inactive-to-active change on the physical input once this transition occurs. You must program your host to control the state of this bit. Do not use the Network Home Proximity bit if you only want to home to the Home Limit Switch.


**Homing Configurations**

The SD4840E2 axis must be correctly configured before one of the homing commands will be accepted. One of the following must be part of the module configuration before you can run the homing commands.

- 1) Configure one of the DC inputs as a Home Input
- 2) Configure the SD4840E2 to use an encoder and home to the encoder Z-pulse

- NOTE** 
- 1) You do not have to configure and use CW or CCW Limits. If you choose to configure the module this way, then the SD4840E2 has no way to automatically prevent overtravel during a homing operation. You must prevent overtravel by some external means, or ensure that the homing command is issued in the direction that will result in reaching the homing input directly.
  - 2) When using one of the DC inputs as a Home Input, you can use a bit in the network data as a home proximity input. Using this bit is completely optional.
  - 3) When using an encoder's Z-pulse as the homing sensor, any DC input you configure as a Home Input will function as a hardware home proximity sensor. Using this feature is completely optional.

**Homing Profiles**

- NOTE** 
- The CW Find Home command is used in all of these examples. The CCW Find Home command will generate the same profiles in the opposite direction.

**Home Input Only Profile**

Figure R4.1 below shows the move profile generated by a CW Find Home command when you use the Home Input without the Network Home Proximity bit.

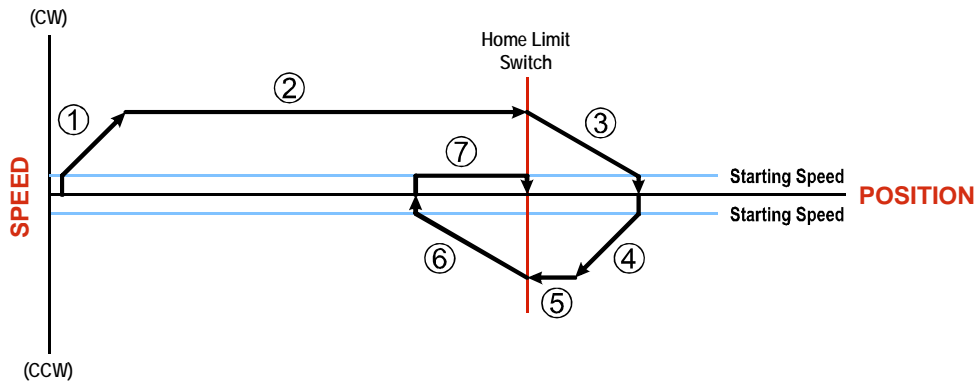



Figure R4.1 Home Input Profile

- 1) Acceleration from the configured Starting Speed to the Programmed Speed
- 2) Run at the Programmed Speed until the Home Input activates
- 3) Deceleration to the Starting Speed and stop, followed by a two second delay.
- 4) Acceleration to the Programmed Speed opposite to the requested direction.
- 5) Run opposite the requested direction until the Home Input transitions from Active to Inactive
- 6) Deceleration to the Starting Speed and stop, followed by a two second delay.
- 7) Return to the Home Input at the configured Starting Speed. Stop when the Home Input transitions from inactive to active.

- NOTE** 
- If the Home Input is active when the command is issued, the move profile begins at step 5 above.

## Homing Profiles (continued)

### Profile with Proximity Input

Figure R4.2 below shows the move profile generated by a CW Find Home command when you use:

- Home Input with Network Home Proximity bit
- Marker Pulse home with Home Input as proximity sensor
- Marker Pulse home with Network Home Proximity bit

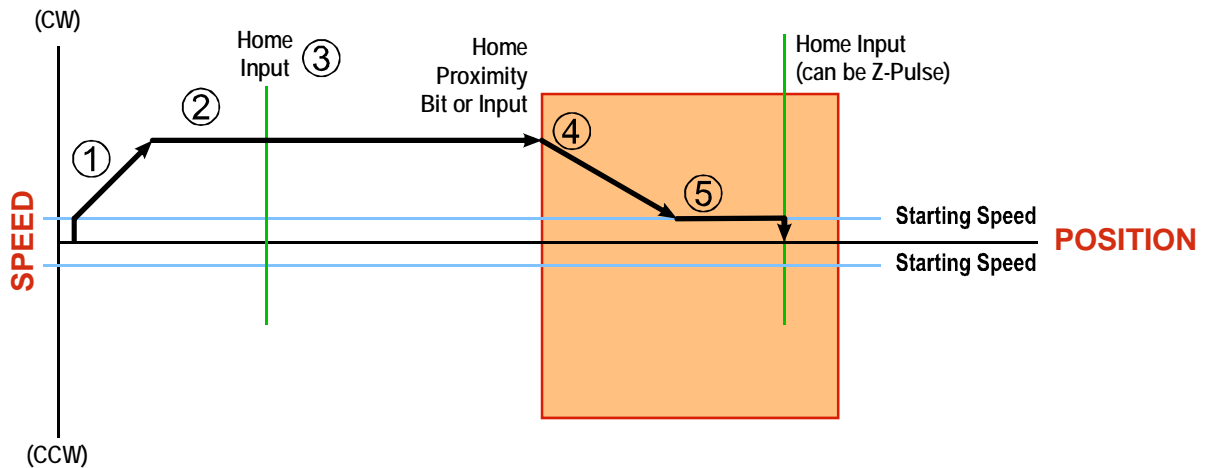



Figure R4.2 Homing with Proximity

- 1) Acceleration from the configured Starting Speed to the Programmed Speed
- 2) Run at the Programmed Speed
- 3) Ignores homing input, (Home Input or Marker Pulse) because proximity input has not made a 0→1 transition.
- 4) Deceleration towards the Starting Speed when the proximity input (Backplane bit or Home Input) transitions from its inactive to active state. The axis will stop as soon as the Home Input becomes active.
- 5) The Starting Speed is the minimum speed the profile will run at. If the axis decelerates to the Starting Speed before reaching the Home Input, it will continue at this speed.

**NOTE**  Figure R4.2 shows the Proximity Input, which is either the Network Home Proximity bit or the Home Input, staying active until the SD4840E2 reaches its home position. This is valid, but does not have to occur. As stated in step 4, the SD4840E2 starts to hunt for the home position as soon and the Proximity Input makes a 0→1 transition.

**Homing Profiles (continued)**

**Profile with Overtravel Limit**

Figure R4.3 below shows the move profile generated by a CW Find Home command when you use:

- CW Overtravel Limit
- Home Input without Network Home Proximity Bit

The profile is generated when you encounter an overtravel limit in the direction of travel. (In this example, hitting the CW limit while traveling in the CW direction.) Hitting the overtravel limit associated with travel in the opposite direction is an Immediate Stop condition. The axis will stop all motion and issue a *Home Invalid* error to your host.

The SD4840E2 will stop the axis with an error if both overtravel limits are activated while the unit is trying to find the home position.

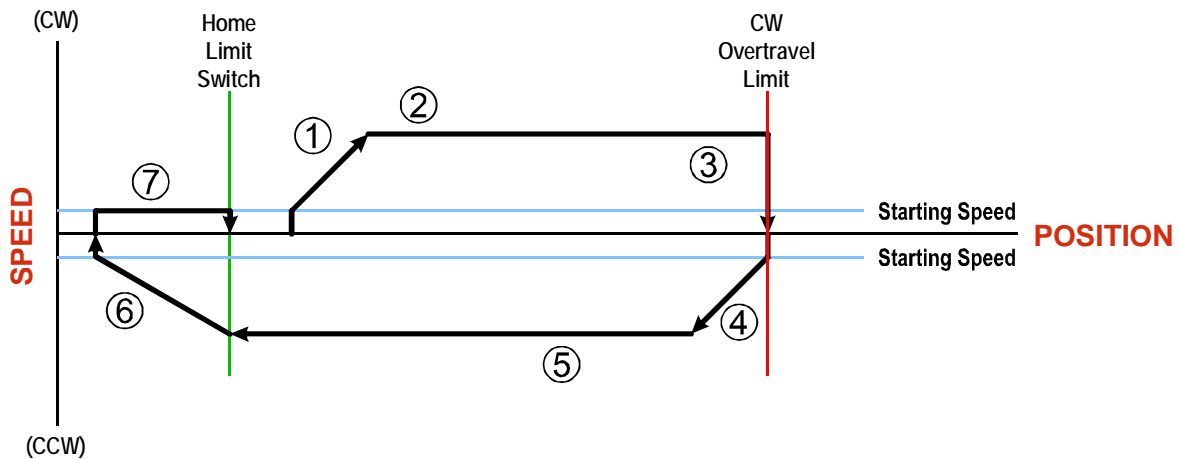


Figure R4.3 Profile with Overtravel Limit

- 1) Acceleration from the configured Starting Speed to the Programmed Speed
- 2) Run at the Programmed Speed
- 3) Hit CW Limit and immediately stop, followed by a two second delay.
- 4) Acceleration to the Programmed Speed opposite to the requested direction.
- 5) Run opposite the requested direction until the Home Input transitions from Active to Inactive
- 6) Deceleration to the Starting Speed and stop, followed by a two second delay.
- 7) Return to the Home Input at the configured Starting Speed. Stop when the Home Input transitions from inactive to active.

**NOTE** ⚠ If the overtravel limit is active when the Find Home Command is active, the profile will begin at step 4.

## CONFIGURATION MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to configure the SD4840E2 as well as the formats of the Network Input Data that contains the responses from the module. Each SD4840E2 requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data.

### **Modes of Operation**

The SD4840E2 has two operating modes, Configuration Mode and Command Mode. You switch between these modes by changing the state of a single bit in the Network Output Data.

#### **Configuration Mode**

Configuration Mode gives you the ability to select the proper configuration for your application without having to set any switches. The SD4840E2 must be configured on every power up. Additionally, the motor will not receive power until the module is configured properly.

#### **Command Mode**

This mode gives you the ability to program and execute stepper moves, and reset errors when they occur. The SD4840E2 will always power up in this mode, but a valid configuration must be sent to the SD4840E2 before it will apply power to the motor or allow you to issue move commands. The command data formats are described in the following chapter.

### **Power Up Behavior**

An SD4840E2 will always power up in Command Mode. If available, the unit will use its stored configuration data to configure itself. The SD4840E2 will then check for valid network command data and will only enable the motor driver section if the Enable\_Driver bit is set.

### **Configuration Mode Data Format**

An SD4840E2 requires twenty bytes of Output Data as well as twenty bytes of Input Data. Many of the hosts that can be used with the SD4840E2 only support sixteen bit integers. Sixteen bit integers support a range of values from -32,768 to 32,767 or 0 to 65,535. The Starting Speed parameter, which is programmed as part of the configuration data, can exceed this range. This parameter is transmitted in two separate words. The table below shows how values are split.

Value	First Word	Second Word
12	0	12
12,345	12	345

Table R5.1 Multi-Word Format Examples

### **Command Mode Data Formats**

When issuing commands to the SD4840E2, there are several parameters that are larger than sixteen bits. These parameters are:

- Target Position
- Programmed Speed
- Stopping Distance
- Minimum Registration Move Distance
- Position Preset Value
- Encoder Preset Value

**Command Mode Data Formats (continued)**

Likewise, when reading data back from a unit while it is in Command Mode, there are values that are larger than sixteen bits. These data values are:

- Motor Position
- Encoder Position
- Captured Encoder Position

By default, these thirty-two bit parameters and data values are written to and read from the SD4840E2 using the multi-word format described above. When configuring the unit, it is possible to program it to use a 32-bit double integer format instead of the custom format shown above.


There are two separate programming bits. The *Binary\_Output\_Format* Bit, controls the format of the programmable parameters written to the unit when issuing commands. The *Binary\_Input\_Format* Bit, controls the format of the data values written to the host controller by the SD4840E2.

When using the signed thirty-two bit format, there is an additional parameter named *Binary\_Endian*. This parameter controls the order of the 16 bit words within the 32 bit field. Least significant word first is called *little endian*, most significant word first is called *big endian*. EtherNet/IP controllers use *little endian* format, while Modbus and PROFINET controllers use *big endian* by default.

Examples of the formats are given below.

Value	Multi-Word Format		32 bit Signed Integer Little Endian Format		32 bit Signed Integer Big Endian Format	
	First Word	Second Word	First Word	Second Word	First Word	Second Word
12	0	12	16#000C	16#0000	16#0000	16#000C
-12	0	-12	16#FFF4	16#FFFF	16#FFFF	16#FFF4
1,234,567	1,234	567	16#D687	16#0012	16#0012	16#D687
-7,654,321	-7,654	-321	16#344F	16#FF8B	16#FF8B	16#344F

Table R5.2 Position Data Format Examples

**NOTE**  The range of values when using the multi-word format is -32,768,000 to 32,767,999. When used in continuous rotation applications, such as control of a conveyor belt, it is possible to overflow these values. When any of the three position values overflow, the value of the associated data words will become indeterminate. AMCI strongly suggests using the signed 32-bit integer format for continuous rotation applications.

## Output Data Format

The correct format for the Network Output Data when the SD4840E2 is in Configuration Mode is shown below. The EtherNet/IP and Modbus/TCP addresses are shown.

EtherNet/IP Word	Modbus/TCP Register	Configuration Data	Range
0	1024	Configuration Word 0	See below
1	1025	Configuration Word 1	See below
2	1026	Starting Speed: Upper Word	Combined value between 1 and 1,999,999 steps/sec.
3	1027	Starting Speed: Lower Word	
4	1028	Motor Steps/Turn	200 to 32,767
5	1029	Hybrid Control Gain	0, 1, 2, or 3
6	1030	Encoder Pulses/Turn	0 to 32,767
7	1031	Idle Current Percentage	0 to 100%
8	1032	Motor Current (X10)	1 to 40, Represents 0.1 to 4.0 Arms
9	1033	Current Loop Gain	1 to 80

Table R5.3 Network Output Data Format: Configuration Mode

## Configuration Word 0 Format

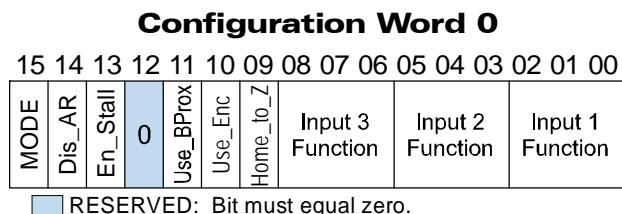


Figure R5.1 Configuration Mode: Control Word Format

- Bit 15: Mode\_Select Bit** – “1” for Configuration Mode Programming, “0” for Command Mode Programming. The SD4840E2 powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), unless:
- A valid configuration was written to the unit when it connected to the network, or
  - A valid configuration has been written to the flash memory of the SD4840E2.
- The SD4840E2 will not power the motor or accept commands until a valid configuration is written to it.
- Bit 14: Disable\_Anti-Resonance Bit** – “0” enables the anti-resonance feature of the SD4840E2. “1” disables the anti-resonance feature. The Anti-resonance feature will provide smoother operation in most cases. If you are still experiencing resonance problems with this feature enabled, disable this feature and test the machine again.
- Bit 13: Enable\_Stall\_Detection Bit** – “0” disables motor stall detection. “1” enables motor stall detection. Only valid when an encoder is used and attached to the motor controlled by the SD4840E2. The Quadrature Encoder Enable Bit, (bit 10 of this word), must be set. The encoder pulses/turn parameter, (Configuration Word 6) must also be programmed to a value that is four times the number of lines in your encoder.
- Bit 12: Reserved** – Must equal zero.



**Output Data Format (continued)**

**Control Word Format (continued)**

- Bit 11: Use\_Backplane\_Proximity Bit** – “0” when the Backplane\_Proximity\_Bit is not used when homing the SD4840E2. “1” when the Backplane\_Proximity\_Bit is to used when homing the SD4840E2. Note that this bit is not the Backplane\_Proximity\_Bit, but enables or disables its operation. Do not use the Backplane\_Proximity\_Bit if you only want to home to the Home Limit Switch. (Leave this bit equal to “0”.)
- Bit 10: Use\_Encoder Bit** – “0” when the quadrature encoder inputs are not used. “1” to enable the quadrature encoder inputs. You must also program the Encoder\_Resolution parameter in configuration word 6.
- Bit 9: Home to Encoder Z Pulse** – Set to “1” to home the machine to the encoder’s Z pulse. The Quadrature Encoder Enable Bit, bit 10, must also be set. You must also program the Encoder\_Resolution parameter in Configuration word 6. If a Discrete DC Input is configured as a Home Input, it will act as a Home Proximity Input.
- Bits 8-6: Input 3 Function Bits** – See Table Below
- Bits 5-3: Input 2 Function Bits** – See Table Below
- Bits 2-0: Input 1 Function Bits** – See Table Below

Bits			Function	Available On
8	7	6		
5	4	3		
2	1	0	Function	Available On
0	0	0	General Purpose Input	The input is not used in any of the functions of the SD4840E2, but it’s status is reported in the Network Data. This allows the input to be used as a discrete DC input to the host controller.
0	0	1	CW Limit	Input defines the mechanical end point for CW motion.
0	1	0	CCW Limit	Input defines the mechanical end point for CCW motion.
0	1	1	Start Indexed Move	Starts a move that is held in memory.
0	1	1	Start Indexed Move / Capture Encoder Value	When the encoder is enabled, the encoder position value is captured whenever this input transitions. An inactive-to-active state transition will also trigger an Indexed Move if one is pending in the SD4840E2.
1	0	0	Stop Jog or Registration Move	Brings a Jog or Registration Move to a controlled stop.
1	0	0	Stop Jog or Registration Move & Capture Encoder Value	When the encoder is enabled, the encoder position value is captured when the input triggers a controlled stop to a Jog or Registration move.
1	0	1	Emergency Stop	All motion is immediately stopped when this input makes an inactive-to-active transition.
1	1	0	Home	Used to define the home position of the machine. When homing to the Z-pulse of the encoder, (Bit 9 of the Control Word set to “1”), this input will act as a Home Proximity input.
1	1	1	Invalid Combination	This bit combination is reserved.

Table R5.4 Configuration Word 0 Format

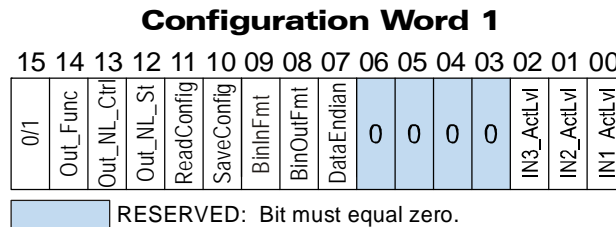
**Output Data Format (continued)****Configuration Word 1 Format**

Figure R5.2 Configuration Word 1 Format

- Bit 15:** **Reserved** – State ignored.
- Bit 14:** **Output\_Functionality** – “0” configures Output 1 to be a Fault Output. The output will conduct current until a fault occurs. “1” configures Output 1 to be a general purpose output whose state is determined by a bit in the Command Mode Network Output Data. This output is in its ON state when power is applied to the SD4840E2 and it has not yet been configured.
- Bit 13:** **Output\_State\_Control\_On\_Network\_Lost** – “0” will keep Output 1 at its last value. “1” will set the state of Output 1 to the value specified in Bit 12 of this word.
- Bit 12:** **Output\_State\_On\_Network\_Lost** – When bit 13 of this word is set, Output 1 will be set to the state of this bit if the network connection is lost.
- Bit 11:** **Read\_Present\_Configuration** – If this bit is set when you enter Configuration Mode, the SD4840E2 responds by placing the present configuration data in the Network Input Data. You cannot write new configuration data to the module while this bit is set. The format of the Configuration Data is given in the *Input Data Format* section of this chapter, starting on page 61.
- Bit 10:** **Save\_Configuration** – The SD4840E2 will store the configuration data to flash memory when this bit makes a 0→1 transition. The validity of the configuration data is checked before being written to the flash memory. If the data is not correct, the transition on this bit is ignored. Once the write to flash is completed, the SD4840E2 will write 0xAAAA into the last status word if the write was successful and the Status LED will begin flashing green, or the SD4840E2 will write 0xEEEE into the last status word if the write was unsuccessful and the Status LED will begin flashing red. Once the SD4840E2 does this, it will stop responding to commands and you must cycle power to the module. This design decision is to prevent the flash memory from constant write commands. The flash memory has a minimum of 10,000 write cycles.
- Bit 9:** **Binary\_Input\_Format** – Set to “0” to have the Motor Position, Encoder Position, and Trapped Encoder Position reported in the multi-word format. Set to “1” to have the Motor Position, Encoder Position, and Trapped Encoder Position reported in signed 32-bit integer format. When this parameter is set to “1”, the Binary\_Endian parameter in bit 7 sets the word order of the 32-bit value. See *Position Data Format Examples* found on page 56 for examples of the different formats.
- Bit 8:** **Binary\_Output\_Format** – Set to “0” to program the multi-word parameters of commands in the multi-word format. Set to “1” to program the multi-word parameters in signed 32-bit integer format. When this parameter is set to “1”, the Binary\_Endian parameter in bit 7 sets the word order of the 32-bit value. See *Position Data Format Examples* found on page 56 for examples of the different formats.



This bit only affects multi-word parameters in command data. The Starting Speed, which is programmed as part of the configuration data, is always programmed in the multi-word format.

***Output Data Format (continued)***

**Configuration Word Format (continued)**

**Bit 7:** **Binary\_Endian** – Only used when bits 8 and/or 9 above are set to “1”, set to “0” to the 32-bit values in little endian format. (EtherNet/IP) Set to “1” to program the 32-bit values in big endian format. Modbus TCP and PROFINET systems typically use big endian format, but you should refer to your PLC’s documentation to verify the format used by your processor. See [Position Data Format Examples](#) found on page 56 for examples of the different formats.

**Bits 6 - 3: Reserved** – Must equal zero.

**Bit 2:** **Input3\_Active\_Level** – Determines the active state of Input 3. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

**Bit 1:** **Input2\_Active\_Level** – Determines the active state of Input 2. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

**Bit 0:** **Input1\_Active\_Level** – Determines the active state of Input 1. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

**Notes on Other Configuration Words**

- See [Configuration Mode Data Format](#) found on page 55 for examples of the multi-word data format used when programming the Starting Speed.
- Hybrid Control Gain controls the “stiffness” of the shaft at zero speed when using the Hybrid Control feature. With a higher gain setting, the SD4840E2 will increase the motor current more quickly to compensate for shaft loading.
- Changes to the Idle Current only take effect at the *end of the first move after re-configuration*.
- You should start with Current Loop Gain setting of 5 for all AMCI motors and can adjust the setting from there based on your system requirements. Refer to the [Current Loop Gain](#) section on page 39 for additional information.

## Input Data Format

The correct format for the Network Input Data when the SD4840E2 is in Configuration Mode is shown below. The EtherNet/IP and Modbus/TCP addresses are shown.

EtherNet/IP Word	Modbus/TCP Register	Configuration Data
0	0	Control Word (see definition below)
1	1	Mirror of Configuration Word 1
2	2	Mirror of Starting Speed: Upper Word
3	3	Mirror of Starting Speed: Lower Word
4	4	Mirror of Motor Steps/Turn
5	5	Mirror of Hybrid Control Gain
6	6	Mirror of Encoder Pulses/Turn
7	7	Mirror of Idle Current Percentage
8	8	Mirror of Motor Current (X10)
9	9	Mirror of Current Loop Gain or Status message when writing Configuration data to flash memory.

Table R5.5 Network Input Data Format: Configuration Mode

### Control Word Format (Word 0)

When the Configuration data is valid and accepted, this word mirrors the value of the Configuration Word 0 written to the SD4840E2. When the SD4840E2 is not configured, or the data written to it is invalid, then this word has the same format of Status Word 1 when the SD4840E2 is in Command Mode. This format is explained in the *Status Word 0 Format* section starting on page 79. On power up, the value of this word will be 6408h unless a valid configuration had been previously written to the unit.

