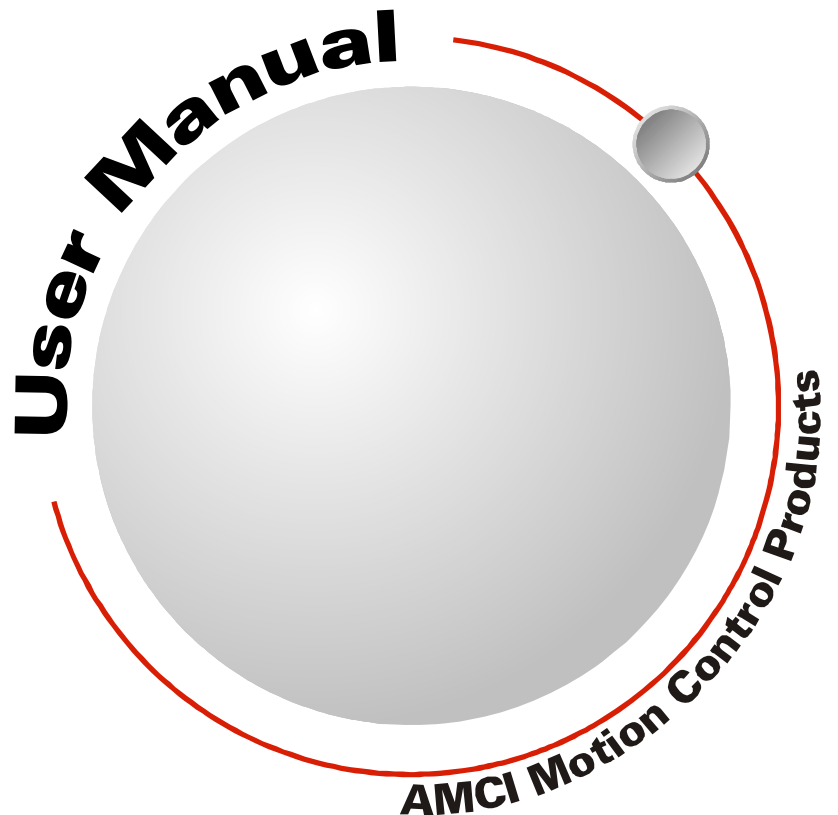


# **SMD23P & SMD24P Integrated Stepper Indexer/Driver/Motor with PROFIBUS Interface**

**SMD23P-130-M12  
SMD23P-240-M12  
SMD23P-130E-M12  
SMD23P-240E-M12  
SMD23P-130-M12P  
SMD23P-240-M12P  
SMD23P-130E-M12P  
SMD23P-240E-M12P  
SMD24P-350-M12  
SMD24P-350E-M12  
SMD24P-350-M12P  
SMD24P-350E-M12P**



# GENERAL INFORMATION

## ***Important User Information***

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

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

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

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

This manual is copyright 2014 by Advanced Micro Controls Inc. You may reproduce this manual, in whole or in part, for your personal use, provided that this copyright notice is included. You may distribute copies of this complete manual in electronic format provided that they are unaltered from the version posted by Advanced Micro Controls Inc. on our official website: [www.amci.com](http://www.amci.com). You may incorporate portions of this documents in other literature for your own personal use provided that you include the notice "Portions of this document copyright 2014 by Advanced Micro Controls Inc." You may not alter the contents of this document or charge a fee for reproducing or distributing it.

## ***Standard Warranty***

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

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

## ***Returns Policy***

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

## ***24 Hour Technical Support Number***

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

If you require additional technical support, call (860) 583-7271. 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.

## ***We Want Your Feedback***

Manuals at AMCI are constantly evolving entities. Your questions and comments on this manual are both welcomed and necessary if this manual is to be improved. Please direct all comments to: Technical Documentation, AMCI, 20 Gear Drive, Terryville CT 06786, or fax us at (860) 584-1973. You can also e-mail your questions and comments to [techsupport@amci.com](mailto:techsupport@amci.com)

# TABLE OF CONTENTS

## **General Information**

Important User Information .....	2
Standard Warranty .....	2
Returns Policy .....	2
24 Hour Technical Support Number .....	2
We Want Your Feedback .....	2

## **ABOUT THIS MANUAL**

Audience .....	7
Applicable Units .....	7
Revision Record .....	7
Revision History .....	7
Navigating this Manual .....	8
Manual Conventions .....	8
Trademark Notices .....	8
Where To Go From Here .....	9

## **Chapter 1: Introduction to the SMD23P and SMD24P**

The SMD23P and SMD24P Families .....	11
General Functionality .....	12
Encoder Functionality .....	12
Indexer Functionality .....	13
Driver Functionality .....	14
Available Discrete Inputs .....	14
Home Input .....	14
CW Limit Switch or	
CCW Limit Switch .....	14
Start Indexer Move Input .....	15
Emergency Stop Input .....	15
Stop Jog or	
Registration Move Input .....	15
Capture Encoder Position Input .....	15
General Purpose Input .....	15
Status LED's .....	16
SMD23/4P Connectors .....	16
Power & Input Connector .....	16
PROFIBUS Connector .....	17
Compatible Connectors and Cordsets .....	17
Connectors .....	17
PROFIBUS Cordsets .....	18
Cordsets .....	18
Cables .....	18
Power Supply Sizing .....	18
Specifications .....	19
Torque and Power Curves .....	20

## **Chapter 2: Move Profiles**

Units of Measure .....	23
Definition of Count Direction .....	23
Definition of Home Position .....	23
Definition of Starting Speed .....	23
Definition of Target Position .....	24
Relative Coordinates .....	24
Absolute Coordinates .....	24
Definition of Acceleration Types .....	24
Linear Acceleration .....	24
Triangular S-Curve	
Acceleration .....	25
Trapezoidal S-Curve	
Acceleration .....	25
A Simple Move .....	26
Controlled and Immediate Stops .....	27
Host Control .....	27
Hardware Control .....	27
Basic Move Types .....	27
Relative Move .....	27
Controlled Stop .....	28
Immediate Stop .....	28
Absolute Move .....	28
Controlled Stop .....	29
Immediate Stop .....	29
±Jog Move .....	29
Controlled Stop .....	30
Immediate Stop .....	30
±Registration Move .....	31
Controlled Stop .....	32
Immediate Stop .....	32
Encoder Moves .....	32
Controlled Stop Conditions .....	33
Immediate Stop Conditions .....	33
Assembled Moves .....	33
Blend Moves .....	33
Controlled Stop .....	34
Immediate Stop .....	34
Dwell Move .....	35
Controlled Stop .....	35
Immediate Stop .....	s 35
Assembled Move Programming .....	36
Control Bits – Output Data .....	36
Control Bits – Input Data .....	36
Programming Routine .....	36
Saving an Assembled	
Move in Flash .....	36

## Chapter 2: Move Profiles (continued)

Indexed Moves .....	37
Controlling Moves In Progress .....	38
Find Home Moves .....	38
Jog Moves .....	38
Registration Moves .....	38
Absolute, Relative and	
Encoder Moves .....	38
Assembled Moves .....	38
Stall Detection with	
SMD23/4P_E Units .....	39
Idle Current Reduction .....	39

## Chapter 3: Homing the SMD23/4P

Definition of Home Position .....	41
Position Preset .....	41
±Find Home Commands .....	41
Homing Inputs .....	41
Physical Inputs .....	41
Backplane Inputs .....	41
Homing Configurations .....	42
Homing Profiles .....	42
Home Input Only Profile .....	42
Profile with Backplane	
Home Proximity Bit .....	43
Profile with Overtravel Limit .....	44

## Chapter 4: Installing the SMD23/4P

Location .....	45
Safe Handling Guidelines .....	45
Prevent Electrostatic Damage .....	45
Prevent Debris From	
Entering the Unit .....	45
Remove Power Before Servicing	
in a Hazardous Environment .....	45
Mounting .....	46
SMD23P Outline Drawing .....	46
SMD24P Outline Drawing .....	47
SMD23/4P-M12 Mounting .....	47
SMD23/4P-M12P Mounting .....	47
Connecting the Load .....	47
Network Connector .....	48
Profibus Terminators .....	48

## Chapter 4: Installing the SMD23/4P (continued)

Compatible Wiring .....	48
Cordsets .....	48
Cables .....	48
Connectors .....	49
Power & Input Connector .....	49
Power Wiring .....	50
Input Wiring .....	51

## Chapter 5: Setting the PROFIBUS Address

Required Equipment .....	53
RS-485 Port Configuration .....	53
Cabling .....	54
Computer Configuration .....	55
Connecting to the SMD23/4P .....	56
Using the PROFIBUS-DP	
Configuration Tools Utility .....	56

## Chapter 6: Host Configuration

Basic Steps .....	59
Download the GSD file .....	59
GSD File Installation .....	59
Configure the PROFIBUS Network .....	60
Adding the SMD23/4P to the	
PROFIBUS Network .....	61

## Chapter 7: Configuration Mode Data Format

Modes of Operation .....	63
Configuration Mode .....	63
Command Mode .....	63
Multi-Word Format .....	63
Output Data Format .....	64
Control Word Format .....	64
Configuration Word Format .....	66
Notes on Other	
Configuration Words .....	67
Input Data Format .....	67
Control Word Format (Word 0) ....	67
Invalid Configurations .....	68

# Table of Contents

---

## **Chapter 8: Command Mode Data Format**

Multi-Word Format .....	69
Command Bits Must Transition .....	69
Output Data Format .....	69
Command Bits MSW .....	70
Command Bits LSW .....	71
Command Blocks .....	73
Absolute Move .....	73
Relative Move .....	73
Hold Move .....	74
Resume Move .....	74
Immediate Stop .....	75
+Find Home (CW) .....	75
-Find Home (CCW) .....	76
+Jog Move (CW) .....	76
+Registration Move (CW) .....	77
-Jog Move (CCW) .....	77
-Registration Move (CCW) .....	78
Preset Position .....	78
Reset Errors .....	79
Run Assembled Move .....	79
Preset Encoder Position .....	80
Programming Blocks .....	81
First Block .....	81
Segment Block .....	81
Input Data Format .....	82
Status Bits MSW Format .....	82
Status Bits LSW Format .....	84
Motor Position Data .....	85
Encoder Position Data .....	85
Captured Encoder Position Data ...	85
Notes on Clearing a Driver Fault .....	85

*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 AMCI's SMD23/4P and SMD24P families of Integrated Stepper Indexer/Driver/Motors with PROFIBUS interface. 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 all of the units in the SMD23/4P family.

Model Number	Description
SMD23/4P-130-M12	Size 23 motor, 130 oz-in holding torque with sealed M12 connectors for IP50 rating.
SMD23/4P-240-M12	Size 23 motor, 240 oz-in holding torque with sealed M12 connectors for IP50 rating.
SMD23/4P-130E-M12	Same as SMD23/4P-130-M12 with integral encoder.
SMD23/4P-240E-M12	Same as SMD23/4P-240-M12 with integral encoder.
SMD23/4P-130-M12P	Same as SMD23/4P-130-M12 with FDA approved clear coating, shaft and wire seals for IP67 rating.
SMD23/4P-240-M12P	Same as SMD23/4P-240-M12 with FDA approved clear coating, shaft and wire seals for IP67 rating.
SMD23/4P-130E-M12P	Same as SMD23/4P-130E-M12 with FDA approved clear coating, shaft and wire seals for IP67 rating.
SMD23/4P-240E-M12P	Same as SMD23/4P-240E-M12 with FDA approved clear coating, shaft and wire seals for IP67 rating.
SMD24P-350-M12	Size 24 motor, 350 oz-in holding torque with sealed M12 connectors for IP50 rating.
SMD24P-350E-M12	Same as SMD24P-350-M12 with integral encoder.
SMD24P-350-M12P	Same as SMD24P-350-M12 with FDA approved clear coating, shaft and wire seals for IP67 rating.
SMD24P-350E-M12P	Same as SMD24P-350E-M12 with FDA approved clear coating, shaft and wire seals for IP67 rating.

Part Number Description



In general, the only difference between the SMD23 and SMD24 units are the outline drawings. They are functionally equivalent. When information in this manual applies to all of the units in the families and the name "SMD23/4P" is used. When a part of this manual only applies to units with encoders, the name "SMD23/4P-E" is used.

## Revision Record

This manual, 940-0S140, is the initial release of this manual. It was first released March 25<sup>th</sup>, 2014.

### Revision History

940-0S140 First Release. March 25, 2014.

## ***Navigating this Manual***

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

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

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.

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

## ***Manual Conventions***

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



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



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



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

The following table shows the text formatting conventions:

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

## ***Trademark Notices***

The AMCI logo is a trademark, and “AMCI” is a registered trademark of Advanced Micro Controls Inc.

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



## ABOUT THIS MANUAL

---

### ***Where To Go From Here***

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

CHP Num.	Chapter Title	Intended Audience
1	<i>Introduction to the SMD23P and SMD24P</i>	Anyone new to the SMD23/4P. This chapter gives a basic overview of the features available on the unit, typical applications, and electrical specifications.
2	<i>Move Profiles</i>	Anyone that needs detailed information on how the unit can be used to control motion in your application.
3	<i>Homing the SMD23/4P</i>	Anyone that needs detailed information on how the home position of the SMD23/4P can be set.
4	<i>Installing the SMD23/4P</i>	Anyone that must install an SMD23/4P on a machine. Includes information on mounting, grounding, and wiring specific to the units.
5	<i>Setting the PROFIBUS Address</i>	Anyone that must set the PROFIBUS address of the SMD23/4P using the PROFIBUS-DP Configuration Tool software that is available from AMCI.
6	<i>Host Configuration</i>	Anyone needing guidance on configuring a PROFIBUS host controller to communicate with the SMD23/4P. A Siemens SIMATIC S7-317 CPU is used as an example host.
9	<i>Configuration Mode Data Format</i>	Anyone that needs detailed information on the format of the network data to and from the SMD23/4P that is used to configure the unit.
10	<i>Command Mode Data Format</i>	Anyone that needs detailed information on the format of the network data to and from the SMD23/4P that is used to command the unit.

***Notes***

## INTRODUCTION TO THE SMD23P AND SMD24P

This manual is designed to get you quickly up and running with the SMD23P and SMD24P lines of Integrated Stepper Indexer/Driver/Motors. 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 SMD23P and SMD24P Families***

The SMD23 and SMD24 are growing product lines from AMCI with a simple concept: a low-cost, stepper indexer, driver, and motor that can be attached to any popular industrial network. All Profibus SMD23P and SMD24P units use the Decentralized Periphery profile, "Profibus-DP", which is the most commonly used Profibus profile. Each SMD23/4P appears to be a communications adapter to the PROFIBUS Master.

There are presently twelve members of the SMD23/4P family. Each unit has M12 connectors for power, I/O, and PROFIBUS connections. The following six units have an IP50 environmental rating:

- SMD23P-130-M12: 130 oz-in motor
- SMD23P-240-M12: 240 oz-in motor
- SMD24P-350-M12: 350 oz-in motor
- SMD23P-130E-M12: 130 oz-in motor with built in encoder
- SMD23P-240E-M12: 240 oz-in motor with built in encoder
- SMD24P-350E-M12: 350 oz-in motor with built in encoder

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

Finally, if an IP67 rating is important, the units can be ordered with a shaft seal, wire potting, and an FDA approved clear epoxy by appending "P" to the part number.

- SMD23/4P-130-M12P
- SMD23/4P-240-M12P
- SMD24P-350-M12P
- SMD23/4P-130E-M12P
- SMD23/4P-240E-M12P
- SMD24P-350E-M12P



Figure 1.1 SMD23/4P



### ***The SMD23/4P and SMD24P Families (continued)***

#### **General Functionality**

Each member of the SMD23/4P family has three integrated parts:

- An indexer that accepts commands over a PROFIBUS-DP connection
- A 3.4Arms micro-stepping driver that accepts 24 to 48Vdc as its input power source
- A high torque size 23 (130 or 240 oz-in holding torque) or size 24 (350 oz-in holding torque) stepper motor

An incremental encoder is also available for applications that require additional position feedback.

The availability of the PROFIBUS connection makes the SMD23/4P 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 SMD23/4P 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 SMD23/4P 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 SMD23/4P 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 SMD23/4P to be very similar to these products.

The SMD23/4P 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 SMD23/4P automatically switches from RMS to peak current control when the motor is idle to prevent overheating the motor.

The SMD23/4P units have two DC inputs. 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 encoder position on SMD23/4P-E units.)
- Stop Manual or Registration Move Input
- Start Indexer Move
- Emergency Stop Input
- General Purpose Input

#### **Encoder Functionality**

All SMD23/4P units can be ordered with an internal 1,024 count/turn incremental encoder. Using an encoder gives you the ability to:

- Make absolute and relative moves based on the encoder count instead of the stepper pulse count
- Detect motor stall conditions

## ***Indexer Functionality***

The table below lists the functionality offered by the indexer built into the SMD23/4P.

Feature	Description
PROFIBUS-DP	Allows easy setup and communication with a wide range of host controllers.
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 Input, 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 SMD23/4P 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.
Encoder Move	SMD23/4P/E units that are ordered with an internal encoder can perform moves based on the encoder position instead of the motor position. The encoder position sets the point that the move begins to decelerate.
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 memory. 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	When an SMD23/4P is purchased with the encoder option, the encoder can be used to verify motion when a move command is issued.

Table 1.1 Indexer Functionality



## Driver Functionality

This table summarizes the features of the stepper motor driver portion of an SMD23/4P.

Feature	Benefits
RMS Current Control	RMS current control give an SMD23/4P 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 an SMD23/4P 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 SMD23/4P sets a warning bit in the network data when the temperature of the unit approaches its safe operating threshold.
Over Temperature Protection	Protects your SMD23/4P from damage by removing power from the motor if the internal temperature of the driver exceeds a safe operating threshold.

Table 1.2 Driver Functionality

## Available Discrete Inputs

The SMD23/4P has two 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 SMD23/4P uses these inputs is fully programmable as well as their active states. 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 SMD23/4P 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 SMD23/4P that will cause the unit to seek this sensor. How the SMD23/4P actually finds the Home sensor is described in the [Homing the SMD23/4P](#) chapter starting on page 41.

