AMCI ADVANCED MICRO CONTROLS INC.

Manual #: 940-0S340

SMD23E-B
Integrated Stepper
Indexer/Driver/Motor

SMD23E-B130E
SMD23E-B240E

User Manual

AMCI Motion Control Products

EtherNet/IP™

Modbus

GENERAL INFORMATION

Important User Information

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

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

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

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

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

Notes

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 set-up, installation, and operation of the SMD23E-B Integrated Stepper Indexer/Driver/Motors from AMCI. It is written for the engineer responsible for incorporating these units into a design, as well as the engineer or technician responsible for their actual installation.

Applicable Units

This manual applies to SMD23E-B130E and SMD23E-B240E products. These part numbers use a new motor that is sealed to IP65 instead of the older version that carried an IP50 rating. There are changes between the two motor flanges that affect mounting in some cases.

With the exception of the outline drawings, all of the information in this manual applies to the older SMD23E-130E and SMD23E-240E units. If you have an older unit are need an outline drawing, contact AMCI technical support (techsupport@amci.com) for further assistance.

New Model Number	Retired Model Number	Description
SMD23E-B130E	SMD23E-130E	SMD23E with 130 oz-in IP65 motor.
SMD23E-B240E	SMD23E-240E	SMD23E with 240 oz-in IP65 motor.

Part Number Description

Revision Record

This manual, 940-0S340, is the first release of this manual. It was first released February 4th, 2026.

Revision History

940-0S340 Initial Release.

Navigating this Manual

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

Bookmarks of all the chapter names, section headings, and sub-headings were created in the PDF file to help navigate it.

Throughout this manual you will also find ***blue text that functions as a hyperlink*** in HTML documents. Clicking on the text will immediately jump you to the referenced section of the manual. If you are reading a printed manual, most links include page numbers.

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 the first time a new term is introduced.
Cross Reference	When viewing the PDF version of the manual, clicking on the cross reference text jumps you to referenced section.
HTML Reference	When viewing the PDF version of the manual, clicking on the HTML reference text will open your default web browser to the referenced web page.

Trademark Notices

The AMCI logo and "SynchroStep" are trademarks 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.

Manual Layout

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

- If you are curious about the SMD23E-B products, this manual contains the information you need to determine if these products are the right choice for your application. Chapter 1, ***SMD23E-B Specifications***, was written for you. The chapter contains all of the information you will need to specify the right SMD23E-B product for your application.
- If you need to install and use an SMD23E-B, then the rest of the manual is written for you. To simplify installation and configuration, the rest of the manual is broken down into *references* and *tasks*. References give you the information you need to integrate an SMD23E-B into your system and tasks give you specific procedures to complete the integration.

Manual Layout (continued)

The table below gives a brief description of each chapter's contents to help you find the information you need to do your job.

Section Title	Page	Intended Audience
SMD23E-B Specifications	11	Anyone new to the SMD23E-B. This chapter gives a basic overview of the features available on the unit, typical applications, and electrical specifications.
Motion Control	23	Anyone that needs detailed information on how the unit can be used to control motion in your application.
Calculating Move Profiles	45	Reference information on calculating detailed move profiles.
Homing The SMD23E-B	55	Anyone that needs detailed information on how the home position of the SMD23E-B can be set.
Configuration Mode Data Format	59	Anyone that needs detailed information on the format of the network data to and from the SMD23E-B used to configure the unit.
Command Mode Data Format	67	Anyone that needs detailed information on the format of the network data to and from the SMD23E-B used to command the unit.
Configuring Network Interfaces	85	Anyone that needs guidance on how to configure a laptop or PC to communicate with a SMD23E-B.
Installation	87	Anyone that must install an SMD23E-B on a machine. Includes information on mounting, grounding, and wiring specific to the units.
Set the IP Address & Network Protocol	91	Anyone that must set the IP address and communications protocol of the SMD23E-B.
Implicit Communications With an EDS	97	Task instructions that cover how to add an SMD23E-B to an EtherNet/IP host that supports the use of EDS files.
Implicit Communications Without an EDS	103	Task instructions that cover how to add and SMD23E-B to a project as a generic device. This configuration is for EtherNet/IP hosts that do not support EDS files but support implicit communications.
EtherNet/IP Explicit Messaging	107	Task instructions for adding message instructions to your controller's program that read and write data to the SMD23E-B.
Modbus TCP Data Addressing	113	Anyone that needs information on configuring a Modbus TCP host.

Notes

REFERENCE 1

SMD23E-B Specifications

This manual is designed to get you quickly up and running with the SMD23E-B Integrated Stepper Indexer/Driver/Motor. As such, it assumes you have a basic understanding of stepper systems, such as the resolution you want to run your motor at, and the reasons why you'd want to use Idle Current Reduction and the reasons why you wouldn't. If these terms or ideas are new to you, we're here to help. AMCI has a great deal of information on our website and we are adding more all the time. If you can't find what you're looking for at http://www.amci.com, send us an e-mail or call us. We're here to support you with all of our knowledge and experience.

The SMD23E-B Family

The SMD23E-B is a product line from AMCI with a simple concept: a low-cost, stepper indexer, driver, and motor that can be attached to any popular industrial network. Each SMD23E-B attaches to an Ethernet network and communicates with either the EtherNet/IP or Modbus TCP protocols. The SMD23E-B units appear as a network node to your controller.

There are presently two members of the SMD23E-B family:

- SMD23E-B130E: 130 oz-in motor with built in encoder
- SMD23E-B240E: 240 oz-in motor with built in encoder

The built-in 1,024 count/turn encoder gives you additional functionality, such as position verification, moves based on encoder position, and stall detection.

Figure R1.1 SMD23E-B

General Functionality

Each member of the SMD23E-B family has three integrated parts:

- An indexer that accepts commands over an Ethernet connection using the EtherNet/IP or Modbus TCP protocol
- A 3.4Arms micro-stepping driver that accepts 24 to 48Vdc as its input power source
- A high torque size 23 stepper motor (130 or 240 oz-in holding torque)
- An incremental encoder for position feedback and stall detection.

The availability of the Ethernet/IP and Modbus TCP protocols makes the SMD23E-B units easy to integrate into a wide variety of controller systems. This 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 SMD23E-B unit
- All motion logic is programmed in the host, eliminating the need to learn a separate motion control language
- Eliminating the separate indexer and driver lowers Total System Cost

An SMD23E-B is powered by a nominal 24 to 48Vdc power source, and can accept surge voltages of up to 60Vdc without damage. The output motor current is fully programmable from 1.0Arms to 3.4Arms which makes the SMD23E-B suitable to a wide range of applications. In addition to the Motor Current setting, the Motor Steps per Turn, Idle Current Reduction, and Anti-Resonance Circuit features are also fully programmable. If you have used other stepper indexer products from AMCI you will find programming an SMD23E-B to be very similar to these products.

The SMD23E-B Family (continued)

General Functionality (continued)

The SMD23E-B contains a true RMS motor current control driver. This means that you will always receive the motor's rated torque regardless of the *Motor Steps/Turn* setting. (Drivers that control the peak current to the motor experience a 30% decrease in motor torque when microstepping a motor.) The SMD23E-B automatically switches from RMS to peak current control when the motor is idle to prevent overheating the motor.

The SMD23E-B units have three DC inputs that are used by the indexer. Configuration data from the host sets the function of these inputs. Each input can be individually configured as a:

- CW or CCW Limit Switch
- Home Limit Switch
- Capture Position Input (Will capture the encoder position.)
- Stop Manual or Registration Move Input
- Start Indexer Move
- Emergency Stop Input
- General Purpose Input

Motion Axis Functionality (*SynchroStep* Technology)

On host platforms that offer motion control, such as the RA ControlLogix platform, a programmed axis can be used to supply data to the SMD23E-B by copying position and velocity data from the axis to the registers assigned to the unit. This allows the unit to perform complex motion profiles that are programmed using your host's motion control instructions.

Encoder Functionality

All SMD23E-B units have an internal 1,024 count/turn incremental encoder. Using an encoder gives you the ability to:

- Verify motor position at the end of a move
- Detect motor stall conditions

Torque and Power Curves

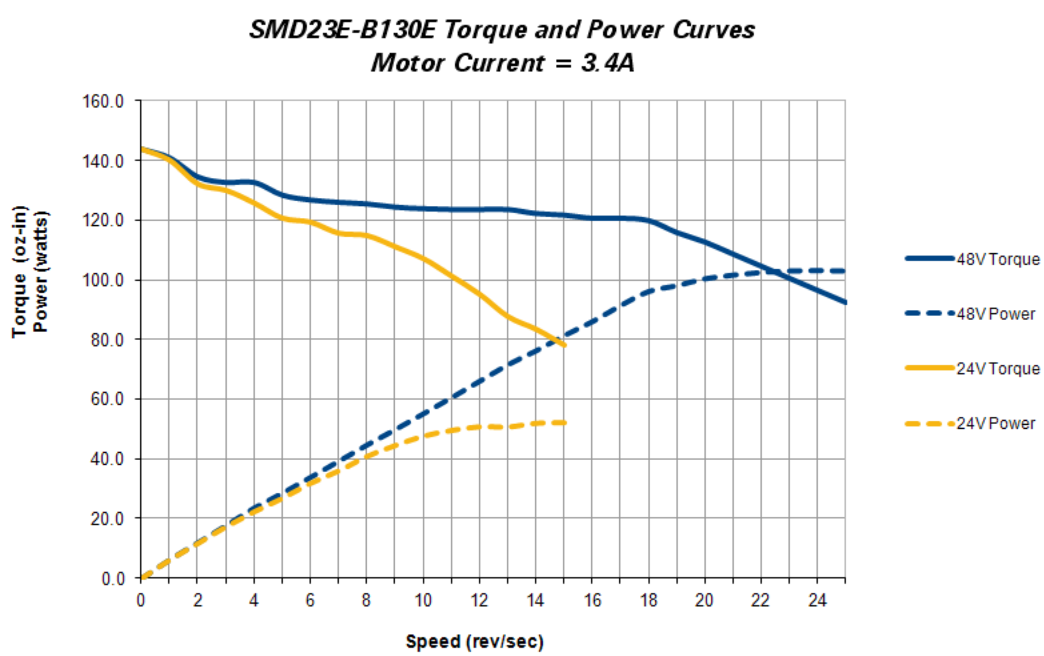


Figure R1.2 SMD23E-B130E Torque and Power Curves

Torque and Power Curves (continued)

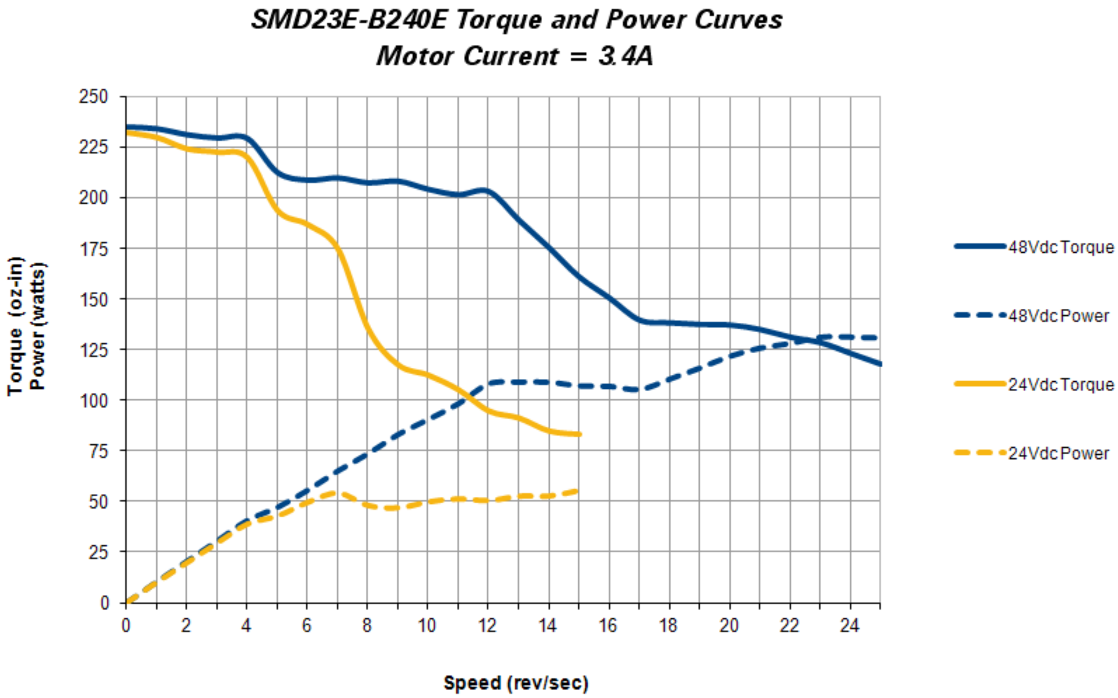


Figure R1.3 SMD23E-B240E Torque and Power Curves

Power Supply Sizing

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

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

Example 1: An SMD23E-B130 will be running at a maximum of 7 RPS and a 48 Vdc supply will be used. Based on the power curve in figure R1.2 on the previous page, the combinations will generate a maximum of 40 Watts. Therefore a 48 Vdc supply with a power range of 60 W to 70 W can be used in the application.

Example 2: An SMD23E-B240 will be running at a maximum of 9 RPS and a 24 Vdc supply will be used. Based on the power curve in figure R1.3 on the previous page, the power at this speed is 45 W, but the maximum power over the entire speed range is 55 W, which occurs at 7 RPS. Therefore, the 55 Watt value should be used, and the 24 Vdc supply should be able to provide 83 W to 97 W.

Power Supply Sizing (continued)

Table R1.1 below shows the suggested power supply sizes based on the maximum power the motor can generate over its entire speed range.

		SMD23E-B130			SMD23E-B240		
		Motor Power	150% Supply	175% Supply	Motor Power	150% Supply	175% Supply
Supply Voltage	24 Vdc	53W	80 W	93W	57 W	86 W	100 W
	48 Vdc	105 W	158 W	184 W	135 W	203 W	237 W

Table R1.1 Suggested Power Supply Ratings

Regeneration (Back EMF) Effects

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

The stepper motors used in the SMD23E-B units are all low inductance motors. Back EMF is typically not an issue unless there is a gearhead attached to the motor and the gearhead is driven by hand. In these instances, the speed of the motor is multiplied by the ratio of the gearhead. The resulting high speed of the motor's shaft can lead to large enough voltage spikes to damage the attached power supply.

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

Specifications

Driver Type

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

Physical Dimensions

See page 88

Weight

SMD23E-B-130E .... 2.20 lbs. (1.00 kg.)
SMD23E-B-240E .... 2.70 lbs. (1.23 kg.)
All weights are without mating connectors

Maximum Shaft Loads

SMD23E-B130E: 300 g*cm^2

SMD23E-B240E: 480 g*cm^2

Rotor Inertia

Radial: 19 lbs (85N) at end of shaft

Axial: 3.37 lbs (15N)

Maximum Operating Temperature

203°F /95°C (Over temperature fault removes motor current at this temperature.)

Inputs

Electrical Characteristics: . . . . Differential. 2500 Vac/dc opto-isolated. Can be wired as single ended inputs.

DC Inputs accept 3.5 to 27Vdc without the need for an external current limiting resistor. Inputs require 5 mA to activate.

Faults

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

Motor Current

Programmable from 1.0 to 3.4 Arms in 0.1 Amp steps.

Motor Counts per Turn

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

Incremental Encoder

Internal incremental encoder that supplies 1,024 counts per turn.

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.

Environmental Specifications

Input Power .... 24 to 48Vdc, surge to 60Vdc without damage to the unit.

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

IP Rating SMD23E-B: IP20

Status LED's

See ***Status LED's*** section starting on page 18.

Connectors

All mating connectors are available separately under the following AMCI part numbers. The MS-2X4 is included with SMD23E-B units.

Connector	AMCI Part #	Wire	Strip Length	Connection Type
I/O	MS-2X4	24 - 16 AWG	0.275 inches	Spring Cage Connector

Indexer Functionality

The table below lists the functionality offered by the indexer built into the SMD23E-B.

Feature	Description
Ethernet/IP or Modbus TCP	Allows easy setup and communication with a wide range of host controllers such as the latest PLC's from Allen-Bradley.
Programmable Inputs	Each of the inputs can be programmed as a Home Limit, Over Travel Limit, Capture Input, Manual Jog Stop, Start Indexer Move, E-Stop, or a General Purpose Input.
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. An SMD23E-B homes to a discrete input and can use a bit in the Network Data as a home proximity input.
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.
Jog Move	Allows you to jog the motor in either direction based on an input bit from your host controller.
Registration Move	Allows you to jog the motor in either direction based on an input bit from your host controller. When a controlled stop is issued, the move will output a programmable number of steps before coming to a stop.
Blend Move	Allows you to perform a sequence of relative moves without stopping between them.
Dwell Move	Allows you to perform a sequence of relative moves with a stop between each move that has a programmable length of time. Used to create highly accurate move profiles that avoid network latency issues.
Indexer Move	Allows you to program a move that is held in the host's output registers. The move is run when one of the programmable inputs makes a transition.
Hold Move	Allows you to suspend a move, and optionally 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	The encoder can be used to verify motion when a move command is issued.
SynchroStep Move	Allows the SMD23E-B to follow a PLC's motion axis.

Table R1.2 Indexer Functionality

Stall Detection with the SMD23E-B

When Stall Detection is enabled, the unit monitors the encoder for position changes, regardless of whether or not a move is in progress. If the error between the encoder position and the motor position exceeds forty-five degrees, the SMD23E-B responds in the following manner:

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

Driver Functionality

This table summarizes the features of the stepper motor driver portion of an SMD23E-B.

Feature	Benefits
RMS Current Control	RMS current control give an SMD23E-B the ability to drive the motor at its fully rated power when microstepping. Peak current controllers typically experience a 30% drop in power when microstepping a motor.
Programmable Motor Current	RMS current supplied to the motor can be programmed from 1.0 to 3.4 amps in 0.1 amp increments. Reducing the motor current to the minimum needed for your application will significantly reduce the motors operating temperature
Programmable Idle Current Reduction	Extends motor life by reducing the motor current when motion is not occurring. This extends the life of the motor by reducing its operating temperature.
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 each SMD23E-B 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.
Over Temperature Detection	An SMD23E-B sets a warning bit in the network data when the temperature of the unit approaches its safe operating threshold.
Over Temperature Protection	Protects your SMD23E-B from damage by removing power from the motor if the internal temperature of the driver exceeds a safe operating threshold.

Table R1.3 Driver Functionality

Idle Current Reduction

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

Idle current reduction should be used whenever possible. By reducing the current, you are reducing the I^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 SMD23E-B always switch from RMS to peak current control when the motor is idle to prevent motor damage due to excessive heating.)

Status LED's

Each SMD23E-B has two status LED's that show module and network status. As shown in figure R1.4, these LED's are located on the rear cover. An additional status LED on the RJ45 connector shows the status of the physical Ethernet link.

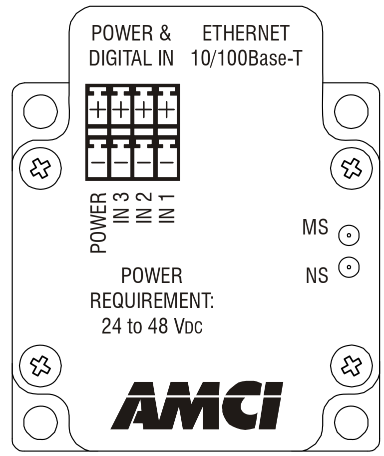


Figure R1.4 Rear Cover Status LED's

Module Status (MS) LED

The Module LED is a bi-color red/green LED that show the general status of the unit.

LED State	Definition
Off	No Power
Alternating Red/Green	Initializing: Power up Self-Test
	Communications failure. There is a communications error between the main processor and the Ethernet co-processor within the unit. You must cycle power to the SMD23E-B to attempt to clear this fault.
Flashing Green	Initializing: Waiting for valid physical connection to the network.
	Successful write to flash memory. Power must be cycled to the unit before additional commands can be written to it.
Steady Green	Drive and Network are operational.
Flashing Red	During Initialization: IP Address Conflict
	If the Network Status LED is also flashing, 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	Overtemperature fault. Remove power from the unit and allow it to cool to clear the fault.

Table R1.4 Module Status LED States

Power Up Behavior

- **Blinking Green:** The unit will blink the Module Status LED green during initialization.

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 SMD23E-B is configured to for.

LED State	EtherNet/IP Definition	Modbus TCP Definition
Off	No Power	No power or no TCP connections
Alternating Red/Green	Power up Self-Test	Power up Self-Test
Flashing Green	Ethernet connection, but no CIP connections	Indicates number of concurrent connections with a two second delay between groups. The SMD23E-B supports up to 3 concurrent connections.
Steady Green	Valid Ethernet network and CIP connections	Should not occur. LED should always flash when network is connected.
Flashing Red	Network Connection Timeout	Not implemented in Modbus TCP
Steady Red	Duplicate IP address on network.	

Table R1.5 Network Status LED States

SMD23E-B Connectors

Input Connector

As shown in figure R1.5, the Input Connector is located on the top of the unit. Connections to the three differential digital inputs and power supply connections are made at this connector. The mating connector is supplied with the SMD23E-B. Spares are also available from AMCI under the part number MS-2X4. They are also available from Phoenix Contact under their part number 173 88 27.

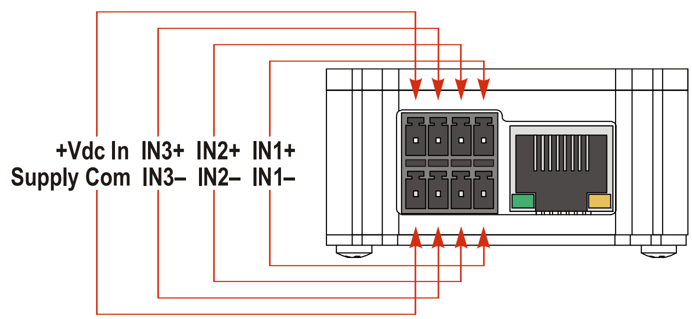


Figure R1.5 I/O Connector

Ethernet Port

The Ethernet port is also located on the top of the SMD23E-B. The connector is a standard RJ-45 jack that will accept any standard CAT 5+ cable. The Ethernet port on the SMD23E-B is an "auto-sense" port that will automatically switch between 10baseT and 100baseT depending on the network equipment it is attached to. The port also has "auto switch" capability. This means that a standard cable can be used when connecting the SMD23E-B to any device, including a personal computer.

Link Status LED

This LED is on when there is a physical link between the Ethernet port of the SMD23E-B and the Ethernet port of the device the SMD23E-B is plugged into. This LED will flash when data is being transmitted over the Ethernet link.