### Invalid Configurations

The following configurations are invalid:

- 1) Setting any of the reserved bits in the configuration words.
- 2) Setting any parameter to a value outside of its valid range. This includes setting the Lower Word of the Starting Speed to a value greater than 999.
- 3) You configure two or more inputs to have the same specific function, such as two CW Limit Switches. It is possible to program the SD4840E2 to have two or three General Purpose Inputs.
- 4) Setting the *Home to Encoder Z-Pulse Bit* without configuring the SD4840E2 to use the encoder.
- 5) Setting the *Stall Detection Enable Bit* without configuring the SD4840E2 to use the encoder.
- 6) Using an encoder and not setting the Encoder Pulses/Turn parameter in word 6 to a valid value.
- 7) Setting the Input Configuration bits for any input to "111". See table R5.4 on page 58 for more information.

***Notes***

# REFERENCE 6

## COMMAND MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to command the SD4840E2 as well as the formats of the Network Input Data that contains the responses from the driver. The SD4840E2 requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data.

### Data Format

An SD4840E2 requires twenty bytes of Output Data as well as twenty bytes of Input Data. In most cases the data is represented as ten 16-bit (single) integers. Sixteen bit integers support a range of values from -32,768 to 32,767 or 0 to 65,535. When issuing commands to the SD4840E2, there are several parameters that are larger than sixteen bits. These parameters are:

- Target Position
- Programmed Speed
- Stopping Distance
- Minimum Registration Move Distance
- Position Preset Value
- Encoder Preset Value

Likewise, when reading data back from a unit while it is in Command Mode, there are values that are larger than sixteen bits. These data values are:

- Motor Position
- Encoder Position
- Captured Encoder Position

By default, these thirty-two bit parameters and data values are written to and read from the SD4840E2 using the multi-word format described below. When configuring the SD4840E2, it is possible to program it to use a 32-bit double integer format instead of the custom format shown above.

There are three configuration bits that control the data format when the SD4840E2 is in command mode. The *Binary\_Output\_Format* Bit, controls the format of the programmable parameters written to the SD4840E2 when issuing commands. The *Binary\_Input\_Format* bit controls the format of the data values written to the host controller by the SD4840E2. When either of these parameters are set to their 32-bit signed integer format settings, the *Binary\_Endian* bit determines if the 32-bit values are stored and transmitted least significant bits first or most significant bits first. Examples of the formats are given below.

Value	Multi-Word Format		32 bit Signed Integer Little Endian Format		32 bit Signed Integer Big Endian Format	
	First Word	Second Word	First Word	Second Word	First Word	Second Word
12	0	12	16#000C	16#0000	16#0000	16#000C
-12	0	-12	16#FFF4	16#FFFF	16#FFFF	16#FFF4
1,234,567	1,234	567	16#D687	16#0012	16#0012	16#D687
-7,654,321	-7,654	-321	16#344F	16#FF8B	16#FF8B	16#344F

Table R6.1 Position Data Format Examples


#### NOTE

The range of values when using the multi-word format is -32,768,000 to 32,767,999. When used in continuous rotation applications, such as control of a conveyor belt, it is possible to overflow these values. When any of the three position values overflow, the value of the associated data words will become indeterminate. AMCI strongly suggests using the signed 32-bit integer format for continuous rotation applications.

**Command Bits Must Transition**

Commands are only accepted when the command bit makes a 0→1 transition. The easiest way to do this is to write a value of zero into the Command Bits MSW before writing the next command.

This condition also applies when switching from Configuration Mode to Command Mode.

**NOTE**  If your move only occurs once, it is probably because your code does not reset the command bit to zero before issuing another command.

**Output Data Format**

The following table shows the format of the output network data words when writing command data to the SD4840E2. The EtherNet/IP and Modbus/TCP addresses are shown.

EtherNet/IP Word	Modbus/TCP Register	Function
0	1024	Command Word 0
1	1025	Command Word 1
2	1026	Command Parameters  <i>Word meaning depends on the command set to the SD4840E2</i>
3	1027	
4	1028	
5	1029	
6	1030	
7	1031	
8	1032	
9	1033	

Table R6.2 Command Mode Data Format

**Command Word 0**

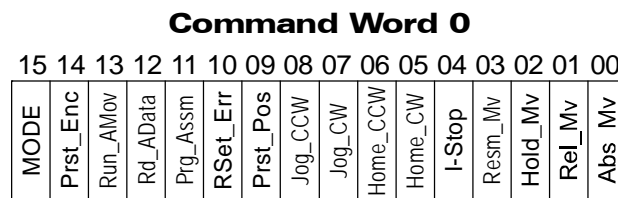


Figure R6.1 Command Word 0 Format

- Bit 15: Mode Select** – “1” for Configuration Mode Programming “0” for Command Mode Programming. The SD4840E2 powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), if a valid configuration has not been stored in flash memory. The SD4840E2 will not power the motor or accept commands until a valid configuration is written to it or read from flash memory.
- Bit 14: Preset\_Encoder** – When this bit makes a 0→1 transition, the SD4840E2 will preset the Encoder Position to the value stored in Output Words 2 and 3.



**Command Word 0 (continued)**

- Bit 13: Run\_Assembled\_Move** – When this bit makes a 0→1 transition, the SD4840E2 will run the Assembled Move already stored in memory.
- **Assembled Move Type – Command Word 1, Bit 9:** This bit determines the type of move that is run. When this bit equals “0”, a Blend Move is run. When this bit equals “1”, a Dwell Move is run. When starting a Dwell Move, the Dwell Time is programmed in word 9 of the Command Data. The value is programmed in milliseconds and can range from 0 to 65,536.
  - **Blend Move Direction – Command Word 1, Bit 4:** This bit is used to determine the direction that the Blend Move will be run in. When this bit equals “0”, the Blend Move runs in the clockwise direction. When this bit equals “1”, the Blend Move is run in the counter-clockwise direction.
- Bits 12 & 11: Program\_Assembled & Read\_Assembled\_Data** – These bits are used to program the segments of an Assembled Move before the move can be run. Their use is explained in the *Assembled Move Programming* section of this manual starting on page 33.
- Bit 10: Reset\_Errors** – When set to “1” the SD4840E2 will clear all existing errors and attempt to use the present data to run a new command. This command also resets the *Move Complete* bit.
- Bit 9: Preset\_Motor\_Position** – When set to “1” the SD4840E2 will preset the Motor Position to the value stored in Output Words 2 and 3. This command also resets the *Move\_Complete* and *Position\_Invalid* bits.
- Bit 8: Jog\_CCW** – When this bit makes a 0→1 transition, the SD4840E2 will run a Jog Move in the counter-clockwise direction. The full explanation of a *CW/CCW Jog Move* can be found starting on page 26.
- **Run Registration Move – Command Word 1, Bit 7:** When this bit equals “0”, and a Jog Move command is issued, it will run as a standard Jog Move. When this bit equals “1” and a Jog Move command is issued, the move will run as a Registration Move.
  - **Enable Electronic Gearing Mode – Command Word 1, Bit 6:** When this bit equals “1” the SD4840E2 will switch its operation to *Electronic Gearing* mode as described on page 37. While in this mode, the two Jog Move bits are used to enable motor motion. One of these two bits must equal “1” before the motor will follow a change in encoder position.
- Bit 7: Jog\_CW** – When this bit makes a 0→1 transition, the SD4840E2 will run a Jog Move in the clockwise direction. The full explanation of a *CW/CCW Jog Move* can be found starting on page 26.
- **Run Registration Move – Command Word 1, Bit 7:** When this bit equals “0”, and a Jog Move command is issued, it will run as a standard Jog Move. When this bit equals “1” and a Jog Move command is issued, the move will run as a Registration Move.
  - **Enable Electronic Gearing Mode – Command Word 1, Bit 6:** When this bit equals “1” the SD4840E2 will switch its operation to *Electronic Gearing* mode as described on page 37. While in this mode, the two Jog Move bits are used to enable motor motion. One of these two bits must equal “1” before the motor will follow a change in encoder position.
- Bit 6: Find\_Home\_CCW** – When set to “1” the SD4840E2 will attempt to move to the Home Limit Switch in the counter-clockwise direction. A full explanation of homing can be found in the *Homing the SD4840E2* chapter starting on page 51.
- Bit 5: Find\_Home\_CW** – When set to “1” the SD4840E2 will attempt to move to the Home Limit Switch in the clockwise direction. A full explanation of homing can be found in the *Homing the SD4840E2* chapter starting on page 51.
- Bit 4: Immediate\_Stop** – When set to “1” the SD4840E2 will stop all motion without deceleration. The Motor Position value will become invalid if this bit is set during a move. Setting this bit when a move is not in progress will not cause the Motor Position to become invalid.

**Command Word 0 (continued)**

- Bit 3: Resume\_Move** – Set to “1” to resume a move that you previously placed in a hold state. Use of the Resume Move bit can be found in the *Controlling Moves In Progress* section starting on page 36. Note that a move in its hold state does not need to be resumed. The move is automatically cancelled if another move is started in its place.
- Bit 2: Hold\_Move** – Set to “1” to hold a move. The move will decelerate to its programmed Starting Speed and stop. The move can be completed by using the Resume Move bit. The use of the Hold Move and Resume Move bits is further explained in the *Controlling Moves In Progress* section starting on page 36.
- Bit 1: Relative\_Move** – Set to “1” to perform a Relative Move using the data in the rest of the Command Data. The full explanation of a *Relative Move* can be found starting on page 24.
- Bit 0: Absolute\_Move** – Set to “1” to perform an Absolute Move using the data in the rest of the Command Data. The full explanation of an *Absolute Move* can be found starting on page 25.

**Command Word 1**

**Command Word 1**

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
En_Driver	V_Enc_Follow	GPOut_State	V_Pos_Follow	BP_Prox	Clr_Drv_Fit	AsMv_Type	Index_Cmd	Reg_Move	En_EIGear	Save_Asmv	Rev_BlendDir	EN_Hybrid	Enc_Reg_Mv	Current_Key1	Current_Key0

Figure R6.2 Command Word 1 Format

- Bit 15: Enable\_Driver** – “0” to disable the motor current, “1” to enable motor current. A valid configuration must be written to the SD4840E2 before the driver can be enabled.
- Bit 14: Virtual\_Encoder\_Follower** – When using the SD4840E2 as an axis follower, set this bit to ‘1’ to close the position loop with respect to the encoder position. This bit is only available when the unit is configured to use encoder feedback. This bit must equal ‘0’ if the Virtual\_Position\_Follower bit, bit 12 of this word, is set to ‘1’.
- Bit 13: General\_Purpose\_Output\_State** – When the output is configured as a general purpose output point instead of the Fault Output, this bit controls the state of the output. When this bit equals a “1”, the output is on and conducts current.
- Bit 12: Virtual\_Position\_Follower** – When using the SD4840E2 as an axis follower, set this bit to ‘1’ to close the position loop with respect to the motor position. This bit must equal ‘0’ if the Virtual\_Encoder\_Follower bit, bit 14 of this word, is set to ‘1’.
- Bit 11: Backplane\_Proximity\_Bit** – When the SD4840E2 is configured to use the Network Home Proximity bit, the SD4840E2 will ignore the state of the Home Input as long as this bit equals “0”. This bit must equal “1” before a transition on the Home Input can be used to home the machine. These statements also apply when the encoder Z-pulse is used to home the machine. Please note that you cannot use the Network Home Proximity bit and a discrete input configured as a Home Proximity Input at the same time. You must choose one or the other when configuring your machine. Further information on using the Network Home Proximity bit can be found in the *Homing the SD4840E2* chapter starting on page 51.
- Bit 10: Clear\_Driver\_Fault** – If this bit is set when a Reset Errors Command is issued, (Command Word 0 Bit 10) the SD4840E2 will attempt to clear driver errors such as an overtemperature or motor short fault. Note that the driver must be disabled (Command Word 1, Bit 15 = 0), when using this command.

**Command Word 1 (continued)**

- Bit 9: Assembled\_Move\_Type** – When this bit equals “0”, a Blend Move is started when the Run Assembled Move bit, (Command Word 1, Bit 13) makes a 0→1 transition. When this bit equals “1”, a Dwell Move is started on the transition. The direction of a Blend Move is controlled by the Blend Move Direction bit, (Command Bits LSW, Bit 4). In a Dwell Move, the Dwell Time between segments is programmed in Word 9 of the command data.
- Bit 8: Indexed\_Command** – If this bit is set when a move command is issued, the SD4840E2 will not run the move immediately, but will instead wait for an inactive-to-active transition on an input configured as a *Start Indexer Move* input. The move will be run on every inactive-to-active transition on the input provided that the valid data remains in the network output data registers of the SD4840E2.
- Bit 7: Registration\_Move** – When this bit equals “0”, and a Jog Move command is issued, it will run as a standard Jog Move. When this bit equals “1” and a Jog Move command is issued, the move will run as a Registration Move.
- Bit 6: Enable\_Electronic\_Gearing\_Mode** – Set to “1” to put the SD4840E2 in Electronic Gearing mode. Set to “0” for normal operation. A full description of *Electronic Gearing* mode starts on page 37.
- Bit 5: Save\_Assembled\_Move** – Set this bit to save the programmed Assembled Move to flash memory. This bit must be set when the Move Program Mode makes a 1→0 transition at the end of the Assembled Move programming cycle to write the data to flash memory. The SD4840E2 responds by flashing the Status LED when the writing is complete. If the LED is flashing green, the write to flash memory was successful. If it flashes red, then there was an error in writing the data. In either case, power must be cycled to the SD4840E2 before you can continue. This design decision is to prevent the flash memory from constant write commands. The flash memory has a minimum of 10,000 write cycles.
- Bit 4: Reverse\_Blend\_Direction** – When you command a Blend Move to run, this bit determines the direction of rotation. Set to “0” for a clockwise Blend Move, ‘1’ for a counter-clockwise Blend Move.
- Bit 3: Hybrid\_Control\_Enable** – Set to “0” to operate the SD4840E2 as a standard stepper system. Set to “1” to enable Hybrid Control. *Hybrid Control* is explained starting on page 38.
- Bit 2: Encoder\_Registration\_Move** – Set to “1” when you command an Absolute or Relative move to cause the move distance to be based on the encoder position, not on the motor position. Set to “0” for a standard Relative or Absolute move based on motor position. Please note that an Encoder Registration Move does not stop at the programmed encoder position. It starts to decelerate at this position. A full description of *Encoder Registration Moves* starts on page 29.
- Bits 1&0: Current\_Key1 and Current\_Key0** – These bits can be used to set the motor current “on-the-fly” by setting the bits to a value of “10”. When these bits are “10”, the base motor current will be set to the value contained in word 8, the ninth word, of the command data.
- ▶When these bits are set to “10”, changes in output word 8 are acted upon immediately. The range of values for word 8 are 0 to 40. (0.0 to 4.0 amps) Values outside of this range are ignored and the last valid motor current setting will continue to be used.
  - ▶When these bits are set to “00”, “01”, or “11” the motor current is left at the last accepted value.
  - ▶Note that this procedure sets the *base* motor current. Any idle current reduction value set in the Configuration data will still affect the actual current delivered to the motor when the motor is idle.

**Command Blocks**

The following section lists the output data format for the sixteen different commands. The EtherNet/IP and Modbus TCP addresses are shown.

**Absolute Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0001
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Abs. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	1027	Abs. Target Position: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R6.3 Absolute Move Command Block

**Relative Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0002
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Rel. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	1027	Rel. Target Position: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R6.4 Relative Move Command Block

**Command Blocks (continued)****Hold Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0004
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Unused		See Note Below

Table R6.5 Hold Move Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

**Resume Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0008
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R6.6 Resume Move Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command. This is typically the case when resuming a move, the words are listed as “Unused” to highlight that the target position of a held move cannot be changed when the move is resumed.

**Command Blocks (continued)**

**Immediate Stop**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0010
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Unused		See Note Below

Table R6.7 Immediate Stop Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

**CW Find Home**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0020
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R6.8 Find Home CW Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

**Command Blocks (continued)****CCW Find Home**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0040
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R6.9 Find Home CCW Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

**CW Jog**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0080
1	1025	<i>Command Word 1</i>		See pg. 66 Bits 7 & 6 must equal "00"
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R6.10 Jog CW Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.



**Command Blocks (continued)**

**CW Registration Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0080
1	1025	<i>Command Word 1</i>		See pg. 66 Bits 7 & 6 must equal "10"
2	1026	Stopping Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
3	1027	Stopping Distance: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps per Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
9	1033	Min. Reg. Move Distance: Lower Word		

Table R6.11 Registration Move CW Command Block

**Synchrostep (Virtual Axis Follower) Move**

EtherNet/IP or PROFINET Word	Modbus TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0080 or 16#0100
1	1025	<i>Command Word 1</i>		See pg. 66 Bit 14 or 12 must equal "1" and the other "0". Bit 7 must equal "0"
2	1026	Lower 16 bits of 32 bit Position	Steps	Signed 32 bit double integer value
3	1027	Upper 16 bits of 32 bit Position		
4	1028	Lower 16 bits of 32 bit Velocity	Steps per Second	Signed 32 bit double integer value
5	1029	Upper 16 bits of 32 bit Velocity		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Proportional Coefficient		1 to 50
9	1033	Network Delay or Position Unwind		0 to 20 milliseconds or 21 to 65,535

Table R6.12 Synchrostep Move Command Block

When using motor position loop control, set the Proportional Coefficient to 1 or 2. When using encoder position loop control, set the Proportional Coefficient to a value between 10 and 50.

Word 9: If its value is between 0 and 20, the move is considered a Linear Axis Follower move and the value in word 9 is the Network Delay value. If the value is between 21 and 65,535, the move is considered a Circular Axis Move and the value in word 9 is the Position Unwind value.

**Command Blocks (continued)****CCW Jog**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0100
1	1025	<i>Command Word 1</i>		See pg. 66 Bits 7 & 6 must equal "00"
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R0.1 Jog CCW Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

**CCW Registration Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0100
1	1025	<i>Command Word 1</i>		See pg. 66 Bits 7 & 6 must equal "10"
2	1026	Stopping Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
3	1027	Stopping Distance: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps per Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
9	1033	Min. Reg. Move Distance: Lower Word		

Table R6.13 Registration Move CCW Command Block

**Command Blocks (continued)**

**Encoder Follower Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0080 or 16#0100
1	1025	<i>Command Word 1</i>		See pg. 66 Bit 6 must equal "1"
2	1026	Electronic Gearing Numerator		1 to 255
3	1027	Electronic Gearing Denominator		1 to 255
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Acceleration Jerk		0 to 5000

Table R6.14 Encoder Follower Move Command Block

**Preset Position**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0200
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Position Preset Value: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	1027	Position Preset Value: Lower Word		
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Unused		See Note Below

Table R6.15 Preset Position Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

- Presetting the position will also reset the *Move Complete* status bit.

**Command Blocks (continued)****Reset Errors**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0400
1	1025	<i>Command Word 1</i>		See pg. 66 Set bit 10 to clear driver faults
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Unused		See Note Below

Table R6.16 Reset Errors Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command. Resetting errors will also reset the *Move Complete* status bit.

Resetting errors will also reset the *Move\_Complete* status bit in the Network Input Data. Resetting errors will not reset the *Position\_Invalid* or *Configuration\_Error* bits.

**Run Assembled Move**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#2000
1	1025	<i>Command Word 1</i>		See pg. 66 Blend Move: Bit 9 = "0" Dwell Move: Bit 9 = "1" Blend Move direction set by Bit 4.
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Unused with Blend Move Dwell Time with Dwell Move	milliseconds	0 to 65,535

Table R6.17 Run Assembled Move Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

**Command Blocks (continued)**

**Preset Encoder Position**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#4000
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Encoder Preset Value: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	1027	Encoder Preset Value: Lower Word		
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Motor Current	0.1 Amps	0 to 40 (0.0 to 4.0 amps)
9	1033	Unused		See Note Below

Table R6.18 Preset Encoder Position Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

## Programming Blocks

The following blocks are used to program an Assembled Move. Both of the moves, Blend Move, and Dwell Move, are programmed exactly the same way. The bit configuration used when starting the move determines which type of Assembled Move is run.

### First Block

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0800
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused		See Note Below

Table R6.19 Assembled Move First Programming Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values from the previous command.

Once the first block is transmitted, the SD4840E2 responds by setting bits 8 and 9 in Status Word 1. (See [Status Word 0 Format](#) starting on page 79.) Once these are set, you can then start transmitting Segment Blocks.

### Segment Block

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#1800
1	1025	<i>Command Word 1</i>		See pg. 66
2	1026	Rel. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	1027	Rel. Target Position: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/ms/sec	1 to 5000
7	1031	Deceleration	Steps/ms/sec	1 to 5000
8	1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	1033	Acceleration Jerk		0 to 5000

Table R6.20 Assembled Move Segment Programming Block

Note that each Segment Block starts with bits 11 and 12 set in the Command Bits MSW word (16#1800). When the SD4840E2 sees bit 12 of Command Bits MSW set, it will accept the block and reset bit 9 in Status Word 1. When your program sees this bit reset, it must respond by resetting bit 12 of Command Bits MSW. The SD4840E2 will respond to this by setting bit 9 in Status Word 1 and the next Segment Block can be written to the SD4840E2. You can write a maximum of sixteen Segment Blocks for each Assembled Move.

**Input Data Format**


The correct format for the Network Input Data when the SD4840E2 is in Command Mode is shown below. The EtherNet/IP and Modbus/TCP addresses are shown.

EtherNet/IP Word	Modbus/TCP Register	Command Mode Input Data
0	0	Status Word 0
1	1	Status Word 1
2	2	Motor Position (See format below)
3	3	Motor Position (See format below)
4	4	Encoder Position (See format below)
5	5	Encoder Position (See format below)
6	6	Captured Encoder Position (See format below)
7	7	Captured Encoder Position (See format below)
8	8	Programmed Motor Current (X10)
9	9	Value of Acceleration Jerk Parameter

Table R6.21 Network Input Data Format: Command Mode

**Format of Position Data Values**

The format of the Motor Position, Encoder Position, and Captured Encoder Position values is controlled by the Binary\_Input\_Format bit in the configuration data written to the SD4840E2. (See [Configuration Word 1 Format](#) on page 59.) When the Data Format bit equals “0”, the position values are reported using the multi-word data format described in table R6.22 below. When the Data Format bit equals “1”, the position values are reported as 32-bit signed integers. The position of the least significant bit is controlled by the Binary\_Endian parameter value in the Configuration Word 1.

**NOTE**  The range of values when using the multi-word format is -32,768,000 to 32,767,999. When used in continuous rotation applications, such as control of a conveyor belt, it is possible to overflow these values. When any of the three position values overflow, the value of the associated data words will become indeterminate. AMCI strongly suggests using the signed 32-bit integer format for continuous rotation applications.

Value	Multi-Word Format		32 bit Signed Integer Little Endian Format		32 bit Signed Integer Big Endian Format	
	First Word	Second Word	First Word	Second Word	First Word	Second Word
12	0	12	16#000C	16#0000	16#0000	16#000C
-12	0	-12	16#FFF4	16#FFFF	16#FFFF	16#FFF4
1,234,567	1,234	567	16#D687	16#0012	16#0012	16#D687
-7,654,321	-7,654	-321	16#344F	16#FF8B	16#FF8B	16#344F

Table R6.22 Position Data Format Examples




**Input Data Format (continued)****Status Word 0 Format**

Status Word 0															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Mode_Flag	Module_OK	Config_Err	Command_Err	Input_Err	Pos_Invalid	Wait_AssySeg	In_Assy_Mode	Move_Cmp	Decelerating	Accelerating	At_Home	Stopped	In_Hold_State	Moving_CCW	Moving_CW

Figure R6.3 Command Mode: Status Word 1 Format

- Bit 15: Mode\_Flag** – “1” for Configuration Mode Programming, “0” for Command Mode Programming. The SD4840E2 powers up in Command Mode and shows a configuration error unless a valid configuration was previously saved in flash memory. The SD4840E2 will not power the motor or accept commands until a valid configuration is written to it.
- Bit 14: Module\_OK** – “1” when the SD4840E2 is operating without a fault, “0” when an internal fault condition exists.
- Bit 13: Configuration\_Error** – “1” on power up before a valid configuration has been written to the SD4840E2 or after any invalid configuration has been written to the module. “0” when the SD4840E2 has a valid configuration written to it or if valid configuration data is read from flash memory on power up.

**NOTE**  This bit is also set when stall detection is enabled and the SD4840E2 is in Configuration Mode. If you are using stall detection you must consider the state of this bit as well as the Mode Bit (bit 15) when determining if a configuration error exists.

**Bit 12: Command\_Error** – “1” when an invalid command has been written to the SD4840E2. This bit can only be reset by the Reset Errors bit, Command Bits MSW, Bit 10.