### 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.

### ***Available Discrete Inputs (continued)***

#### **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 SMD23/4P to run critically timed moves that cannot be reliably started from the network due to data transfer lags.

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

#### **Emergency Stop Input**

When an input is defined as an Emergency Stop, or E-Stop Input, motion will immediately stop when this input becomes active. The driver remains enabled and power is supplied to the motor. In addition to stopping moves in progress, moves 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 available, the quadrature encoder position data will be captured when the DC input makes an inactive-to-active transition if it is configured as a Stop Jog or Registration Move Input. The encoder position data is not captured if a Jog or Registration Move is not in progress. If you want to capture encoder position data on every transition of a DC input, configure it as a Start Indexer Move Input.

#### **Capture Encoder Position Input**

As described in the [\*Start Indexer Move Input\*](#) and [\*Stop Jog or Registration Move Input\*](#) sections above, an SMD23/4P can be configured to capture the encoder position value on a transition on 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 SMD23/4P, but the input state is reported in the network data so their state can be monitored by your host controller.

## Status LED's

As shown in figure 1.2 below, the NR25 has two status LED's on the rear cover. The tables below list the various states of the LED's and their meaning.

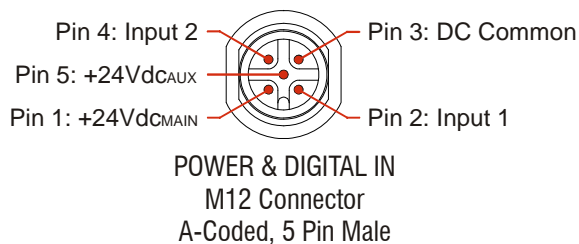
NS LED (BUS State)	MS LED (DEVICE State)	Definition
Off	Off	No Power
Off	On	Normal Operation / Data Exchange Mode
On	Off	1) No communications 2) SMD23/4P has not received data for an extended period of time. Possible bus line interruption.
On	On	NR25 is operational, but has not received any configuration data since power up. Possible bus line interruption or wiring error.
On	Flashing	Parameter or Configuration Error. Invalid data or incorrect data length was received by the unit.
Flashing	On	New address received from Profibus Master.
Synchronous Flashing		Address cannot be read from Flash Memory on power up. Default address of 97 is presently being used.
Alternate Flashing		Fatal Internal Error. Cycle power to the unit. If the error persists, contact AMCI for further assistance.

Table 1.3 Network Status LED States

## SMD23/4P Connectors

### Power & Input Connector

As shown in figure 1.2, the 'Power & Digital Input' Connector is located on the back of the unit below the PROFIBUS 'Network' connector. All digital input and power supply connections are made at this connector. Figure 1.3 shows the pinout of the connector when viewed from the back of the SMD23/4P.



*View shows the front of the unit's connector, which is the same view as the back of its mating connector.*

Figure 1.3 M12 Power and Digital Input Connector

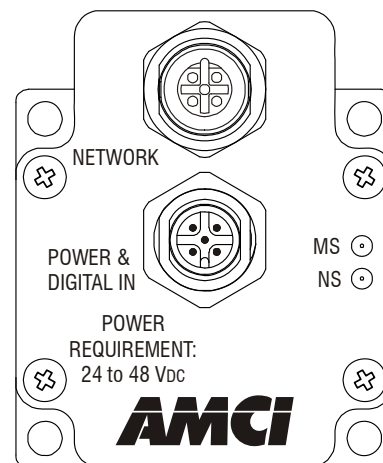


Figure 1.2 SMD23/4P Rear Cover

Digital inputs on the SMD23/4P units are single ended and referenced to the DC Common pin. There are two power pins.  $DCPower_{MAIN}$  powers both the control electronics and the motor.  $DCPower_{AUX}$  powers only the control electronics. Using the  $DCPower_{AUX}$  pin is optional. If your application requires you to cut power to your motor under some conditions, using the  $DCPower_{AUX}$  pin allows you to cut power to your motor without losing your network connection. The connector is a standard five pin A-coded M12 connector that is rated to IP67 when the mate is properly attached.

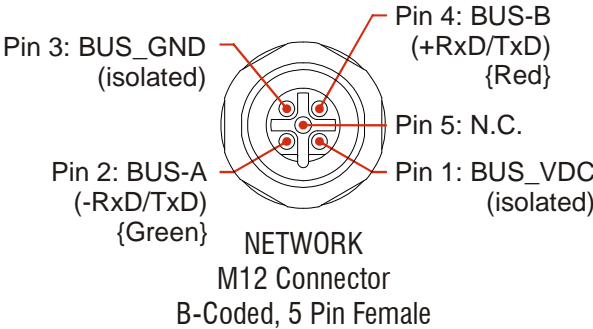
- 18 AWG wire should be used for the  $VdC_{MAIN}$  and DC Common connections.



## SMD23/4P-M12 Connectors (continued)

### PROFIBUS Connector

Figure 1.2 also shows the placement of the sealed PROFIBUS Connector, while figure 1.4 shows the connector pinout when viewed from the back of the SMD23/4P. The connector is a standard five pin B-coded female M12 connector that is rated to IP67 when the mate is properly attached.



*View shows the front of the unit's connector, which is the same view as the back of its mating connector.*

Figure 1.4 M12 PROFIBUS Connector Pinout

## Compatible Connectors and Cordsets

Many different connectors and cordsets are available on the market, all of which will work with the SMD23/4P provided that the manufacturer follows the connector and Ethernet standards. AMCI has reviewed the following connectors and PROFIBUS cordsets for compatibility with the SMD23/4P.

### Connectors

AMCI #	Binder #	Description
MS-39	99-0437-105-05	NETWORK mating connector, B-coded, male, straight. Screw terminal connections. 6 to 8 mm dia. cable. Plastic body. IP67 rated when properly installed.
	99-0437-145-05	NETWORK mating connector, B-coded, male, right-angle. Screw terminal connections. 6 to 8 mm dia. cable. Plastic body. IP67 rated when properly installed.
MS-31	99-0436-12-05	Mating connector for Power Connector. Female, 5 pin A-coded. Screw terminal connections. 6 to 8 mm dia. cable. Straight, plastic body. IP67 rated when properly installed.
	99-0436-52-05	Mating connector for Power Connector. Female, 5 pin A-coded. Screw terminal connections. 6 to 8 mm dia. cable. Right angle, plastic body. IP67 rated when properly installed.

Table 1.4 Compatible Connectors



### ***Compatible Connectors and Cordsets (continued)***

#### **PROFIBUS Cordsets**

There are many different suppliers of Profibus interconnections. Installing a pre-assembled and tested cordset is the most common way to wire a Profibus system. For those companies wishing to make their own cables, cable and connectors are also available from a wide variety of suppliers. A small sampling of companies is given below.

##### ***Cordsets***

- Phoenix Contact: <http://www.phoenixcontact.com>
- Turck Inc.: <http://www.turck.com> or <http://www.turck-usa.com>
- Lumberg Automation: <http://www.lumberg-automationusa.com>

##### ***Cables***

- Belden Inc. <http://www.belden.com> (Belden part # 3079A or 3079E are suggested)
- Phoenix Contact: <http://www.phoenixcontact.com> (Phoenix Contact part # 2744652)

### ***Power Supply Sizing***

An external 24 to 48Vdc power supply is required to power the control electronics and the stepper motor, so it must be rated to supply current to motor. The general rule of thumb is to specify a supply that can output the maximum current for your motor. The maximum motor current for the SMD23/4P is 3.4 Arms, but this can be limited under software configuration.

By using the torque curves of your motor, it is possible to calculate the maximum power your motor will output and size your supply accordingly. If you use power calculations, double the calculated power when sizing your supply to account for power conversion losses and guarantee stall-free operation.

SMD23/4P-M12 units have an additional pin that supplies power to the electronics only. This allows you to remove power from the motor without losing the network connection.

- 18 AWG wire should be used for the VdcMAIN and DC Common connections.



## INTRODUCTION TO THE SMD23P AND SMD24P

### Specifications

#### Driver Type

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

#### Physical Dimensions

See page 46 for the SMD23P  
See page 47 for the SMD24P

#### Weight

SMD23/4P-130(E)-M12(P) .. 2.30 lbs. (1.05 kg.)  
SMD23/4P-240(E)-M12(P) .. 2.80 lbs. (1.27 kg.)

All weights are without mating connectors

#### Inputs

Electrical Characteristics: .....  
Single ended sinking. 2500 Vac/dc  
opto-isolated.

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

#### 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 3.4Arms in 0.1 Amp  
steps.

#### DCPower<sub>AUX</sub> Current

70 mA @ 24Vdc, 40mA @48Vdc

#### Motor Counts per Turn

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

#### Incremental Encoder

Internal incremental encoder available as an option.  
Encoder 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 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 ..... SMD23/4P-M12: IP50

..... SMD23/4P-M12P: IP67

#### Status LED's

See *Status LED's* section starting on page 16.

#### Connectors

All mating connectors are available separately  
under the following AMCI part numbers.

Connector	AMCI Part #	Wire	Strip Length	Connection Type
Power	MS-31	18 AWG max.	0.197 inches	Screw Terminals
Network	MS-39	18 AWG max.	0.197 inches	Screw Terminals



## Torque and Power Curves

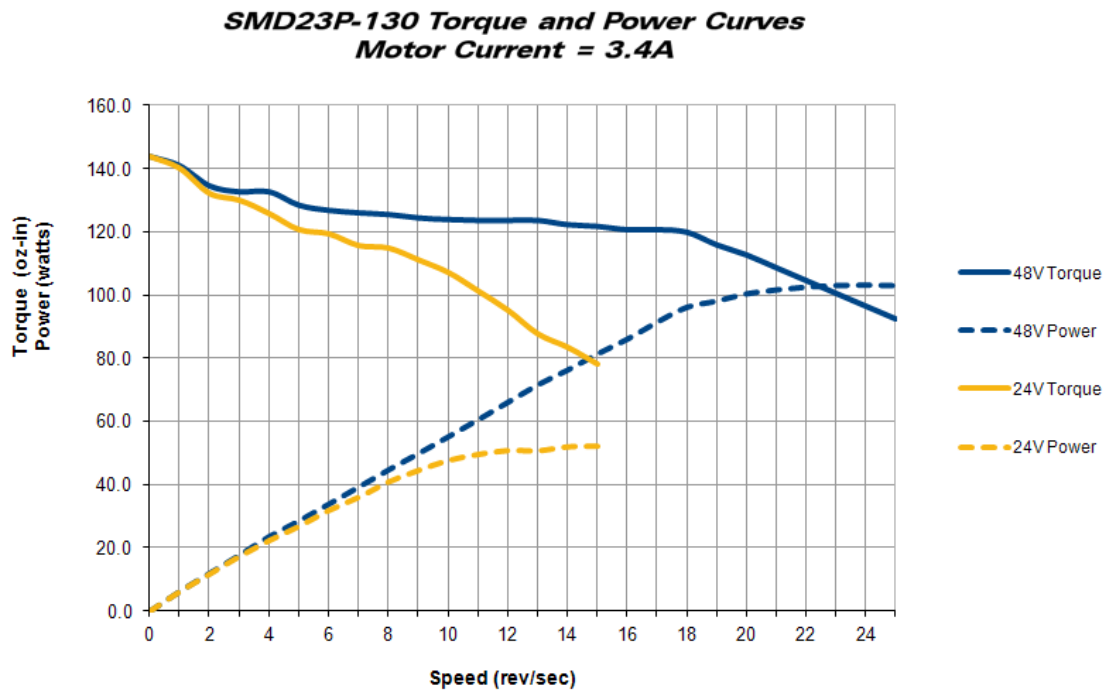


Figure 1.5 SMD23P-130 Torque and Power Curves

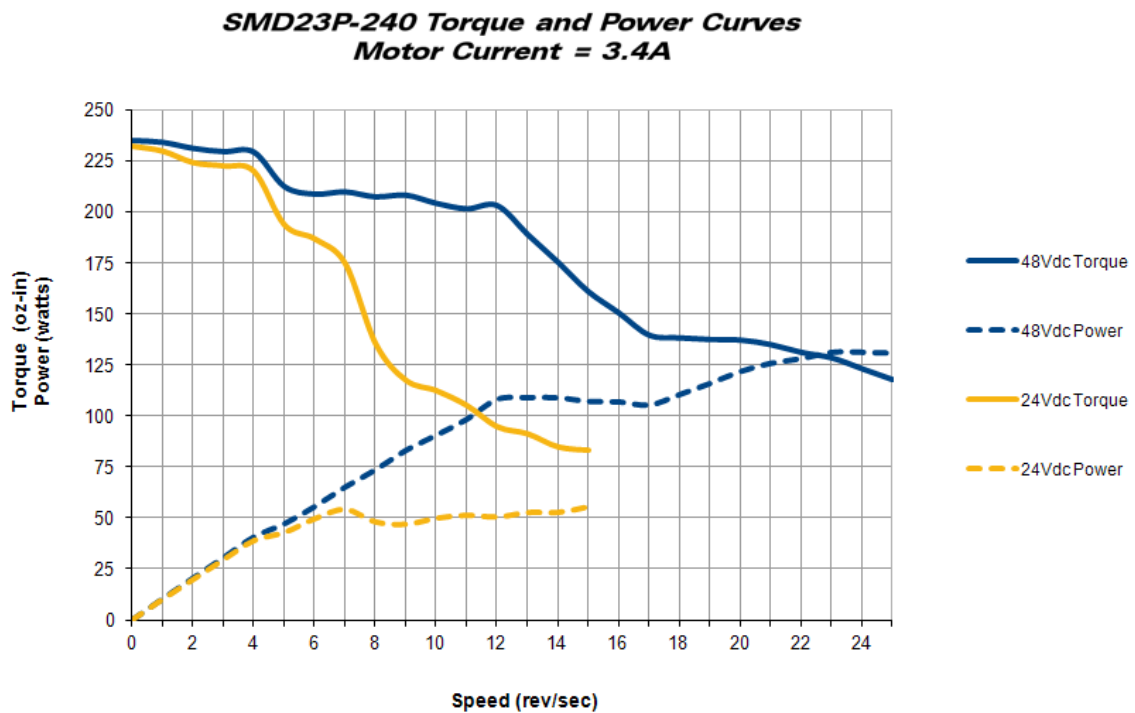


Figure 1.6 SMD23P-240 Torque and Power Curves

## Torque and Power Curves (continued)

### *SMD24P-350 Torque and Power Curves* *Motor Current = 3.4A*

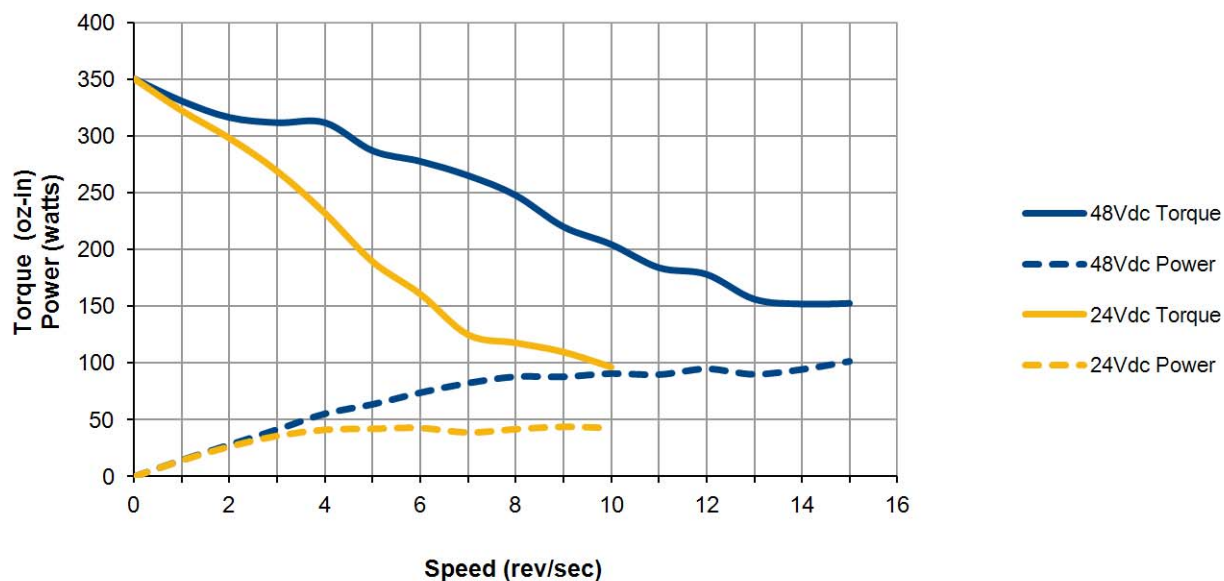


Figure 1.7 SMD24P-350 Torque and Power Curves



***Notes***

## CHAPTER 2

# MOVE PROFILES

When a move command is sent to an SMD23/4P, the unit calculates the entire profile before starting the move or issuing an error message. This chapter explains how the SMD23/4P calculates profiles and the different move types that are available. Appendix A, Calculating Move Profiles, provides example profile calculations if your application requires precise move profiles.

### ***Units 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<sup>2</sup>. However, when programming an SMD23/4P, all acceleration and deceleration values must be programmed in the unit of measure of steps/millisecond/second.

- To convert from steps/second<sup>2</sup> to steps/millisecond/second, divide the value by 1000. This must be done when converting from a value used in the equations to a value programmed into an SMD23/4P.
- To convert from steps/millisecond/second to steps/second<sup>2</sup>, multiply the value by 1000. This must be done when converting from the value programmed into an SMD23/4P to the value used in the equations.

### ***Definition of 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.

### ***Definition of 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  $\pm$ Find Home commands. If you use the unit's  $\pm$ Find Home commands, the motor position and encoder registers will automatically be set to zero once the home position is reached. Defining a Home Position is completely optional. Some applications, such as those that use the SMD23/4P for speed control, don't require position data at all.

### ***Definition of Starting Speed***

The Starting Speed is the speed that 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. In micro-stepping applications, this setting also limits the amount of time needed for acceleration and deceleration. AMCI does not specify a default value because it is very dependent on motor size and attached load.