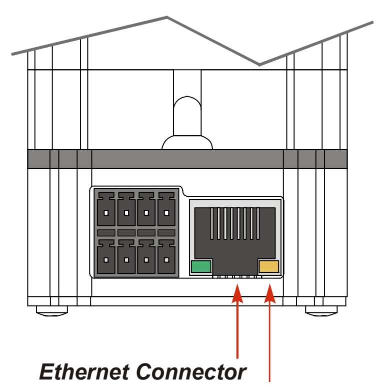


Figure R1.6 Ethernet Port Location

Available Discrete Inputs

The SMD23E-B has three discrete DC inputs that accept 3.5 to 27Vdc signals. (5 to 24Vdc nominal) They can be wired as differential, sinking, or sourcing inputs. How your SMD23E-B uses these inputs is fully programmable. The active state of each input is also programmable. Programming their active states allow them to act as Normally Open(NO) or Normally Closed(NC) contacts.

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. An SMD23E-B 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 SMD23E-B that will cause the unit to seek this sensor. How the SMD23E-B actually finds the Home sensor is described in the ***Homing The SMD23E-B*** chapter starting on page 55.

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 jog 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 an SMD23E-B to run critically timed moves that cannot be reliably started from the network due to data transfer lags.

If the quadrature encoder is enabled, the quadrature encoder position data will be captured whenever a Start Indexer Move Input input makes any transition, both active-to-inactive and inactive-to-active.

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. Any type of move, including a Jog or Registration Move, cannot 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 Move will bring the move to a controlled stop. The controlled stop is triggered on an inactive-to-active state change on the input. Only Jog Moves and Registration Moves can be stopped this way, all other moves ignore this input.

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

Capture Encoder Position Input

As described in the ***Start Indexer Move Input*** and ***Stop Jog or Registration Move Input*** sections above, an SMD23E-B can be configured to capture the encoder position value on a transition of a discrete DC input.

General Purpose Input

If your application does not require all of the inputs, you can configure the unused inputs as General Purpose Inputs. The inputs are not used by the SMD23E-B, but the input state is reported in the network data so their state can be monitored by your host controller.

Network I/O Bits

Position Status Bit

The SMD23E-B unit has a single bit in the input registers that reports the status of the motor position value. The name of this bit is Position_Invalid, and it is set the "1" when the reported motor position may not match the corresponding position on the machine. The Position_Invalid bit us always set on power up. A successful Find_Home command or a Preset_Position command will reset this bit to "0". Absolute Moves cannot be run while The Position_Invalid bit is set. Attempting one will result in a command error response from the module. All other move types can be run while this bit is set.

Motion Status Bits

The SMD23E-B unit has five bits in the input registers that report on the state of a move in progress. These bits are Stopped, Moving_CW, Moving_CCW, Accelerating, and Decelerating and are fully described in the following chapter.

Command Status Bits

The SMD23E-B unit has four bits in the input registers that report on the state of the last command issued to the module. These bits are Move_Complete, In_Hold_State, At_Home, and Command_Error. They are fully described in the following chapter.

Module Status Bits

The SMD23E-B unit has sixteen bits in the input registers that report on the state of the module. These bits are fully described in the following chapter.

Motion Command Bits

The following motion command bits are mutually exclusive. This means that only one of these bits can be at a logic "1" state at a time. The module will issue a command error if two or more of these bits are a logic "1" at the same time.

- Absolute Move
- Relative Move
- Hold Move
- Resume Move
- Find Home CW
- Find Home CCW
- Jog CW†
- Jog CCW†
- Run Blend CW
- Run Blend CCW
- Run Dwell Move
- Preset Current Position
- Preset Encoder Position
- Immediate Stop
- Reset Errors
- Store Minimum Registration Distance

† The two Jog bits are also used to trigger Registration Moves.

Blend Move Programming Bits

There are two bits in the network output data and two bits in the network input data that are used to program a Blend Move. The two output bits are named Program_Assembled and Read_Assembled_Data. The two input bits are named In_Assembled_Mode and Wait_For_Assembled_Segment. Their use is described in the ***Assembled Move Programming*** section of the following chapter, starting on page 40.

Notes

REFERENCE 2
MOTION CONTROL

When a move command is sent to an SMD23E-B, 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.

General Definitions

Unit of Measure

Distance: Every distance is measured in steps. Your programming will give you a specific number of steps needed to complete one rotation of the motor shaft. It is up to you to determine how many steps are required to travel the required 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 an SMD23E-B, 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 an SMD23E-B.
- 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 an SMD23E-B to the value used in the equations.

Motor Position

Clockwise moves will always increase the motor position register that is 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.

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 registers will automatically be set to zero once the home position is reached. Note that the unit must be configured to use one of the DC inputs as a Home Input before *Find_ Home* commands can be used.

Defining a Home Position is completely optional. Some applications, such as those that use the SMD23E-B for speed control, don't require position data at all.

Count Direction

Clockwise moves will always increase the motor position register that is 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 1,999,999 steps/second. This value is typically used to start the move above the motor's low frequency resonances and in micro-stepping applications to limit the amount of time needed for acceleration and deceleration. AMCI does not specify a default value in this manual because it is very dependent on motor size and attached load.

General Definitions (continued)

Target Position

The Target Position is 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 SMD23E-B 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 counter-clockwise 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.

Position Status Bit

The SMD23E-B unit has one position status bit, the Position_Invalid bit. This bit is set when the reported motor position may not correspond to the actual machine position. The reported motor position is a bi-directional counter that tracks the number, and direction, of motor steps. The actual machine position is the physical position of the axis on the machine.

Absolute Moves cannot be run while the Position_Invalid bit is set. All other move types can be run while the Position_Invalid bit is set.

There are two ways to reset the Position_Invalid bit.

1) Use a Find_Home command. If the command completes successfully, the motor position will be reset to zero and the Position_Invalid bit will be reset to zero.
2) Use a Preset_Position command. Once the command completes, the motor position will be set to the commanded value and the Position_Invalid bit will be reset to zero.

Five operations will always set the Position_Invalid bit.

1) Cycle power to the SMD23E-B.
2) Write Configuration data to the SMD23E-B.
3) Issue an Immediate_Stop command from the host. (Even when motion is not occurring.)
4) Activate an Emergency Stop input while motion is occurring.
5) Activate a CW or CCW end limit switch while motion is occurring.

Motion Status Bits

The SMD23E-B unit has five motion status bits:

- **Moving_CW:** Set while the SMD23E-B unit is moving clockwise.
- **Moving_CCW:** Set while the SMD23E-B unit is moving counter-clockwise.
- **Accelerating:** Set while the move is in its acceleration phase.
- **Decelerating:** Set while the move is in its deceleration phase.
- **Stopped:** Set when motion is not occurring.

Command Status Bits

The SMD23E-B unit has four command status bits:

- **Move_Complete:** Set at the end of any move that has a predefined stopping point. These moves are the Absolute, Relative, and Blend moves.
- **In_Hold_State:** Relative and Absolute Moves can be interrupted after they have started. The move decelerates to a stop and is placed in a "Hold State". This bit is on while the move is in its Hold State. Once it its Hold State, a move can be restarted or a new move can be issued.
- **At_Home:** This bit is set to "1" when a homing command completes successfully. It is reset to "0" when the next command is accepted.
- **Command_Error:** This bit is set to "1" when there is an error in a command.

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 SMD23E-B supports three types of accelerations and decelerations. The type of acceleration used is controlled by the *Acceleration Jerk* parameter.

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.

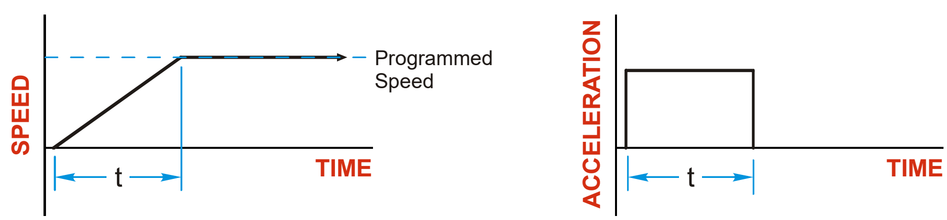


Figure R2.1 Linear Acceleration

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 considerably longer than a Linear Acceleration to achieve the same velocity. Below is an example where the total time of acceleration is twice that of a linear acceleration move.

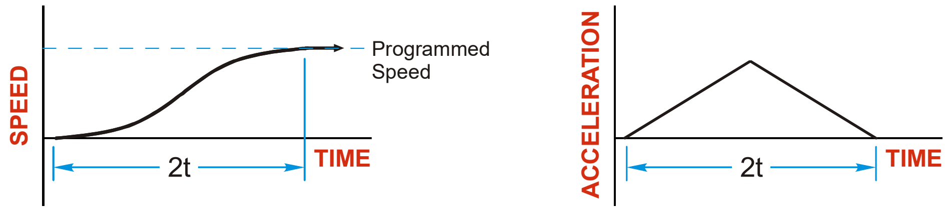


Figure R2.2 Triangular S-Curve Acceleration

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 R2.3 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.

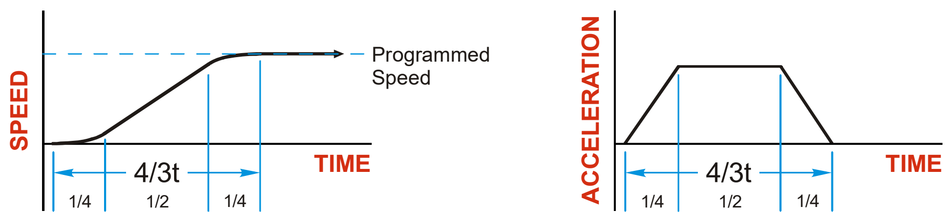


Figure R2.3 Trapezoidal S-Curve Acceleration

How to Control Moves in Progress

There are two ways of stopping a move in progress.

- **Controlled Stops:** The motor decelerates at the move's programmed deceleration value until it reaches the configured Starting Speed. The move stops at this point. If the Position_Invalid bit is reset to "0" at the end of the move, the motor position value is still considered valid. The machine does not need to be homed again before Absolute Moves can be run.
- **Immediate Stops:** The motor stops immediately without deceleration. Because it is possible for the inertia of the load to pull the motor beyond the stopping point, the motor position value is considered invalid after an Immediate Stop. The Position_Invalid bit will be set to "1" and the machine must be homed again before Absolute Moves can be run.

The SMD23E-B also has the ability to "hold" a relative and absolute move. The move is brought to a controlled stop, and the move is placed in a Hold State. The move can later be resumed and it will run to its completion. One example of the use of the Hold Move feature is on an axis that runs until there is an out of material condition. After the material is replenished, the axis can be restarted.

Note that a held move does not have to be resumed. If a new move is written to the unit while the previous move is in its held state, the previous move is canceled and the new move begins from the current position.

Available Inputs

The table below shows the available DC input types and available network bits that affect the available moves. The complete table is shown here for easy reference. The appropriate rows are repeated in the sections below that explain each move in detail.

Note that the SMD23E-B has three physical digital inputs, each of which can be programmed for one of seven different functions. One of these input functions is General Purpose Input. An input with this function type has its state reported over the network, but has no effect on moves.

	DC Digital Input Types							Backplane Bits			
	General Purpose	CW L.S.	CCW L.S.	Home L.S.	E-Stop	Stop Jog or Reg	Start Indexed	Indexed_Cmd	Hold Move	Resume Move	Immediate Stop
Absolute Move		1	1		2		3,4	3	5	6	2
Relative Move		1	1		2		3,4	3	5	6	2
CW Find Home		7	8	9	2			3	10		2
CCW Find Home		8	7	9	2			3	10		2
Jog_CW Move		11	12		2	13	3,4	3	5	6	2
Jog_CCW Move		12	11		2	13	3,4	3	5	6	2
CW Registration Move		11	12		2	14	3,4	3	10		2
CCW Registration Move		12	11		2	14	3,4	3	10		2
CW Blend Move		1	1		2		3,4	3	10		2
CCW Blend Move		1	1		2		3,4	3	10		2
Dwell Move		1	1		2		3,4	3	10		2
SynchroStep Move		1	1		2				10		2

Table R2.1 Control Inputs

See the numbered notes on the following page.
A blank cell means that the state of the input as no effect on the move.

How to Control Moves in Progress (continued)

Available Inputs (continued)

1) An active limit switch input will immediately stop all motion, and prevent further motion in that direction.
2) An inactive-to-active transition on an Emergency Stop Input, or an active Immediate_Stop command will immediately stop all motion. The Position_Invalid bit will be set if motion was occurring when either of these conditions became true.
3) If the Indexed_Command bit is set when the motion command is written to the unit, the move will not start until there is an inactive-to-active transition on the Start Indexed input.
4) If the encoder is enabled, a transition on the Start Indexed input will capture the encoder position and copy it to the Captured Encoder Position registers.
5) An active Hold command will start to decelerate the move. The move will stop when the configured Starting Speed is reached and the move will enter its Hold State.
6) An active Resume command will resume a move if it is in its hold state. Activating the Resume command when a move is not in its hold state will generate a Command Error response from the unit. If the move is running at this time, the move will continue.
7) An inactive-to-active transition on this limit switch input will immediately stop all motion. The unit will wait for two seconds, and then begin motion in the opposite direction while searching for the Home Input.
8) An inactive-to-active transition on this limit switch input will immediately stop all motion. The module will set the Position_Invalid and Input_Error bits and abort the Find_Home command.
9) Transitions on a Home Limit Switch input type will cause the controller to finish its homing sequence. Detailed explanations of the homing sequences can be found in the ***Homing The SMD23E-B*** chapter, starting on page 55.
10) Find_Home, Registration, Blend, Dwell, and SynchroStep moves cannot be held. If the Hold command is triggered during one of these moves, the unit will immediately begin to decelerate the move and bring the move to a controlled stop. The move is not put into a hold state.
11) Activating a CW limit will bring a Jog_CW or a CW Registration Move to an immediately stop, and prevent further motion in that direction. Activating a CCW limit will bring a Jog_CCW or a CCW Registration Move to an immediately stop, and prevent further motion in that direction.
12) Activating a CCW limit during a Jog_CW or a CW Registration Move or activating a CW limit during a Jog_CCW or a CCW Registration Move has no effect.
13) An inactive-to-active transition on a Stop Jog or Registration Move input will start to decelerate the move. The move will stop when the configured starting speed is reached. If the encoder is enabled, the transition will also capture the encoder position and copy it to the Captured Encoder Position registers.
14) An inactive-to-active transition on a Stop Jog or Registration Move input will cause the unit to start to output the programmed number of steps at the end move. If the encoder is enabled, the transition will also capture the encoder position and copy it to the Captured Encoder Position registers. Note that this input is ignored until the programmed Minimum Registration Move Distance is exceeded.

A Simple Move

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

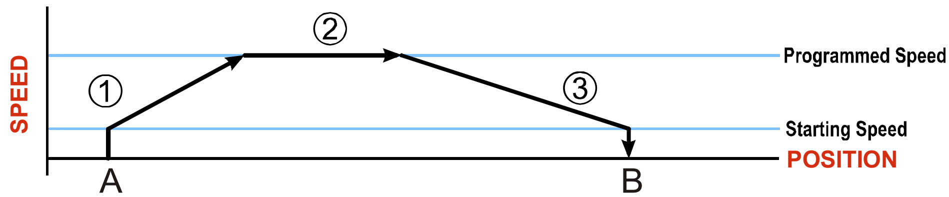


Figure R2.4 A Trapezoidal Profile

1) The move begins at point A, where the motor jumps from rest to the configured *Starting Speed*. The motor then accelerates at the programmed *Acceleration Value* until the speed of the motor reaches the *Programmed Speed*. Both the Acceleration Value and the Programmed Speed are programmed when the move command is sent to the SMD23E-B.
2) The motor continues to run at the Programmed Speed until it reaches the point where it must decelerate before reaching point B.
3) The motor decelerates at the *Deceleration Value*, which is also programmed by the move command, until the speed reaches the Starting Speed, which occurs at the Target Position (B). The motor stops at this point. Note that the acceleration and deceleration values can be different in the move.

Figure R2.4 above shows a Trapezoidal Profile. A Trapezoidal Profile occurs when the Programmed Speed is reached during the move. This occurs when the number of steps needed to accelerate and decelerate are less than the total number of steps in the move.

Figure R2.5 below shows a Triangular Profile. A Triangular Profile occurs when the number of steps needed to accelerate to the Programmed Speed and decelerate from the Programmed Speed are greater than the total number of steps in the move. In this case, the profile will accelerate as far as it can before it has to decelerate to reach the Starting Speed at the Target Position. The Programmed Speed is never reached.

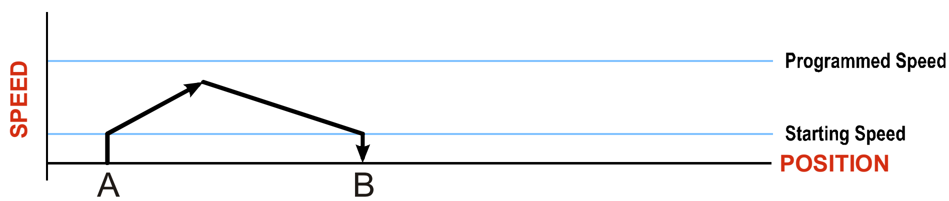


Figure R2.5 A Triangular Profile

Basic Move Types

Absolute Move

Absolute Moves move from the Current Position (A) to a given position (B). (The SMD23E-B 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.

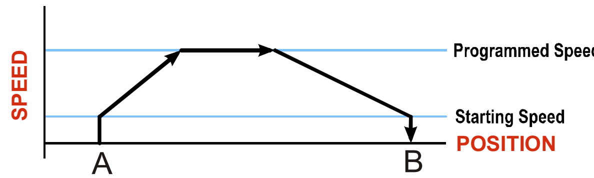


Figure R2.6 Absolute Move

NOTE 1) The Home Position of the machine must be set before running an Absolute Move. (The Position_Invalid module status bit cannot be set to "1".) See chapter 4, ***Homing The SMD23E-B***, which starts on page 55, for information on homing the machine.

2) 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.

Motion Status Bits

The motion status bits function as described in the ***Motion Status Bits*** section on page 25.

Control Inputs

	DC Digital Input Types							Backplane Bits			
	General Purpose	CW L.S.	CCW L.S.	Home L.S.	E-Stop	Stop Jog or Reg	Start Indexed	Indexed_Cmd	Hold Move	Resume Move	Immediate Stop
Absolute Move		1	1		2		3,4	3	5	6	2

Table R2.2 Absolute Move Control Inputs

1) An active limit switch input will immediately stop all motion, and prevent further motion in that direction.
2) An inactive-to-active transition on an Emergency Stop Input, or an active Immediate_Stop command will immediately stop all motion. The Position_Invalid bit will be set if motion was occurring when either of these conditions became true.
3) If the Indexed_Command bit is set when the motion command is written to the unit, the move will not start until there is an inactive-to-active transition on the Start Indexed input.
4) If the encoder is enabled, a transition on the Start Indexed input will capture the encoder position and copy it to the Captured Encoder Position registers.
5) An active Hold command will start to decelerate the move. The move will stop when the configured Starting Speed is reached and the move will enter its Hold State.
6) An active Resume command will resume an absolute or relative move if it is in its hold state. Activating the Resume command when a move is not in its hold state will generate a Command Error response from the unit. If the move is running at this time, the move will continue.

Basic Move Types (continued)

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.

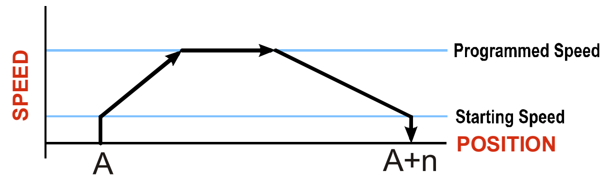


Figure R2.7 Relative Move

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 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.

Motion Status Bits

The motion status bits function as described in the ***Motion Status Bits*** section on page 25.

Control Inputs

	DC Digital Input Types							Backplane Bits			
	General Purpose	CW L.S.	CCW L.S.	Home L.S.	E-Stop	Stop Jog or Reg	Start Indexed	Indexed_Cmd	Hold Move	Resume Move	Immediate Stop
Relative Move		1	1		2		3,4	3	5	6	2

Table R2.3 Relative Move Control Inputs

1) An active limit switch input will immediately stop all motion, and prevent further motion in that direction.
2) An inactive-to-active transition on an Emergency Stop Input, or an active Immediate_Stop command will immediately stop all motion. The Position_Invalid bit will be set if motion was occurring when either of these conditions became true.
3) If the Indexed_Command bit is set when the motion command is written to the unit, the move will not start until there is an inactive-to-active transition on the Start Indexed input.
4) If the encoder is enabled, a transition on the Start Indexed input will capture the encoder position and copy it to the Captured Encoder Position registers.
5) An active Hold command will start to decelerate the move. The move will stop when the configured Starting Speed is reached and the move will enter its Hold State.
6) An active Resume command will resume an absolute or relative move if it is in its hold state. Activating the Resume command when a move is not in its hold state will generate a Command Error response from the unit. If the move is running at this time, the move will continue.

Basic Move Types (continued)

Jog Moves

Jog Moves move in the programmed direction as long as the command is active. Two commands are available. The Jog_CW move will rotate the shaft in a clockwise direction while the Jog_CCW move will rotate the shaft in a counter-clockwise direction. These commands are often used to give the operator manual control over the axis.

Jog Moves are typically used to drive the machine under direct operator control, but they can also be used when you are interested in controlling the speed of the shaft instead of its position. One such application is driving a conveyor belt. In this application you are only interested in driving the conveyor at a specific speed, and you may need to vary the speed based on environmental conditions.

The CW Limit and CCW Limit inputs behave differently for Jog Moves and 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 Jog_CW can be issued while the CCW limit switch is active. This allows you to move off of an activated end limit switch.

As shown below, a Jog 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 SMD23E-B 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 an SMD23E-B that is less than the starting speed, the Jog Move will continue at the previously programmed speed.

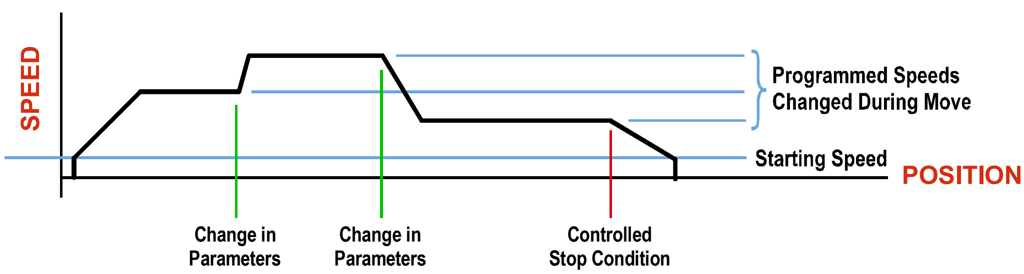


Figure R2.8 Jog Move

Basic Move Types (continued)

Jog Moves (continued)

Motion Status Bits

The motion status bits function as described in the ***Motion Status Bits*** section on page 25.

