ADVANCED MICRO CONTROLS INC.

Manual #: 940-0S172

Networked Stepper Indexer/Driver

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

E2 Technology

User Manual

AMCI Motion Control Products

EtherNet/IP™

PROFINET®

Modbus

UL® LISTED
IND. CONT. EQ.
E231137

GENERAL INFORMATION

Important User Information

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. 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

ABOUT THIS MANUAL

Read this chapter to learn how to navigate through this manual and familiarize yourself with the conventions used in it. The last section of this chapter highlights the manual's remaining chapters and their target audience.

Audience

This manual explains the installation and operation of AMCI's Networked Stepper Indexer/Drivers. It is written for the engineer responsible for incorporating these products into a design as well as the engineer or technician responsible for their actual installation.

Applicable Units

This manual applies to the SD17060E2 and SD31045E2 units.

- "E2" units released prior to January, 2019, support the EtherNet/IP and Modbus TCP protocols. They are factory configured for the EtherNet/IP protocol.
- "E2" units that were released on or after January 1st, 2019, support the EtherNet/IP, Modbus TCP, and PROFINET protocols, These units are factory configured for the EtherNet/IP protocol.

The serial numbers for all SD units are date coded and follows the format D'MMYYnnnn'. 'MMYY' is the month and year of manufacturer and 'nnnn' is a sequential number that resets every month.

Each unit contains a two port Ethernet switch, which simplifies network wiring. When the EtherNet/IP protocol is used, the unit can act as a node in a Device Level Ring (DLR). When PROFINET is enabled, the unit supports the Media Redundancy Protocol (MRP) and can be incorporated in PROFINET installations that use a redundant ring topology.

With the exception of power requirements and output current, the information in this manual applies equally to the SD17060E2 and SD31045E2 units. The term "Networked Driver" is used when the information applies to all drivers.

Navigating this Manual

This manual is designed to be used in both printed and on-line forms. Its on-line form is a PDF document, which requires Adobe Acrobat Reader version 7.0+ to open it. 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. If you decide to print out this manual, all sections contain an even number of pages which allows you to easily print out a single chapter on a duplex (two-sided) printer.

Manual Conventions

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

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

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

WARNING — **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.
Emphasis Font	Font used for parameter names and the first time a new term is introduced.
Cross Reference	When viewing the PDF version of the manual, clicking on a blue cross reference jumps you to referenced section of the manual.
HTML Reference	When viewing the PDF version of the manual, clicking on a red cross reference opens your default web browser to the referenced section of the AMCI website if you have Internet access.

Trademarks and Other Legal Stuff

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, 940-0S172 is the second release of this manual. It was first released on September 28th, 2018. It changes manual layout and adds information on the PROFINET protocol.

Revision History

940-0S171: May 31, 2016 - Changes to IEC 60417 grounding symbols.

940-0S170: May 23, 2016 - Initial Release

Manual Layout

You will most likely read this manual for one of two reasons:

- If you are curious about the Networked Stepper Indexer/Driver products from AMCI, this manual contains the information you need to determine if these products are the right products for your application. The first chapter, ***SD17060E2 & SD31045E2 Specifications*** contains all of the information you will need to fully specify the right product for your application.
- If you need to install and use a Networked Stepper Indexer/Driver product from AMCI, then the rest of the manual is written for you. To simplify installation and configuration, the rest of the manual is broken down into *tasks* and *references*. Using a Networked Stepper Indexer/Driver requires you to complete multiple tasks, and the manual is broken down into sections that explain how to complete each one.

Section Title	Page #	Section Description
SD17060E2 & SD31045E2 Specifications	11	Complete specifications for the SD17060E2 and SD31045E2 products.
UL/cUL Recognized Installations	23	Installation guidelines that must be followed in order for an installation of a Networked Driver to meet UL/cUL criterion.
Motion Control	25	Reference information on how the driver can be used to control motion in your application.
Calculating Move Profiles	47	Reference information on calculating detailed move profiles.
Homing an AMCI Networked Driver	57	Reference information on how the home position of the SD17060E2 or SD31045E2 can be set.
Configuration Mode Data Format	61	Reference information on the format of the network data to and from the SD17060E2 or SD31045E2 that is used to configure it.
Command Mode Data Format	69	Reference information on the format of the network data to and from the SD17060E2 or SD31045E2 that is used to command it.
General Installation Guidelines	91	Reference information on grounding, wiring, and surge suppression, that is applicable to any control system installation.
Installing an AMCI Networked Driver	97	Task instructions covering how to install an SD17060E2 or SD31045E2 on a machine. Includes information on mounting, grounding, and wiring specific to the units.
Set the IP Address	109	Task instructions that covers the options for setting the IP address on an SD17060E2 or SD31045E2.
Using the EDS File with EtherNet/IP	117	Task instructions that cover how to add an SD17060E2 or SD31045E2 to a EtherNet/IP host that supports the use of EDS files.
Implicit Communications Without EDS	123	Task instructions for adding a Networked Driver to a project as a generic device. This configuration is for EtherNet/IP hosts that do not support EDS files while supporting implicit communications.
EtherNet/IP Explicit Messaging	127	Task instructions for adding message instructions to you host controller program that write data explicitly to the SD17060E2 or SD31045E2.
Modbus TCP Configuration	133	Task instructions for communicating with an SD17060E2 or SD31045E2 using the Modbus TCP protocol.
PROFINET Network Configuration	137	Task instructions that covers how to add a Networked Drive to a PROFINET network.
Optional: Configure Your Network Interfaces	23	Instructions for the optional task of configuring network interfaces on your computer or laptop.

Notes

REFERENCE 1

SD17060E2 & SD31045E2 Specifications

This manual is designed to get you up and running quickly with either the SD17060E2 or the SD31045E2 from AMCI. 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.

Networked Stepper Indexer/Drivers

The AMCI SD17060E2 is a 6.0Arms micro-stepping driver with a 170Vdc internal bus voltage. The AMCI SD31045E2 is a 4.4Arms micro-stepping driver with a maximum internal bus voltage of 310Vdc. What makes each drive unique is its built-in indexer that accepts configuration and command data from a host system over the internal Ethernet port. The SD17060E2 and SD31045E2 drivers can be configured to use either Ethernet/IP, PROFINET, or Modbus/TCP protocols, which makes these units easy to integrate into a wide variety of controller systems.

The SD1706E2 and SD31045E2 are members of the growing line of products from AMCI that incorporate our *E2 Technology*. E2 Technology by AMCI is an innovative new multi-protocol approach to Ethernet distributed I/O.

E2 Technology products are simple and intuitive, allowing easy transition between multiple common industrial Ethernet protocols without the need to physically switch parts. An advanced web server integrated into all AMCI E2 Technology devices facilitates simple device configuration and troubleshooting via web-browser. Furthermore, an impressive array of advanced features for each supported protocol has been incorporated into the devices to meet many unique application requirements.

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 separate 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

Networked Stepper Indexer/Drivers (continued)

General Functionality (continued)

Both Networked Drivers can be powered from a nominal 115 Vac, 50/60 Hz source and they will have a 170 Vdc motor bus voltage with this input supply. The SD31045E2 Drivers can also be powered by a nominal 230 Vac 50/60 Hz source and they will have a 310 Vdc motor bus voltage with this input supply.

The output motor current is fully programmable on these Networked Drivers. The SD17060E2 Drivers can be programmed from 1.0 Arms to 6.0 Arms, while the SD31045E2 Drivers can be programmed from 1.0 Arms to 4.4 Arms. It is even possible to change the motor current setting while the move is in progress. These ranges of output currents makes the Networked Drivers compatible with the complete line of stepper motors 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 the programming to be very similar to these products.

All Networked Drivers from AMCI are true RMS motor current control drivers. This means that you will always receive the motor's rated torque regardless of the *Motor_Resolution* setting. (Drivers that control the peak current to the motor experience a 30% decrease in motor torque when microstepping a motor.) The Networked Drivers automatically switch from RMS to peak current control when the motor is idle to prevent overheating the motor.

In addition to power and motor hookups, the Networked Drivers have 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. Each input 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
- A/B Encoder Inputs
- General Purpose Input

Network Data Format

In order to support any host that communicates with the EtherNet/IP, PROFINET, or Modbus TCP protocols, the format of the data read from and written to the Networked Driver while in command mode is completely programmable. The format of the network input and output data can be programmed separately.

The smallest data size used by the Networked Driver 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.

Networked Stepper Indexer/Drivers (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 *Data 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) 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*. Rockwell Automation controllers use *little endian* format, while the default Modbus format is *big endian*.

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 Data Endian parameter should be set to *little endian*.

Specifications

Driver Type

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

SD17060E2 Drivers: 170Vdc output bus.
SD31045E2 Drivers: up to 310Vdc output bus.

Physical Dimensions

Width:	SD17060E2 Drivers: 2.1 inches max.
	SD31045E2 Drivers: 2.7 inches max.
Depth:	4.0 inches max.
Height:	6.2 inches
	7.0 inches with mounting tabs

Weight

SD17060E2 Drivers: 2.4 lbs. (1.1 kg.) max.
SD31045E2 Drivers: 2.7 lbs. (1.25 kg.) max.

Inputs

Electrical Characteristics for all Inputs: . . . . Differential. UL 1577 recognized, 3750 Vrms for 1 minute. Can be wired as single ended inputs.

Accepts 3.5 to 27Vdc without the need for an external current limiting resistor.

Output

Electrical Characteristics:
Open Collector/Emitter. Opto-isolated. (2500 Vac rms (f=60 Hz, t=1 min. duration)
30Vdc, 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 Earth Ground
Over Temp	Heat Sink temperature exceeds 90° C (195° F)
No Motor	The motor interlock terminals are not connected.

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

Motor Interlock

Two pins that must be shorted together before power will be applied to motor. Can be shorted together through mechanical relay contacts.

Motor Current

SD17060E2 Drivers: Programmable from 1.0 to 6.0 Arms in 0.2 Arms steps.

SD31045E2 Drivers: Programmable from 1.0 to 4.4 Arms in 0.2 Arms steps.

Motor 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 Fuses

10 Amp Slow Blow. Both Line and Neutral are fused. Fuses are not user replaceable.

Environmental Specifications

Input Power	SD17060E2 Drivers: 95 to 132Vac, 50/60 Hz, 5.0 Apk max. SD31045E2 Drivers: 95 to 264Vac, 50/60 Hz, 5.0 Apk max.
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. 4, 6, or 8 lead motor
Insulation	Minimum 500Vdc phase-to-phase and phase-to-case
Inductance	0.3 mH minimum. 2.5 to 45 mH recommended

Connectors

All mating connectors are included with driver.

Connector	Wire	Strip Length	Min./Max. Tightening Torque
I/O	28 - 16 AWG	0.275 inches	1.91/2.23 lb-in (0.22/0.25 Nm)
Motor	24 - 12 AWG	0.275 inches	4.46/5.35 lb-in (0.5/0.6 Nm)
Power	24 - 12 AWG	0.275 inches	4.46/5.35 lb-in (0.5/0.6 Nm)

Indexer Functionality

The table below lists the functionality offered by the indexer built into the AMCI Networked Drivers

Feature	Description
Multiple Network Protocols	The SD17060E2 and SD31045E2 units can be configured to communicate with EtherNet/IP, PROFINET, or Modbus TCP protocols. This allows easy setup and communication with a wide range of host controllers.
EtherNet/IP-DLR	The SD17060E2 and SD31045E2 units act as Beacon-Based Nodes in Device Level Ring environments. This functionality results in the fastest response to changes in the redundant network.
PROFINET-MRP	The SD17060E2 and SD31045E2 units have Media Redundancy Protocol support, which adds redundancy to the PROFINET protocol.
Programmable Inputs	Each of the three inputs can be programmed as a Home Limit, Over Travel Limit, Capture Input, Jog Stop, or E-Stop Input. They can also be programmed to accept a quadrature encoder.
Programmable Output	The single output on the Networked Driver can be programmed as a Fault Output or as a general purpose DC output point.
Programmable Parameters	Starting Speed, Running Speed, Acceleration, Deceleration, and Accel/Decel Types are fully programmable.
Homing	Allows you to set the machine to a known position. The Networked Driver 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 a bit from your controller.
Registration Move	Allows you to jog the motor in either direction based on a command bit from your controller. When a controlled stop is issued, the move will output a programmable number of steps before coming to a stop.
Encoder Registration Move	Uses the count from a quadrature encoder to determine when a move begins to decelerate and stop. Commonly used in spooling/despooling applications.
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.
Assembled Move	Allows you to perform a sequence of relative moves. These moves can be performed with or without stopping between them.
Indexer Move	Allows you to program a move that is run when 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 Networked Driver is configured to accept an encoder, the encoder can be used to verify motion when a move command is issued.
Electronic Gearing	The Networked Driver 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 full programmable and can be changed on-the-fly.

Table R1.1 Indexer Functionality

Driver Functionality

This table summarizes the features of the stepper motor driver portion of the SD17060E2 and SD31045E2.

Feature	Benefits
170 Vdc or 310 Vdc Output Bus	A high voltage output bus means you can derive more high speed torque (power) from your stepper motor.
RMS Current Control	RMS current control give the Networked Driver 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 in 0.2 amp increments. This allows you to use a Networked 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 impedances, 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 Networked Driver 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.
Motor Interlock	Safety feature that prevents power from being applied to the motor connector if a wire jumper does not exist between two of its pins. A *mechanical contact* can be used to disable the current to the motor. The voltage between the two interlock pins never exceeds 5Vdc, but 115/230Vac may be present between either of the Interlock pins and either of the power supply inputs.
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	Protects the Networked Driver 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

Available Inputs

Each Networked Driver has three DC inputs that accept 3.5 to 27Vdc signals. (5 to 24Vdc nominal) They can be wired as differential, sinking, or sourcing inputs. How the Networked Driver uses these inputs is fully programmable as well as their active states. (Each input can be programmed to act as a Normally Open (NO) or Normally Closed (NC) input.)

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. Each Networked Driver allows you to define this starting position in two 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 Networked Driver that will cause the unit to seek this sensor. How the Networked Driver actually finds the home sensor is described in the reference chapter ***Homing an AMCI Networked Driver*** starting on page 57.

Available Inputs (continued)

CW Limit Switch or CCW Limit Switch

Each input can be defined as a CW or CCW Limit Switch. When used 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 a Start Indexer Move Input makes a inactive-to-active state transition. This allows a Networked Driver to run critically timed moves that cannot be reliably started from the network due to data transfer lags.

If Inputs 1 and 2 are programmed as quadrature encoder inputs and Input 3 is programmed as a Start Indexer Move Input, then the quadrature encoder position data will be captured whenever Input 3 transitions. An inactive-to-active state transition 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.

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 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 with this input type, all other moves will ignore this input.

If Inputs 1 and 2 are programmed as quadrature encoder inputs and Input 3 is programmed as a Stop Jog or Registration Move Input, then the quadrature encoder position data will be captured when Input 3 makes an inactive-to-active transition to bring a Jog or Registration Move to a controlled stop. 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 Input 3, 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, a Networked Driver can be configured to capture the encoder position value on a transition on Input 3.

Encoder Feedback

Each Networked Driver can be configured to accept a quadrature encoder instead of discrete inputs. Inputs 1 and 2 can be programmed to accept the ±A and ±B signals from a quadrature encoder. When using an encoder, you have two additional choices when configuring Input 3. First, the ±Z signal can be wired to Input 3 and used to home the machine. Second, Input 3 can be used to capture the encoder position when it transitions if it is programmed as a Capture Encoder Position Input or a Jog Move Stop Input.

Additional information on the ways to use an encoder with a Networked Driver is described in the ***Encoder Functionality*** section on page 18.

Available Inputs (continued)

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 Networked Driver, but the input state is reported in the network data.

Encoder Functionality

Inputs 1 and 2 can be programmed to accept the ±A and ±B signals from a quadrature encoder. When using an encoder, you have two additional choices when configuring Input 3. First, the ±Z signal can be wired to Input 3 and used to home the machine. Second, Input 3 can be used to capture the encoder position when it transitions if it is programmed as a Capture Encoder Position Input or a Jog Move Stop Input.

Using an encoder input gives you the ability to:

- ➤ Home the machine to the encoder marker pulse
- ➤ Detect motor stall conditions
- ➤ Run Encoder Registration Moves

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.

Encoder Registration Move

The encoder position acts as a registration mark in an Encoder Registration Move. Typically, the encoder is mounted separately from the motor. With this mode, the Networked Driver will drive the motor until the desired encoder count is reached, at which time the motor begins to decelerate and stop. This functionality is useful in spool winding or unwinding applications, where the motor drives the spool and the encoder measures the material length as it enters or exits the spool. As the spool diameter increases or decreases, the programmed distance remains the same because you are measuring material, not the number of turns.

Network Interface Description

The network interface is located on the top of the drive, towards the tabbed mounting surface. The network status LED's are also located in this area.

Ethernet Interface

The Ethernet Interface on the SD17060E2 and SD31045E2 drivers has two standard RJ-45 jacks that accept any standard 100baseT cable. Both jacks are used in EtherNet/IP DLR and PROFINET MRP applications. If you are using the Modbus TCP protocol, or are not using DLR or MRP functionality, then you can connect to either jack. Both Ethernet ports on the Ethernet Drivers have "auto switch" capability. This means that a standard cable can be used when connecting the Ethernet Driver to any device. Crossover cables are never needed.

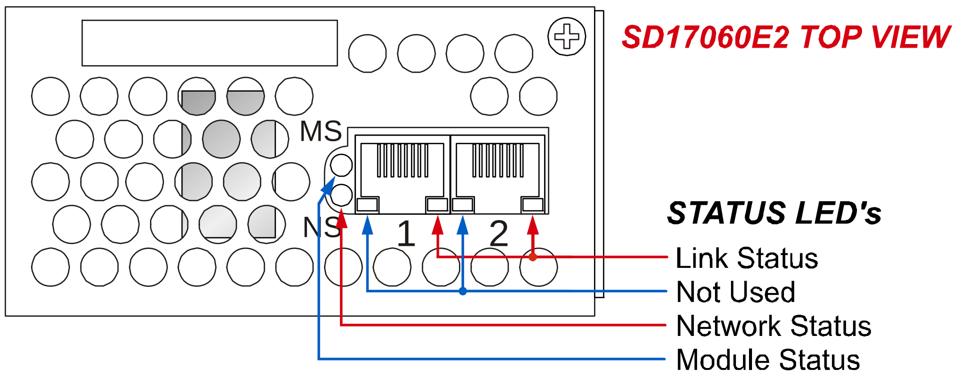


Figure R1.1 Ethernet Interface Location

The two ports are internally connected to an ethernet switch. In non-DLR applications, either port can be used. You also have the ability to daisy chain a second device off of the Networked Driver if this would simplify your network wiring.

There are two LED's on each of the RJ-45 jacks. The left LED is not used. The right LED shows the link status of the Ethernet connection. This LED will be on if the hardware connection is correct.

There are two additional LED's to the left of the jacks. The Module Status (MS) LED gives additional information on the status of the unit. The Network Status (NS) LED gives additional information on the status of the network protocol.

Status LED's

Front Panel Status LED's

- **Power:** This green LED should be on when power is applied to the drive. If the LED is off, the input voltage is below 95Vac or one or both of the internal fuses are blown. The fuses are not field replaceable.
- **Status:**

Steady Green	Driver OK
Steady Red	Short Circuit Fault
	Over Temperature Fault
	Interlock Missing
Blinking Green ...	Successful write to Flash memory. Cycle power to the drive.
Blinking Red	Failure to write to Flash memory. Cycle power to the drive before attempting another write.

A Networked Driver will only detect motor errors when the motor current is enabled.

Status LED's (continued)

Ethernet Status LED's

Link Status – On when there is a physical link between the Ethernet port of the Networked Driver and the Ethernet port of the device the driver is plugged into. This LED will flash when data is being transmitted over the Ethernet link.

Module Status (MS) LED – The Module Status LED is a bi-color red/green LED. The state of the LED depends on the state of the network adapter module.

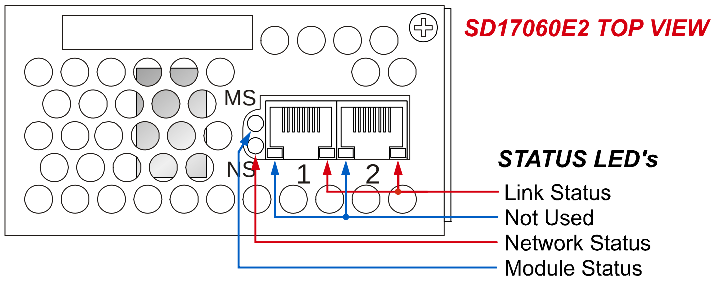


Figure R1.2 Ethernet Status LED's

LED State	EtherNet/IP Definition	Modbus TCP Definition	PROFINET Definition
Off	No Power		
Alternating Red/Green	Power up Self-Test (Occurs very quickly on power up.)		
Flashing Green	Waiting for valid physical connection to the network.		Not Implemented
Steady Green	Drive and Network are operational.		Device Name or IP Address are set.
Flashing Red	Initializing: IP Address Conflict		Initializing: Device Name or IP Address are not set.
	If the Network Status LED is also flashing red, the IP Address or Network Protocol has been changed. Cycle power to the unit to continue. If the Network Status LED is in any other state, a write to flash memory has failed. Cycle power to the unit to clear this fault.		
Steady Red	Major Fault. Cycle power to the unit to attempt to clear the fault.		

Table R1.3 Module Status LED States

Status LED's (continued)

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 OR No IP Address		No power, duplicate IP address on the network, mis-match in Device Name, or no connection to IO Controller.
Alternating Red/Green	Power up Self-Test (Occurs very quickly on power up.)		
Flashing Green	Ethernet connection, but no CIP connections	Indicates number of connec-tions with 2 second delay between groups. The unit supports up to 5 concurrent connections.	On-line, Stop state. A connec-tion with the IO Controller is established and it is in its STOP state.
Steady Green	Valid Ethernet network and CIP connections	Not Implemented	On-line, Run state. A connec-tion with the IO Controller is established and it is in its RUN state.
Flashing Red	If the MS LED is steady green: Network Connection Timeout	Not Implimented	Not Implimented
	If the MS LED is blinking green: IP Address or Network Protocol changed: Cycle power		
Steady Red	Duplicate IP address on network.		Not Implemented

Table R1.4 Network Status LED States

Notes

REFERENCE 2

UL/cUL Recognized Installations

The SD17060E2 and SD31045E2 drivers are Underwriter Laboratory Inc.® listed devices. It is listed as "Industrial Control Equipment" under the control number 60GB. The UL file number is E231137. These units are appropriate for UL and cUL applications.

UL Required Information

If your installation is to meet UL requirements, you must be aware of the following information when using the SD17060E2 or SD31045E2.

- Maximum surrounding air temperature is +50°C
- The driver does not incorporate internal motor overload protection.
- The driver does not provide motor over temperature protection.
- The driver does not provide overspeed protection.
- The driver shall be used in pollution degree 1 or 2 environments. If the driver is mounted in an enclosure, this enclosure must meet these requirements.
- All wiring to the driver shall be R/C (AVLV2), minimum rating of 80°C, 300V, except secondary low-voltage circuit wiring.
- Use 75°C copper conductors only.
- Terminals shall be tighten to manufacturer's recommended torques
- Power Connector shall be rated for a minimum 12A, 600V, in a pollution degree 2 environment†
- Motor Connector shall be rated for a minimum 16A, 600V, in a pollution degree 2 environment†
- I/O Connector shall be rated for a minimum 8A, 300V, in a pollution degree 2 environment†

† The mating connectors supplied with these units meet these requirements. Additional mating connectors can be ordered directly from Phoenix Contact under the following part numbers:

Location	Phoenix Contact Part Number
Power	1737822
Motor	1912919
I/O	1803633

Table R2.1 Mating Connectors

Notes

REFERENCE 3

MOTION CONTROL

When a move command is sent to a Networked Driver, the unit 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^2. However, when programming a Networked Driver, all acceleration and deceleration values must be programmed in the unit of measure of steps/second/millisecond.

- To convert from steps/second^2 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 a Networked Driver.
- To convert from steps/second/millisecond to steps/second^2, multiply the value by 1000. This must be done when converting from the value programmed into a Networked Driver 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 Networked Driver for speed control, don't require position data at all.

Count Direction

Clockwise moves will always 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 +Jog Move command, will result in a clockwise rotation of the 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 2,999,999 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.

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 moves use relative coordinates.

- The range of values for the Target Position when it is treated as an offset is ±8,388,607 counts. 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 ±8,388,607 counts. The only way to move beyond ±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 ±8,388,607 counts. The move will be clockwise 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 ±8,388,607 counts. However, you cannot move beyond ±8,388,607 counts with an Absolute Move. The only way to move beyond ±8,388,607 counts is with multiple relative moves or jog commands.

Definition of Acceleration Types

With the exception of Registration Moves, all move commands, including homing commands, allow you to define the acceleration type used during the move. The Networked Driver supports three types of accelerations and decelerations. The type of acceleration used is controlled by the *Acceleration Jerk* parameter.

Detailed move profile calculations, including the effect of the *Acceleration Jerk* parameter, can be found in the reference section, ***Calculating Move Profiles***, starting on page 47.

Linear 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 Linear Acceleration. Note that other values will work correctly, but you may notice a quick change in velocity at the beginning of the acceleration phase.

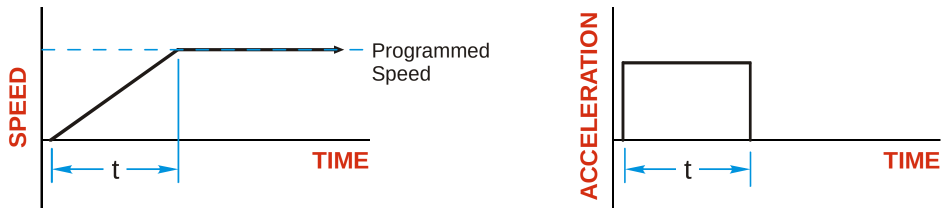


Figure R3.1 Linear Acceleration

Definition of Acceleration Types (continued)

Triangular S-Curve Acceleration

When the Acceleration Jerk parameter equals one, the axis accelerates (or decelerates) at a constantly changing rate that is slowest at the beginning and end of the acceleration phase of the move. The Triangular S-Curve type offers the smoothest acceleration, but it takes twice as long as a Linear Acceleration to achieve the same velocity.

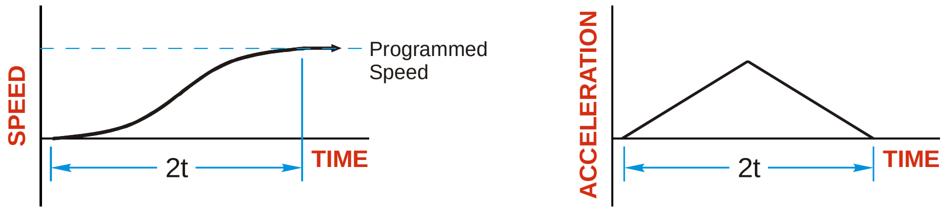


Figure R3.2 Triangular S-Curve Acceleration

Trapezoidal S-Curve Acceleration

When the Acceleration Jerk parameter is in the range of 2 to 5,000, Trapezoidal S-Curve acceleration is used. The Trapezoidal S-Curve acceleration is a good compromise between the speed of Linear acceleration and the smoothness of Triangular S-Curve acceleration. Like the Triangular S-Curve, this acceleration type begins and ends the acceleration phase smoothly, but the middle of the acceleration phase is linear. Figure R3.3 shows a trapezoidal curve when the linear acceleration phase is half of the total acceleration time. With this setting, the Trapezoidal S-Curve acceleration only requires 33% more time to achieve the same velocity as a Linear Acceleration.

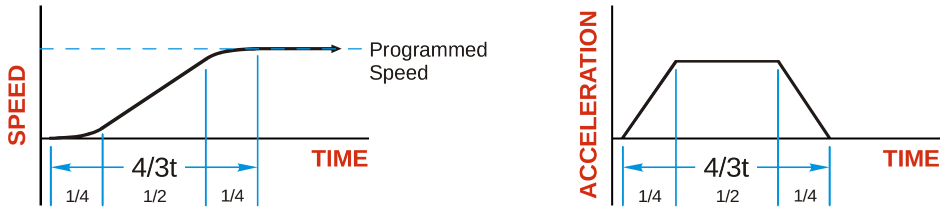


Figure R3.3 Trapezoidal S-Curve Acceleration

A Simple Move

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

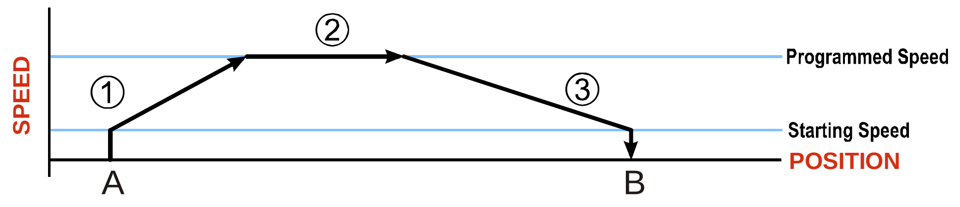


Figure R3.4 A Trapezoidal Profile

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 Networked Driver.

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 R3.4 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 R3.5 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 it has to decelerate to reach the Starting Speed at the Target Position. The Programmed Speed is never reached.

