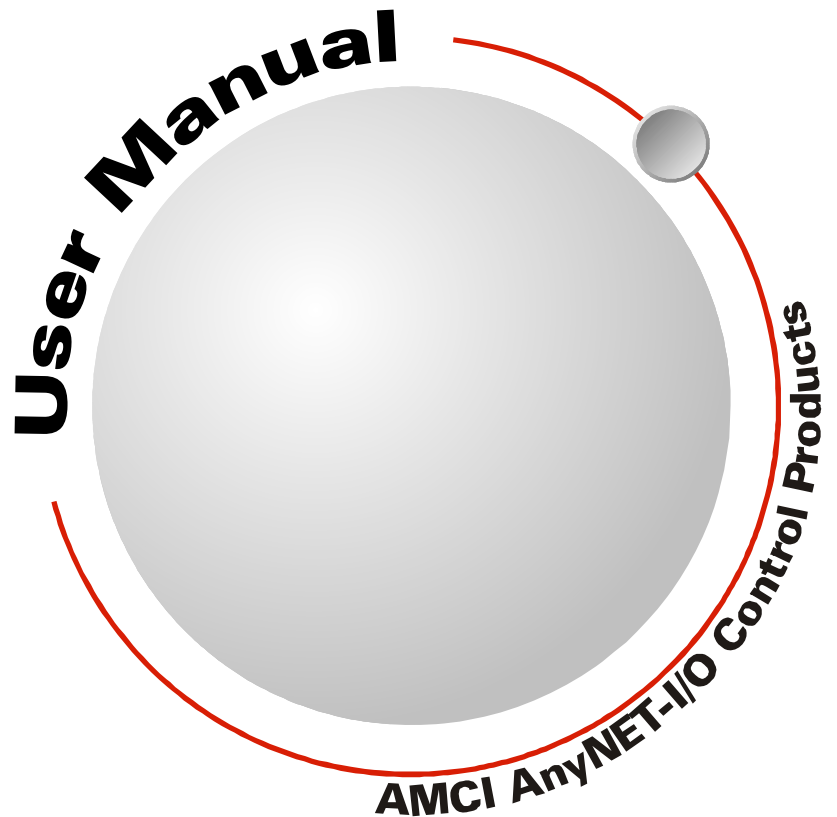


# **ANG1 & ANG1M AnyNET-I/O Stepper Motor Indexer/Driver**



# 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 2012 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 2012 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
Trademark Notices .....	7
Revision Record .....	7
Revision History .....	7
Navigating this Manual .....	7
Manual Conventions .....	8
Where To Go From Here .....	8

## **Chapter 1: Introduction to the ANG1(M)**

AnyNET-I/O .....	9
The ANG1 and ANG1M .....	9
General Functionality .....	9
Encoder Functionality .....	10
Indexer Functionality .....	11
Driver Functionality .....	12
I/O Connector .....	12
Power Supply Sizing .....	12
Available Discrete Inputs .....	13
Home Input .....	13
CW Limit Switch or CCW Limit Switch .....	13
Start Indexer Move Input .....	13
Emergency Stop Input .....	13
Stop Jog or Registration Move Input .....	13
Capture Encoder Position Input .....	13
General Purpose Input .....	13
Encoder Feedback Inputs .....	14
Available Discrete Output .....	14
Modbus Network Connector .....	15
Mating Connector .....	15
Network Status LED .....	15
Motor Connector .....	15

## **Chapter 1: Introduction to the ANG1(M) (continued)**

Front Panel .....	16
AnyNet-I/O Address Settings .....	16
Modbus Network Address Settings .....	16
Status LED .....	17
Micro USB Connector .....	17
Specifications .....	18