Control Inputs

	DC Digital Input Types							Backplane Bits			
	General Purpose	CW L.S.	CCW L.S.	Home L.S.	E-Stop	Stop Jog or Reg	Start Indexed	Indexed_Cmd	Hold Move	Resume Move	Immediate Stop
Jog_CW Move		11	12		2	13	3,4	3	5	6	2
Jog_CCW Move		12	11		2	13	3,4	3	5	6	2

Table R2.4 Jog Move Control Inputs

A blank cell means that the state of the input as no effect on the move.

2) An inactive-to-active transition on an Emergency Stop Input, or an active Immediate_Stop command will immediately stop all motion. The Position_Invalid bit will be set if motion was occurring when either of these conditions became true.

3) If the Indexed_Command bit is set when the motion command is written to the unit, the move will not start until there is an inactive-to-active transition on the Start Indexed input.

4) If the encoder is enabled, a transition on the Start Indexed input will capture the encoder position and copy it to the Captured Encoder Position registers.

5) An active Hold command will start to decelerate the move. The move will stop when the configured Starting Speed is reached and the move will enter its Hold State.

6) An active Resume command will resume a move if it is in its hold state. Activating the Resume command when a move is not in its hold state will generate a Command Error response from the unit. If the move is running at this time, the move will continue.

11) Activating a CW limit will bring a Jog_CW or a CW Registration Move to an immediately stop, and prevent further motion in that direction. Activating a CCW limit will bring a Jog_CCW or a CCW Registration Move to an immediately stop, and prevent further motion in that direction.

12) Activating a CCW limit during a Jog_CW or a CW Registration Move or activating a CW limit during a Jog_CCW or a CCW Registration Move has no effect.

13) An inactive-to-active transition on a Stop Jog or Registration Move input will start to decelerate the move. The move will stop when the configured starting speed is reached. If the encoder is enabled, the transition will also capture the encoder position and copy it to the Captured Encoder Position registers.

Basic Move Types (continued)

Registration Moves

Similar to a Jog Move, a Registration Move will travel in the programmed direction as long as the command is active. When the command terminates under Controlled Stop conditions, the SMD23E-B will output a programmed number of steps as part of bringing the move to a stop. Note that all position values programmed with a Registration Move are relative values, not absolute machine positions.

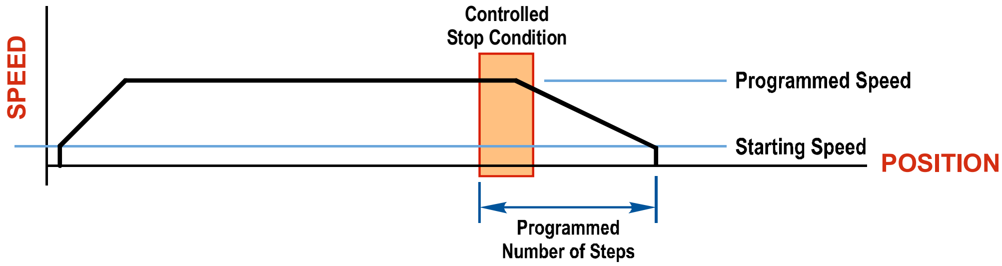


Figure R2.9 Registration Move

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 SMD23E-B 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 SMD23E-B 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 effects when the Stop Condition is applied to the move. As shown in the second diagram, Controlled Stop conditions are level triggered, not edge triggered. If a Controlled Stop Condition occurs before the Minimum Registration Move Distance is reached and stays active, the move will begin its controlled stop once the Minimum Registration Move Distance is reached.

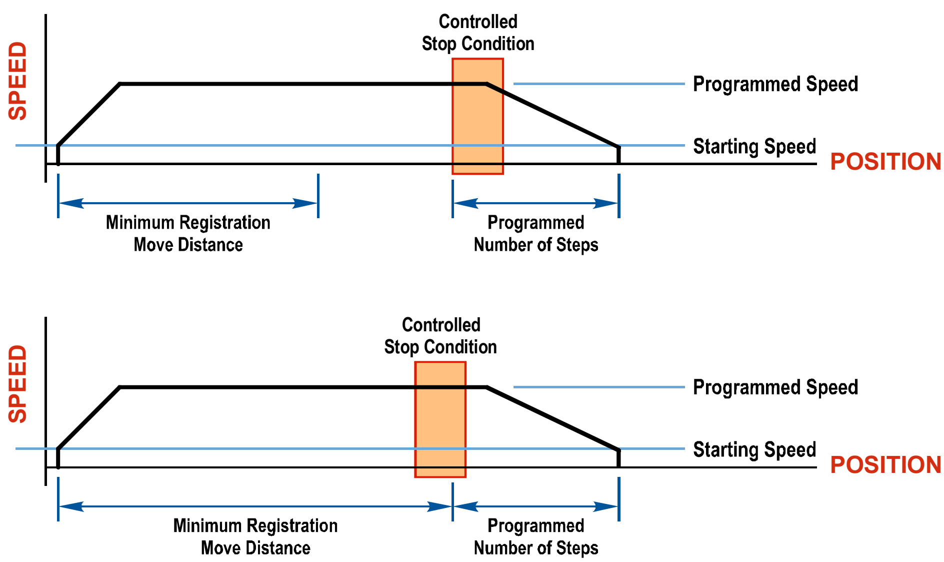


Figure R2.10 Min. Registration Move Distance

Basic Move Types (continued)

±Registration Move (continued)

Motion Status Bits

The motion status bits function as described in the ***Motion Status Bits*** section on page 25.

Control Inputs

	DC Digital Input Types							Backplane Bits			
	General Purpose	CW L.S.	CCW L.S.	Home L.S.	E-Stop	Stop Jog or Reg	Start Indexed	Indexed_Cmd	Hold Move	Resume Move	Immediate Stop
CW Registration Move		11	12		2	14	3,4	3	10		2
CCW Registration Move		12	11		2	14	3,4	3	10		2

Table R2.5 Registration Move Control Inputs

A blank cell means that the state of the input as no effect on the move.

2) An inactive-to-active transition on an Emergency Stop Input, or an active Immediate_Stop command will immediately stop all motion. The Position_Invalid bit will be set if motion was occurring when either of these conditions became true.

3) If the Indexed_Command bit is set when the motion command is written to the unit, the move will not start until there is an inactive-to-active transition on the Start Indexed input.

4) If the encoder is enabled, a transition on the Start Indexed input will capture the encoder position and copy it to the Captured Encoder Position registers.

10) Find_Home, Registration, Blend, Dwell, and SynchroStep moves cannot be held. If the Hold command is triggered during one of these moves, the unit will immediately begin to decelerate the move and bring the move to a controlled stop. The move is not put into a hold state.

11) Activating a CW limit will bring a Jog_CW or a CW Registration Move to an immediately stop, and prevent further motion in that direction. Activating a CCW limit will bring a Jog_CCW or a CCW Registration Move to an immediately stop, and prevent further motion in that direction.

12) Activating a CCW limit during a Jog_CW or a CW Registration Move or activating a CW limit during a Jog_CCW or a CCW Registration Move has no effect.

14) An inactive-to-active transition on a Stop Jog or Registration Move input will cause the unit to start to output the programmed number of steps at the end move. If the encoder is enabled, the transition will also capture the encoder position and copy it to the Captured Encoder Position registers. Note that this input is ignored until the programmed Minimum Registration Move Distance is exceeded.

Basic Move Types (continued)

SynchroStep (Virtual Axis) Moves

The SMD23E-B has the ability to be treated as a motion axis, with the host controller periodically sending position and velocity data to the unit. The loop is closed by the SMD23E-B by controlling the velocity of the motor.

Programming is similar to a Jog Move in that the position, velocity, acceleration, and deceleration data can be changed while the command is active. This is optional in a Jog Move. When the SMD23E-B is used as a motion axis, this data must be updated periodically for accurate control.

SynchroStep Moves have the following characteristics:

- Position and Velocity are programmed as 32 bit double integer values.
- The loop can be closed with respect to the internal motor position or the encoder position.
- 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 SMD23E-B2. 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.

NOTE When using the SMD23E-B 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.

Motion Status Bits

The motion status bits function as described in the ***Motion Status Bits*** section on page 25.

Control Inputs

	DC Digital Input Types							Backplane Bits			
	General Purpose	CW L.S.	CCW L.S.	Home L.S.	E-Stop	Stop Jog or Reg	Start Indexed	Indexed_Cmd	Hold Move	Resume Move	Immediate Stop
SynchroStep Move		1	1		2				10		2

Table R2.6 SynchroStep Move Control Inputs

A blank cell means that the state of the input as no effect on the move.

1) An active limit switch input will immediately stop all motion, and prevent further motion in that direction.
2) An inactive-to-active transition on an Emergency Stop Input, or an active Immediate_Stop command will immediately stop all motion. The Position_Invalid bit will be set if motion was occurring when either of these conditions became true.
10) Find_Home, Registration, Blend, Dwell, and SynchroStep moves cannot be held. If the Hold command is triggered during one of these moves, the unit will immediately begin to decelerate the move and bring the move to a controlled stop. The move is not put into a hold state.

Assembled Moves

Assembled Moves are an option when the move profile is well defined before the move begins. Consider using this functionality for simple profiles. An Assembled Move is programmed into the SMD23E-B before it is run and the unit precisely controls position and velocity throughout the entire move.

All of the moves explained so far must be run individually to their completion or must be stopped before another move can begin. The SMD23E-B 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 an SMD23E-B:

- **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 SMD23E-B 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.

Blend Moves

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 SMD23E-B before the move can be initiated.
    - The SMD23E-B 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 Moves*** which is explained on page 38.
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.

Assembled Moves (continued)

Blend Moves (continued)

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.

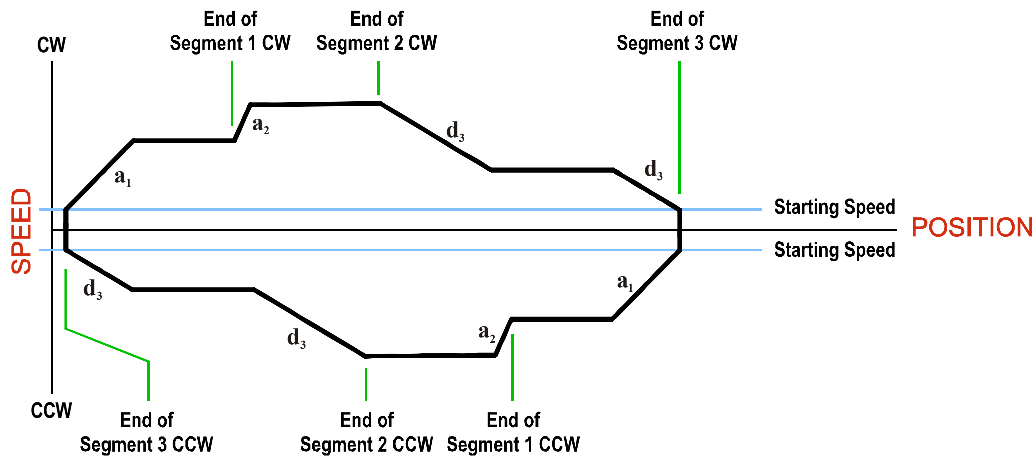


Figure R2.11 Blend Move

NOTE

1) You do not have to preset the position or home the machine before you can use a Blend Move. Because the Blend Move is based on Relative Moves, it can be run from any location.
2) The Blend Move is stored in the internal memory of the SMD23E-B 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 41, 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 SMD23E-B 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.

Dwell Moves

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 an SMD23E-B 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. 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.

Assembled Moves (continued)

Dwell Moves (continued)

You can accomplish this Dwell Move with a series of six relative moves that are sent down to the SMD23E-B sequentially. The two advantages of a Dwell Move in this case are that the SMD23E-B will be more accurate with the Dwell Time then you can be in your control program, and Dwell Moves simplify your program's logic.

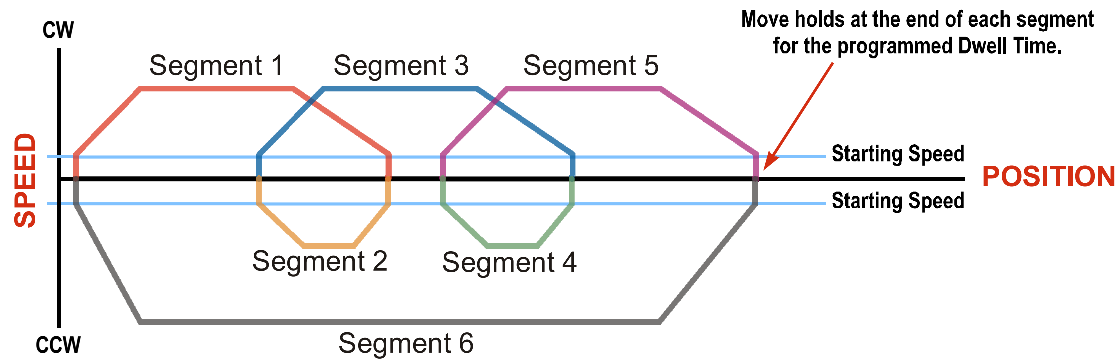


Figure R2.12 Dwell Move

NOTE

1) You do not have to preset the position or home the machine before you can use a Dwell Move. Because the Dwell Move is based on Relative Moves, it can be run from any location.

2) The Dwell Move is stored in the internal memory of an SMD23E-B 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 SMD23E-B. As described in ***Saving an Assembled Move in Flash*** on page 41, 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 SMD23E-B and enter Command Mode.

Motion Status Bits

The motion status bits function as described in the ***Motion Status Bits*** section on page 25.

Control Inputs

	DC Digital Input Types							Backplane Bits			
	General Purpose	CW L.S.	CCW L.S.	Home L.S.	E-Stop	Stop Jog or Reg	Start Indexed	Indexed_Cmd	Hold Move	Resume Move	Immediate Stop
CW Blend Move		1	1		2		3,4	3	10		2
CCW Blend Move		1	1		2		3,4	3	10		2
Dwell Move		1	1		2		3,4	3	10		2

Table R2.7 Blend and Dwell Control Inputs

See the numbered notes on the following page.
A blank cell means that the state of the input as no effect on the move.

Assembled Moves (continued)

Control Inputs (continued)

1) An active limit switch input will immediately stop all motion, and prevent further motion in that direction.
2) An inactive-to-active transition on an Emergency Stop Input, or an active Immediate_Stop command will immediately stop all motion. The Position_Invalid bit will be set if motion was occurring when either of these conditions became true.
3) If the Indexed_Command bit is set when the motion command is written to the unit, the move will not start until there is an inactive-to-active transition on the Start Indexed input.
4) If the encoder is enabled, a transition on the Start Indexed input will capture the encoder position and copy it to the Captured Encoder Position registers.
10) Find_Home, Registration, Blend, Dwell, and SynchroStep moves cannot be held. If the Hold command is triggered during one of these moves, the unit will immediately begin to decelerate the move and bring the move to a controlled stop. The move is not put into a hold state.

Assembled Move Programming

All of the segments in a Blend or Dwell Move must be written to the SMD23E-B 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 SMD23E-B as part of the command.

Control Bits – Output Data

- **Program_Assembled bit –** Set this bit to tell the SMD23E-B that you want to program a Blend or Dwell Move Profile. The SMD23E-B will respond by setting the *In_Assembled_Mode* bit in the Network Input Data. At the beginning of the programming cycle, the SMD23E-B will also set the *Wait_For_Assembled_Segment* bit to signify that it is ready for the first segment.
- **Read_Assembled_Data bit –** Set this bit to tell the SMD23E-B that the data for the next segment is available in the remaining data words.

Control Bits – Input Data

- **In_Assembled_Mode bit –** The SMD23E-B 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 *Read_Assembled_Data* and *Wait_For_Assembled_Segment* bits.
- **Wait_For_Assembled_Segment bit –** The SMD23E-B will set this bit to signal the host that it is ready to accept the data for the next segment.

Assembled Moves (continued)

Assembled Move Programming (continued)

Programming Routine

1) The host sets the *Program_Assembled* bit in the Network Output Data.
2) The SMD23E-B responds by setting both the *In_Assembled_Mode* and *Wait_For_Assembled_Segment* bits in the Network Input Data.
3) When the host detects that the *Wait_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 SMD23E-B checks the data, and when finished, resets the *Wait_For_Assembled_Segment* bit. If an error is detected, it also sets the *Command_Error* bit.
5) When the host detects that the *Wait_For_Assembled_Segment* bit is reset, it resets the *Read_Assembled_Data* bit.
6) The SMD23E-B detects that the *Read_Assembled_Data* bit is reset, and sets the *Wait_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 16.
8) After the last segment has been transferred, the host exits Assembled Move Programming Mode by resetting the *Program_Assembled* bit.
9) The SMD23E-B resets the *In_Assembled_Mode* bit and the *Wait_For_Assembled_Segment* bit.

Saving an Assembled Move in Flash

The SMD23E-B also contains the *Save_To_Flash* bit that allows you to store the Assembled Move in flash memory. This allows you to run the Assembled Move right after power up, without having to go through a programming sequence first. To use this bit, you follow the above programming routine with the *Save_To_Flash* bit set. When you reach step 9 in the sequence, the SMD23E-B responds by resetting the *In_Assembled_Mode* and *Wait_For_Assembled_Segment* bits as usual and then it will flash the Status LED. If the LED is flashing green, the write to flash memory was successful. If it flashes red, then there was an error in writing the data. In either case, power must be cycled to the SMD23E-B before you can continue. With a limit of 10,000 write cycles, the design decision that requires you to cycle power to the SMD23E-B 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 SMD23E-B 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 SMD23E-B 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 Ox0000 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 Manual 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.

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, which results in an exponential, not linear, drop in motor temperature. This means that even a small reduction in the idle current can have a significant effect on the temperature of the motor.

NOTE Note that the reduction values are "to" values, not "by" values. Setting a motor current to 2Arms and the current reduction to 25% will result in an idle current of 0.5Apk. (The SMD23E-B always switches from RMS to peak current control when the motor is idle to prevent motor damage due to excessive heating.)

Stall Detection with SMD23E-B Units

When Stall Detection is enabled, the SMD23E-B 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 Pulses/Turn* parameter to 1,024 in the Configuration Data of the SMD23E-B.

When Stall Detection is enabled, the unit monitors the encoder for position changes, regardless of whether or not a move is in progress. If the error between the encoder position and the motor position exceeds forty-five degrees, the SMD23E-B responds in the following manner:

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

Note that a move does not have to be in progress for stall detection to be useful. By using Idle Current Reduction, it is possible to remove motor current when a move is not in progress. By enabling stall detection, the unit can notify the system if the motor shaft moves more than forty-five degrees while power is removed from the motor.

Notes

REFERENCE 3

CALCULATING MOVE PROFILES

This reference was added for customers that must program very precise profiles. Understanding this section is not necessary before programming the SMD23E-B and therefore can be considered optional. Two different approaches are presented here. The constant acceleration example takes given parameters and calculates the resulting profile. The variable acceleration example starts with a desired speed profile and calculates the required parameters.

The equations in this reference use a unit of measure of steps/second/second (steps/second^2) for acceleration and deceleration. However, when programming the SMD23E-B, 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 SMD23E-B.
- 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 SMD23E-B 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. ($141^2 \approx 20{,}000$)

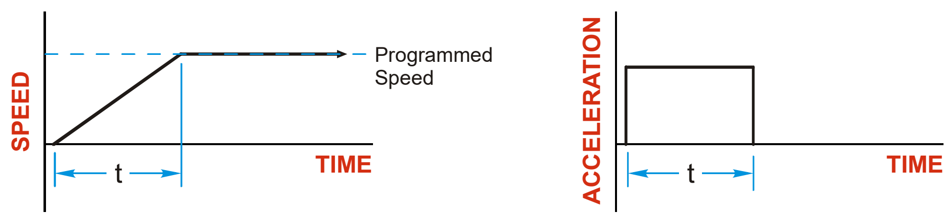
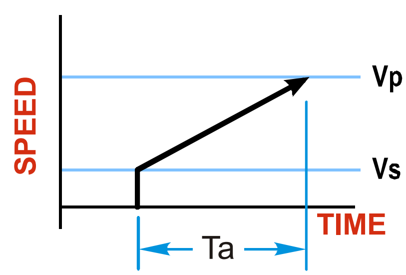


Figure R3.1 Constant Acceleration Curves

Variable Definitions

The following variables are used in these equations:



- V_S = Configured Starting Speed of the move
- V_P = Programmed Speed of the move
- **a** = Acceleration value. Must be in the units of 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 R3.1 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 R3.1 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 R3.2 for a move profile.

Name	Value	SMD23E-B 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 R3.2 Sample Values

From figure R3.1:

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

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

If a move with the above acceleration, deceleration, starting speed, and programmed speed has a length greater than 449,984 steps, the SMD23E-B will generate a Trapezoidal profile. If the move is equal to 449,984 steps, the unit will generate a Triangular profile and the it will output one pulse at the programmed speed. If the move is less than 449,984 steps, the unit will generate a Triangular profile and the programmed speed will not be reached.

In the case of a Triangular profile where the programmed speed is not reached, it is fairly easy to calculate the maximum speed (V_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 R3.2, assume a total travel distance of 300,000 steps.