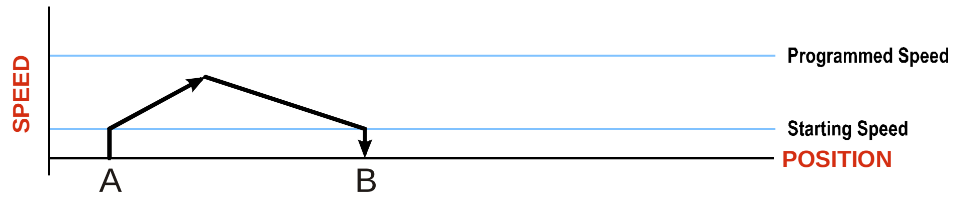


Figure R3.5 A Triangular Profile

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 motion regardless of the speed the motor is running at. Since 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. The machine must be homed or the position must be preset before Absolute Moves can be run again.

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 29, 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

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 *Find Home* commands are explained in the reference section, ***Homing an AMCI Networked Driver***, which starts on page 57. The ***CW/CCW Jog Move*** commands are fully explained on page 31, and the ***CW/CCW Registration Move*** commands are fully explained on page 33.

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. Note that power is not removed from the motor.

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 ±8,388,607 counts. Positive offsets will result in clockwise moves, while negative offsets result in counter-clockwise moves.

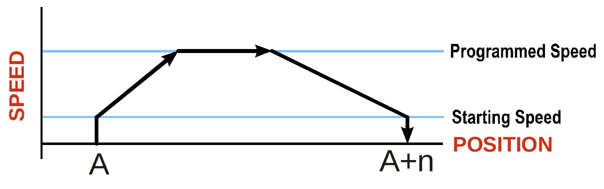


Figure R3.6 Relative Move

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° multiple times without recalculating new target positions in your controller. If you perform the same action with Absolute Moves, you would have to calculate your 30° position followed by your 60° position, followed by your 90° position, etc.

Basic Move Types (continued)

Relative Move (continued)

Relative Moves can be brought to a Controlled Stop by using the Hold Move Command from your host controller. When the command is accepted, the axis will immediately decelerate at the programmed rate and stop. When stopped successfully, the Networked Driver will set a *In_Hold_State* bit in the input data table. The Relative Move can be restarted with the Resume Move command from the host controller or the move can be aborted by starting another move. 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. The use of the Hold_Move and Resume_Move bits is further explained in the ***Controlling Moves In Progress*** section starting on page 42.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the Network Output Data.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW or 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.

Absolute Move

Absolute Moves move from the Current Position (A) to a given position (B). (The Networked Driver calculates the direction and 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 ±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.

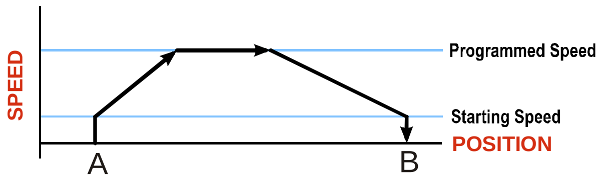


Figure R3.7 Absolute Move

NOTE

1) The *Home Position* of the machine must be set before running an Absolute Move. See the reference section, ***Homing an AMCI Networked Driver***, which starts on page 57, 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° will move the table to the correct position regardless of where the move starts from.

Basic Move Types (continued)

Absolute Move (continued)

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 the move can be aborted by starting another move. The use of the Hold_Move and Resume_Move bits is explained in the ***Controlling Moves In Progress*** section starting on page 42.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the Network Input Data.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW or 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.

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 increase the motor position count while the CCW Jog Move will decrease the motor position count. These commands are often used to give the operator manual control over the axis.

Jog Moves are also 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.

Basic Move Types (continued)

CW/CCW Jog Move (continued)

As shown below, a Jog Moves 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 Networked Driver 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 unit that is less than the starting speed, the Jog Move will continue at the previously programmed speed.

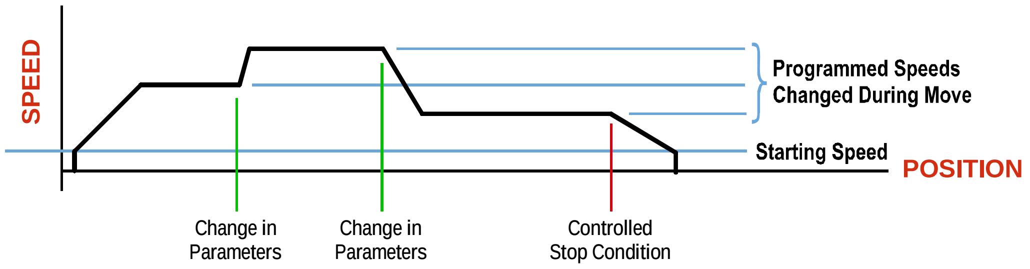


Figure R3.8 Jog Move

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 42.

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 or 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 CW Jog Move while the CCW Limit Switch is active.

Basic Move Types (continued)

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 increase the motor position count while the CCW Registration Moves decrease the motor position count. When the command terminates under Controlled Stop conditions, the Networked Driver 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.

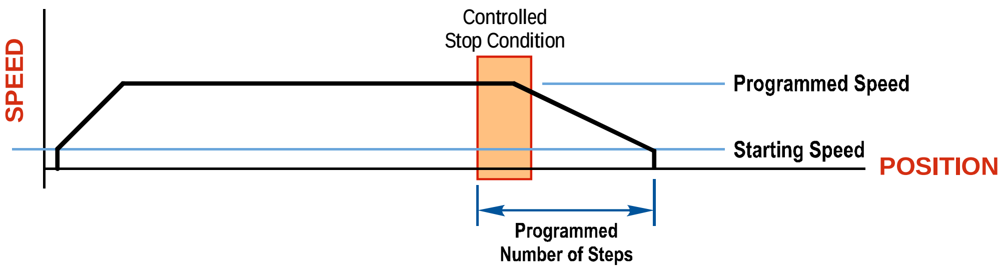


Figure R3.9 Registration Move

NOTE 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 Networked Driver will decelerate at the programmed Deceleration value until it has output the Programmed Number of Steps and then stop the move without further deceleration.

An additional feature of the Registration Moves 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.

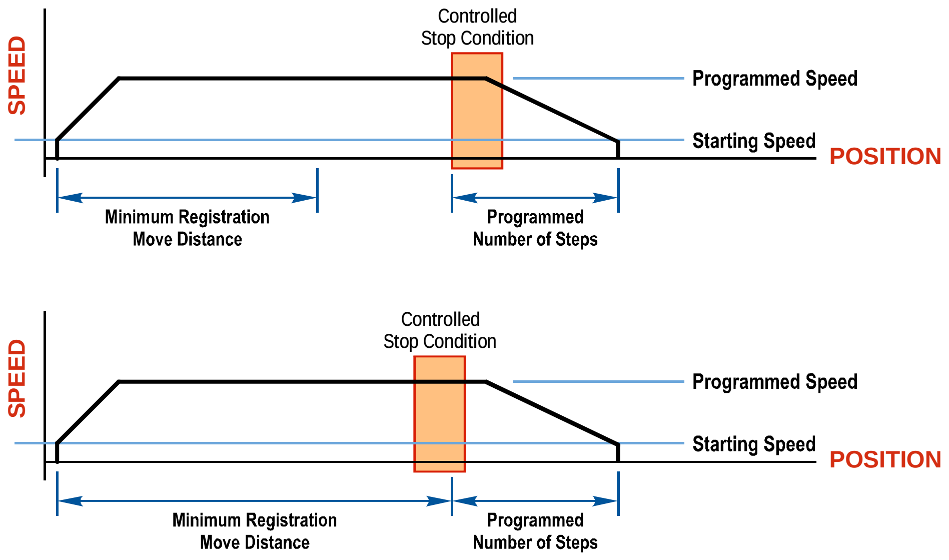


Figure R3.10 Min. Registration Move Distance

Basic Move Types (continued)

CW/CCW Registration Move (continued)

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 Networked Driver responds by using the programmed Deceleration value to bring the move to a stop, without using the value of the Programmed Number of Steps parameter. 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.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW or 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 CW Registration Move while the CCW Limit Switch is active.

Encoder Registration Move

When the Networked Driver is configured to use a quadrature encoder, the position value from the encoder can be used as a registration mark. Once the encoder count is reached, the move will begin to decelerate and stop when the move slows to the configured Starting Speed. Absolute and relative type move types are supported. When programming an absolute move, the target position defines the encoder position that must be reached before the move begins to decelerate and stop. When programming a relative move, the sign of the target position determines the direction of the move and its magnitude determines the distance traveled.

NOTE

1) The encoder is typically mounted separate from the motor. For example, the encoder can measure the length of a flexible and/or stretchable material, or material on a spool. The motor is used to drive the material for the length measured by the encoder, regardless of the slipping, flexing, or stretching, of the material as it is driven by the motor.
2) This is an open-loop move. The encoder does not provide feedback in a closed-loop, servo control fashion. The encoder position is used to decide when the move deceleration begins, not the final position of the move.
3) You do not have to preset the position or home the machine before you can use a relative Encoder Registration Move.

Basic Move Types (continued)

Encoder Registration Move (continued)

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.

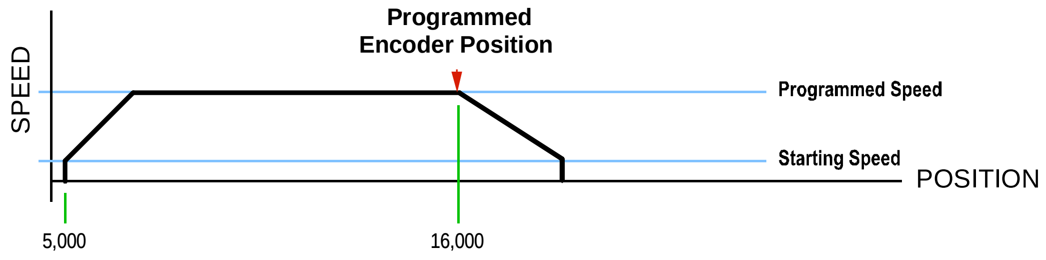


Figure R3.11 Encoder Registration Move

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. The use of the Hold_Move and Resume_Move bits is explained in the ***Controlling Moves In Progress*** section starting on page 42.

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/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.

Assembled Moves

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

- **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. A Blend Move can be run in either direction, and the direction is set when the move command is issued.
- **Dwell Move -** A Dwell Move gives you the ability to string multiple relative moves together, and the Networked Driver will stop between each move for a programed *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 Networked Driver before the move can be initiated.
   - The Networked Driver 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 37.
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.

NOTE The deceleration value programmed with segment 3 is used twice in the segment. Once to decelerate from the Programmed Speed of segment 2 and once again to decelerate at the end of the move.

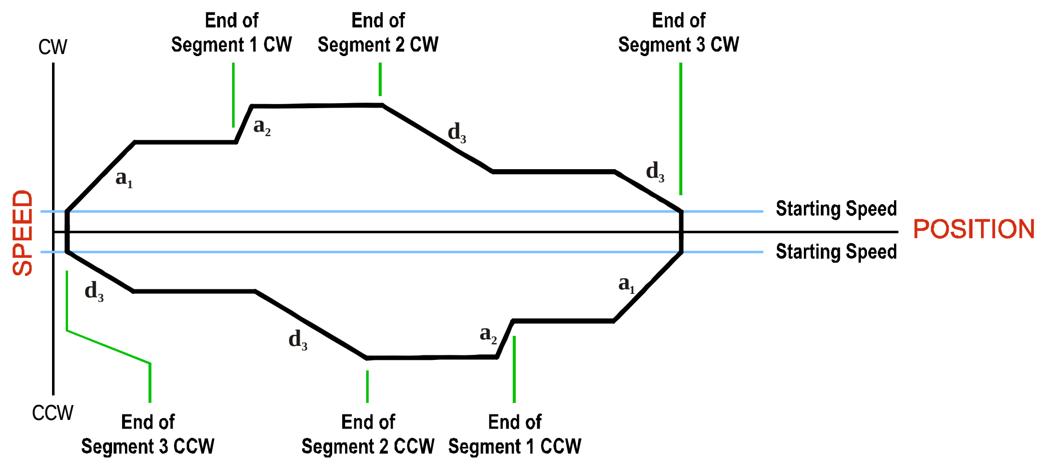


Figure R3.12 Blend Move

Assembled Moves (continued)

Blend Move (continued)

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 Networked Driver 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 unit. As described in ***Saving an Assembled Move in Flash*** on page 39, 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 Networked Driver 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.

Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold_Move control bit in the Network Output Data. When this occurs, the Networked Driver 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. A Blend Move that is brought to a controlled stop with the Hold_Move bit cannot be restarted. The use of the Hold_Move bit is explained in the ***Controlling Moves In Progress*** section starting on page 42.

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 or 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.

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 a Networked Driver 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 programed *Dwell Time*. The Dwell Time is programmed as part of the command that starts the move. The Dwell Time is the same for all 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 shaft rotation while a negative segment will result in a counter-clockwise shaft rotation.

Assembled Moves (continued)

Dwell Move (continued)

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 *could* accomplish this Dwell Move with a series of six relative moves that are sent down to the Networked Driver sequentially. The two advantages of a Dwell Move in this case are that the Networked Driver will be more accurate with the Dwell Time then you can be in your control program, and Dwell Moves simplify your program's logic.

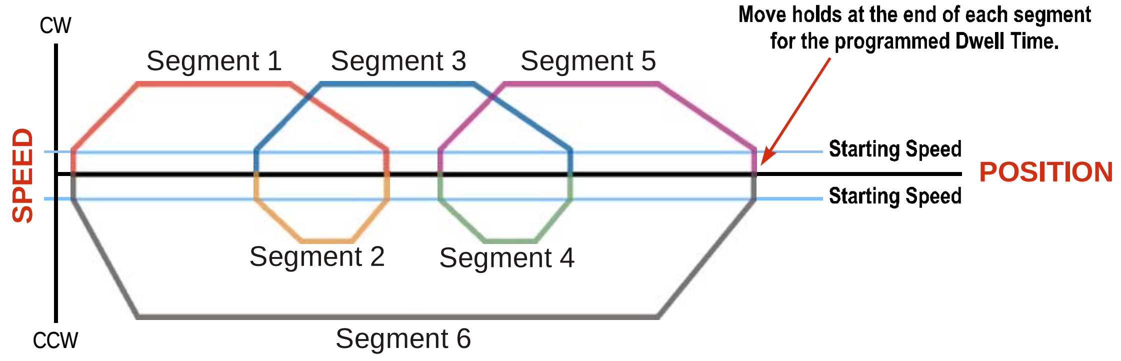


Figure R3.13 Dwell Move

NOTE

1) You do not have to preset the position or home the machine before you using 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 Networked Driver 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 Networked Driver. As described in ***Saving an Assembled Move in Flash*** on page 39, 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 your Networked Driver 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 Networked Driver 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. A Dwell Move that is brought to a controlled stop with the Hold_Move bit 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 or 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.

Assembled Move Programming

All of the segments in a Blend or Dwell Move must be written to the Networked Driver 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 signs of the segment's Target Positions are ignored and all segments are run in the same direction. In the case of a Dwell Move, the signs of the segment's Target Positions determine the direction of the segment. For Dwell Moves, the Dwell Time is sent to the Networked Driver as part of the command.

Control Bits – Output Data

- **Program_Assembled bit –** Set this bit to tell the Networked Driver that you want to program a Blend or Dwell Move Profile. The Networked Driver will respond by setting the *In_Assembled_Mode* bit in the Network Input Data. At the beginning of the programming cycle, the Networked Driver will also set the *Waiting_For_Assembled_Segment* bit to signify that it is ready for the first segment.
- **Read_Assembled_Data bit –** Set this bit to tell the Networked Driver that the data for the next segment is available in the remaining data words.

Control Bits – Input Data

- **In_Assembled_Mode bit –** The Networked Driver 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 –** The Networked Driver will set this bit to signal the host that it 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 Networked Driver 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 Networked Driver 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 Networked Driver 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 Networked Driver resets the *In_Assembled_Mode* and the *Waiting_For_Assembled_Segment* bits.

Saving an Assembled Move in Flash

The Networked Driver 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 and have the *Save_Assembled_to_Flash* bit set during step 8. At step 9 in the sequence, the Networked Driver responds by resetting the *In_Assembled_Mode* and *Transmit Blend Move Segments* bits as usual and then flashes 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 Networked Driver before you can continue. With a limit of 10,000 write cycles, the design decision that requires you to cycle power to the Networked Driver 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 inputs instead of a command from the network. If the *Indexed Move* bit is set when the command is issued, the Networked Driver 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 Networked Driver until the triggered move is finished.
- As stated above, a move can be run multiple times as long at 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 16#0000 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, the Networked Driver 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 Networked Driver 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 have the encoder option.
- 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 Networked Driver. 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 Networked Driver units, this parameter has a range of 21 to 65,535.

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

NOTE When using the Networked Driver 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 Networked Driver 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 Networked Driver 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.

A sample program that demonstrates virtual axis programming on the ControlLogix platform is available on the AMCI website at ***https://www.amci.com/industrial-automation-support/sample-programs/***. It can be found in the *Stepper Motor Control* section of that page. The sample includes the setup required to run the axis as a high priority event driven periodic task. If you are using a different host controller that supports motion axis programming, feel free to contact AMCI technical support for assistance in programming your controller.

Blended 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. A Blend Move is programmed into the Networked Driver before it is run and the unit precisely controls position and velocity throughout the entire move. Additional information on ***Blend Move*** functionality can be found starting on page 36.

Controlling Moves In Progress

Each Networked Driver 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 a new move of any type from the stopped position.

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

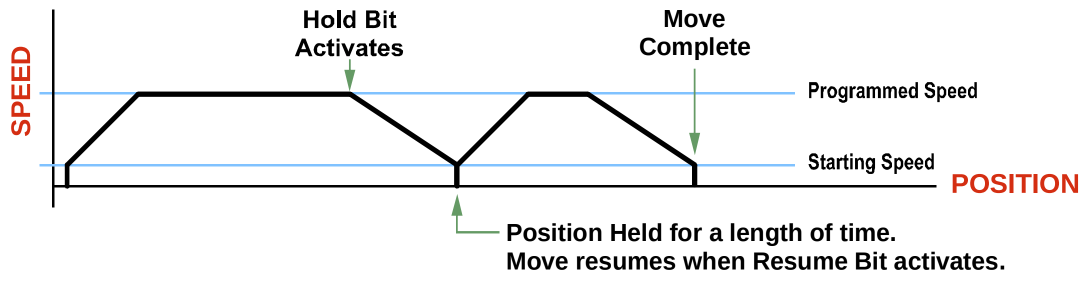


Figure R3.14 Hold/Resume a Move Profile

Find Home Moves

A Find Home command can be placed in a Hold state but cannot be resumed. This give 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 Networked Driver 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 Networked Driver 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 give 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 a Networked Driver is Electronic Gearing. A quadrature encoder is required but it is not mounted on the motor controlled by the Networked Driver. 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 Networked Driver has three parameters that allow you to set any turns ratio you want between the encoder and the motor.

Motor_Resolution

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 Networked Driver. It has a range of 200 to 32,767. If you set the Motor_Resolution parameter to four times the number of lines of your encoder, then by default your motor will complete one rotation for every rotation of the encoder.

ELGearing Multiplier and Divisor

The ratio of these two parameters sets the number of motor rotations per encoder rotation. Each parameter has a range of 1 to 255.

How It Works

The Networked Driver always uses 4X decoding when counting pulses from the encoder. If you set the Motor_Resolution parameter to four times the number of encoder lines, and set both of the ELGearing Multiplier and Divisor parameters to 1, then the motor will complete one rotation for every rotation of the encoder's shaft.

Once placed in Electronic Gearing mode, the Networked Driver monitors the Jog Move command bits in the Network Output Registers. When either of these bits are set, the encoder inputs are monitored for a change in position. When a change is sensed, the Networked Driver 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 Networked Driver will accelerate or decelerate the motor to match the new ratio.

Encoder position data can be captured and reported to the host controller while in Electronic Gearing mode by configuring Input 3 as a Capture Encoder Position input.

NOTE You must set the Acceleration and Deceleration parameters before you can put the Networked Driver in Electronic Gearing mode.

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 CCW 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_Resolution parameter.

The Z pulse is not used to correct the encoder position once per turn, so you can actually program the Motor_Resolution parameter 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_Resolution 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.

Stall Detection

Another feature available when using an encoder is stall detection. The encoder must be mounted on the motor controlled by the Networked Driver, which means that you cannot use Stall Detection when using the Electronic Gearing feature. When Stall Detection is enabled, the Networked Driver monitors the encoder inputs for changes while a move is in progress. If the encoder inputs do not change as expected, the move stops and an error bit is reported to your host controller.

In order for the Stall Detection to work correctly, you must program the *Encoder_Resolution* parameter to its correct value in the Configuration Data of the Networked Driver. The Networked Driver always uses X4 decoding when determining the encoder position value, so the *Encoder_Resolution* parameter must be set to four times the number of encoder lines. (When using a 1,024 line encoder, the *Encoder_Resolution* parameter must equal 4,096 for stall detection to work correctly.)

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^2R 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 4Arms and the current reduction to 25% will result in an idle current of 1Apk. (The Networked Driver always switches from RMS to peak current control when the motor is idle to prevent motor damage due to excessive heating.)

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.

Assuming a stable line voltage of 115Vac, the following gains can be used for AMCI motors. These gain settings are factory suggestions and are average settings for our motors. Your system may benefit from increasing or decreasing these settings. In general, increase the setting by one or two counts to improve high speed performance or decrease the settings by one or two for quieter low speed operation.

		115 Vac GAIN SETTINGS						
	MOTOR CURRENT →	1 A	1.5 A	2 A	3 A	4 A	5 A	6 A
MOTOR PART #	SM2340-130			2	3	4		
	SM2340-240			4	5	6		
	SM34-450			6	9	11		
	SM34-850			11	14	17		
	SM34-1100			12	17	21		
	SM23-130 (Series)	4	6	7				
	SM23-130 (Parallel)			2	3	4		
	SM23-240 (Series)	6	8	10				
	SM23-240 (Parallel)			4	5	6		
	SM42-1800				10	11	12	14

Table R3.1 Optional 115Vac Current Loop Gain Settings

Current Loop Gain (continued)

When using the SD31045E2 with a stable 230 Vac line voltage, the following gains can be used for AMCI motors. These gain settings are factory suggestions and are average settings for our motors. Your system may benefit from increasing or decreasing these settings. In general, increase the setting by one or two counts to improve high speed performance or decrease the settings by one or two for quieter low speed operation.

		230Vac GAIN SETTINGS					
	MOTOR CURRENT →	1 A	1.5 A	2 A	3 A	4 A	4.4 A
MOTOR PART #	SM2340-130			1	1	1	
	SM2340-240			2	3	3	
	SM34-450			3	4	5	
	SM34-850			6	7	8	
	SM34-1100			8	9	10	
	SM23-130 (Series)	2	3	3			
	SM23-130 (Parallel)			1	1	1	
	SM23-240 (Series)	3	4	5			
	SM23-240 (Parallel)			2	3	3	
	SM42-1800				5	6	7

Table R3.2 Optional 230Vac Current Loop Gain Settings

REFERENCE 4
Calculating Move Profiles

This reference was added because some of our customers must program very precise profiles. Understanding this section is not necessary before programming the Networked Driver and it 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 appendix use a unit of measure of steps/second/second (steps/second^2) for acceleration and deceleration. However, when programming the Networked Driver, all acceleration and deceleration values must be programmed in the unit of measure of steps/second/millisecond.

- To convert from steps/second^2 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 Networked Driver.
- To convert from steps/second/millisecond to steps/second^2, multiply the value by 1000. This must be done when converting from the value programmed into the Networked Driver 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 steps/sec^2. For example, if the choose acceleration is 20,000 steps/sec^2, the smoothest transition occurs when the starting speed is 141. ($141^2 \approx 20{,}000$)

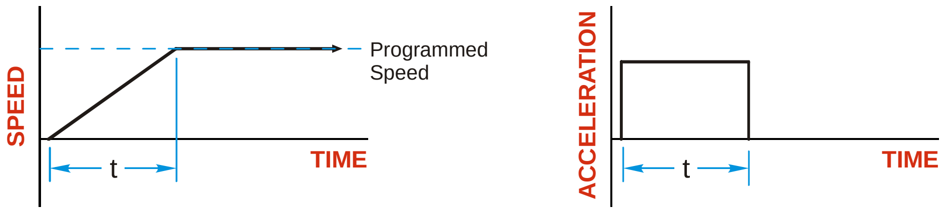
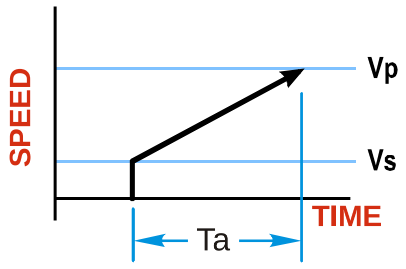


Figure R4.1 Constant Acceleration Curves

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 steps/second^2
- **d** = Deceleration value. Must be in the units of 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 R4.1 gives the equations to calculate Time, Distance, and Acceleration values for a constant acceleration move.

Acceleration Type	T_A or T_D (Time to Accelerate or Decelerate)	D_A or D_D (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 R4.1 Acceleration Equations

If the sum of the D_A and D_D 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_A and D_D 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_A and D_D 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 R4.2 for a move profile.

Name	Value	SMD34E2 Parameter Values
Acceleration (a)	20,000 steps/sec^2	20
Deceleration (d)	25,000 steps/sec^2	25
Starting Speed (V_S)	141 steps/sec	141
Programmed Speed (V_P)	100,000 steps/sec	100,000

Table R4.2 Sample Values

From figure R4.1:

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 Networked Driver will generate a Trapezoidal profile. If the move is equal to 449,984 steps, the Networked Driver will generate a Triangular profile and the unit will output one pulse at the programmed speed. If the move is less than 449,984 steps, the Networked Driver 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_M) attained during the move. Because the move is always accelerating or decelerating, the total distance traveled is equal to the sum of D_A and D_D.

$D_A = T_A*(V_M + V_S)/2$ 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 R4.2, assume a total travel distance of 300,000 steps.

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

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

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

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

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

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

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

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

Once you have calculated the maximum speed, you can substitute this value into the time and distance formulas in table R4.1 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_T below), from the second equation.

D_P = (Total Number of Steps) – (D_A + D_D)

$T_T = T_A + T_D + D_P/V_P$

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

$T_T = T_A + T_D$

S-Curve Acceleration Equations

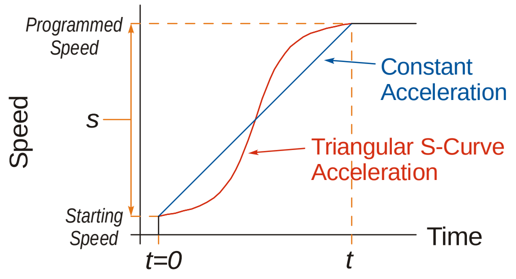
When the Acceleration Jerk parameter value is in the range of 1 to 5,000, the Networked Driver uses this value to smoothly change the acceleration value applied during the move. In this case, the speed of the move does not increase linearly, but exponentially, resulting in an "S" shaped curve. This limits 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 Networked Driver below sixteen bits, the Acceleration Jerk parameter programmed into the driver does not have units of steps/sec^3. The Acceleration Jerk parameter equals ({100 * jerk in steps/sec^3} / acceleration in steps/sec^2). This translates to the jerk property in steps/sec^3 equalling ({Acceleration Jerk parameter/100} * 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 0.01*a* to 50*a* where "*a*" is the acceleration value in steps/sec^2. For example, if the acceleration is programmed to 20,000 steps/sec^2, then the value of the jerk property used by the unit can be programmed to be between 200 steps/sec^3 (0.01*20,000) and 1,000,000 steps/sec^3 (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 R4.2 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 = Programmed Speed – Starting Speed

$$\text{Acceleration} = \frac{\text{speed}}{\text{time}} \qquad \text{jerk} = \frac{\text{acceleration}}{\text{time}} \qquad \text{Unit's Acceleration Jerk Parameter(J)} = \frac{100j}{a}$$

$$a = \frac{s}{t} \qquad j = \frac{a}{t} \qquad j = \frac{Ja}{100}$$

$$at = s \qquad jt = a$$

Figure R4.2 Move Profile Example

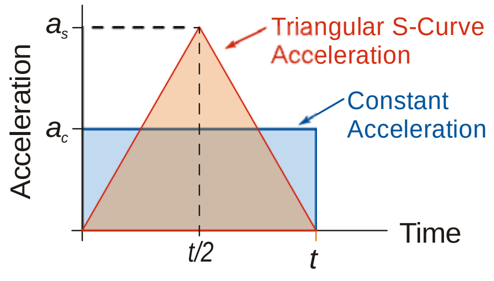


Figure R4.3 Triangular Acceleration

Speed is equal to acceleration multiplied by the time it is applied. This is shown graphically in figure R4.3 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:

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

$$a_s = 2a_c$$

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.