## **Chapter 2: Move Profiles**

Units of Measure .....	19
Definition of Count Direction .....	19
Definition of Home Position .....	19
Definition of Starting Speed .....	19
Definition of Target Position .....	20
Definition of Acceleration Types .....	20
Linear Acceleration .....	20
Triangular S-Curve Acceleration .....	21
Trapezoidal S-Curve Acceleration .....	21
A Simple Move .....	22
Controlled and Immediate Stops .....	23
Host Control .....	23
Hardware Control .....	23
Basic Move Types .....	23
Relative Move .....	23
Controlled Stop Conditions .....	24
Immediate Stop Conditions .....	24
Absolute Move .....	24
Controlled Stop Conditions .....	25
Immediate Stop Conditions .....	25
±Jog Move .....	25
Controlled Stop Conditions .....	26
Immediate Stop Conditions .....	26
±Registration Move .....	27
Controlled Stop Conditions .....	28
Immediate Stop Conditions .....	28
Encoder Moves .....	28
Controlled Stop Conditions .....	29
Immediate Stop Conditions .....	29

**Chapter 2: Move Profiles  
(continued)**

Assembled Moves ..... 29  
 Blend Moves ..... 29  
     Controlled Stop Conditions ..... 30  
     Immediate Stop Conditions ..... 30  
 Dwell Move ..... 31  
     Controlled Stop Conditions ..... 31  
     Immediate Stop Conditions ..... 31  
 Assembled Move Programming ..... 32  
     Control Bits – Output Data ..... 32  
     Control Bits – Input Data ..... 32  
     Programming Routine ..... 32  
     Saving an Assembled  
         Move in Flash ..... 32  
 Indexed Moves ..... 33  
 Controlling Moves In Progress ..... 34  
     Find Home Moves ..... 34  
     Jog Moves ..... 34  
     Registration Moves ..... 34  
     Absolute, Relative, and  
         Encoder Moves ..... 34  
     Assembled Moves ..... 34  
 Encoder Follower ..... 35  
     Motor Steps/Turn ..... 35  
     ELGearing Multiplier  
         and Divisor ..... 35  
     How It Works ..... 35  
     Controlled Stop Conditions ..... 35  
     Immediate Stop Conditions ..... 35  
     Advanced Ratio Control ..... 36  
 Stall Detection ..... 36  
 Hybrid Control ..... 36  
 Idle Current Reduction ..... 37  
 Current Loop Gain ..... 37

**Chapter 3: Homing the ANG1(M)**

Definition of Home Position ..... 39  
 Position Preset ..... 39  
 ±Find Home Commands ..... 39  
 Homing Inputs ..... 39  
     Physical Inputs ..... 39  
     Backplane Inputs ..... 39  
 Homing Configurations ..... 40  
 Homing Profiles ..... 40  
     Home Input Only Profile ..... 40  
     Profile with Proximity Input ..... 41  
     Profile with Overtravel Limit ..... 42

**Chapter 4: Installing the ANG1(M)**

Location ..... 43  
 Safe Handling Guidelines ..... 43  
     Prevent Electrostatic Damage ..... 43  
     Prevent Debris From  
         Entering the Module ..... 43  
     Remove Power Before Servicing  
         in a Hazardous Environment ..... 43  
 Mounting ..... 44  
     Minimum Spacing ..... 44  
     DIN Rail Installation ..... 44  
     Installing IC-5 Connectors  
         (as needed) ..... 45  
     Mounting the ANG1(M) Module .. 45  
 Addressing ..... 45  
     AnyNet-I/O Stack Addressing ..... 45  
     Modbus Network Addressing ..... 46  
 Modbus Network Connector ..... 46  
 Modbus Network Wiring ..... 47  
     4-Wire Bus ..... 47  
 Network Status LED ..... 48  
 I/O Connector Pin Out ..... 48  
 Power Wiring ..... 48  
 Input Wiring ..... 49  
 Output Wiring ..... 50  
 Encoder Wiring ..... 50  
     Differential Wiring ..... 50  
     Single Ended Wiring ..... 51  
 Installing the Stepper Motor ..... 52  
     Outline Drawings ..... 52  
     Mounting the Motor ..... 52  
     Connecting the Load ..... 52  
     Extending the Motor Cable ..... 52  
     Installing the Motor Cable ..... 52  
 Connecting the Motor ..... 53  
     Motor Connector ..... 53  
     Motor Wiring ..... 53