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

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

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

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

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

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

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

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

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

Total Time Equations

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

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

$T_P = T_A + T_D + D_P/V_P$

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

$T_P = T_A + T_D$

S-Curve Acceleration Equations

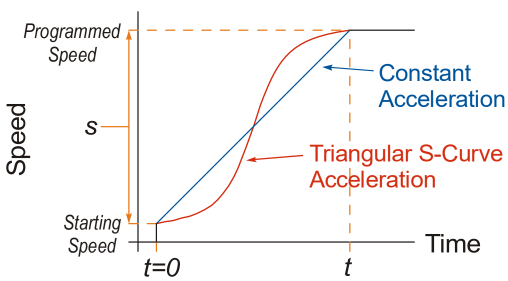
When the Acceleration Jerk parameter value is in the range of 1 to 5,000, the SMD23E-B uses this value to smoothly change the acceleration value applied during the move. In this case, the speed of the move does not increase linearly, but exponentially, resulting in an "S" shaped curve. This limits mechanical 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 SMD23E-B 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 equaling ({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 R3.2 shows the speed profile of a move during its acceleration phase. The figure shows the desired triangular S-curve acceleration in red along with the equivalent constant acceleration in blue. The equivalent constant acceleration is equal to the change in speed divided by the time it takes to achieve this change in speed. This is the value that would have to be used if the Jerk parameter was left at zero and we will use this information to calculate the S-curve acceleration and the value of the Jerk Parameter.



s = Programmed Speed – Starting Speed

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

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

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

Figure R3.2 Move Profile Example

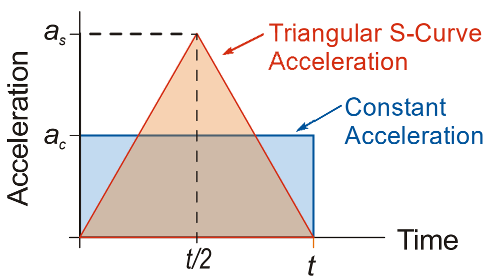


Figure R3.3 Triangular Acceleration

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

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

$$a_s = 2a_c$$

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 $a_s = a_c$

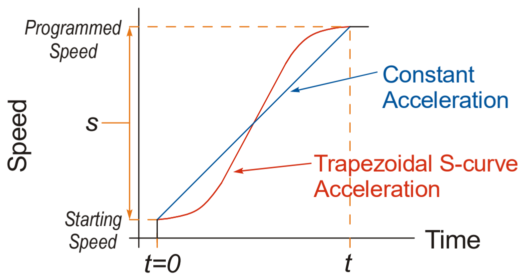
The above examples assume that you can increase the programmed acceleration value to keep the acceleration time the same. If your constant acceleration value is the maximum your system will allow, then using S-curve accelerations will lengthen the time needed to accelerate to your desired speed.

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



S = Programmed Speed – Starting Speed

$$\text{Acceleration} = \frac{\text{speed}}{\text{time}} \qquad a = \frac{s}{t} \qquad at = s$$

$$\text{jerk} = \frac{\text{acceleration}}{\text{time}} \qquad j = \frac{a}{t} \qquad jt = a$$

$$\text{SMD Acceleration Jerk Parameter (J)} = \frac{100j}{a} \implies j = \frac{Ja}{100}$$

Figure R3.4 Move Profile Example

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

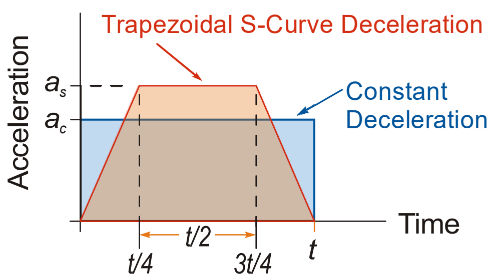


Figure R3.5 Trapezoidal Acceleration

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

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

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

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

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

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.

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

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

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

$$Ja_s t = 400a_s$$

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

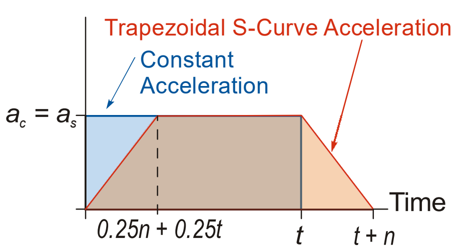
This value represents the ideal Acceleration Jerk parameter value for a trapezoidal S-curve acceleration with a constant acceleration for half of the phase. Setting the value lower than this will result in a shorter constant period, while setting the value greater than this will result in a longer constant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

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

When $a_s = a_c$

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

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



$$a_c(t) = a_c(.5n + .5t) + a_c(.25n + .25t)$$
$$a_c(t) = a_c((.5n + .5t) + (.25n + .25t))$$
$$t = .75n + .75t$$
$$0.25t = .75n$$
$$t = 3n$$
$$t/3 = n \Rightarrow t+n = t + t/3 = 4/3t = 1.3333t$$

Figure R3.6 Trapezoidal S-curve Time Increase Example

In this case the value of the Acceleration Jerk parameter should be based on the new, longer time. For example, assume an equivalent constant acceleration of 15,000 steps/sec^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

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

S_m = midpoint of change in speed

J = Acceleration Jerk parameter

j = physical jerk property

a_f = calculated final acceleration

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

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

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

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 R3.7, two additional calculations must be made. The first is the time (t_1) it takes to jerk to the programmed acceleration value. The second is the time (t_2) it takes to accelerate to half of the required change in speed (S_m).

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

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

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

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

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

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

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

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

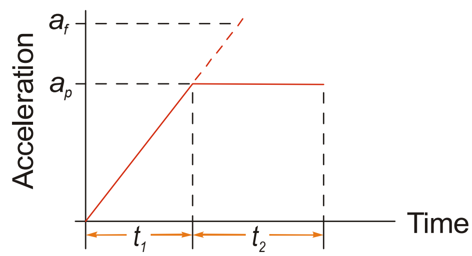


Figure R3.7 Calculating Trapezoidal S-Curve

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 4

HOMING THE SMD23E-B

This chapter explains the various ways of homing the SMD23E-B. 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 an SMD23E-B must be set to an appropriate value. If you use the unit's 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 quadrature encoder is enabled.

NOTE Defining a Home Position is completely optional. Some applications, such as those that use the SMD23E-B for speed control, don't require position data at all.

The SMD23E-B has one position status bit, the Position_Invalid bit. This bit is set when the reported motor position may not correspond to the actual machine position. The reported motor position is a bi-directional counter that tracks the number, and direction, of the motor steps. The actual machine position is the physical position of the axis on the machine.

With the exception of Absolute Moves, an SMD23E-B can perform any of its move commands while the Position_Invalid bit is set.

Position Preset

One of the ways to define the Home Position is to issue the Preset Position command to the SMD23E-B. Before doing this, your host controller will need a way of sensing position. The machine position data must be brought into the host, the correct preset value calculated, and this value written to the SMD23E-B with the Position Preset command. There is a separate Preset_Encoder command to preset the encoder position. The motor and encoder position values can be preset anywhere in the range of –8,388,607 to +8,388,607.

Find_Home Commands

The other choice is to use the module's Find_Home commands to order the SMD23E-B to find the Home Position based on sensors brought into the unit. The Find_Home_CW command begins searching by rotating the motor shaft clockwise and ends when the home sensor triggers while the SMD23E-B is rotating clockwise at the starting speed. The Find_Home_CCW command operates in the same way but starts and ends with counter-clockwise rotation.

Homing Inputs

Four inputs can be used when homing the module. These inputs are either physical inputs attached to the module 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

- **Backplane_Proximity_Bit:** The SMD23E-B can be configured to ignore changes on the physical homing input until the Backplane_Proximity_Bit makes a 0→1 transition. The SMD23E-B 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

An SMD23E-B must have one of its DC inputs configured as the home input before one of the Find_Home commands can be issued.

NOTE You do not have to configure and use CW or CCW Limits. If you choose to configure the module this way, then the SMD23E-B has no way to automatically prevent overtravel during a homing operation. In this case you must do one of the following.

- Prevent overtravel by some external means
- Ensure that the homing command is issued in the direction that will result in reaching the homing input directly
- Enable the encoder and the stall detection option

Homing Profiles

NOTE The Find_Home_CW command is used in all of these examples. The Find_Home_CCW command will generate the same profiles in the opposite direction.

Home Input Only Profile

Figure R4.1 below shows the move profile generated by a Find_Home_CW command when you use the Home Input without the Backplane_Proximity_Bit.

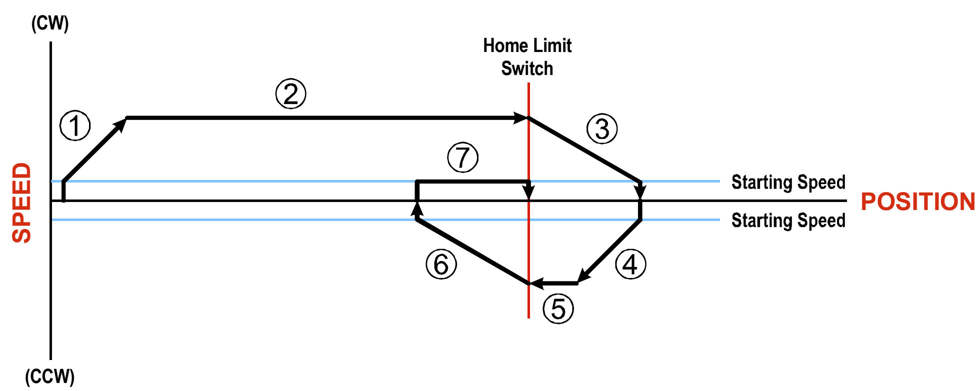


Figure R4.1 Home Input Profile

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

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

Homing Profiles (continued)

Profile with Network Home Proximity Bit

Figure R4.2 below shows the move profile generated by a Find_Home_CW command when you use the Home Input with the Backplane_Proximity_Bit in the network output registers.

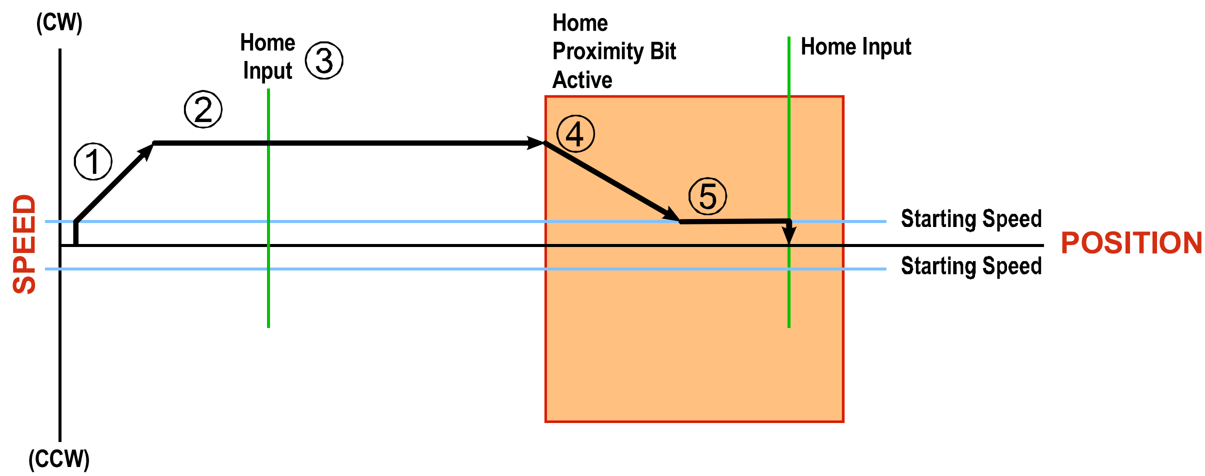


Figure R4.2 Homing with Proximity

1) Acceleration from the configured Starting Speed to the Programmed Speed
2) Run at the Programmed Speed
3) Ignores the Home Input because Backplane_Proximity_Bit has not made a 0→1 transition.
4) Deceleration towards the Starting Speed when the Backplane_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 R4.2 shows the Backplane_Proximity_Bit staying active until the SMD23E-B reaches its home position. This is valid, but does not have to occur. As stated in step 4, the SMD23E-B starts to hunt for the home position as soon and the Backplane_Proximity_Bit makes a 0→1 transition

Homing Profiles (continued)

Profile with Overtravel Limit

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

- CW Overtravel Limit
- Home Input without using the Backplane_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 will set the Input_Error bit in the network input data.

The SMD23E-B will stop and set the Input_Error bit if both overtravel limits are activated while the unit is trying to find the home position.

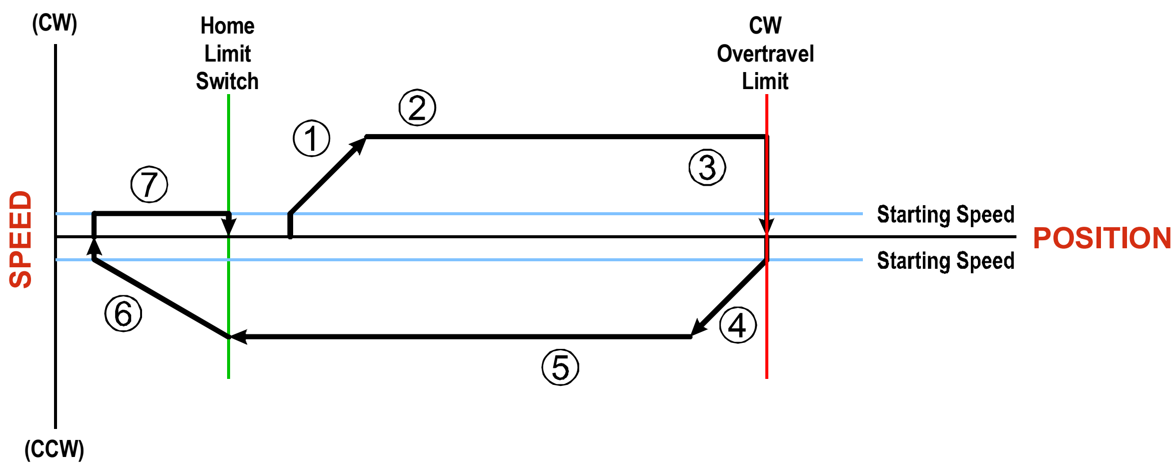


Figure R4.3 Profile with Overtravel Limit

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

NOTE If the overtravel limit is active when the Find_Home command is active, the profile will begin at step 4.

REFERENCE 5

Configuration Mode Data Format

This chapter covers the formats of the Network Output Data used to configure an SMD23E-B as well as the formats of the Network Input Data that contains the configuration responses from the device.

Modes of Operation

An SMD23E-B has two operating modes, Configuration Mode and Command Mode. You switch between these modes by changing the state of a single bit in the Network Output Data.

Configuration Mode

Configuration Mode gives you the ability to select the proper configuration for your application without having to set any switches. The ladder logic needed to configure a unit is included in the sample programs available from AMCI. This method simplifies change over if the unit ever needs to be replaced.

A valid configuration can be saved to the unit's Flash memory and the unit will use this as a default configuration on every power up. If you use this method, you can still write down a new configuration to the unit at any time. The new configuration is stored in RAM and is lost on power down unless you issue a command to store the new configuration in Flash.

Command Mode

This mode gives you the ability to program and execute stepper moves, and reset errors when they occur. The SMD23E-B units will always power up in this mode. The command data formats are described in the following chapter.

NOTE The SMD23E-B will not accept move commands, and the motor will not receive power, until the unit has a valid configuration.

Power Up Behavior

An SMD23E-B will always power up in Command Mode. If available, the unit will use its stored configuration data to configure itself. The SMD23E-B will then check for valid network command data and will only enable the motor driver section if the Enable_Driver bit is set.

Configuration Mode Multi-Word Data Format

An SMD23-B requires twenty bytes of Output Data as well as twenty bytes of Input Data. Many of the hosts that can be used with the SMD23E-B only support sixteen bit integers. Sixteen bit integers support a range of values from -32,768 to 32,767 or 0 to 65,535. The Starting Speed parameter, which is programmed as part of the configuration data, can exceed this range. This parameter is transmitted in two separate words. The table below shows how values are split.

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

Table R5.1 Multi-Word Format Examples

Command Mode Data Formats

When issuing commands to the SMD23E-B, there are several parameters that are larger than sixteen bits. These parameters are:

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

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

- Motor Position
- Encoder Position
- Captured Encoder Position

By default, these thirty-two bit parameters and data values are written to and read from the SMD23E-B using the multi-word format described above. When configuring the unit, it is possible to program it to use a 32-bit double integer format instead of the custom format shown above.

There are two separate programming bits. The Binary_Output_Format Bit, controls the format of the programmable parameters written to the unit when issuing commands. The Binary_Input_Format Bit, controls the format of the data values written to the host controller by the SMD23E-B.

When using the signed thirty-two bit format, there is an additional parameter named Binary_Endian. Rockwell PLCs use little endian format. Modbus processors typically use big endian format, but you should refer to your PLC's documentation to verify the format used by your processor.

Examples of the formats are given below.

	Multi-Word Format		32 bit Signed Integer Little Endian Format		32 bit Signed Integer Big Endian Format	
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 R5.2 Position Data Format Examples

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

Output Data Format

The correct format for the Network Output Data when the SMD23E-B is in Configuration Mode is shown below. EtherNet/IP and Modbus TCP addresses are also shown.

EtherNet/IP Word	Modbus TCP Register	Configuration Data	Range	Default
0	1024	Configuration Word 0	See below	See below
1	1025	Configuration Word 1	See below	See below
2	1026	Starting Speed: Upper Word	Combined value between 1 and 1,999,999 steps/sec.	50
3	1027	Starting Speed: Lower Word		
4	1028	Motor Steps/Turn	200 to 32,767	1024
5	1029	Reserved	Set to zero	0
6	1030	Encoder Pulses/Turn	Set to 1,024	1024
7	1031	Idle Current Percentage	0 to 100%	20
8	1032	Motor Current (X10)	10 to 34, Represents 1.0 to 3.4 Arms	28
9	1033	Reserved	Set to zero	0

Table R5.3 Network Output Data Format: Configuration Mode

Configuration Word 0 Format

Configuration Word 0

15	14	13	12	11	10	09	08 07 06	05 04 03	02 01 00
MODE	Dis_AR	En_Stall	0	Use_BProx	Use_Enc	0	Input 3 Function	Input 2 Function	Input 1 Function

RESERVED: Bit must equal zero.

Figure R5.1 Configuration Mode: Control Word Format

Bit 15: **Mode Bit –** "1" for Configuration Mode Programming, "0" for Command Mode Programming. An SMD23E-B powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), unless a valid configuration has been written to the flash memory of the SMD23E-B. The SMD23E-B will not power the motor or accept commands until a valid configuration is written to it.

Bit 14: **Disable_Anti-Resonance Bit –** "0" enables the anti-resonance feature of the SMD23E-B. "1" disables the anti-resonance feature. The Anti-resonance feature will provide smoother operation in most cases. If you are still experiencing resonance problems with this feature enabled, disable this feature and test the machine again.

Bit 13: **Enable_Stall_Detection Bit –** "0" disables motor stall detection. "1" enables motor stall detection. The encoder pulses/turn must be programmed to a value of 1,024 and the encoder must be enabled by setting bit 10 to "1". (See Bit 10 below.)

Bit 11: **Use_Backplane_Proximity Bit –** "0" when Home Proximity bit is not used when homing the SMD23E-B. "1" when the Home Proximity bit is used when homing the SMD23E-B. Note that this bit is not the Home Proximity Bit, but enables or disables its operation. Do not use the Home Proximity bit if you only want to home to the Home Limit Switch. (Leave this bit equal to "0".)

Bit 10: **Use_Encoder Bit –** "0" when Quadrature Encoder is not used. "1" to enable a Quadrature Encoder.

Configuration Mode - Output Data Format (continued)

Configuration Word 0 Format (continued)

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

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

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

Bits				
8	7	6		
5	4	3		
2	1	0	Function	Available On
0	0	0	General Purpose Input	The input is not used in any of the functions of the SMD23E-B, but it's status is reported in the Network Data. This allows the input to be used as a discrete DC input to the host controller.
0	0	1	CW Limit	Input defines the mechanical end point for CW motion.
0	1	0	CCW Limit	Input defines the mechanical end point for CCW motion.
0	1	1	Start Indexed Move	Starts the move that is currently located in the output registers.
0	1	1	Start Indexed Move / Capture Encoder Value	When the encoder is enabled on an SMD23E-B/E, the encoder position value is captured whenever this input transitions. An inactive-to-active state transition will also trigger an Indexed Move if one is pending in the SMD23E-B/E.
1	0	0	Stop Jog or Registration Move	Brings a Jog or Registration Move to a controlled stop.
1	0	0	Stop Jog or Registration Move & Capture Encoder Value	When the encoder is enabled on an SMD23E-B/E, the encoder position value is captured when the input triggers a controlled stop to a Manual or Registration move.
1	0	1	Emergency Stop	All motion is immediately stopped when this input makes an inactive-to-active transition.
1	1	0	Home	Used to define the home position of the machine.
1	1	1	Invalid Combination	This bit combination is reserved.

Table R5.4 Configuration Data: Input Function Selections

Configuration Mode - 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	0	0	0	ReadConfig	SaveConfig	BinInFrmt	BinOutFrmt	DataEndian	0	0	0	0	ActSt-3	ActSt-2	ActSt-1

RESERVED: Bit must equal zero.
RESERVED: Bit value is ignored .

Figure R5.2 Configuration Mode: Config Word Format

Bit 15: Reserved – State Ignored.

Bits 14 - 12: Reserved – Must equal zero.

Bit 11: Read_Present_Configuration Bit – If this bit is set when you enter Configuration Mode, the SMD23E-B 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 64.

Bit 10: Save_Present_Configuration Bit – An SMD23E-B 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 unit will write 16#AAAA into the last word of the Network Input Data and the Module Status LED will start flashing green. If the write is unsuccessful, the unit will write 16#EEEE into the last word of the Network Input Data and the Module Status LED will start flashing red. Once the unit 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 SMD23E-B 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 an SMD23E-B 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 one out on the machine.
2) The endurance of the flash memory is a minimum of 10,000 write cycles.

Bit 9: Binary_Input_Format Bit – Set to "0" to have the Motor Position, Encoder Position, and Trapped Encoder Position reported in the multi-word format. Set to "1" to have the Motor Position, Encoder Position, and Trapped Encoder Position reported in signed 32-bit integer format. When this parameter is set to "1", the Binary_Endian parameter in bit 7 sets the word order of the 32-bit value. See ***Position Data Format Examples*** found on page 60 for examples of the different formats.

Bit 8: Binary_Output_Format Bit – Set to "0" to program the multi-word parameters in the multi-word format. Set to "1" to program the multi-word parameters in signed 32-bit integer format. When this parameter is set to "1", the Binary_Endian parameter in bit 7 sets the word order of the 32-bit value. See ***Position Data Format Examples*** found on page 60 for examples of the different formats.

Bit 7: Binary_Endian Bit – Only used when bits 8 and/or 9 above are set to "1", set to "0" to program the 32-bit values in little endian format. Set to "1" to program the 32-bit values in big endian format. Rockwell processors use little endian format. See ***Position Data Format Examples*** found on page 60 for examples of the different formats.

Bits 6 - 3: Reserved – Must equal zero.

Configuration Mode - Output Data Format (continued)

Configuration Word 1 Format (continued)

Bit 2: **Input3_Active_Level Bit –** Determines the active state of Input 3. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Bit 1: **Input2_Active_Level Bit –** Determines the active state of Input 2. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Bit 0: **Input1_Active_Level Bit –** Determines the active state of Input 1. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Notes on Other Configuration Words

- Information on the ***Configuration Mode Multi-Word Data Format*** used when programming the Starting Speed can be found on page 59.
- Changes to the Idle Current only take effect at the *end of the first move after re-configuration*.