**Bit 11: Input\_Error** – “1” when:

- Emergency Stop input has been activated.
- Either of the End Limit Switches activates during any move operation except for homing.
- Starting a Jog Move in the same direction as an active End Limit Switch.
- If the opposite End Limit Switch is reached during a homing operation.

This bit is reset by a **Reset Errors** command. The format of the command is given on page 75.

**Bit 10: Position\_Invalid** – “1” when:

- A configuration is written to the SD4840E2
- The motor position has not been preset
- The machine has not been homed
- The Network Connection has been lost and re-established.
- An Immediate or Emergency Stop has occurred
- An End Limit Switch has been reached
- A motor stall has been detected.

Absolute moves cannot be performed while the position is invalid.

**Bit 9: Waiting\_For\_Assembled\_Segment** – The SD4840E2 sets this bit to tell the host that it is ready to accept the data for the next segment of your assembled move profile. Its use is explained in the **Assembled Move Programming** section of this manual starting on page 33.

**Input Data Format (continued)**

**Status Word 0 Format (continued)**

- Bit 8:** **In\_Assembled\_Mode** – The SD4840E2 sets this bit to signal the host that it is ready to accept assembled move profile programming data. Its use is explained in the *Assembled Move Programming* section of this manual starting on page 33.
- Bit 7:** **Move\_Complete** – Set to “1” when the present Absolute, Relative, Jog, Registration, or Assembled Move command completes without error. This bit is reset to “0” when the next move command is written to the SD4840E2, when the position is preset, or a Reset Errors command is issued to the unit. This bit is also set along with the Command Error bit (Bit 12 of this word), when any Jog Move or Registration Move parameters are outside of their valid ranges. This bit is not set on a command error for any other type of command. Finally, this bit is not set at the end of a homing operation.
- Bit 6:** **Decelerating** – Set to “1” when the present move is decelerating. Set to “0” at all other times.
- Bit 5:** **Accelerating** – Set to “1” when the present move is accelerating. Set to “0” at all other times.
- Bit 4:** **At\_Home** – Set to “1” when a homing command has completed successfully, “0” at all other times.
- Bit 3:** **Stopped** – Set to “1” when the motor is not in motion. Note that this is stopped for any reason, not just a completed move. For example, an Immediate Stop command during a move will set this bit to “1”, but the Move Complete Bit, (bit 7 above) will not be set.
- Bit 2:** **In\_Hold\_State** – Set to “1” when a move command has been successfully brought into a Hold State. Hold States are explained in the Controlling Moves In Progress section starting on page 22.
- Bit 1:** **Moving\_CCW** – Set to “1” when the motor is rotating in a counter-clockwise direction.
- Bit 0:** **Moving\_CW** – Set to “1” when the motor is rotating in a clockwise direction.

**Status Word 1 Format**

**Status Word 1**

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Drive_Enabled	Stall_Detected	Cmd_Ack	0	Heartbeat_Bit	Limit_Condition	Inv_Jog_Cng	Motion_Lag	Driver_Fault	Connect_Lost	PLC_In_PRG	Temp_90°C	0	IN3_Active	IN2_Active	IN1_Active

Figure R6.4 Command Mode: Status Word 2 Format

- Bit 15:** **Driver\_Is\_Enabled** – Present state of global Driver Enable Bit. When set to “0”, current is not available to the motor under any condition. When set to “1”, the motor driver section of the SD4840E2 is enabled and current is available to the motor but current may be removed for other reasons. Motor current is removed if there is a Driver Fault (Bit 7 below), or if the motor is idle and Idle Current Reduction is programmed to its *To 0%* setting. Note that the motor will still receive power under an E-Stop condition.
- Bit 14:** **Stall\_Detected** – Set to “1” when a motor stall has been detected.
- Bit 13:** **Output\_State** – Present actual state of DC Output. When this bit is set to “1”, the output is in its on state and conducts current.
- Bit 12:** **Reserved Bit** – Will always equal zero.
- Bit 11:** **Heartbeat Bit** – This bit will change state approximately every 500 milliseconds. Monitor this bit to verify that the module and network connection are operating correctly.

**NOTE** This bit is only available when in Command Mode.

**Input Data Format (continued)****Status Word 1 Format (continued)**

- Bit 10: Limit\_Condition** – This bit is set if an End Limit Switch is reached during a move. This bit will be reset when the Limit Switch changes from its active to inactive state, or when a Reset Errors Command is issued.
- Bit 9: Invalid\_Jog\_Change** – Set during a Jog Move if parameters are changed to invalid values. Parameters that can be changed during a Jog Move are Programmed Speed, Acceleration, and Deceleration. Set while in Electronic Gearing mode if the Numerator or Denominator are set outside their range of 1 to 255.
- Bit 8: Motion\_Lag** – When using the Hybrid Control feature, this bit will equal “1” if the actual rotor position is more than 360° behind the commanded position. This indicates a heavily loaded motor and the encoder position should be monitored for a stalled condition.
- Bit 7: Driver\_Fault** – If the driver section of the SD4840E2 is enabled, this bit will be a “1” during a Over-temperature Fault, or a Short Circuit Fault. This fault can be cleared by issuing a *Reset Errors* programming block with the Clear Driver Fault bit, (Command Word 1, bit 10) set to “1” and the Driver Enable bit is set to “0”. For additional information, see *Notes on Clearing a Driver Fault* on page 82.
- Bit 6: Connection\_Was\_Lost** – If the physical network connection is lost at any time, this bit will be set when the connection is re-established. The Input Error bit will also be set. Note that this bit is not set if the communication loss is not due to a physical error.
- Bit 5: PLC\_In\_PROG\_Mode** – On ControlLogix and CompactLogix platforms, this bit equals “1” when the PLC is in Program mode and “0” when in Run mode. This bit will always equal “0” on all other platforms.
- Bit 4: Temperature\_Above\_90C** – This bit is set to “1” when the processor internal temperature exceeds 90°C. At this point, the heatsink temperature is typically near 83°C. If this bit trips often and you want to lower the operating temperature of the module, consider installing a fan below the unit to allow additional airflow through the SD4840E2.
- Bit 3: Reserved Bit** – Will always equal zero.
- Bit 2: IN3\_Active** – “1” when Input 3 is in its active state. The active state of the input is programmed as explained in the *Configuration Word 1 Format* section starting on page 59.
- Bit 1: IN2\_Active** – “1” when Input 2 is in its active state. The active state of the input is programmed as explained in the *Configuration Word 1 Format* section starting on page 59.
- Bit 0: IN1\_Active** – “1” when Input 1 is in its active state. The active state of the input is programmed as explained in the *Configuration Word 1 Format* section starting on page 59.

**Notes on Clearing a Driver Fault**

A Driver Fault occurs when there is an over temperature condition, or a short circuit condition in the motor. When a Driver Fault occurs, the SD4840E2 sets bit 7 of Status Word 1 in the Network Input Data. (See *Status Word 1 Format* on page 80 for a full description of Status Word 1.) *Once you have cleared the fault condition, you can reset the Driver Fault with the following Command Block:*


**Reset Driver Fault**

EtherNet/IP Word	Modbus/TCP Register	Function	Units	Range
0	1024	<i>Command Word 0</i>		16#0400
1	1025	<i>Command Word 1</i>		16#0400
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused		See Note Below

Table 0.2 Reset Driver Fault Command Block

Unused words are ignored by the SD4840E2 and can be any value, including parameter values in the previous command.

Once the command block is accepted by the SD4840E2, it will respond by resetting bit 7 in Status Word 1 of the Network Input Data.

**NOTE**  After this procedure, there will still be no current to the motor. This is because the Driver Enabled Bit (bit 15 of Command Word 1 in the Network Output Data) must be reset when writing down the Reset Driver Fault Command Block. Setting this bit in the next command block will re-enable the motor.

## MOTOR AND POWER SUPPLY SIZING

This section is intended for anyone that need performance data on compatible AMCI stepper motors or guidelines for using a foreign stepper motor.

### 1.1 Sizing Your Motor

Your motor choice is based on the output torque you need, the mounting space you have, and your budgetary constraints. Torque curves for all of the motors available from AMCI are presented on the following pages. Torque curves show the performance of the motor at 4 Arms, which is the maximum setting for the SD4840E2.

There are a few things to remember when choosing your motor based on torque curves.

- 1) The torque curves in this manual are for the SD4840E2. You cannot use these curves to accurately determine the amount of torque from an AMCI motor when it is attached to a different drive. Nor can you accurately determine the amount of torque from a motor when attached to an SD4840E2 if its torque curves were generated using a different drive. In general, if an output bus of the foreign drive is not the same as the voltage supplied to the SD4840E2, then the torque curves will be less accurate at higher speeds.
- 2) Make sure that the motor can provide the needed torque over the entire speed range of your application. Available torque drops as speed increases, so evaluate the motor's torque at its highest operating speed.
- 3) All of the torque curves show when the motor's windings are attached to the SD4840E2 in parallel. Parallel attached motors have the advantage of more torque at high speeds when compared to series attached motors.

A simple guideline is to use the largest motor your mounting space and budgetary constraints allow. Because the  $I^2R$  losses in the motor's windings manifest themselves as heat, the maximum allowable motor temperature limits the motor's current. Using the largest motor possible may allow you to use a lower current setting on the SD4840E2 drive. This lowers the  $I^2R$  losses, and the operating temperature of the motor, which increases the motor's life.

### 1.2 Determining Your Motor Current Setting

Your motor current setting is based on the amount of torque needed from the motor. If you decide to use a lower current setting than the value listed in the curve, be aware that a reduction in current proportionally reduces the holding torque. However, a reduction in current may not lead to a proportionally reduction in torque at high speeds, especially if the motor is series connected. At high speeds, motor torque is limited by the voltage bus of the drive and the inductance of the motor. (The simplest explanation is that the drive does not have enough time to establish the full current through the motor before it must switch the current to the winding.)

Because of this, its difficult to calculate the exact amount of high speed torque a motor will give you when you reduce its current setting. Its often easier to determine your optimum current setting by testing your machine at various current settings and then deciding which setting gives you the best performance.

#### 1.2.1 A Note on Microstepping

Many microstepping drives control the peak current through the motor. At low speeds, this type of current control drops the available torque of a micro-stepped motor to approximately 70.7% of that available when the motor is full stepped.

However, the SD4840E2 controls the RMS current through the motor. Therefore, the current supplied by the SD4840E2 when microstepping is always the power equivalent of the full step current. This means that the motors' full torque is always available. At very low speeds, the SD4840E2 automatically switches to peak current control to prevent motor damage.

**1.3 Torque and Power Curves**

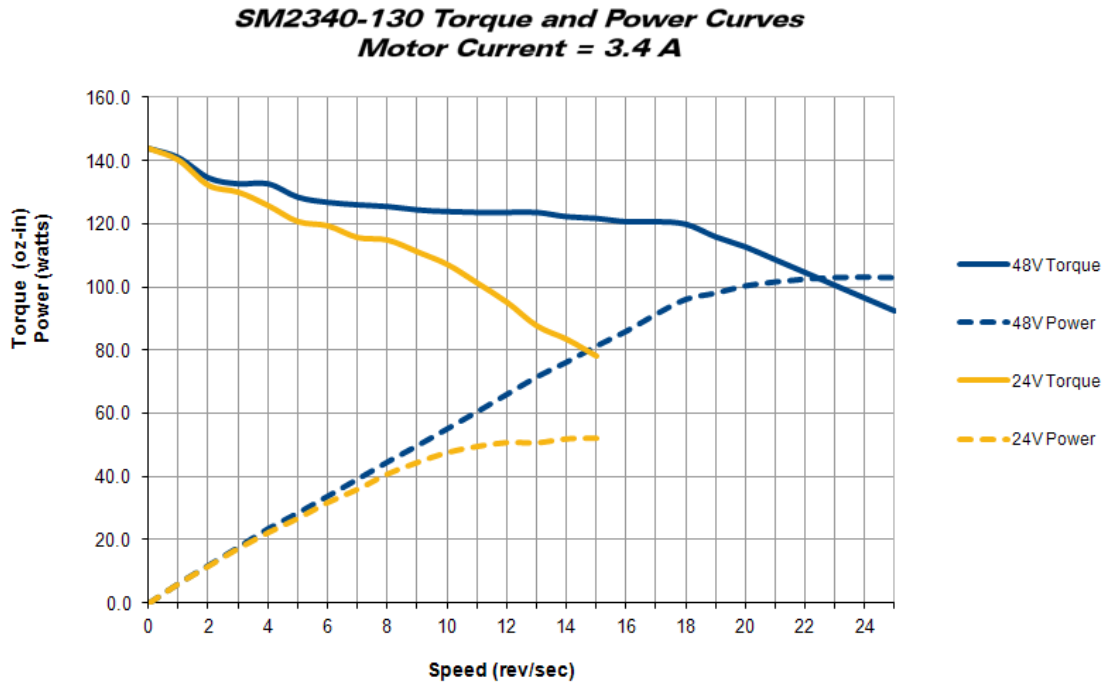


Figure T1.1 SM2340-130 Torque and Power Curves

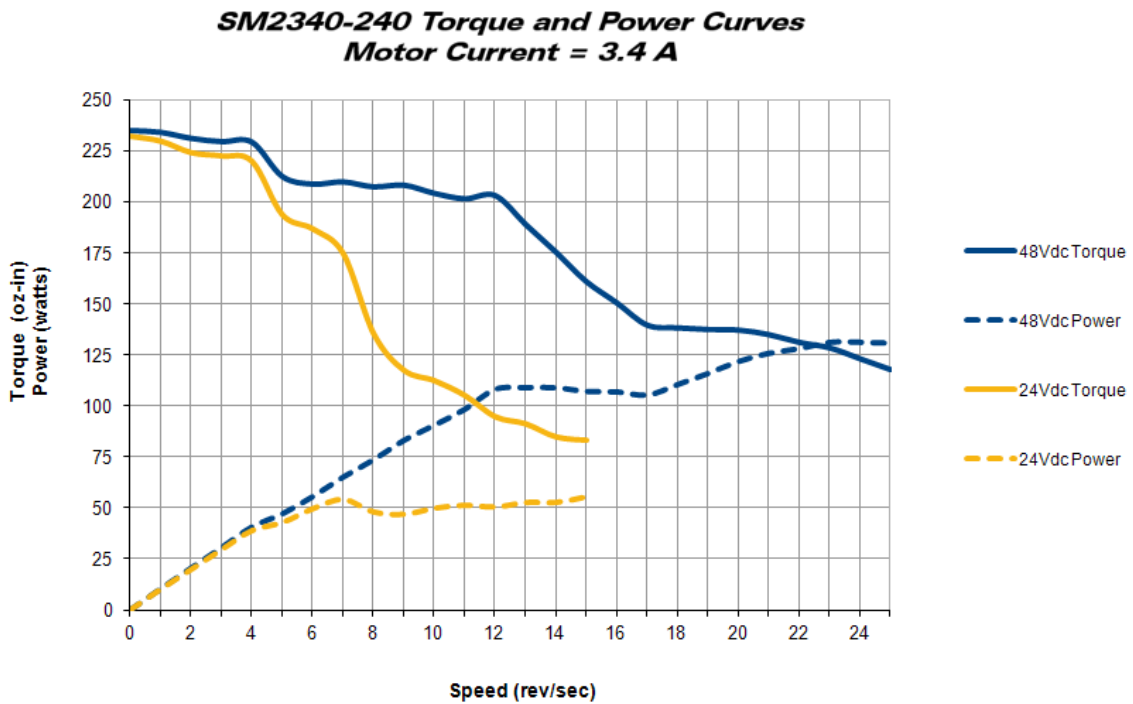


Figure T1.2 SM2340-240 Torque and Power Curves

1.3 Torque and Power Curves (continued)

**SM34-450 Torque and Power Curves**  
**Motor Current = 4.0A**

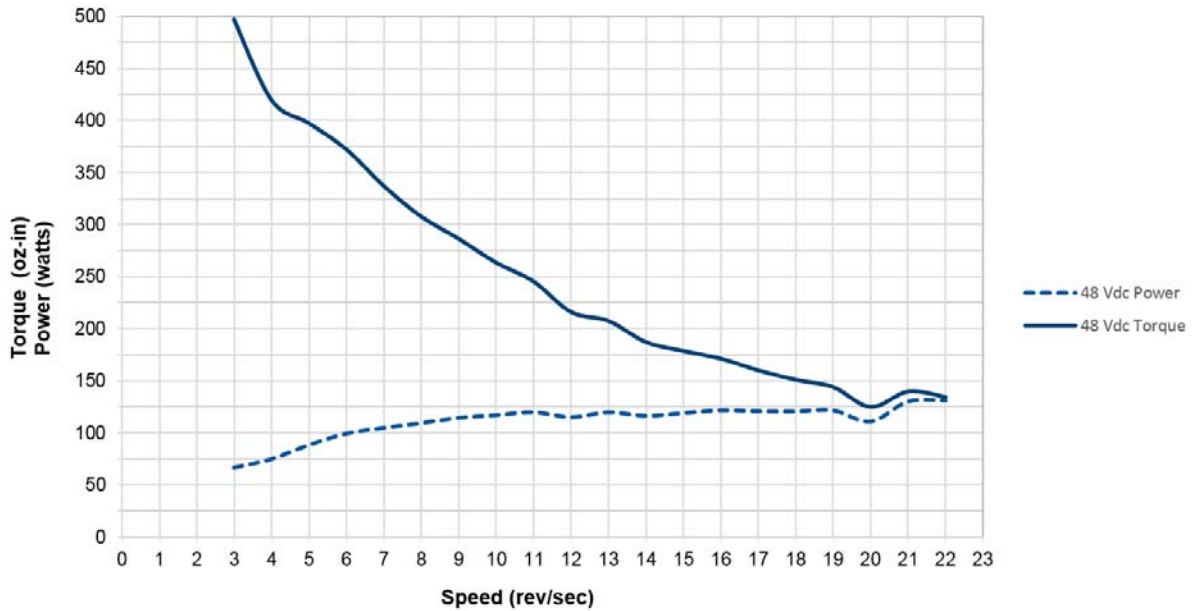


Figure T1.3 SM34-450 Torque and Power Curves

**SM34-850 Torque and Power Curves**  
**Motor Current = 4.0A**

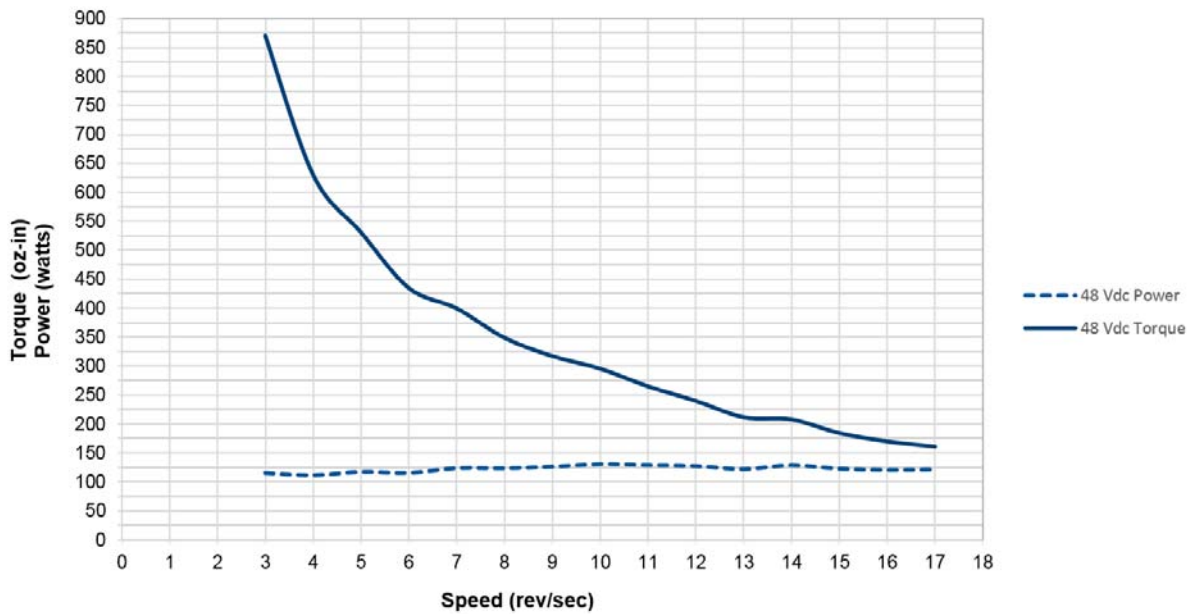


Figure T1.4 SM34-850 Torque and Power Curves



**1.4 Power Supply Sizing**

The power supply can be sized based on the power the motor must generate during its operation. As a general guideline, your supply should be able to produce 150% to 175% of the power the motor can produce. The power and torque curves on the previous pages can be used to determine the maximum power the motor can generate over its speed range.

Note that the power value that you should use is the *maximum* power value over the range of speeds that the motor will be operated at. The power generated by the motor decreases towards the end of its usable speed. Therefore, the power generated at your machine’s operating point may be less than the maximum the motor can generate at a lower speed.

**Example 1:** An SD2340-130 will be running at a maximum of 7 RPS and a 48 Vdc supply will be used. Based on the power curve in figure T1.1 on page 84, the combinations will generate a maximum of 40 Watts. Therefore a 48 Vdc supply with a power range of 60 W to 70 W can be used in the application.

**Example 2:** An SD2340-240 will be running at a maximum of 9 RPS and a 24 Vdc supply will be used. Based on the power curve in figure T1.2 on page 84, the power at this speed is 45 W, but the maximum power over the entire speed range is 55 W, which occurs at 7 RPS. Therefore, the 55 Watt value should be used, and the 24 Vdc supply should be able to generate 83 W to 97 W.

The two tables below shows the suggested power supply sizes based on the maximum power the motor can generate over its entire speed range.

		SM2340-130			SM2340-240			SM34-450 & SM34-850		
		Motor Power	150% Supply	175% Supply	Motor Power	150% Supply	175% Supply	Motor Power	150% Supply	175% Supply
Supply Voltage	24 Vdc	53W	80 W	93W	57 W	86 W	100 W	Not Applicable		
	48 Vdc	105 W	158 W	184 W	135 W	203 W	237 W	135 W	203 W	237 W

Table T1.1 SM23 Suggested Power Supply Ratings

**Regeneration (Back EMF) Effects**

All motors generate electrical energy when the mechanical speed of the rotor is greater than the speed of the rotating magnetic fields set by the drive. This is known as regeneration, or back EMF. Designers of systems with a large mass moment of inertia or high deceleration rates must take regeneration effects into account when selecting power supply components.

All AMCI stepper motors are all low inductance motors. Back EMF is typically not an issue unless there is a gearhead attached to the motor and it is driven by hand. In these instances, the motor acts as a generator. With the speed of the motor multiplied by the ratio of the gearhead, this can lead to large enough voltage spikes to damage the attached power supply.

The first line of defense against regenerative events is an appropriately sized power supply. The additional capacitance typically found in a larger supplies can be used to absorb the regenerative energy. If your application has high deceleration rates, then a supply that can deliver 175% of peak motor power should be used.

The second line of defense is a regeneration resistor, also known as a braking resistor. Braking resistors, and their control circuitry, are built into AMCI AC powered drives. They are not included in the SD4840E2 products because of the limited ability to dissipate the heat generated by the resistor. An external braking resistor and control circuitry can be added to the system if needed.

## TASK 2

# INSTALLING THE SD4840E2

This chapter gives detailed information on installing the SD4840E2. The SD4840E2 is designed to be DIN rail mounted. The network connection allows the drive to be mounted near the motor, which minimized power loss in the motor cable.

## 2.1 Safe Handling Guidelines

### 2.1.1 Prevent Electrostatic Damage

#### ⚠ CAUTION

Electrostatic discharge can damage the SD4840E2. Follow these guidelines when handling the module.

- 1) Touch a grounded object to discharge static potential before handling the module.
- 2) Work in a static-safe environment whenever possible.
- 3) Wear an approved wrist-strap grounding device.
- 4) Do not touch the pins of the I/O connector.
- 5) Do not disassemble the module.
- 6) Store the module in its anti-static bag and shipping box when it is not in use.