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

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

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

$$Ja_st = 200a_s$$

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

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 $a_s = a_c$

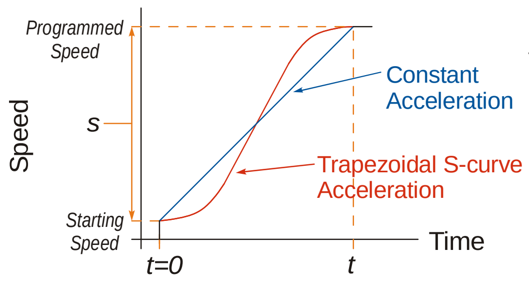
The above examples assume that you can increase the programmed acceleration value to keep the acceleration time the same. If your constant acceleration value is the maximum your system will allow, then using S-curve accelerations will lengthen the time needed to accelerate to your desired speed.

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^2 that is applied for 2.0 seconds. If the acceleration value must remain at 20,000 steps/sec^2, 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 R4.4 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.



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

$$\text{Acceleration} = \frac{\text{speed}}{\text{time}} \qquad \text{jerk} = \frac{\text{acceleration}}{\text{time}} \qquad \text{Unit's Acceleration Jerk Parameter(J)} = \frac{100j}{a}$$

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

$$at = s \qquad jt = a \qquad \Longrightarrow \qquad j = \frac{Ja}{100}$$

Figure R4.4 Move Profile Example

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

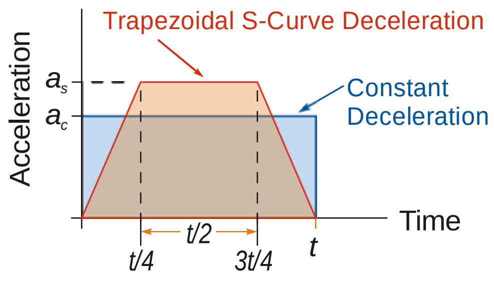


Figure R4.5 Trapezoidal Acceleration

Speed is equal to acceleration multiplied by the time it is applied. This is shown graphically in figure R4.5 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.

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

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

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

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

This means that a trapezoidal S-curve acceleration profile that is 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.

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

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

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

$$Ja_st = 400a_s$$

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

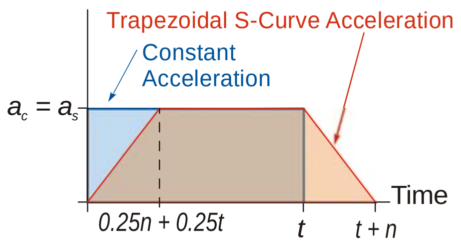
This value represents the ideal Acceleration Jerk parameter value for a trapezoidal S-curve acceleration with a constant acceleration for half of the phase. Setting the value lower than this will result in a shorter constant period, while setting the value greater than this will result in a longer constant 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 R4.6. 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.



$$a_c(t) = a_c(.5n + .5t) + a_c(.25n + .25t)$$

$$a_c(t) = a_c((.5n + .5t) + (.25n + .25t))$$

$$t = .75n + .75t$$

$$0.25t = .75n$$

$$t = 3n$$

$$t/3 = n \Rightarrow t+n = t + t/3 = 4/3t = 1.3333t$$

Figure R4.6 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^2 that is applied for 2.0 seconds. If the acceleration value must remain at 15,000 steps/sec^2, 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^2. 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

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

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

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

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

S_m	=	midpoint of change in speed
J	=	Acceleration Jerk parameter
j	=	physical jerk property
a_f	=	calculated final acceleration

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

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

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

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

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^2, 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

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

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

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

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

S_m	=	midpoint of change in speed
J	=	Acceleration Jerk parameter
j	=	physical jerk property
a_f	=	calculated final acceleration

Just as displacement $= \frac{1}{2}at^2$**, 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^2, the resulting acceleration is a trapezoidal S-curve. As shown in figure R4.7, 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).

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

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

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

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

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

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

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

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

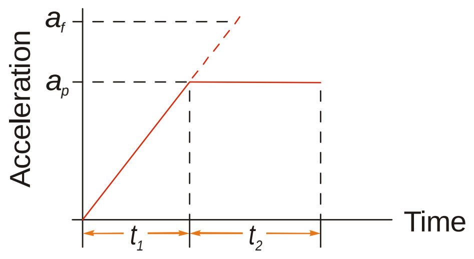


Figure R4.7 Calculating Trapezoidal S-Curve

The time for this acceleration phase is 2(t1 + t2), which equals 2(0.2500 sec + 0.1336 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

REFERENCE 5
HOMING AN AMCI NETWORKED DRIVER

This chapter explains the various ways of homing a Networked Driver. Inputs used to home the unit are introduced and diagrams that show how the unit 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 a Networked Driver must be set to an appropriate value. If you use the unit's *CW/CCW Find Home* commands, the motor position register will automatically be set to zero once the home position is reached. *The Encoder Position register will also be reset to zero if the encoder is available and enabled.*

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

With the exception of Absolute Moves, a Networked Driver 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 Networked Driver. On units with integral encoders, both the motor position and the encoder position can be preset separately, and the motor position can also be preset to the encoder position. The motor and encoder position values can be preset anywhere in the range of –8,388,607 to +8,388,607.

CW/CCW Find Home Commands

The other choice is to use the driver's Find Home commands to order the Networked Driver 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 positive direction and ends when the home sensor triggers while the Networked Driver is rotating in the positive direction at the starting speed. The CCW Find Home command operates in the same way but starts and ends with motion in the negative direction.

Homing Inputs

Four inputs can be used when homing the driver. These inputs are either physical inputs attached to the unit or bits in the network output data words.

Physical Inputs

- **Home Input:** This input is used to define the actual home position of 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

- **Backplace_Proximity_Bit:** A Networked Driver can be configured to ignore changes on the physical homing input until the Backplace_Proximity_Bit makes a 0→1 transition. The Networked Driver 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.

Homing Configurations

A Networked Driver must have one of its DC inputs configured as the home input before one of the *CW/CCW Find Home* commands can be issued.

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 Networked Driver 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) You can use a bit in the Network Output Data (the Backplace_Proximity_Bit) as a home proximity input. Using this bit is completely optional and prevents the Home Input from being acted upon until the Backplace_Proximity_Bit makes a 0→1 transition.

Homing Profiles

NOTE

1) 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.
2) The homing diagrams show CW and CCW direction on their vertical axes. These are the directions that will occur if using an AMCI motor that is wired as shown in this manual. It is possible to reverse the direction of rotation by swapping one connections on one of the motor windings.

Home Input Only Profile

Figure R5.1 below shows the move profile generated by a +Find Home command when you use the Home Input without the Backplace_Proximity_Bit.

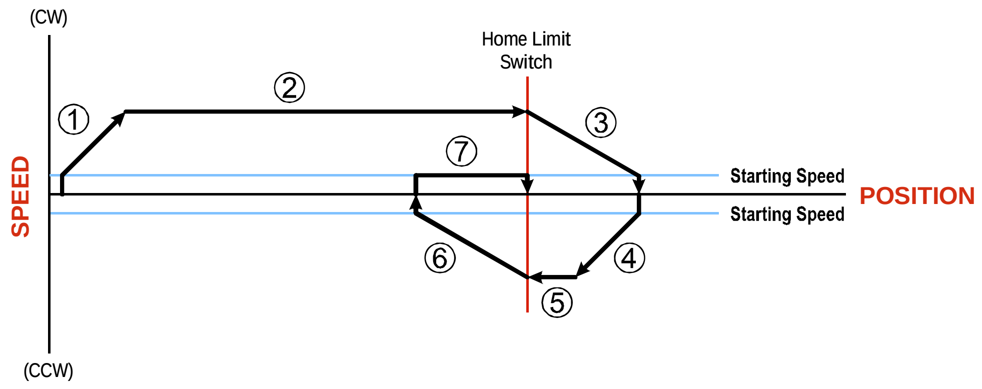


Figure R5.1 Home Input Profile

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 Network Backplace_Proximity_Bit

Figure R5.2 below shows the move profile generated by a +Find Home command when you use the Home Input with Network Backplace_Proximity_Bit.

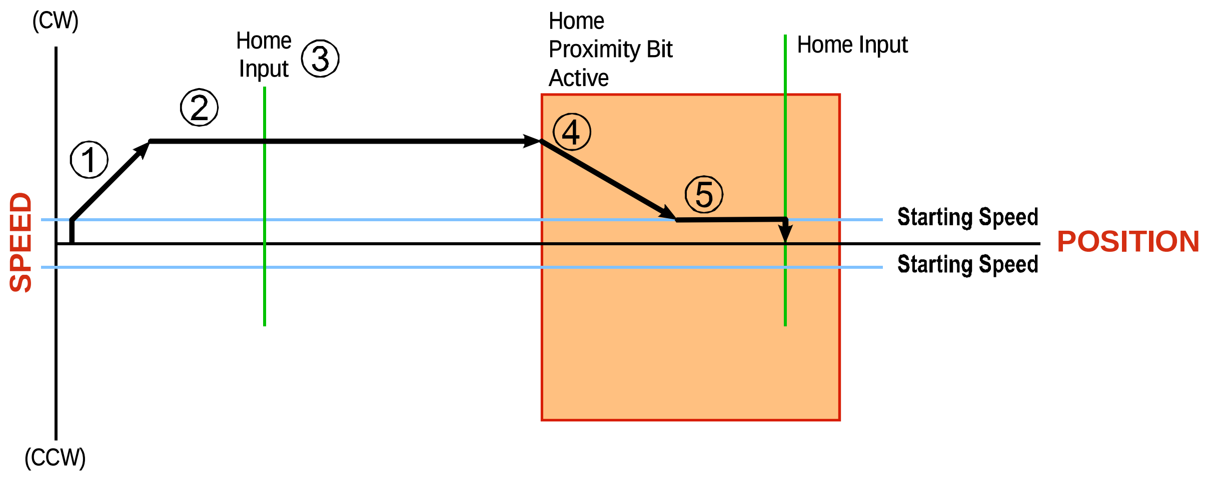


Figure R5.2 Homing with Proximity

1) Acceleration from the configured Starting Speed to the Programmed Speed
2) Run at the Programmed Speed
3) Ignores the Home Input because Backplace_Proximity_Bit has not made a 0→1 transition.
4) Deceleration towards the Starting Speed when the Backplace_Proximity_Bit transitions from 0 to 1. 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 R5.2 shows the Backplane Backplace_Proximity_Bit staying active until the Networked Driver reaches its home position. This is valid, but does not have to occur. As stated in step 4, the Networked Driver starts to hunt for the home position as soon and the Backplane Backplace_Proximity_Bit makes a 0→1 transition

Homing Profiles (continued)

Profile with Overtravel Limit

Figure R5.3 below shows the move profile generated by a +Find Home command when you use:

- CW Overtravel Limit
- Home Input without the Backplace_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 motor will stop all motion and issue a *Home Invalid* error to your host.

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

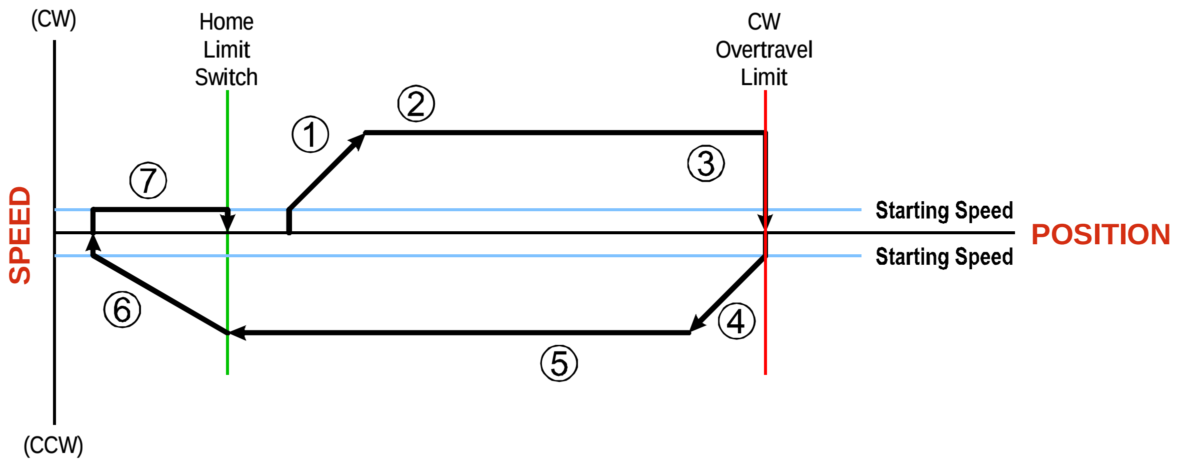


Figure R5.3 Profile with Overtravel Limit

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.

REFERENCE 6
CONFIGURATION MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to configure the AMCI Networked Driver as well as the formats of the Network Input Data that contains the responses from the device. An AMCI Networked Driver requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data.

Modes of Operation

Each AMCI Networked Driver 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. Typically, the Networked Driver should be configured on every power up. This will help ease system maintenance if you ever need to replace the drive. However, note that the motor will not receive power until the driver is properly configured.

Command Mode

This mode gives you the ability to program and execute stepper moves, and reset errors when they occur. The Networked Driver will always power up in this mode, but a valid configuration must be sent to the Networked Driver 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

A Networked Driver will always power up in Command Mode. The unit will use its stored configuration data to configure the unit if it is available. The Networked Driver will then check for valid network command data and will only enable the motor driver section if the Enable_Driver bit is set.

➤ If using EtherNet/IP with an EDS file, the Networked Driver is sent configuration data when it connects to the network. This data is used to configure the unit. (This data is used in place of any configuration that may be stored in flash memory.) The Networked Driver will then check for valid network command data and will only enable the motor driver section if the Enable_Driver bit is set.

Word Addressing

Data format tables in this reference chapter contain EtherNet/IP, PROFINET, and Modbus TCP address columns. Modbus addresses given in the tables follow the zero-based addressing method. If you need additional information on Modbus addressing schemes, please refer to the task chapter ***Modbus TCP Configuration***, starting on page 133.

Configuration Mode Data Format

A Networked Driver requires twenty bytes of Output Data as well as twenty bytes of Input Data. In EtherNet/IP applications that use an EDS file, these twenty bytes will be a mixture of single and double integers. In EtherNet/IP applications that do not use an EDS file, the Networked Driver is added to the network as a generic device. In this case the data is represented as ten 16-bit (single) integers. This ten word format is also used in Modbus TCP and PROFINET applications.

Configuration Mode Data Format (continued)

Multi-Word Data Format

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. In applications that do not use an EDS file, this parameter is transmitted in two separate words. The table below shows how values are split. Note that negative values are written as negative numbers in both words.

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

Table R6.1 Multi-Word Format Examples

NOTE EtherNet/IP applications that use a non-generic EDS file specify the Starting Speed in the configuration registers that are created when the Networked Driver was added to the project. The EDS file creates a double precision integer for the Starting Speed parameter and it is programmed as a thirty-two bit integer. The multi-word format shown above is not used.

Command Mode Data Formats

When issuing commands to the Networked Driver, 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 Networked Driver 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 Networked Driver using the multi-word format described above. When configuring the Networked Driver, 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 Networked Driver when issuing commands. The **Binary_Input_Format Bit,** controls the format of the data values written to the host controller by the Networked Driver.

NOTE EtherNet/IP applications that use a non-generic EDS file have the parameters and data values listed above defined as double integers. In these applications, the **Binary_Output_Format Bit,** and the **Binary_Input_Format Bit** must both be set to "1" so that the data is transferred as thirty-two bit integers.

When using the signed thirty-two bit format, there is an additional parameter named Data Endian. Host controllers that use the EtherNet/IP protocol typically use *little endian* format, while controllers that use PROFINET or the Modbus protocol typically use the *big endian* format.

Command Mode Data Formats (continued)

Examples of the formats are given below.

	Multi-Word Format		32 bit Signed Integer Little Endian Format (EtherNet/IP)		32 bit Signed Integer Big Endian Format (PROFINET & Modbus TCP)	
Value	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 R6.2 Position Data Format Examples

NOTE When using the Modbus-TCP protocol, the host controller should use "signed 16-bit integer" as the data type when using the multi-word format.

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 Networked Driver is in Configuration Mode is shown below.

EtherNet/IP or PROFINET 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 Resolution	200 to 32,767
5	1029	Reserved	Must be set to zero
6	1030	Encoder Resolution	0 to 32,768
7	1031	Idle Current Reduction	0 to 100%
8	1032	Motor Current (X10)	10 to 60, Even numbers only. Represents 1.0 to 6.0 Arms
9	1033	Current Loop Gain	1 to 40

Table R6.3 Network Output Data Format: Configuration Mode

Output Data Format (continued)

Configuration Word 0 Format

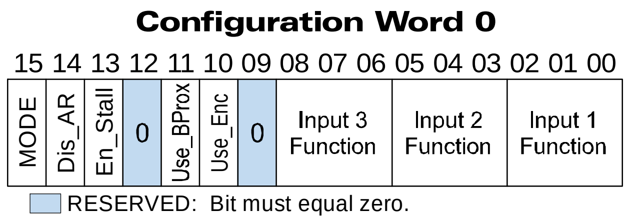


Figure R6.1 Configuration Word 0 Format

Bit 15: **Mode –** "1" for Configuration Mode Programming, "0" for Command Mode Programming. Unless a configuration has been saved to Flash memory, the Networked Driver powers up in Command Mode and shows a configuration error. (Hexadecimal value of 6408h.) The Networked Driver will not power the motor or accept commands until a valid configuration is available.

Bit 14: **Disable_Antiresonance –** "1" disables the antiresonance feature. "0" enables the anti-resonance feature of the Networked Driver. 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 –** "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 Networked Driver. The Encoder_Resolution parameter must also be programmed and must be four times the line count of your encoder.

Bit 11: **Use_Backplane_Proximity –** "0" when the Backplace_Proximity_Bit is not used when homing the Networked Driver. "1" when the Backplace_Proximity_Bit is used when homing the Networked Driver. Note that this bit is not the Backplace_Proximity_Bit, but enables or disables its operation. Do not use the Backplace_Proximity_Bit if you only want to home to the Home Limit Switch. (Leave this bit equal to "0".)

Bit 10: **Use_Encoder –** "0" when Quadrature Encoder is not used. "1" to enable a Quadrature Encoder. You must also configure Inputs 1 and 2 to accept quadrature signals and enter a value in Word 6 for the Encoder_Resolution parameter in addition to setting this bit.

Bit 9: **Reserved –** Must equal zero.

Output Data Format (continued)

Configuration Word 0 Format (continued)

Bits 8-6: Input 3 Function – See Table Below

Bits 5-3: Input 2 Function – See Table Below

Bits 2-0: Input 1 Function – See Table Below

Bits				
8	7	6		
5	4	3		
2	1	0	Function	Available On
0	0	0	General Purpose Input	All Inputs - The input is not used in any functions of the Networked Driver, but it's status is reported in the Network Data. This allows the input to be used as a discrete DC input to your controller.
0	0	1	CW Limit	All Inputs
0	1	0	CCW Limit	All Inputs
0	1	1	Start Indexed Move	All Inputs
0	1	1	Start Indexed Move / Capture Encoder Value	All Inputs, or Input 3 when Inputs 1 & 2 are configured as quadrature encoder inputs. The encoder position value is captured whenever Input 3 transitions. An inactive-to-active state transition will also trigger an Indexed Move if one is pending in the Networked Driver.
1	0	0	Stop Jog or Registration Move	All Inputs
1	0	0	Stop Jog or Registration Move & Capture Encoder Value	All Inputs, or Input 3 when Inputs 1 & 2 are configured as quadrature encoder inputs. The encoder position value is captured when Input 3 triggers a controlled stop to a Jog or Registration move.
1	0	1	Emergency Stop	All Inputs
1	1	0	Home	All Inputs when using discrete sensors. Input 3 only when using quadrature encoder. Use this configuration value when homing to the Z-Pulse of an encoder.
1	1	1	Quadrature Encoder Input	Inputs 1 and 2 only. Input 1 is channel A. Input 2 is channel B.

Table R6.4 Configuration Data: Input Function Selections

NOTE When using a quadrature encoder, you must set bit 10, the Use_Encoder Bit, in addition to programming the inputs to accept quadrature signals. You must also program the Encoder_Resolution parameter in configuration word 6.

Output Data Format (continued)

Configuration Word 1 Format

Configuration Word 1

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0/1	OUT1Fn	OUT1AOCL	OUT1SACL	ReadConfig	SaveConfig	BinInFrmt	BinOutFrmt	DataEndian	0	0	0	0	IN3_ActLvl	IN2_ActLvl	IN2_ActLvl

🟦 RESERVED: Bit must equal zero.

Figure R6.2 Configuration Mode: Config Word Format

Bit 15: **Reserved –** State ignored.

Bit 14: **OUT1_Function –** “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 an ON state when power is applied to the Networked Driver and it has not yet been configured.

Bit 13: **OUT1_Action_on_Connection_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: **OUT1_State_at_Connection_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 Networked Driver responds by placing the present configuration data in the Network Input Data. You cannot write new configuration data to the unit 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 68.

Bit 10: **Save_Present_Configuration –** An AMCI Networked Driver will store its configuration data to flash memory when this bit makes a 0→1 transition. The validity of the configuration data is checked before being written. If the data is not correct, the transition on this bit is ignored. If the write to flash completes successfully, the Networked Driver will write 16#AAAA into the last word of the Network Input Data and the Status LED will start flashing green. If the write is unsuccessful, the Networked Driver will write 16#EEEE into the last word of the Network Input Data and the Status LED will start flashing red. Once the Networked Driver issues its response to the Save_Present_Configuration command, it stops responding to commands and you must cycle power to the unit. This design decision prevents the Networked Driver from responding to constant save commands from the host controller.

NOTE

1) This feature was added to support users whose host controllers have very limited functionality. Consider the consequences of using this feature. Adding the code necessary to write down the configuration to a Networked Drive on power up or network connection is fairly straight forward on most PLC based hosts. Adding this code allows you to easily change the configuration in the host and easily configure a new drive if you ever need to swap a drive out of the machine. Sample code from AMCI includes this functionality.

2) The endurance of the Flash memory is a minimum of 10,000 write cycles.

3) If using the EDS file to configure the Networked Driver, the configuration data should be entered on the Configuration tab when setting the properties of the driver. This configuration data is written to the driver whenever the network connection is established.

Output Data Format (continued)

Configuration Word 1 Format (continued)

Bit 9: **Binary_Input_Format –** Set to “0” to have the Motor Position, Encoder Position, and Trapped Encoder Position reported in the ***Multi-Word Data Format*** shown in table R6.1 on page 62. Set to “1” to have the Motor Position, Encoder Position, and Trapped Encoder Position reported as signed 32-bit integers with the location of the least significant bits determined by the value of the Binary_Endian bit. (See bit 7 below.)

Bit 8: **Binary_Output_Format –** Set to “0” to program the Target Position and Programmed Speed parameters in the ***Multi-Word Data Format*** shown in table R6.1 on page 62. Set to “1” to program the Target Position and Programmed Speed parameters in signed 32-bit integer format with the location of the least significant bits determined by the value of the Binary_Endian bit. (See bit 7 below.)

Bit 7: **Binary_Endian –** Only used when bits 8 and/or 9 (Binary_Output_Format and Binary_Input_Format) are set to “1”. Set to “0” to have 32-bit values transmitted in little endian format. Set to “1” to have the 32-bit values transmitted in big endian format. EtherNet/IP applications should reset this bit to “0”. Modbus TCP and PROFINET applications should start with this bit set to “1”. These protocols typically use big endian format, but you should refer to you PLC’s documentation to verify the format used by your processor. See table R6.2, ***Position Data Format Examples*** found on page 63 for examples of the different formats.

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

Bit 2: **IN3_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: **IN2_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: **IN1_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.

NOTE The AMCI Networked Driver does not report the state of the inputs when they are configured as encoder inputs (ABZ). The driver reports the state of the *Active Level Bit* for the input.

Notes on Other Configuration Words

- Information on the ***Multi-Word Data Format*** used when programming the Starting Speed can be found on page 62.
- Changes to the Idle Current only take effect at the *end of the first move after re-configuration.*
- ***Current Loop Gain*** settings for all AMCI motors can be found starting on page 45.

Input Data Format

The correct format for the Network Input Data when the Networked Driver is in Configuration Mode is shown below.

EtherNet/IP or PRIOFINET Word	Modbus/TCP Register	Configuration Data
0	0	Configuration Word 0
1	1	Mirror of Output Data Config Word
2	2	Mirror of Starting Speed.
3	3	
4	4	Mirror of Motor_Resolution
5	5	0
6	6	Mirror of Encoder_Resolution
7	7	Mirror of Idle_Current_Reduction
8	8	Mirror of Motor Current (X10)
9	9	Mirror of Current Loop Gain

Table R6.5 Network Input Data Format: Configuration Mode

Configuration Word 0 Format

When the Configuration data is valid and accepted, this word mirrors the value of the Configuration Word 0 written to the Networked Driver. When the Networked Driver is not configured, or the data written to it is invalid, then this word has the same format of Status Word 0 when the Networked Driver is in Command Mode. This format is explained in the ***Status Word 0 Format*** section starting on page 85. On power up, the value of this word will be 6408h if the Networked Driver does not have a valid configuration in its flash memory.

NOTE When in Configuration Mode, bit 13 of word 0 is set to "1" when stall detection is enabled. When in Command Mode, bit 13 of word 0 is set to "1" when there is a configuration error. When using the state of bit 13 of word 0 in your logic, always include the state of bit 15 of word 0 to assure that you are only acting on the bit when in the proper mode.

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 function, such as two CW Limit Switches.
4) You configure the Networked Driver to use an encoder, but you do not configure Inputs 1 and 2 as Quadrature Encoder Inputs.
5) You configure the Networked Driver to use an encoder, but you do not set the Encoder_Resolution value in word 6.
6) Setting the Enable_Stall_Detection Bit without configuring the Networked Driver to use an encoder.

REFERENCE 7
COMMAND MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to command the SD17060E2 or SD31045E2 as well as the formats of the Network Input Data that contains the responses from the device. The Networked Driver requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data. Note that the Networked Driver, when in its factory default state, will power up in Configuration Mode. The Configuration Mode data format is described in the previous chapter.

Data Format

A Networked Driver requires twenty bytes of Output Data as well as twenty bytes of Input Data. In EtherNet/IP applications that use a non-generic EDS file, these 20 bytes will be a mixture of single and double integers. In EtherNet/IP applications that do not use an EDS file, or a generic EDS file, the Networked Driver is added to the network as a generic device. In this case the data is represented as ten 16-bit (single) integers. This ten word format is also used in Modbus TCP and PROFINET applications.

Sixteen bit integers support a range of values from -32,768 to 32,767 or 0 to 65,535. When issuing commands to the Networked Driver, 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 Networked Driver 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 Networked Driver using the multi-word format described below. When configuring the Networked Driver, it is possible to program it to use a 32-bit double integer format instead of the custom format shown below.

There are three configuration bits that control the data format when the Networked Driver is in command mode. The **Binary_Output_Format Bit,** controls the format of the programmable parameters written to the Networked Driver when issuing commands. The **Binary_Input_Format Bit,** controls the format of the data values written to the host controller by the Networked Driver. When either of these parameters are set to their 32-bit signed integer format settings, the Data_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.

	Multi-Word Format		32 bit Signed Integer Little Endian Format (EtherNet/IP)		32 bit Signed Integer Big Endian Format (Modbus TCP specification)	
Value	**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 R7.1 Position Data Format Examples

Data Format (continued)

NOTE EtherNet/IP applications that use a non-generic EDS file have the parameters and data values listed above defined as double integers. In these applications, the Binary_Output_Format Bit, and the Binary_Input_Format Bit must both be set to “1”. The Data_Endian Bit must be reset to “0”. The data will then be transferred as thirty-two bit signed integers in little endian format.

NOTE When using the Modbus-TCP protocol, use “signed 16-bit integer” as the data type when using the multi-word format.

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 Word 0 before writing the next command.

This condition also applies when switching from Configuration Mode to Command Mode. If a bit is set in Configuration Word 0 while in Configuration Mode and you switch to Command Mode with the same bit set, the command will not occur because the bit must transition between writes to the unit.

Output Data Format

The following table shows the format of the output network data words when writing command data to the SD17060E2 or SD31045E2.

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

Figure R7.1 Command Data Format

Command Word 0

Command Word 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MODE	Prst_Enc	Run_AMov	Rd_AData	Prg_Assm	RSet_Err	Prst_Pos	Jog_CCW	Jog_CW	Home_CCW	Home_CW	I-Stop	Resm_Mv	Hold_Mv	Rel_Mv	Abs_Mv

Figure R7.2 Command Word 0 Format

Bit 15: Mode_Select – "1" for Configuration Mode Programming "0" for Command Mode Programming. A factory default SD17060E2 or SD312045E2 powers up in Command Mode and shows a configuration error. (Hexadecimal value of 6408h.) The Networked Driver will not power the motor or accept commands until a valid configuration is written to it.

Bit 14: Preset_Encoder – When set to "1" the Networked Driver will preset the Encoder Position to the value stored in Output Words 2 and 3.

Bit 13: Run_Assembled_Move – When set to "1" the Networked Driver 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.
- **Reverse_Blend_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 11 & 12: 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 39.

Bit 10: Reset_Errors – When set to "1" the Networked Driver will clear all existing errors, reset the *Move_Complete* bit, and attempt to use the present data to run a new command.

Bit 9: Preset_Position – When set to "1" the Networked Driver will preset the Motor Position to the value stored in Output Words 2 and 3 and reset the Move_Complete bit in the Network Input Data.

Bit 8: Jog_CCW – When set to "1" the Networked Driver 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 31.