Input Data Format

The format for the Network Input Data when an SMD23-B is in Configuration Mode is shown below. EtherNet/IP and Modbus TCP addresses are both shown.

EtherNet/IP 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: Upper Word
3	3	Mirror of Starting Speed: Lower Word
4	4	Mirror of Motor Steps/Turn
5	5	0000
6	6	Mirror of Encoder Pulses/Turn
7	7	Mirror of Idle Current Percentage
8	8	Mirror of Motor Current (X10)
9	9	0000 or Status message when writing Configuration data to flash memory.

Table R5.5 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 SMD23E-B. When the data written to it is invalid, the unit remains in Command Mode and sets the Configuration Error bit in the first word written back to the host controller. The format of this word is explained in the ***Status Word 0 Format*** section starting on page 82. On power up, the value of this word will be 6408h unless a valid configuration had been previously written to the flash memory of the unit.

Starting Speed Format

The Starting Speed parameter is always programmed using the ***Configuration Mode Multi-Word Data Format*** as described on page 59.

Input Data Format (continued)

Stall Detect Enable

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 based on the mode of the SMD23E-B.

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 when the output data format is set to its default AMCI multi-word setting.
3) Configuring the two inputs to have the same function, such as two CW Limit Switches. (An error does not occur if both are configured as General Purpose Inputs.)
4) Setting the *Stall Detection Enable Bit* without configuring the SMD23E-B to use its built in encoder.
5) Setting the Input Configuration bits for any input to “111”. See table R5.4 on page 62 for more information.

Notes

COMMAND MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to command the SMD23E-B as well as the formats of the Network Input Data that contains the responses from the unit. An SMD23E-B requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data.

Power Up Behavior

An SMD23E-B will always power up in Command Mode. If available, the unit will use its stored configuration data to configure itself. The SMD23E-B will then check for valid network command data and will only enable the motor driver section if the Enable_Driver bit is set.

Data Format

An SMD23E-B requires twenty bytes of Output Data as well as twenty bytes of Input Data. In most cases the data is represented as ten 16-bit (single) integers. Sixteen bit integers support a range of values from -32,768 to 32,767 or 0 to 65,535. When issuing commands to the SMD23E-B, there are several parameters that are larger than sixteen bits. These parameters are:

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

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

- Motor Position
- Encoder Position
- Captured Encoder Position

By default, these thirty-two bit parameters and data values are written to and read from the SMD23E-B using the multi-word format described below. When configuring the SMD23E-B, it is possible to program it to use a 32-bit double integer format instead of the custom format shown below.

There are two separate programming bits. The Binary_Output_Format Bit, controls the format of the programmable parameters written to the unit when issuing commands. The Binary_Input_Format Bit, controls the format of the data values written to the host controller by the SMD23E-B.

When either of these parameters are set to their 32-bit signed integer format settings, the Binary_Endian bit determines if the 32-bit values are stored and transmitted least significant bits first (little endian) or most significant bits first (big endian). Rockwell PLCs use little endian format. Modbus processors typically use big endian format, but you should refer to your PLC's documentation to verify the format used by your processor. Examples of the formats are given below.

	Multi-Word Format		32 bit Signed Integer Little Endian Format		32 bit Signed Integer Big Endian Format	
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 R6.1 Position Data Format Examples

Data Format (continued)

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

Command Bits Must Transition

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

This condition also applies when switching from Configuration Mode to Command Mode. If a bit is set in the Configuration Word 0 (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 SMD23E-B. EtherNet/IP and Modbus TCP addresses are both shown.

EtherNet/IP 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 SMD23E-B
3	1027	
4	1028	
5	1029	
6	1030	
7	1031	
8	1032	
9	1033	

Table R6.2 Command Mode Data Format

Command Word 0 Bits

Command Word 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MODE	PrstEnc	RunAsMv	ReadAsData	PrgAMv	Reset_Err	Pos_Prst	Jog_CCW	Jog_CW	Home_CCW	Home_CW	Imd_Stop	Resm_Mv	Hold_Mv	Rel_Mv	Abs_Mv

Figure R6.1 Command Bits MSW Format

Bit 15: **Mode Bit –** "1" for Configuration Mode Programming "0" for Command Mode Programming. The SMD23E-B powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), if a valid configuration has not been stored in flash memory. An SMD23E-B will not power the motor or accept commands until a valid configuration is written to it or read from flash memory.

Bit 14: **Preset Encoder Bit –** When set to "1" the SMD23E-B will preset the Encoder Position to the value stored in Output Words 2 and 3. A command error will occur if the encoder is not enabled in the unit's configuration data.

Bit 13: **Run Assembled Move –** When set to "1" an SMD23E-B 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. This bit is ignored while running a Dwell Move.

Bits 11 & 12: Program_Assembled & Read_Assembled_Data Bits – 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 40.

Bit 10: **Reset_Errors –** When set to "1" the SMD23E-B will clear command, input, and stall detect errors. This command will also reset the Move_Complete status bit. The Reset_Errors command *will not* reset the Position_Invalid status bit.

Bit 9: **Preset_Position Bit –** When set to "1" the SMD23E-B will preset the Motor Position to the value stored in Output Words 2 and 3. This command also resets the Move_Complete bit.

Bit 8: **Jog_CCW Move –** When set to "1" the SMD23E-B will run a Jog Move in the counter-clockwise direction. A full explanation of a ***Jog Moves*** can be found starting on page 32.

- **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.

Bit 7: **Jog_CW Move –** When set to "1" the SMD23E-B will run a Jog Move in the clockwise direction. A full explanation of a ***Jog Moves*** can be found starting on page 32.

- **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.

Bit 6: **Find_Home_CCW Move –** When set to "1" the SMD23E-B will attempt to move to the Home Limit Switch in the counter-clockwise direction. A full explanation of homing can be found in the ***Homing The SMD23E-B*** chapter starting on page 55.

Command Word 0 Bits (continued)

Bit 5: Find_Home_CW Move – When set to “1” the SMD23E-B will attempt to move to the Home Limit Switch in the clockwise direction. A full explanation of homing can be found in the ***Homing The SMD23E-B*** chapter starting on page 55.

Bit 4: Immediate_Stop Bit – When set to “1” the SMD23E-B will stop all motion without deceleration. The Motor Position value will become invalid when this bit is set, even if no motion is occurring when it is set.

Bit 3: Resume_Move Bit – 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 for each move type in the section ***How to Control Moves in Progress*** starting on page 27. Note that a move in its hold state does not need to be resumed. The move is automatically canceled if another move is started in its place.

Bit 2: Hold_Move Bit – 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 or it can be aborted by simply starting another move. Use of the Hold_Move and Resume_Move bits can be found for each move type in the section ***How to Control Moves in Progress*** starting on page 27.

Bit 1: Relative_Move Bit – 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 31.

Bit 0: Absolute_Move Bit – 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 Bits

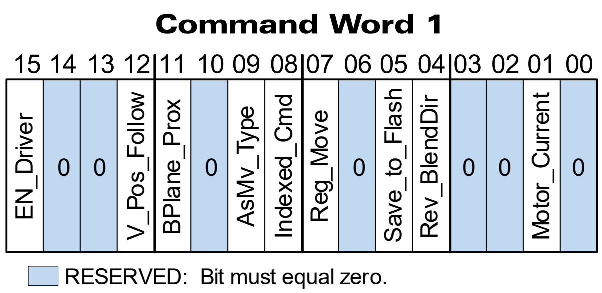


Figure R6.2 Command Bits LSW Format

Bit 15: Enable_Driver Bit – “0” to disable the motor current, “1” to enable motor current. A valid configuration must be written to the SMD23E-B before the driver can be enabled.

Bit 14: Reserved – Must equal “0”.

Bit 13: Reserved – Must equal “0”.

Bit 12: Virtual_Position_Follower – When using the SMD23E-B as an axis follower (SynchroStep Mode), set this bit to ‘1’ to close the position loop with respect to the motor position.

Bit 11: Backplane_Proximity_Bit – When the SMD23E-B is configured to use the Backplane Home 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 Home Proximity bit can be found in the ***Homing Inputs*** section starting on page 55.

Bit 10: Reserved – Must equal “0”.

Command Word 1 Bits (continued)

Bit 9: **Assembled_Move_Type –** When this bit equals “0”, a Blend Move is started when the Run Assembled Move bit, (Command Word 1, Bit 13) makes a 0 → 1 transition. When this bit equals “1”, a Dwell Move is started on the transition. The direction of a Blend Move is controlled by the Blend Move Direction bit, (Command Word 0, 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 SMD23E-B will not run the move immediately, but will instead wait for an inactive-to-active transition on an input configured as a *Start Indexer Move* input. The move will run on every inactive-to-active transition on the DC input provided the move command data, including this bit, remains in the Network Output Data registers.

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: **Reserved –** Must equal “0”.

Bit 5: **Save_to_Flash -** This bit can be used to save a programmed Assembled Move to flash memory. This bit must be set when the Program_Assembled bit (Command Word 0, bit 12) makes a 1 → 0 transition at the end of the programming cycle. The unit responds by flashing the Status LED when the writing is complete. If the LED is flashing green, the write to flash memory was successful. If it flashes red, then there was an error in writing the data. In either case, power must be cycled to the unit before you can continue. This design decision is to protect the flash memory from constant write commands. The flash memory has a minimum of 10,000 write cycles.

Bit 4: **Reverse_Blend_Direction –** When you command a Blend Move to run, this bit determines the direction of rotation. Set to “0” for a clockwise Blend Move, “1” for a counter-clockwise Blend Move.

Bits 3-2: **Reserved –** Must equal “0”.

Bit 1: **Motor Current –** Set to “1” to program the motor current to the value in word 8 of the command block. The motor current can set as a separate command or as part of a move command. The new current value will be used for all future moves. If reset to “0” when a move command is issued, the motor current will remain unchanged and the last programmed motor current will be used. The motor current is also programmed when Configuration data is written to the unit.

Bit 0: **Reserved –** Must equal “0”.

Command Blocks

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

Absolute Move

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	***Command Word 0 Bits***		16#0001
1	1025	***Command Word 1 Bits***		See pg. 70
2	1026	Abs. Target Position: Upper Word	Steps	Combined value between –8,388,607 and +8,388,607
3	1027	Abs. Target Position: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Motor Current	0.1 amps	0 to 34. Ignored if bit 1 of Command Word 0 is not set.
9	1033	Acceleration Jerk		0 to 5000

Table R6.3 Absolute Move Command Block

Relative Move

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	***Command Word 0 Bits***		16#0002
1	1025	***Command Word 1 Bits***		See pg. 70
2	1026	Rel. Target Position: Upper Word	Steps	Combined value between –8,388,607 and +8,388,607
3	1027	Rel. Target Position: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Motor Current	0.1 amps	0 to 34. Ignored if bit 1 of Command Word 0 is not set.
9	1033	Acceleration Jerk		0 to 5000

Table R6.4 Relative Move Command Block

Command Blocks (continued)

Hold Move

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	***Command Word 0 Bits***		16#0004
1	1025	***Command Word 1 Bits***		See pg. 70
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused		See Note Below

Table R6.5 Hold Move Command Block

Unused words are ignored by an SMD23E-B and can be any value, including parameter values in the previous command.

Resume Move

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	***Command Word 0 Bits***		16#0008
1	1025	***Command Word 1 Bits***		See pg. 70
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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Motor Current	0.1 amps	0 to 34. Ignored if bit 1 of Command Word 0 is not set.
9	1033	Acceleration Jerk		0 to 5000

Table R6.6 Resume Move Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command. This is typically the case when resuming a move, the words are listed as “Unused” to highlight that the target position of a held move cannot be changed when the move is resumed.

Command Blocks (continued)

Immediate Stop

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0010
1	1025	*Command Word 1 Bits*		See pg. 70
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused		See Note Below

Table R6.7 Immediate Stop Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

Find Home CW

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0020
1	1025	*Command Word 1 Bits*		See pg. 70
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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Motor Current	0.1 amps	0 to 34. Ignored if bit 1 of Command Word 0 is not set.
9	1033	Acceleration Jerk		0 to 5000

Table R6.8 Find Home CW Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

Command Blocks (continued)

Find Home CCW

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0040
1	1025	*Command Word 1 Bits*		See pg. 70
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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Motor Current	0.1 amps	0 to 34. Ignored if bit 1 of Command Word 0 is not set.
9	1033	Acceleration Jerk		0 to 5000

Table R6.9 Find Home CCW Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

Jog Move CW

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0080
1	1025	*Command Word 1 Bits*		See pg. 70 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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Motor Current	0.1 amps	0 to 34. Ignored if bit 1 of Command Word 0 is not set.
9	1033	Acceleration Jerk		0 to 5000

Table R6.10 Jog Move CW Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

Command Blocks (continued)

Registration Move CW

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0080
1	1025	*Command Word 1 Bits*		See pg. 70 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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
9	1033	Min. Reg. Move Distance: Lower Word		

Table R6.11 Registration Move CW Command Block

Jog Move CCW

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0100
1	1025	*Command Word 1 Bits*		See pg. 70 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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Motor Current	0.1 amps	0 to 34. Ignored if bit 1 of Command Word 0 is not set.
9	1033	Acceleration Jerk		0 to 5000

Table R6.12 Jog Move CCW Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

Command Blocks (continued)

Registration Move CCW

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0100
1	1025	*Command Word 1 Bits*		See pg. 70 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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
9	1033	Min. Reg. Move Distance: Lower Word		

Table R6.13 Registration Move CCW Command Block

SynchroStep (Virtual Axis Follower) Moves

The SynchroStep requires a controller with the capacity for motion axis programming. This table shows the format of the data that must be periodically sent to the SMD23E-B.

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0080 or 16#0100
1	1025	*Command Word 1 Bits*		See pg. 70 Bit 12 must equal “1”. Bits 14 and 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/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	Proportional Coefficient		1 to 50
9	1033	Network Delay OR Position Unwind		0 to 20 OR 21 to 65,535

Table R6.14 Axis Follower Move Command Block

NOTE The type of move is controlled by the value in word 9. If the value is between 0 and 20, it is considered the network delay value and the move is considered to be on a linear axis. If the value is greater than 20, it is considered to be the Position Unwind value and the move is considered to be on a circular axis. Note that word 9 is considered to be an unsigned integer. In systems that only support signed integers, the values of 32,768 through 65,535 will appear as -32,768 through -1.

Command Blocks (continued)

Preset Position

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0200
1	1025	*Command Word 1 Bits*		See pg. 70
2	1026	Position Preset Value: Upper Word	Steps	Combined value between –8,388,607 and +8,388,607
3	1027	Position Preset Value: Lower Word		
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused		See Note Below

Table R6.15 Preset Position Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

- Presetting the position will also reset the Move_Complete status bit.

Reset Errors

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#0400
1	1025	*Command Word 1 Bits*		See pg. 70
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused		See Note Below

Table R6.16 Reset Errors Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

- Resetting errors will also reset the Move_Complete status bit.

Command Blocks (continued)

Run Assembled Move

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#2000
1	1025	*Command Word 1 Bits*		See pg. 70 Blend Move: Bit 9 = "0" Dwell Move: Bit 9 = "1" Blend Move direction set by Bit 4.
2	1026	Unused		See Note Below
3	1027	Unused		See Note Below
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused with Blend Move Dwell Time with Dwell Move	milliseconds	0 to 65,535

Table R6.17 Run Assembled Move Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

Preset Encoder Position

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	*Command Word 0 Bits*		16#4000
1	1025	*Command Word 1 Bits*		See pg. 70
2	1026	Encoder Preset Value: Upper Word	Steps	Combined value between –8,388,607 and +8,388,607
3	1027	Encoder Preset Value: Lower Word		
4	1028	Unused		See Note Below
5	1029	Unused		See Note Below
6	1030	Unused		See Note Below
7	1031	Unused		See Note Below
8	1032	Unused		See Note Below
9	1033	Unused		See Note Below

Table R6.18 Preset Encoder Position Command Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values in the previous command.

Programming Blocks

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

First Block

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	***Command Word 0 Bits***		16#0800
1	1025	***Command Word 1 Bits***		See pg. 70
2 - 9	1026 - 1033	Unused		See Note Below

Table R6.19 Assembled Move First Programming Block

Unused words are ignored by the SMD23E-B and can be any value, including parameter values from the previous command.

Once the first block is transmitted, the SMD23E-B responds by setting bits 8 and 9 in the Status Word 0. (See ***Status Word 0 Format*** starting on page 82.) Once these are set, you can then start transmitting Segment Blocks.

Segment Block

EtherNet/IP Word	Modbus TCP Register	Function	Units	Range
0	1024	***Command Word 0 Bits***		16#1800
1	1025	***Command Word 1 Bits***		See pg. 70
2	1026	Rel. Target Position: Upper Word	Steps	Combined value between –8,388,607 and +8,388,607
3	1027	Rel. Target Position: Lower Word		
4	1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	1029	Programmed Speed: Lower Word		
6	1030	Acceleration	Steps/sec/ms	1 to 5000
7	1031	Deceleration	Steps/sec/ms	1 to 5000
8	1032	*Reserved*		Must equal zero for compatibility with future releases.
9	1033	Acceleration Jerk		0 to 5000

Table R6.20 Assembled Move Segment Programming Block

Note that each Segment Block starts with bits 11 and 12 set in the Command Bits MSW word (16#1800). When the SMD23E-B sees bit 12 of Command Bits MSW set, it will accept the block and reset bit 9 in the Status Bits MSW word. When your program sees this bit reset, it must respond by resetting bit 12 of Command Bits MSW. The SMD23E-B will respond to this by setting bit 9 in the Status Bits MSW word and the next Segment Block can be written to the SMD23E-B. 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 an SMD23E-B is in Command Mode is shown below. EtherNet/IP and Modbus TCP addresses are both shown.

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

Table R6.21 Network Input Data Format: Command Mode

Format of Position Data Values

The format of the Motor Position, Encoder Position, and Captured Encoder Position values is controlled by the Binary_Input_Format bit in the configuration data written to the SMD23E-B. (See ***Configuration Word 1 Format***, bit 9, on page 63.) When the Binary_Input_Format bit equals "0", the position values are reported using the AMCI multi-word format shown below. When the Binary_Input_Format bit equals "1", the position values are reported as 32-bit signed integers, with the least significant bits in the lower word.

NOTE When using the multi-word format, the range of values 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.

Examples of the two formats are given below.

	Multi-Word Format		32 bit Signed Integer Little Endian Format		32 bit Signed Integer Big Endian Format	
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

Figure R6.3 Position Data Format Examples

Input Data Format (continued)

Status Word 0 Format

Status Word 0

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

Figure R6.4 Command Mode: Status Bits MSW Format

Bit 15: **Mode_Flag –** Set to "1" if in Configuration Mode. Reset to "0" if in Command Mode.

Bit 14: **Module_OK –** Set to "1" when the SMD23E-B is operating without a fault. Reset to "0" when an internal fault condition exists.

Bit 13: **Configuration_Error –** Set to "1" on power up before a valid configuration has been written to the SMD23E-B or after any invalid configuration has been written to the unit. Reset to "0" when the SMD23E-B has a valid configuration written to it or if valid configuration data is read from flash memory on power up.

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 –** Set to "1" when an invalid command has been written to the SMD23E-B. This bit can only be reset by a ***Reset Errors*** command. The format of the command is given on page 78.

Bit 11: **Input_Error –** Set to "1" when:

- ➤The 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.
- ➤And 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 78.

Bit 10: **Position_Invalid –** Set to "1" when:

- ➤ Power is cycled to the SMD23E-B
- ➤ A configuration is written to the SMD23E-B
- ➤ The motor position has not been preset
- ➤ The machine has not been homed
- ➤ An Immediate or Emergency Stop has occurred
- ➤ An End Limit Switch has been reached
- ➤ A motor stall has been detected.

The bit is reset to "0" when the motor position has been defined. Absolute moves cannot be performed while this bit is set to "1".

Bit 9: **Wait_For_Assembled_Segment –** The SMD23E-B 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 40.

Input Data Format (continued)

Status Word 0 Format (continued)

Bit 8: **In_Assembled_Mode –** The SMD23E-B 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 40.

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 SMD23E-B, when the position is preset, or when a Reset Errors command is issued to the unit. This bit is also set along with the Command Error bit (Bit 12 of this word), when any Jog Move or Registration Move parameters are outside of their valid ranges. This bit is not set on a command error for any other type of command. Finally, this bit is not set at the end of a homing operation.

Bit 6: **Decelerating –** Set to “1” when the present move is decelerating. Set to “0” at all other times.

Bit 5: **Accelerating –** Set to “1” when the present move is accelerating. Set to “0” at all other times.

Bit 4: **At_Home –** Set to “1” when a homing command has completed successfully, “0” at all other times.

Bit 3: **Stopped –** Set to “1” when the motor is not in motion. Note that this is stopped for any reason, not just a completed move. For example, an Immediate Stop command during a move will set this bit to “1”, but the Move_Complete bit, (bit 7 above) will not be set.

Bit 2: **In_Hold_State Bit –** Set to “1” when a move command has been successfully brought into a Hold State. Use of the Hold_Move and Resume_Move command bits can be found for each move type in the section ***How to Control Moves in Progress*** starting on page 27.

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_Is_Enabled	Stall_Detected	Acknowledge_Flag	0	Heartbeat_Bit	Limit_Condition	Inv_Jog_Change	0	Driver_Fault	Conn_Was_Lost	PLC_In_PGM	Temp_Above_90C	0	IN3_Active	IN2_Active	IN1_Active

Figure R6.5 Command Mode: Status Bits LSW Format

Bit 15: **Drive_Is_Enabled –** When set to “0”, current is not available to the motor under any condition. When set to “1”, the motor driver section of the SMD23E-B is enabled and current is available to the motor, but current may be removed for other reasons. Motor current is removed if there is a Driver Fault (Bit 7 below), or if the motor is idle and Idle Current Reduction is programmed to its *To 0%* setting. Note that the motor will still receive power under an E-Stop condition.

Bit 14: **Stall_Detected –** Set to “1” when a motor stall has been detected. This is only available when the SMD23E-B has been configured to detect stalls. This bit will always equal “0” on SMD23E-B units that do not have the encoder and stall detection enabled. This bit is reset by a Reset Errors or Preset Motor Position commands.

Input Data Format (continued)

Status Word 1 Format (continued)

Bit 13: **Acknowledge_Flag –** Normally “0”. This bit is set to “1” when one of the following commands completes successfully:

- Preset Position
- Preset Encoder Position
- Reset Errors

This bit resets to “0” when the command bit is reset to “0” by the host controller.

Bit 12: **Reserved –** 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 The Heart Beat bit is only available when in Command Mode.