## Definition of 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 SMD23/4P moves use relative coordinates.

- The range of values for the Target Position when it is treated as an offset is  $\pm 8,388,607$  counts. Positive offsets will result in clockwise moves, while negative offsets result in counter-clockwise moves.
- The Current Motor Position that is reported back to the host has a range of -32,768,000 to 32,767,999 counts. The only way to move beyond  $\pm 8,388,607$  counts is with multiple relative moves or jog commands.

### Absolute Coordinates

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

- The range of values for the Target Position when it is treated as an actual position on the machine is  $\pm 8,388,607$  counts. The move will be clockwise (positive) if the Target Position is greater than the Current Position and counter-clockwise (negative) if the Target Position is less than the Current Position.
- The Current Motor Position register that is reported back to the host has a range of -32,768,000 to 32,767,999 counts. However, you cannot move beyond  $\pm 8,388,607$  counts with an Absolute Move. The only way to move beyond  $\pm 8,388,607$  counts is with multiple relative moves or jog commands.

## Definition of Acceleration Types

With the exception of Registration Moves, all move commands, including homing commands, allow you to define the acceleration type used during the move. The SMD23/4P 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. For the smoothest transition from the starting speed, the starting speed should be equal to the square root of the acceleration in steps/sec<sup>2</sup>. For example, if the choose acceleration is 20,000 steps/sec<sup>2</sup>, the smoothest transition occurs when the starting speed is 141. ( $141^2 \approx 20,000$ ). Note that other values will work correctly, but you may notice a quick change in velocity at the beginning of the acceleration phase.

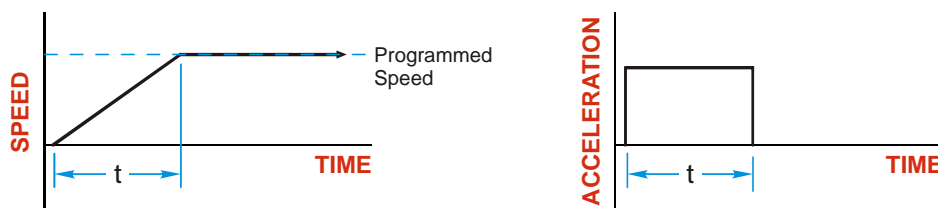


Figure 2.1 Linear Acceleration



## Definition of Acceleration Types (continued)

### Triangular S-Curve Acceleration

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

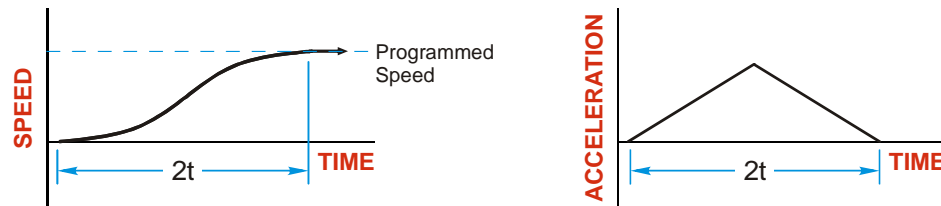


Figure 2.2 Triangular S-Curve Acceleration

### 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 2.3 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.

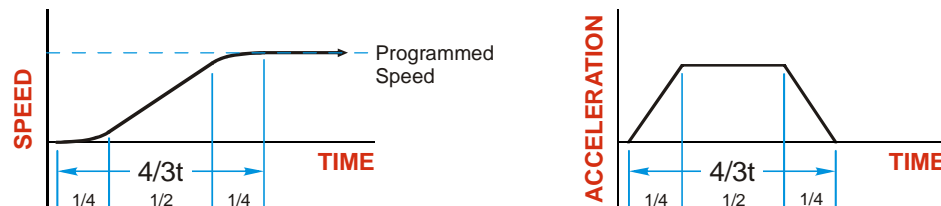


Figure 2.3 Trapezoidal S-Curve Acceleration

## A Simple Move

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

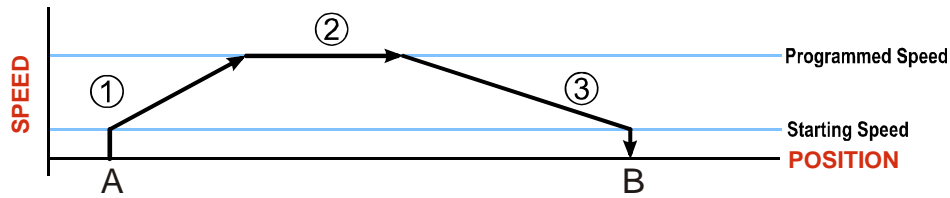


Figure 2.4 A Trapezoidal Profile

- 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*.
- 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* until the speed reaches the *Starting Speed*, which occurs at the Target Position (B). The motor stops at this point.
  - The *Starting Speed* parameter is programmed when the SMD23/4P is configured by the host controller. The *Acceleration Value*, *Deceleration Value*, and *Programmed Speed* parameters are programmed when the move command is sent to the SMD23/4P.

Figure 2.4 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 2.5 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* may never be reached.

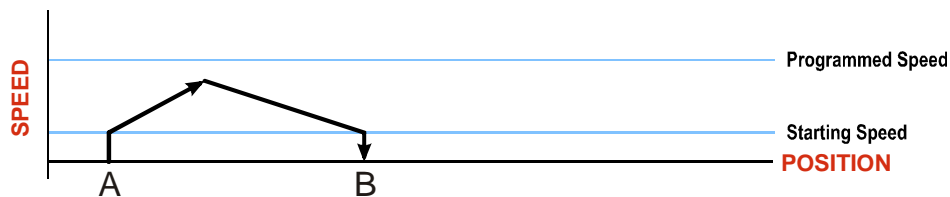


Figure 2.5 A Triangular Profile

## Controlled and Immediate Stops

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

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

## Host Control

**Hold Move Command:** This command can be used with some moves to bring the axis to a Controlled Stop. The move can be resumed and finished, or it can be aborted. Not all moves are affected by this command. The section *Basic Move Types*, starting on page 27, describes each move type in detail, including if the move is affected by this command.

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

## Hardware Control

**CW Limit and CCW Limit Inputs:** In most cases, activating these inputs during a move will bring the axis to an Immediate Stop. The exceptions are the  $\pm$ Find Home commands, the  $\pm$ Jog Move commands, and the  $\pm$ Registration Move commands. The  $\pm$ Find Home commands are explained in chapter 3, *Homing the SMD23/4P*, which starts on page 41. The  $\pm$ Jog Move commands are fully explained on page 29, and the  $\pm$ Registration Move commands are fully explained on page 31.

**Emergency Stop Input:** It is possible to configure an input as an Emergency Stop Input. When an Emergency Stop Input is activated, the axis will come to an Immediate Stop, regardless of the direction of travel. Note that power is not removed from the motor.

## Basic Move Types

### Relative Move

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

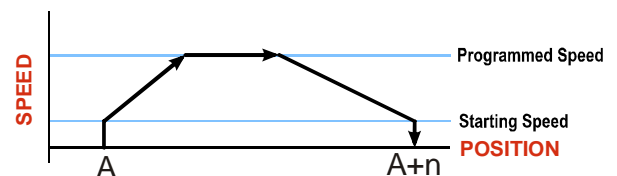


Figure 2.6 Relative Move

### NOTE

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

## Basic Move Types (continued)

### Relative Move (continued)

Relative Moves can be brought to a Controlled Stop by using the Hold Move Command from your host controller. When the command is accepted, the axis will immediately decelerate at the programmed rate and stop. When stopped successfully, the SMD23/4P will set a *Hold State* bit in the input data table. The Relative Move can be restarted with the Resume Move command from the host controller or the move can be aborted by starting another move. The Resume Move command allows you to change the move's Programmed Speed, Acceleration Value and Type, and the Deceleration Value and Type. The Target Position cannot be changed with the Resume Move Command.

#### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will not be the final position you commanded unless you toggle the Hold Move control bit during the deceleration stage of the move. You can resume a held Relative Move by using the Resume Move command. The use of the Hold Move and Resume Move bits is further explained in the [Controlling Moves In Progress](#) section starting on page 38.

#### Immediate Stop Conditions

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

### Absolute Move

Absolute Moves move from the Current Position (A) to a given position (B). (The SMD23/4P calculates the direction and the number of steps needed to move to the given position and moves that number of steps.) A trapezoidal profile is shown to the right, but Absolute Moves can also generate triangular profiles. The command's Target Position can be in the range of  $\pm 8,388,607$  counts. The move will be clockwise if the Target Position is greater than the Current Position and counter-clockwise if the Target Position is less than the Current Position.

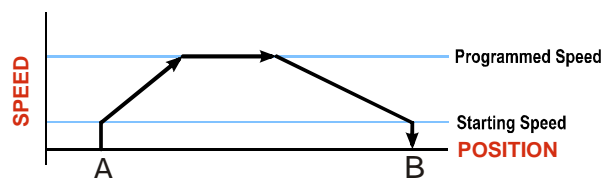


Figure 2.7 Absolute Move

#### NOTE

- 1) The *Home Position* of the machine must be set before running an Absolute Move. See chapter 3, [Homing the SMD23/4P](#), which starts on page 41, for information on homing the machine.
- 2) The Motor Position must be valid before you can use an Absolute Move. The Motor Position becomes valid when you preset the position or home the machine.
- 3) Absolute Moves allow you to move your machine without having to calculate relative positions. If you are controlling a rotary table, you can drive the table to any angle without having to calculate the distance to travel. For example an Absolute Move to 180° will move the table to the correct position regardless of where the move starts from.

### **Basic Move Types (continued)**

#### **Absolute Move (continued)**

##### *Controlled Stop Conditions*

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will not be the final position you commanded unless you toggle the Hold Move control bit during the deceleration stage of the move. You can resume a held Absolute Move by using the Resume Move bit or the move can be aborted by starting another move. The use of the Hold Move and Resume Move bits is explained in the [Controlling Moves In Progress](#) section starting on page 38.

##### *Immediate Stop Conditions*

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

#### **±Jog Move**

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



The Target Position value must be zero during a Jog Move. If it is not zero, you will perform a [±Registration Move](#), and the final stopping position may be unexpected.

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 CW +Jog can be issued while the CCW limit switch is active. This allows you to move off of an activated end limit switch.

## Basic Move Types (continued)

### ± Jog Move (continued)

As shown below, a Jog Move begins at the programmed Starting Speed, accelerates at the programmed rate to the Programmed Speed and continues until a stop condition occurs. If it is a *Controlled Stop Condition*, the SMD23/4P 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 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. Figure 2.8 below shows an example of a Jog Move where the speed and acceleration values are changed while the move is in progress.

- Once the move is initiated with the Programmed Speed greater than the Starting Speed, changing the Programmed Speed to a value less than the Starting Speed will cause the SMD23/4P to issue an invalid parameter error. The Jog Move will continue to run at the last valid speed.

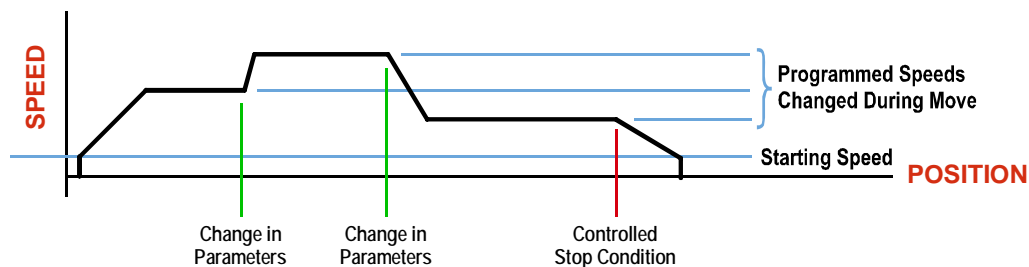


Figure 2.8 Jog Move

### Controlled Stop Conditions

- The Jog Move Command bit is reset to “0”.
- An inactive-to-active transition on an input configured as a *Stop Jog or Registration Move Input*.
- You toggle the Hold Move control bit in the Network Output Data. The use of the Hold Move and Resume Move bits is explained in the [Controlling Moves In Progress](#) section starting on page 38.

### Immediate Stop Conditions

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



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

## Basic Move Types (continued)

### ±Registration Move

Similar to a Jog Move, a Registration Move will travel in the programmed direction as long as the command is active. +Registration Moves result in clockwise rotation of the shaft, –Registration Moves result in counter-clockwise rotation of the shaft. When the command terminates under Controlled Stop conditions, the SMD23/4P 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.

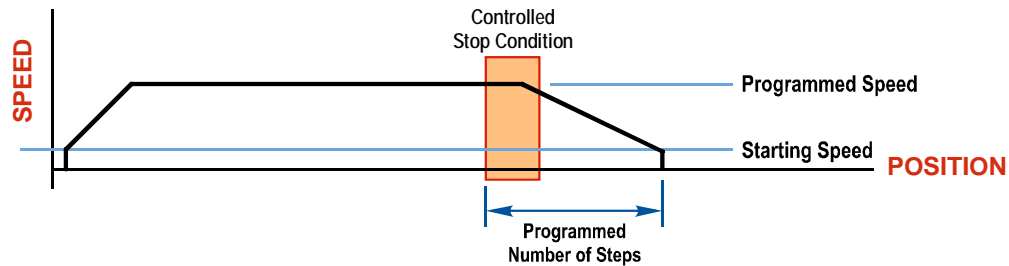


Figure 2.9 Registration Move

**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 SMD23/4P 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 SMD23/4P 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 has been reached.

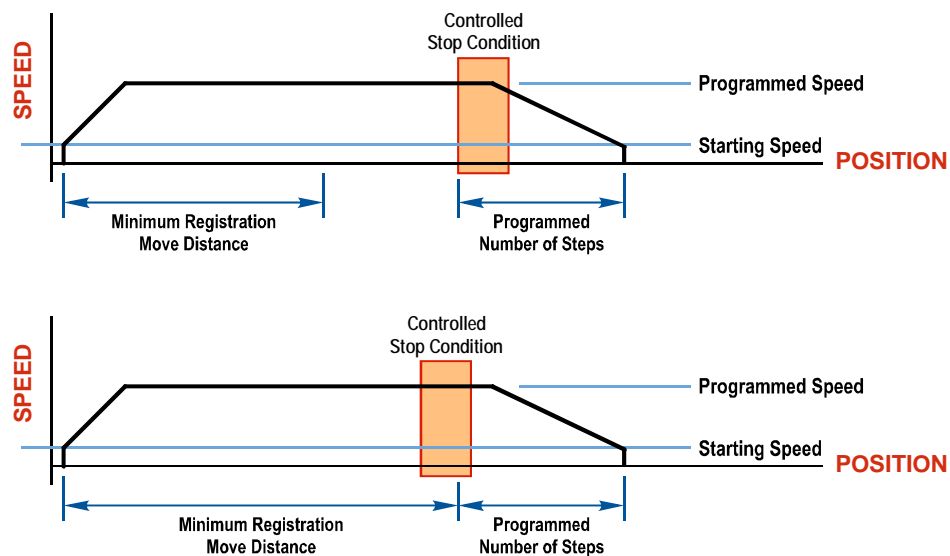


Figure 2.10 Min. Registration Move Distance



## Basic Move Types (continued)

### ±Registration Move (continued)

#### Controlled Stop Conditions

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

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

- You toggle the Hold Move control bit in the Network Output Data. The SMD23/4P responds by using the programmed Deceleration value to bring the move to a stop, without using the value of the Programmed Number of Steps parameter. A Registration Move does not go into the Hold State if the Hold Move control bit is used to stop the move and it cannot be restarted.

#### Immediate Stop Conditions

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

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

## Encoder Moves

When an SMD23/4P is ordered with an integral quadrature encoder, (SMD23/4P-xxxE part numbers), the position value from the encoder can be used to control the move instead of the position of the motor. Absolute and relative type moves are both supported.

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

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

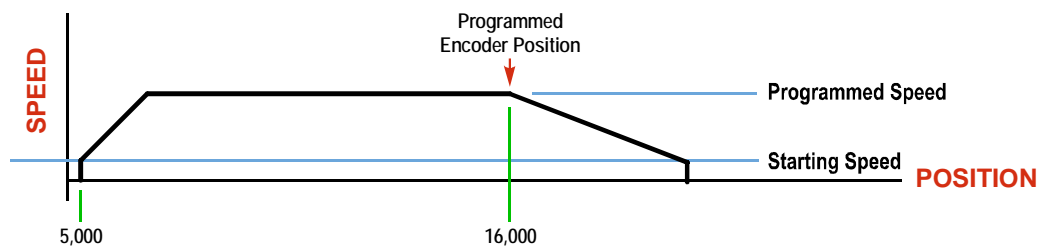


Figure 2.11 Encoder Move



## Encoder Moves (continued)

### Controlled Stop Conditions

- The move completes without error
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Encoder Move by using the Resume Move bit or the move can be aborted by starting another move. The use of the Hold Move and Resume Move bits is explained in the [Controlling Moves In Progress](#) section starting on page 38.

### Immediate Stop Conditions

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

## Assembled Moves

All of the moves explained so far must be run individually to their completion or must be stopped before another move can begin. The SMD23/4P 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 SMD23/4P:

- **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 SMD23/4P will stop between each move for a programmed *Dwell Time*. Because motion stops between each segment, a Dwell Move allows you to reverse direction during the move.

### 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 SMD23/4P before the move can be initiated.
  - The SMD23/4P supports Blend Moves with up to sixteen segments.
- 2) Each segment is programmed as a relative move. Blend Moves cannot be programmed with absolute coordinates.
- 3) All segments run in the same direction. The sign of the target position is ignored and only the magnitude of the target position is used. The move's direction is controlled by the bit pattern used to start the move. If you want to reverse direction during your move, consider using the [Dwell Move](#) which is explained on page 35.
- 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.

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.