- **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 – Command Word 1, Bit 6:** When this bit equals "1" the Networked Driver will switch its operation to ***Electronic Gearing*** mode as described on page 43. 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.
- **Axis Follower Mode – Command Word 1, Bits 12 and 14:** If either of these bits equal "1" when the Registration_Move bit equals "0", the Networked Driver will enter its Axis Follower Mode when the Jog Move bit is set to "1".

Command Word 0 (continued)

Bit 7: **Jog_CW –** When set to “1” the Networked Driver will run a Jog Move in the clockwise direction. The full explanation of a ***CW/CCW Jog Move*** can be found starting on page 31.

- **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 – Command Word 1, Bit 6:** When this bit equals “1” the Networked Driver will switch its operation to ***Electronic Gearing*** mode as described on page 43. 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.
- **Axis Follower Mode – Command Word 1, Bits 12 and 14:** If either of these bits equal “1” when the Registration_Move bit equals “0”, the Networked Driver will enter its Axis Follower Mode when the Jog Move bit is set to “1”.

Bit 6: **Find_Home_CCW –** When set to “1” the Networked Driver 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 an AMCI Networked Driver*** reference chapter starting on page 57.

Bit 5: **Find_Home_CW –** When set to “1” the Networked Driver will attempt to move to the Home Limit Switch in the clockwise direction. A full explanation of homing can be found in the ***Homing an AMCI Networked Driver*** reference chapter starting on page 57.

Bit 4: **Immediate_Stop –** When set to “1” the Networked Driver 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.

Bit 3: **Resume_Move –** Set to “1” to resume a move that you previously placed in a hold state. Use of the Resume_Move and Hold_Move bits can be found in the ***Controlling Moves In Progress*** section of this manual starting on page 42. 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. Use of the Hold_Move and Resume_Move bits can be found in the ***Controlling Moves In Progress*** section of this manual starting on page 42.

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 29.

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 30.

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	OUT1_State	V_Pos_Follow	BP_Prox	Clr_Drv_Flt	AsMv_Type	Index_Cmd	Reg_Move	En_ElGear	SvA_to_Flash or Curent_Key2	Rev_BlendDir	0	Enc_Reg_Mv	Current_Key1	Current_Key0

Figure R7.3 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 Networked Driver before the driver can be enabled.

Bit 14: Virtual_Encoder_Follower – When using the Networked Driver 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: OUT1_Set_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 networked Driver 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: Backplace_Proximity_Bit – When the Networked Driver is configured to use the Backplace_Proximity_Bit, the unit 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. Further information on using the Backplace_Proximity_Bit can be found in the ***Profile with Network Backplace_Proximity_Bit*** section found on page 59.

Bit 10: Clear_Driver_Fault – If this bit is set when a Reset Errors Command is issued, (Command Word 0 Bit 10) the Networked Driver will attempt to clear driver errors such as a missing interlock jumper or motor short fault. Note that the driver must be disabled (Command Word 1, Bit 15 = 0), when using this command.

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 Reverse_Blend_Direction bit, (Command Word 1, 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 Networked Driver 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.

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 – Set to "1" to put the Networked Driver in Electronic Gearing mode. Set to "0" for normal operation. A full description of Electronic Gearing mode starts on page 23.

Bit 5: Save_Assembled_to_Flash - Set this bit to save the data of a programmed Assembled Move. This bit is only acted upon this way when the Program_Assembled bit (Command Word 0, Bit 11 makes a 1→0 transition as explained in the ***Assembled Move Programming*** section of this manual starting on page 39.
OR
Motor_Current_Key2 - See ***Description of Motor Current Keys*** on the following page.

Command Word 1 (continued)

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: **Reserved –** Must be set to “0”.

Bit 2: **Encoder_Registration_Move –** This bit is used with the Relative_Move (Command Word 0, bit 1) or Absolute_Move (Command Word 0, bit 0) bits. Set this bit to “1” to command an Encoder Registration Move. This move will run until the encoder count reaches the programmed relative or absolute value and then the move will begin to decelerate and stop. Must equal “0” for normal relative or absolute moves. A full description of an ***Encoder Registration Move*** can be found starting on page 18.

Bit 1: **Motor_Current_Key1 –** See the description below.

Bit 0: **Motor_Current_Key0 –** See the description below.

Description of Motor Current Keys

With the exceptions of the CW/CCW Registration Move commands, it is possible to change the motor current “on the fly” while in Command Mode. To do this, place the new motor current in Word 8 and set the key bits as shown below.

- Control Word 2: Bit 5, (Key2) – “1”
- Control Word 2: Bit 1, (Key1) – “0”
- Control Word 2: Bit 0, (Key0) – “1”

This feature does not work with CW/CCW Registration Move commands because Words 8 and 9 are used to store the Minimum Registration Distance parameter. For all other commands, the Networked Driver will respond by changing the motor current and writing the new value into the Network Input Data, Word 8. Once the motor current has been set, reset the key bits to all zeros to prevent further changes to the motor current. The motor current can be changed at any time, including during a move.

NOTE Setting the motor current to an invalid value does not force a command error. The value will be ignored and the Networked Driver will use the motor current value programmed while in Configuration Mode. Invalid values are:

- Value whose last digit is not even
- Values outside the range of 1.0 to 6.0 amps on the SD17060E2
- Values outside the range of 1.0 to 4.4 amps on the SD31045E2

Command Blocks

The following section lists the output data format for the sixteen different commands.

Absolute Move

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0001
1	1025	***Command Word 1***		See pg. 73
2	1026	Abs. Target Position: Upper Word	Steps	Combined value between –8,388,608 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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Acceleration Jerk		0 to 5000

Table R7.2 Absolute Move Command Block

Relative Move

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0002
1	1025	***Command Word 1***		See pg. 73
2	1026	Rel. Target Position: Upper Word	Steps	Combined value between –8,388,608 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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Acceleration Jerk		0 to 5000

Table R7.3 Relative Move Command Block

Command Blocks (continued)

Hold Move

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0004
1	1025	***Command Word 1***		See pg. 73
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 R7.4 Hold Move Command Block

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

Resume Move

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0008
1	1025	***Command Word 1***		See pg. 73
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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Acceleration Jerk		0 to 5000

Table R7.5 Resume Move Command Block

Unused words are ignored by the Networked Driver 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 or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0010
1	1025	***Command Word 1***		See pg. 73
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 R7.6 Immediate Stop Command Block

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

Find Home CW

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0020
1	1025	***Command Word 1***		See pg. 73
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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Acceleration Jerk		0 to 5000

Table R7.7 Find Home CW Command Block

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

Command Blocks (continued)

Find Home CCW

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0040
1	1025	***Command Word 1***		See pg. 73
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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Acceleration Jerk		0 to 5000

Table R7.8 Find Home CCW Command Block

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

Jog CW

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0080
1	1025	***Command Word 1***		See pg. 73 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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Acceleration Jerk		0 to 5000

Table R7.9 Jog Move CW Command Block

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

Command Blocks (continued)

Registration Move CW

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0080
1	1025	***Command Word 1***		See pg. 73 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 R7.10 Registration Move CW Command Block

Virtual Axis Follower Move

EtherNet/IP or PROFINET Word	Modbus TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0080 or 16#0100
1	1025	***Command Word 1***		See pg. 73 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 R7.11 Axis Follower 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 is 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)

Jog CCW

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0100
1	1025	***Command Word 1***		See pg. 73 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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Acceleration Jerk		0 to 5000

Table R7.12 Jog CCW Command Block

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

Registration Move CCW

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0100
1	1025	***Command Word 1***		See pg. 73 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 R7.13 Registration Move CCW Command Block

Command Blocks (continued)

Encoder Follower Move

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0080 or 16#0100
1	1025	***Command Word 1***		See pg. 73 Bit 6 must equal “1”
2	1026	Electronic Gearing Numerator		1 to 255
3	1027	Electronic Gearing Denominator		1 to 255
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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Unused		See Note Below

Table R7.14 Encoder Follower Move Command Block

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

Preset Position

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0200
1	1025	***Command Word 1***		See pg. 73
2	1026	Position Preset Value: Upper Word	Steps	Combined value between –8,388,608 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	Unused		See Note Below
9	1033	Unused		See Note Below

Table R7.15 Preset Position Command Block

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

Presetting the position will also reset the *Move_Complete* status bit in the Network Input Data.

Command Blocks (continued)

Reset Errors

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0400
1	1025	***Command Word 1***		See pg. 73 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	Unused		See Note Below
9	1033	Unused		See Note Below

Table R7.16 Reset Errors Command Block

Unused words are ignored by the Network Driver and can be any value, including parameter values in the previous command. Resetting errors will also reset the *Move_Complete* status bit in the Network Input Data.

Run Assembled Move

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#2000
1	1025	***Command Word 1***		See pg. 73 Blend Move: Bit 9 = "0" Dwell Move: Bit 9 = "1" Reverse_Blend_Direction is 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 A	SD17060E2: 10 to 60 SD31045E2: 10 to 44 *Even numbers only.*
9	1033	Unused with Blend Move Dwell Time with Dwell Move	milliseconds	0 to 65,535

Table R7.17 Run Assembled Move Command Block

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

Command Blocks (continued)

Preset Encoder Position

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#4000
1	1025	***Command Word 1***		See pg. 73
2	1026	Encoder Preset Value: Upper Word	Steps	Combined value between –8,388,608 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	Unused		See Note Below
9	1033	Unused		See Note Below

Table R7.18 Preset Encoder Position Command Block

Unused words are ignored by the Networked Driver 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 or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0800
1	1025	***Command Word 1***		See pg. 73
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 R7.19 Assembled Move First Programming Block

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

Once the first block is transmitted, the Networked Driver responds by setting bits 8 and 9 in Status Word 0. (See ***Status Word 0 Format*** starting on page 85.) Once these are set, you can then start transmitting Segment Blocks.

Segment Block

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#1800
1	1025	***Command Word 1***		See pg. 73
2	1026	Rel. Target Position: Upper Word	Steps	Combined value between –8,388,608 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	Unused		See Note Below
9	1033	Acceleration Jerk		0 to 5000

Table R7.20 Assembled Move Segment Programming Block

Note that each Segment Block starts with bits 11 and 12 set in the Command Word 0 word (16#1800). When the Networked Driver sees bit 12 of Command Word 0 set, it will accept the block and reset bit 9 in Status Word 0. When your program sees this bit reset, it must respond by resetting bit 12 of Command Word 0. The Networked Driver will respond to this by setting bit 9 in Status Word 0 and the next Segment Block can be written to the Networked Driver. 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 Networked Driver is in Command Mode is shown below.

EtherNet/IP or PROFINET Word	Modbus/TCP Register	Command Mode Input Data
0	0	Status Word 0
1	1	Status Word 1
2	2	Motor Position: Upper Word
3	3	Motor Position: Lower Word
4	4	Encoder Position: Upper Word
5	5	Encoder Position: Lower Word
6	6	Trapped Encoder Position: Upper Word
7	7	Trapped Encoder Position: Lower Word
8	8	Programmed Motor Current (X10)
9	9	Value of Acceleration Jerk Parameter

Table R7.21 Network Input Data Format: Command Mode

Format of Position Data Values

The format of the Motor Position, Encoder Position, and Trapped Encoder Position values is controlled by the Binary_Input_Format bit in the configuration data written to the Networked Driver. (See ***Configuration Word 1 Format***, bit 9 starting on page 66.) When the Binary_Input_Format bit equals "0", the position values are reported with the bottom three digits of the value in the lower word (000 - 999) and the remaining digits in the upper word. See ***Position Data Format Examples*** on page 69 for an explanation of this format. When the Binary_Input_Format bit equals "1", the position values are reported as 32-bit signed integers, with the location of the least significant bits dependent on the selected endian format.

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.

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 R7.4 Command Mode: Status Word 0 Format

Bit 15: Mode_Flag – Set to "1" if in Configuration Mode, and set to "0" if in Command Mode. The Networked Driver powers up in Command Mode and shows a configuration error. The Networked Driver will not power the motor or accept commands until a valid configuration is written to it.

Bit 14: Module_OK – "1" when the Networked Driver is operating without a fault, "0" when an internal fault condition exists.

Input Data Format (continued)

Status Word 0 Format (continued)

Bit 13: Configuration_Error – "1" on power up before a valid configuration has been written to the Networked Driver or after any invalid configuration has been written to the driver. "0" when the Networked Driver has a valid configuration written to it.

NOTE When in Command Mode, bit 13 of word 0 is set to "1" when there is a configuration error. When in Configuration Mode, bit 13 of word 0 is set to "1" when stall detection is enabled. When using the state of bit 13 of word 0 in your logic, always include the state of bit 15 of word 0 to assure that you are only acting on the bit when in the proper mode.

Bit 12: Command_Error – "1" when an invalid command has been written to the Networked Driver. This bit can only be reset by the Reset_Errors bit, Command Word 0, Bit 10. Note that setting the motor current to a value outside the range of 1.0 to 6.0 amps on the SD17060E2, or 1.0 to 4.4 amps on the SD31045E2, does not force a command error. The driver simply ignores the new value and uses the last accepted value for the motor current.

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 82.

Bit 10: Position_Invalid – "1" when:

- ➤A configuration is written to the Networked Driver
- ➤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 Networked Driver 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 39.

Bit 8: In_Assembled_Mode – The Networked Driver 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 39.

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 Networked Driver, 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.

Input Data Format (continued)

Status Word 0 Format (continued)

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 is 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	OUT1_State	0	Heartbeat_Bit	Limit_Condition	Inv_Jog_Par	0	Driver_Fault	Connect_Lost	PLC_In_PRG	Temp_90C	0	IN3Active	IN2_Active	IN1_Active

Figure R7.5 Command Mode: Status Word 1 Format

Bit 15: **Drive_Is_Enabled –** Set to “1” when the motor driver section of the Networked Driver is enabled and current is available to the motor. Set to “0” when the motor driver section is disabled. If this bit is set to “1”, the motor current remains present when an E-Stop input is active. Motor current is removed if there is a Driver_Fault (Bit 7 below) regardless of the state of this bit. Motor current is also removed if the motor is idle and Idle Current Reduction is programmed to its *To 0%* setting.

Bit 14: **Stall_Detected –** Set to “1” when a motor stall has been detected.

Bit 13: **OUT1_State –** Present actual state of Output 1. 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 unit and network connection are operating correctly. Note that this bit is only available while the Networked Driver is in Command Mode.

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: **Reserved –** Will always equal zero.

Input Data Format (continued)

Status Word 1 Format (continued)

Bit 7: **Driver_Fault –** If the driver section of the Networked Driver is enabled, this bit will be a "1" during a Over temperature Fault, a Short Circuit Fault, or when the Interlock Jumper is missing. 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". For additional information, see ***Notes on Clearing a Driver Fault*** on page 89. Note that this bit is not set if the Interlock Jumper is missing when power is applied to the driver.

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. Note that this bit is not set if the communication loss is on the protocol level, not the physical level.

Bit 5: **PLC_in_PROG_Mode –** On ControlLogix and CompactLogix platforms, this bit will equal "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 above 80°C. If this bit trips often and you want to lower the operating temperature of the unit, consider changing how the device is mounted, or installing a fan to force additional airflow through the device.

Bit 3: **Reserved –** 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 66.

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 66.

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 66.

NOTE The Networked Driver does not report the state of the inputs when they are configured as encoder inputs (ABZ). The driver reports the state of the *Active Level Bit* for the input which is set in Configuration Mode.

Notes on Clearing a Driver Fault

A Driver Fault occurs when there is an over temperature condition, a short circuit condition in the motor, or the Interlock jumper is missing.

When a Driver Fault occurs, the driver sets bit 7 of Status Word 1 in the Network Input Data. (See ***Status Word 1 Format*** on page 87 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 or PROFINET Word	Modbus/TCP Register	Function	Units	Range
0	1024	***Command Word 0***		16#0400
1	1025	***Command Word 1***		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 R7.22 Reset Driver Fault Command Block

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

Once the command block is accepted by the Networked Driver, 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 Enabled_Driver 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.

Notes

REFERENCE 8
General Installation Guidelines

This chapter gives general information on installing electronic controls in an industrial environment including the importance of proper wiring, grounding, and surge suppression. If you are responsible for installing an SD17060E2 or SD31045E2, make sure you are familiar with these practices and follow them when installing the unit.

These instructions assume a solidly grounded system, which is used in a vast majority of modern industrial systems. As defined by the IEEE, a solidly grounded system is one in which the neutral of a generator, power transformer, or grounding transformer is directly connected to the system ground or earth. The installation of ungrounded systems is not covered by these instructions.

WARNING This chapter is presented as a tool in the hopes of avoiding common installation problems. It is not a substitute for the safety practices called out in local electrical codes or, in the United States, the National Electrical Code published by the National Fire Protection Association. If *any* conflicts exist, local and national codes must be followed. *It is the responsibility of the user* to determine what installation practices must be followed to conform to all local and national codes.

Safety is the single most important objective of any electrical installation. The system must be safe to use and it must be able to detect unsafe conditions and remove power when these conditions occur.

The second important objective of an electrical installation is proper and consistent operation. Proper operation can be achieved through:

- Proper Grounding
- Suppression of electrical noise generated by the machine.
- Suppression of electrical noise coming in from the environment. (Other machines, power surges, etc.)

Grounding

Proper grounding is the single most important consideration for a safe installation. Proper grounding also ensures that unwanted electrical currents, such as those induced by electromagnetic noise, will be quickly shunted to ground instead of flowing through the machine.

Earth ground connections on electronic controls are one of two types.

- **Protective Earth Connection:** Point that must be tied to Earth for the safe operation of the device. (Protection against electric shock should the device be damaged.)

- **Functional Earth Connection:** Point that must be tied to Earth to improve noise immunity of the device.

For all AMCI Networked Drivers, the Protective Earth Connection is any mounting surface of the unit. (The SD31045E2 Drivers also have a grounding lug on their heatsinks.) The Functional Earth Connection of the Networked Drivers is the ground terminal (GND) on the power connector.

Grounding Electrode System

The Grounding Electrode System is the common name for the building's earth ground infrastructure. This system *defines* earth-ground potential within a building and is the central ground for all electrical equipment.

Ground Bus

AMCI strongly suggests the use of a ground bus in each system enclosure. The ground bus is simply a metal bar with studs or tapped holes for accepting grounding connections in the enclosure. The ground bus is the only component in the enclosure that is directly connected to your Grounding Electrode System. Therefore, the ground bus is becomes the central grounding point for the enclosure and its equipment.

Grounding (continued)

Grounding Electrode Conductor

Connection of the ground bus to the Grounding Electrode System is made with the Grounding Electrode Conductor. The Grounding Electrode Conductor should be the shortest length of stranded wire or braid possible. The American NEC allows a length of up to six feet, but shorter is better. A shorter length will help shunt high frequency noise to the Grounding Electrode System.

In extremely noisy environments, it is sometimes advisable to run two wires from the ground bus to the GES. The length of these two wires should differ by 20% to 30% so they offer different impedances to the noise frequencies being shunted to earth ground.

The minimum sizes for the Grounding Electrode Conductor are #8 AWG stranded wire or 1" braid, but larger sizes may be needed depending on the total ground fault current your system can generate. The actual size is determined by the size of the largest conductor in the system.

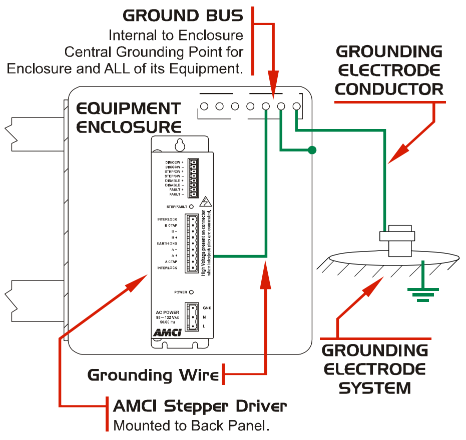

GROUND BUS
Internal to Enclosure
Central Grounding Point for Enclosure and ALL of its Equipment.
GROUNDING ELECTRODE CONDUCTOR
EQUIPMENT ENCLOSURE
GROUNDING ELECTRODE SYSTEM
Grounding Wire
AMCI Stepper Driver
Mounted to Back Panel.

Figure R8.1 Grounding Components

The American NEC allows the use of a solid wire for the Grounding Electrode Conductor but this should be avoided. High frequency currents (electrical noise) travel along the surface of a conductor. Heavy gauge stranded wire and braid both have large surface areas, solid conductors do not. Therefore, stranded wire and braid have a much lower effective resistance to electrical noise.

Grounding Wires

Grounding wires are used to connect the pieces of equipment to your ground bus. AMCI strongly suggests using #8 AWG stranded wire or 1" wire braid for all grounding connections. The American NEC allows the use of solid wire for grounding wires but this should be avoided for the same reasons they should not be used for the Grounding Electrode Conductor.

Avoiding Grounding Problems

As stated before, there are two reasons to properly ground a system. The first reason is to protect human life. A properly designed grounding system will be able to detect potential shock currents and remove power from the system. The second reason is to improve system reliability by shunting electrical noise currents to earth instead of allowing them to flow through the system where they can disrupt the operation of electronic controls.

The entire grounding system, even though conductive, is not meant to carry current except under fault conditions.

NOTE **Under normal operating conditions, the grounding system should not carry any current at all.**

Ground Fault Interrupt (GFI) circuits depend on this condition to work correctly. Any ground current flowing through a GFI is considered a fault condition and the GFI immediately opens to remove power from the faulted circuit.

By this definition, electrical noise and other transient currents such as inductive spikes are considered fault currents. The fact that these currents are very short lived and of relatively small power allows them to be safely shunted without tripping a Ground Fault Interrupt circuit.

Avoiding Grounding Problems (continued)

Grounding problems occur when the grounding system carries currents during normal operation or when it cannot shunt high frequency electrical currents to earth.

AMCI generally encounters three types of grounding problems in systems that use AMCI equipment.

- **Ground Loops:** A ground loop occurs when AC or DC return currents can travel through the system ground path in addition to their normal return path. This can cause damaging currents to electronics that share the power supply.

  Ground loops in AC systems can occur when the neutral conductor is bonded to the Grounding Electrode System in more than one place. Under the right conditions, AC line currents may end up flowing through the protective ground system, which may include exposed metal surfaces and therefore represent a touch shock hazard. Designing an effective ground fault detection system is well beyond the scope of this manual. Please refer to the NEC and other design guidelines for your area.

- **Ground Shift:** Remote machines or monitoring stations will usually be connected to a different point on the Grounding Electrode System than a local system. A voltage potential can exist between these stations due to resistance between the grounding electrodes and earth. A similar problem exists on machines that are not properly bonded together. The resistance of a poor bond in the system will result in a voltage potential across the connection when current is forced through it. Incorrectly installed sensor or communications cables that run between these systems can be damaged by these Ground Shift potentials.

- **High Impedance Grounds:** The grounding system shows a high impedance to high frequency noise currents and these signals are not properly shunted to earth. The high impedance to high frequency noise is caused by capacitance and inductance in the system. Use #8 AWG stranded wire or 1” wire braid for all ground connections and keep connections as short as possible to minimize capacitance and inductance in the grounding system.

Surge (EMI) Suppression

Incoming Power

In many systems, three phase power is brought to the machine and the control system is powered from one of these phases. In these cases, good quality, clean AC power can only be achieved by properly conditioning the three phase power, not just the single phase used by the control system. This is generally achieved by placing surge suppressors across each phase where the three phase enters the system and at all inductive loads that are on the three phase branch. This includes inductive loads *on any other machine* that may be powered by the same feeder circuit.

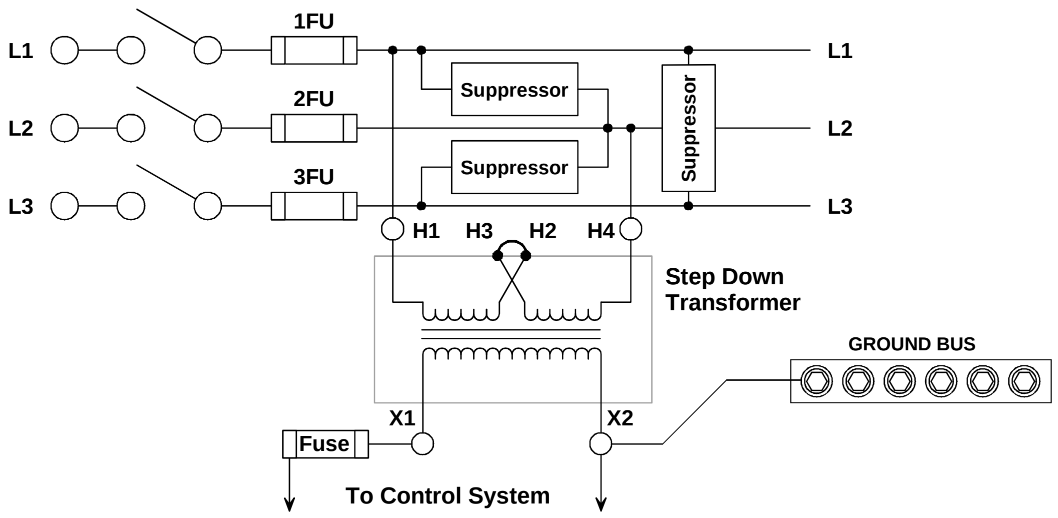


Figure R8.2 Surge Suppression on Power Inputs

Surge (EMI) Suppression (continued)

Inductive Loads

NOTE All inductive devices in the system, such as AC and DC motors, motor starters, contactors, relays and solenoids, must have surge suppression devices installed across their coils.

NOTE Because the Networked Driver is responsible for commutating the stepper motor, surge suppressors cannot be installed at the motor connector. The best way to avoid noise issues when using a stepper motor is proper cable installation. Keep the motor wires as short as possible. Consider twisting the winding pairs together to minimize radiated noise. Use shielded cable if you must extend the cable and tie the shield to earth ground at the Networked Driver motor connector only.

System Environment

When designing the physical layout of your system it is important to review the environment your control system will be placed in. This should include reviews of both the physical and electrical environments.

- Check the quality of the AC power that will feed the machine.
- Check for adequate surge suppression on machines that will be near your new system.

A review of the electrical environment includes checking the quality of the AC power that will feed your new system as well as checking the quality of the point where your system will be connected to the building's Grounding Electrode System.

When checking the quality of the AC power for your system, be sure to check every machine that may be powered by the same feeder circuit. It is in AMCI's experience that nuisance faults can be caused by another system on the same feeder circuit that does not have adequate surge suppression devices.

System Layout Considerations

The first step to achieving a clean system layout is classifying the wires and cables used in your system based on the amount of power they carry. Refer to the following table to help you categorize your wires and cables.

Category	Description	Examples
1	**Control and AC Power -** High power conductors that are more tolerant of electrical noise but are also more likely to carry electrical noise than the other categories. Corresponds to IEEE levels 3 & 4	AC power lines High power digital AC I/O lines High power digital DC I/O lines High power I/O lines typically connect to devices such as mechanical switches, relays, and solenoids. Stepper Motor wiring also falls into this category.
2	**Signal and Communication -** Low power conductors that are less tolerant of electrical noise but are also less likely to carry electrical noise than category 1. Corresponds to IEEE levels 1 & 2	Analog I/O lines Low power digital AC I/O lines Low power digital DC I/O lines Ethernet or other Communications Cables Low power I/O lines typically connect to devices such as proximity switches, photo sensors, and encoders.
3	**Intra-enclosure -** Similar characteristics to category 2 conductors, these conductors interconnect system components within an enclosure.	Low voltage DC power cables Communications Cables

Table R8.1 Conductor Categories

Wiring Categories for the SD17060E2 or SD31045E2

- AC Power Input: Class 1 (AC power line)
- DC Input cables (also included encoder cabling): Class 2 (Low power digital DC input line)
- DC output cable: Class 2 (Low Power digital DC output line)
- Motor Outputs: Class 1 (High Power DC outputs)

System Layout Considerations (continued)

Minimize Voltages in the System Enclosure

You will want to minimize the voltage in the enclosure that houses your control system to minimize EMI and voltage transients. In many cases, three phase is used to power the machine's motors and the control system is powered by one of the phases. Ideally, the three phase power, it's disconnect, fuses, filtering, and all three phase controls such as motor starters, are housed in their own enclosure. The single phase used to power the control system is then brought into a separate enclosure built exclusively for the control system.

Power Supply Sizing

A properly sized power supply is vital to system operation. The best guideline that we can give you is to buy the best supply your budget allows and consider the surge requirements of the components you are attaching to the supply.

Component Placement

Once you have established the proper categories of all of the system's wires and cables and determined all of your components you can determine proper component placement and conductor routing within the system enclosure.

CAUTION The following guidelines are for noise immunity only. Follow all local codes for safety requirements.

There are three general guidelines to follow when placing components:

- Keep as much physical space between the classes of conductors as possible.
- Minimize the distance that conductors run in parallel with each other to minimize capacitive coupling.
- If conductors must cross, they should do so at right angles to minimize inductive coupling.

If you have a PLC in your system, be sure to consider the spacing of the modules within the rack. Do not place an AC output module next to low level DC input module unless absolutely necessary. For example, consider placing low level input modules on the left side of the rack, high power AC output modules on the right side, and medium power I/O in the center. This should help you maximize the spacing of I/O wiring within the enclosure.