Bit 10: **Limit_Condition –** This bit is set to “1” 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 Bit –** 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. Issuing a Reset Errors command will not reset this bit. This bit will reset when valid jog data is sent from the host controller.

Bit 8: **Reserved –** Will always equal zero.

Bit 7: **Driver_Fault –** If the driver section of the SMD23E-B is enabled, this bit will be a “1” during an Overtemperature Fault. Even though the driver is enabled, it will not supply current to the motor until the motor’s temperature decreases to a safe value. At this point the fault will clear itself.

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_PGM –** On ControlLogix and CompactLogix platforms, this bit equals “1” when the PLC is in Program mode and “0” when in Run mode. This bit will always equal “0” on all other platforms.

Bit 4: **Temperature_Above_90C –** This bit is set to “1” when the processor internal temperature exceeds 90°C. At this point, the heatsink temperature is typically near 83°C. If this bit trips often and you want to lower the operating temperature of the unit, consider changing how the device is mounted, or installing a fan to force additional airflow around the unit. This bit will clear itself when the temperature decreases.

Bit 3: **Reserved Bit –** Will always equal zero.

Bit 2: **Input3 State Bit –** “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 63.

Bit 1: **Input2 State Bit –** “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 63.

Bit 0: **Input1 State Bit –** “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 63.

Notes on Clearing a Driver Fault

A Driver Fault occurs when there is an over temperature condition. When a Driver Fault occurs, the SMD23E-B sets bit 7 of the Status Word 1 in the Network Input Data. The only way to clear this fault is to lower the temperature of the motor. Once the motor’s temperature decreases to a safe value, the fault will clear itself.

REFERENCE 7

CONFIGURING NETWORK INTERFACES

This section lists suggestions for configuring the network interfaces on your computer or laptop before attaching to the SMD23E-B.

Firewall Settings

Firewalls are hardware devices or software that prevent unwanted network connections from occurring. Firewall software has been present on Windows based computers since XP, and it may prevent your computer from communicating with the SMD23E-B. Configuring your firewall to allow communication with the SMD23E-B is beyond the scope of this manual.

Disable All Unused Network Interfaces

Routing and default gateway setting on your computer might prevent connection to the SMD23E-B. When using the Net Configurator utility, broadcast packets that are used to find the SMD23E-B often go out the wrong port. The easiest way to avoid this problem is to temporarily disable all network interfaces that are not attached to the SMD23E-B.

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

Configure Your Network Interface

Before you can communicate with the SMD23E-B, your network interface must be on the same subnet as the device.

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 10 and 11, press the [Win+X] keys and select “Command Prompt” or “Terminal” from the resulting popup. There is no need to run the command prompt as the administrator.

Configure Your Network Interface (continued)

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 SMD23E-B unit. Figure R7.1 shows the output of an ipconfig command that shows the "Local Area Connection 2" interface on the 192.168.0.xxx subnet.

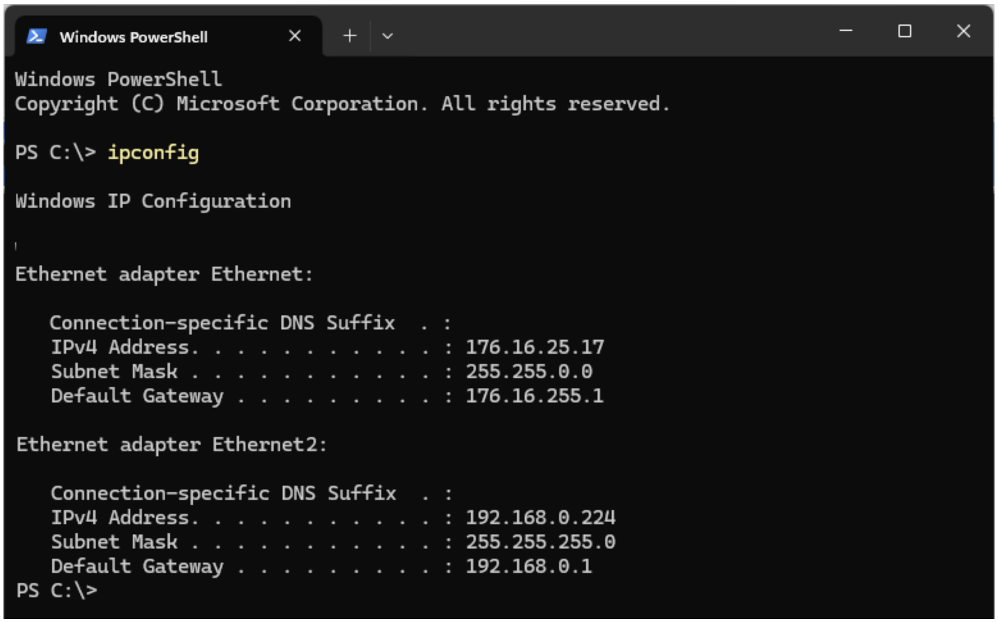


Figure R7.1 ipconfig Command

If your present address is not in the 192.168.0.xxx range, type in 'ncpa.cpl' at the command prompt and hit [Enter] on the keyboard. This will open the *Network Connections* window.

Right click on the appropriate interface and select "Properties". 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 SMD23E-B that has factory default settings.) The default gateway and DNS server settings can be ignored.

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' is the IP address of the SMD23E-B. The computer will ping the unit and the message "Reply from aaa.bbb.ccc.ddd: bytes=32 time<10ms TTL=128" should appear four times.

If the message "Request timed out." or "Destination host unreachable" appears, then one of four things has occurred:

- You set a new IP address, but have not yet cycled power to the SMD23E-B.
- You did not enter the correct address in the ping command.
- The IP address of the SMD23E-B is not set correctly.
- The SMD23E-B and the computer are not on the same subnet.

TASK 1
INSTALLATION

1.1 Location

SMD23E-B units are suitable for use in an industrial environment 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.

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 SMD23E-B. Follow these guidelines when handling the unit.

1) Touch a grounded object to discharge static potential before handling the unit.
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 network connector or I/O connector.
5) Do not disassemble the unit
6) Store the unit in its shipping box when it is not in use.

1.2.2 Prevent Debris From Entering the Unit

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

1.2.3 Remove Power Before Servicing in a Hazardous Environment

WARNING Remove power before removing or installing any SMD23E-B units in a hazardous environment.

1.3 Operating Temperature Guidelines

Due to the onboard electronics, the maximum operating temperature of the SMD23E-B is limited to 203°F/95°C. The motor by itself has a maximum operating temperature of 266°F/130°C. Depending on the operating current setting, move profiles, idle time, and the idle current reduction setting, it is possible to exceed these temperatures in a thermally isolated environment. As explained in the mounting section, mounting the SMD23E-B to a large metal heatsink is the best way to limit the operating temperature of the device. Operating temperature should be monitored during system startup to verify that the maximum motor temperature remains below its 203°F/95°C specification. SMD23E-B devices have an onboard thermistor and will remove motor current if the operating temperature of the electronics exceeds this limit. These ovetemperature faults are also reported in the Network Input Data.

1.4 Mounting

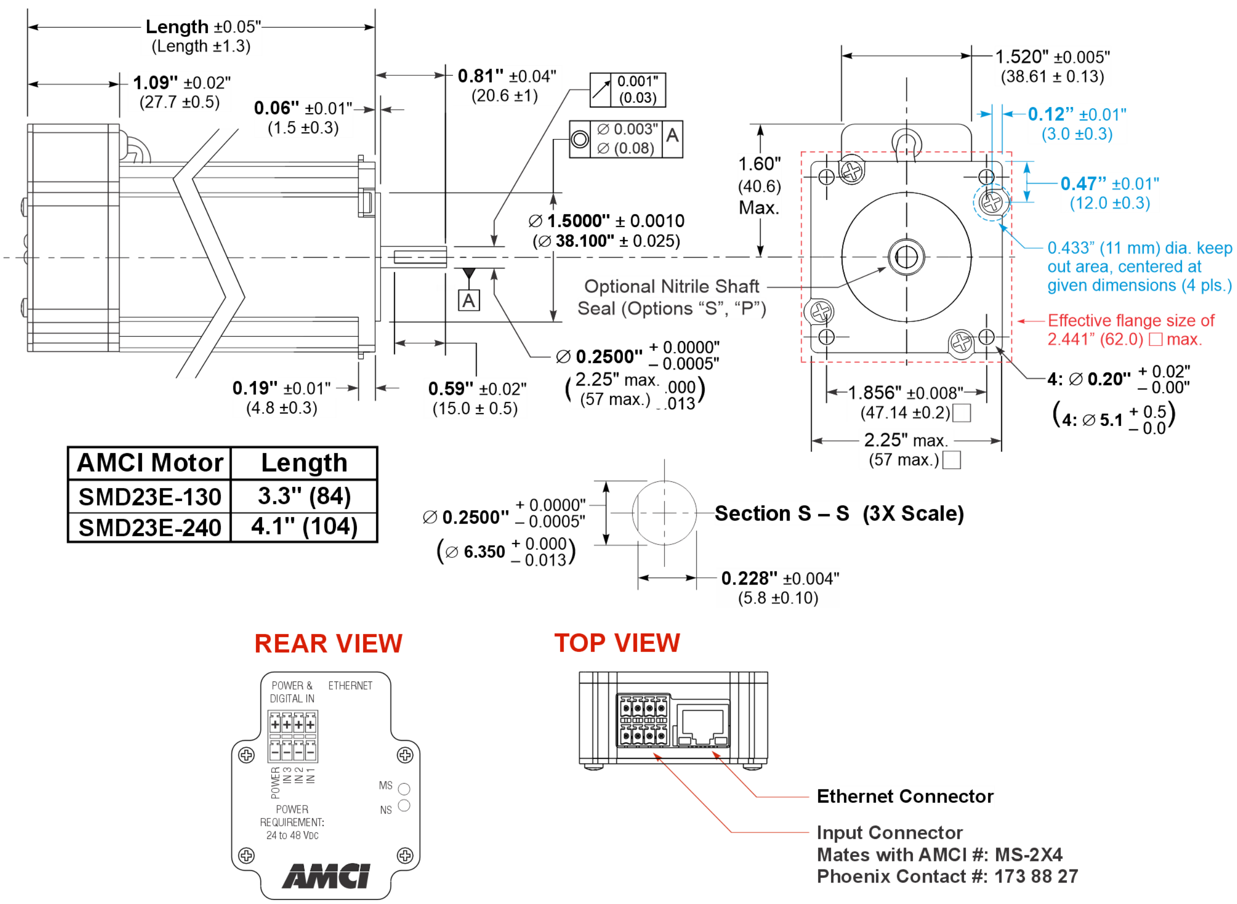
All AMCI motor have flanges on the front of the motor for mounting. This flange also acts as a heatsink, so motors should be mounted on a large, unpainted metal surface. Mounting a motor in this fashion will allow a significant amount of heat to be dissipated away from the motor, which will increase the unit'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 SMD23E-B.

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.

1.4.1 SMD23E-B Outline Drawing



AMCI Motor	Length
SMD23E-130	3.3" (84)
SMD23E-240	4.1" (104)

Figure T1.1 SMD23E-B Outline Drawing

1.5 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. Limit shaft loading to the following maximums:

- Radial: 19 lbs (85N) at end of shaft
- Axial: 3.37 lbs (15N)

1.6 Network Connection

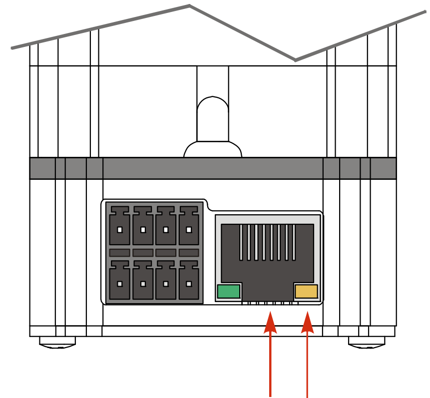


Figure T1.2 Ethernet Port Location

The Ethernet connector is located on the top of the SMD23E-B. The connector is a standard RJ-45 jack that will accept any standard, shielded or non-shielded, 100baseT cable. Because the port can run at 100 Mbit speeds, Category 5, 5e, or 6 cable should be used.

The Ethernet port on the SMD23E-B has "auto switch" capability. This means that a standard cable can be used when connecting the SMD23E-B to any device. A crossover cable is not necessary when connecting an SMD23E-B directly to a PC.

The three status LED's are fully described in the ***Status LED's*** section of this manual starting on page 18.

1.7 I/O Connector Pin Out

The I/O Connector is located on the top of the SMD23E-B. The mate for this connector is included with the unit. It is also available from AMCI under the part number MS-2X4 and Phoenix Contact under their part number 173 88 27. Figure T1.3 shows the pin out for the I/O connector.

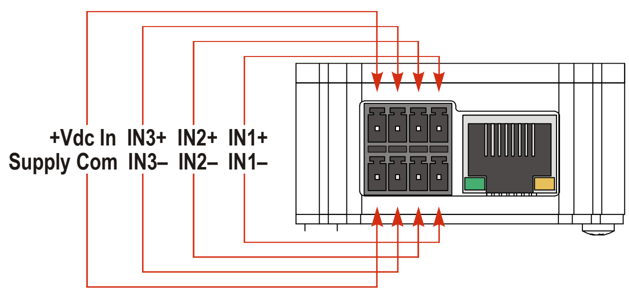


Figure T1.3 I/O Connector

1.8 Power Wiring

The SMD23E-B accepts 24 to 48Vdc as its input power. The power pins are the outer two pins of the MS-2X4 connector. See figure T1.3 for proper connections to the MS-2X4. The connector will accept 16 to 24 AWG wire. AMCI strongly suggests using 16 AWG wire to minimize power losses.

CAUTION Do not apply 120 Vac to any pins of the SMD23E-B. If this occurs, the unit will be damaged and you will void the unit's warranty.

1.9 Input Wiring

Figure T1.4 below shows how to wire discrete DC differential, sourcing, and sinking sensors to Inputs 1, 2, and 3 of the SMD23E-B.

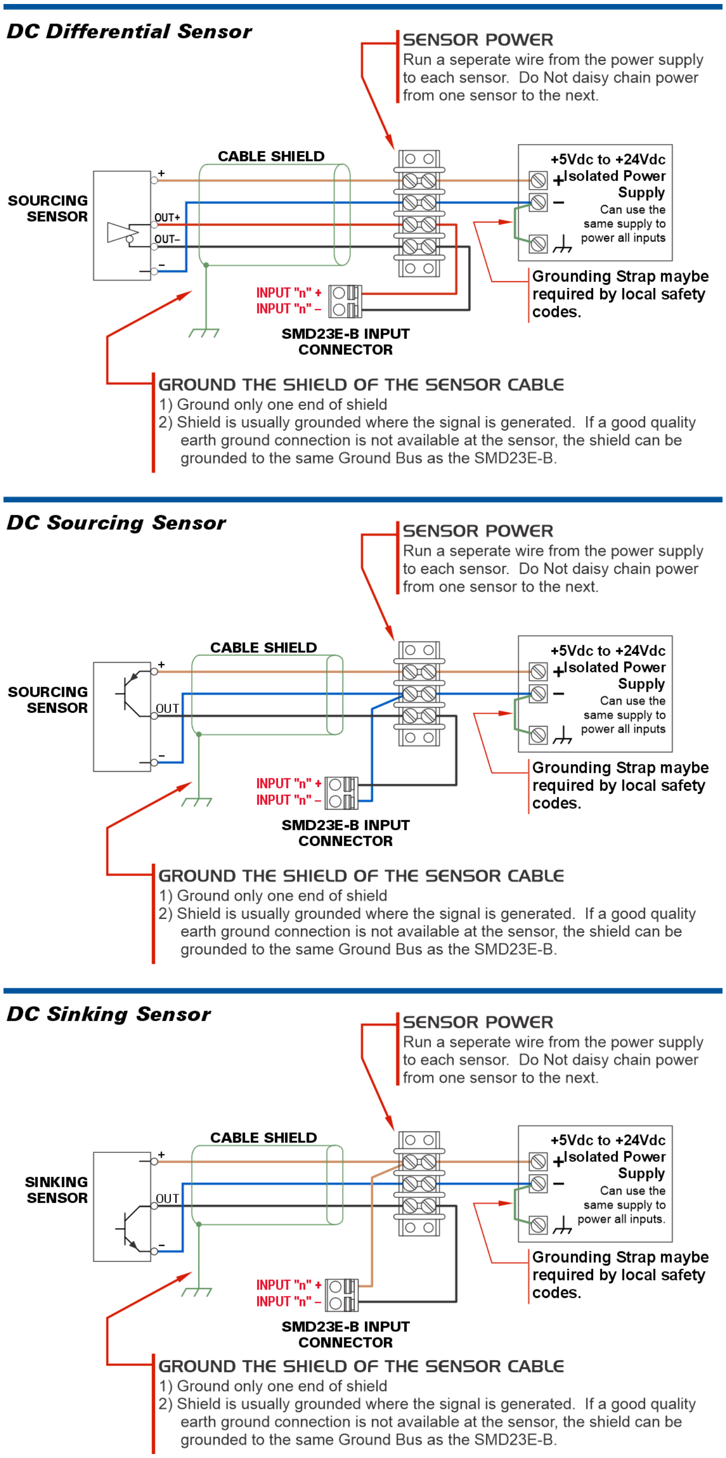


Figure T1.4 Input Wiring

Input Specifications:

Differential. 2500 Vac/dc opto-isolated.† Can be wired as single ended inputs. Accepts 3.5 to 27Vdc without the need for an external current limiting resistor.

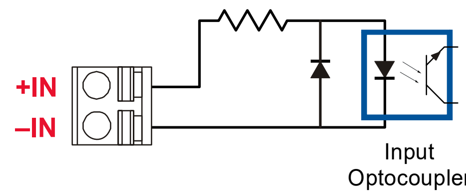


Figure T1.5 Simplified Input Schematic

If using 48 Vdc sensors, you are required to install a 5 kilohm resistor in series with the input. You will damage the SMD23E-B if you apply 48Vdc directly to the input.

Because they are low power signals, cabling from the sensor to the SMD23E-B 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 SMD23E-B.

† This ratings refer to the optocouplers ability to prevent damaging voltages from crossing its isolation barrier to the rest of the SMD23E-B. If 120 Vac voltage is applied to an input, the input itself will still be destroyed.

TASK 2

Set the IP Address & Network Protocol

This task covers the methods for setting the IP address of an SMD23E-B. It also covers the procedure for changing the communications protocol from EtherNet/IP to Modbus TCP or vice versa.

2.1 Determine the Best Method for Setting the IP Address

There are two methods for setting the IP address on an SMD23E-B. Table T2.1 below outlines the available methods and when you can use them.

Method	Restrictions	Starting Page
Use Factory Defaults	1) The machine must use 192.168.0.xxx subnet. 2) The 192.168.0.50 address must be available.	See Below
Use the AMCI NET Configurator Utility	No restrictions on use. The software can be used to set the IP addressing information as well as the network protocol. The IP address will be stored in nonvolatile memory and used on subsequent power-ups. The software can also be used later to configure the configuration parameters of the SMD23E-B.	92

Table T2.1 Methods for Setting the IP Address

NOTE There is a MAC address label on each SMD23E-B which has a writable surface. There is room on the label for writing the programmed IP address of the unit. It is a best practice to use this label to document the IP address of the unit in case it is ever re-purposed.

2.2a Use Factory Defaults

Each SMD23E-B ships from the factory with a default IP address of 192.168.0.50, a subnet mask of 255.255.255.0, and a default gateway of 192.168.0.1. The default network protocol is EtherNet/IP. If these setting are acceptable for your application, no further action is required for this task.

NOTE An SMD23E-B retains its IP address information and network protocol settings in nonvolatile memory. Therefore, the address of your SMD23E-B may be different if your unit is not directly from the factory.

The factory default address for the SMD23E-B unit 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 ***Test Your Network Interface*** which starts on page 86.

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.

An FAQ is available on the AMCI website that covers how to use the Wireshark program.
https://www.amci.com/files/3315/7652/6333/FAQ_Determing_unknown_IP_address.pdf

2.2b Use the AMCI NET Configurator Utility

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

PREREQUISITE: ***Installation*** (page 87) You must be able to power the device and make a network connection before the NET Configurator utility can be used to set the IP address.

PREREQUISITE: ***Configuring Network Interfaces*** which starts on page 85. The network interfaces on your computer must be on the same subnet before you can communicate with the unit.

2.2b.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 NET Configurator Utility for all networked products...".

2.2b.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.

2.2b.3 Verify that Your Host Controller is Disconnected from the SMD23E-B

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 SMD23E-B, the AMCI Net Configurator utility must act as a bus master. Therefore, physically disconnect your host controller from the SMD23E-B before starting the Net Configurator utility.

2.2b.4 Apply or Cycle Power to the SMD23E-B

Cycling power to the SMD23E-B will reset any connections it may have with the host controller.

2.2b Use the AMCI NET Configurator Utility (continued)

2.2b.5 Start the AMCI Net Configurator Utility

Double click on the utility's icon. A welcome screen similar to the one in figure T2.1 below will appear.

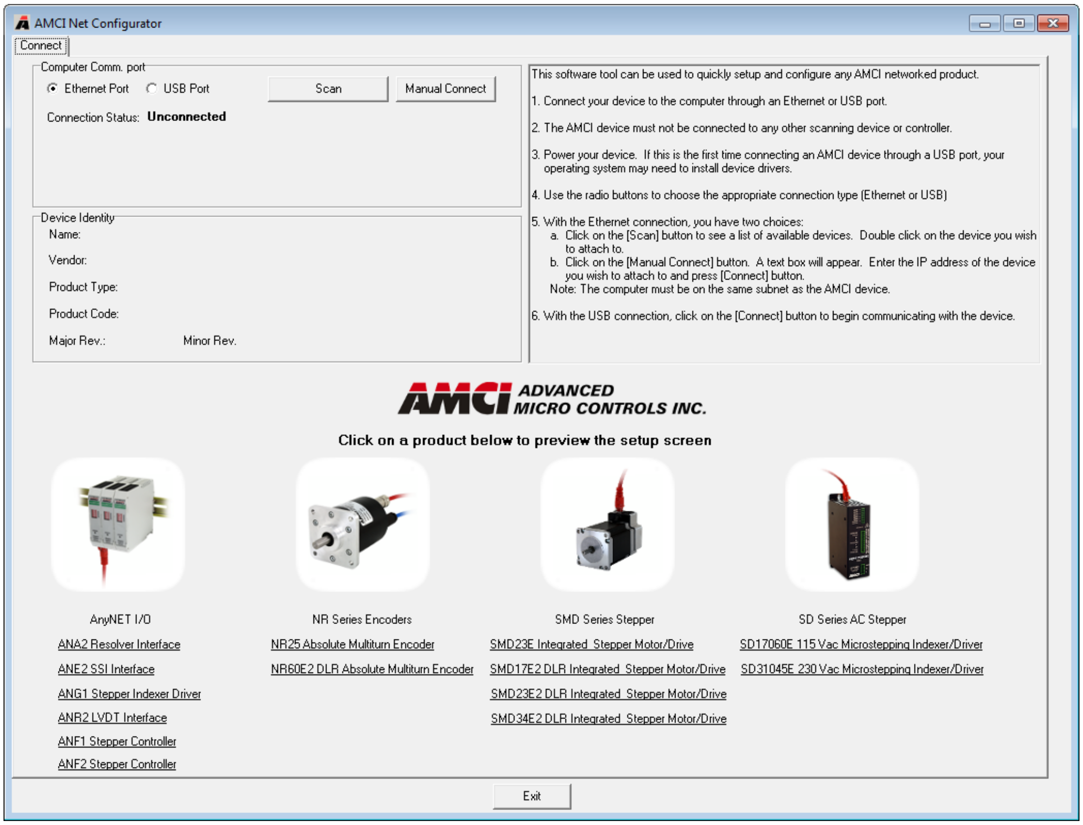


Figure T2.1 Net Configurator Welcome Screen

2.2b Use the AMCI Net Configurator Utility (continued)

2.2b.6 Press the [SCAN] button and Connect to the SMD23E-B

Pressing the [Scan] button will open the window shown in figure T2.2. The SMD23E-B will appear in the scan list only if the unit and your network interface are on the same subnet. Optionally, you can press the [Manual Connect] button and enter the IP address of the unit.