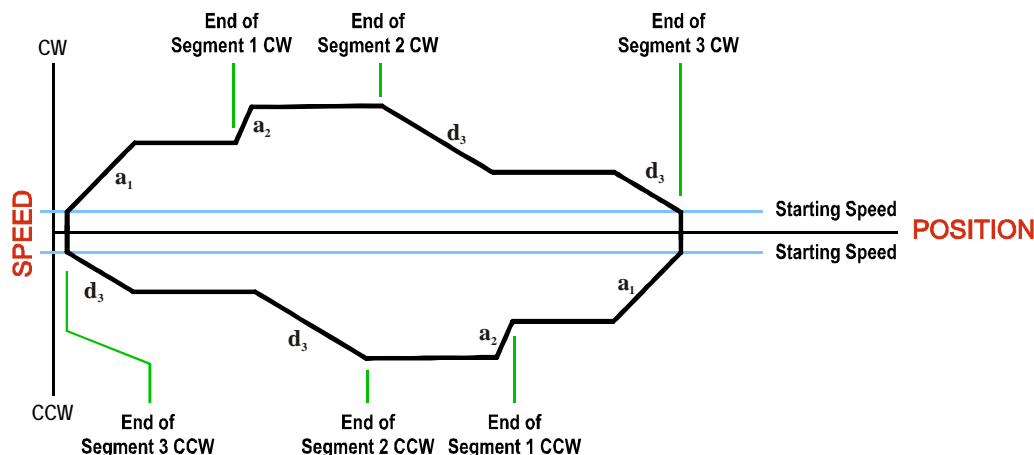


Figure 2.12 Blend Move

#### NOTE

- 1) You do not have to preset the position or home the machine before you can use a Blend Move. Because the Blend Move is based on Relative Moves, it can be run from any location.
- 2) The Blend Move is stored in the internal memory of the SMD23/4P 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 SMD23/4P. As described in [Saving an Assembled Move in Flash](#) on page 36, 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 enter Command Mode.
- 3) In addition to the bit used to command the move, there is a control bit that specifies which direction the Blend Move is run in.

### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. When this occurs, the SMD23/4P decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. A Blend Move that is brought to a controlled stop with the Hold Move bit cannot be restarted. The use of the Hold Move bit is explained in the [Controlling Moves In Progress](#) section starting on page 38.

### Immediate Stop Conditions

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

## Assembled Moves (continued)

### Dwell Move

A Dwell Move gives you the ability to string multiple relative moves together and run all of them sequentially with a single start condition. Like a Blend Move, a Dwell Move is programmed into an SMD23/4P 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. 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 operation so chips can be relieved from the bit.

You can accomplish this Dwell Move with a series of six relative moves that are sent down to the SMD23/4P sequentially. The two advantages of a Dwell Move in this case are that the SMD23/4P will be more accurate with the Dwell Time then you can be in your control program, and Dwell Moves simplify your program's logic.

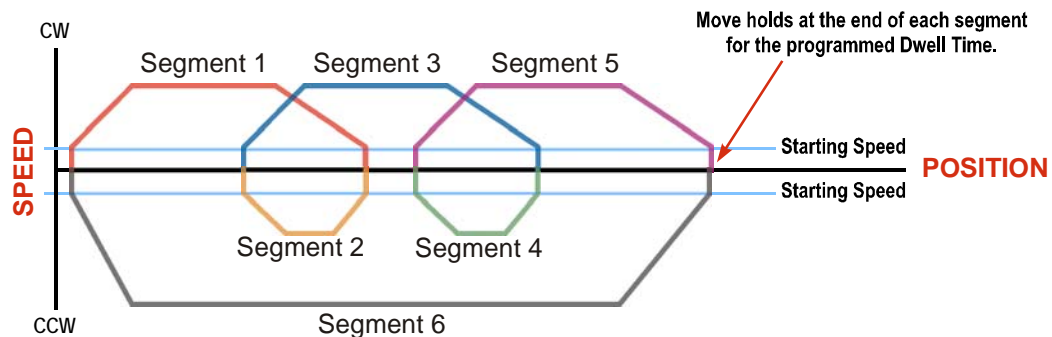


Figure 2.13 Dwell Move

#### 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 SMD23/4P 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 SMD23/4P. As described in [Saving an Assembled Move in Flash](#) on page 36, 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 SMD23/4P and enter Command Mode.

### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. When this occurs, the SMD23/4P decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. A Dwell Move that is brought to a controlled stop with the Hold Move bit cannot be restarted.

### Immediate Stop Conditions

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



## ***Assembled Move Programming***

All of the segments in a Blend or Dwell Move must be written to the SMD23/4P 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 SMD23/4P as part of the command.

### **Control Bits – Output Data**

- **Program Move bit** – Set this bit to tell the SMD23/4P that you want to program a Blend or Dwell Move Profile. The SMD23/4P will respond by setting the *Move Program Mode* bit in the Network Input Data. At the beginning of the programming cycle, the SMD23/4P will also set the *Transmit Move Segment* bit to signify that it is ready for the first segment.
- **Program Move Segment bit** – Set this bit to tell the SMD23/4P that the data for the next segment is available in the remaining data words.

### **Control Bits – Input Data**

- **Move Program Mode bit** – The SMD23/4P sets this bit to tell you that it is ready to accept segment programming data in the output data words. The actual transfer of segment data is controlled by the *Program Move Segment* and *Transmit Move Segment* bits.
- **Transmit Move Segment bit** – The SMD23/4P will set this bit to signal the host that it is ready to accept the data for the next segment.

### **Programming Routine**

- 1) The host sets the *Program Move* bit in the Network Output Data.
- 2) The SMD23/4P responds by setting both the *Move Program Mode* and *Transmit Move Segment* bits in the Network Input Data.
- 3) When the host detects that the *Transmit Move Segment* bit is set, it writes the data for the first segment in the Network Output Data and sets the *Program Move Segment* bit.
- 4) The SMD23/4P checks the data, and when finished, resets the *Transmit Move Segment* bit. If an error is detected, it also sets the *Command Error* bit.
- 5) When the host detects that the *Transmit Move Segment* bit is reset, it resets the *Program Move Segment* bit.
- 6) The SMD23/4P detects that the *Program Move Segment* bit is reset, and sets the *Transmit Move 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 Move* bit.
- 9) The SMD23/4P resets the *Move Program Mode* bit and the *Transmit Blend Move Segment* bit.

### **Saving an Assembled Move in Flash**

The SMD23/4P also contains the *Write Assembled Move* 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 *Write Assembled Move* bit set. When you reach step 9 in the sequence, the SMD23/4P responds by resetting the *Move Program Mode* and *Transmit Blend Move Segments* bits as usual and then it will flash the Status LED. If the LED is flashing green, the write to flash memory was successful. If it flashes red, then there was an error in writing the data. In either case, power must be cycled to the SMD23/4P before you can continue. With a limit of 10,000 write cycles, the design decision that requires you to cycle power to the SMD23/4P 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 SMD23/4P 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 SMD23/4P until the triggered move is finished.
- As stated above, a move can be run multiple times as long as the move command data remains unchanged. If you wish to program a second move and run it as an Indexed Move type, then you must have a 0→1 transition on the move command bit before the new parameters are accepted. The easiest way to accomplish this is by writing a value of 0x0000 to the command word between issuing move commands.
- A Jog Move that is started as an Indexed Move will come to a controlled stop when the command bit in the Network Output Data is reset to zero.
- It is possible to perform an Indexed Registration Move by configuring two inputs for their respective functions. The first input, configured as a *Start Indexed Move Input*, starts the move and the second, configured as a *Stop 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.

## Controlling Moves In Progress

An SMD23/4P has the ability to place a running move on hold and later resume the move if an error did not occur while the move was in its Hold state. One potential application for this feature is bringing a move to a controlled stop when your controller senses an end-of-stock condition. The move can be put in its Hold state until the stock is replenished and then the move can be resumed.

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

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

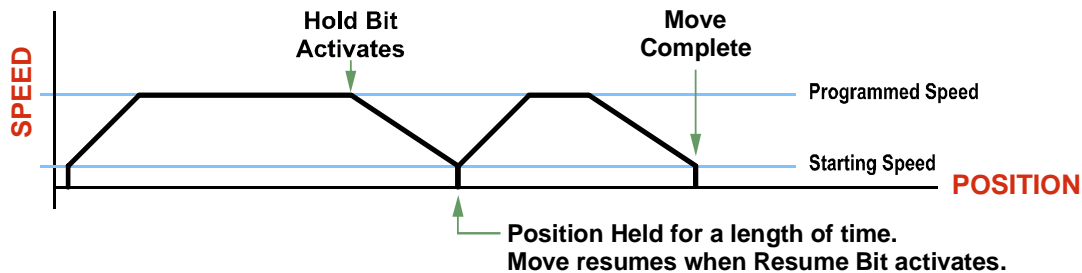


Figure 2.14 Hold/Resume a Move Profile

## Find Home Moves

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

## Jog Moves

Jog Moves can be placed in a Hold state and resumed if error conditions, such as programming errors, have not occurred. New Acceleration, Deceleration, and Programmed Speed parameters can be written to the SMD23/4P while a Jog Move is in its hold state. If these parameters are accepted without error, the move can be resumed and it will use the new parameter values.

## Registration Moves

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

## Absolute, Relative and Encoder Moves

Absolute, Relative and Encoder Moves can be placed in a Hold state and resumed if error conditions, such as programming errors, have not occurred. New Acceleration, Deceleration, and Programmed Speed parameters can be written to the SMD23/4P while these moves are in their hold states. If the parameters are accepted without error, the move can be resumed and it will use the new parameter values. Note that a change to the Target Position is ignored.

## Assembled Moves

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



### ***Stall Detection with SMD23/4P\_E Units***

One of the additional features available to you when you order an SMD23/4P with an encoder is *Stall Detection*. When Stall Detection is enabled, the SMD23/4P\_E 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 SMD23/4P\_E.

### ***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 0.5A<sub>pk</sub>. (The SMD23/4P always switches from RMS to peak current control when the motor is idle to prevent motor damage due to excessive heating.)

***Notes***




# HOMING THE SMD23/4P

---

This chapter explains the various ways of homing the SMD23/4P. 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 can be any position on your machine that you can sense and stop at. Once at the Home Position, the motor position register of an SMD23/4P must be set to an appropriate value. If you use the module's  $\pm$ 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 available and enabled.

**NOTE**  Defining a Home Position is completely optional. Some applications, such as those that use the SMD23/4P for speed control, don't require position data at all.

With the exception of Absolute Moves, an SMD23/4P can still perform all of its move commands if the Home Position is not defined.

### Position Preset

One of the ways to define the Home Position is to issue the Preset Position command to the SMD23/4P. 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 SMD23/4P with the Position Preset command. The motor and encoder position values can be preset anywhere in the range of -8,388,607 to +8,388,607.

### $\pm$ Find Home Commands

The other choice is to use the module's Find Home commands to order the SMD23/4P to find the Home Position based on sensors brought into the unit. The +Find Home command begins searching by rotating the motor shaft clockwise and ends when the home sensor triggers while the SMD23/4P is rotating clockwise at the starting speed. The -Find Home 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

- **Home Proximity Bit:** The SMD23/4P can be configured to ignore changes on the physical homing input until the Home Proximity Bit makes a 0→1 transition. The SMD23/4P 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 SMD23/4P must have one of its DC inputs configured as the home input before one of the  $\pm$  Find Home commands can be issued.

### NOTE

- 1) You do not have to configure and use CW or CCW Limits. If you choose to configure the module this way, then the SMD23/4P has no way to automatically prevent overtravel during a homing operation. You must prevent overtravel by some external means, or ensure that the homing command is issued in the direction that will result in reaching the homing input directly.
- 2) You can use a bit in the Network Output Data as a home proximity input. Using this bit is completely optional.

## Homing Profiles

### NOTE

The +Find Home command is used in all of these examples. The –Find Home command will generate the same profiles in the opposite direction.

### Home Input Only Profile

Figure 3.1 below shows the move profile generated by a +Find Home command when you use the Home Input without the Backplane Home Proximity bit.

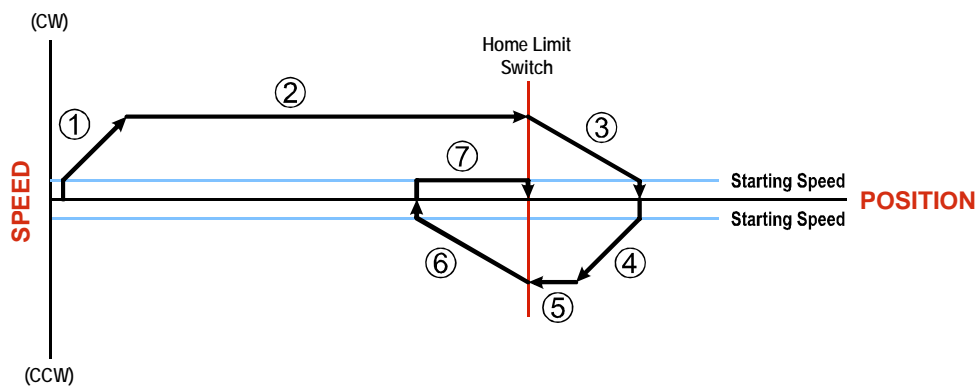


Figure 3.1 Home Input Profile

- 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 Backplane Home Proximity Bit

Figure 3.2 below shows the move profile generated by a +Find Home command when you use the Home Input with Backplane Home Proximity bit.

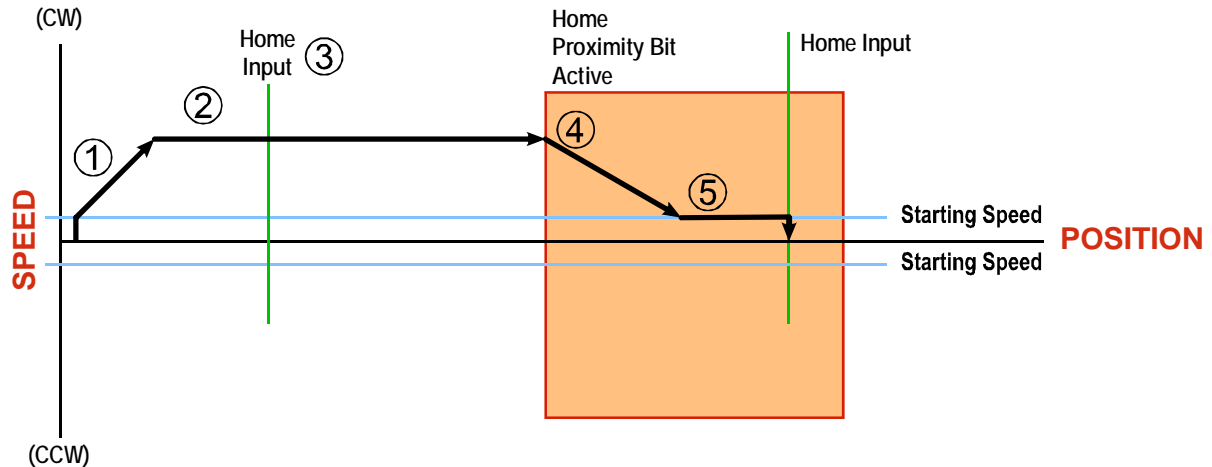


Figure 3.2 Homing with Proximity

- 1) Acceleration from the configured Starting Speed to the Programmed Speed
- 2) Run at the Programmed Speed
- 3) Ignores the Home Input because Home Proximity Bit has not made a 0→1 transition.
- 4) Deceleration towards the Starting Speed when the Home 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 3.2 shows the Backplane Home Proximity Bit staying active until the SMD23/4P reaches its home position. This is valid, but does not have to occur. As stated in step 4, the SMD23/4P starts to hunt for the home position as soon and the Backplane Home Proximity Bit makes a 0→1 transition

## Homing Profiles (continued)

### Profile with Overtravel Limit

Figure 3.3 below shows the move profile generated by a +Find Home command when you use:

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

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

The SMD23/4P will stop with an error if both overtravel limits are activated while the unit is trying to find the home position.

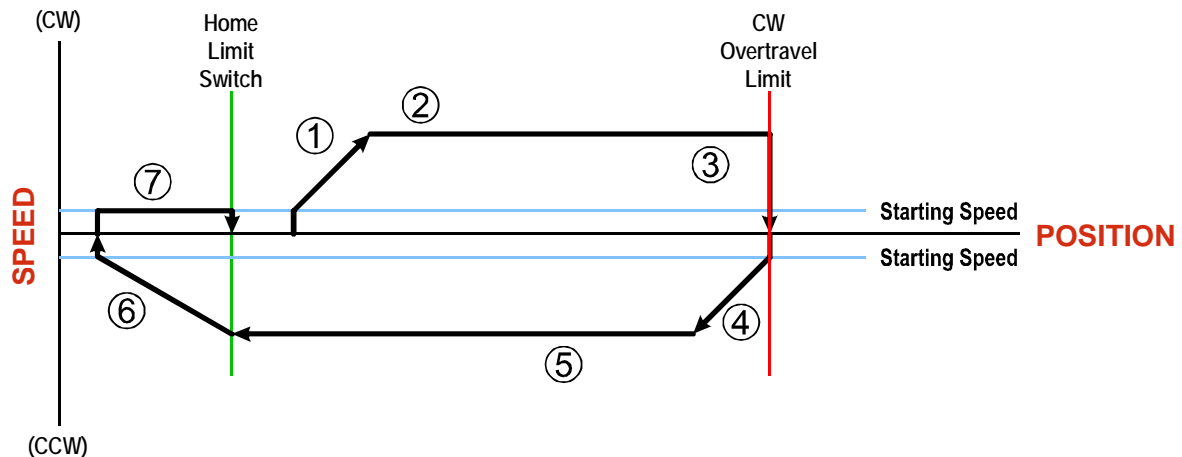


Figure 3.3 Profile with Overtravel Limit

- 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.



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

## CHAPTER 4

# INSTALLING THE SMD23/4P

This chapter gives detailed information on installing the SMD23/4P units. The person responsible for installing the SMD23/4P should be familiar with general installation practices.

### Location

SMD23/4P units are IP50 rated and are suitable for use in an industrial environment that meets 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).

SMD23/4P-M12P units are IP67 rated and are suitable for use in an industrial environment that meets the following criteria:

- Continuous conductivity occurs due to conductive dust, rain, or other wet conditions.
- 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 4* and *Over Voltage Category II* designations of the International Electrotechnical Commission (IEC).

### Safe Handling Guidelines

#### Prevent Electrostatic Damage



#### CAUTION

Electrostatic discharge can damage any SMD23/4P product if the discharge is directed at the I/O pins. 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) Do not touch the pins of the network connector or I/O connector.
- 4) Do not disassemble the unit.

#### 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, specifically the M12 connectors. Debris may cause damage to the unit or unintended machine operation with possible personal injury.