### 2.1.2 Prevent Debris From Entering the Module

#### ⚠ WARNING

During DIN rail mounting of all devices, be sure that all debris (metal chips, wire strands, tapping liquids, etc.) is prevented from falling into the module. Debris may cause damage to the module or unintended machine operation with possible personal injury. The DIN rail for the modules should be securely installed and grounded before the modules are mounted on it.

### 2.1.3 Remove Power Before Servicing

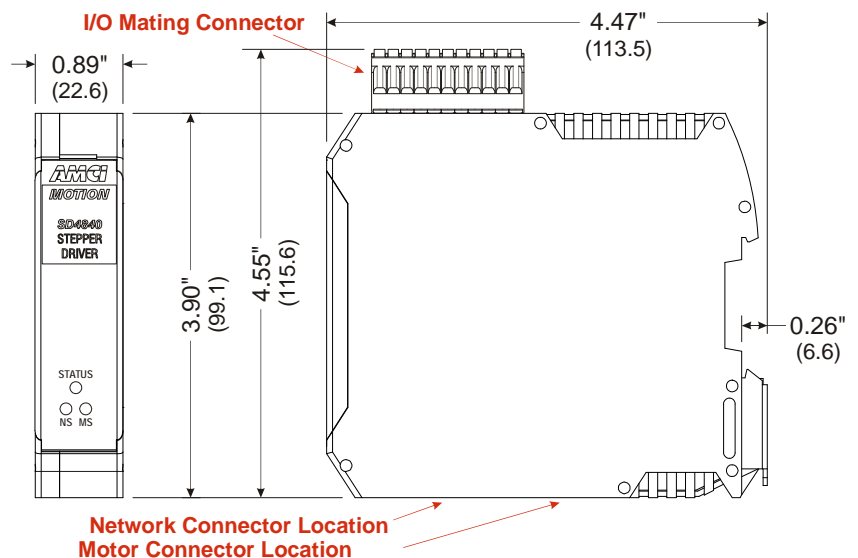
#### ⚠ WARNING

Remove power before removing or installing any modules.

## 2.2 Mounting

### 2.2.1 Dimensions

Figure T2.1 shows the dimensions of an SMD4840E2.



**2.2 Mounting (continued)**

**2.2.2 Minimum Spacing**

As shown in figure T2.2, you must maintain a minimum spacing of 2 inches (50.8 millimeters) from enclosure walls, wireways, adjacent equipment, etc. for adequate system ventilation.

Also note that the SD4840E2 drives must be mounted in the orientation shown in the figure. Mounting the system in any other orientation will decrease the efficiency of the ventilation slots in the top and bottom of each module which may lead to system overheating and malfunction.

When installing multiple SD4840E2 drives that are all running at the full 4.0 A motor current, it is possible for the drives to overheat. It is difficult to calculate when this can occur because it is based on not only the current and duty cycle of the motor, but also such variables as enclosure size and ambient temperature. If overheating does occur, you have two choices. You can install a cooling fan beneath the drives to force additional air up through the modules or you can increase the space between the drives to limit mutual heating.

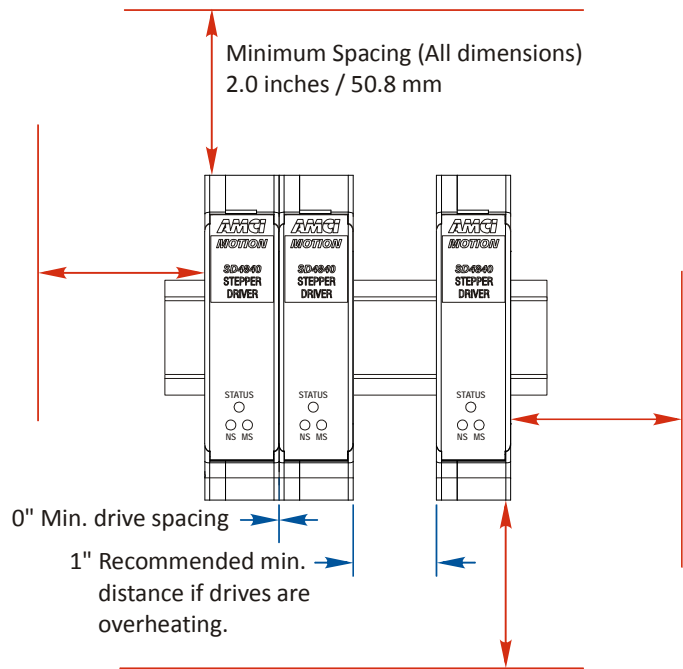


Figure T2.2 Ventilation Spacing

The SD4840E2 has a bit in the network data that signals when the module is close to overheating. This bit is the Temperature Warning bit, and is available in Status Word 2 of the network input data while in Command Mode. This bit is further explained in the *Status Word 1 Format* section starting on page 80.

**2.2.3 Mounting the SD4840E2 Module**

Mounting an SD4840E2 is a very simple process thanks to the design of the enclosure.

- 1) Engage the top of the clip on the back of the enclosure with the top of the DIN rail
- 2) Rotate the module down until the metal bracket snaps on to the DIN Rail.

Once all of your drives are installed, it is strongly suggested to use the end caps from Phoenix Contact with the part number of 271 37 80 to secure the modules on the DIN Rail. These end caps prevent the drives from sliding along the DIN rail if it is subjected to shock or vibration during machine operation.

### 2.3 I/O Connector Pin Out

The I/O Connector is located on the top of the module. The mate for this connector is included with the SD4840E2. It is available from AMCI under the part number MS-2x11 and is also available from Phoenix Contact under their part number 173 88 98. Figure T2.3 shows the pin out for the I/O connector.

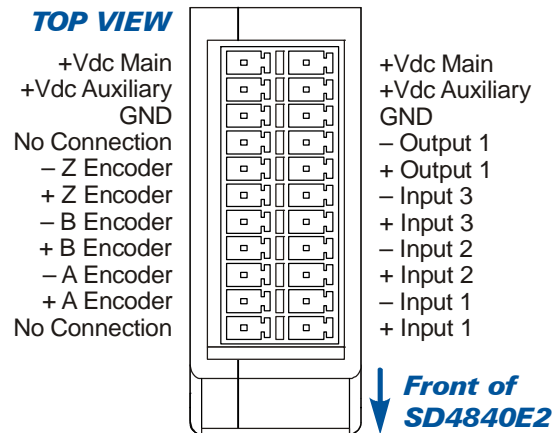


Figure T2.3 I/O Connector

### 2.4 Power Wiring

The SD4840E2 accepts 24 to 48Vdc for its input power.

- ▶ The +Vdc Main pin powers the entire module. This includes the control electronics and the motor drive circuitry. AMCI strongly suggests using 18 AWG or larger wire for the power connections. The MS-2x11 connector will accept up to 16 gauge wire.
- ▶ The +Vdc Auxiliary pin powers the control electronics only. Using this pin is optional. By applying power to this pin, you can remove power from +Vdc Main to remove power from the motor only. The control electronics will remain powered and the drive will not lose its network connection. This feature was incorporated into the design of the SD4840E2 for customers that must remove power from the motor due to safety considerations. Maximum current requirements on the +Vdc Auxiliary pin is 70 mA at 24Vdc.

The +Vdc Main, +Vdc Auxiliary, and GND connections each have two pins on the MS-2x11 I/O connector that are internally connected together. You can wire to either or both pins for each connection.

**CAUTION**

The I/O connector is rated for a maximum current of 8 amps per pin. It is possible to daisy chain the +Vdc Auxiliary connections, but AMCI strongly suggests avoiding this practice for the +Vdc Main connections. It is strongly suggested to run separate wires from your power supply to each drive for the +Vdc Main connections.

**CAUTION**

Do not apply 120 Vac to any pins of the SD4840E2. If this occurs, the unit will be damaged and you will void the unit's warranty.

2.4 Power Wiring (continued)

**WARNING**

The SD4840E2 units do not have a circuit to limit inrush current when power is applied. If the power supply voltage is applied through the switching of contacts, damage to the contacts or contact welding may occur.

- ▶ Use a power supply that limits the peak output current to a value below the contact switching limit.
- ▶ The best practice is to remove switching contact from the power supply to the SD4840E2 entirely. Remove power from the SD4840E2 by switching off its power supply.

**WARNING**

Remove power from the SD4840E2 before servicing. Do not connect or disconnect the MS-2x11 connector while power is applied.

Figure T1.4 below shows how to wire power to the SD4840E2 units. Note that +Vdc Auxiliary is only used when you introduce a circuit for removing power from the motor.

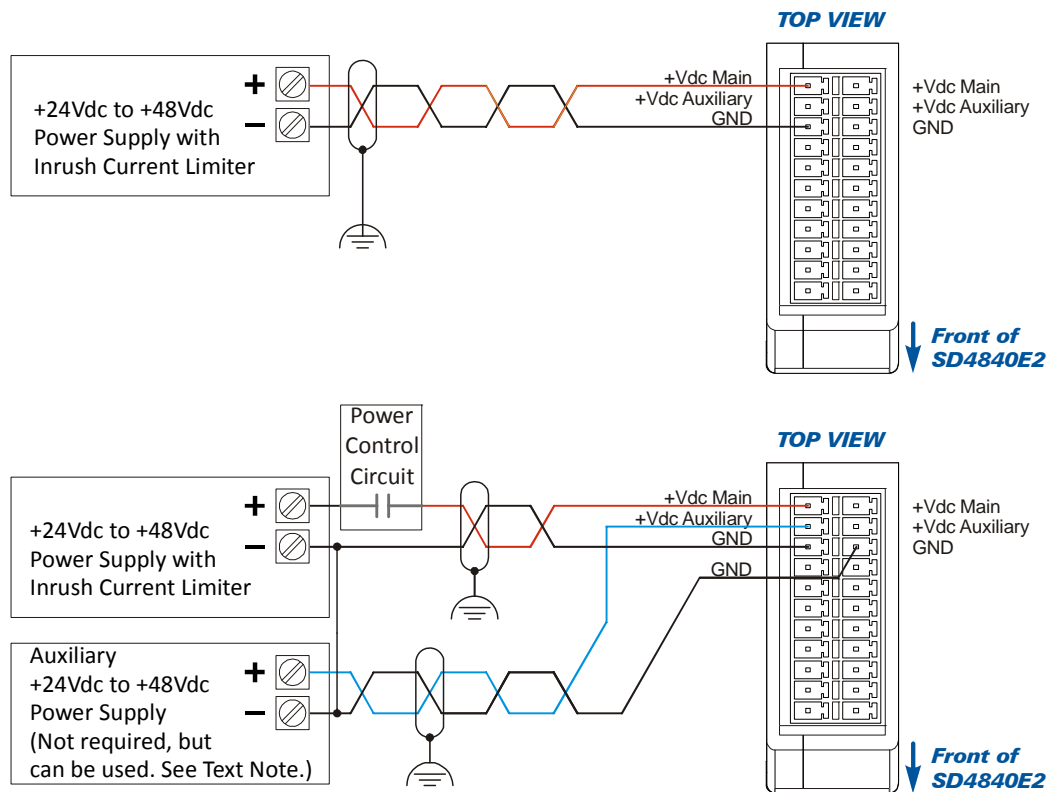


Figure T2.4 SD4840E2 Power Wiring

**NOTE**

Twisted pair cable with an overall shield should be used for all power supply wiring to limit EMI. The shield must be connected to the earth ground point used by the power supply. These cables must be routed away from all of the module's I/O to further limit the possibility of injecting noise into the I/O signals.

**NOTE**

As shown in the figure above, it is possible, *but not required*, to use a second power supply when powering the +Vdc Auxiliary power input. The +Vdc Auxiliary pin can also be powered by the +Vdc Main supply. If a separate supply is used, it must have a nominal output voltage of 24 to 48 Vdc. The output voltage does not have to match the voltage applied to the +Vdc Main power input. The commons of the two supplies must be electrically tied together.

### 2.5 Input Wiring

Figure T2.5 below shows how to wire discrete DC sourcing or sinking sensors to Inputs 1, 2, and 3 of the SD4840E2.

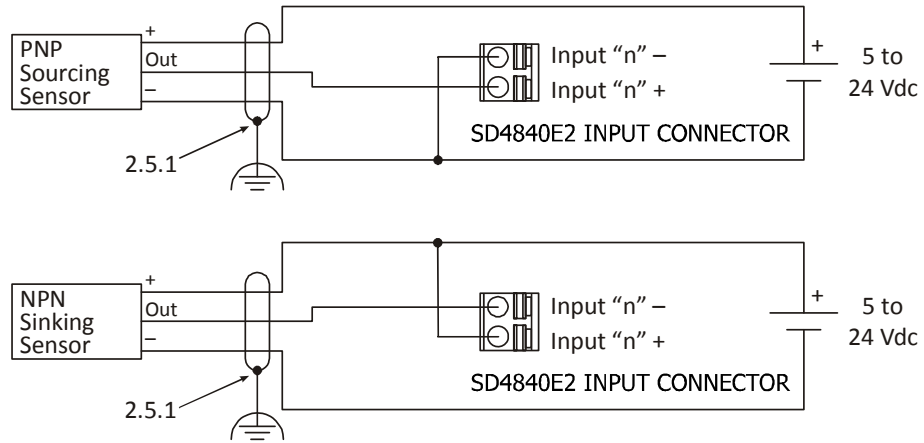


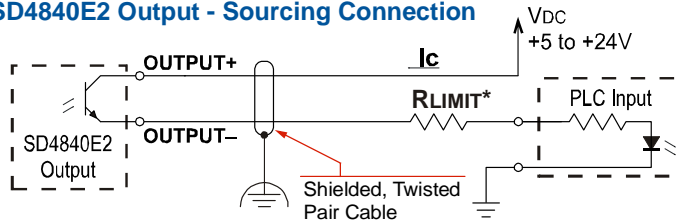
Figure T2.5 Input Wiring

#### 2.5.1 Cable Shields

Because they are low power signals, cabling from the sensor to the SD4840E2 should be done using a twisted pair cable with an overall shield. The shield should be grounded at the end when the signal is generated, which is the sensor end. If this is not practical, the shield should be grounded to the same ground bus as the SD4840E2.

### 2.6 Output Wiring

#### SD4840E2 Output - Sourcing Connection

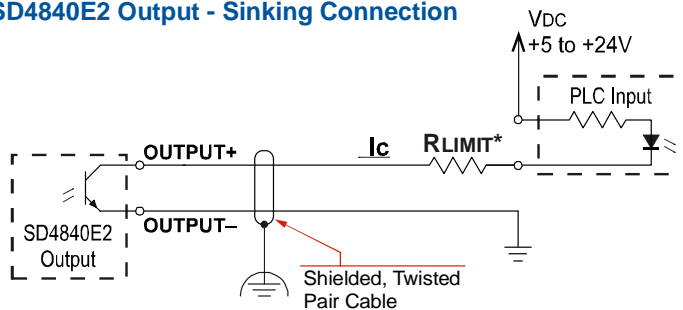


The SD4840E2 output is an optically isolated transistor that is capable of driving a typical PLC input. Both ends are uncommitted, so it can be wired as a sinking or sourcing output.

#### Electrical Specifications

VDC max: 30Vdc	VCE SAT: 1Vdc @ 20 mA
Ic max: 20 mA	Power Dissipation: 20 mW max.

#### SD4840E2 Output - Sinking Connection



#### RLIMIT

A resistor may be needed to limit the current through the output. The value, and power rating of the resistor is dependent on the value of Vdc, the voltage drop across the input, and the current requirements of the input.

Figure T2.6 Output Wiring

2.7 Encoder Wiring

2.7.1 Differential Wiring

The figure below shows how to wire a 5 Vdc differential encoder to the SD4840E2. There is no standard when it comes to the color code of the encoder’s wires. A document named ‘encoder specs’, is available on the AMCI website ([www.amci.com](http://www.amci.com)) that lists the color codes of encoders used by AMCI. The direct link to the document is: <https://www.amci.com/files/6714/5495/8474/encoder-specifications-rev.pdf>

Because they are low power signals, cabling from the encoder to the SD4840E2 should be done using a twisted pair cable with an overall shield. The shield should be grounded at the end when the signal is generated, which is the encoder end. If this is not practical, the shield should be grounded to the same ground bus as the SD4840E2.

As shown in the following diagram, the No Connection pins on the SD4840E2 can be used to make the connections between the encoder and its external power supply. This may eliminate the need for an external terminal block when wiring the encoder to the unit.

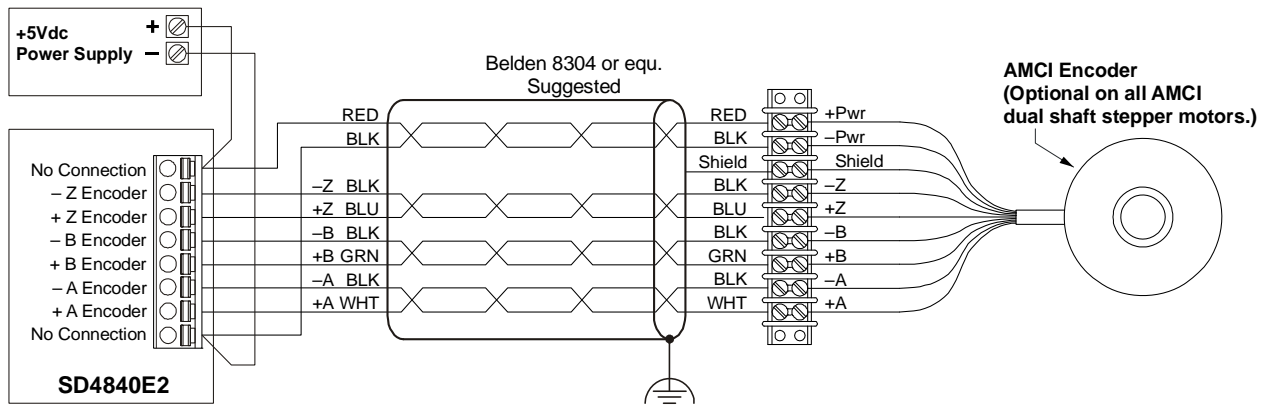


Figure T2.7 Sample Differential Encoder Wiring



## 2.7 Encoder Wiring (continued)

### 2.7.2 Single Ended Wiring

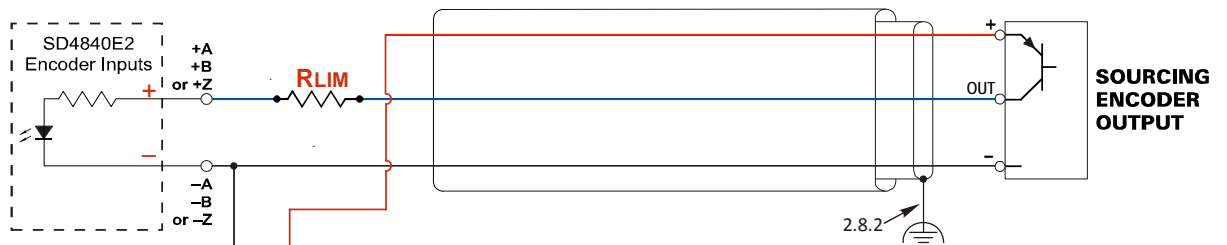
Figure T2.8 below shows how to wire the encoder inputs to both a single ended sourcing and single ended sinking encoder outputs.



The encoder inputs on the SD4840E2 are rated for 5 Vdc only. You must use a current limiting resistor on each input if the outputs of your encoder are greater than 5 Vdc. Appropriate current limiting resistors are shown in the figure below.

Because they are low power signals, cabling from the encoder to the SD4840E2 should be done using a twisted pair cable with an overall shield. The shield should be grounded at the end when the signal is generated, which is the encoder end. If this is not practical, the shield should be grounded to the same ground bus as the SD4840E2.

#### SD4840E2 Encoder Input Connection to Sourcing Encoder Output



**RLIM:** The inputs are designed to accept 5Vdc but can use any voltage up to 24Vdc with the appropriate current limiting resistor. See the table above for the required resistor based on the supply voltage.

Supply	RLIM
5 Vdc	None
12 Vdc	2.0 Kohm
15 Vdc	2.4 Kohm
24 Vdc	4.7 Kohm

#### SD4840E2 Encoder Input Connection to Sinking Encoder Output

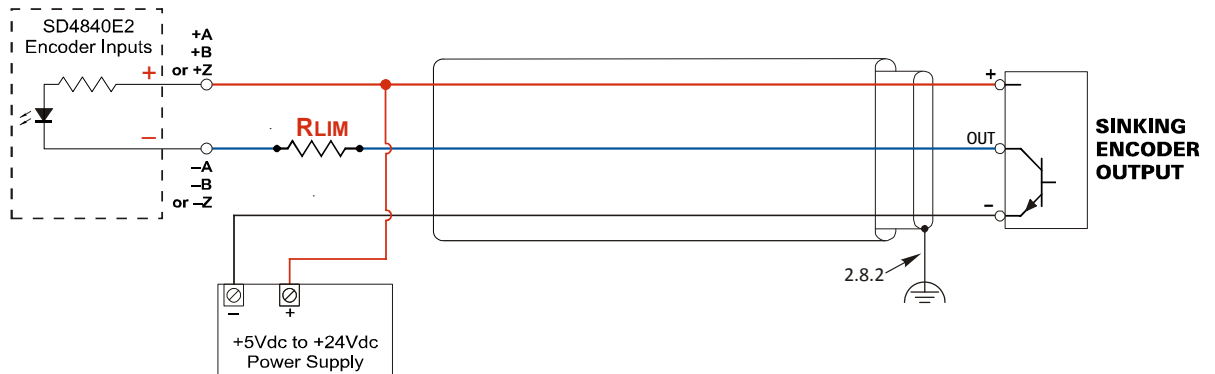


Figure T2.8 Single Ended Encoder Wiring

## 2.8 Installing the Stepper Motor

### 2.8.1 Outline Drawings

Outline drawings for all of our motors can be found on our website, [www.amci.com](http://www.amci.com), in the *Support -> User Manuals* section. A direct link to the section is: <https://www.amci.com/industrial-automation-support/user-manuals>. The drawings are available as Adobe Acrobat pdf files. A document that is simply called *wiring* lists all of the wiring color codes for all AMCI motors. If you do not have internet access contact AMCI and we will fax the information to you.

## 2.8 Installing the Stepper Motor (continued)

### 2.8.2 Mounting the Motor

All AMCI motor have flanges on the front of the motor for mounting. This flange also acts as a heatsink, so motors should be mounted on a large, unpainted metal surface. Mounting a motor in this fashion will allow a significant amount of heat to be dissipated away from the motor, which will increase the motor's life by reducing its operating temperature. If you cannot mount the motor on a large metal surface, you may need to install a fan to force cooling air over the motor.

Motors should be mounted using the heaviest hardware possible. AMCI motors can produce high torques and accelerations that may weaken and shear inadequate mounting hardware.

**NOTE** 

- 1) The motor case must be grounded for proper operation. This is usually accomplished through its mounting hardware. If you suspect a problem with your installation, such as mounting the motor to a painted surface, then run a bonding wire from the motor to a solid earth ground point near it. Use a minimum #8 gauge stranded wire or 1/2" wire braid as the grounding wire
- 2) Do not disassemble *any* stepper motor. A significant reduction in motor performance will result.

### 2.8.3 Connecting the Load

Care must be exercised when connecting your load to the stepper motor. Even small shaft misalignments can cause large loading effects on the bearings of the motor and load. The use of a flexible coupler is *strongly* recommended whenever possible.

### 2.8.4 Extending the Motor Cable

Even though it is possible to extend the cable length an additional forty feet, AMCI recommends installing the SD4840E2 as close to the motor as possible. This will decrease the chances of forming a ground loop, and has the added benefit of limiting the amount of power loss in the motor cable. If you must extend the cable, you should use a cable with twisted pairs 18 AWG or larger and an overall shield. The motor connector will accept up to 12 AWG wire. The exact gauge that you use will depend on the length of the run and expected temperature rise in your application. Belden 9552 is a suggested 18 AWG cable.

### 2.8.5 Installing the Motor Cable

**NOTE** 

- 1) All of the motor connections are high power, high voltage signals. Cable from the motor can be installed in conduit along with ac/dc power lines or high power ac/dc I/O. It cannot be installed in conduit with low power cabling such as I/O cabling or Ethernet cabling attached to the SD4840E2.
- 2) If you decide to extend the motor cable, treat the shield as a signal carrying conductor when installing the motor cable. Do not connect the shield to earth ground at any junction box.