System Layout Considerations (continued)

Conduits to Enclosure

When designing the layout of you system be sure to include enough conduits to house the different categories of cabling. To guard against coupling noise from one conductor to another, follow the spacing guidelines in table R8.2 below. These spacing values should be followed for routing cables both inside and outside of an enclosure. Of course, sometimes these guidelines cannot be followed, such as when the connection points on a controller are spaced closer together than these guidelines recommend.

Category	Guidelines
1	These conductors can be routed in the same cable tray, raceway, or conduit as machine power conductors of up to 600Vac. Power conductors cannot feed larger than 100HP devices.
2	➤ If they must cross power feeds, they must do so at right angles. ➤ Route in a raceway or conduit separate from all category 1 conductors and properly shield where applicable. ➤ Any metal wireway or conduit that houses the conductor must be bonded along its entire length and bonded to the enclosure at its entry point. ➤ If in a continuous metal wireway or conduit, route at least 3" from category 1 conductors of less than 20A, 6" from category 1 conductors of greater than 20A and 12" from any category 1 conductor of any amperage of the circuit is greater than 100 kVA. ➤ If not in a continuous metal wireway or conduit, route at least 6" from category 1 conductors of less than 20A, 12" from category 1 conductors of greater than 20A and 24" from any category 1 conductor of any amperage of the circuit is greater than 100 kVA. ➤ Route at least 5 feet away from high-voltage enclosures or sources of rf/microwave radiation.
3	Route these conductors external to all raceways in the enclosure or in a raceway separate from all category 1 conductors. Use the same spacing rules given for category 2 conductors.

Table R8.2 Conductor Routing Guidelines

Installing an AMCI Networked Driver

The Networked Driver is designed to be mounted in an enclosure that is close to the motor. On small machines, this may be the same enclosure that the rest of the control system in mounted in. On large machines, this is typically a small enclosure that only houses the Networked Driver.

- A minimum of two conduits are required for the enclosure that houses a Networked Driver. One conduit is for the DC I/O and ethernet cable, and the other is for AC power and the motor connections. Three conduits may be required if the code in your area does not allow you to run the motor connections with AC power or if the noise generated on the motor connections is injected into your AC lines.
- If 230Vac is the only line voltage available in a system that is using the SD17060E2 Drivers, a step down transformer that is rated for a minimum of 800VA must be installed. Consider installing this transformer in the enclosure with the SD17060E2 Networked Driver. This will prevent any noise placed on the AC line by the SD17060E2 from leaving the enclosure.
- For the SD31045E2 Drivers, if 115Vac is the only line voltage available in the system and you require the 310Vdc motor bus, a step up transformer that is rated for a minimum of 800VA must be installed. Consider installing this transformer in the enclosure with the SD31045E2. This will prevent any noise placed on the AC line by the SD31045E2 from leaving the enclosure.
- Like all motors, stepper motors are noise generators and must be properly grounded when installed.
- Keep the motor wires as short as possible. Consider twisting the winding pairs together to minimize radiated noise. Use shielded cable if you must extend the cable and tie the shield to earth ground at the motor connector of the Networked Driver only.

TASK 1
INSTALLING AN AMCI NETWORKED DRIVER

This chapter gives detailed information on installing an AMCI Networked Driver. The person responsible for installing the driver should be familiar with the general installation guidelines given in the previous chapter.

1.1 Location

AMCI Networked Drivers are suitable for use in industrial environments that meet the following criteria:

- Only non-conductive pollutants normally exist in the environment, but an occasional temporary conductivity caused by condensation is expected.
- Transient voltages are controlled and do not exceed the impulse voltage capability of the product's insulation.

Note that these criteria apply to the system as a whole. These criteria are equivalent to the *Pollution Degree 2* and *Over Voltage Category II* designations of the International Electrotechnical Commission (IEC).

1.2 Safe Handling Guidelines

1.2.1 Prevent Electrostatic Damage

CAUTION Electrostatic discharge can damage the Networked Driver. Follow these guidelines when handling the unit.

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 bus connector or 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.

1.2.2 Prevent Debris From Entering the Unit

WARNING While mounting a devices, be sure that all debris (metal chips, wire strands, tapping liquids, etc.) is prevented from falling into the driver. Debris may cause damage to the unit or unintended machine operation with possible personal injury.

1.2.3 Remove Power Before Servicing

WARNING Remove power before removing or installing a Networked Driver.

1.3 Airflow and Wiring Space

To ensure adequate space for wiring and convectional cooling, you need:

- 2.0" (50 mm) of space above and below the driver
- 1.5" (37 mm) of space to the left and right of the driver
- 1.0" (25 mm) of space in front of the driver

NOTE Drivers should be mounted with the orientation shown in the outline drawings below. Holes are in the top and bottom of the driver cover for convectional cooling. If you have an active cooling system such as enclosure fans, the drivers should still be mounted as shown below, but you should be able to mount the drivers closer together.

1.4 Outline Drawings

1.4.1 SD17060E2 Outline Drawing

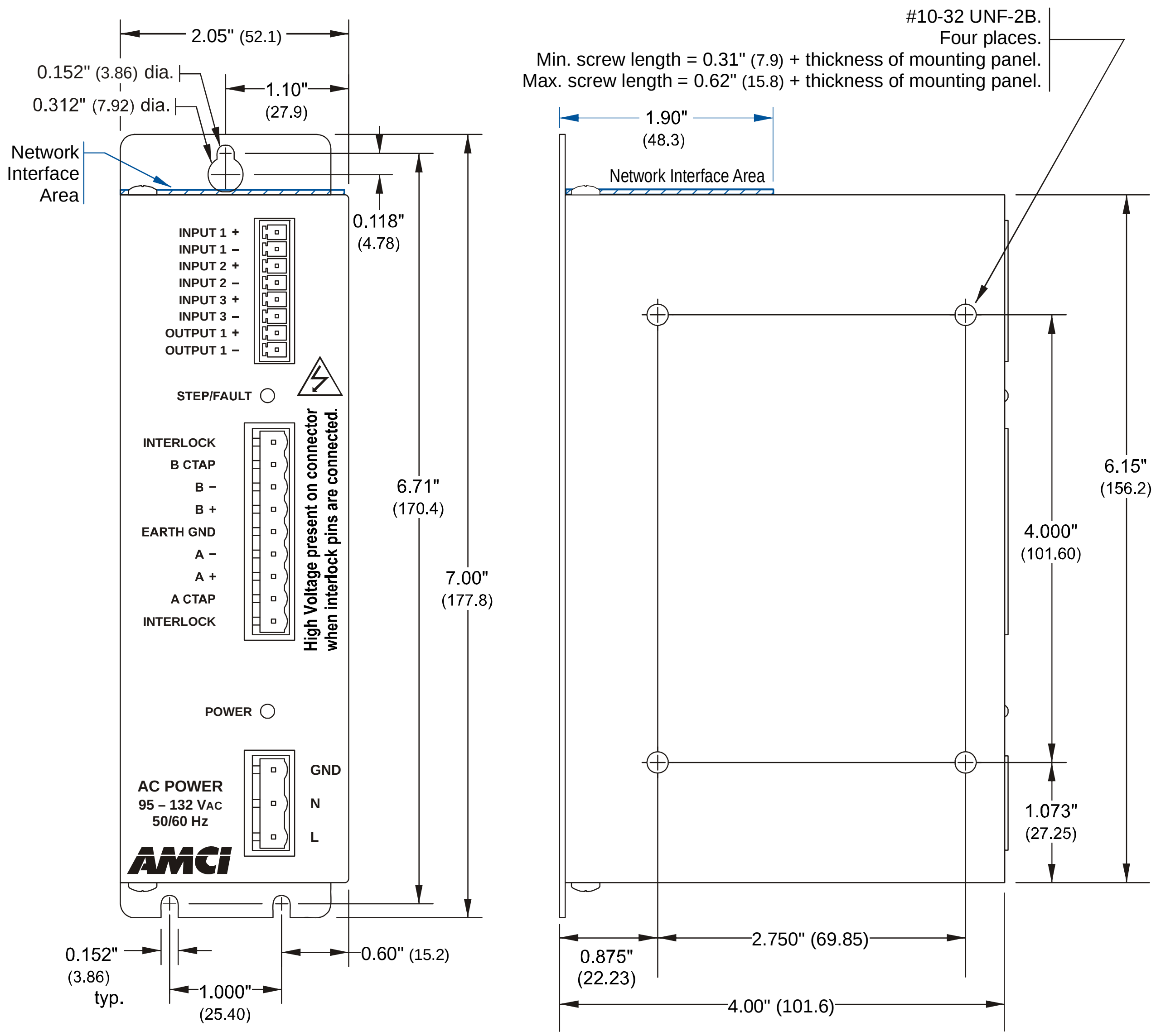


Figure T1.1 SD17060E2 Outline Drawing

1.4 Outline Drawings (continued)

1.4.2 SD31045E2 Outline Drawing

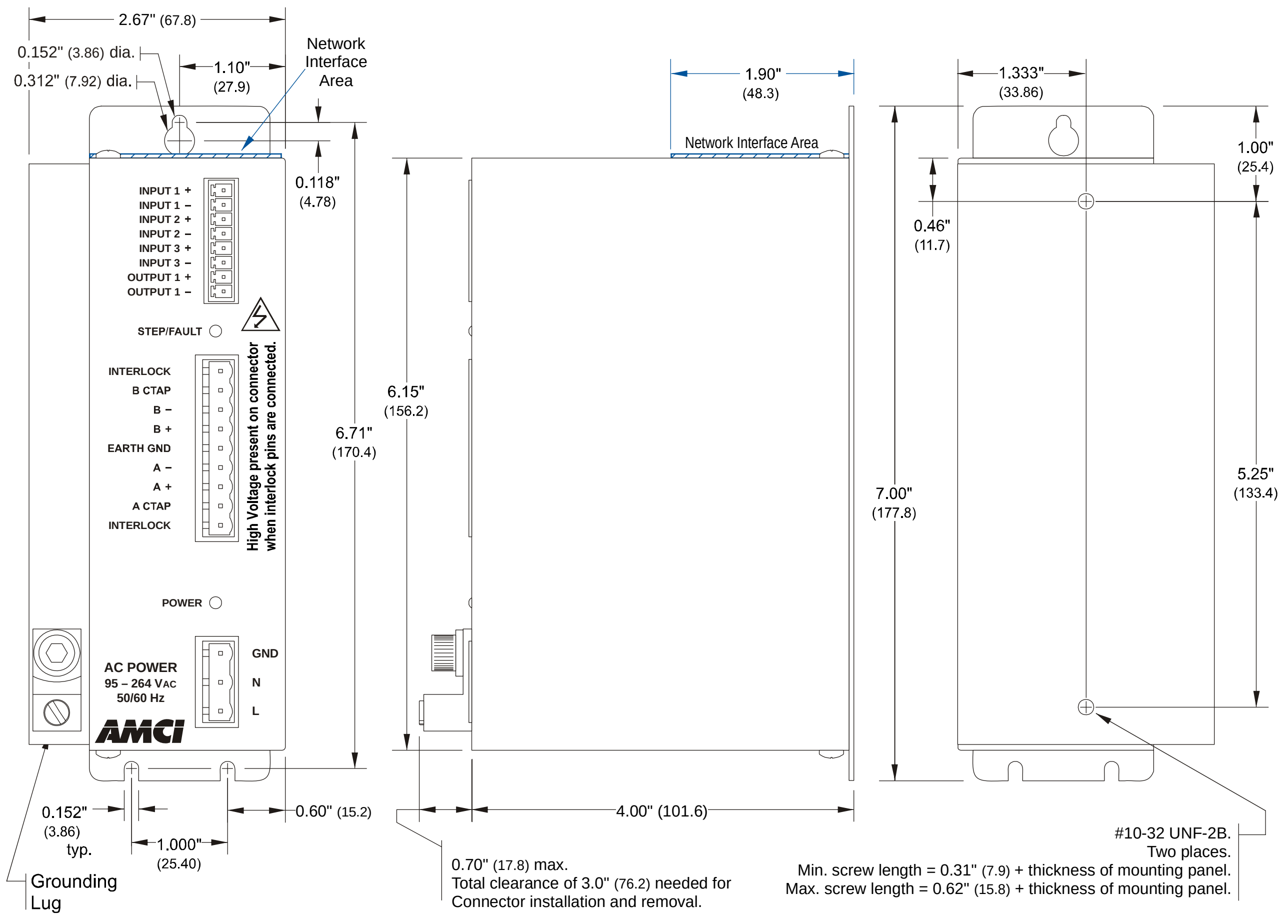


Figure T1.2 SD31045E2 Outline Drawing

1.5 Mounting Methods

There are two ways to mount an AMCI Networked Driver. The first method is with the #10-32 screws into the side panel of the SD17060E2 or the back panel of the SD31045E2. The second method is by the mounting tabs. Mounting tabs are for #6 screws.

WARNING When using the panel mounting method that uses the #10-32 screws, minimum and maximum screw lengths must be observed to prevent a screw from shorting to the PC board or components.

NOTE There are airflow holes in the top and bottom of the enclosure. To ensure adequate convectional airflow, the drive must be mounted in the orientation shown in the drawing.

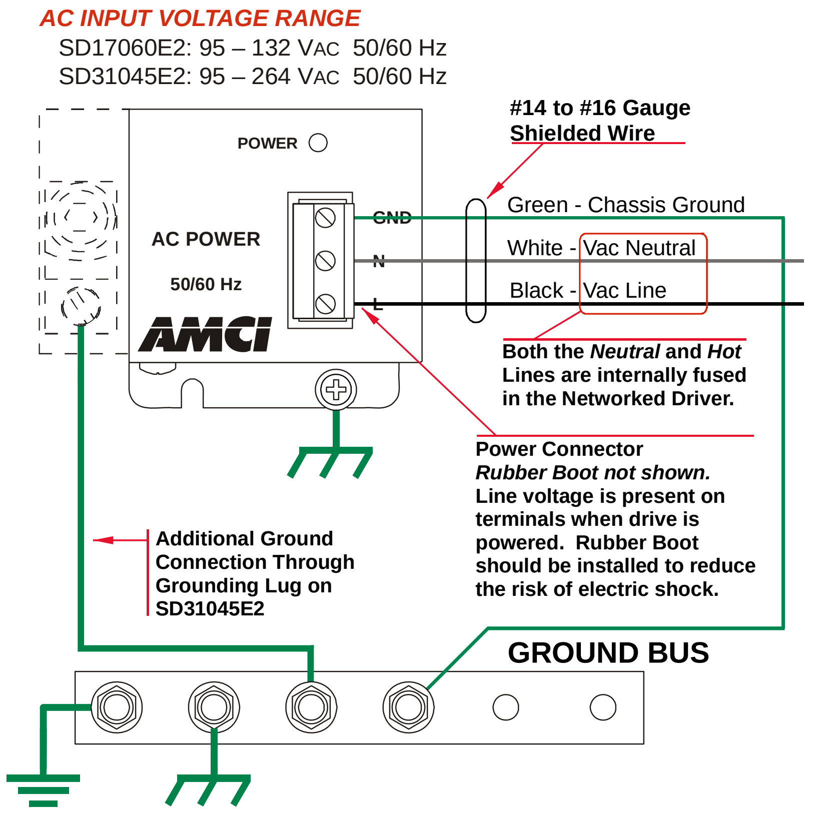
1.6 Grounding and Powering the System

WARNING The chassis must be connected to earth ground. Failure to properly ground the chassis leaves the potential for severe electrical hazard and/or problems with normal operation.

The Networked Driver must be grounded for proper operation. The **GND** connection on the power connector is connected to the enclosure of the Networked Driver and is a sufficient grounding point for most applications. When mounting the Networked Driver on a surface that is electrically conductive and grounded, you should take steps to ensure that the two are electrically bonded together. If necessary, remove paint from the surfaces of the panel that contact the mounting bolts to ensure adequate electrical bonding.

AC power connections are made to the Networked Driver using the power connector kit that ships with the driver. This kit includes the power connector and rubber boot. Figure T1.3 below shows how to properly wire and ground the driver.

NOTE For clarity, the rubber boot is not shown in the figure. When installing the power cable, slide the rubber boot onto the cable before wiring the connector. When you're sure the wiring is correct, slide the boot over the connector to cover the screw heads.



AC INPUT VOLTAGE RANGE
SD17060E2: 95 – 132 VAC 50/60 Hz
SD31045E2: 95 – 264 VAC 50/60 Hz
POWER
AC POWER
50/60 Hz
AMCI
GND
N
L
#14 to #16 Gauge Shielded Wire
Green - Chassis Ground
White - Vac Neutral
Black - Vac Line
Both the Neutral and Hot Lines are internally fused in the Networked Driver.
Power Connector
Rubber Boot not shown.
Line voltage is present on terminals when drive is powered. Rubber Boot should be installed to reduce the risk of electric shock.
Additional Ground Connection Through Grounding Lug on SD31045E2
GROUND BUS

Figure T1.3 Power and Grounding Connections

NOTE

1) Input power must be 95 to 132 Vac, 50/60 Hz for the SD17060E2 and 95 to 264 Vac, 50/60 Hz for the SD31045E2. Input power must able to supply 800VA for proper operation.
2) Never attempt to power an SD17060E2 with 230Vac. Doing so will damage the driver and void its warranty.

Both the Neutral and the Line power connections are internally fused in the AMCI Networked Drivers. These fuses are not field replaceable, so external fuses or circuit breakers should also be used. They must be rated for at least 10 amps.

1.7 Installing the Stepper Motor

1.7.1 Outline Drawings

Outline drawings for all of our motors can be found on our website, *www.amci.com*, in the *PDF Documents* section. They're 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.

1.7.2 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.

1.7.3 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.

1.7.4 Extending the Motor Cable

Even though it is possible to extend the cable length an additional forty feet, AMCI recommends installing the Networked Driver 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. Belden 9554 (eight wire), 9553 (six wire) and 9552 (four wire) meet these specifications for 18 AWG+. For long runs, consider using 14 AWG wire. Belden 1070A (eight wire), 1527A (six wire) and 1069A (four wire) meet these specifications for 14 AWG.

1.7.5 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 Networked Driver.
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.

1.8 Connecting the Motor

1.8.1 Motor Connector

The motor connector is shown in figure T1.4. The two Interlock terminals are a safety feature. When these two terminals are not connected, the driver will not power the motor outputs, the Status LED turns on red, and it activates the Fault Output. For normal operation, these two terminals must be connected together with a short wire.

The two center tap pins, A CTAP and B CTAP, are there for wiring convenience only. They are electrically isolated from the rest of the driver and are not used to power the motor. The **EARTH GND** pin is for the shields of the motor cable. This pin is directly connected to the GND pin of the power connector on the Networked Driver.

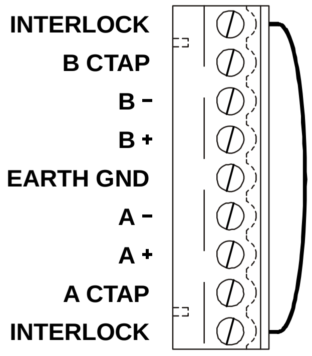


Figure T1.4 Motor Connector with Interlock Jumper

NOTE When powered, the motor connector represents a shock hazard because it has 170/310 Vdc present on its terminals. A rubber boot that is included with the connector must be installed but is not shown in the following figures for clarity. When installing the motor cable, slide the rubber boot onto the cable before wiring the connector. When you're sure the wiring is correct, slide the boot over the connector to cover the screw heads.

WARNING Always remove power from the Networked Driver 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 to the Earth Ground terminal of the Motor Connector on the Networked Driver.

1.8.2 Interlock Wiring

The motor interlock is designed to prevent 170/310 Vdc power from being supplied to the motor connector unless its mate is installed. As shown in figure T1.4 above, a short wire has to be installed between the two outer pins of the mating connector or the motor will not receive power when it is plugged into the Networked Driver.

NOTE It is possible to use the Interlock terminals as an alternative way of disabling motor current. This is accomplished by wiring the two pins through a *mechanical contact*. The wires to the Interlock pins should be kept as short as possible, which means using a mechanical relay in the cabinet that houses the Networked Driver if you need to control the Interlock from a source outside the cabinet. Only mechanical relays should be used. The voltage between the two interlock pins will never exceed 5 Vdc, but 115/230 Vac may be present between either of the two interlock pins and either of the power supply pins.

When the Networked Driver detects a break in the Interlock wiring, it will issue a *Driver Fault* error. This error is latched and can only be cleared after the interlock connection is re-established. The fault can be cleared over the network by following the procedure outlined in ***Notes on Clearing a Driver Fault*** which starts on page 89 or by cycling power to the driver.

1.8 Connecting the Motor (continued)

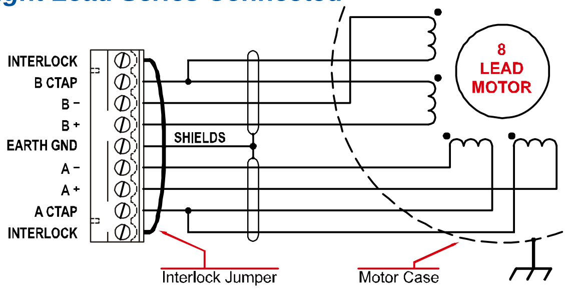
1.8.3 Motor Wiring

A Networked Driver 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.

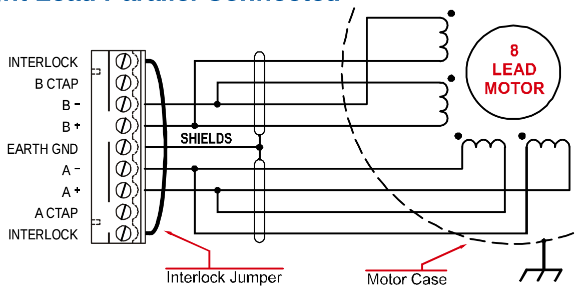
A wiring document for all of the motors ever sold by AMCI is available on our website. This single document contains all of the information necessary to connect any AMCI motor to any AMCI driver. At the time of this manual revision, the wiring manual can be found in the *PDF Documents* section of the website. It is under the *Stepper Motor* heading, and link is simply called "wiring".

Figure T1.5 shows how to wire a motor to a Networked Driver in series, parallel, or center-tap configurations.

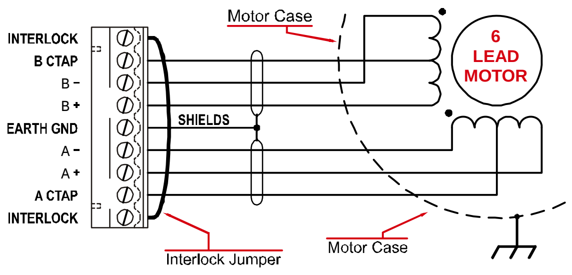
Eight Lead Series Connected



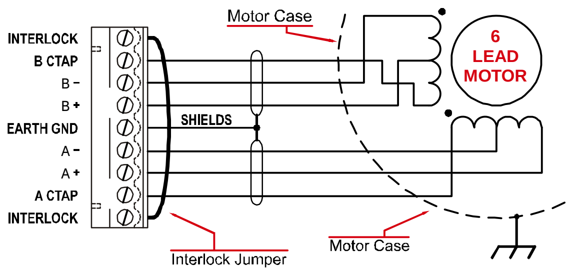
Eight Lead Parallel Connected



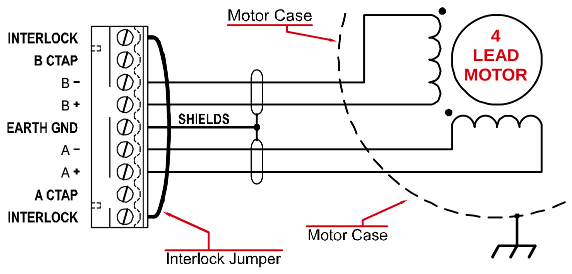
Six Lead Series Connected



Six Lead Center Tap Connected



Four Lead Connected



NOTES

1) Refer to the torque vs.speed curves on your motor's specifications sheet to determine if you should wire the motor to the Networked Driver in series, parallel, of center-tap configurations.
2) Motor connections should be tight. 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)

Figure T1.5 Motor Wiring

1.9 Digital Input Wiring

1.9.1 Cable Shields

Because they are low power signals, cabling from the sensor to the Networked Driver 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 Networked Driver.

1.9.2 Input Wiring

All three inputs are differential, and can be wired to sinking or sourcing sensors without requiring a pull up or pull down resistor. They accept 3.5 to 27 Vdc without the need for an external current limiting resistor. If using a 48 Vdc sensor, a 5 kilohm resistor that is wired in series with the input pin is required. Figure T1.7 below shows how to wire a discrete DC sourcing or sinking sensor to an input on the Networked Driver.

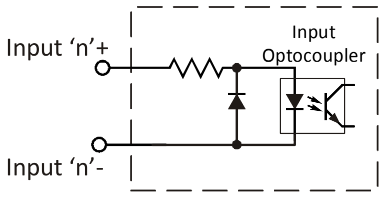


Figure T1.6 Input Schematic

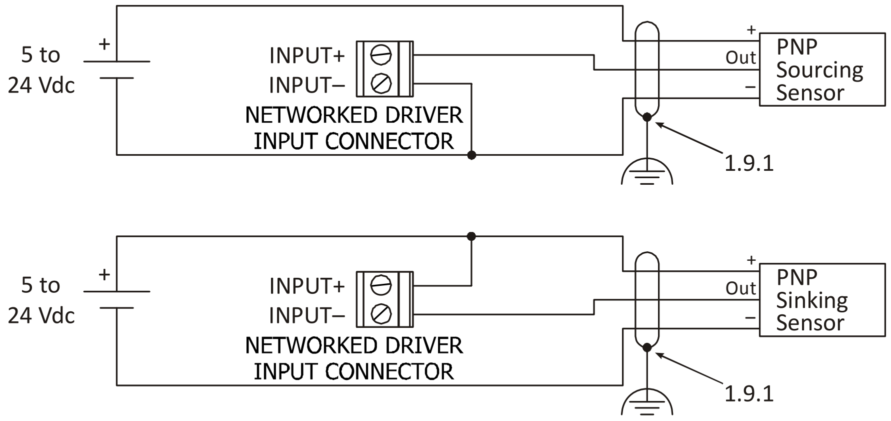


Figure T1.7 Wiring - Input 3

Encoders typically have differential outputs, but must be wired to the driver in a specific way. More information on wiring an encoder is given later in this chapter.

1.9 I/O Wiring (continued)

1.9.3 Output Wiring

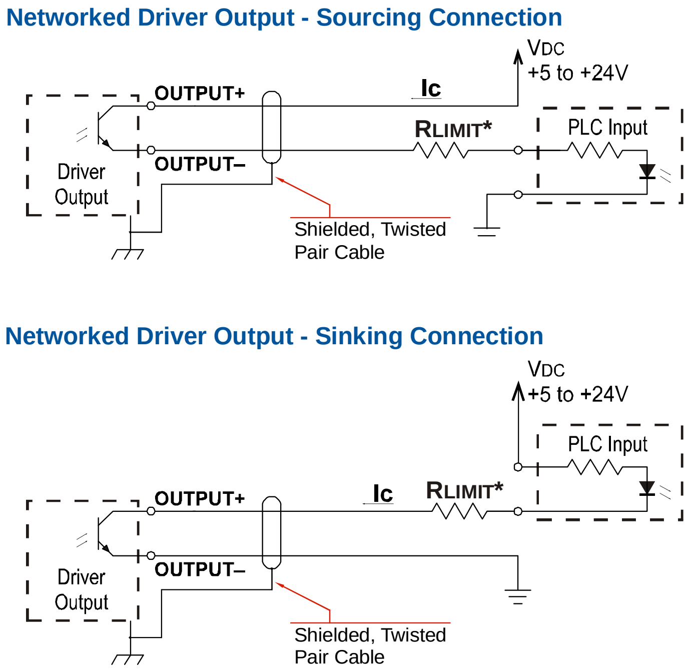

Networked Driver Output - Sourcing Connection
VDC
+5 to +24V
OUTPUT+
Ic
RLIMIT*
PLC Input
OUTPUT–
Driver Output
Shielded, Twisted Pair Cable
Networked Driver Output - Sinking Connection
VDC
+5 to +24V
PLC Input
OUTPUT+
Ic
RLIMIT*
OUTPUT–
Driver Output
Shielded, Twisted Pair Cable

Figure T1.8 Output Wiring

The output on the Networked Driver 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	$V_{CE\,SAT}$ = 1Vdc @ 20 mA
Ic max: 20 mA	Power Dissipation = 20 mW max.

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.

1.9.4 Encoder Wiring

The figure below shows how to wire a 5Vdc differential encoder to a Networked Driver.

- The ±A channel must be wired to Input 1 for proper operation
- The ±B channel must be wired to Input 2 for proper operation
- The ±Z channel is optional. If used, it must be wired to Input 3. The Z channel is only used when homing the Networked Driver.