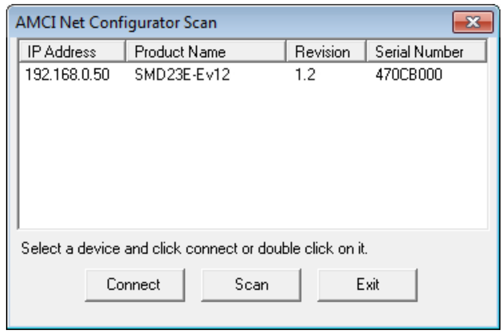


Figure T2.2 Scan for SMD23E-B

If scanning for the SMD23E-B, click on the IP Address of the unit and click on the [Connect] button. The Net Configurator utility will connect to the unit.

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

Figure T2.3 below shows the screen that results when you are connected to the SMD23E-B. In order to change the IP Address of the unit, you must first click on the checkbox next to the text “Allow IP configuration changes. You will need to restart the device.” Once the checkbox is selected, the [Set IP Address] and protocol select buttons will be enabled.

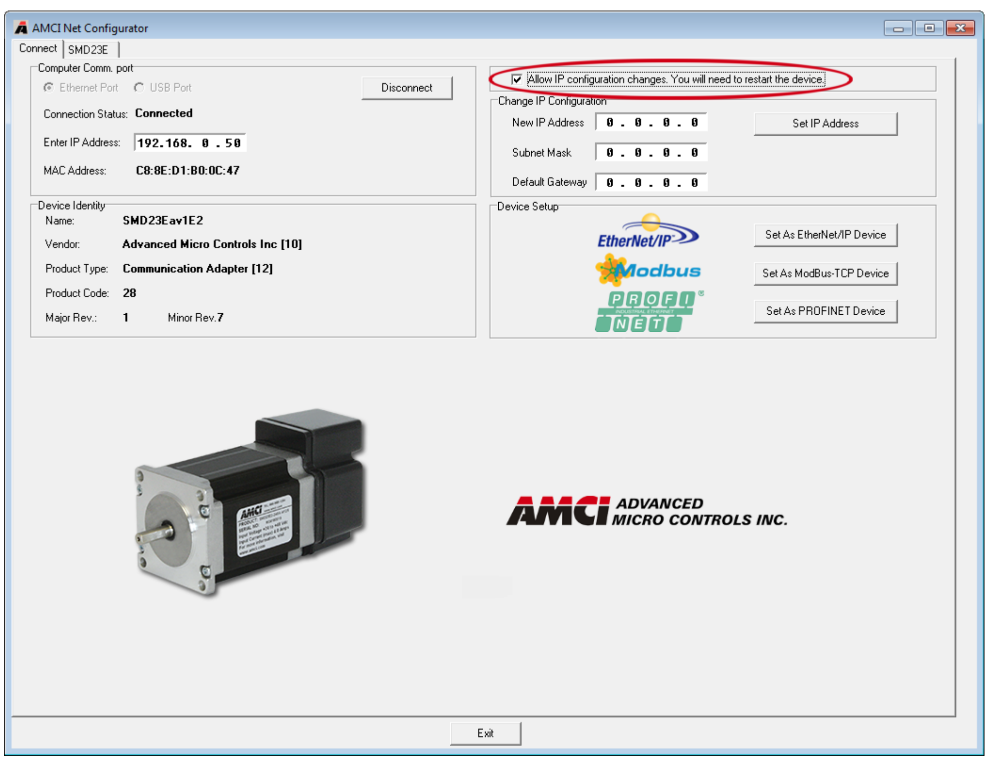


Figure T2.3 Enable IP Address Changes

2.2b Use the AMCI Net Configurator Utility (continued)

2.2b.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! In order to comply with the ODVA specification, it must be set to a valid address on the chosen subnet. Because the Default Gateway is often not used in device level networks, if you do not have a required value for it, AMCI suggests setting the Default Gateway to the IP address of your host controller.

2.2b.9 Set the Communications Protocol

The factory default protocol for the SMD23E-B is EtherNet/IP. In order to use the Modbus TCP protocol, simply click on the appropriate button.

2.2b.10 Write the New IP Address to the SMD23E-B

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.

2.2b.11 Remove Power from the SMD23E-B

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

Task Complete

Notes

Implicit Communications With an EDS

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

Note: Use of an EDS file is completely optional. The SMD23E-B can always be added to a system as a generic module. If you are using RSLogix 5000 version 19 and below, or RSLogix 500, adding the unit as a generic module is the only option available.

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

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. This will open the EDS Wizard.

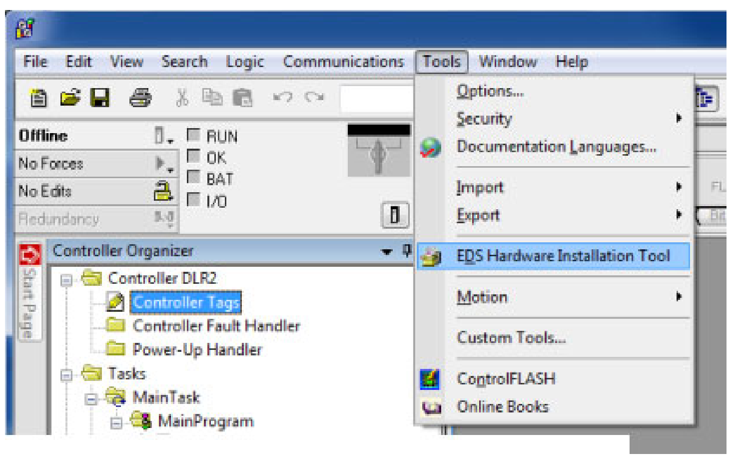


Figure T3.1 Opening the EDS Wizard

2) 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 >].

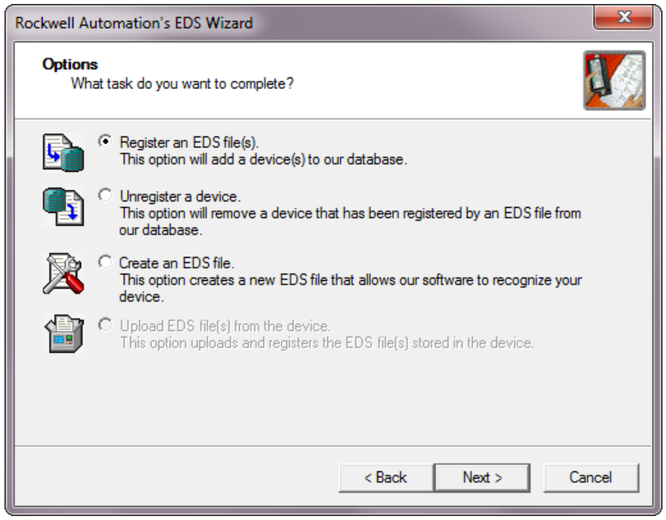


Figure T3.2 EDS Options Screen

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

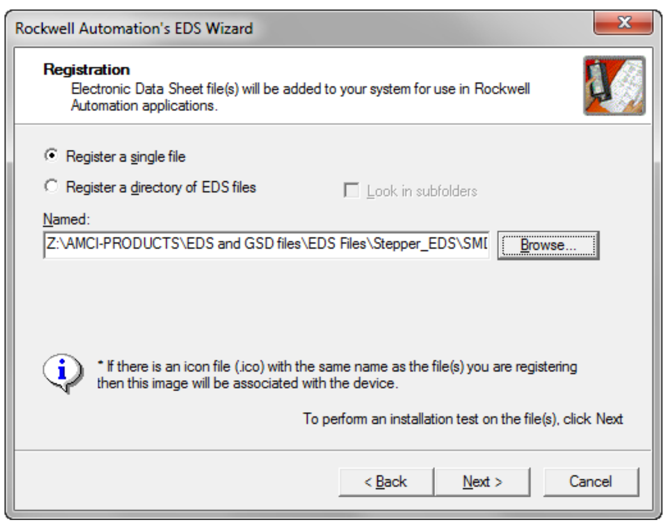


Figure T3.3 EDS Registration Screen

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.

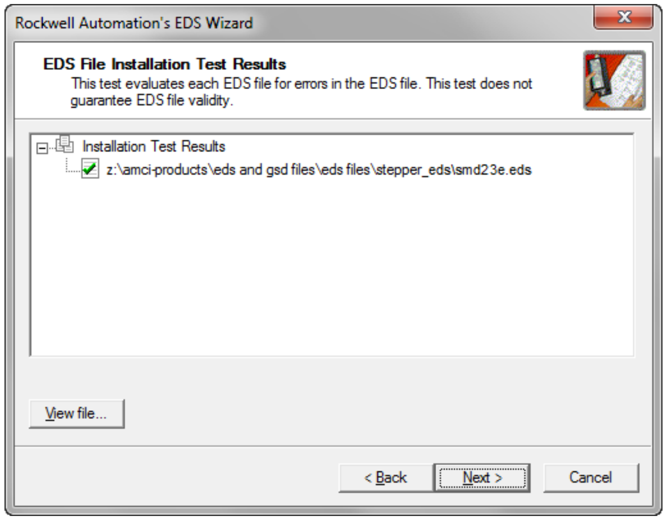


Figure T3.4 EDS Test Screen

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

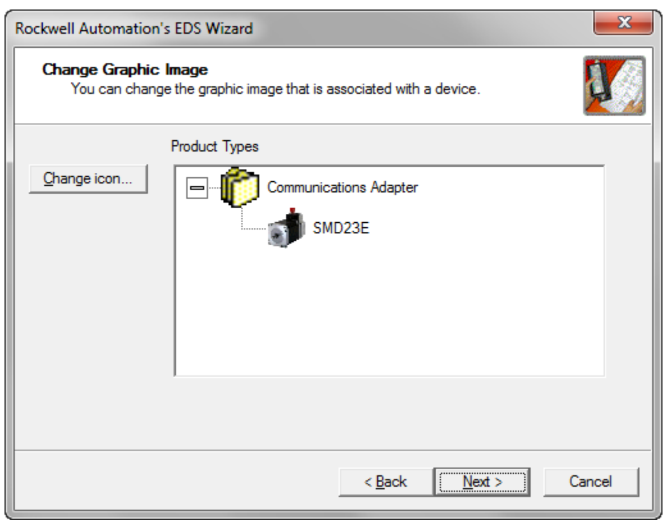


Figure T3.5 Change ECS Icon Screen

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.
9) Click on the [Finish] button to exit the EDS wizard.

3.3 Host System Configuration

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

If the Ethernet port is built into processor, the only step you have to take before adding an AMCI SMD23E-B 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 SMD23E-B to Your Project

You can add an AMCI SMD23E2-B 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.

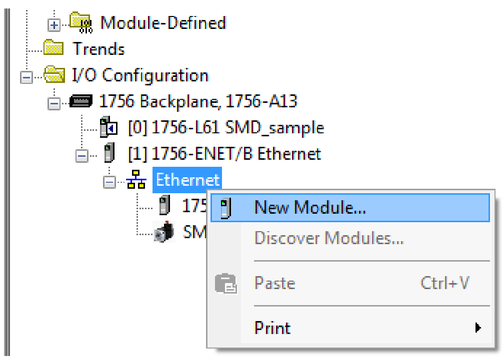


Figure T3.6 Adding an AMCI Ethernet Driver

2) In the resulting Select Module Type screen, select "Advanced Micro Controls Inc. (AMCI)" in the Vendor Filters. This will limit the results to catalog numbers from AMCI.
3) Select "SMD23E" in the resulting list.
4) Click on the [Create] button to create the module.
5) Click on [Close] if necessary to close the Select Module Type screen.

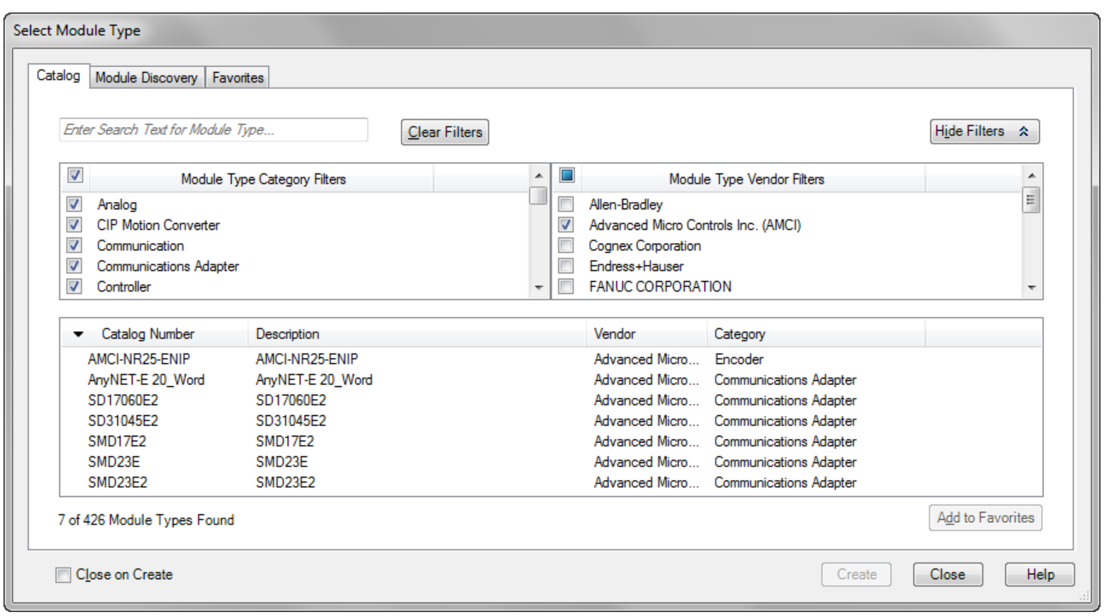


Figure T3.7 Selecting the Networked Driver

3.5 Configure the SMD23E-B Driver

If you are continuing from step 3.4, the resulting New Module screen is used to configure the network connection between the SMD23E-B and your controller. If you need to open the screen to perform this task at a later time, right click on the unit in the project tree and then select "Properties" from the drop-down menu

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

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.

3.5.3 Configuration Tab

The Configuration tab is used to define the configuration data that is written down to the SMD23E-B when the device connects to the network. You can also click on the [Apply] button to write down the configuration data to the unit at any time.

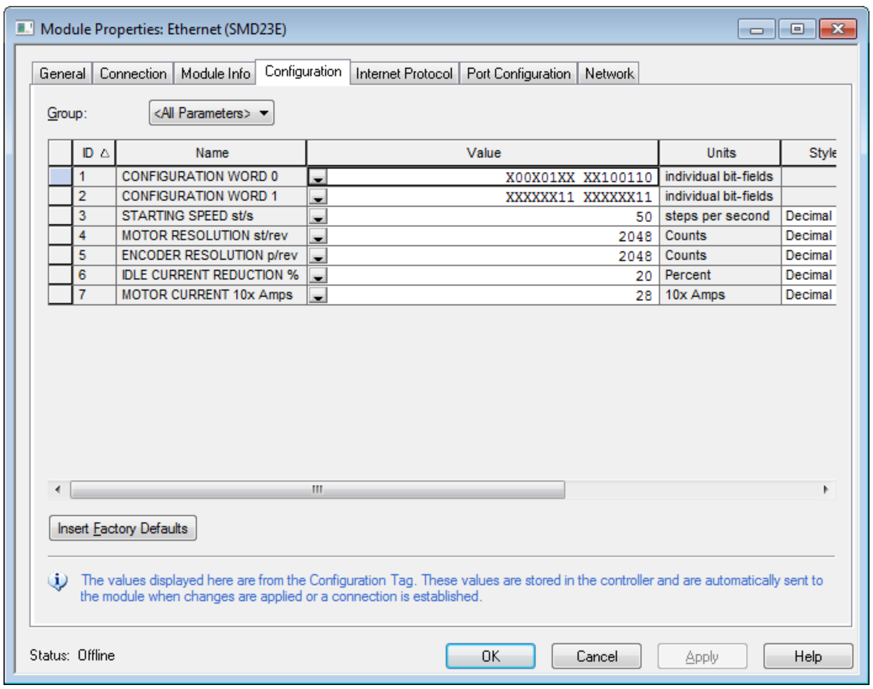


Figure T3.8 Networked Driver Configuration with EDS File

The EDS file defines tags that are used to configure the SMD23E-B. These tags follow the format of the Configuration Data given in reference chapter ***Configuration Mode Data Format***, starting on page 59.

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

3.6 Buffering the I/O Data

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

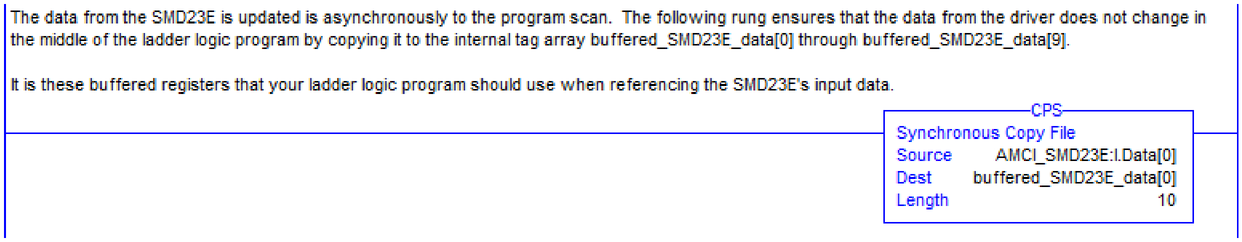


Figure T3.9 Buffer I/O Data

- 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.

Implicit Communications Without an EDS

An AMCI SMD23E 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 SMD23E 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 Add the SMD23E

You can add the SMD23E 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.

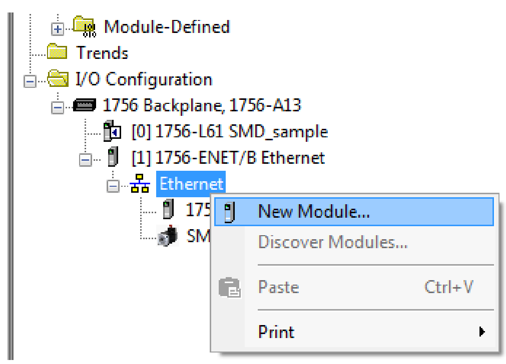


Figure T4.1 Adding an AMCI SMD23E

4.2 Add the SMD23E (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.

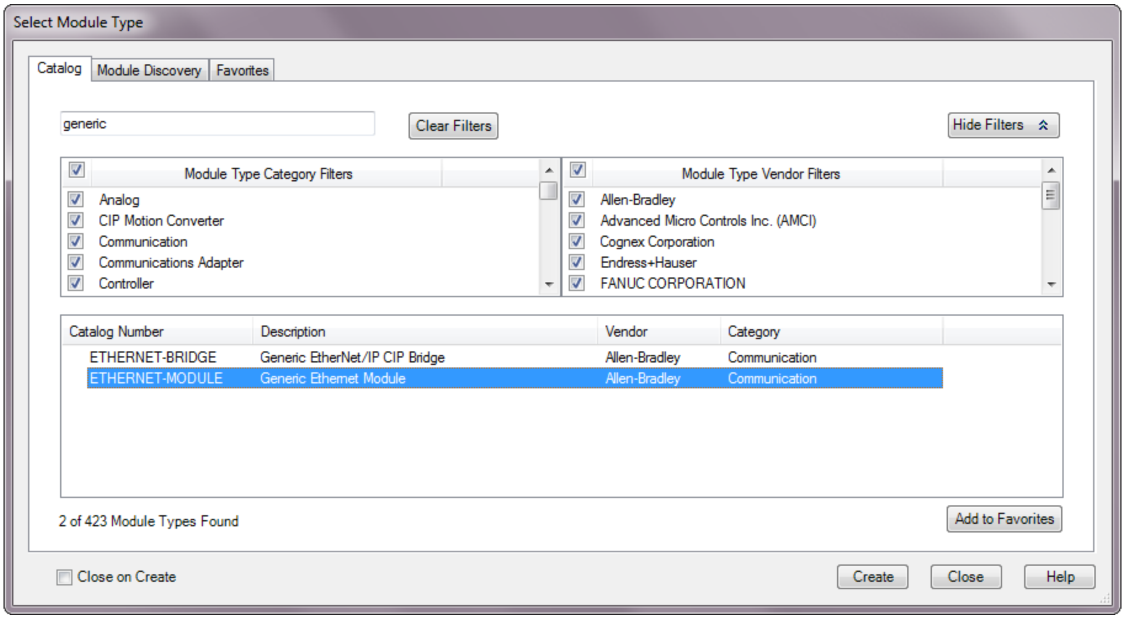


Figure T4.2 Selecting a Generic Device

4.2 Add the SMD23E (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.

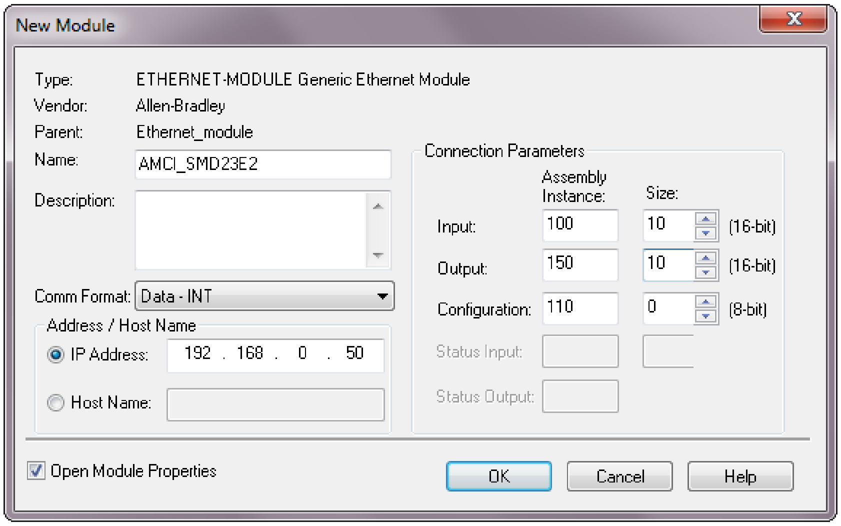


Figure T4.3 Configuration Screen - Generic Device

- **Name: A** descriptive name for the SMD23E
- **Comm Format: D**ata - INT

CAUTION The Comm Format defaults to Data - DINT. The SMD23E will not be able to communicate with the host controller if this format is not changed when the device is added to the system. Once added, the Comm Format cannot be changed. The device must be deleted and again added to the project if the Comm Format is incorrect.