## 2.9 Connecting the Motor

### 2.9.1 Motor Connector

The motor connector is included with the SD4840E2. Spares are available from AMCI under the part number MS-4M as well as directly from Phoenix Contact under their part number 187 80 37. Motor connections should be tight, as loose connections may lead to arcing which will heat the connector. Phoenix Contact specifies a tightening torque of 4.4 to 5.4 lb-in (0.5 to 0.6 Nm)

**NOTE** When powered, the motor connector may represent a shock hazard because the full DC input voltage may be present on its terminals.

**WARNING** Always remove power from the SD4840E2 before connecting or disconnecting the motor.

**NOTE**

- 1) Never connect the motor leads to ground or to a power supply.
- 2) Always connect the cable shield from your motor's cable to Earth Ground. It is best to connect the cable shields to the ground bus of the system. Do not connect the shields to the DIN rail. If you connect the motor shields to the DIN Rail and the grounding connection from the DIN rail to the Grounding Bus fails over time, then you will eventually have a condition where electrical noise is injected into the SD4840E2, which may result in future system errors.

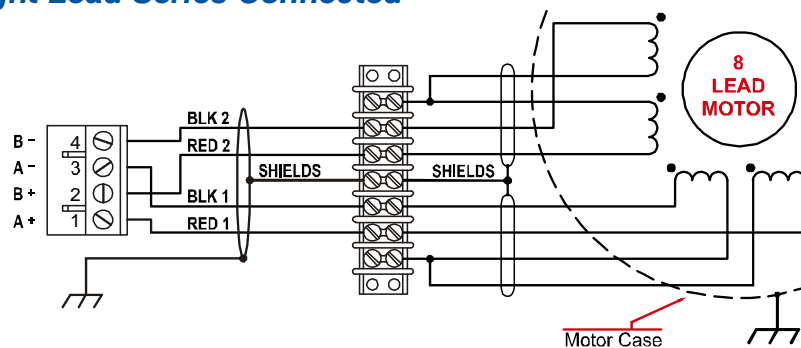
### 2.9.2 Motor Wiring

The SD4840 will work with many different motors, including those not sold by AMCI. This section assumes that you have already chosen your motor and you are looking for wiring information. No wire colors are given in the figures below because there is no single industry wide color coding scheme for stepper motors. You must refer to your motor data sheets for this information.

A wiring document for all of the motors ever sold by AMCI is available on our website. This single document contains all of the information necessary to connect any AMCI motor to any AMCI driver. It can be found in the *Support -> User Manuals* section. A direct link to the section is: <https://www.amci.com/industrial-automation-support/user-manuals>. The document is simply called "wiring".

Figure T2.9, which is continued on the following page, shows how to wire a motor to the SD4840E2 in series, parallel, or center-tap configurations.

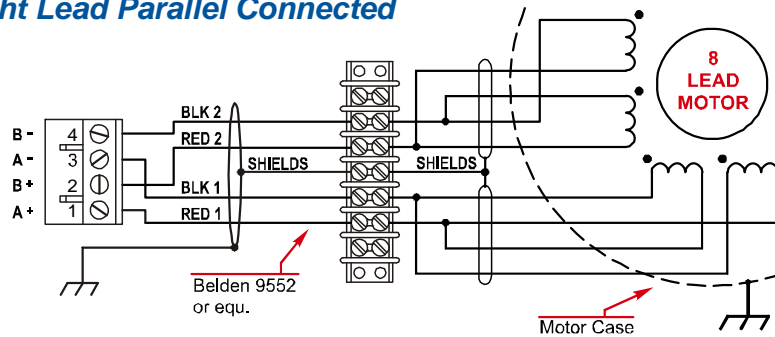
#### Eight Lead Series Connected



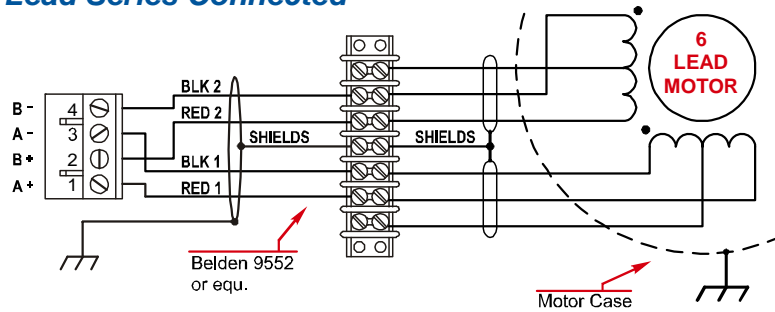
**2.9 Connecting the Motor (continued)**

*2.9.2 Motor Wiring (continued)*

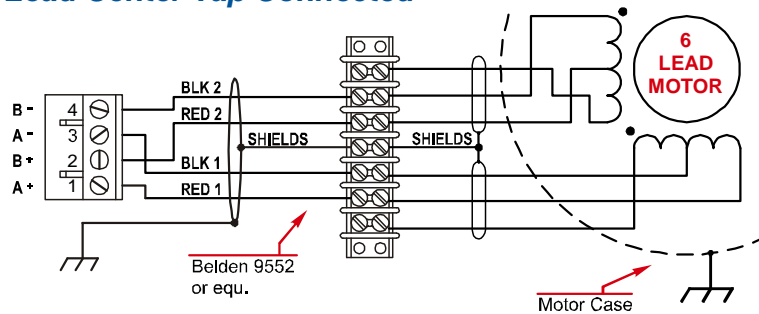
**Eight Lead Parallel Connected**



**Six Lead Series Connected**



**Six Lead Center Tap Connected**



**Four Lead Connected**

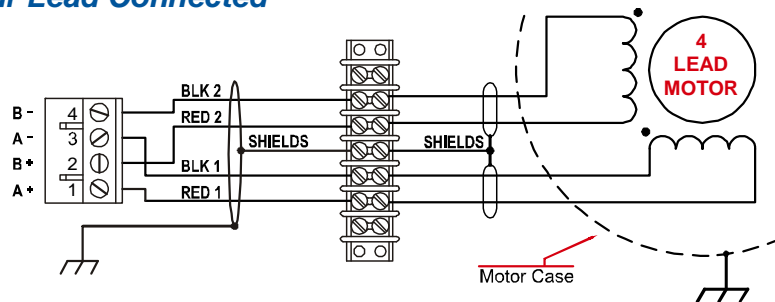


Figure T2.9 Motor Wiring

## 2.10 Ethernet Connections

The Ethernet connectors are located on the bottom of the unit. The connectors are standard RJ-45 jacks that will accept any standard 100baseT cable. Because the port can run at 100 Mbit speeds, Category 5, 5e, or 6 cable should be used.

The Ethernet ports on the units are “auto-sense” ports that will automatically switch between 10baseT and 100baseT depending on the network equipment they are attached to. The ports also have “auto switch” capability. This means that a standard cable can be used to connect the SD4840E2 to any device, including a personal computer.

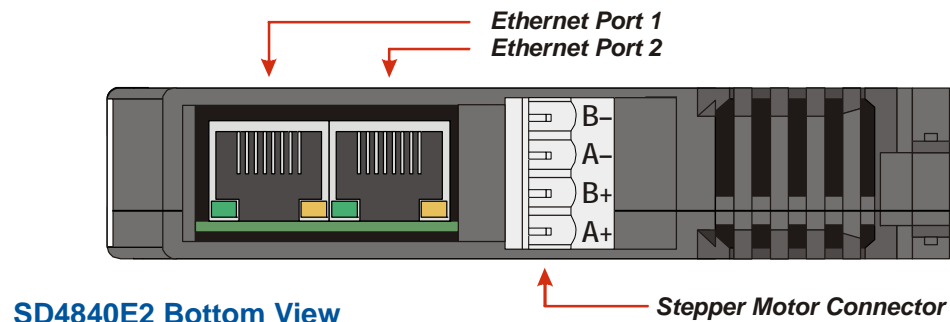


Figure R6.5 Ethernet Port Locations

The Network Status LED's are fully described on page 18 in the *Status LED* section of the Specifications reference.

## 2.11 EtherNet/IP Connections

### 2.11.1 Non-DLR Applications

The Networked Driver has two Ethernet ports with a built-in Ethernet switch connecting the two. In non-DLR applications, either port can be used to attach the unit to the network. The remaining port can be used to extend the network to another device if this would reduce wiring costs.

### 2.11.2 DLR Applications

In Device Level Ring applications, the Networked Driver functions as Beacon-Based Ring Nodes. In these applications, both ports are used when wiring the ring, daisy chaining from one unit in the ring to the next.

## 2.12 PROFINET Connections

### 2.12.1 Non-MRP Applications

The Networked Driver has two Ethernet ports with a built-in Ethernet switch connecting the two. In non-MRP applications, either port can be used to attach the unit to the network. Also, both ports can be used in non-MRP applications. For example, if two units are located some distance from your controller, then you need only run one cable from your controller to the first unit. The second unit can then be attached to the first with a short cable. There is no need to run two cables from your controller.

### 2.12.2 MRP Applications

In Media Redundancy Protocol applications, the Networked Driver functions as a Media Redundancy Client (MRC). In these applications, both ports are used when wiring the ring, daisy chaining from one unit in the ring to the next.

## 2.13 Modbus TCP Connections


The Networked Driver has two Ethernet ports with a built-in Ethernet switch connecting the two. Either port can be used to attach the unit to the network. The remaining port can be used to extend the network to another device if this would reduce wiring costs.

***Notes***

## TASK 3

# SET THE IP ADDRESS AND PROTOCOL

This section is intended for the engineer or technician responsible for setting the IP address of an AMCI SD4840E2.


**NOTE**  If you are using an SD4840E2 in a PROFINET environment, you can skip this task. The IP Address is written to the unit as part of the standard PROFINET configuration. You can proceed to Step 8, *PROFINET Network Configuration*, which starts on page 127


### 3.1 Determine the Best Method for Setting the IP Address

There are three methods for setting the IP address on an SD4840E2. Table T3.1 below outlines the available methods and when you can use them.

Method	Restrictions	Section
<i>Use Factory Default Settings</i>	1) The machine must use 192.168.0.xxx subnet. 2) The 192.168.0.50 address must be available.	3.2a
<i>Use the Embedded Web Server</i>	No restrictions on use. This is the preferred method. The internal webserver can be used to set the SD4840E2 to any IPv4 address. The IP address and protocol will be stored in nonvolatile memory and used on subsequent power-ups.	3.2b
<i>Use the AMCI NET Configurator Utility</i>	Unit must be initially configured for EtherNet/IP. The software can be used to set the SD4840E2 to any IPv4 address and protocol. The IP address and protocol choice will be stored in nonvolatile memory and used on subsequent power-ups.	3.2c

Table T3.1 Methods for Setting the IP Address

**NOTE**  There is a MAC address label on each SD4840E2 which has a writable surface. There is room on the label for writing the programmed IP address of the unit. It is a best practice to use this label to document the IP address of the unit in case it is ever repurposed.

**NOTE**  When using EtherNet/IP, the SD4840E2 units also supports the DHCP protocol. You will need an EtherNet/IP DHCP server, such as the one available from Rockwell Automation, in order to use this protocol. The AMCI Net Configurator utility offers the same functionality and should be used unless your company policy prevents you from installing third party utilities.

### 3.2a Use Factory Default Settings

The factory default address for the SD4840E2 units is 192.168.0.50 with a subnet mask of 255.255.255.0. The easiest way to verify this address is with the ping command as described in steps A.3 and A.4 of the Optional Task *Configure Your Network Interfaces* which starts on page 133.

If the driver does not respond to this address, then it may take some effort to determine the correct address. There is a label on the driver that lists the MAC address of the device. There is space on the label for noting the IP address of the device if it is changed. If the address was not documented, a program called Wireshark (<https://www.wireshark.org/>) can be used to determine the address of the driver.

**Task Complete**



**3.2b Use the Embedded Web Server**

**PREREQUISITE:** You must know the present IP address of the SD4840E2. The factory default address is 192.168.0.50.

**PREREQUISITE:** Task 2.4: *Power Wiring* found on page 89. You must be able to power the SD4840E2.

**PREREQUISITE:** Tasks: 2.10 and 2.11, 2.12, or 2.13, starting on page 97. You must attach your SD4840E2 to your computer.

**PREREQUISITE:** Optional Task A: *Configure Your Network Interfaces*. (page 133) The network interfaces on your computer must be on the same subnet before you can communicate with an SD4840E2.

**3.2b.1 Disconnect the SD4840E2 from the host controller and cycle power to the unit**

This ensures that the unit does not have any open connections to the host controller.

**3.2b.2 Start your web browser and connect to the SD4840E2**

The internal HTML pages should work with any browser. Once your web browser is running, enter the present IP address of the SD4840E2 into the address bar. The default address is 192.168.0.50. The unit will respond with the following page.

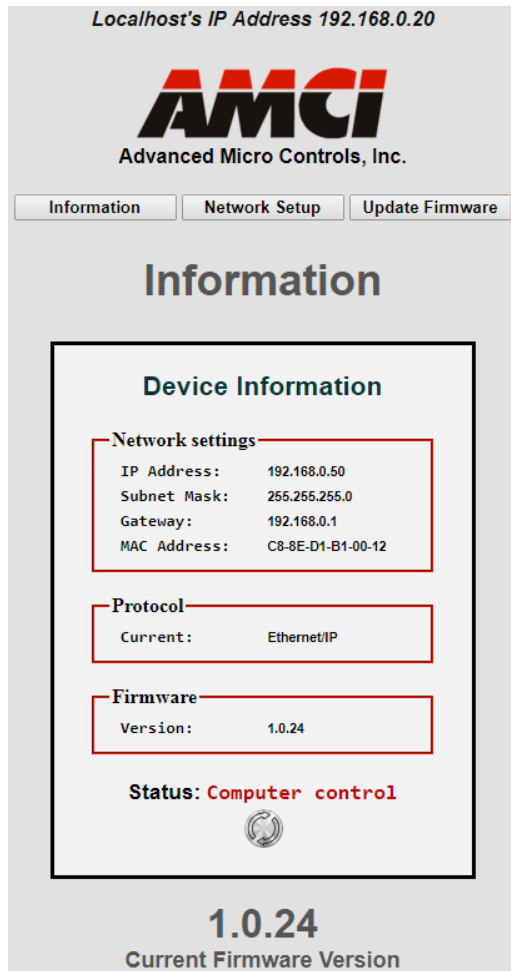


Figure T3.1 SD4840E2 Information Webpage

### 3.2b Use the Embedded Web Server (continued)

#### 3.2b.3 Network Setup Page

- 1) Click on the [Network Setup] button to switch to the Network Setup page shown below. This page shows the current IP address settings, as well as the configured protocol.

Localhost's IP Address 192.168.0.20

**AMCI**  
Advanced Micro Controls, Inc.

Information Network Setup Update Firmware

## Network Setup

### Device Configuration

#### Network settings

IP Address:

Subnet Mask:

Gateway:

#### Protocol

Ethernet/IP

Modbus-TCP

PROFINET (Protocol default address of 0.0.0.0 is set)

Write Configuration

**1.0.24**  
Current Firmware Version

Figure T3.2 SD4840E2 Network Setup Web Page

- 2) Enter your desired values into the IP Address, Subnet Mask, and Default Gateway fields.



The Default Gateway setting is not optional! It must be set to a valid address on the chosen subnet. Because the Default Gateway is often not used in device level networks, if you do not have a required value for it, AMCI suggests setting the Default Gateway to the IP address of your host controller.

- 3) If need be, click on the proper radio button to select the required protocol.
- 4) Click on the [Write Configuration] button to write the new configuration to the unit. If there are any errors with the data, the unit will display a warning message instead of accepting the new values.

**3.2b Use the Embedded Web Server (continued)**

*3.2b.3 Network Setup Page (continued)*

5) If the values are accepted, the following pages will be displayed while the data is being written to the unit.

**NOTE** ⚠ Wait for the pop up window to appear before cycling power to the SD4840E2. Cycling power before this window appears may corrupt the non-volatile memory of the unit. The SD4840E2 unit will also flash its Network Status LED red to indicate that power must be cycled.

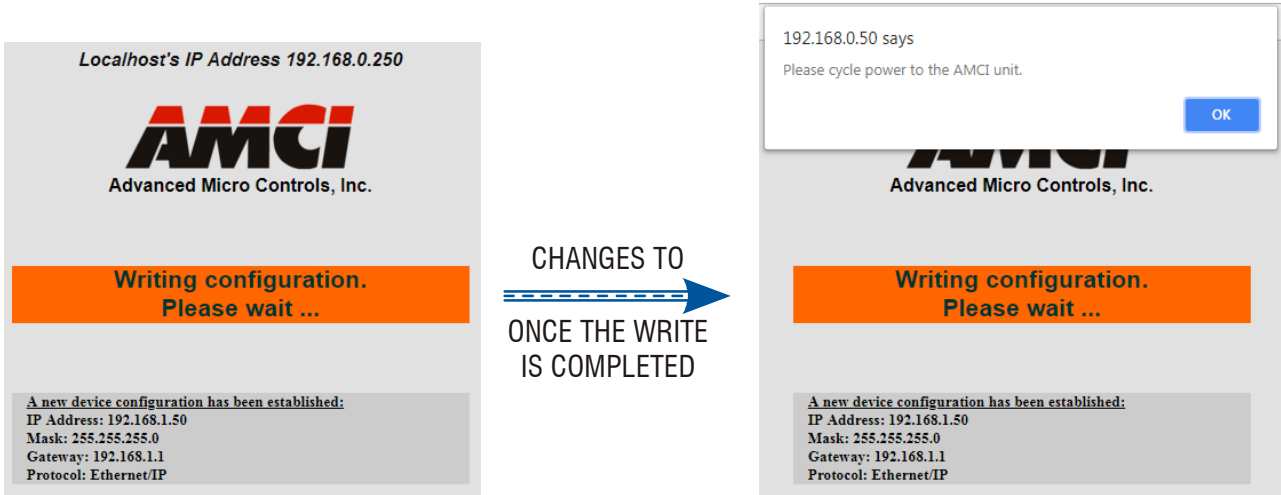


Figure R6.6 Write Configuration to Flash Memory Pages

6) Once instructed to, cycle power to the unit. You can now enter the new IP address into the address bar of your web browser to reconnect with the SD4840E2.

**Task Complete**

**3.2c Use the AMCI NET Configurator Utility**

**PREREQUISITE:** The unit must be configured for the EtherNet/IP protocol. EtherNet/IP is the communications protocol used by the AMCI NET Configurator utility. Once connected to the unit, you can select any IP address and protocol supported by the device.

**PREREQUISITE:** You must know the present IP address. The factory default address is 192.168.0.50.

**PREREQUISITE:** Task 2.4: *Power Wiring* found on page 89. You must be able to power the SD4840E2.

**PREREQUISITE:** Tasks: 2.10 and 2.11, 2.12, or 2.13, starting on page 97. You must attach your SD4840E2 to your computer.

**PREREQUISITE:** Optional Task A: *Configure Your Network Interfaces*. (page 133) The network interfaces on your computer must be on the same subnet before you can communicate with the unit.

*3.2c.1 Download the AMCI Net Configurator Utility*

The AMCI Net Configurator utility is available on our website, [www.amci.com](http://www.amci.com). The latest version available should be used. It can be found in our *Support* section under *Software*. The program exists as a ZIP file, and at the time of this writing, the link was “AMCI Configuration software for all networked products...”.

### 3.2c Use the AMCI Net Configurator Utility (continued)

#### 3.2c.2 Install the AMCI Net Configurator Utility

Once downloaded, simply extract the program from the ZIP file and run the program to install the AMCI Net Configurator utility on your computer. The software installs as most products do, giving you the option to change the file locations before installing the utility. Once the install is complete, a link to the utility is available on the Start Menu.

The install process only copies the utility to the designated location and creates links to the Start Menu. No changes are made to your registry settings.

#### 3.2c.3 Verify that Your Host Controller is Disconnected from the SD4840E2

EtherNet/IP is not a multi-master protocol. There can be only one bus master on the network at a time. In order to program the SD4840E2, the AMCI Net Configurator utility must act as a bus master. Therefore, physically disconnect your host controller from the SD4840E2 before starting the Net Configurator utility.

#### 3.2c.4 Apply or Cycle Power to the SD4840E2

Cycling power to the SD4840E2 will reset any connections it may have with the host controller.

#### 3.2c.5 Start the AMCI Net Configurator Utility

Double click on the utility's icon. A welcome screen similar to the one in figure T3.3 below will appear.

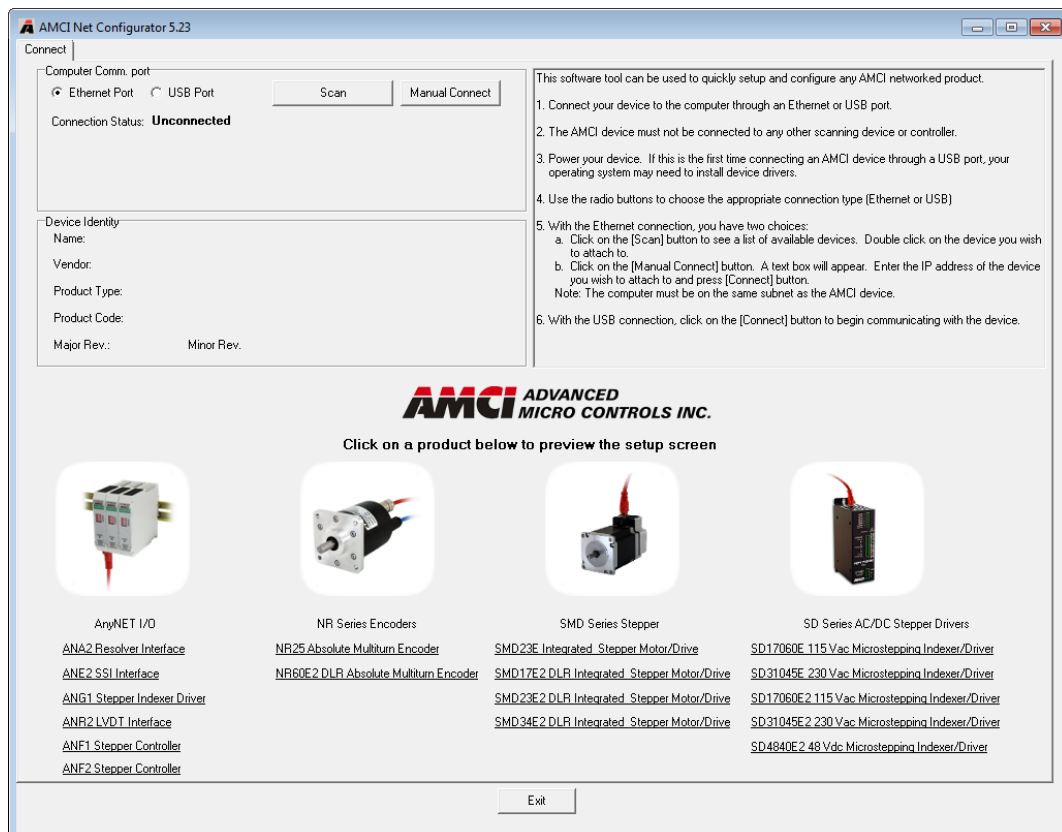


Figure T3.3 Net Configurator Welcome Screen

**3.2c Use the AMCI Net Configurator Utility (continued)**

**3.2c.6 Press the [SCAN] button and Connect to the SD4840E2**

Pressing the [Scan] button will open the window shown in figure T3.4. The SD4840E2 will appear in the scan list only if the unit and your network interface are on the same subnet. Optionally, you can press the [Manual Connect] button and enter the IP address of the unit.

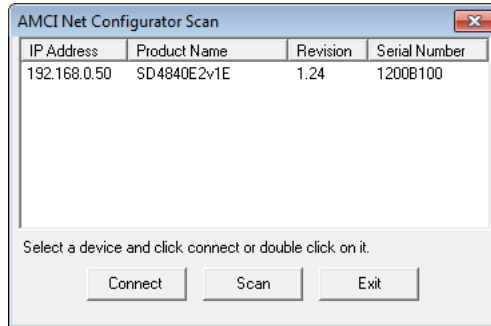


Figure T3.4 Scan for SD4840E2

If scanning for the SD4840E2, click on the IP Address of the unit and click on the [Connect] button. The Net Configurator utility will connect to the unit.