#### Remove Power Before Servicing in a Hazardous Environment



#### WARNING

Remove power before removing or installing any SMD23/4P units in a hazardous environment.

## Mounting

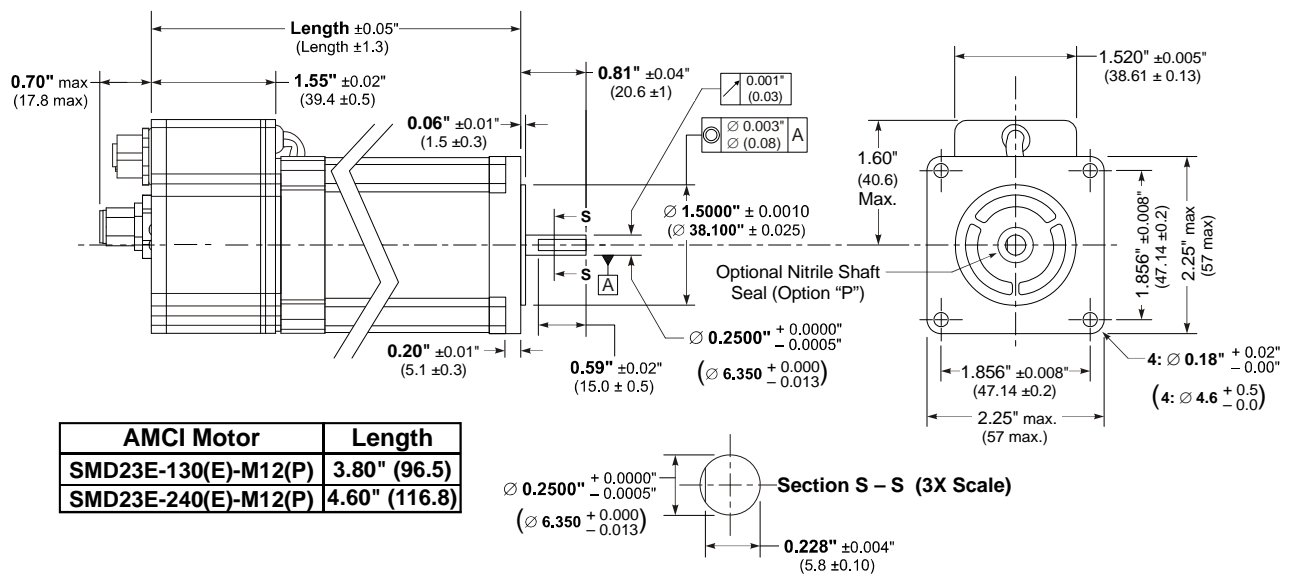
All AMCI motor have flanges on the front of the motor for mounting. This flange also acts as a heatsink, so motors should be mounted on a large, unpainted metal surface. Mounting a motor in this fashion will allow a significant amount of heat to be dissipated away from the motor, which will increase 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 SMD23/4P. Programming in an idle current reduction can also significantly reduce the temperature of the motor.

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

### NOTE

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

## SMD23P Outline Drawing



### REAR VIEW

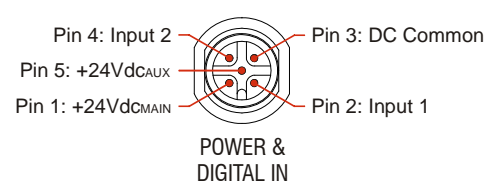
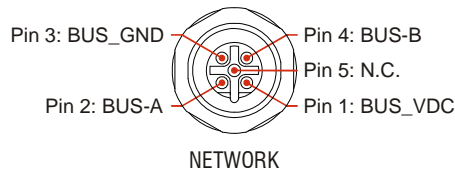
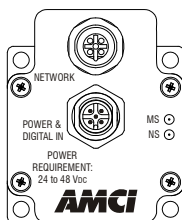
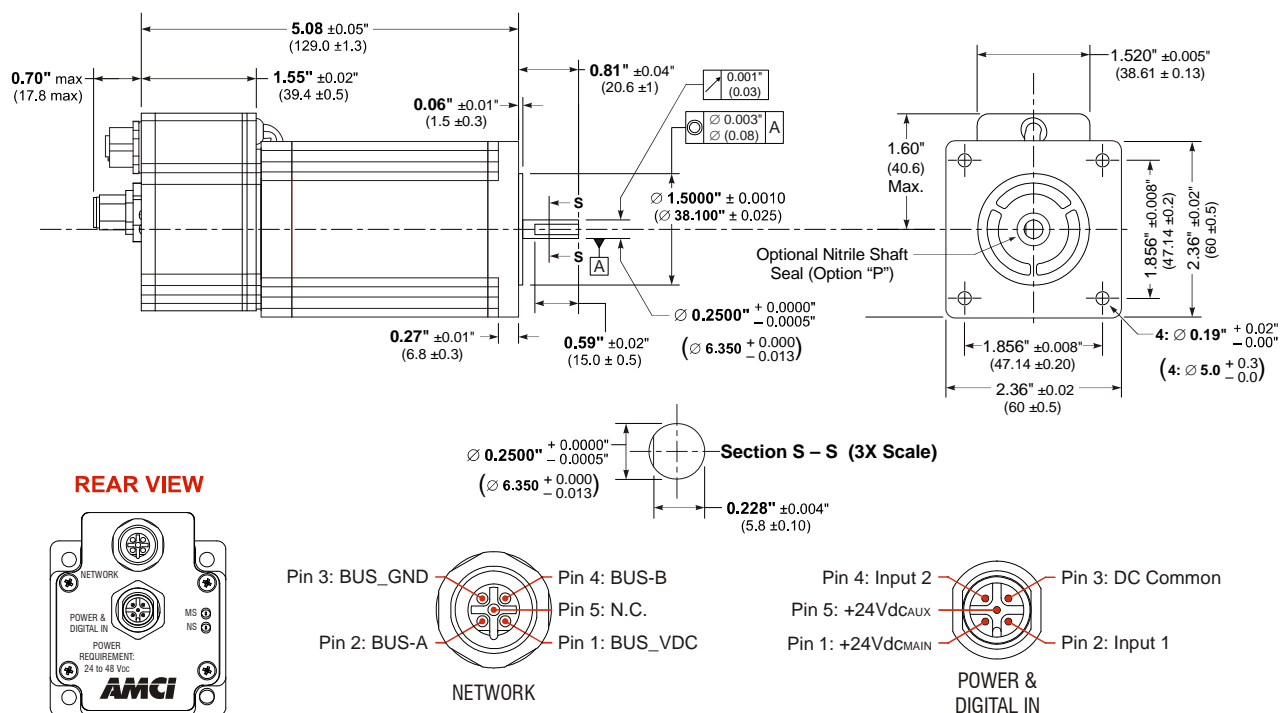


Figure 4.1 SMD23P Outline Drawing

## SMD24P Outline Drawing



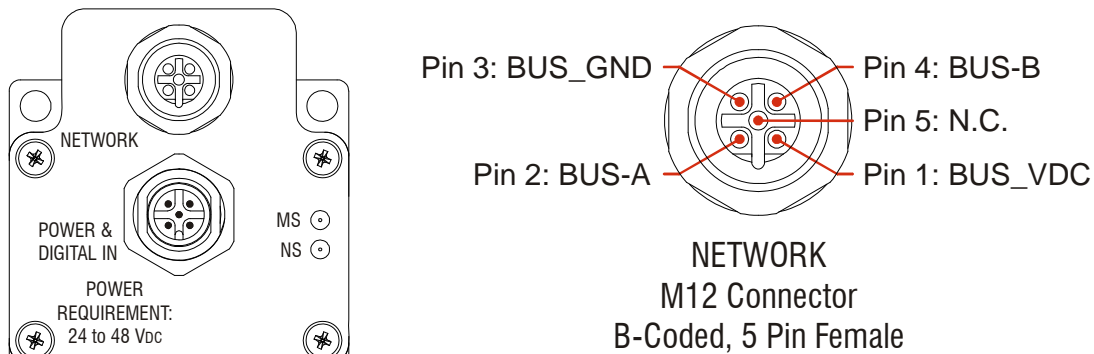
## SMD23/4P-M12 Mounting

## SMD23/4P-M12P Mounting

## Connecting the Load

## Network Connector

The diagram below shows the pinout of the SMD23/4P network connector.



*View shows the front of the unit's connector, which is the same view as the back of its mating connector.*

Figure 4.3 Network (BUS\_OUT) Connector Pinout

## Profibus Terminators

The Profibus must be terminated on both ends for proper operation. The network connector on the SMD23/4P has the additional signals required to terminate the Profibus network. The network can be terminated on the connector itself, but standard procedure is to terminate the network on a separate terminator so that devices can be removed from the network without disrupting the connections of the other devices.

## Compatible Wiring

There are many suppliers of Profibus interconnection products. Installing a pre-assembled and tested cordset is the most common way to wire a Profibus system. For those companies wishing to make their own cables, cable and connectors are also available from a wide variety of suppliers. A small sampling of companies is given below.

### Cordsets

- Phoenix Contact: <http://www.phoenixcontact.com>
- Turck Inc.: <http://www.turck.com> or <http://www.turck-usa.com>
- Lumberg Automation: <http://www.lumberg-automationusa.com>

### Cables

- Belden Inc. <http://www.belden.com> (Belden part # 3079A or 3079E are suggested)
- Phoenix Contact: <http://www.phoenixcontact.com> (Phoenix Contact part # 2744652)



# INSTALLING THE SMD23/4P

## Compatible Wiring (continued)

### Connectors

AMCI uses Binder connectors in our SMD23/4P units. The table below lists the Binder part number of the straight and right angle mating connectors for the network connector.

**NOTE** These mating connectors are “same source” suggestions, which means that you are not required to use them. Any industry standard M12 B-coded 5 pin connector will mate with the SMD23/4P network connectors.

AMCI #	Binder #	Description
MS-39	99-0437-105-05	Network mating connector, B-coded, male, straight. Screw terminal connections. 6 to 8 mm dia. cable. Plastic body. IP67 rated when properly installed.
	99-0437-145-05	Network mating connector, B-coded, male, right-angle. Screw terminal connections. 6 to 8 mm dia. cable. Plastic body. IP67 rated when properly installed.

Table 4.1 Suggested Mating Connectors - Network

### Power & Input Connector

The Input Connector is located on the back of the SMD23/4P below the network connector. Figure 4.4 shows the pin out for the Input Connector when viewed from the back of the unit.

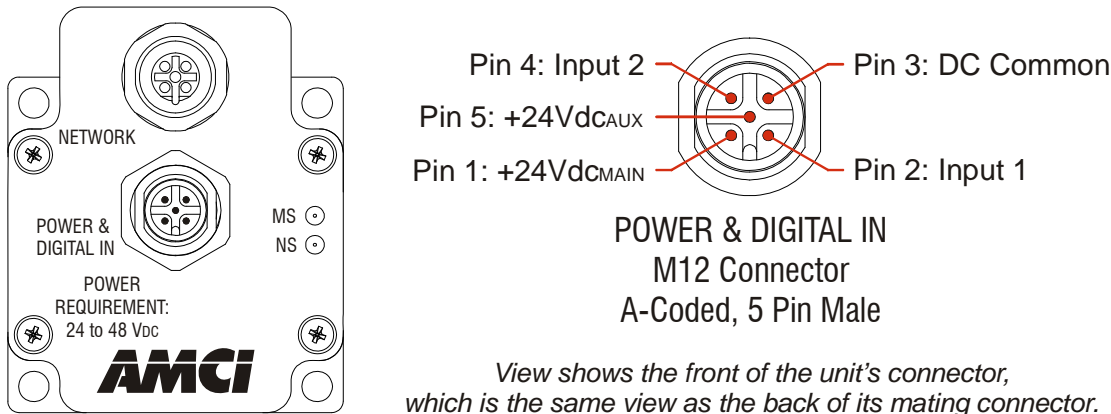


Figure 4.4 I/O Connector

Digital inputs on the SMD23/4P units are single ended and referenced to the DC Common pin. There are two power pins. DCPower<sub>MAIN</sub> powers both the control electronics and the motor. DCPower<sub>AUX</sub> powers only the control electronics. Using the DCPower<sub>AUX</sub> pin is optional. If your application requires you to cut power to your motor under some conditions, using the DCPower<sub>AUX</sub> pin allows you to cut power to your motor without losing your network connection. The connector is a standard five pin A-coded M12 connector that is rated to IP67 when the mate is properly attached.

Many different connectors and cordsets are available on the market, all of which will work with the SMD23/4P provided that the manufacturer follows the M12 A-coding standards. The following mating connector is available from AMCI.

AMCI #	Binder #	Description
MS-31	99-0436-12-05	Mating connector for Power Connector. Female, 5 pin A-coded. Screw terminal connections. 6 to 8 mm dia. cable. Straight, IP67 rated when properly installed.

Table 4.2 Compatible Input Connector

### Power Wiring

The SMD23/4P accepts 24 to 48Vdc as its input power. AMCI strongly suggests using 18 AWG or larger wire for the power connections. The MS-31 connector will accept up to 18 gauge wire.



#### CAUTION

Do not apply 120 Vac to any pins of the SMD23/4P. If this occurs, the unit will be damaged and you will void the unit's warranty.

Figure 4.5 below shows how to wire power to the SMD23/4P units. Note that Pin 5, DC Power<sub>AUX</sub>, is only used when you introduce a circuit for removing power from the motor.

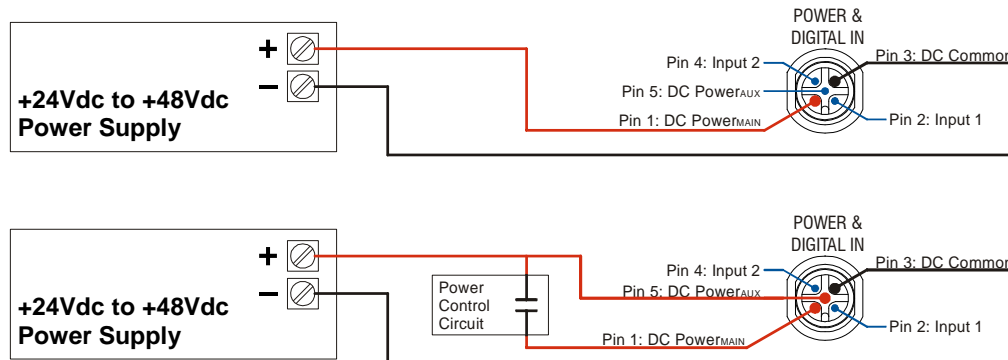


Figure 4.5 M12 Power Wiring

# INSTALLING THE SMD23/4P

## Input Wiring

Figure 4.6 below shows how to wire discrete DC differential, sourcing, and sinking sensors to Inputs 1 and 2 of the SMD23/4P.

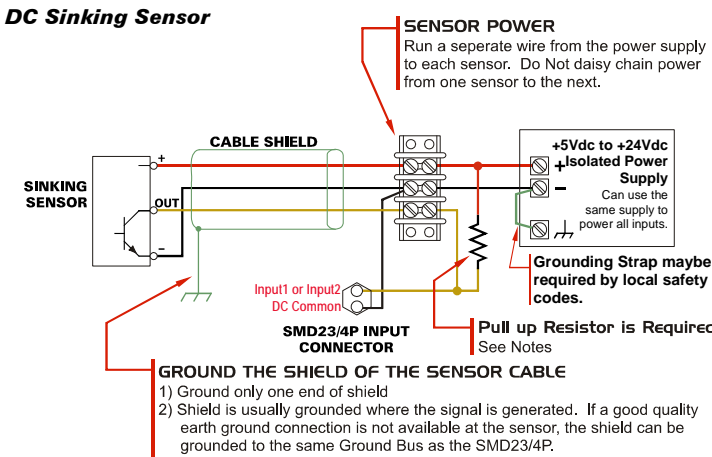
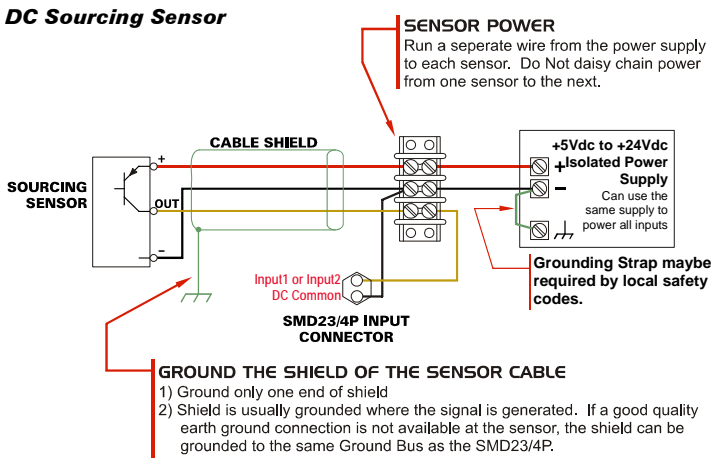
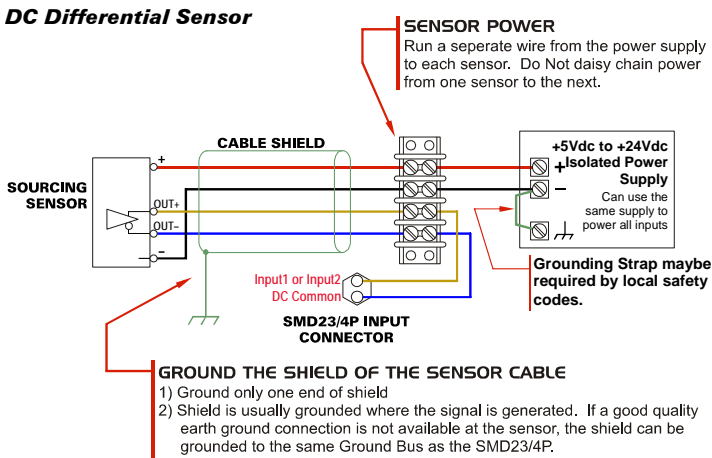


Figure 4.6 Input Wiring

## Input Specifications:

Single Ended, 2500 Vac/dc opto-isolated.<sup>†</sup>  
Accepts 3.5 to 27Vdc without the need for an external current limiting resistor.