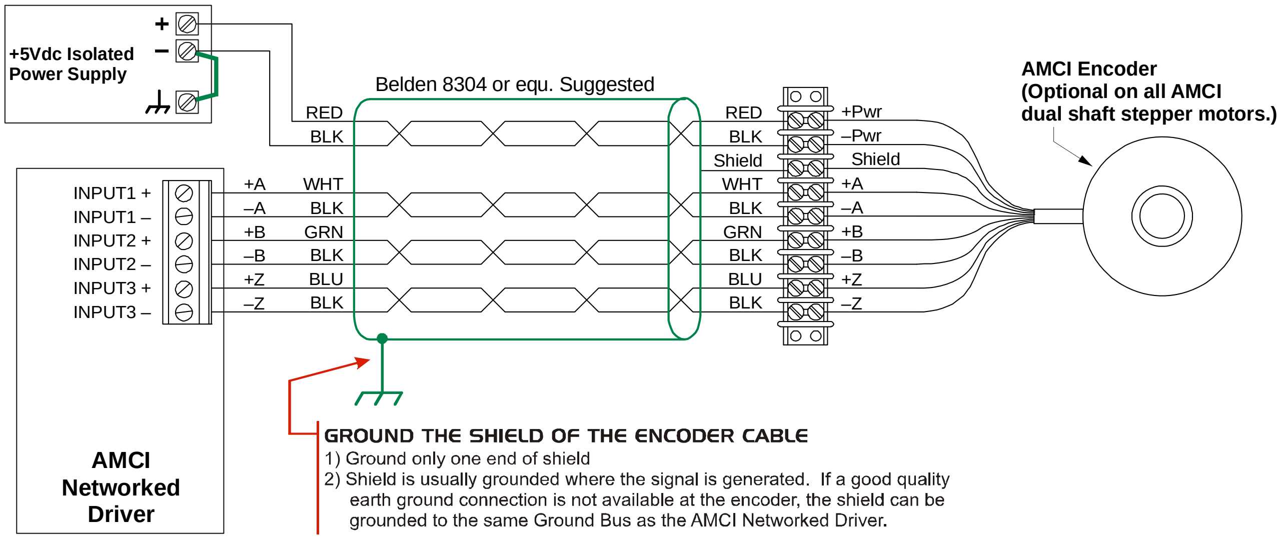

+5Vdc Isolated Power Supply
+
–
Belden 8304 or equ. Suggested
AMCI Encoder (Optional on all AMCI dual shaft stepper motors.)
RED
BLK
INPUT1 +
INPUT1 –
INPUT2 +
INPUT2 –
INPUT3 +
INPUT3 –
+A
–A
+B
–B
+Z
–Z
WHT
BLK
GRN
BLK
BLU
BLK
Shield
+Pwr
–Pwr
AMCI Networked Driver
GROUND THE SHIELD OF THE ENCODER CABLE
1) Ground only one end of shield
2) Shield is usually grounded where the signal is generated. If a good quality earth ground connection is not available at the encoder, the shield can be grounded to the same Ground Bus as the AMCI Networked Driver.

Figure T1.9 Sample Encoder Wiring

1.10 Ethernet Connections

The Ethernet connectors are located on the top of the SD17060E2 and SD31045E2 units. 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 SD17060E2 or SD31045E2 to any device, including a personal computer.

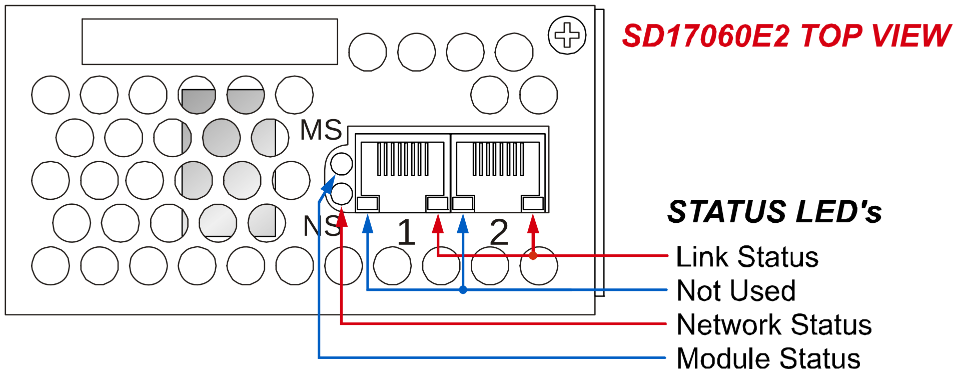


Figure T1.10 Ethernet Port Location

The Network Status LED’s are fully described on page 19 in the ***Status LED’s*** section of the Specifications reference.

1.11 EtherNet/IP Connections

1.11.1 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.

1.11.2 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.

1.12 PROFINET Connections

1.12.1 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.

1.12.2 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.

1.13 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 2
SET THE IP ADDRESS

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

2.1 Determine the Best Method for Setting the IP Address

There are three methods for setting the IP address on a Networked Driver. Table T2.1 below outlines the available methods and when you can use them.

Method	Restrictions	Section
Use Factory Default Settings	1) The machine must use 192.168.0.xxx subnet. 2) The 192.168.0.50 address must be available.	2.2a
Use the Embedded Web Server	No restrictions on use. This is the preferred method. The internal webserver can be used to set the Networked Driver to any IPv4 address. The IP address and protocol will be stored in nonvolatile memory and used on subsequent power-ups.	2.2b
Use the AMCI NET Configurator Utility	Unit must be initially configured for EtherNet/IP. The software can be used to set the Networked Driver to any IPv4 address and protocol. The IP address and protocol choice will be stored in non-volatile memory and used on subsequent power-ups.	2.2c

Table T2.1 Methods for Setting the IP Address

NOTE There is a MAC address label on each Networked Driver. This label 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 Networked Drivers 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.

2.2a Use Factory Default Settings

The factory default address for the Networked Driver 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 143.

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

2.2b Use the Embedded Web Server

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

PREREQUISITE: Task 1.6: ***Grounding and Powering the System*** found on page 100. You must be able to power the Networked Driver.

PREREQUISITE: Tasks: 1.10 and 1.11, 1.12, or 1.13, starting on page 106. You must attach your Networked Driver to your computer.

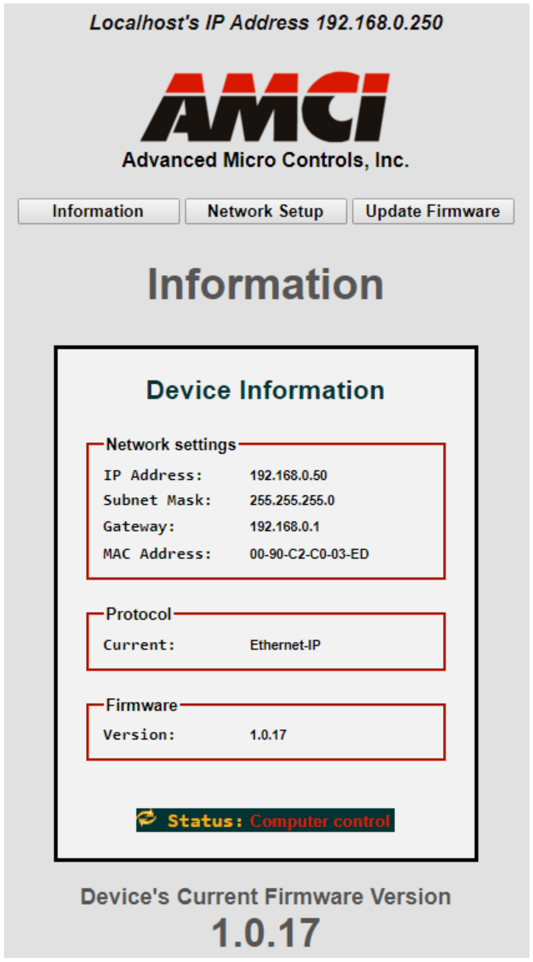
PREREQUISITE: Optional Task A: ***Configure Your Network Interfaces***. (page 143) The network interfaces on your computer must be on the same subnet before you can communicate with a Networked Driver.

2.2b.1 Disconnect the Networked Driver 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.

2.2b.2 Start your web browser and connect to the Networked Driver

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


Localhost's IP Address 192.168.0.250
Advanced Micro Controls, Inc.
Information
Network Setup
Update Firmware

Information

Device Information

Network settings

IP Address:	192.168.0.50
Subnet Mask:	255.255.255.0
Gateway:	192.168.0.1
MAC Address:	00-90-C2-C0-03-ED

Protocol

Current:	Ethernet-IP

Firmware

Version:	1.0.17

Status: Computer control

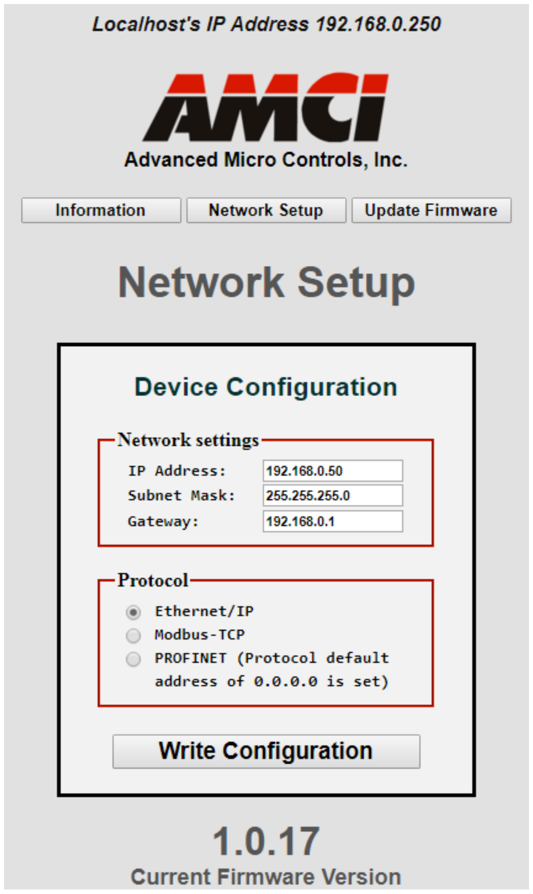
Device's Current Firmware Version
1.0.17

Figure T2.1 Networked Driver Information Webpage

2.2b Use the Embedded Web Server (continued)

2.2b.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 192.168.0.250
Advanced Micro Controls, Inc.
Information
Network Setup
Update Firmware
Write Configuration

Network Setup

Device Configuration

Network settings

IP Address:	192.168.0.50
Subnet Mask:	255.255.255.0
Gateway:	192.168.0.1

Protocol

- [✓ Ethernet/IP]
- Modbus-TCP
- PROFINET (Protocol default address of 0.0.0.0 is set)

1.0.17
Current Firmware Version

Figure T2.2 Networked Driver Network Setup Web Page

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

NOTE 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.

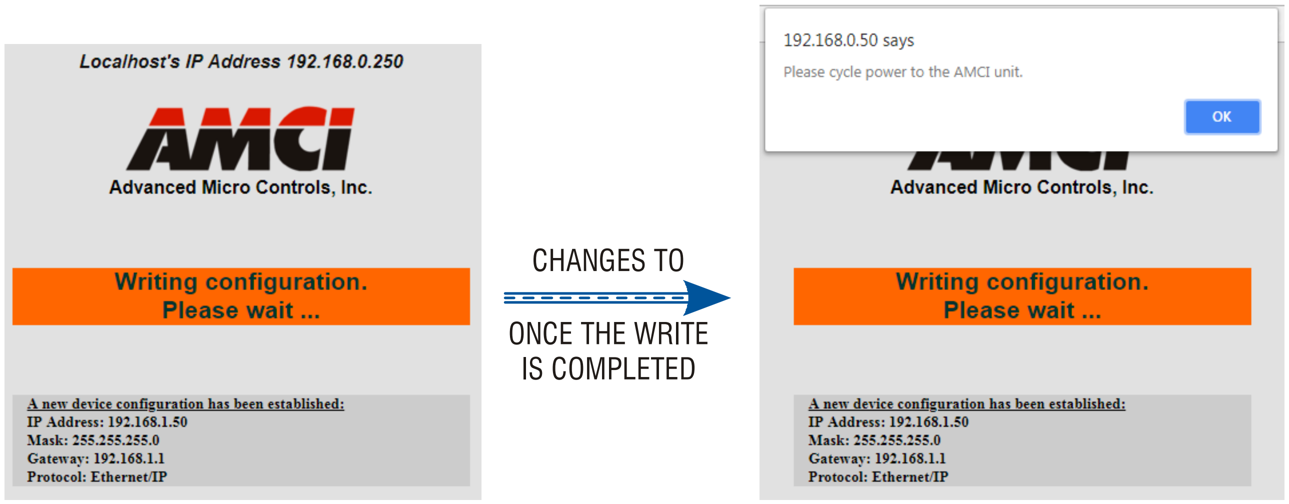
3) If need be, click on the proper radio button to select the required protocol.
4) Click on the [Write Configuration] button to write the new configuration to the unit. If there are any errors with the data, the unit will display a warning message instead of accepting the new values.

2.2b Use the Embedded Web Server (continued)

2.2b.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 Networked Driver. Cycling power before this window appears may corrupt the non-volatile memory of the unit. The Networked Driver will also flash its Network Status LED red to indicate that power must be cycled.


Localhost's IP Address 192.168.0.250

Advanced Micro Controls, Inc.

Writing configuration.
Please wait ...

A new device configuration has been established:
IP Address: 192.168.1.50
Mask: 255.255.255.0
Gateway: 192.168.1.1
Protocol: Ethernet/IP

192.168.0.50 says
Please cycle power to the AMCI unit.

Figure T2.3 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 Networked Driver.

Task Complete

2.2c 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 1.6: ***Grounding and Powering the System*** found on page 100. You must be able to power the Networked Driver.

PREREQUISITE: Tasks: 1.10 and 1.11, 1.12, or 1.13, starting on page 106. You must attach your Networked Driver to your computer.

PREREQUISITE: Optional Task A: ***Configure Your Network Interfaces***. (page 143) The network interfaces on your computer must be on the same subnet before you can communicate with the unit.

2.2c.1 Download the AMCI Net Configurator Utility

The AMCI Net Configurator utility is available on our website, ***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...”.

2.2c.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.

2.2c.3 Verify that Your Host Controller is Disconnected from the Networked Driver

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 Networked Driver, the AMCI Net Configurator utility must act as a bus master. Therefore, physically disconnect your host controller from the Networked Driver before starting the Net Configurator utility.

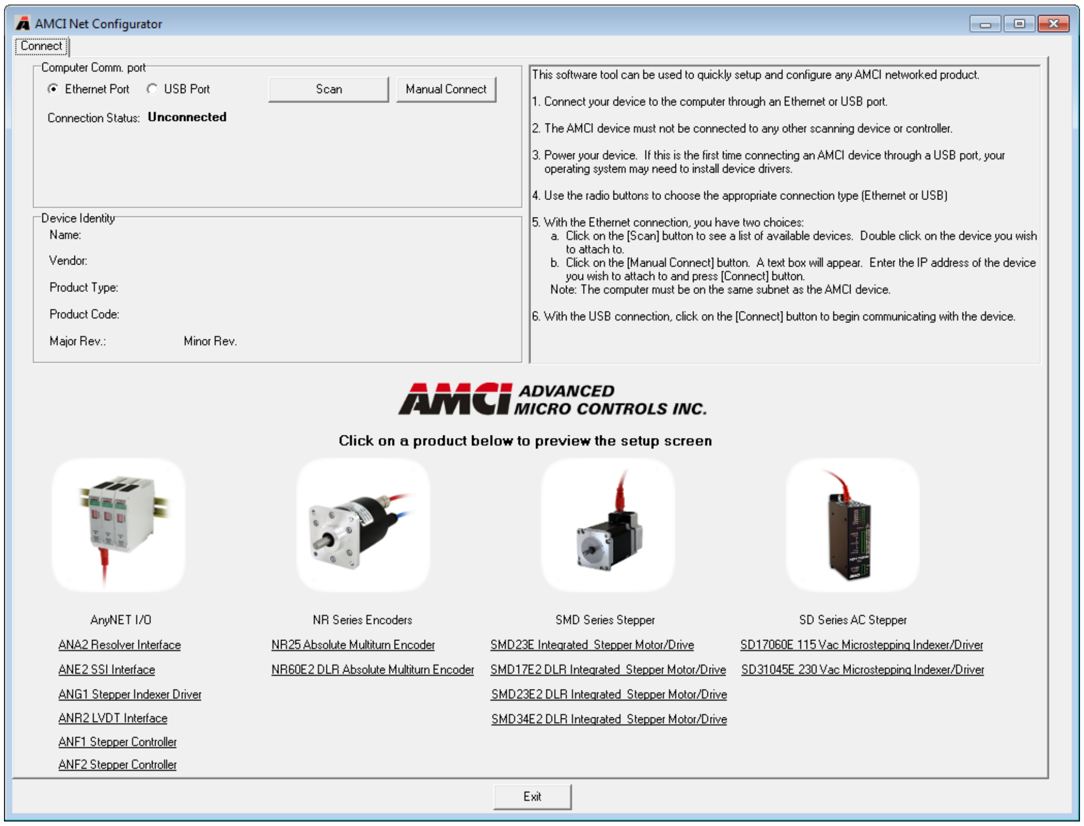
2.2c.4 Apply or Cycle Power to the Networked Driver

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

2.2c Use the AMCI Net Configurator Utility (continued)

2.2c.5 Start the AMCI Net Configurator Utility

Double click on the utility's icon. A welcome screen similar to the one in figure T2.4 below will appear.



Computer Comm. port: [✓ Ethernet Port] USB Port

Connection Status: **Unconnected**

This software tool can be used to quickly setup and configure any AMCI networked product.

1. Connect your device to the computer through an Ethernet or USB port.
2. The AMCI device must not be connected to any other scanning device or controller.
3. Power your device. If this is the first time connecting an AMCI device through a USB port, your operating system may need to install device drivers.
4. Use the radio buttons to choose the appropriate connection type (Ethernet or USB)
5. With the Ethernet connection, you have two choices:
   a. Click on the [Scan] button to see a list of available devices. Double click on the device you wish to attach to.
   b. Click on the [Manual Connect] button. A text box will appear. Enter the IP address of the device you wish to attach to and press [Connect] button.
   Note: The computer must be on the same subnet as the AMCI device.
6. With the USB connection, click on the [Connect] button to begin communicating with the device.

Click on a product below to preview the setup screen

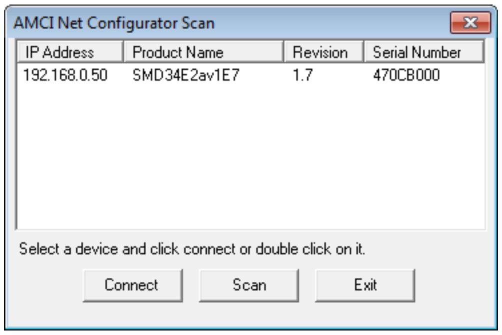
AnyNET I/O	NR Series Encoders	SMD Series Stepper	SD Series AC Stepper
ANA2 Resolver Interface	NR25 Absolute Multiturn Encoder	SMD23E Integrated Stepper Motor/Drive	SD17060E 115 Vac Microstepping Indexer/Driver
ANE2 SSI Interface	NR60E2 DLR Absolute Multiturn Encoder	SMD17E2 DLR Integrated Stepper Motor/Drive	SD31045E 230 Vac Microstepping Indexer/Driver
ANG1 Stepper Indexer Driver		SMD23E2 DLR Integrated Stepper Motor/Drive	
ANR2 LVDT Interface		SMD34E2 DLR Integrated Stepper Motor/Drive	
ANF1 Stepper Controller			
ANF2 Stepper Controller			

Figure T2.4 Net Configurator Welcome Screen

2.2c Use the AMCI Net Configurator Utility (continued)

2.2c.6 Press the [SCAN] button and Connect to the Networked Driver

Pressing the [Scan] button will open the window shown in figure T2.5. The Networked Driver 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.



IP Address	Product Name	Revision	Serial Number
192.168.0.50	SMD34E2av1E7	1.7	470CB000

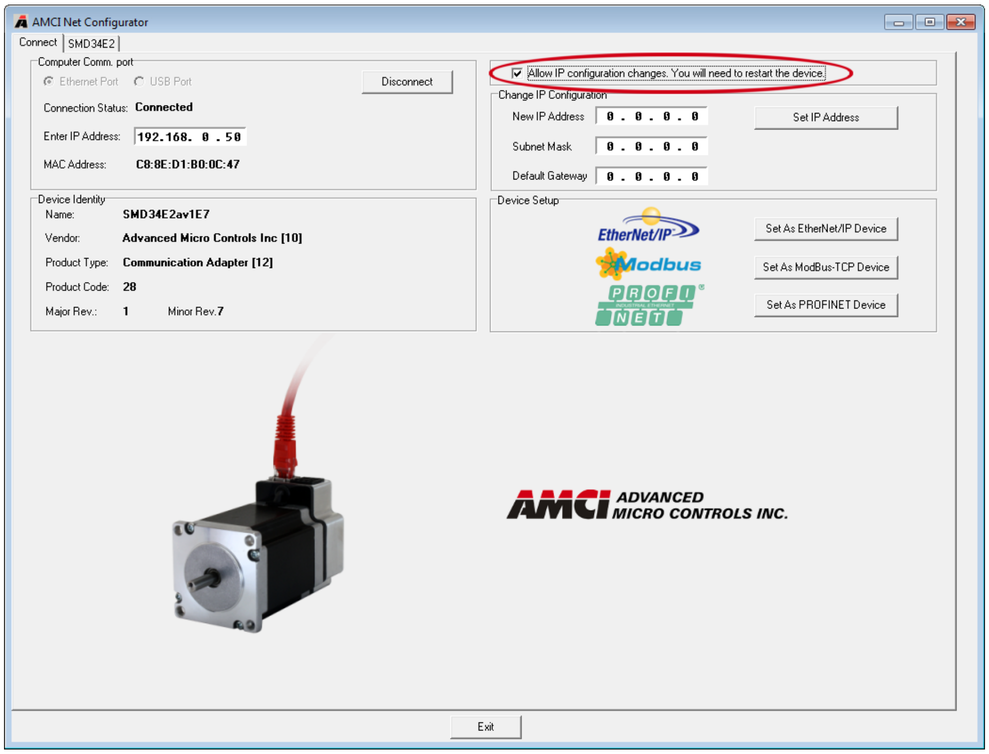
Select a device and click connect or double click on it.

Figure T2.5 Scan for SMD34E2

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

2.2c.7 Click on the "Allow IP..." Checkbox to Access the IP Settings

Figure T2.6 below shows the screen that results when you are connected to the Networked Driver. 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.



Connection Status: Connected
Enter IP Address: 192.168. 0 . 50
MAC Address: C8:8E:D1:B0:0C:47

[✓] Allow IP configuration changes. You will need to restart the device.

Change IP Configuration
New IP Address 0 . 0 . 0 . 0
Subnet Mask 0 . 0 . 0 . 0
Default Gateway 0 . 0 . 0 . 0

Device Identity
Name: SMD34E2av1E7
Vendor: Advanced Micro Controls Inc [10]
Product Type: Communication Adapter [12]
Product Code: 28
Major Rev.: 1 Minor Rev.7

Figure T2.6 Enable IP Address Changes

2.2c Use the AMCI Net Configurator Utility (continued)

2.2c.8 Set the IP Address, Subnet Mask, and Default Gateway

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

NOTE 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.

2.2c.9 Set the Communications Protocol

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

2.2c.10 Write the New IP Address to the Networked Driver

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.

2.2c.11 Remove Power from the Networked Driver

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

Task Complete

USING THE EDS FILE WITH ETHERNET/IP

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 SD17060E2 or SD31045E2 can always be added to a system as a generic module. If you are using a Rockwell Automation PLC, adding the unit as a generic module is the only option available if you are using RSLogix 5000 version 19 and below, or RSLogix 500.

Problems On Some Systems

AMCI is aware of some non-Rockwell Automation systems that have a communications error with the Ethernet Driver when using the EDS file and implicit messaging. The problem stems from the system not allowing the Ethernet Driver to choose the UDP socket number when using point to point communications. If you are using the EDS file and are experiencing communications errors, contact AMCI for a utility that will change the UDP socket used by the Networked Driver.

3.1 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. The ZIP file contains the EDS text file and a custom icon file for the device.

3.2 Install the EDS file

3.2.1 Start the EDS Hardware Installation Tool

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

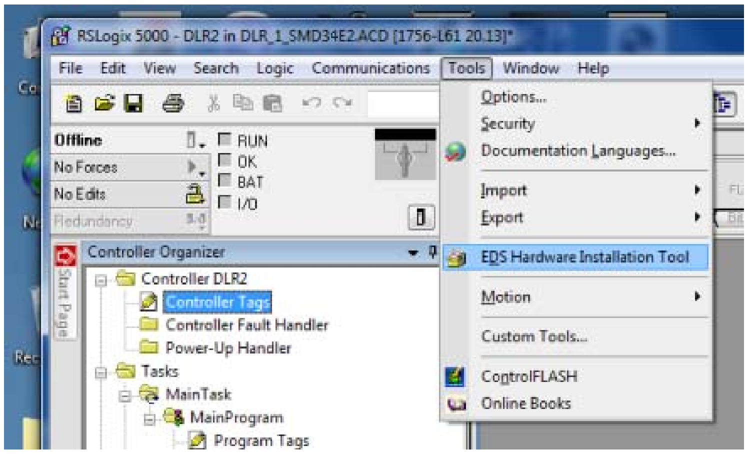


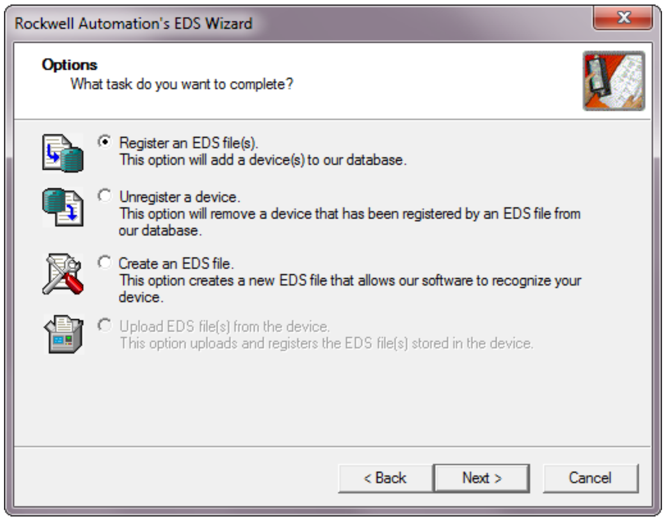
Figure T3.1 Opening the EDS Wizard

2) This will open the EDS Wizard. Click on [Next >] to advance to the Options screen.

3.2 Install the EDS file (continued)

3.2.2 Install the EDS File

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


Rockwell Automation's EDS Wizard

Options
What task do you want to complete?

[✓ Register an EDS file(s).] This option will add a device(s) to our database.

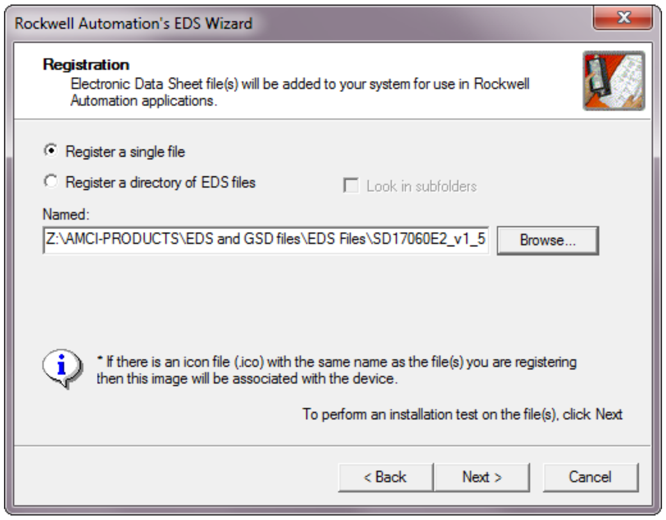
Unregister a device. This option will remove a device that has been registered by an EDS file from our database.

Create an EDS file. This option creates a new EDS file that allows our software to recognize your device.

Upload EDS file(s) from the device. This option uploads and registers the EDS file(s) stored in the device.

Figure T3.2 EDS Options Screen

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


Rockwell Automation's EDS Wizard

Registration
Electronic Data Sheet file(s) will be added to your system for use in Rockwell Automation applications.

[✓ Register a single file]
Register a directory of EDS files
Look in subfolders

Named: Z:\AMCI-PRODUCTS\EDS and GSD files\EDS Files\SD17060E2_v1_5

* If there is an icon file (.ico) with the same name as the file(s) you are registering then this image will be associated with the device.

To perform an installation test on the file(s), click Next

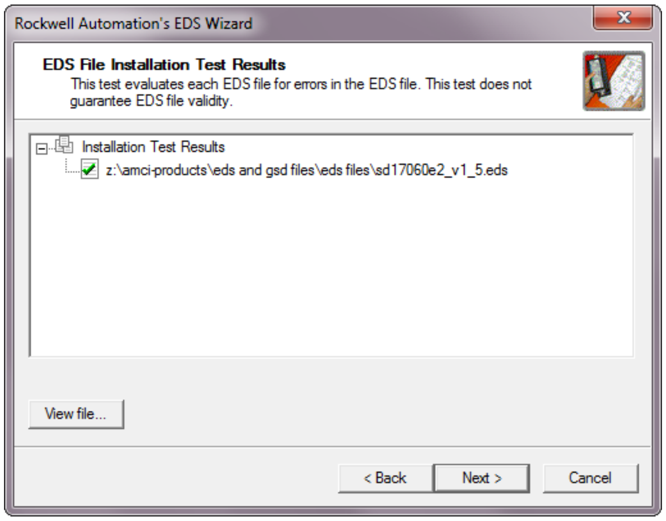
Figure T3.3 EDS Registration Screen

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.