**3.2c.7 Click on the "Allow IP..." Checkbox to Access the IP Settings**

Figure T3.5 below shows the screen that results when you are connected to the SD4840E2. In order to change the IP Address of the unit, you must first click on the checkbox next to the text “Allow IP configuration changes. You will need to restart the device.” Once the checkbox is selected, the [Set IP Address] and protocol select buttons will be enabled.

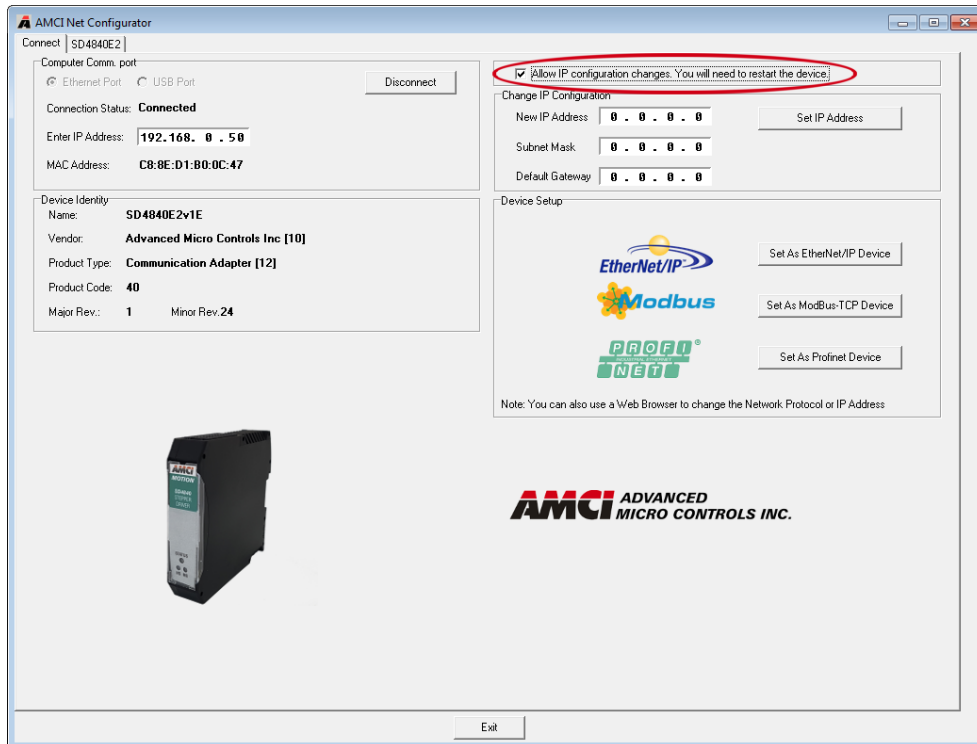


Figure T3.5 Enable IP Address Changes

### **3.2c Use the AMCI Net Configurator Utility (continued)**

#### *3.2c.8 Set the IP Address, Subnet Mask, and Default Gateway*

Enter your desired values into the IP Address, Subnet Mask, and Default Gateway fields.



The Default Gateway setting is not optional! In order to comply with the ODVA specification, it must be set to a valid address on the chosen subnet. Because the Default Gateway is often not used in device level networks, if you do not have a required value for it, AMCI suggests setting the Default Gateway to the IP address of your host controller.

#### *3.2c.9 Set the Communications Protocol*

The factory default protocol for the SD4840E2 is EtherNet/IP. In order to use the Modbus TCP or PROFINET protocols, simply click on the appropriate button.

#### *3.2c.10 Write the New IP Address to the SD4840E2*

Click on the [Set IP Address] button. If there is an error in the settings, the utility will tell you what is wrong. Once they are all correct, the utility will write the new IP address settings to the unit. These settings are automatically saved to nonvolatile memory.

#### *3.2c.11 Remove Power from the SD4840E2*

The new IP address will not be used until power to the unit has been cycled.

### **Task Complete**

### ***Notes***



## TASK 4 (EtherNet/IP Option)

### IMPLICIT COMMUNICATIONS WITH AN EDS

Many EtherNet/IP platforms support the use of EDS files to simplify the addition and configuration of devices. This chapter covers the installation and use of the EDS file for systems that are programmed with Rockwell Automation Studio 5000 version 20 and above. Other systems will follow a similar pattern. Consult your controller's documentation if you need additional information.

**Note:** Use of an EDS file is completely optional. The SD4840E2 can always be added to a system as a generic module. If you are using RSLogix 5000 version 19 and below, or RSLogix 500, adding the unit as a generic module is the only option available.

Using the EDS file simplifies configuration and adds named tags for all input and output data.

#### 4.1 Obtain the EDS file

All AMCI EDS files are located on our website at the following address:

➤ <http://www.amci.com/industrial-automation-support/configuration-files/>

Simply download the ZIP file and extract it to its own directory.

#### 4.2 Install the EDS file

##### 4.2.1 Start the EDS Hardware Installation Tool

1) Once Studio 5000 is running, in the menu bar select Tools → EDS Hardware Installation Tool. This will open the EDS Wizard.

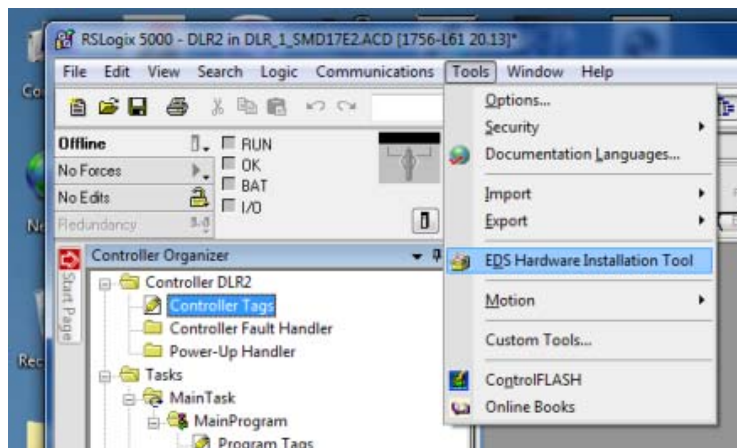


Figure T4.1 Opening the EDS Wizard

2) Click on [Next >] to advance to the Options screen.

**4.2 Install the EDS file (continued)**

*4.2.2 Install the EDS File*

1) On the Options screen, select the Register an EDS file(s) radio button and press [Next >].

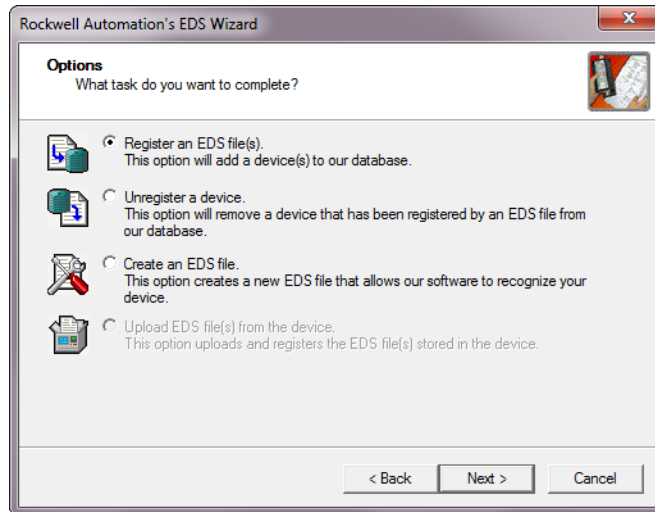


Figure T4.2 EDS Options Screen

2) The registration screen will open. Select the Register a single file radio button.

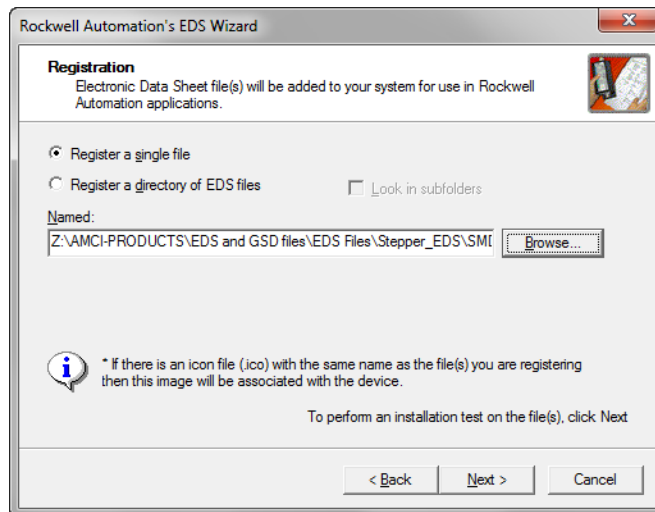


Figure T4.3 EDS Registration Screen

3) Click on the [Browse...] button and browse to the folder that contains the extracted EDS file you downloaded from the AMCI website. Select the EDS file and click on the [Open] button to return to the registration screen. Click on the [Next >] button to advance to the EDS file test screen.

## 4.2 Install the EDS file (continued)

### 4.2.2 Install the EDS File (continued)

- 4) Once at the EDS File Installation Test Results screen, expand the tree as needed to view the results of the installation test for the EDS file. You should see a green check mark next to the file name indicating that the EDS file is correct.

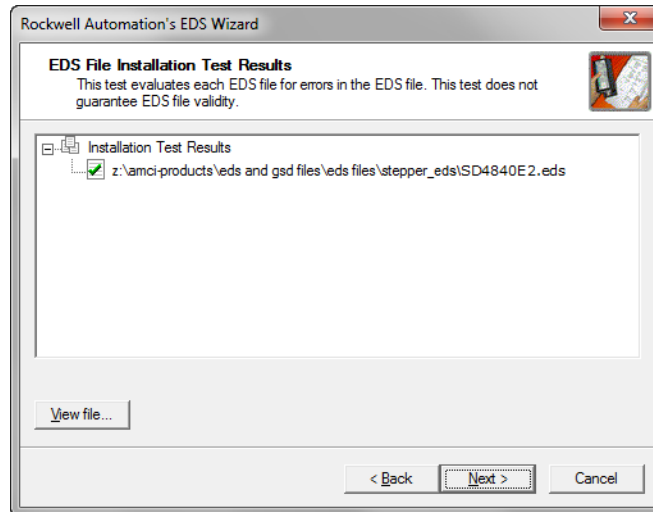


Figure T4.4 EDS Test Screen

- 5) Press on the [Next >] button to advance to the Change Graphic Image screen. The icon file supplied by AMCI is built into the EDS file. If your application requires custom icons that you have created, you can switch to these icons by clicking on the [Change icon...] button.

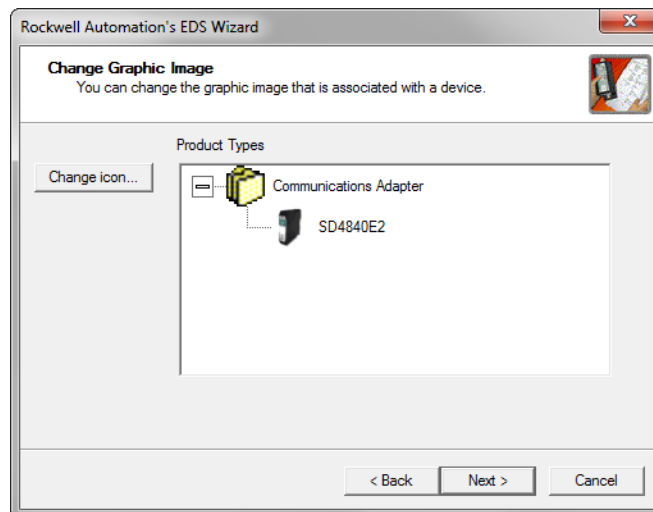


Figure T4.5 Change ECS Icon Screen

- 6) Click on the [Next...] button to advance to the completion screen. The Completion screen tells you that you have successfully completed the wizard.
- 7) Click on the [Finish] button to exit the EDS wizard.

### 4.3 Host System Configuration

Studio 5000 is used to configure both the ControlLogix and CompactLogix platforms. When using these platforms, you have the option of using a separate Ethernet Bridge module or an Ethernet port built into the processor.

If the Ethernet port is built into processor, the only step you have to take before adding an AMCI SD4840E2 is to create a new project with the correct processor or modify an existing project. Once this is done, the Ethernet port will automatically appear in the Project Tree. If you are using an Ethernet bridge module, you will have to add it to the I/O Configuration tree before adding the unit to your project.

Refer to your Rockwell Automation documentation if you need instructions for configuring the ethernet port.

### 4.4 Add the SD4840E2 to Your Project

You can add an AMCI SD4840E2 to the project once the Ethernet port (built-in or bridge module) is configured. As shown in figure T4.6 below, the Ethernet port will be listed under the I/O Configuration tree.

- 1) Right click on the Ethernet port and then click on “New Module...” in the pop-up menu.

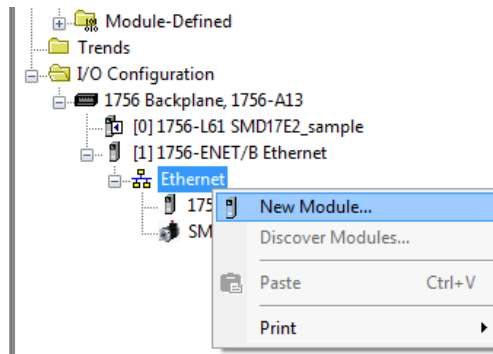


Figure T4.6 Adding an AMCI Ethernet Driver

- 2) In the resulting Select Module Type screen, select “Advanced Micro Controls Inc. (AMCI)” in the Vendor Filters. This will limit the results to catalog numbers from AMCI.
- 3) Select “SD4840E2” in the resulting list.
- 4) Click on the [Create] button to create the module.
- 5) Click on [Close] if necessary to close the Select Module Type screen.

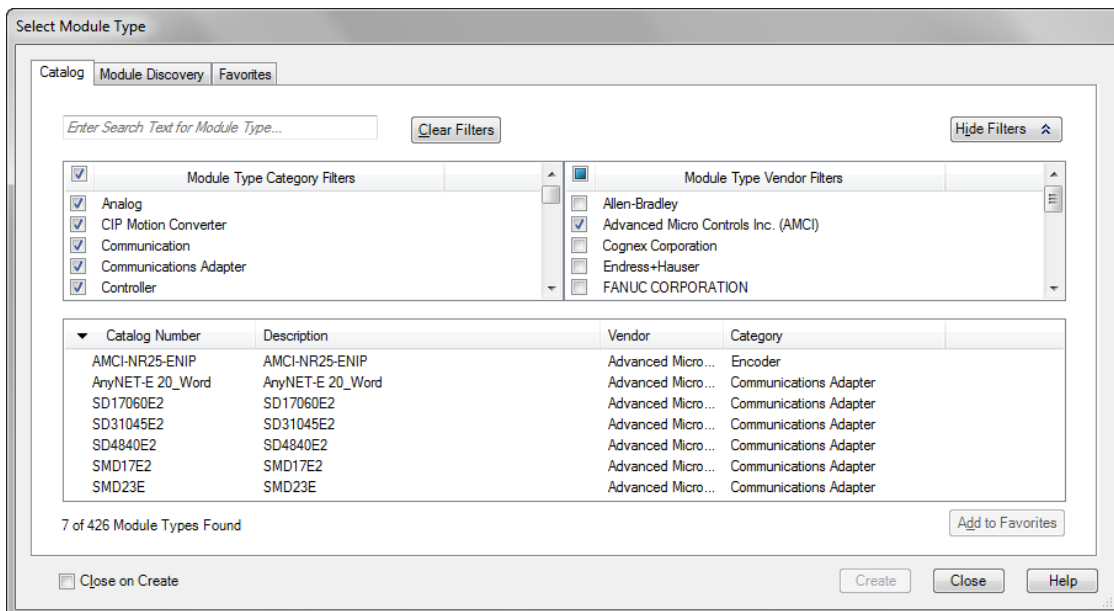



Figure T4.7 Selecting the Networked Driver

## 4.5 Configure the SD4840E2 Driver

If you are continuing from step 4.4, the resulting New Module screen is used to configure the network connection between the SD4840E2 and your controller. If you need to open the screen to perform this task at a later time, right click on the unit in the project tree and then select “Properties” from the drop-down menu

**NOTE**  Tabs that are not listed in the steps below are filled with reasonable defaults by the EDS file.

### 4.5.1 General Tab

The Name, Description, and IP address of the device must be specified here. The [Change...] button allows you to change the Module Definition if needed.

### 4.5.2 Connection Tab

The default RPI time is eight milliseconds. This value can be changed in this tab.

### 4.5.3 Configuration Tab

The Configuration tab is used to define the configuration data that is written down to the SD4840E2 when the device connects to the network. You can also click on the [Apply] button to write down the configuration data to the unit at any time.

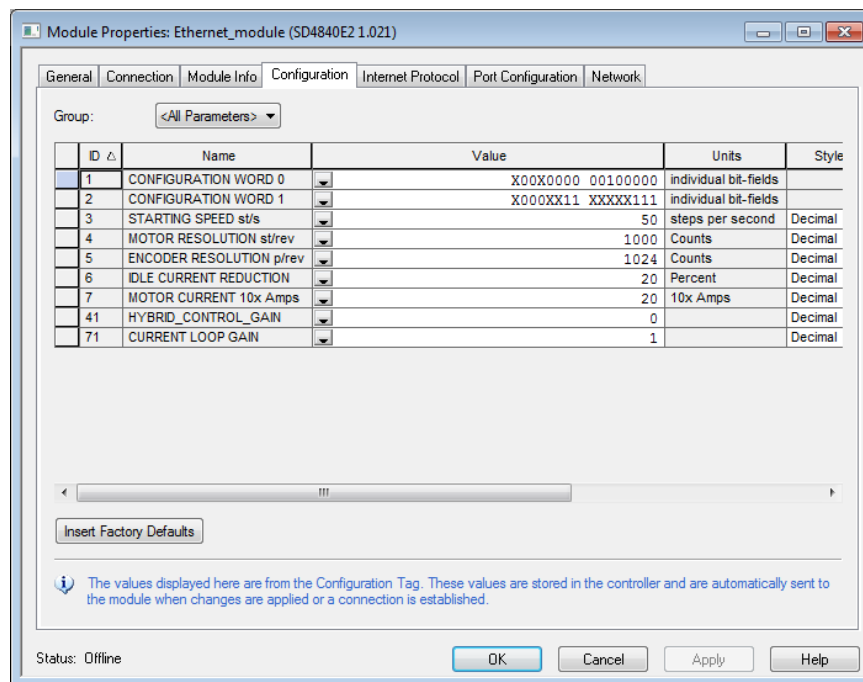




Figure T4.8 Networked Driver Configuration with EDS File

The EDS file defines tags that are used to configure the SD4840E2. These tags follow the format of the Configuration Data given in reference 5, *Configuration Mode Data Format*, starting on page 55.

**NOTE**  When using an EDS setup, the command and response data must be sent as 32-bit binary values. Bit 7 of Configuration Word 1, *Binary\_Endian*, should be set to “0” and bits 8 and 9, *Binary\_Output\_Format* and *Binary\_Input\_Format*, should both be set to “1”.

**NOTE**  When using the EDS file, the Starting Speed is transmitted as a double integer value. The parameter does not use the multi-word format outlined in Configuration Mode Data Format reference chapter.

**4.6 Buffering the I/O Data**

Input and output data is transferred asynchronously to the program scan. The input data tags should be buffered with Synchronous Copy File instructions to guarantee stable data during the program scan.

The data from the SD4840E2 is updated is asynchronously to the program scan. The following rung ensures that the data from the driver does not change in the middle of the ladder logic program by copying it to the internal tag array buffered\_SD4840E2\_data[0] through buffered\_SD4840E2\_data[9].

It is these buffered registers that your ladder logic program should use when referencing the SD4840E2's input data.

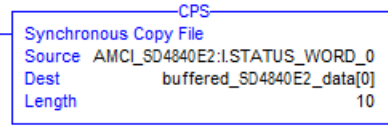


Figure T4.9 Buffer I/O Data

- When copying input data, the data can be converted from byte to integer format by specifying an integer array as the destination for the instruction. The array must contain at least ten integer elements. The length of the copy should be ten.

## TASK 5 (EtherNet/IP Option)

# IMPLICIT COMMUNICATIONS WITHOUT AN EDS

An AMCI SD4840E2 requires a host controller to issue configuration and motion commands to it. This chapter tells you how to configure implicit connections in EtherNet/IP systems that do not use EDS files. If you instead wish to use explicit messaging, refer to the next chapter for information on using message instructions.

Rockwell Automation's RSLogix 5000 version 19 software is used for the example installation in this chapter.

### 5.1 Host System Configuration

RSLogix 5000 is used to configure both the ControlLogix and CompactLogix platforms. When using these platforms, you have the option of using a separate Ethernet Bridge module or an Ethernet port built into the processor.

If the Ethernet port is built into processor, the only step you have to take before adding the SD4840E2 is to create a new project with the correct processor or modify an existing project. Once this is done, the Ethernet port will automatically appear in the Project Tree. If you are using an Ethernet bridge module, you will have to add it to the I/O Configuration tree before adding the driver to your project.

Refer to your Rockwell Automation documentation if you need instructions for configuring the ethernet port.

### 5.2 Add the SD4840E2

You can add the SD4840E2 to the project once the Ethernet port (built-in or bridge module) is configured.

1) Right click on the Ethernet port and then click on "New Module..." in the pop-up menu.

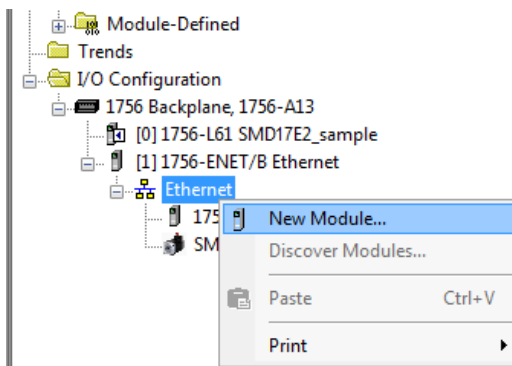


Figure T5.1 Adding an AMCI SD4840E2



**5.2 Add the SD4840E2 (continued)**

- 2) In the resulting Select Module Type screen, type “generic” into the filter as shown in figure T5.2. This will limit the results in the Catalog Number list.
- 3) Select the Catalog Number “ETHERNET-MODULE” in the list.
- 4) Click on the [Create] button to create the module.
- 5) Click on [Close] if necessary to close the Select Module Type screen.

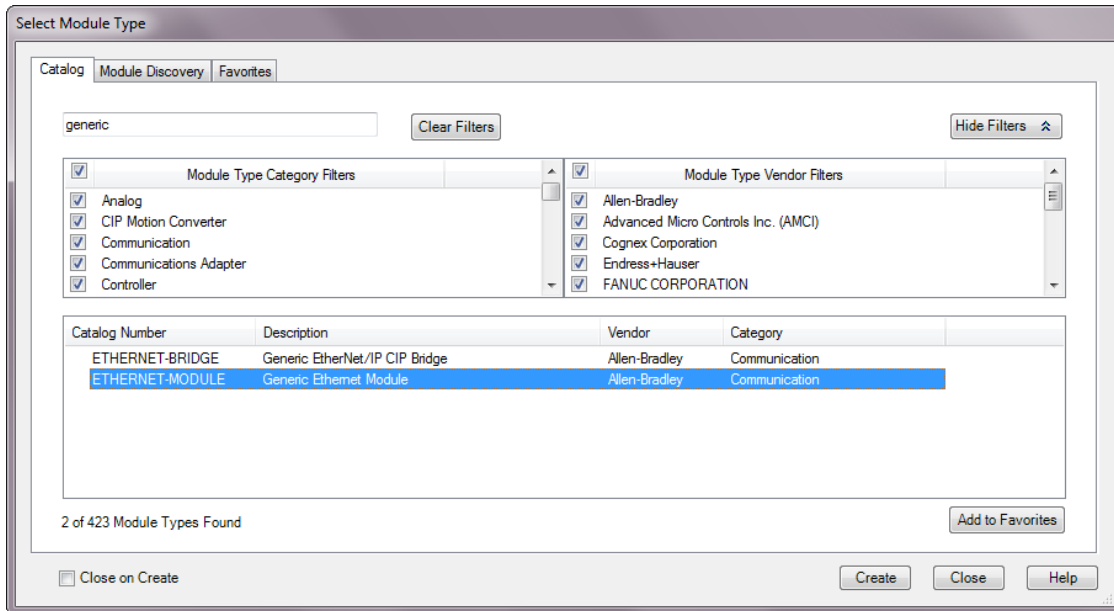


Figure T5.2 Selecting a Generic Device

## 5.2 Add the SD4840E2 (continued)

- 6) Set the following parameters in the Module Properties window. All parameters not listed here are optional. Figure T5.3 shows a completed screen.