- **IP Address:** Must be the address you set for the SMD23E. Refer to the ***Set the IP Address & Network Protocol*** task chapter starting on page 91 for information on setting the IP Address of the unit.
- **Input:** Assembly Instance = 100, Size = 10 words.
- **Output:** Assembly Instance = 150, Size = 10 words.
- **Configuration: A**ssembly Instance = 110, Size = 0

7) Verify that the “Open Module Properties” check box is selected and click on [OK]. The Module Properties window will open. You can set the RPI time as required for your system in this window. The minimum RPI time for an SMD23E 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 SMD23E.

4.3 Configure the SMD23E

The device will join the EtherNet/IP network as soon as the request is made to it. If the SMD23E has a configuration stored in flash memory, it will be used to configure the unit on power up. You can also configure the unit at anytime and store this new configuration to flash. Configuration is accomplished by writing a block of data to the device that is formatted according to the specifications in the ***Configuration Mode Data Format*** reference chapter, which starts on page 59.

It is possible to store configuration data in the flash memory of the SMD23E and this configuration will be used on power up to configure the device. However, writing the configuration data to the unit on power up may simplify system maintenance if the device ever has to be replaced.

4.4 Buffer I/O Data

Data to and from the SMD23E should be buffered once per scan using Synchronous Copy instructions. This is to insure stable input data during the program scan and guarantee that complete command data is delivered to the device. Ten word integer arrays can be used for this purpose.

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

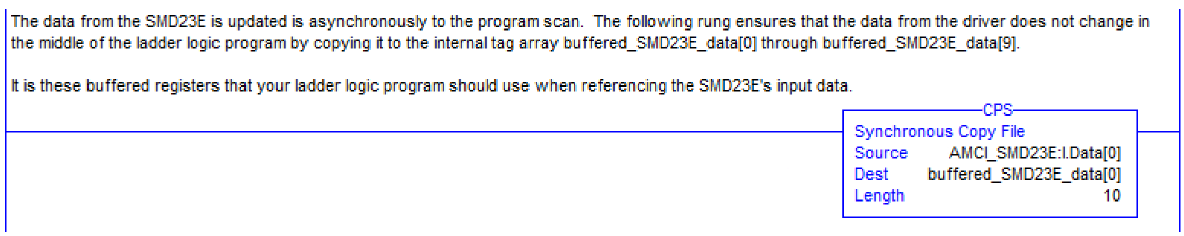


Figure T4.4 Buffer I/O Data

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 SMD23E. 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 will be used as an example in this chapter.

5.1 Required Message Instructions

Only two instructions are required to transfer data between the PLC and the SMD23E. One instruction reads data from the unit and the other writes data to it. The sample programs available from AMCI use this style of programming. The two instructions are alternately triggered using the instruction's ENABLE bits. The remainder of the program controls when data in the source tags of the write instruction changes. The following table gives the required attributes for the instructions.

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

Table 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. Message Instructions do not work correctly in version 10 of RSLogix 500.

5.2 Create Four New Data Files.

- An Integer file to contain the data read from the SMD23E. This file must be at least 10 words in length.
- An Integer file to contain the data written to the SMD23E. 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 Instructions to your Ladder Logic

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

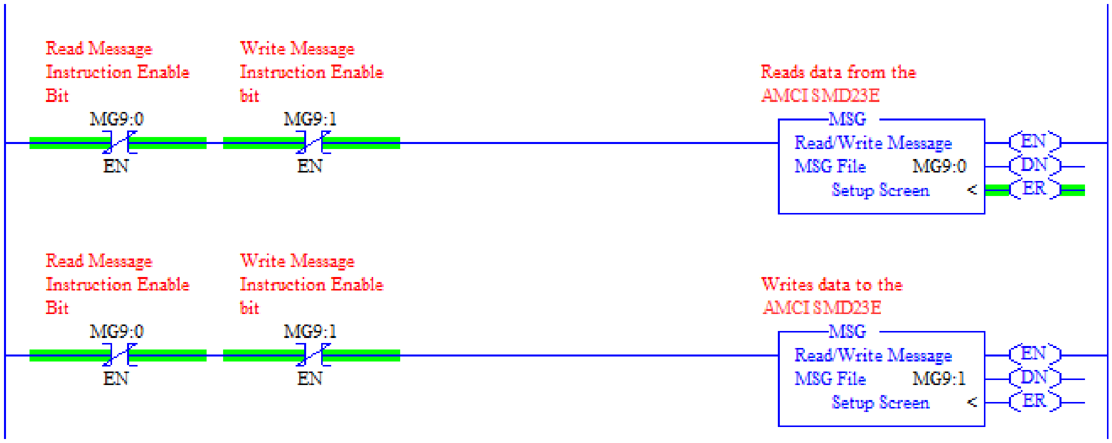


Figure T5.1 Message Instruction Example

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.

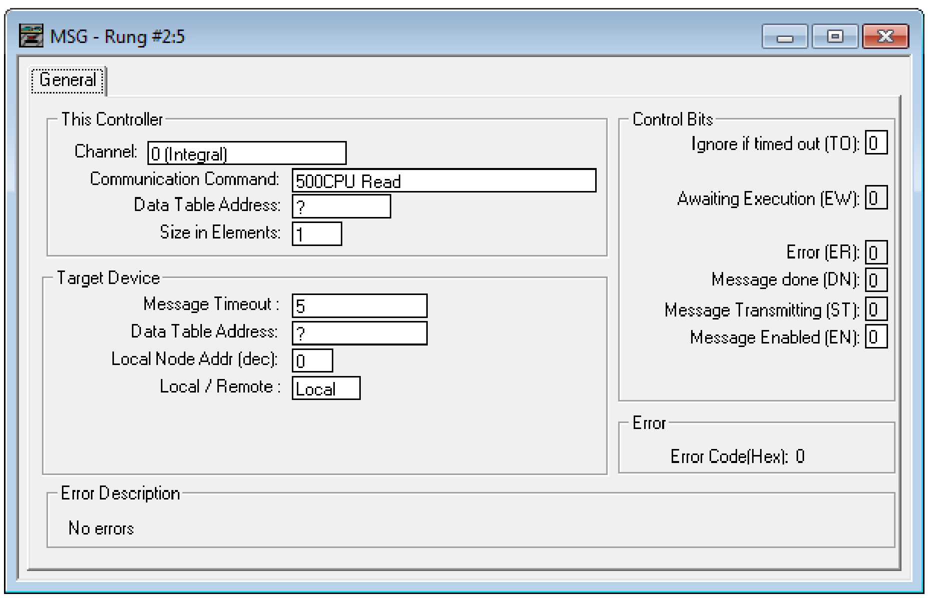


Figure T5.2 Message Instruction Setup Screen

2) Double click in the *Channel* field, click on the t, select "1 (Integral)", and press Enter.
3) Double click in the *Communication Command* field, click on the t, select "CIP Generic" and press Enter.
4) If the Message Instruction is being used to read data from the SMD23E, 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 SMD23E, enter the integer file where the source data will be located in the *Data Table Address (Send)* field and press Enter.
6) Enter "20" as the number of bytes needed in either the *Size In Bytes (Receive)* or *Size In Bytes (Send)* fields. The SMD23E 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 Instructions to your Ladder Logic (continued)

8) Double click in the *Service* field and select "Read Assembly" for a Message Instruction that is being used to read data from the SMD23E, or "Write Assemble" for a Message Instruction that is being used to send data to the SMD23E2 or SMD24E2, 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 SMD23E. Please note that the Data Table Address (Receive) field may be different in your application.

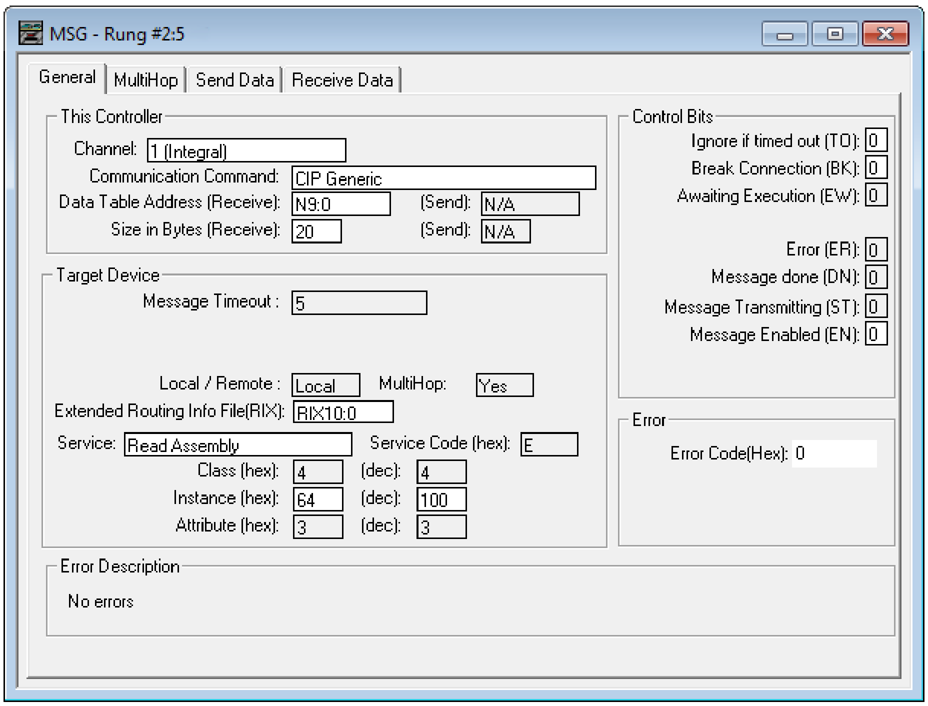


Figure T5.3 Read Message Instruction Setup Screen

5.3 Add the Message Instructions to your Ladder Logic (continued)

The figure below show a typical configuration for Message Instructions being used to write data to the SMD23E. Please note that the Data Table Address (Send) field may be different in your application.

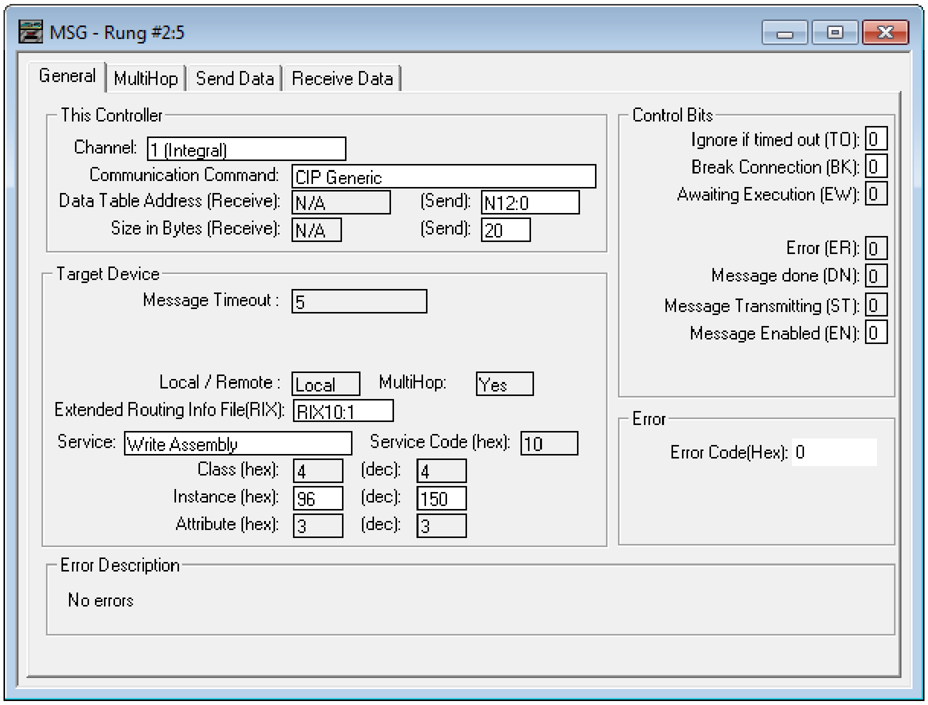


Figure T5.4 Write Message Instruction Setup Screen

Click on the MultiHop tab on the top of the window. As shown in figure T5.5, enter the IP address of the SMD23E and press Enter.

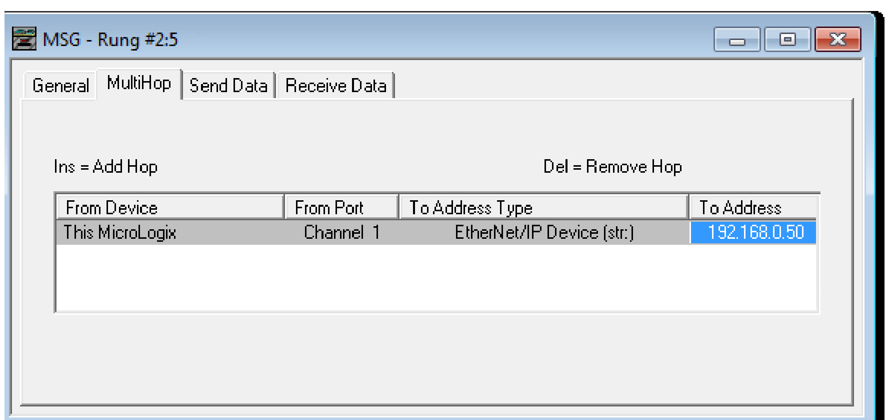


Figure T5.5 Message Instruction MultiHop Settings

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 SMD23E, 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.

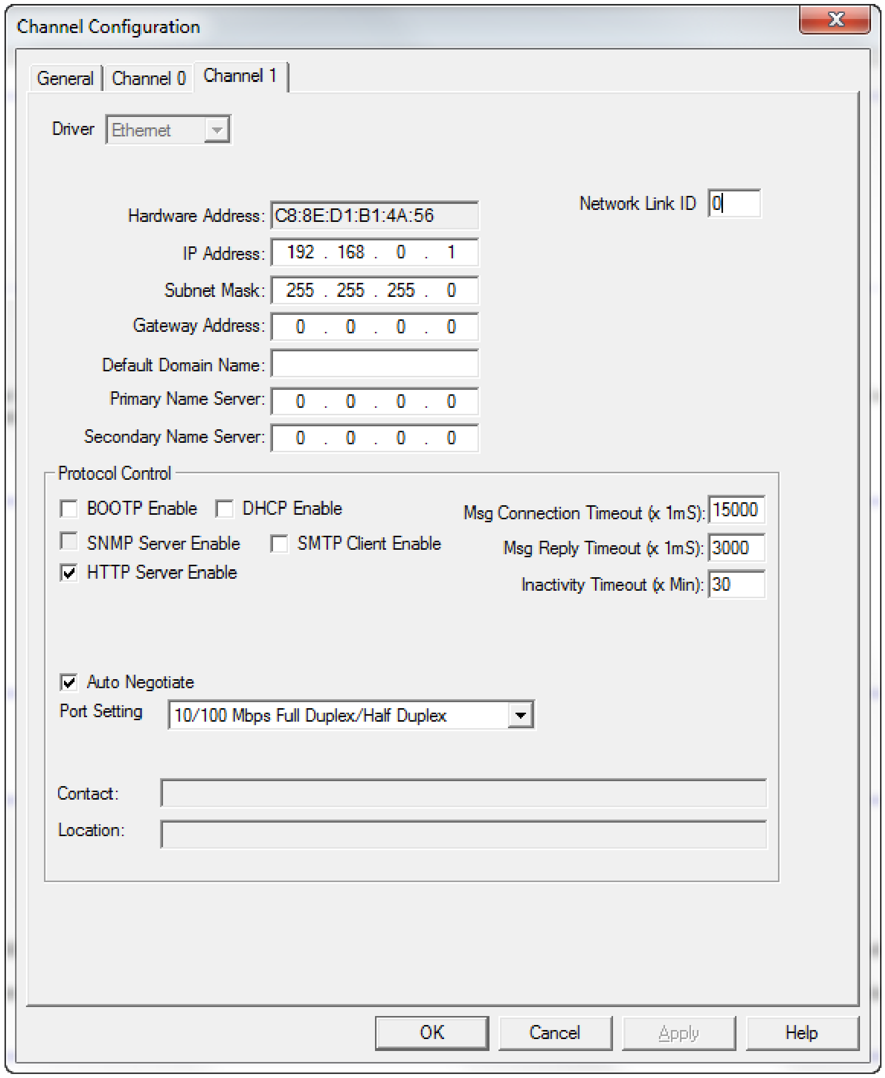


Figure T5.6 MicroLogix Ethernet Configuration Screen

2) Enter the IP address and Subnet Mask of your MicroLogix 1100, (not the address of the SMD23E) and click on [Apply]. The Ethernet Port should now be working.

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

Notes

TASK 6

MODBUS TCP DATA ADDRESSING

An SMD23E-B that has been configured for Modbus TCP protocol requires a host controller to issue configuration data and motion commands. This chapter tell you how the I/O words used by an SMD23E-B are mapped to the Modbus I/O registers.

6.1 Enable Modbus TCP Protocol

All SMD23E-B units are factory configured to use the EtherNet/IP protocol. The AMCI Net Configurator utility can be used to change the protocol to Modbus TCP. This is typically done while setting the IP address. Specifically, follow the steps in section 2.2b, ***Use the AMCI NET Configurator Utility*** which starts on page 92.

6.2 Modbus Addressing Overview

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 Modbus TCP packets.

NOTE All AMCI manuals use zero based addressing.

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 Data Table Mapping

The Modbus data model contains Discrete Input and Input Register tables. These addresses map to the same physical memory locations in the SMD23E-B units.

- These registers hold data that is reported from the SMD23E-B to the host controller. This data is typically command responses and status data.
- Addresses for these data locations 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 Modbus data model also contains Coil and Holding Register tables. These addresses map to the same physical memory locations in the SMD23E-B units.

- These registers hold data that is transfered from the host controller to the SMD23E-B units. This data is configuration or command data.
- 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 16400 through 16415.

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

6.2 Modbus Addressing Overview (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 must define a mapping between your host controller's addressing scheme and the zero based addresses used in this manual when you add the SMD23E-B to your host controller. Refer to your host controller's documentation for information on how to accomplish this.

6.3 SMD23E-B Memory Layout

The SMD23E-B has a starting Input Register address of 0 and a starting Output Register address of 1024. Input Registers hold the data from the SMD23E-B while Output Registers hold the data to be written to the unit. Figure T6.1 shows how an SMD23E-B is mapped to the Modbus data reference. The complete specification for the Modbus protocol can be downloaded at http://www.modbus.org/specs.php.

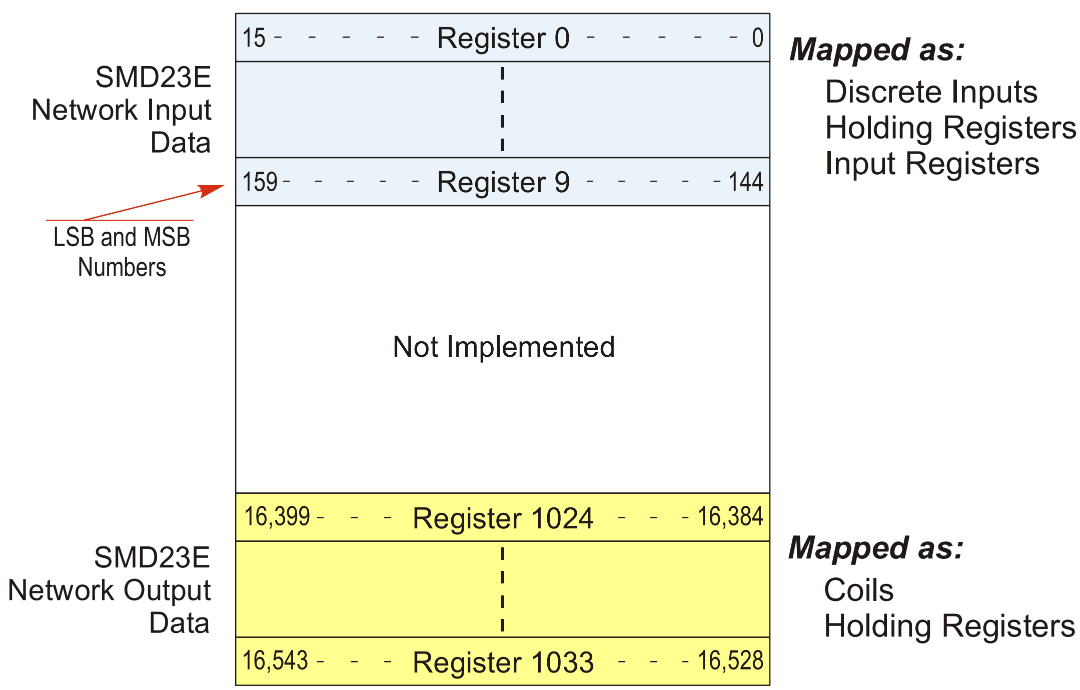


Figure T6.1 Modbus Data Reference Map: SMD23E-B

6.4 Supported Number of Connections

All SMD23E-B units support four 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	SMD23E-B 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 SMD23E-B units. AMCI supports all of these functions so that you can control the SMD23E-B as you see fit. Most applications will use FC4 to read all ten input registers and FC16 to write to all ten output registers.

NOTE Each SMD23E-B buffers the data that it sends over the network. If you use the *Read/Write Registers* function (FC23) to write data to the unit, then the data read with that command will not contain the response to the new 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

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