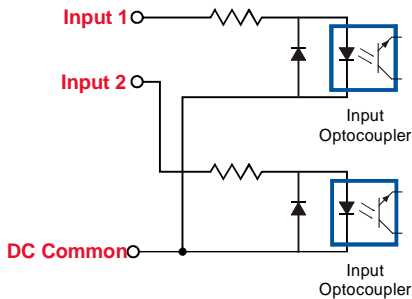


Figure 4.7 Simplified Input Schematic

Because they are low power signals, cabling from the sensor to the SMD23/4P 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 SMD23/4P.

## Sinking Output Sensors Require a Pull Up Resistor

Sinking output sensors require an external pull up resistor because the input to the SMD23/4P also sinks current. Table 4.3 below shows the values of pull up resistors that will allow the SMD23/4P input to activate along with the current that the sensor must be able to sink when it is active.

Input Voltage	Pull Up Resistor	Sensor Current When Active
5	300	16.7mA
12	1.4K	8.6mA
24	3.8K	6.3mA

Table 4.3 Pull Up Resistor

The logical states of the sensor and SMD23/4P input will be reversed. The SMD23/4P input is off when the sensor is active. You can set the logic state of the SMD23/4P input when you configure the unit.

<sup>†</sup> This ratings refer to the optocouplers ability to prevent damaging voltages from crossing its isolation barrier to the rest of the SMD23/4P. If 120 Vac voltage is applied to an input, the input itself will still be destroyed.

***Notes***

# CHAPTER 5

## SETTING THE PROFIBUS ADDRESS

This chapter covers how to configure the Profibus address of an SMD23/4P using Profibus-DP Configuration Tools software available from AMCI. This chapter also explains how to configure the P485USB-3 RS485 adapter for use with Profibus devices.

### Required Equipment

- Windows based computer
- RS485 port, either built into the computer or as an RS232 converter or USB device (The P485USB-3 converter kit from AMCI is an example of a USB device.)
- AMCI Profibus-DP Configuration Tools software
- SMD23/4P unit
- Profibus network cable  
(Can be made from an RS232 cable and an MS-39 connector. Connectors can be ordered from AMCI. Cable and connector are included in the P485USB-3 converter kit.)
- SMD23/4P power cable, using an MS-31 connector that can be ordered from AMCI
- 24 to 48 Vdc power supply

### RS-485 Port Configuration

RS-485 can be used as a 4-wire, full duplex interface or as a 2-wire, half duplex interface. PROFIBUS uses RS-485 in its 2-wire, half duplex configuration. The P485USB-3 converter from AMCI can be configured for either and its factory default settings are for 2-wire, half duplex.

You may have to open the converter to set line termination resistors. The P485USB-3 ships with the line terminators off. AMCI has tested communications over a nine foot RS232 cable at 9600 baud and termination resistors were not required. If your system has a longer cable run between the programming port and the SMD23/4P, termination may be needed.

**NOTE** The termination resistors on the C485USB-3 are 120 ohms between the signals and 750 ohm bias resistors from signal to either VCC or GND. These are RS485 standard values, not PROFIBUS standard values.

Two small, phillips head, screws have to be removed to open the C485USB-3. The screws are behind the gray panel and are the only screws on the device. Once they are removed, the two cover pieces separate and the PC board can be removed.

Figure 5.1 shows the converter PC board and the DIP switches that must also be set before the P485USB-3 can interface with PROFIBUS devices. All three DIP Switches must be in their OFF state, which is down in this picture, which are their factory default settings.

Once the terminating resistors are set, reassemble the P485USB-3 device before proceeding.

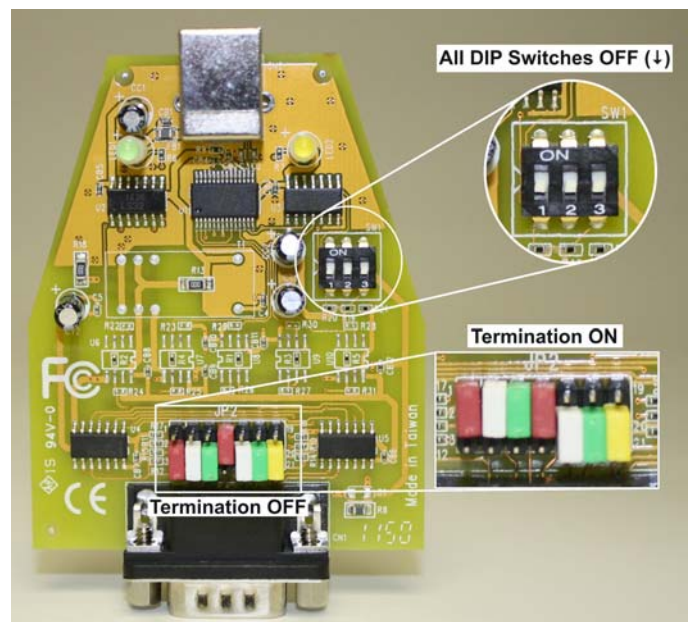


Figure 5.1 C485USB-3 DIP Switch Settings

## Cabling

### Programming Cable

#### ! CAUTION

There is no industry standard pin out for RS-485 connectors. The following section shows how to wire to a P485USB-3 converter as an example. When wiring your bus cable, verify the RS-485 port pin out for your converter. Damage to your SMD23/4P or your RS485 port may occur if the bus cable is not wired correctly.

A sample cable is shown in figure 5.2 below. The AMCI test department uses a standard RS-232 cable with the male connector removed. The remaining female connector and cable is wired to an MS-39 connector. If you use this method to make a programming cable, cut off all unused wires at the cable's outer insulation to isolate them.

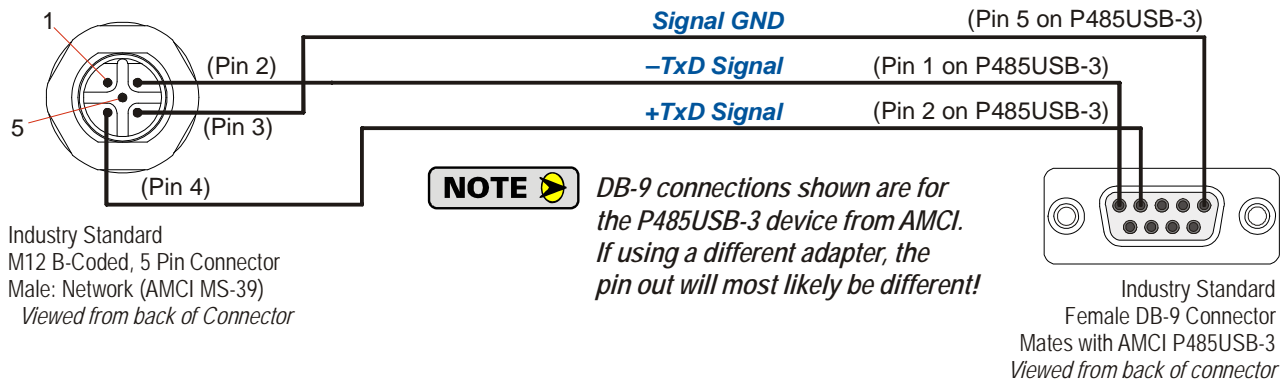


Figure 5.2 Programming Cable

### Power Cable

Power must be brought into the SMD23/4P. The power connector is an industry standard M12, A-coded, 4 pin, female connector. A straight connector is available from AMCI under the part number MS-29. Figure 5.3 shows which pins to connect to in order to power the SMD23/4P.

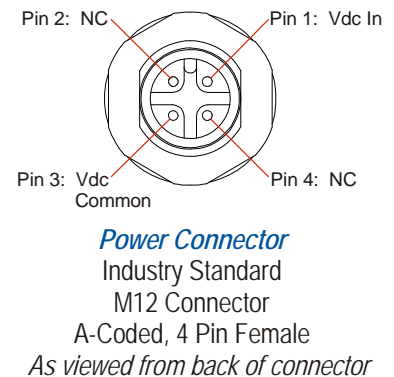


Figure 5.3 Power Connector



### Computer Configuration

#### RS-485 Port

If using a USB serial port device like the P485USB-3, you may have to install device drivers before your computer will recognize the new port. You may need a driver CD or you may need to download the drivers from the manufacturer's website before loading them into Windows. The P485USB-3 uses an FTDI chipset that already has drivers included in Windows XP and above. If they are not pre-loaded on your computer, allow your computer to connect to the Internet when it wants to search for a driver. Your computer will attach to a Microsoft download server and download and install the appropriate drivers.

Once the serial port is attached and the drivers are installed, verify the COM port used by the device. Right click on your "My Computer" icon and select "Manage". The Computer Management program will open.

- Click on "Device Manager"
- Click on the [+] sign next to the "Ports (COM & LPT)" entry in the device tree. Your COM port device will show up in that branch.
- Verify that your device is listed in the range of COM1 to COM4.

If it is not, you will have to change this setting. Start by right clicking on the RS-485 device in the device tree and selecting "Properties" from the drop down menu.

- Select the {Port Settings} tab and then click on the [Advanced] button.
- Click on the drop down list for "COM Port Number:" and select one of the four options. If the COM Port shows "(In Use)" it was probably reserved by a USB device you used before and is actually available.

Figure 5.4 is a screen capture of a P485USB-3 installed as COM 2.

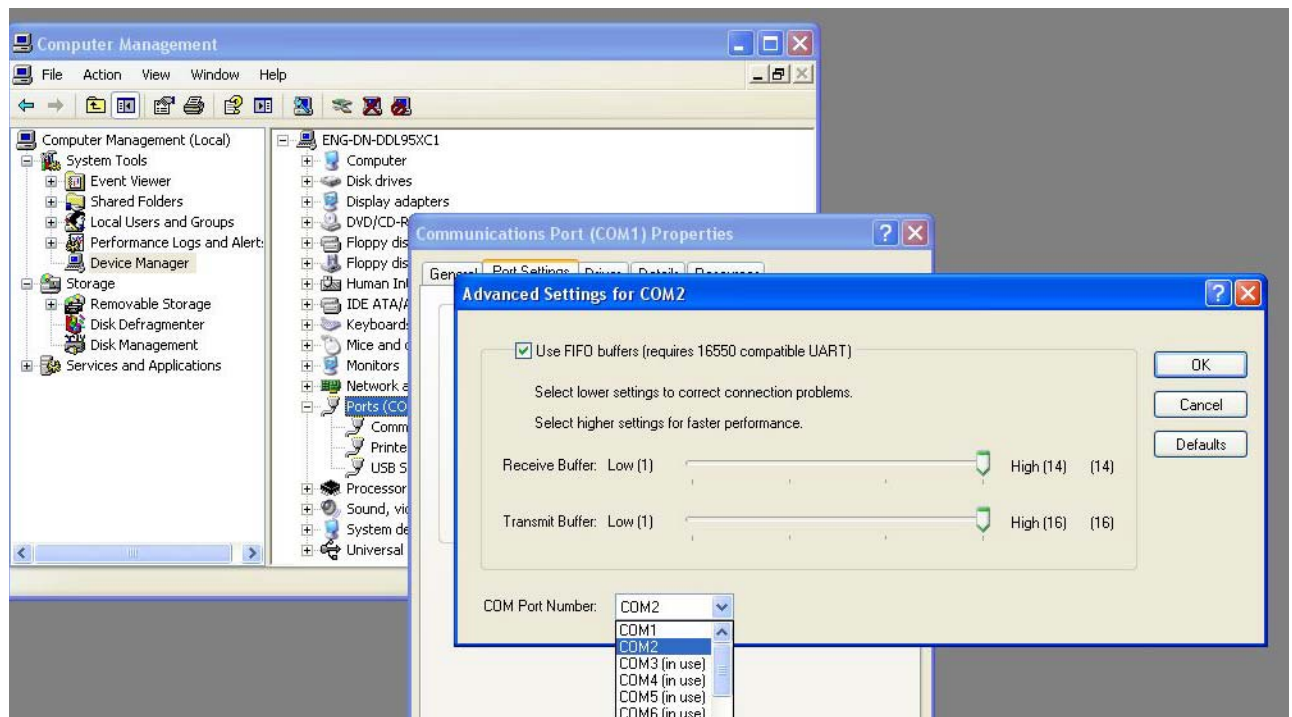


Figure 5.4 Device Manager Settings

- Click on the appropriate [OK] and/or [Apply] buttons to apply the setting and close the windows.
- Close the Computer Management program.

## Computer Configuration (continued)

### AMCI PROFIBUS-DP Configuration Tools software

The PROFIBUS-DP Configuration Tools software is available on our website at: <http://www.amci.com/product-software.asp>. Simply download the software and extract the zip file. Running the resulting exe file will install the program in the correct Program Files folder and create a desktop icon.

- The install program does not make changes to your computer's registry.

### Connecting to the SMD23/4P

- Attach your programming cable to the SMD23/4P and your RS-485 port
- Attach your power cable to the SMD23/4P
- Apply power to the SMD23/4P
- If needed, attach your RS-485 port to your computer. Verify the COM port number for the device

### Using the PROFIBUS-DP Configuration Tools Utility

- 1) Start the program by double clicking on the desktop icon.
- 2) Click on the {Comm. Settings} tab.
- 3) If not using COM1, click on the drop down list for "Port" and select the appropriate port.
- 4) Click on the [Connect Port] button to connect to your RS-485 port.

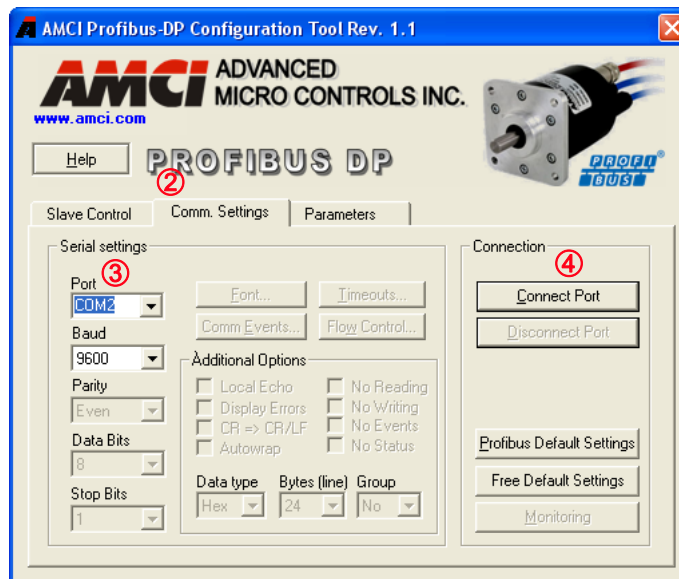


Figure 5.5 Connecting to a COM Port

- At this point, the NS (Network Status) LED on the SMD23/4P should be on and red. The MS (Module Status) LED on the SMD23/4P should be on and green.



## Using the PROFIBUS-DP Configuration Tools Program (continued)

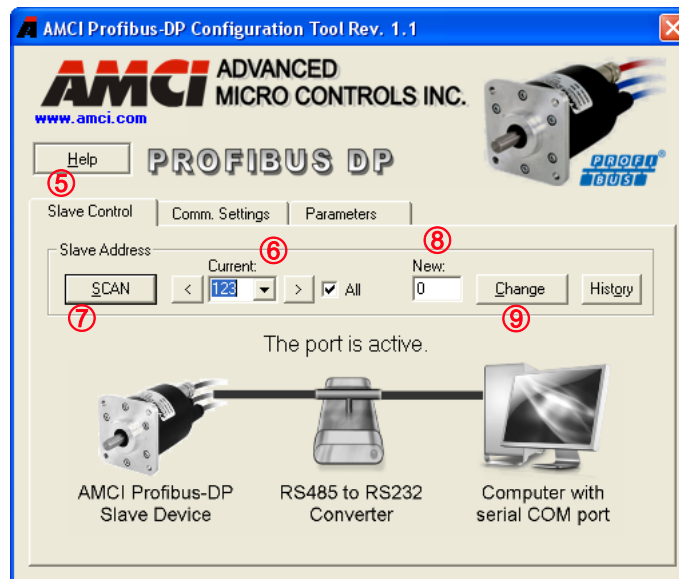


Figure 5.6 DP Configuration Tool - Slave Control Tab

- 5) Click on the {Slave Control} tab.
- 6) If you know the present address of the SMD23/4P, enter it in the “Current:” field. (Factory default address is 97.)
- 7) If you do not know the present address of the SMD23/4P, press the [Scan] button. The program will open a new window and scan every address from 2 to 123. It will then list the address that it found the SMD23/4P at. Double click on the found address to copy it into the “Current:” field.
- 8) Entered the desired SMD23/4P address into the “New:” field.
- 9) Press the [Change] button. The new address will be written to the SMD23/4P and the program will display a message prompt.



Figure 7 Searching for the SMD23/4P



Figure 8 Successful Address Change

- 10) If the change was successful, the NS LED on the SMD23/4P will be blinking red.
- 11) Cycle power to the SMD23/4P and wait for MS LED to be on and green.
- 12) In order to verify the address change, press the [SCAN] button. The program will scan through addresses 2 through 123 and will report the address of the SMD23/4P.

*Notes*

# CHAPTER 6

## Host CONFIGURATION

This chapter outlines the steps commonly needed to get the SMD23/4P communicating with the Profibus master. A Siemens SIMATIC S7-317 controller is used as an example.

### Basic Steps

Configuring a PROFIBUS host requires a few basic steps.

- 1) Download the ZIP archive that contains the GSD files for the SMD23/4P from the [www.amci.com](http://www.amci.com) website.
- 1) Install the GSD file into the configuration software for your host controller.
- 2) Add the SMD23/4P to the PROFIBUS Network.
- 3) Set the I/O word addresses used to communicate with the SMD23/4P.

### Download the GSD file

The GSD files are available on the AMCI website on the <http://www.amci.com/driverfiles.asp> web page. The file is a ZIP archive that has to be extracted to a folder on your computer. Extracting the ZIP file will leave you with four files. Two are GSD files and two are icon files.