Figure T5.3 Configuration Screen - Generic Device

- **Name:** A descriptive name for the SD4840E2
- **Comm Format:** Data - INT

### ! CAUTION

The Comm Format defaults to Data - DINT. The SD4840E2 will not be able to communicate with the host controller if this format is not changed when the device is added to the system. Once added, the Comm Format cannot be changed. The device must be deleted and again added to the project if the Comm Format is incorrect.

- **IP Address:** Must be the address you set for the SD4840E2. Refer to the [Set the IP Address and Protocol](#) task chapter starting on page 99 for information on setting the IP Address of the unit.
- **Input:** Assembly Instance = 100, Size = 10 words.
- **Output:** Assembly Instance = 150, Size = 10 words.
- **Configuration:** Assembly Instance = 110, Size = 0

- 7) Verify that the “Open Module Properties” check box is selected and click on [OK]. The Module Properties window will open. You can set the RPI time as required for your system in this window. The minimum RPI time for an SD4840E2 is 2 milliseconds. When done, click on [OK] to complete the setup.

### Error Code 16#0109

The PLC will generate an Error Code 16#0109 when the Comm Format parameter is not changed from its default of “Data-DINT” to “Data-INT”. This is the most common cause of communication failures with the SD4840E2.

**5.3 Configure the SD4840E2**

The device will join the EtherNet/IP network as soon as the request is made to it. If the SD4840E2 has a configuration stored in flash memory, it will be used to configure the unit on power up. You can also configure the unit at anytime and store this new configuration to flash. Configuration is accomplished by writing a block of data to the device that is formatted according to the specifications in the *Configuration Mode Data Format* reference, starting on page 55.

It is possible to store configuration data in the flash memory of the SD4840E2 and this configuration will be used on power up to configure the device. However, writing the configuration data to the driver on power up may simplify system maintenance if the device ever has to be swapped out.

**5.4 Buffer I/O Data**

Data to and from the SD4840E2 should be buffered once per scan using Synchronous Copy instructions. This is to insure stable input data during the program scan and guarantee that complete command data is delivered to the device. Ten word integer arrays can be used for this purpose.

These data tags should be buffered with Synchronous Copy File instructions to guarantee stable input data during the program scan and guarantee that complete command data is delivered to the device.

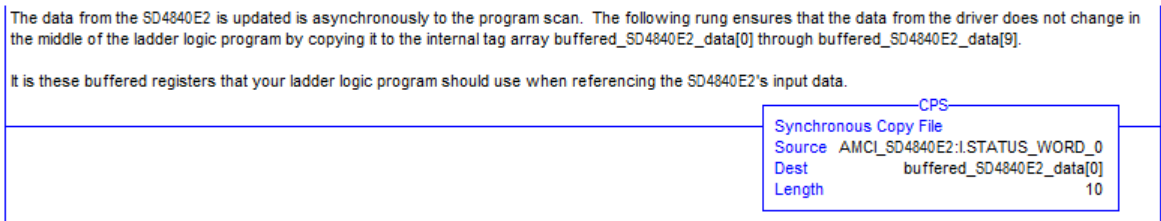


Figure T5.4 Buffer I/O Data

## TASK 6 (EtherNet/IP Option)

### ETHERNET/IP EXPLICIT MESSAGING

All controllers that support EtherNet/IP support explicit messaging. When using explicit messaging, Message Instructions must be added to your program to communicate with the SD4840E2. Explicit messaging can be used on platforms that also support implicit messaging.

Rockwell Automation controllers which are programmed with the RSLogix 500 software only support explicit messaging. A MicroLogix 1100 will be used as an example in this chapter.

#### 6.1 Required Message Instructions

Only two instructions are required to transfer data between the PLC and the SD4840E2. One instruction reads data from the unit and the other writes data to it. The sample programs available from AMCI use this style of programming. The two instructions are alternately triggered using the instruction's ENABLE bits. The remainder of the program controls when data in the source tags of the write instruction changes. The following table gives the required attributes for the instructions.

	Read Instruction	Write Instruction
Service Type	Read Assembly	Write Assembly
Service Code	E (hex)	10 (hex)
Class	4 (hex)	4 (hex)
Instance	100 (decimal)	150 (decimal)
Attribute	3 (hex)	3 (hex)
Length	20 bytes	20 bytes

Table T6.1 Message Instruction Attributes



Only RSLogix 500 version 8.0 or above can be used to configure Message Instructions to communicate with an EtherNet/IP device. Message Instructions do not work correctly in version 10 of RSLogix 500.

#### 6.2 Create Four New Data Files.

- An Integer file to contain the data read from the SD4840E2. This file must be at least 10 words in length.
- An Integer file to contain the data written to the SD4840E2. This file must be at least 10 words in length.
- A Message (MG) data file. This file must have at least two elements, one to control the Read Operation and one to control the Write Operation.
- An Extended Routing Information (RIX) data file. This file is used to store information used by the Message Instructions. This file must have at least two elements, one for the Read Operation and one for the Write Operation.

### 6.3 Add the Message Instructions to your Ladder Logic

The following rungs show how you can alternately read data from and write data to the unit.



Figure T6.1 Message Instruction Example

- 1) Double click on *Setup Screen* text inside the Message Instruction. The following window will open. Note that this is the default window and its appearance will change considerably as you progress through these steps.

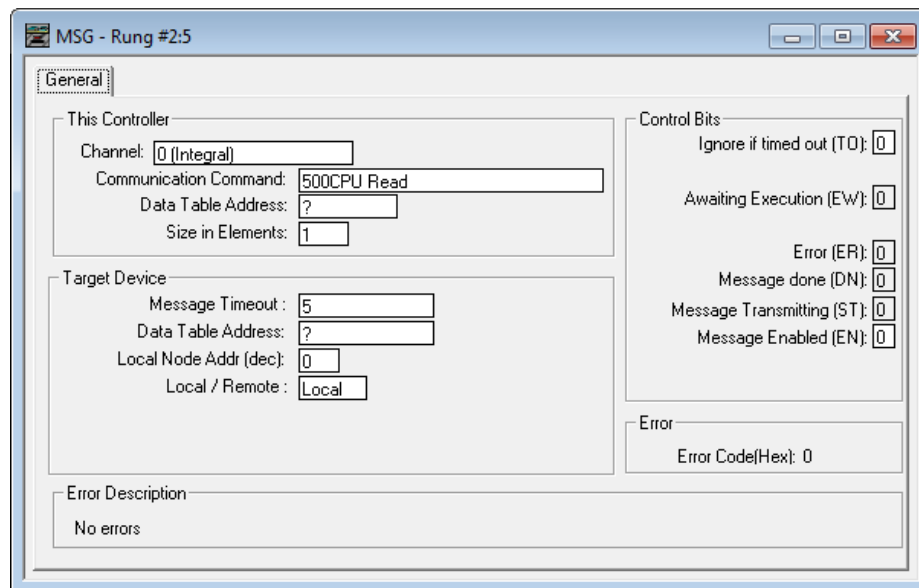


Figure T6.2 Message Instruction Setup Screen

- 2) Double click in the *Channel* field, click on the ▼, select “1 (Integral)”, and press Enter.
- 3) Double click in the *Communication Command* field, click on the ▼, select “CIP Generic” and press Enter.
- 4) If the Message Instruction is being used to read data from the SD4840E2, enter the integer file where the data will be placed in the *Data Table Address (Received)* field and press enter.
- 5) If the Message Instruction is being used to write data to the SD4840E2, enter the integer file where the source data will be located in the *Data Table Address (Send)* field and press Enter.
- 6) Enter “20” as the number of bytes needed in either the *Size In Bytes (Receive)* or *Size In Bytes (Send)* fields. The SD4840E2 requires 20 bytes for both Receive and Send.
- 7) Enter a RIX address in the *Extended Routing Info* field. Please note that each Message Instruction must have its own RIX address.

### 6.3 Add the Message Instructions to your Ladder Logic (continued)

- 8) Double click in the *Service* field and select “Read Assembly” for a Message Instruction that is being used to read data from the SD4840E2, or “Write Assemble” for a Message Instruction that is being used to send data to the SD4840E2, and press Enter.
- 9) For *Read* operations, the *Service Code* field will change to “E” (hex). For *Write* operations, the *Service Code* field will change to “10” (hex). For both read and write operations, the *Class* field will change to “4” (hex), and the *Attribute* field will change to “3” (hex).
- 10) For Read operations, enter a value of 100 decimal (64 hex) in the *Instance* field.  
For Write operations, enter a value of 150 decimal (96 hex) in the *Instance* field.

The figure below show a typical configuration for Message Instructions being used to read data from the SD4840E2. Please note that the Data Table Address (Receive) field may be different in your application.

The screenshot shows the 'MSG - Rung #2:5' configuration window. The 'General' tab is selected. The configuration is as follows:

- This Controller:** Channel: 1 (Integral); Communication Command: CIP Generic; Data Table Address (Receive): N9:0; Size in Bytes (Receive): 20.
- Target Device:** Message Timeout: 5; Local / Remote: Local; MultiHop: Yes; Extended Routing Info File (RIX): RIX10:0.
- Service:** Read Assembly; Service Code (hex): E; Class (hex): 4; Instance (hex): 64; Attribute (hex): 3.
- Control Bits:** Ignore if timed out (TO): 0; Break Connection (BK): 0; Awaiting Execution (EW): 0; Error (ER): 0; Message done (DN): 0; Message Transmitting (ST): 0; Message Enabled (EN): 0.
- Error:** Error Code (Hex): 0.
- Error Description:** No errors.

Figure T6.3 Read Message Instruction Setup Screen

**6.3 Add the Message Instructions to your Ladder Logic (continued)**

The figure below show a typical configuration for Message Instructions being used to write data to the SD4840E2. Please note that the Data Table Address (Send) field may be different in your application.

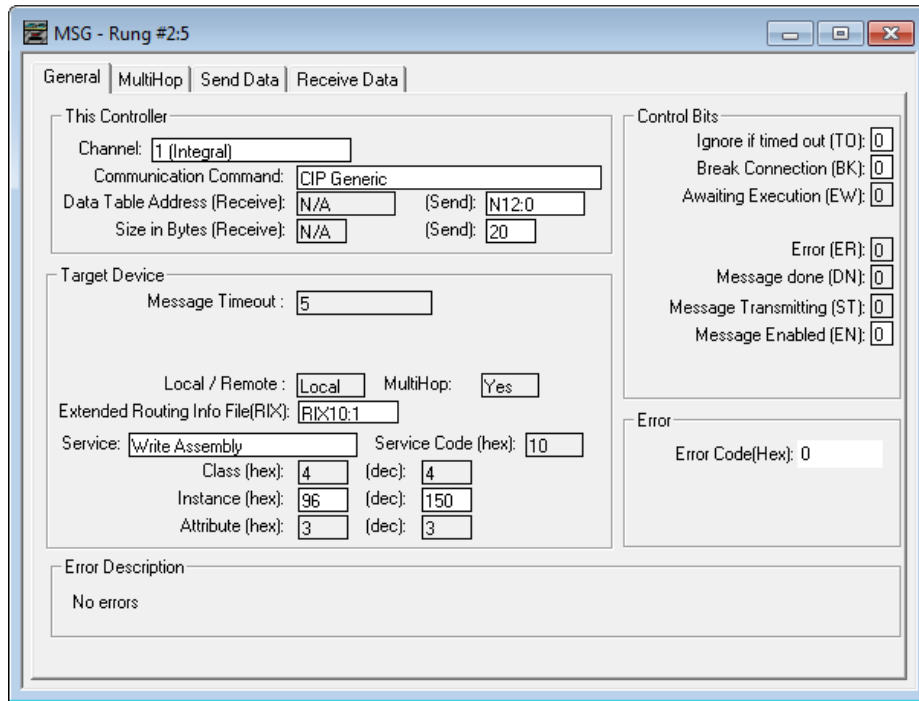


Figure T6.4 Write Message Instruction Setup Screen

Click on the MultiHop tab on the top of the window. As shown in figure T6.5, enter the IP address of the SD4840E2 and press Enter.

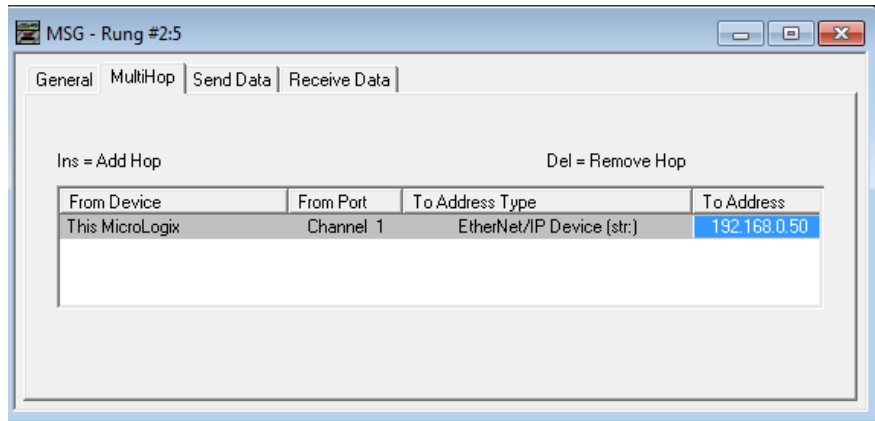


Figure T6.5 Message Instruction MultiHop Settings

After you are finished adding both the read and write message instructions to your program, save and download the program to the PLC.



## 6.4 Troubleshooting

If you are unable to communicate with the SD4840E2, the problem may be that the Ethernet port of your MicroLogix 1100 has not been configured. To check this:

- 1) Double click on Channel Configuration in the Project Tree and then select the Channel 1 tab. The following window will open.

The screenshot shows the 'Channel Configuration' dialog box with the 'Channel 1' tab selected. The 'Driver' is set to 'Ethernet'. The 'Hardware Address' is 'C8:8E:D1:B1:4A:56' and 'Network Link ID' is '0'. The IP configuration is: IP Address: 192.168.0.1, Subnet Mask: 255.255.255.0, Gateway Address: 0.0.0.0, Default Domain Name: (empty), Primary Name Server: 0.0.0.0, and Secondary Name Server: 0.0.0.0. Under 'Protocol Control', 'HTTP Server Enable' is checked, while 'BOOTP Enable', 'DHCP Enable', 'SNMP Server Enable', and 'SMTP Client Enable' are unchecked. 'Msg Connection Timeout (x 1mS)' is 15000, 'Msg Reply Timeout (x 1mS)' is 3000, and 'Inactivity Timeout (x Min)' is 30. 'Auto Negotiate' is checked, and 'Port Setting' is '10/100 Mbps Full Duplex/Half Duplex'. There are also empty fields for 'Contact' and 'Location'. Buttons for 'OK', 'Cancel', 'Apply', and 'Help' are at the bottom.

Figure R6.7 MicroLogix Ethernet Configuration Screen

- 2) Enter the IP address and Subnet Mask of your MicroLogix 1100, (not the address of the SD4840E2) and click on [Apply]. The Ethernet Port should now be working.

**NOTE** AMCI is aware of an issue with the RIX data type in version 10 of RSLogix 500. If you are experiencing communications errors and are running version 10, please contact Rockwell Automation for support.

***Notes***

# TASK 7 (Modbus TCP Option)

## MODBUS TCP CONFIGURATION

An AMCI SD4840E2 that has been configured for the Modbus TCP protocol requires a host controller to issue configuration data and motion commands to the unit. This chapter tell you how the I/O words used by an AMCI SD4840E2 are mapped to the Modbus I/O registers.

### 7.1 Enable Modbus TCP Protocol

The AMCI Net Configurator utility can be used to change the communications protocol used by the SD4840E2. This is typically done while setting the IP address. Specifically, follow the steps in section 3.2c, *Use the AMCI NET Configurator Utility* which starts on page 102.

### 7.2 Modbus Addressing

The register addresses used in this manual are the *Modbus logical reference numbers*<sup>†</sup>, which are unsigned integers starting at zero. This is often called *zero based* addressing. In this scheme, the first register is given an address of zero. This is the actual addressing scheme used in the Modbus packets.

Another common addressing scheme is *one based* or *data model* addressing. In this scheme, the register's number is used as its address, so the first register, Register 1 in the data model, has an address of 1.

#### 7.2.1 Modbus Table Mapping

The Discrete Input and Input Register tables in the Modbus data model map to the same physical memory locations in the SD4840E2 units.

- These registers hold data that is reported from the driver to the host controller. This data is typically command responses and status data.
- Addresses for these registers and inputs start at 0 in zero based addressing.

As examples:

- Discrete Input 0 is the same memory location as bit 0 of the first Input Register.
- Register address 3, the fourth register, contains Discrete Inputs 48 through 63.

The Coil and Holding Register tables in the Modbus data model map to the same physical memory locations in the SD4840E2 units.

- These registers hold data that is from the host controller to the unit. This data is typically commands.
- Addresses for these registers start at 1024 in zero based addressing. Coil addresses start at 16,384 in zero based addressing (1024\*16).

As examples:

- Coil 16384 is the same memory location as bit 0 of the first Holding Register.
- Register address 1025, the address of the second Holding Register, contains Discrete Inputs 16,400 through 16,415 in zero based addressing.

#### 7.2.2 Host Addressing

Your host controller may not use these basic addressing schemes for communicating over a Modbus connection. For example, Modicon controllers use addresses starting at 30000 for Input Registers and addresses starting at 40000 for Holding Registers. GE hosts internally use their %R memory for Holding Registers and %AI memory for Input Registers.

If this is the case, you will define a mapping between your host controller's addressing scheme and the zero based Modbus TCP addresses when you add the SD4840E2 to your host controller. Refer to your host controller's documentation for information on how to accomplish this.

<sup>†</sup> MODBUS Application Protocol Specification V1.1b3, section 4.3: MODBUS Data model. [www.modbus.org](http://www.modbus.org)

**7.3 AMCI Modbus TCP Memory Layout**

The SD4840E2 has a starting Input Register address of 0 and a starting Output Register address of 1024. Input Registers hold the data from the driver while Output Registers hold the data to be written to the unit. Figure T7.1 shows how an SD4840E2 is mapped to the Modbus data reference. The complete specification for the Modbus protocol can be downloaded at <http://www.modbus.org/specs.php>.

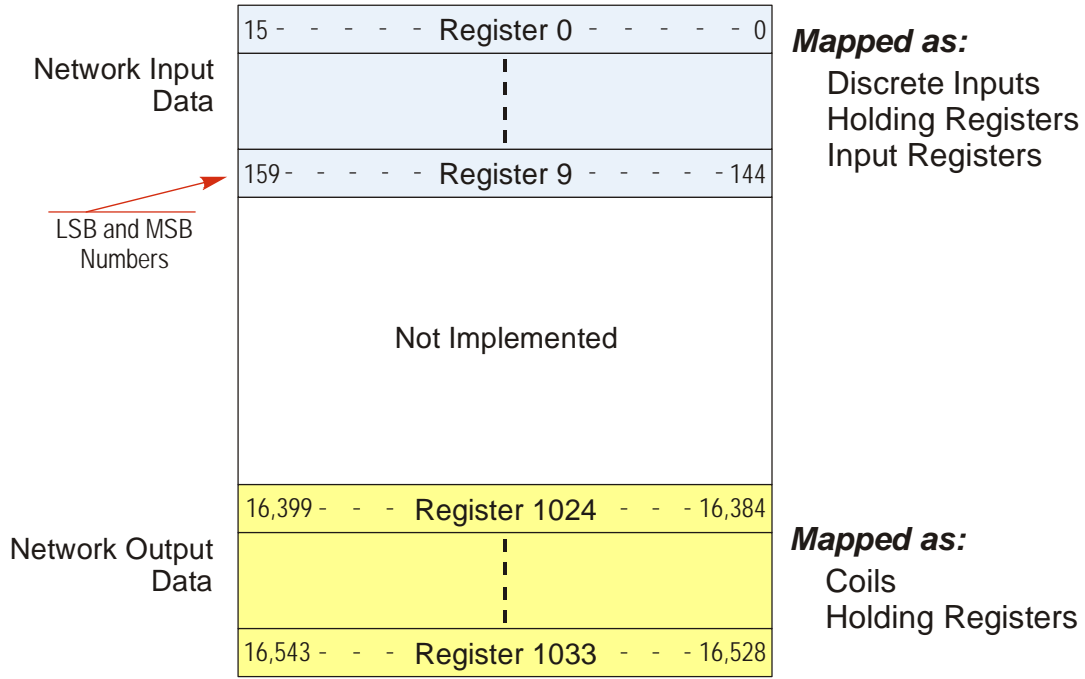


Figure T7.1 Modbus Data Reference Map

**7.4 Supported Number of Connections**

All SD4840E2 units support seven concurrent connections. When connections exist, the Network Status (NS) LED on the back of the unit will flash green. The number of blinks indicate the number of active connections. There is a two second break between groups of flashes.

## 7.5 Supported Modbus Functions

Function Code	Function Name	SD4840E2 Register	Addressing method
1	Read Coils	OUTPUT	Bit: Addresses starting at 16,384
2	Read Discrete Inputs	INPUT	Bit: Addresses starting at 0
3	Read Holding Registers	OUTPUT & INPUT	Word: Out Regs. Starting at 1024 In Regs. Starting at 0
4	Read Input Registers	INPUT	Word: Addresses starting at 0.
5	Write Single Coil	OUTPUT	Bit: Addresses starting at 16,384
6	Write Single Register	OUTPUT	Word: Addresses starting at 1024
15	Write Multiple Coils	OUTPUT	Bit: Addresses starting at 16,384
16	Write Multiple Registers	OUTPUT	Word: Addresses starting at 1024
22	Mask Write Register	OUTPUT	Word: Addresses starting at 1024
23	Read/Write Registers	INPUT/OUTPUT	Word: Out Regs. Starting at 1024 In Regs. Starting at 0

Table T7.1 Supported Modbus Functions

Table T7.1 above lists all of the Modbus functions supported by an SD4840E2. AMCI supports all of these functions so that you can control the unit as you see fit. However, if you are looking for the easiest way to interface with your unit, then you only need to use the *Read/Write Registers* function, which is function code 23.

**NOTE** 

Each SD4840E2 buffers the data that is sent to it over the network. If you use the *Read/Write Registers* function to write data to the unit, the data read with that command will not contain the response to the new write data. The response to the new data will be sent with the next data read.

## 7.6 Supported Modbus Exceptions

Code	Name	Description
01	Illegal function	The SD4840E2 does not support the function code in the query.
02	Illegal data address	The data address received in the query is outside the initialized memory area.
03	Illegal data value	The data in the request is illegal.

Table T7.2 Supported Modbus Exceptions

***Notes***

## TASK 8 (PROFINET Option)

# PROFINET NETWORK CONFIGURATION

This chapter outlines the steps commonly needed to get an SD4840E2 communicating with the PROFINET master. A Siemens SIMATIC S7-1212C controller is used as an example.

### Basic Steps

Configuring a PROFINET host requires a few basic steps.

- 1) Download the ZIP archive that contains the GSDML files for the SD4840E2 from the [www.amci.com](http://www.amci.com) website.
- 2) Install the GSDML file into the configuration software for your host controller.
- 3) Add the SD4840E2 to the PROFINET Network.
- 4) Set the I/O word addresses used to communicate with the unit.

### 8.1 Download the GSDML files

The GSDML files are available on the AMCI website on the <http://www.amci.com/industrial-automation-support/configuration-files/> web page. The file is a ZIP archive that has to be extracted to a folder on your computer. Extracting the ZIP file will leave you with multiple files. One is the GSDML file and the others are icon files for the various devices.