3.2 Install the EDS file (continued)

3.2.2 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.

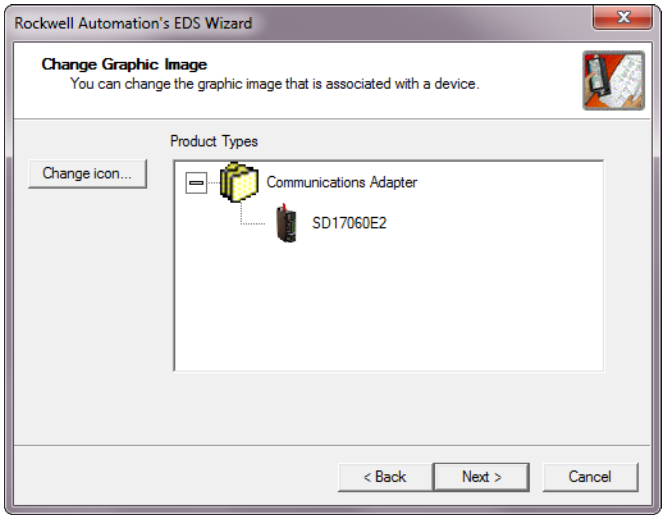

Rockwell Automation's EDS Wizard

EDS File Installation Test Results
This test evaluates each EDS file for errors in the EDS file. This test does not guarantee EDS file validity.

Installation Test Results
✓ z:\amci-products\eds and gsd files\eds files\sd17060e2_v1_5.eds

Figure T3.4 EDS Test Screen

5) Press on the [Next >] button to advance to the Change Graphic Image screen. The Change Graphic Image screen gives you the ability to change the icon associated with the device.


Rockwell Automation's EDS Wizard

Change Graphic Image
You can change the graphic image that is associated with a device.

Product Types
Communications Adapter
SD17060E2

Figure T3.5 Change ECS Icon Screen

6) Click on the [Change icon...] button. In the window that opens, click on [Browse...] and browse to the folder that contains the extracted EDS and icon files you downloaded from the AMCI website.
7) Select the icon file (*.ico) associated with the device. Click on the [Open] button and then on [OK] to return to the Change Graphic Image screen.
8) Click on the [Next...] button to advance to the completion screen. The Completion screen tells you that you have successfully completed the wizard. Click on the [Finish] button to exit the EDS wizard.
9) Click on the [Finish] button to exit the EDS wizard.

3.3 Configure the Ethernet Adapter on Your PLC

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 the AMCI Ethernet Driver 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.

3.4 Add the SD17060E2 or SD31045E2 to Your Project

You can add the AMCI Networked Driver to the project once the Ethernet port (built-in or bridge module) is configured. As shown in figure T3.6 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.

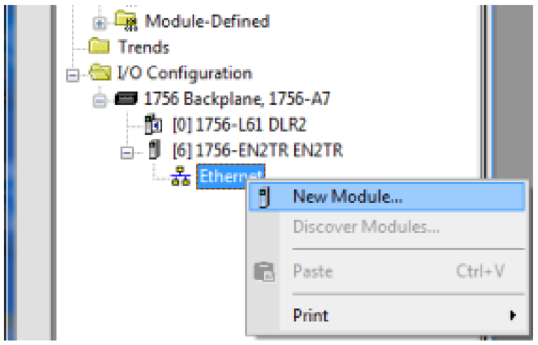
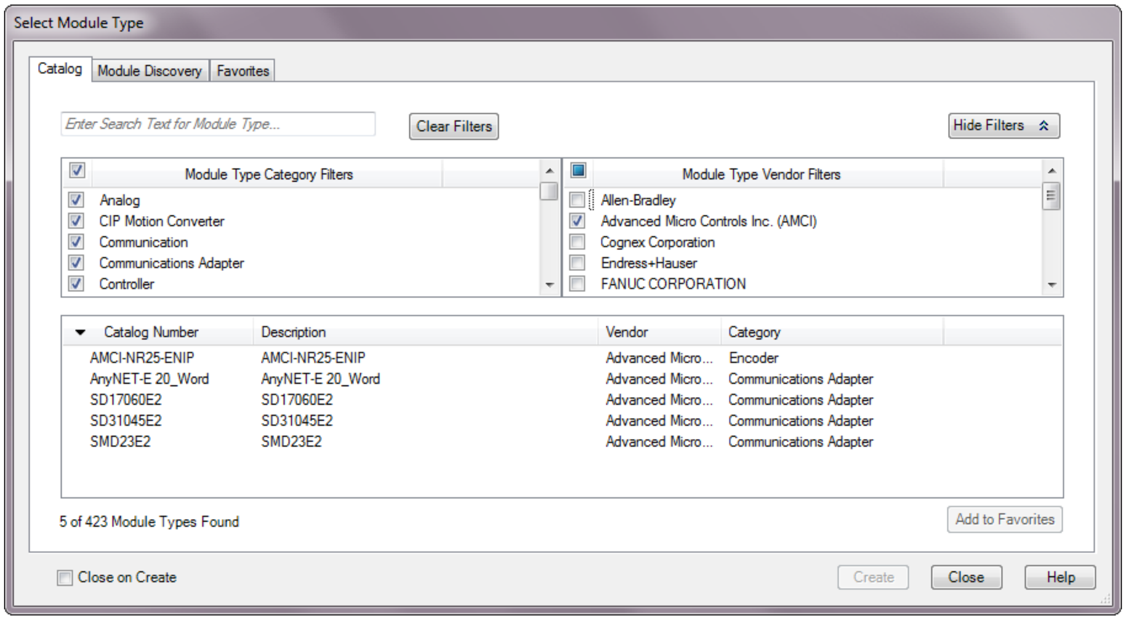


Figure T3.6 Adding an AMCI Ethernet Driver

2) In the resulting Select Module Type screen, select "Advanced Micro Controls In. (AMCI)" in the Vendor Filters. This will limit the results to catalog numbers from AMCI.
3) Select "SD17060E2" or "SD31045E2" 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.



✓	Module Type Category Filters
✓	Analog
✓	CIP Motion Converter
✓	Communication
✓	Communications Adapter
✓	Controller

	Module Type Vendor Filters
☐	Allen-Bradley
✓	Advanced Micro Controls Inc. (AMCI)
☐	Cognex Corporation
☐	Endress+Hauser
☐	FANUC CORPORATION

Catalog Number	Description	Vendor	Category
AMCI-NR25-ENIP	AMCI-NR25-ENIP	Advanced Micro...	Encoder
AnyNET-E 20_Word	AnyNET-E 20_Word	Advanced Micro...	Communications Adapter
SD17060E2	SD17060E2	Advanced Micro...	Communications Adapter
SD31045E2	SD31045E2	Advanced Micro...	Communications Adapter
SMD23E2	SMD23E2	Advanced Micro...	Communications Adapter

5 of 423 Module Types Found

Figure T3.7 Selecting the Networked Driver

3.5 Configure the SD17060E2 or SD31045E2 Driver

If you are continuing from step 3.4, the resulting New Module screen is used to configure the network connection between the Networked Driver and your controller. If you need to open the screen to perform this task at a later time, right click on the SD17060E2 / SD31045E2 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.

3.5.1 General Tab

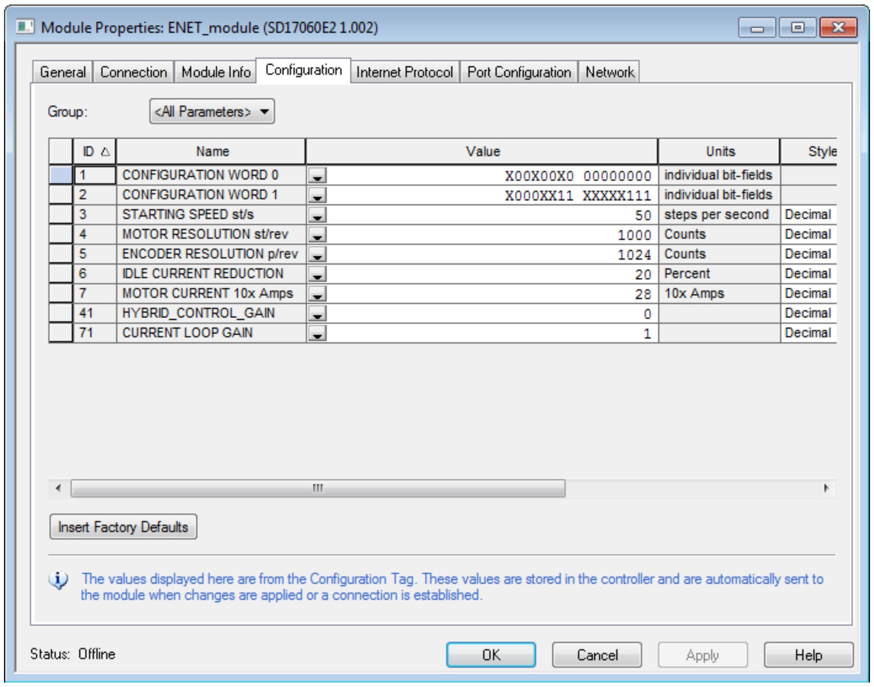
The Name, Description, and IP address of the device must be specified here. The [Change...] button allows you to change the Module Definition if needed.

3.5.2 Connection Tab

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

3.5.3 Configuration Tab

The Configuration tab is used to define the configuration data that is written down to the Networked Driver when the device connects to the network.



Group: <All Parameters>

ID	Name	Value	Units	Style
1	CONFIGURATION WORD 0	X00X00X0 00000000	individual bit-fields	
2	CONFIGURATION WORD 1	X000XX11 XXXXX111	individual bit-fields	
3	STARTING SPEED st/s	50	steps per second	Decimal
4	MOTOR RESOLUTION st/rev	1000	Counts	Decimal
5	ENCODER RESOLUTION p/rev	1024	Counts	Decimal
6	IDLE CURRENT REDUCTION	20	Percent	Decimal
7	MOTOR CURRENT 10x Amps	28	10x Amps	Decimal
41	HYBRID_CONTROL_GAIN	0		Decimal
71	CURRENT LOOP GAIN	1		Decimal

The values displayed here are from the Configuration Tag. These values are stored in the controller and are automatically sent to the module when changes are applied or a connection is established.

Status: Offline

Figure T3.8 Networked Driver Configuration with EDS File

The EDS file defines tags that are used to configure the SD17060E2 or SD31045E2. The two drivers use the same tags. These tags follow the format of the Configuration Data given in reference chapter 6, ***Configuration Mode Data Format***, starting on page 61.

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

NOTE Bits 8 and 9 of Configuration Word 1, *Binary_Output_Format* and *Binary_Input_Format*, should both be set to “1” when using and EDS setup so that command and response data is sent as 32-bit binary values.

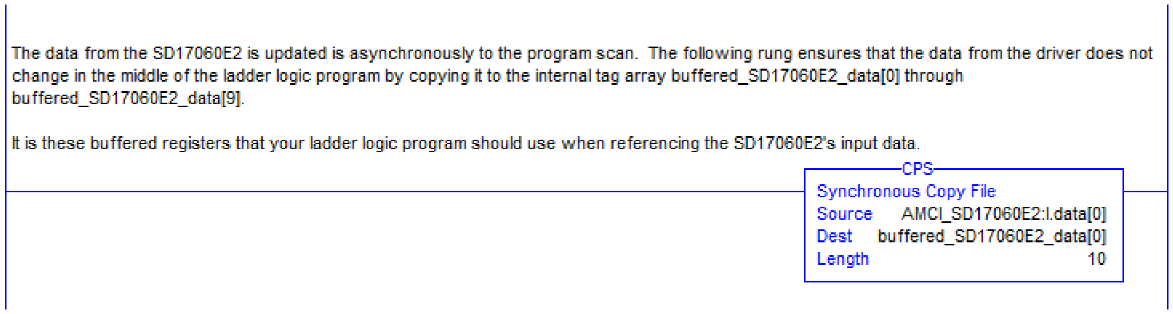
3.5 Configure the SD17060E2 or SD31045E2 Driver (continued)

3.5.3 Configuration Tab (continued)

NOTE Using the [Apply] button to write configuration data to the Networked Driver will disable the motor current. A 0→1 transition on the *Enable_Driver* bit is required to re-enable the motor. Note that you must wait a minimum of 300 milliseconds before re-enabling the motor. See ***Command Word 1*** on page 71 for an explanation of the *Enable_Driver* bit.

3.6 Buffering the I/O Data

Input and output data is transferred asynchronously to the program scan. 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.


The data from the SD17060E2 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_SD17060E2_data[0] through buffered_SD17060E2_data[9].

It is these buffered registers that your ladder logic program should use when referencing the SD17060E2's input data.

```
CPS
Synchronous Copy File
Source    AMCI_SD17060E2:I.data[0]
Dest      buffered_SD17060E2_data[0]
Length    10
```

Figure T3.9 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.

The format of the input and output data is covered in the ***Configuration Mode Data Format*** and ***Command Mode Data Format*** reference chapters, starting on pages 61 and 69 respectively.

TASK 4 (EtherNet/IP Option)

IMPLICIT COMMUNICATIONS WITHOUT EDS

Each AMCI Networked Stepper Driver 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 20 software is used for the example installation in this chapter.

4.1 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 SD17060E2 or SD31045E2 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.

4.2 Adding the SD17060E2 or SD31045E2

You can add the Networked Driver 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.

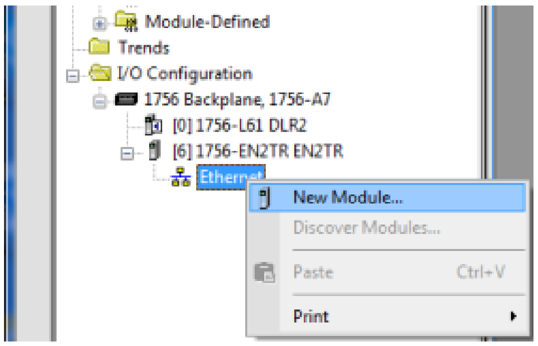
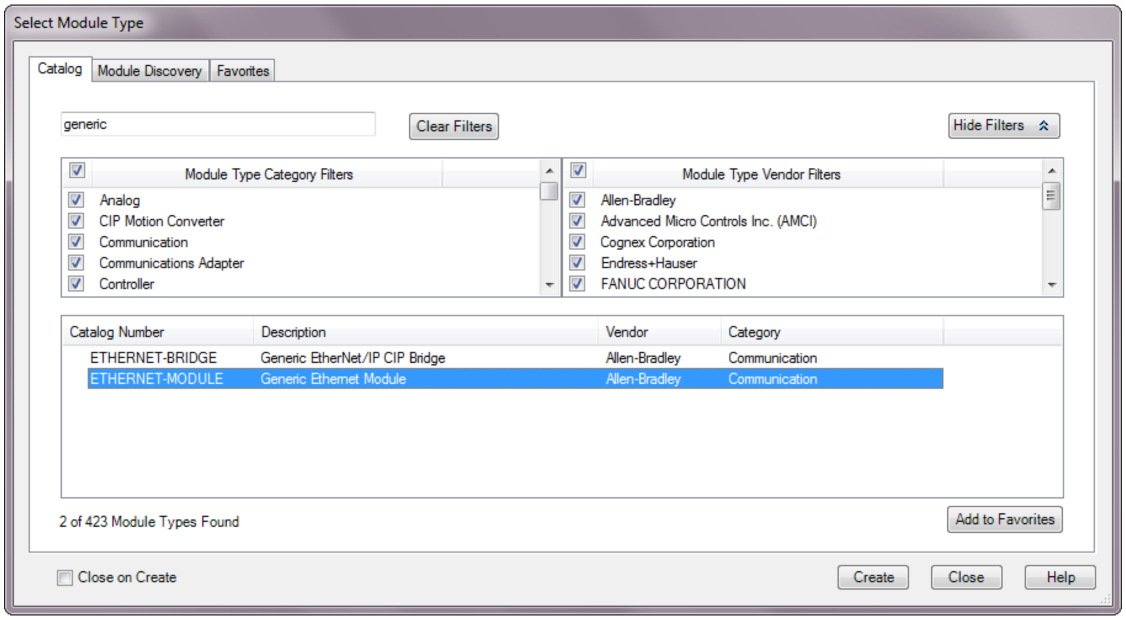


Figure T4.1 Adding an AMCI Networked Driver

4.2 Adding the SD17060E2 or SD31045E2 (continued)

2) In the resulting Select Module Type screen, type “generic” into the filter as shown in figure T4.2. 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.



generic

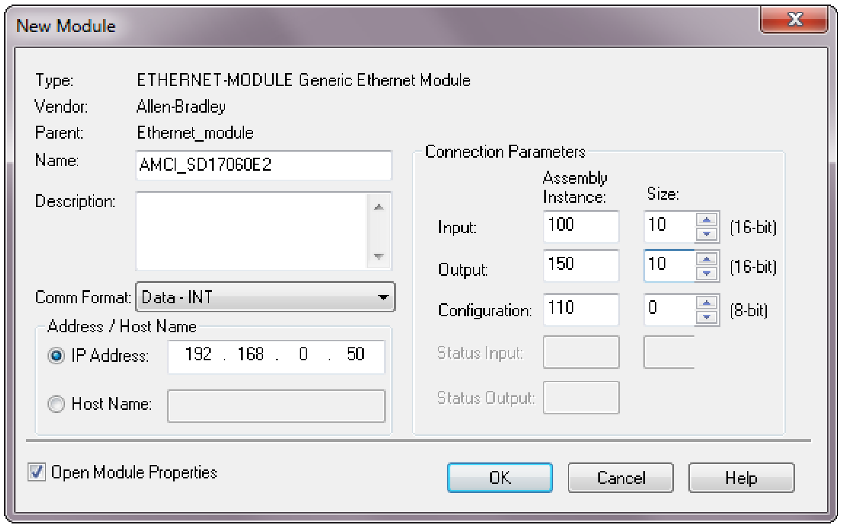
Catalog Number	Description	Vendor	Category
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley	Communication
ETHERNET-MODULE	Generic Ethernet Module	Allen-Bradley	Communication

2 of 423 Module Types Found

Figure T4.2 Selecting the Networked Driver

4.2 Adding the SD17060E2 or SD31045E2 (continued)

6) Set the following parameters in the Module Properties window. All parameters not listed here are optional. Figure T4.3 shows a completed screen.


New Module

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: Ethernet_module
Name: AMCI_SD17060E2
Description:
Comm Format: Data - INT
Address / Host Name
IP Address: 192 . 168 . 0 . 50
Host Name:

Connection Parameters

	Assembly Instance:	Size:	
Input:	100	10	(16-bit)
Output:	150	10	(16-bit)
Configuration:	110	0	(8-bit)
Status Input:			
Status Output:			

Open Module Properties [✓]

Figure T4.3 Configuration Screen - Generic Device

➤ **Name:** A descriptive name for the driver.

➤ **Comm Format:** Data - INT

CAUTION The Comm Format defaults to Data - DINT. The Networked Driver 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 added to the project if the Comm Format is incorrect.

➤ **IP Address:** Must be the address you set for the SD17060E2 or SD31045E2. Refer to the ***Set the IP Address*** task chapter starting on page 109 for information on setting the IP Address of the driver.

➤ **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 a Networked Drive 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 Networked Driver.

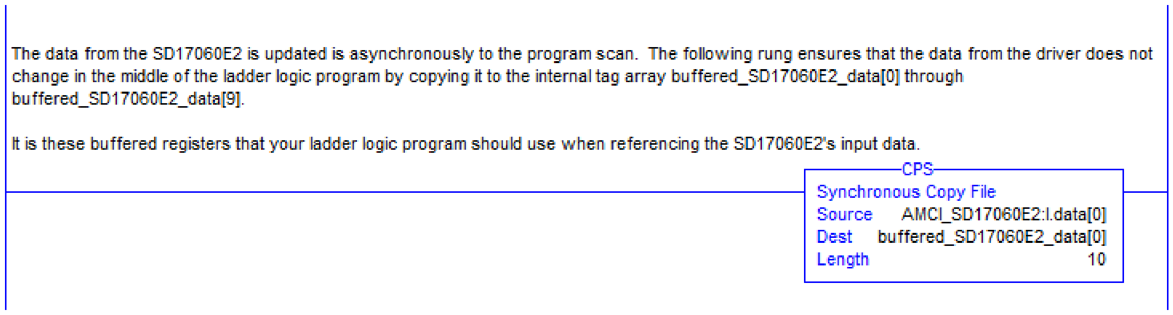
4.3 Configuring the SD17060E2 or SD31045E2

With the configuration assembly instance size set the zero, the device will join the EtherNet/IP network as soon as the request is made to it. If the unit does not have a valid configuration stored in its flash memory, it will show a configuration error in its network input data. 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 chapter, which starts on page 61.

It is possible to store configuration data in the flash memory of the AMCI EtherNet/IP Stepper Driver 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.

4.4 Buffering I/O Data

Input and output data is transferred asynchronously to the program scan. 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.



The data from the SD17060E2 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_SD17060E2_data[0] through buffered_SD17060E2_data[9].

It is these buffered registers that your ladder logic program should use when referencing the SD17060E2's input data.

```
CPS
Synchronous Copy File
Source      AMCI_SD17060E2:I.data[0]
Dest     buffered_SD17060E2_data[0]
Length                           10
```

Figure T4.4 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.

The format of the output and input data is covered in the ***Configuration Mode Data Format*** and ***Command Mode Data Format*** reference chapters, starting on pages 61 and 69 respectively.

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 EtherNet/IP Stepper Driver. Explicit messaging can be use 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 is used as an example in this chapter.

5.1 Required Message Instructions

Two instructions are required to transfer data between the PLC and the EtherNet/IP Stepper Driver. One instruction reads data from the unit and the other writes data to it. 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 T5.1 Message Instruction Attributes

NOTE Only RSLogix 500 version 8.0 or above can be used to configure Message Instructions to communicate with an EtherNet/IP device.

5.2 Create Four New Data Files.

- An Integer file to contain the data from the SD17060E2 or SD31045E2. This file must be at least 10 words in length.
- An Integer file to contain the data sent to the SD17060E2 or SD31045E2. 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.

5.3 Add the Message Instruction(s) to your Ladder Logic

The following rungs show how you can alternately read data from and write data to your Networked Driver.

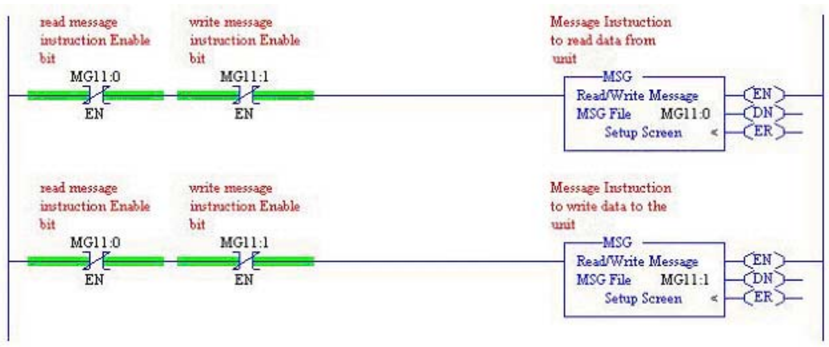
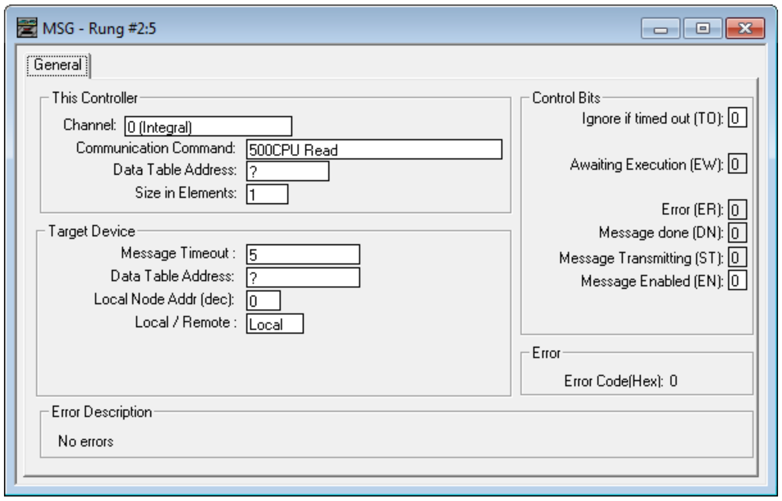

read message instruction Enable bit
write message instruction Enable bit
MG11:0
MG11:1
EN
EN
Message Instruction to read data from unit
MSG
Read/Write Message
MSG File MG11:0
Setup Screen
EN
DN
ER
read message instruction Enable bit
write message instruction Enable bit
MG11:0
MG11:1
EN
EN
Message Instruction to write data to the unit
MSG
Read/Write Message
MSG File MG11:1
Setup Screen
EN
DN
ER

Figure T5.1 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.



This Controller
Channel: 0 (Integral)
Communication Command: 500CPU Read
Data Table Address: ?
Size in Elements: 1

Target Device
Message Timeout : 5
Data Table Address: ?
Local Node Addr (dec): 0
Local / Remote : Local

Control Bits
Ignore if timed out (TO): 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

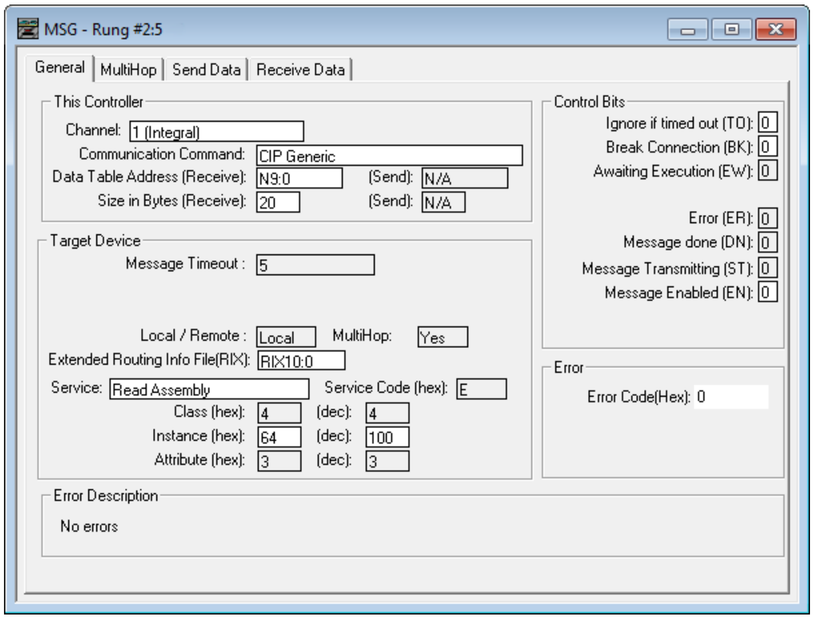
Figure T5.2 Message Instruction Setup Screen

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 SD17060E2 or SD31045E2, 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 SD17060E2 or SD31045E2, 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. Each EtherNet/IP Stepper Driver 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.

5.3 Add the Message Instruction(s) 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 SD17060E2 or SD31045E2, or "Write Assemble" for a Message Instruction that is being used to send data to the SD17060E2 or SD31045E2, 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 SD17060E2 or SD31045E2. Please note that the Data Table Address (Receive) field may be different in your application.

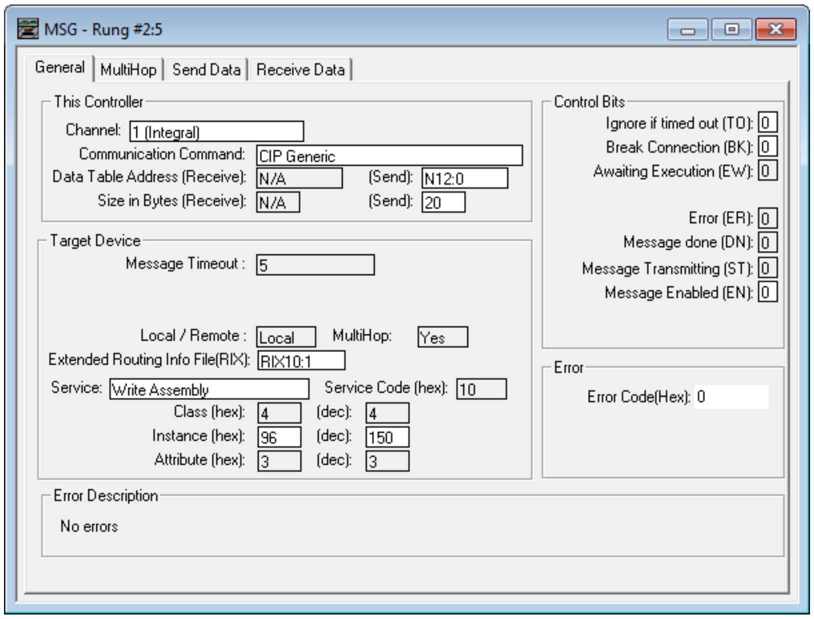


Channel: 1 (Integral)
Communication Command: CIP Generic
Data Table Address (Receive): N9:0 (Send): N/A
Size in Bytes (Receive): 20 (Send): N/A
Message Timeout : 5
Local / Remote : Local MultiHop: Yes
Extended Routing Info File(RIX): RIX10:0
Service: Read Assembly Service Code (hex): E
Class (hex): 4 (dec): 4
Instance (hex): 64 (dec): 100
Attribute (hex): 3 (dec): 3
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 Code(Hex): 0
No errors

Figure T5.3 Read Message Instruction Setup Screen

5.3 Add the Message Instruction(s) to your Ladder Logic (continued)

The figure below show a typical configuration for Message Instructions being used to write data to the SD17060E2 or SD31045E2. Please note that the Data Table Address (Send) field may be different in your application.