### GSD File Installation

- 1) With the SIMATIC manager software started and the correct project open, click on the SIMATIC 300 station icon in the project tree. Double click on the Hardware icon to open the HW Configuration editor.
- 2) Select *Options -> Install GSD File...* from the file menu. The "Install GSD Files" window will open.
- 3) Click on the [Browse] button and navigate to the folder that contains the extracted GSD files you downloaded from the AMCI website. Once at the folder, click on the [OK] button.
- 4) Select the SMD23P.gsd and SMD24P.gsd entries in the file list and click on the [Install] button. In the resulting window, click on the [Yes] button to confirm the installation of the GSD files.
- 5) Click the [Close] button to close the Install GSD File window

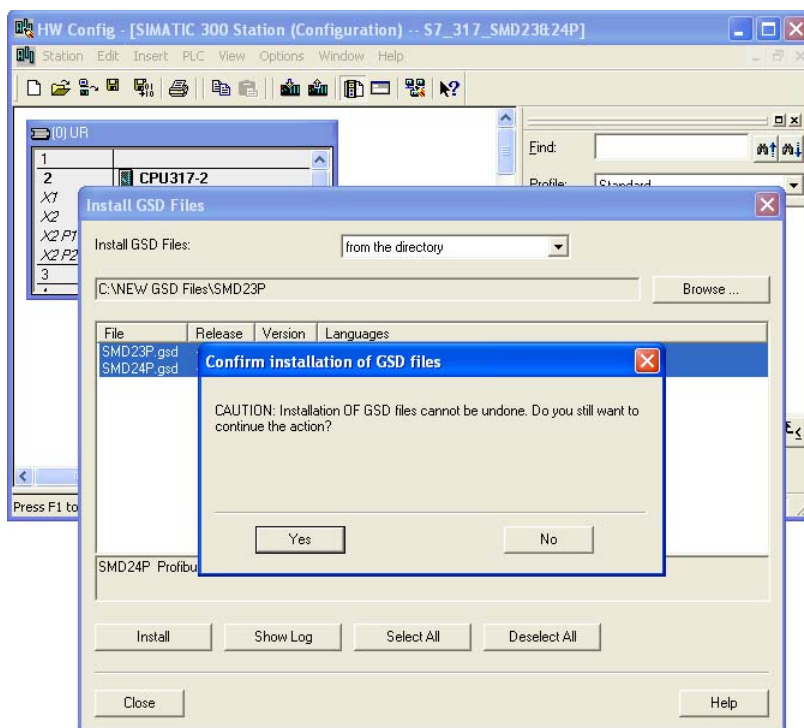


Figure 6.1 GSD File Installation

## Configure the PROFIBUS Network

- 1) If the HW Configuration editor is not open, click on the SIMATIC 300 station icon in the project tree and then double click on the Hardware icon to open the HW Configuration editor.
- 2) If the Hardware Catalog tree is not available on the right side of the HW Configuration Editor window, press [CTRL+K] to open it.
- 3) If not already present, install the processor in the mounting rail by first navigating to the appropriate PLC folder the Hardware Catalog and then dragging the proper icon to slot 2 of the rail. In this example, the folder is: *SIMATIC 300 +> CPU-300 +> CPU 317-2PN/DP +> 6ES7-317-2EJ10-0AB0* and the proper icon is V2.3.
- 4) If the PROFIBUS network is not already configured, you must configure the network parameters, including the PLC's address, Transmission Rate, and Profile. In this example:
  - a) Double click on the MPI/DI device of the processor. This will open the "Properties-MPI/DP" window.
  - b) Set the name of the network.
  - c) Under Interface, click on the Type drop down menu and select "PROFIBUS". This will open the "Properties - PROFIBUS Interface MPI/DP" window.
  - d) Set the PROFIBUS address of the PLC. (Default of 2)
  - e) Click on the [New] button to open the "Properties - New subnet PROFIBUS" window. Use the {General} tab to set the name of the subnet and the {Network Settings} tab to set the bus speed and type. The Type must be set to "DP" in order to communicate with the SMD23/4P.
- 5) Click [OK] to close two open windows in order to get back to the "Properties-MPI/DP" window. Click on the {Operating Mode} tab and confirm that DP Master is selected.
- 6) Close the "Properties-MPI/DP" window by clicking on the [OK] button to get back to HW Configuration Editor screen. You will now see the PROFIBUS network line on the screen.

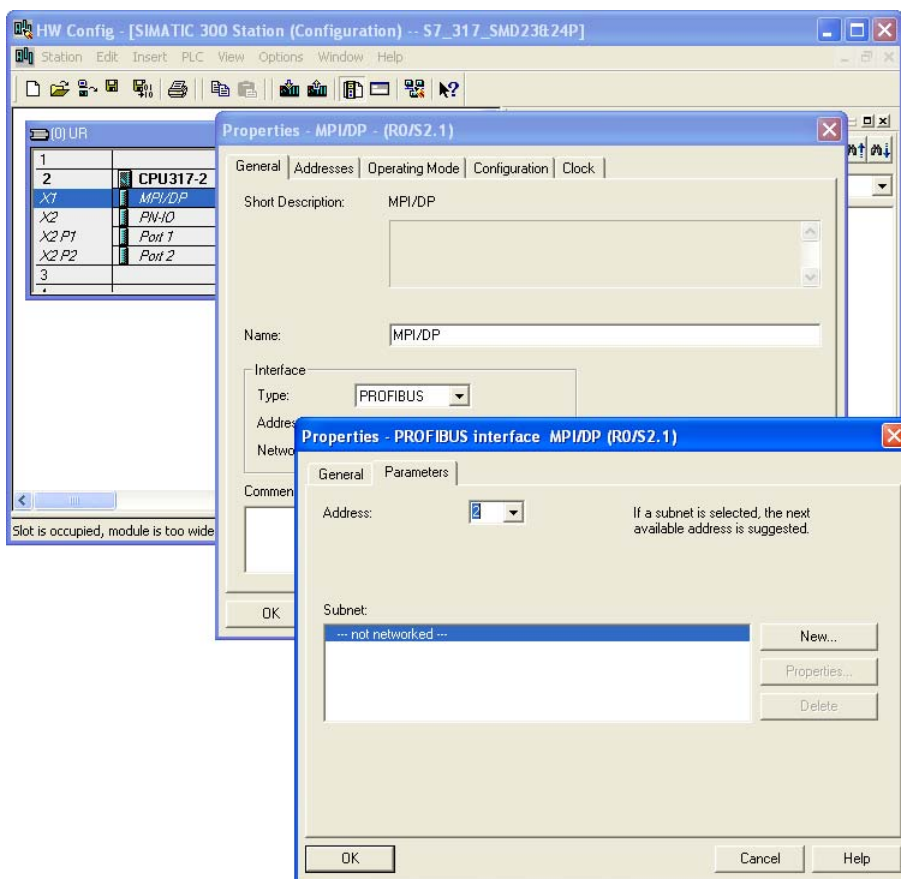


Figure 6.2 PROFIBUS Interface Parameters

## Adding the SMD23/4P to the PROFIBUS Network

- 1) If the Hardware Catalog tree is not available on the right side of the HW Configuration Editor window, press [CTRL+K] to open it.
- 2) Browse to the SMD23P and SMD24P icons by clicking through *PROFIBUS DP* +> *Additional Field Devices* +> *Drives* +> *AMCI\_Products*. Drag and drop the appropriate icon onto the PROFIBUS network line.

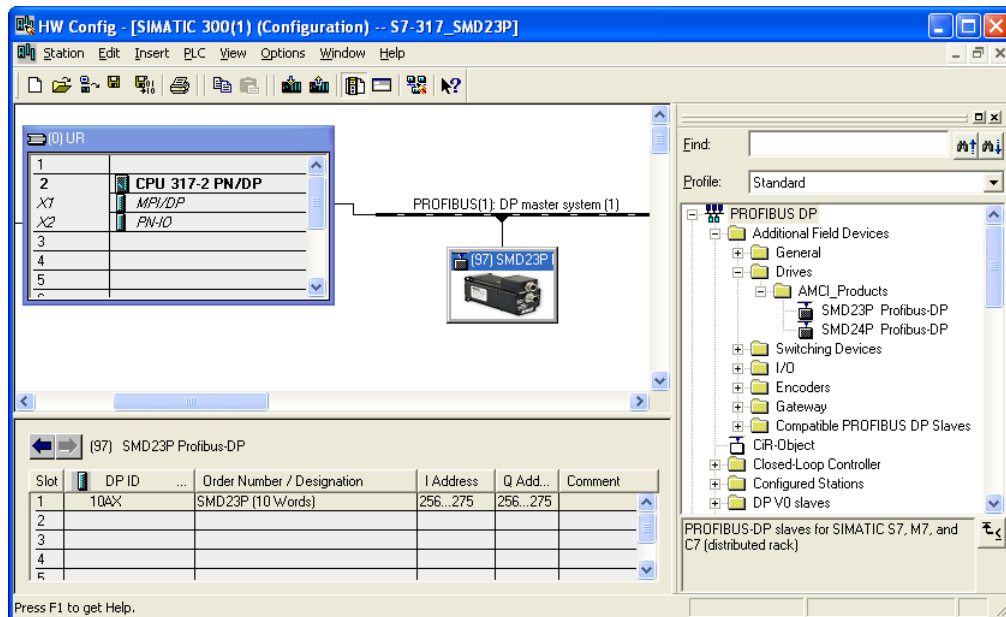


Figure 6.3 SMD23P Added to PROFIBUS Network

- 3) In the “Properties - PROFIBUS interface SMD23/4P Profibus-DP” window that opened when you dropped the icon onto the network line, use the {Parameters} tab to set the network address of the SMD23/4P device. This address must be the same one you set with the AMCI PROFIBUS-DP Configuration Tool software or the factory default of 97. If you have created more than one subnet, select the proper subnet in this tab as well. Use the {General} tab to set the name of the SMD23/4P in your project. Click the [OK] button to close the Properties window.
- 4) With the icon selected, detailed information on the SMD23/4P device is available in the detailed view section of the window, which is below the graphic view of your setup. The default PI (Peripheral Input) and PQ (Peripheral Output) memory locations that were assigned when the SMD23/4P was added are shown in this view. If you want to change these addresses, double click anywhere in the detail view row to open the “Properties - DP slave” window. From here you can set the starting addresses for the SMD23/4P. The ending addresses will automatically update. Click the [OK] button to confirm your selection.

Continue by adding any remaining devices to your PROFIBUS network. Remember to compile and download the changes to your PLC.

***Notes***

## CONFIGURATION MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to configure an SMD23/4P as well as the formats of the Network Input Data that contains the responses from the unit. An SMD23/4P requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data.

### Modes of Operation

An SMD23/4P 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. SMD23/4P units are shipped from the factory without a valid configuration and must be configured on every power up. The ladder logic needed to accomplish this 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 SMD23/4P 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.

#### NOTE

The SMD23/4P will not accept move commands, and the motor will not receive power, until the unit has a valid configuration.

#### Command Mode

This mode gives you the ability to program and execute stepper moves, and reset errors when they occur. An SMD23/4P will always power up in this mode, but a valid configuration must be sent to it before the unit will apply power to the motor or allow you to issue move commands. The command data formats are described in the following chapter.

### Multi-Word Format

Many of the hosts that can be used with an SMD23/4P only support 16 bit integers, which limits the range of values from -32,768 to 32,767 or 0 to 65,535. Many parameters of an SMD23/4P exceed this range. These parameters are transmitted in two separate words. The table below shows how values are split. Note that negative values are written as negative numbers in both words.

Value	First Word	Second Word
12	0	12
12,345	12	345
1,234,567	1,234	567
-7,654,321	-7,654	-321

Table 7.1 Multi-Word Format Examples



### Output Data Format

The correct format for the Network Output Data when the SMD23/4P is in Configuration Mode is shown below.

Data Word	Configuration Data	Range
0	Control Word	See below
1	Configuration Word	See below
2	Starting Speed: Upper Word	Combined value between 1 and 1,999,999 steps/sec.
3	Starting Speed: Lower Word	
4	Motor Steps/Turn	200 to 32,767
5	Reserved	Set to zero
6	Encoder Pulses/Turn	Set to 1,024
7	Idle Current Percentage	0 to 100%
8	Motor Current (X10)	10 to 34, Represents 1.0 to 3.4 Arms
9	Reserved	Set to zero

Table 7.2 Network Output Data Format: Configuration Mode

### Control Word Format

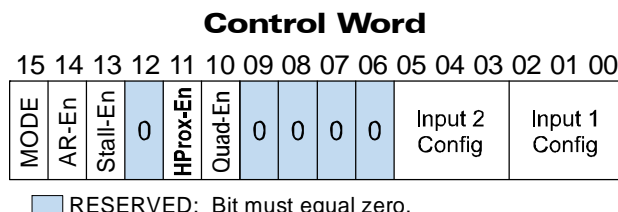


Figure 7.1 Configuration Mode: Control Word Format

- Bit 15: Mode Bit** – “1” for Configuration Mode Programming, “0” for Command Mode Programming. An SMD23/4P powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), unless a valid configuration had been previously written to the flash memory of the unit. The SMD23/4P will not power the motor or accept commands until a valid configuration is written to it or read from its internal flash memory.
- Bit 14: Anti-Resonance Enable Bit** – “0” enables the anti-resonance feature of the SMD23/4P. “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: Stall Detection Enable Bit** – “0” disables motor stall detection. “1” enables motor stall detection. Only valid with the SMD23/4P\_E units that have a built-in encoder. 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: Home Proximity Bit Enable Bit** – “0” when the Home Proximity bit is not used when homing the SMD23/4P. “1” when the Home Proximity bit is used when homing the SMD23/4P. 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: Quadrature Encoder Enable Bit** – “0” when Quadrature Encoder is not used. “1” to enable a Quadrature Encoder. Only valid with the SMD23/4P\_E units that have the built-in encoder.





## Configuration Mode - Output Data Format (continued)

### Control Word Format (continued)

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

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

Bits			Function	Available On
5	4	3		
2	1	0		
0	0	0	General Purpose Input	The input is not used in any of the functions of the SMD23/4P, but it's status is reported in the Network Data. This allows the input to be used as a discrete DC input to the host controller.
0	0	1	CW Limit	Input defines the mechanical end point for CW motion.
0	1	0	CCW Limit	Input defines the mechanical end point for CCW motion.
0	1	1	Start Indexed Move	Starts a move that is held in memory.
0	1	1	Start Indexed Move / Capture Encoder Value	When the encoder is enabled on an SMD23/4P_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 SMD23/4P_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 SMD23/4P_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 7.3 Configuration Mode: Control Word Format



### Configuration Mode - Output Data Format (continued)

#### Configuration Word Format

Configuration Word															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DriveEn	0	0	0	RdConfig	SavConfig	0	0	0	0	0	0	0	0	ActSt-2	ActSt-1

RESERVED: Bit must equal zero.

Figure 7.2 Configuration Mode: Config Word Format

**Bit 15: Driver Enable Bit** – “0” to disable the motor driver circuitry. “1” to enable the motor driver circuitry. When the motor driver is disabled, all voltage is removed from the motor.

**Bits 14 - 12: Reserved** – Must equal zero.

**Bit 11: Read Present Configuration** – If this bit is set when you enter Configuration Mode, the SMD23/4P 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 67.

**Bit 10: Save Configuration** – An SMD23/4P 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 to the flash memory. If the data is not correct, the transition on this bit is ignored. Once the write to flash is completed, the SMD23/4P will write 0xAAAA into the last status word if the write was successful and the MS LED will begin to flash green. If the write is unsuccessful, the SMD23/4P will write 0xEEEE into the last status word and the MS LED will begin to flash red. Once the SMD23/4P issues its response to the Save Configuration command, it stops responding to commands and you must cycle power to the unit. This design decision prevents the SMD23/4P from responding to constant save commands from the host controller. The flash memory has a minimum of 10,000 write cycles.

**Bits 9 - 2: Reserved** – Must equal zero.

**Bit 1: Input 2 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: Input 1 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.



## Configuration Mode - Output Data Format (continued)

### Notes on Other Configuration Words

- Information on the *Multi-Word Format* used when programming the Starting Speed can be found on page 63.
- 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 SMD23/4P is in Configuration Mode is shown below.

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

Table 7.4 Network Input Data Format: Configuration Mode

### Control Word Format (Word 0)

When the Configuration data is valid and accepted, this word mirrors the value of the Control Word written to the SMD23/4P. When the SMD23/4P is not configured, or the data written to it is invalid, then this word has the same format as the Status Bits MSW word when the SMD23/4P is in Command Mode. This format is explained in the *Status Bits MSW 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.



### ***Invalid Configurations***

The following configurations are invalid:

- 1) Setting any of the reserved bits in the configuration words.
- 2) Setting any parameter to a value outside of its valid range. This includes setting the Lower Word of the Starting Speed to a value greater than 999.
- 3) You configure both inputs to have the same function, such as two CW Limit Switches. Note that you can have both inputs configured to be General Purpose Inputs.
- 4) Setting the *Stall Detection Enable Bit* without configuring the SMD23/4P\_E to use the encoder.
- 5) Setting the Input Configuration bits for either input to “111”. See table 7.3 on page 65 for more information.

## CHAPTER 8

# COMMAND MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to command the SMD23/4P as well as the formats of the Network Input Data that contains the responses from the unit. An SMD23/4P requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data.

### Multi-Word Format

Many of the hosts that can be used with the SMD23/4P only support 16 bit integers, which limits the range of values from -32,768 to 32,767 or 0 to 65,535. Many parameters of the SMD23/4P exceed this range. These parameters are transmitted in two separate words. The table below shows how values are split. Note that negative values are written as negative numbers in both words.

Value	First Word	Second Word
12	0	12
1,234,567	1,234	567
-7,654,321	-7,654	-321

Figure 8.1 Multi-Word Format Examples

### 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 Control Word (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 SMD23/4P.