### 8.2 GSDML File Installation

- 1) Open or create a new project that will include the SD4840E2 and open the Project View of the project.
- 2) In the menu, select *Options -> Manage general station description files (GSD)*.
- 3) In the window that opens, click on the [...] button and navigate to the folder that contains the extracted GSDML file you downloaded from the AMCI website. Once at the folder, click on the [OK] button.
- 4) Click on the check box next to the name of the GSD file and click on the [Install] button. The system will install the GSD file.
- 5) Click the [Close] button and wait for the software to finish installing the file and updating the Hardware Catalog.

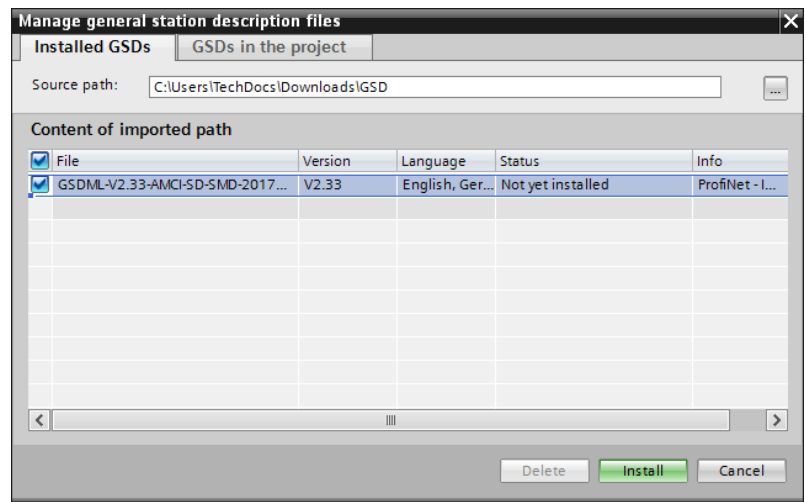


Figure T8.1 GSD File Installation

### 8.3 Configure the PROFINET Network

A CPU must be added to the project and the PROFINET network must be configured before a Networked Stepper Driver can be added to the system.

Refer to Siemens documentation for information on configuring the PROFINET network to suit your application.



### 8.4 Add the SD4840E2 to the PROFINET Network

- 1) With the project open in Project View, double click on “Device & Networks” in the project tree.
- 2) If need be, click on the “Hardware Catalog” vertical tab to open the Hardware Catalog.
- 3) You can search for “SD48”, or browse to the SD4840x2 icon by clicking through *Other field devices* +> *PROFINET IO* +> *IO* +> *Advanced Micro Controls Inc.* +> *AMCI\_Products* +> *AMCI\_Drives*. Drag and drop the appropriate icon onto the PROFINET network.
- 4) Drag the green square on the SD4840x2 icon onto the PROFINET network line to connect the device to the network.

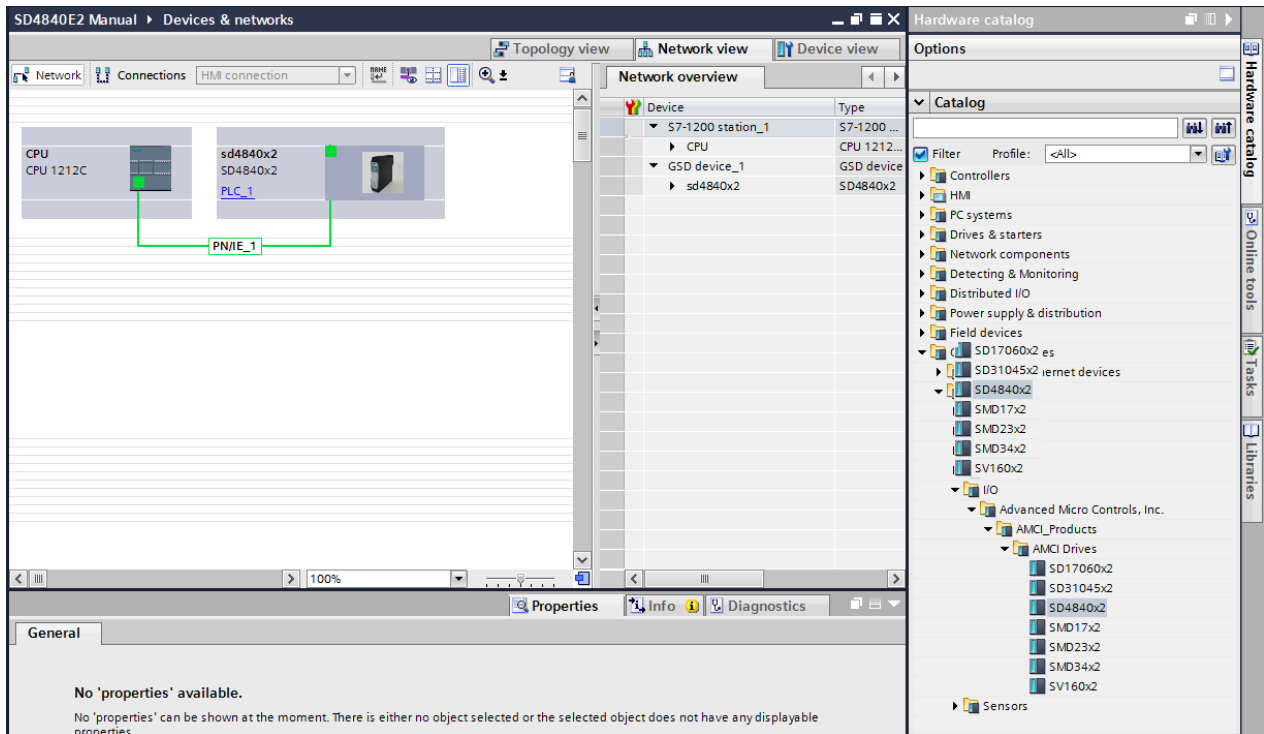


Figure T8.2 Networked Driver Added to PROFINET Network

### 8.4 Add the SD4840E2 to the PROFINET Network (continued)

- 5) Right click on the SD4840x2 icon and select “Properties” from the pop up menu. The Inspector window will open at the bottom of the screen. Under the “General” tab, select the “►General” heading. You can rename the SD4840E2 by changing the Name: field.
- 6) Under the “►PROFINET interface [x1]” heading, select “Ethernet addresses”. Under the IP protocol section, set the desired IP address and subnet mask for the SD4840E2.

The screenshot displays the configuration software interface. The top window shows a network topology with a CPU (CPU 1212C) and an SD4840x2\_Axis1 device connected via a PN/IE\_1 interface. The bottom window shows the Properties dialog for the SD4840x2\_Axis1 device, with the 'Ethernet addresses' tab selected. Red arrows point to 'Step 5' (General tab) and 'Step 6' (Ethernet addresses sub-tab).

Device	Type	Addr...
S7-1200 station_1	S7-1200 station	
▶ CPU	CPU 1212C AC/DC/Rly	
GSD device_1	GSD device	
▶ SD4840x2_Axis1	SD4840x2	

**Properties - SD4840x2\_Axis1 [SD4840x2]**

**General** ← Step 5

  Catalog information

  PROFINET interface [X1]

    General

    Ethernet addresses ← Step 6

    Advanced options

      Interface options

      Media redundancy

      ▶ Real time settings

      ▶ Port 1 [X1 P1 R]

      ▶ Port 2 [X1 P2 R]

  Identification & Maintenance

  Module parameters

**Ethernet addresses**

  Interface networked with

    Subnet: PN/IE\_1

    Add new subnet

**IP protocol**

  IP address: 192 . 168 . 0 . 155

  Subnet mask: 255 . 255 . 255 . 0

Synchronize router settings with IO controller

Use router

  Router address: 0 . 0 . 0 . 0

**PROFINET**

Generate PROFINET device name automatically

  PROFINET device name: sd4840e2\_axis1

  Converted name: sd4840e2xbaxis155bb

  Device number: 1

Figure T8.3 Networked Driver IP Addressing

## 8.5 Configure the SD4840E2

- Continuing with the Inspector Window, click on the “Module parameters” heading. The display will change to show all of the parameters of the SD4840E2. These configuration parameters are written down to the SD4840E2 whenever it connects to the network.

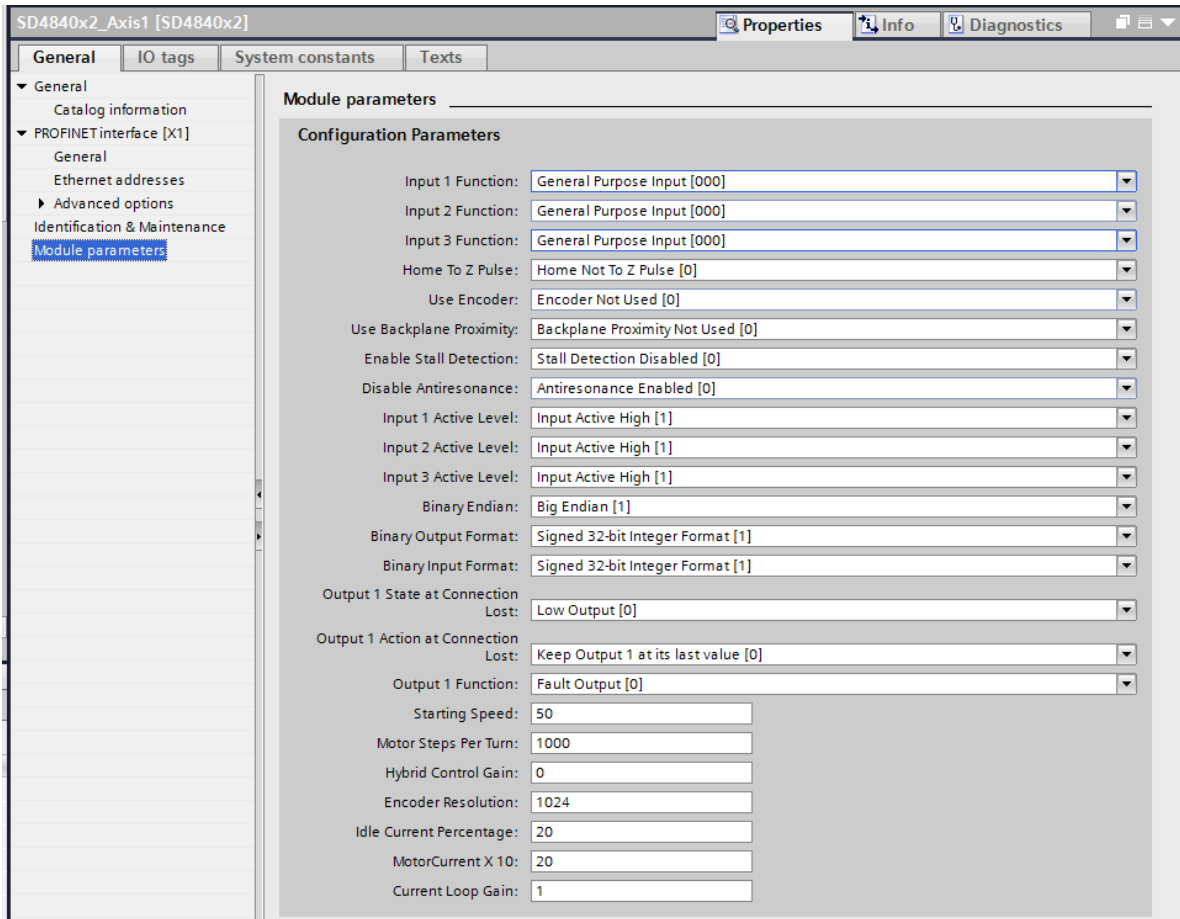


Figure T8.4 Configuration Parameters

- Configuration is done with a combination of drop down menus and value fields. Drop down menus limit you to a set of valid selections. When entering numbers into value fields, a pop up will appear that shows you the valid range of values.

**NOTE** Even though your selections are limited, it is still possible to mis-configure the SD4840E2 by using this screen. For example, it is possible to configure two or more inputs as CW Limit Switches. If an error occurs, the SD4840E2 will show that a configuration error exists after it connects to the network.

**NOTE** This data is only written to the unit when it connects to the network, which is typically on power up. The unit can still be reconfigured at any time by writing the appropriate data to the SD4840E2's output registers.

**NOTE** These configuration parameters cannot be changed while on line. To make any changes, you must go off line, make the changes, save the project, download it, and then go back on line.

## 8.6 Set the I/O Configuration

The SD4840E2 units require 10 Input Words (20 Input Bytes) and 10 Output Words (20 Output Bytes). All required Input and Output Bytes are defined by the GSDML file and divided into suitable modules. These settings are shown in the Table T8.1.

Input / Output Bytes of an SD4840E2	Input / Output Modules of an SD4840E2
20 Input Bytes	Input Module - Slot 1: 20 bytes
20 Output Bytes	Output Module - Slot 2: 20 bytes

Table T8.1 PROFINET I/O Configuration

- 1) With the SD4840x2 icon selected on the PROFINET bus, click on the “Device view” tab. The view in the Hardware Catalog will change. Expand the Module tree to show both the Input and Output modules.
- 2) To map the I/O bytes to the CPU, double click on the “20 bytes IN” and “20 bytes OUT” icons in the Hardware Catalog. The system will automatically assign the next I and Q addresses to the data table.

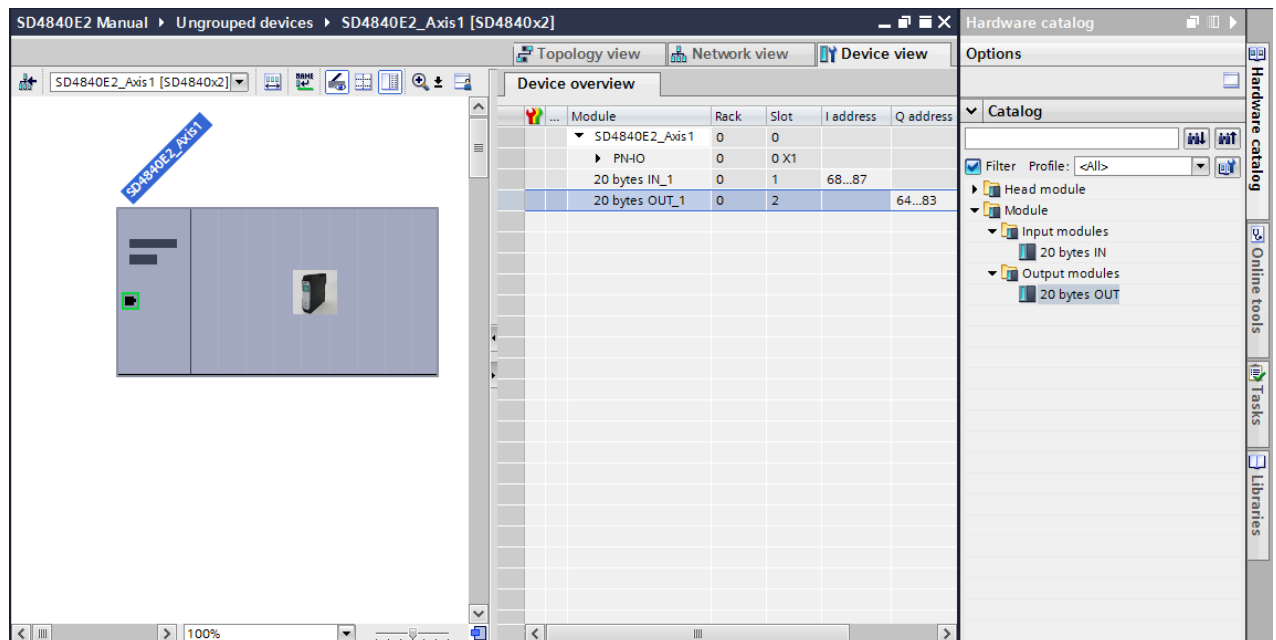


Figure T8.5 I/O Byte Mapping

## 8.7 Verify and Download the New Configuration

- 1) Continue by adding any remaining devices to your PROFINET network.
- 2) Compile and download the project to the CPU.

**MRP Installations**

At this point, the SD4840E2 is configured and ready to use. If you are using the unit in a redundant, ring based, network that uses the Media Redundancy Protocol (MRP), continue with the following instructions.

**Media Redundancy Protocol (MRP) installations require that the SD4840E2 be installed in a ring topology. In these applications, both Ethernet ports are used when wiring the ring, daisy chaining from one unit in the ring to the next. The steps below covers typical software configuration that must also be completed.**

**8.8 Configure the SD4840E2 as an MRC**

The SD4840E2 functions as a Media Redundancy Client (MRC) in an MRP network.

- 1) Switch to Topology view and drag the additional connections between the appropriate ports.
- 2) Click on the SD4840x2 icon to select it. In the Inspector window, select *Advanced options* +> *Media redundancy*. Use the “MRP domain:” drop down menu to select the appropriate domain. Use the “Media redundancy role:” drop down menu to select “Client”.
- 3) Continuing in the Inspector window, select *Advanced options* +> *Port 1* +> *Port interconnection*. Under “Partner port:”, the partner port you assigned to the port when you drew the topology is shown. If you do not know which port will be the partner port in the actual installation, you can use the drop down menu to select “Any partner”.
- 4) If need be, repeat step 3 for Port 2 of the SD4840E2.

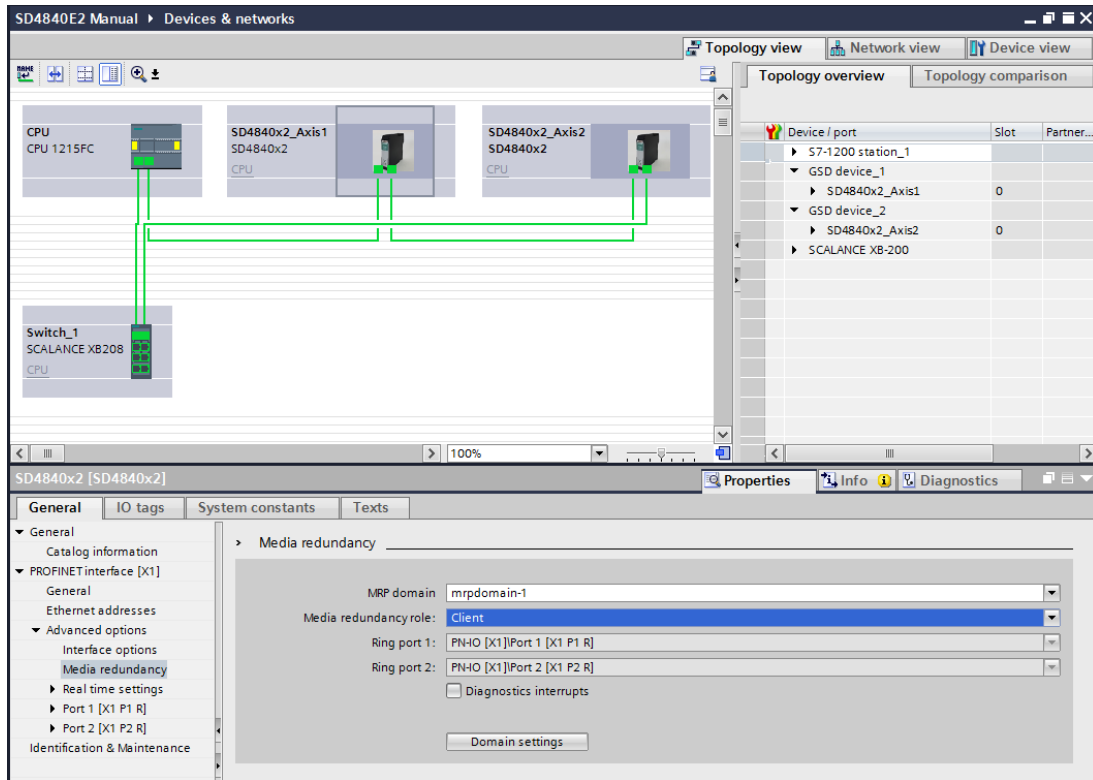


Figure T8.6 MRP Topology and Client settings

- 5) Continue configuring the rest of the devices on the network before compiling the project and downloading it to the CPU.

# OPTIONAL TASK A

## CONFIGURE YOUR NETWORK INTERFACES

### A.1 Firewall Settings

Firewalls are hardware devices or software that prevent unwanted network connections from occurring. Firewall software is present in Windows XP and above and it may prevent your computer from communicating with the SD4840E2. Configuring your firewall to allow communication with the SD4840E2 is beyond the scope of this manual.

AMCI strongly suggests temporarily disabling any firewall software while using the Net Configurator utility. You should enable the firewall once you have finished using the utility.

### A.2 Disable All Unused Network Interfaces

Routing and default gateway setting on your computer can interfere with the proper operation of the Net Configurator software. The Net Configurator software uses broadcast packets to locate devices on the network, and sometimes these packets are sent out through the default gateway instead of the interface attached to the AMCI product. The easiest way to avoid this problem is to temporarily disable all network interfaces that are not attached to the SD4840E2.

**NOTE** This includes all wireless interfaces as well as all Bluetooth interfaces.

### A.3 Configure Your Network Interface

Before you can communicate with an SD4840E2, your network interface must be on the same subnet as the driver.

**NOTE** The rest of this procedure assumes you are using the 192.168.0.xxx subnet. If you are not, you will have to adjust the given network addresses accordingly.

The easiest way to check the current settings for your NIC is with the 'ipconfig' command.

- ▶ For Windows 7, click on the [Start] button, and type "cmd" in the "Search programs and files" text box. Press [Enter] on the keyboard.
- ▶ For Windows 8 and 10, press the [Win+X] keys and select "Command Prompt" from the resulting popup. There is no need to run the command prompt as the administrator, so do not select "Command Prompt (Admin)".

A DOS like terminal will open. Type in 'ipconfig', press [Enter] on the keyboard and the computer will return the present Address, Subnet Mask, and Default Gateway for all of your network interfaces. If your present address is 192.168.0.xxx, where 'xxx' does not equal 50, and your subnet mask is 255.255.255.0, then you are ready to configure your SD4840E2. Figure A.1 shows the output of an ipconfig command that shows the "Local Area Connection 2" interface on the 192.168.0.xxx subnet.

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .                : 176.16.25.17
    Subnet Mask . . . . .              : 255.255.0.0
    Default Gateway . . . . .          : 176.16.255.1
    DHCP Class ID . . . . .            : 

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .                : 192.168.0.224
    Subnet Mask . . . . .              : 255.255.255.0
    Default Gateway . . . . .          : 

C:\>
```

Figure A.1 ipconfig Command

**A.3 Configure Your Network Interface (continued)**

If your present address is not in the 192.168.0.xxx range, type in 'ncpa.cpl' at the command prompt and hit [Enter] on the keyboard.

- For Windows 7, this open the *Network Connections* window. Double click on the appropriate interface. In the window that opens, select "Internet Protocol Version 4 (TCP/IP v4)" from the list and then click on the [Properties] button.
- For Windows 8 and 10, this open the Network Connections window. Double click on the appropriate interface. In the window that opens, select "Internet Protocol Version 4 (TCP/IP v4)" from the list and then click on the [Properties] button.

Set the address and subnet mask to appropriate values. (192.168.0.1 and 255.255.255.0 will work for an SD4840E2 that has factory default settings.) The default gateway and DNS server settings can be ignored.

**A.4 Test Your Network Interface**

Going back to the terminal you opened in the last step, type in 'ping aaa.bbb.ccc.ddd' where 'aaa.bbb.ccc.ddd' in the IP address of the SD4840E2. The computer will ping the unit and the message "Reply from aaa.bbb.ccc.ddd: bytes=32 time<10ms TTL=128" should appear four times.

If the message "Request timed out." or "Destination host unreachable" appears, then one of four things has occurred:

- You set a new IP address, but have not yet cycled power to the SD4840E2
- You did not enter the correct address in the ping command.
- The IP address of the SD4840E2 is not set correctly.
- The SD4840E2 and the computer are not on the same subnet.

***Notes***





***ADVANCED MICRO CONTROLS INC.***

20 GEAR DRIVE, TERRYVILLE, CT 06786 T: (860) 585-1254 F: (860) 584-1973

[www.amci.com](http://www.amci.com)

***LEADERS IN ADVANCED CONTROL PRODUCTS***