This Controller
Channel: 1 (Integral)
Communication Command: CIP Generic
Data Table Address (Receive): N/A (Send): N12:0
Size in Bytes (Receive): N/A (Send): 20

Target Device
Message Timeout : 5
Local / Remote : Local MultiHop: Yes
Extended Routing Info File(RIX): RIX10:1
Service: Write Assembly Service Code (hex): 10
Class (hex): 4 (dec): 4
Instance (hex): 96 (dec): 150
Attribute (hex): 3 (dec): 3

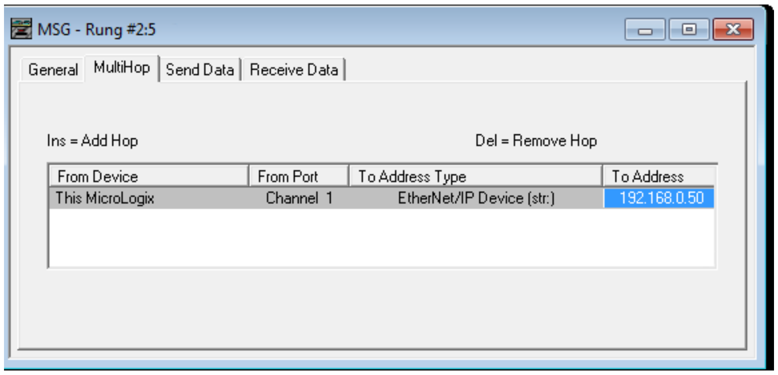
Control Bits
Ignore if timed out (TO): 0
Break Connection (BK): 0
Awaiting Execution (EW): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

Error Description
No errors

Figure T5.4 Write Message Instruction Setup Screen

Click on the MultiHop tab on the top of the window. As shown in figure T5.5, enter the IP address of the SD17060E2 or SD31045E2 and press Enter.



Ins = Add Hop Del = Remove Hop

From Device	From Port	To Address Type	To Address
This MicroLogix	Channel 1	EtherNet/IP Device (str.)	192.168.0.50

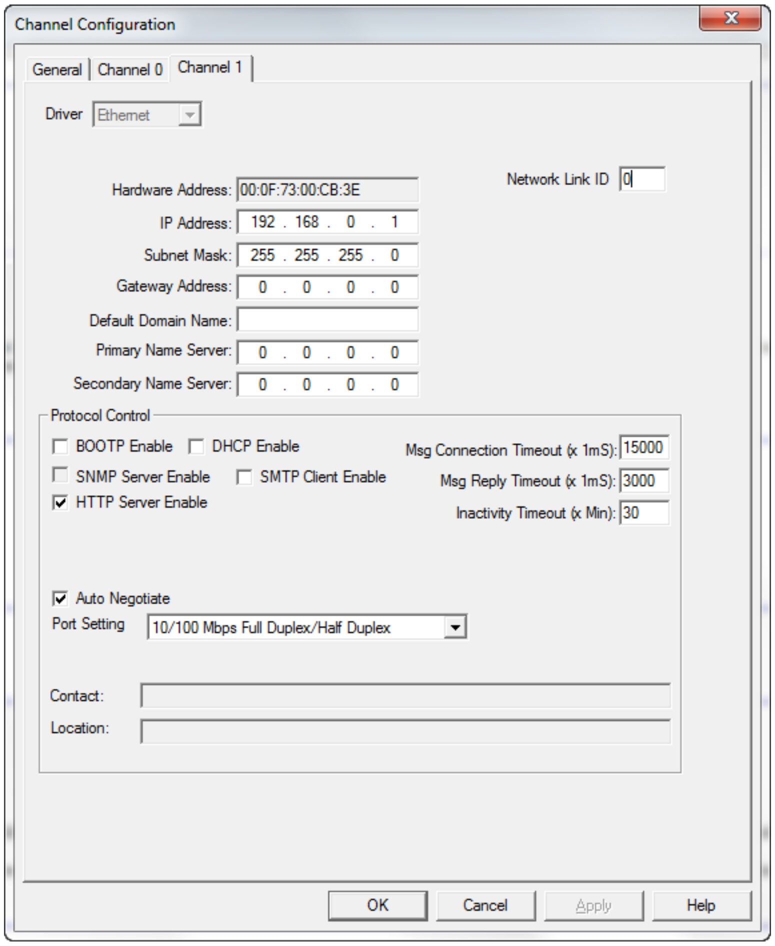
Figure T5.5 Message Instruction MultiHop Settings

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

5.4 Troubleshooting

If you are unable to communicate with the SD17060E2 or SD31045E2, 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.



Driver: Ethernet

Hardware Address: 00:0F:73:00:CB:3E

IP Address: 192 . 168 . 0 . 1

Subnet Mask: 255 . 255 . 255 . 0

Gateway Address: 0 . 0 . 0 . 0

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name Server: 0 . 0 . 0 . 0

Network Link ID: 0

☐ BOOTP Enable ☐ DHCP Enable

☐ SNMP Server Enable ☐ SMTP Client Enable

☑ HTTP Server Enable

Msg Connection Timeout (x 1mS): 15000

Msg Reply Timeout (x 1mS): 3000

Inactivity Timeout (x Min): 30

☑ Auto Negotiate

Port Setting: 10/100 Mbps Full Duplex/Half Duplex

Contact:

Location:

Figure R0.1 MicroLogix Ethernet Configuration Screen

2) Enter the IP address and Subnet Mask of your MicroLogix 1100, (not the address of the SD17060E2 or SD31045E2) 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

MODBUS TCP CONFIGURATION

An AMCI Networked Driver configured for the Modbus TCP protocol requires a host controller to issue configuration data and motion commands to these units. This chapter tell you how the I/O words used by an AMCI Networked Driver are mapped to the Modbus I/O registers.

6.1 Enable Modbus TCP Protocol

The embedded webserver can be used to change the communications protocol used by the SD17060E2 or SD31045E2. This is typically done while setting the IP address. Specifically, follow the steps in section 2.2b, ***Use the Embedded Web Server***, which starts on page 110. The AMCI Net Configurator utility can also be used to change the communications protocol. Follow the steps in section 2.2c, ***Use the AMCI NET Configurator Utility*** which starts on page 113.

6.2 Modbus Addressing

The register addresses used in this manual are the *Modbus logical reference numbers*†, 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.

6.2.1 Modbus Table Mapping

The Discrete Input and Input Register tables in the Modbus data model map to the same physical memory locations in the SD17060E2 and SD31045E2 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 SD17060E2 and SD31045E2 units.

- These registers hold data that is from the host controller to the driver. 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.

† MODBUS Application Protocol Specification V1.1b3, section 4.3: MODBUS Data model. ***www.modbus.org***

6.2 Modbus Addressing (continued)

6.2.2 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 SD17060E2 or SD31045E2 to your host controller. Refer to your host controller's documentation for information on how to accomplish this.

6.3 AMCI Modbus TCP Stepper Driver Memory Layout

The Stepper Driver 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 T6.1 shows how an AMCI Networked Driver is mapped to the Modbus data reference. The complete specification for the Modbus protocol can be downloaded at http://www.modbus.org/specs.php.

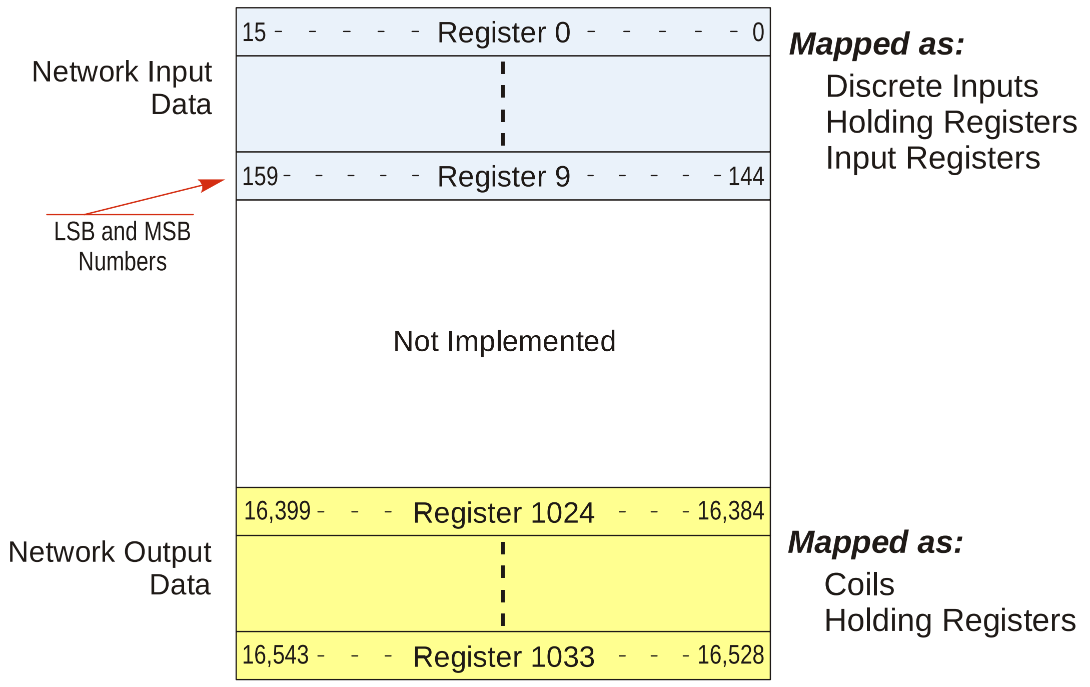


Figure T6.1 Modbus Data Reference Map

6.4 Supported Number of Connections

All Networked Drivers 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.

6.5 Supported Modbus Functions

Function Code	Function Name	Networked Driver 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 T6.1 Supported Modbus Functions

Table T6.1 above lists all of the Modbus functions supported by the Networked Drivers. AMCI supports all of these functions so that you can control the driver 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 Networked Driver buffers the data that it sends over the network. If you use the *Read/Write Registers* function to write configuration data to the unit, then the data read with that command will not contain the response to the new configuration data. The response to the new data will be sent with the next data read.

6.6 Supported Modbus Exceptions

Code	Name	Description
01	Illegal function	The module 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 T6.2 Supported Modbus Exceptions

Notes

TASK 7 (PROFINET Option)

PROFINET Network Configuration

This chapter outlines the steps commonly needed to get a Networked Driver 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.

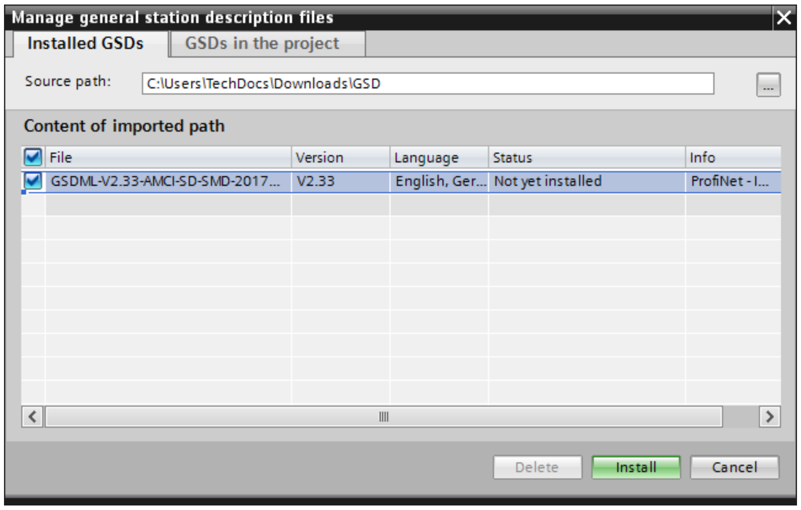
1) Download the ZIP archive that contains the GSDML files for the Networked Driver from the www.amci.com website.
2) Install the GSDML file into the configuration software for your host controller.
3) Add the Networked Driver to the PROFINET Network.
4) Set the I/O word addresses used to communicate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

7.1 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.

7.2 GSDML File Installation

1) Open or create a new project that will include the Networked Driver 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.



Source path: C:\Users\TechDocs\Downloads\GSD

Content of imported path

File	Version	Language	Status	Info
☑ GSDML-V2.33-AMCI-SD-SMD-2017...	V2.33	English, Ger...	Not yet installed	ProfiNet - I...

Figure T7.1 GSD File Installation

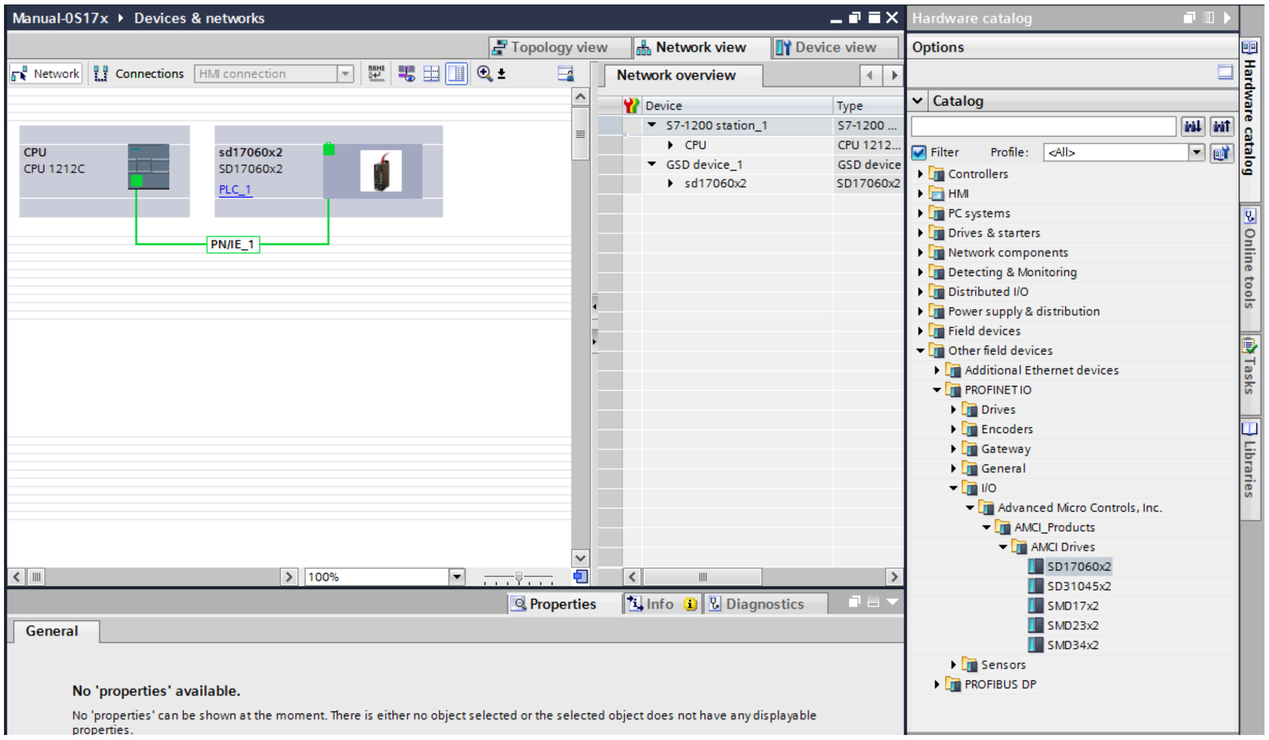
7.3 Configure the PROFINET Network

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

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

7.4 Add the Networked Driver 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 “SD”, or browse to the SD17060x2 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 SD17060x2 icon onto the PROFINET network line to connect the device to the network.



Device	Type
S7-1200 station_1	S7-1200 ...
CPU	CPU 1212...
GSD device_1	GSD device
sd17060x2	SD17060x2

No 'properties' available.

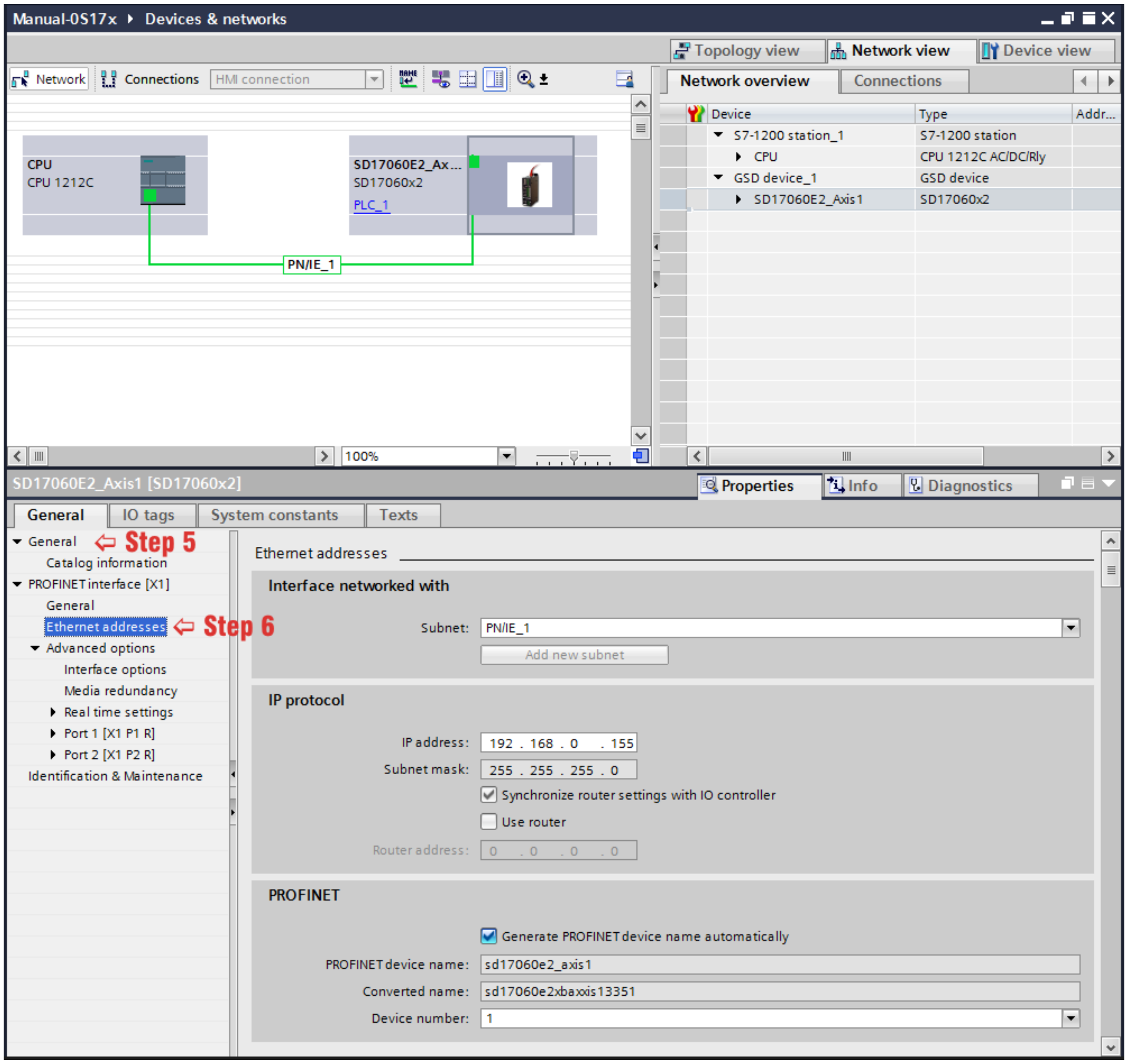
No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

Figure T7.2 Networked Driver Added to PROFINET Network

7.4 Add the Networked Driver to the PROFINET Network (continued)

5) Right click on the SD17060x2 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 Networked Driver 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 Networked Driver.



Device	Type	Addr...
S7-1200 station_1	S7-1200 station	
CPU	CPU 1212C AC/DC/Rly	
GSD device_1	GSD device	
SD17060E2_Axis1	SD17060x2	

Ethernet addresses

Interface networked with

Subnet: PN/IE_1

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: sd17060e2_axis1
Converted name: sd17060e2xbaxis13351
Device number: 1

Figure T7.3 Networked Driver IP Addressing

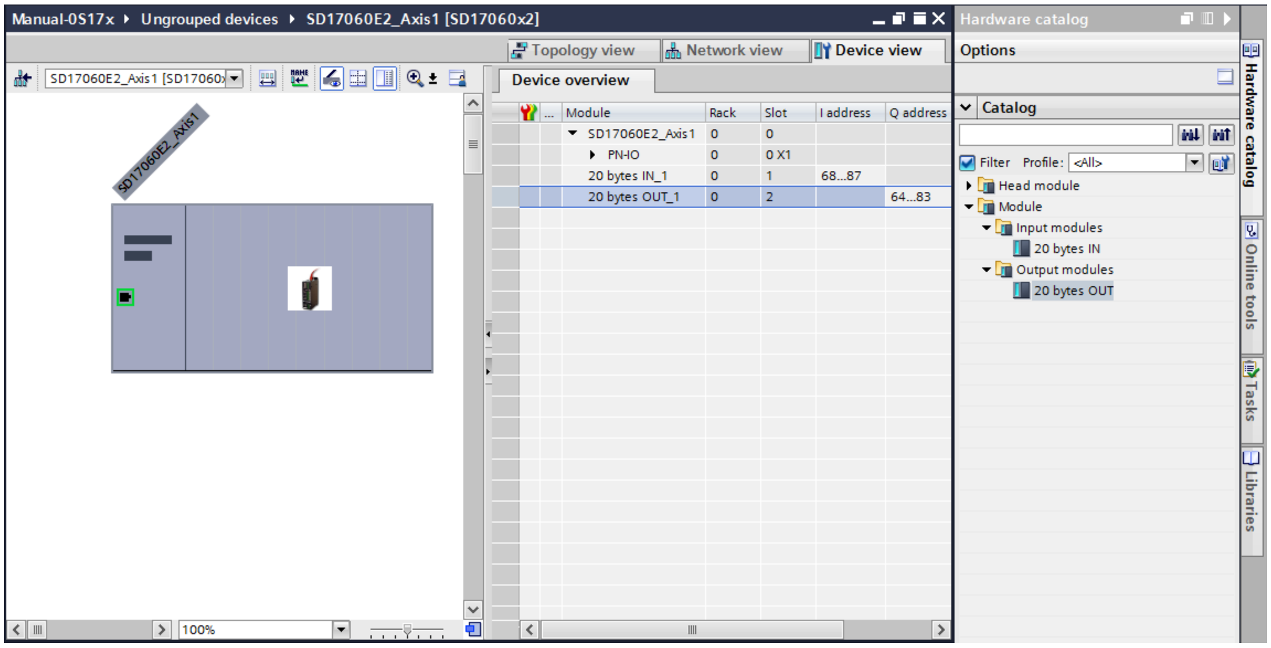
7.5 Set the I/O Configuration

The Networked Drivers 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 T7.1.

Input / Output Bytes of a Networked Driver	Input / Output Modules of a Networked Driver
20 Input Bytes	Input Module - Slot 1: 20 bytes
20 Output Bytes	Output Module - Slot 2: 20 bytes

Table T7.1 PROFINET I/O Configuration

1) With the SD17060x2 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.



Device overview

Module	Rack	Slot	I address	Q address
SD17060E2_Axis1	0	0		
PN-IO	0	0 X1		
20 bytes IN_1	0	1	68...87	
20 bytes OUT_1	0	2		64...83

Figure T7.4 I/O Byte Mapping

7.6 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

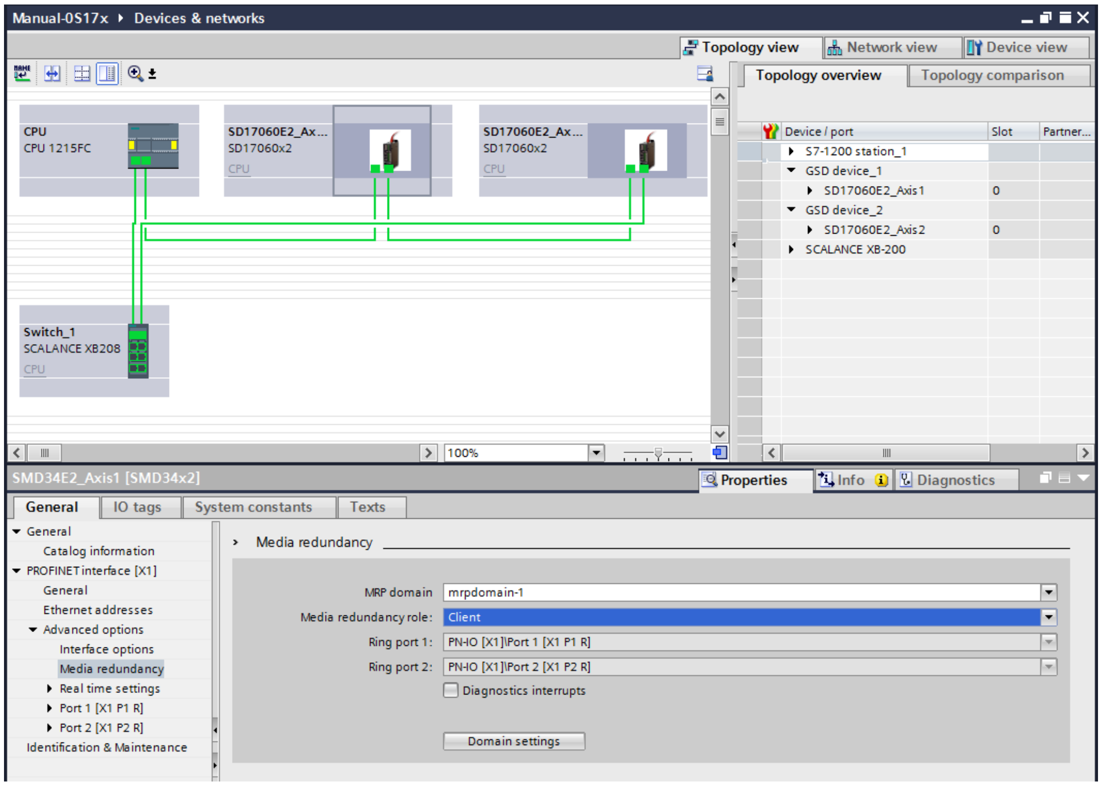
At this point, the Networked Driver is configured and ready to use. If you are using the unit in a redundant, ring based, network that uses the Media Redundancy Protocol (MRP), continue with the following instructions.

Media Redundancy Protocol (MRP) installations require that the Networked Driver 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.

7.7 Configure the SD17060x2 as an MRC

The Networked Driver 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 SD17060x2 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 Networked Driver.



Device / port	Slot	Partner...
▸ S7-1200 station_1		
▾ GSD device_1		
▸ SD17060E2_Axis1	0	
▾ GSD device_2		
▸ SD17060E2_Axis2	0	
▸ SCALANCE XB-200		

Media redundancy

MRP domain: mrpdomain-1
Media redundancy role: Client
Ring port 1: PN-IO [X1]\Port 1 [X1 P1 R]
Ring port 2: PN-IO [X1]\Port 2 [X1 P2 R]
☐ Diagnostics interrupts

Figure T7.5 MRP Topology and Client settings

7.7 Configure the SD17060x2 as an MRC (continued)

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 Networked Driver. Configuring your firewall to allow communication with the Networked Driver 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 stepper driver.

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

A.3 Configure Your Network Interface

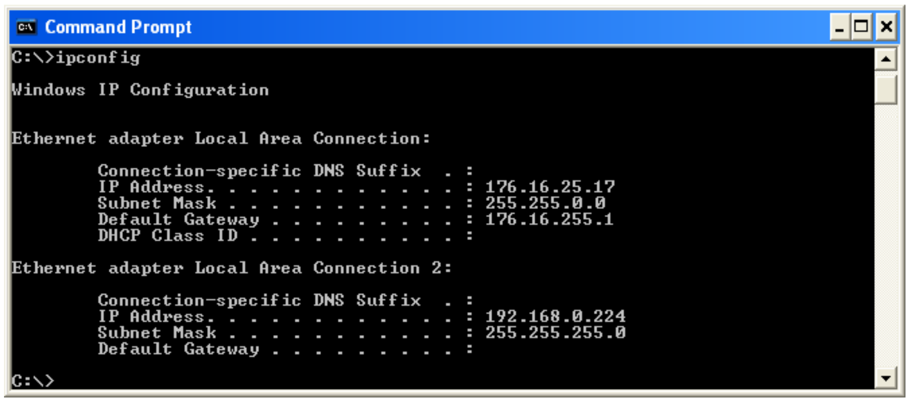
Before you can communicate with the Networked Driver, your network interface must be on the same subnet as the driver.

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 Networked Driver. 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 in not in the 192.168.0.xxx range, type in ‘ncpa.cpl’ at the command prompt and hit [Enter] on the keyboard.

- In 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.
- In 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 SD17060E2 or SD31045E2 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 SD17060E2 or SD31045E2. The computer will ping the Networked Driver 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 SD17060E2 or SD31045E2
- You did not enter the correct address in the ping command.
- The IP address of the SD17060E2 or SD31045E2 is not set correctly.
- The SD17060E2 or SD31045E2 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

LEADERS IN ADVANCED CONTROL PRODUCTS