Data Word	Function
0	Command Bits MSW
1	Command Bits LSW
2	Command Parameters  Word meaning depends on the command set to the SMD23/4P
3	
4	
5	
6	
7	
8	
9	

Table 8.1 Command Mode Data Format

**Command Bits MSW**

Command Bits MSW															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MODE	PrstEnc	RunAMv	PrgAMS	PrgAMv	ClrErr	PrstPos	-JogMv	+JogMv	-Home	+Home	I-Stop	ResMv	HoldMv	RelMv	AbsMv

Figure 8.2 Command Bits MSW Format

- Bit 15: Mode Bit** – “1” for Configuration Mode Programming “0” for Command Mode Programming. The SMD23/4P powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), if a valid configuration was not previously stored in flash memory. This is the factory default behavior. An SMD23/4P 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** – Valid only with an SMD23/4P\_E, when set to “1” the SMD23/4P\_E will pre-set the Encoder Position to the value stored in Output Words 2 and 3.
- Bit 13: Run Assembled Move** – When set to “1” an SMD23/4P will run the Assembled Move already stored in memory.
- **Assembled Move Type – Command Bits LSW, Bit 9:** This bit determines the type of move that is run. When this bit equals “0”, a Blend Move is run. When this bit equals “1”, a Dwell Move is run. When starting a Dwell Move, the Dwell Time is programmed in word 9 of the Command Data. The value is programmed in milliseconds and can range from 0 to 65,536.
  - **Blend Move Direction – Command Bits LSW, Bit 4:** This bit is used to determine the direction that the Blend Move will be run in. When this bit equals “0”, the Blend Move runs in the clockwise direction. When this bit equals “1”, the Blend Move is run in the counter-clockwise direction.
- Bits 11 & 12: Program Move & Program Move Segment 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 36.
- Bit 10: Reset Errors** – When set to “1” the SMD23/4P will clear all existing errors and attempt to use the present data to run a new command. This command also resets the *Move Complete* bit.
- Bit 9: Preset Motor Position Bit** – When set to “1” the SMD23/4P 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 Move (CCW)** – When set to “1” the SMD23/4P will run a Jog Move in the counter-clockwise direction. A full explanation of a [±Jog Move](#) can be found starting on page 29.
- **Run Registration Move – Command Bits LSW, 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 Move (CW)** – When set to “1” the SMD23/4P will run a Jog Move in the clockwise direction. A full explanation of a [±Jog Move](#) can be found starting on page 29.
- **Run Registration Move – Command Bits LSW, 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.

**Command Bits MSW (continued)**

- Bit 6: Find Home CCW** – When set to “1” the SMD23/4P 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 SMD23/4P](#) chapter starting on page 41.
- Bit 5: Find Home CW** – When set to “1” the SMD23/4P 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 SMD23/4P](#) chapter starting on page 41.
- Bit 4: Immediate Stop Bit** – When set to “1” the SMD23/4P will stop all motion without deceleration. The Motor Position value will become invalid if this bit is set during a move. Setting this bit when a move is not in progress will not cause the Motor Position to become invalid.
- Bit 3: Resume Move 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 in the [Controlling Moves In Progress](#) section of this manual starting on page 38. Note that a move in its hold state does not need to be resumed. The move is automatically cancelled if another move is started in its place.
- Bit 2: Hold Move 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 in the [Controlling Moves In Progress](#) section of this manual starting on page 38.
- 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 27.
- 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 28.

**Command Bits LSW**

Command Bits LSW															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DriveEn	0	0	0	BkHPX	0	AMType	IndxMv	RunReg	0	SavAMv	AMvDir	HyCtrl	EncMv	0	0

Figure 8.3 Command Bits LSW Format

- Bit 15: Driver Enable Bit** – “0” to disable the motor current, “1” to enable motor current. A valid configuration must be written to the SMD23/4P before the driver can be enabled.
- Bits 14 – 12: Reserved** – Must equal “0”.
- Bit 11: Backplane Home Proximity Bit** – When the SMD23/4P 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 41.

**Command Bits LSW (continued)**

- Bit 10: Reserved** – Must equal “0”.
- Bit 9: Assembled Move Type** – When this bit equals “0”, a Blend Move is started when the Run Assembled Move bit, (Command Word 1, Bit 13) makes a 0 → 1 transition. When this bit equals “1”, a Dwell Move is started on the transition. The direction of a Blend Move is controlled by the Blend Move Direction bit, (Command Bits LSW, Bit 4). In a Dwell Move, the Dwell Time between segments is programmed in Word 9 of the command data.
- Bit 8: Run As An Indexed Move** – If this bit is set when a move command is issued, the SMD23/4P 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 command data, including this bit, must remain in the Network Output Data registers while performing an Indexed Move.
- Bit 7: Run 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 Assembled Move** – Set this bit to save the programmed Assembled Move to flash memory. This bit must be set when the Move Program Mode makes a 1 → 0 transition at the end of the programming cycle to write the data to flash memory. The SMD23/4P responds by flashing the MS LED when the write 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 SMD23/4P 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: Blend Move Direction** – When you command a Blend Move to run, this bit determines the direction of rotation. Set to “0” for a clockwise Blend Move, ‘1’ for a counter-clockwise Blend Move.
- Bit 3: Reserved** – Must equal “0”.
- Bit 2: Encoder Move** – Valid with an SMD23/4P\_E only, if this bit to “1” when you command an Absolute or Relative move, the move distance will be based on the encoder position, not on the motor position. Set to “0” for a standard Relative or Absolute move based on motor position. Please note that an encoder move does not stop at the programmed encoder position. It starts to decelerate at this position. A full description of *Encoder Moves* starts on page 32.
- Bit 1: Reserved** – Must equal “0”.
- Bit 0: Reserved** – Must equal “0”.



## Command Blocks

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

### Absolute Move

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0001
1	<i>Command Bits LSW</i>		See pg. 71
2	Abs. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	Abs. Target Position: Lower Word		
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.2 Absolute Move Command Block

### Relative Move

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0002
1	<i>Command Bits LSW</i>		See pg. 71
2	Rel. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	Rel. Target Position: Lower Word		
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.3 Relative Move Command Block

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

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0004
1	<i>Command Bits LSW</i>		See pg. 71
2	Unused		See Note Below
3	Unused		See Note Below
4	Unused		See Note Below
5	Unused		See Note Below
6	Unused		See Note Below
7	Unused		See Note Below
8	Unused		See Note Below
9	Unused		See Note Below

Table 8.4 Hold Move Command Block

Unused words are ignored by an SMD23/4P and can be any value, including parameter values in the previous command.

**Resume Move**

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0008
1	<i>Command Bits LSW</i>		See pg. 71
2	Unused		See Note Below
3	Unused		See Note Below
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.5 Resume Move Command Block

Unused words are ignored by the SMD23/4P 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

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0010
1	<i>Command Bits LSW</i>		See pg. 71
2	Unused		See Note Below
3	Unused		See Note Below
4	Unused		See Note Below
5	Unused		See Note Below
6	Unused		See Note Below
7	Unused		See Note Below
8	Unused		See Note Below
9	Unused		See Note Below

Table 8.6 Immediate Stop Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

### +Find Home (CW)

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0020
1	<i>Command Bits LSW</i>		See pg. 71
2	Unused		See Note Below
3	Unused		See Note Below
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.7 +Find Home (CW) Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

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

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0040
1	<i>Command Bits LSW</i>		See pg. 71
2	Unused		See Note Below
3	Unused		See Note Below
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.8 –Find Home (CCW) Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

**+Jog Move (CW)**

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0080
1	<i>Command Bits LSW</i>		See pg. 71 Bits 7 & 6 must equal “00”
2	Unused		See Note Below
3	Unused		See Note Below
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.9 +Jog Move (CW) Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

## Command Blocks (continued)

### +Registration Move (CW)

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0080
1	<i>Command Bits LSW</i>		See pg. 71 Bits 7 & 6 must equal "10"
2	Stopping Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
3	Stopping Distance: Lower Word		
4	Programmed Speed: Upper Word	Steps per Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
9	Min. Reg. Move Distance: Lower Word		

Table 8.10 +Registration Move (CW) Command Block

### –Jog Move (CCW)

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0100
1	<i>Command Bits LSW</i>		See pg. 71 Bits 7 & 6 must equal "00"
2	Unused		See Note Below
3	Unused		See Note Below
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.11 –Jog Move (CCW) Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

**Command Blocks (continued)****–Registration Move (CCW)**

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0100
1	<i>Command Bits LSW</i>		See pg. 71 Bits 7 & 6 must equal “10”
2	Stopping Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
3	Stopping Distance: Lower Word		
4	Programmed Speed: Upper Word	Steps per Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
9	Min. Reg. Move Distance: Lower Word		

Table 8.12 –Registration Move (CCW) Command Block

**Preset Position**

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0200
1	<i>Command Bits LSW</i>		See pg. 71
2	Position Preset Value: Upper Word	Steps	Combined value between –8,388,607 and +8,388,607
3	Position Preset Value: Lower Word		
4	Unused		See Note Below
5	Unused		See Note Below
6	Unused		See Note Below
7	Unused		See Note Below
8	Unused		See Note Below
9	Unused		See Note Below

Table 8.13 Preset Position Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

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

## Command Blocks (continued)

### Reset Errors

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0400
1	<i>Command Bits LSW</i>		See pg. 71
2	Unused		See Note Below
3	Unused		See Note Below
4	Unused		See Note Below
5	Unused		See Note Below
6	Unused		See Note Below
7	Unused		See Note Below
8	Unused		See Note Below
9	Unused		See Note Below

Table 8.14 Reset Errors Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

- Resetting errors will also reset the *Move Complete* status bit.

### Run Assembled Move

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#2000
1	<i>Command Bits LSW</i>		See pg. 71 Blend Move: Bit 9 = "0" Dwell Move: Bit 9 = "1" Blend Move direction set by Bit 4.
2	Unused		See Note Below
3	Unused		See Note Below
4	Unused		See Note Below
5	Unused		See Note Below
6	Unused		See Note Below
7	Unused		See Note Below
8	Unused		See Note Below
9	Unused with Blend Move Dwell Time with Dwell Move	milliseconds	0 to 65,535

Table 8.15 Run Assembled Move Command Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values in the previous command.

**Command Blocks (continued)****Preset Encoder Position**

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#4000
1	<i>Command Bits LSW</i>		See pg. 71
2	Encoder Preset Value: Upper Word	Steps	Combined value between –8,388,607 and +8,388,607
3	Encoder Preset Value: Lower Word		
4	Unused		See Note Below
5	Unused		See Note Below
6	Unused		See Note Below
7	Unused		See Note Below
8	Unused		See Note Below
9	Unused		See Note Below

Table 8.16 Preset Encoder Position Command Block

Unused words are ignored by the SMD23/4P 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

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#0800
1	<i>Command Bits LSW</i>		See pg. 71
2-9	Unused		See Note Below

Table 8.17 Assembled Move First Programming Block

Unused words are ignored by the SMD23/4P and can be any value, including parameter values from the previous command.

Once the first block is transmitted, the SMD23/4P responds by setting bits 8 and 9 in the Status Bits MSW word. (See *Status Bits MSW Format* starting on page 82.) Once these are set, you can then start transmitting Segment Blocks.

### Segment Block

Data Word	Function	Units	Range
0	<i>Command Bits MSW</i>		16#1800
1	<i>Command Bits LSW</i>		See pg. 71
2	Rel. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
3	Rel. Target Position: Lower Word		
4	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
5	Programmed Speed: Lower Word		
6	Acceleration	Steps/ms/sec	1 to 5000
7	Deceleration	Steps/ms/sec	1 to 5000
8	<i>Reserved</i>		Must equal zero for compatibility with future releases.
9	Acceleration Jerk		0 to 5000

Table 8.18 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 SMD23/4P 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 SMD23/4P will respond to this by setting bit 9 the Status Bits MSW word and the next Segment Block can be written to the SMD23/4P. 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 SMD23/4P is in Command Mode is shown below.

Data Word	Command Mode Input Data
0	Status Bits MSW
1	Status Bits LSW
2	Motor Position: Upper Word
3	Motor Position: Lower Word
4	Encoder Position: Upper Word
5	Encoder Position: Lower Word
6	Captured Encoder Position: Upper Word
7	Captured Encoder Position: Lower Word
8	Value of Motor Current (X10)
9	Last programmed value of the Acceleration Jerk Parameter

Table 8.19 Network Input Data Format: Command Mode

### Status Bits MSW Format

Status Bits MSW															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MODE	CrtOK	ConErr	CmdErr	InErr	PosInvl	Xmt_MS	PgmMov	MvCmp	Decel	Accel	Homed	Stopped	Hold	MvCCW	MvCW

Figure 8.4 Command Mode: Status Bits MSW Format

**Bit 15: Mode Bit** – “1” for Configuration Mode Programming, “0” for Command Mode Programming. An SMD23/4P powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), unless a valid configuration had been previously written to the flash memory of the unit. The SMD23/4P will not power the motor or accept commands until a valid configuration is written to it or read from its internal flash memory.

**Bit 14: Controller OK Bit** – “1” when the SMD23/4P is operating without a fault, “0” when an internal fault condition exists.

**Bit 13: Configuration Error Bit** – “1” on power up before a valid configuration has been written to the SMD23/4P or after any invalid configuration has been written to the unit. “0” when the SMD23/4P has a valid configuration written to it or if valid configuration data is read from flash memory on power up.

**Bit 12: Command Error Bit** – “1” when an invalid command has been written to the SMD23/4P. This bit can only be reset by the Reset Errors bit, Command Bits MSW, Bit 10.

**Bit 11: Input Error Bit** – “1” when:

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

This bit is reset by a [Reset Errors](#) command. The format of the command is given on page 79.

***Input Data Format (continued)*****Status Bits MSW Format (continued)**

**Bit 10: Position Invalid Bit** – “1” when:

- A configuration is written to the SMD23/4P
- The motor position has not been preset
- The machine has not been homed
- The Network Connection has been lost and re-established.
- An Immediate or Emergency Stop has occurred
- An End Limit Switch has been reached
- A motor stall has been detected.

Absolute moves cannot be performed while the position is invalid.

**Bit 9: Transmit Move Segment Bit** – The SMD23/4P 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 36.

**Bit 8: Program Move Mode Bit** – The SMD23/4P 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 36.

**Bit 7: Move Complete Bit** – 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 SMD23/4P, when the position is preset, or a Reset Errors command is issued to the unit. This bit is also set along with the Command Error bit (Bit 12 of this word), when any Jog Move or Registration Move parameters are outside of their valid ranges. This bit is not set on a command error for any other type of command. Finally, this bit is not set at the end of a homing operation.

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

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

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

**Bit 3: Axis Stopped Bit** – 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: Hold State Bit** – Set to “1” when a move command has been successfully brought into a Hold State. Hold States are explained in the Controlling Moves In Progress section starting on page 22.

**Bit 1: CCW Move Bit** – Set to “1” when the motor is rotating in a counter-clockwise direction.

**Bit 0: CW Move Bit** – Set to “1” when the motor is rotating in a clockwise direction.

**Input Data Format (continued)****Status Bits LSW Format**

Status Bits LSW															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DrvEn	Stall	0	0	HeartBt	LmtCon	InvPar	0	DrvFlt	NetLoss	0	TempWarn	0	0	IN2St	IN1St

Figure 8.5 Command Mode: Status Bits LSW Format

- Bit 15: Driver Enabled Bit** – Present state of global Driver Enable Bit. When set to “0”, current is not available to the motor under any condition. When set to “1”, the motor driver section of the SMD23/4P 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: Motor Stall Detected Bit** – Set to “1” when a motor stall has been detected.
- Bit 13: Reserved Bit** – Will always equal zero.
- Bit 12: Reserved Bit** – Will always equal zero.
- Bit 11: Heart Beat Bit** – This bit will change state approximately every 500 milliseconds. Monitor this bit to verify that the unit and network connection are operating correctly. Note that this bit is only available when in Command Mode.
- Bit 10: Limit Condition Active Bit** – This bit is set if an End Limit Switch is reached during a move. This bit will be reset when the Limit Switch changes from its active to inactive state, or when a Reset Errors Command is issued.
- Bit 9: Invalid Parameter Change Bit** – Set during a Jog Move if parameters are changed to invalid values. The only way to clear this bit once set is to write correct parameter values to the SMD23/4P. Parameters that can be changed during a Jog Move are Programmed Speed, Acceleration, and Deceleration.
- Bit 8: Reserved Bit** – Will always equal zero.
- Bit 7: Driver Fault Bit** – If the driver section of the SMD23/4P is enabled, this bit will be a “1” during an Overtemperature Fault. This bit can only be cleared by cycling power to the SMD23/4P.
- Bit 6: Network Lost Error Bit** – If the physical network connection is lost at any time, this bit will be set when the connection is re-established. The Input Error bit will also be set. Note that this bit is not set if the communication loss is not due to a physical error.
- Bit 5: Reserved Bit** – Will always equal zero.
- Bit 4: Temperature Warning Bit** – 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 installing a fan to force additional air-flow around the unit.
- Bit 3: Reserved Bit** – Will always equal zero.
- Bit 2: Reserved Bit** – Will always equal zero.
- 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 Format](#) section starting on page 66.
- 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 Format](#) section starting on page 66.

***Input Data Format (continued)*****Motor Position Data**

These two inputs words report the motor position based on the number of steps output by the SMD23/4P. The range for the position value is -32,768,000 to 32,767,999. The data is transmitted in the *Multi-Word Format* described on page 69. Note that both words in the multi-word format will be negative if the position value is negative.

**Encoder Position Data**

These two inputs words report the encoder position based on the quadrature pulses from the internal encoder of an SMD23/4P\_E. The range for the encoder position value is -32,768,000 to 32,767,999. The data is transmitted in the *Multi-Word Format* described on page 69. Note that both words in the multi-word format will be negative if the encoder position value is negative.

**Captured Encoder Position Data**

These two inputs words report the position of the internal encoder of an SMD23/4P\_E when an input transitions for its inactive to its active state. The input must be configured as a Capture Encoder Value input or as a Stop Jog and Capture Encoder Value input. The range for the captured encoder position value is -32,768,000 to 32,767,999. The data is transmitted in the *Multi-Word Format* described on page 69. Note that both words in the multi-word format will be negative if the captured encoder position value is negative.

***Notes on Clearing a Driver Fault***

A Driver Fault occurs when there is an over temperature condition. When a Driver Fault occurs, the SMD23/4E sets bit 7 of the Status Bits LSW word in the Network Input Data. The only way to clear this fault is to lower the temperature of the motor and cycle power to the SMD23/4P.



***ADVANCED MICRO CONTROLS INC.***

20 GEAR DRIVE, TERRYVILLE, CT 06786 T: (860) 585-1254 F: (860) 584-1973

[www.amci.com](http://www.amci.com)

***LEADERS IN ADVANCED CONTROL PRODUCTS***