# TABLE OF CONTENTS

---

## **Chapter 5: Network Configuration**

USB Connection .....	55
Installing the AMCI NET Software .....	55
USB Driver Installation .....	55
Using the AMCI NET Software .....	56
Connect to your ANG1E .....	56
The Meaning of the Product Name .....	57
Setting Modbus Serial Line Parameters .....	57
Configuring an ANG1(M) module (optional) .....	58

## **Chapter 6: Modbus Interface**

AnyNET-I/O Stack Memory Layout .....	59
AnyNET-I/O Stack Addressing .....	59
Supported Modbus Functions .....	60
Supported Modbus Exceptions .....	60

## **Chapter 7: Configuration Mode Data Format**

Modes of Operation .....	61
Configuration Mode .....	61
Command Mode .....	61
Multi-Word Format .....	61
Output Data Format .....	62
Control Word Format .....	62
Configuration Word Format .....	64
Notes on Other Configuration Words .....	65
Input Data Format .....	65
Control Word Format (Word 0) ....	65
Invalid Configurations .....	66

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

Multi-Word Format .....	67
Command Bits Must Transition .....	67
Output Data Format .....	67
Command Bits MSW .....	68
Command Bits LSW .....	69
Command Blocks .....	71
Absolute Move .....	71
Relative Move .....	71
Hold Move .....	72
Resume Move .....	72
Immediate Stop .....	73
+Find Home (CW) .....	73
-Find Home (CCW) .....	74
+Jog Move (CW) .....	74
+Registration Move (CW) .....	75
-Jog Move (CCW) .....	75
-Registration Move (CCW) .....	76
Encoder Follower Move .....	76
Preset Position .....	77
Reset Errors .....	77
Run Assembled Move .....	78
Preset Encoder Position .....	78
Programming Blocks .....	79
First Block .....	79
Segment Block .....	79
Input Data Format .....	80
Status Word 1 Format .....	80
Status Word 2 Format .....	82
Motor Position Data .....	83
Encoder Position Data .....	83
Captured Encoder Position Data ...	83
Notes on Clearing a Driver Fault .....	84
Reset Driver Fault .....	84

## **Chapter A: Choosing Your Motor**

Sizing Your Motor .....	85
Determining Your Motor Current Setting	86
A Note on Microstepping .....	86

*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 ANG1 and ANG1M AnyNET-I/O Stepper Motor Indexer / Driver. It is written for the engineer responsible for incorporating these modules into a design, as well as the engineer or technician responsible for its actual installation.

## **Applicable Units**

This manual applies to both the ANG1M and the ANG1 modules. The ANG1M has an integral RS485 port while the ANG1 does not. This port allows the ANG1M to connect itself, and up to five other modules, to a Modbus Serial Line network.

**NOTE**  Most of this manual applies to both the ANG1M and the ANG1 modules. When a part of this manual only applies to one of the units, the name of the module is used. When the section applies equally to each unit, the text "ANG1(M)" is used instead of listing both part numbers.

## **Trademark Notices**

The AMCI logo and "AnyNET-I/O" are trademarks of Advanced Micro Controls Inc.

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

## **Revision Record**

This manual, 940-0A020, is the first revision of this manual. It was first released May 4, 2012.

### **Revision History**

940-0A020 Initial Release.

## **Navigating this Manual**

This manual is designed to be used in both printed and on-line formats. Its on-line form is a PDF document, which requires Adobe Acrobat Reader version 6.0+ 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 6.0 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.

## 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	<b><i>INTRODUCTION TO THE ANGI(M)</i></b>	Anyone new to the ANGI(M). This chapter gives a basic overview of the features available on the unit, typical applications, and electrical specifications.
2	<b><i>MOVE PROFILES</i></b>	Anyone that needs detailed information on how the module can be used to control motion in your application.
3	<b><i>HOMING THE ANGI(M)</i></b>	Anyone that needs information on how the ANGI(M) can find the home position of your machine.
4	<b><i>INSTALLING THE ANGI(M)</i></b>	Anyone that must install an ANGI(M) on a machine. Includes information on mounting, grounding, and wiring specific to the units.
5	<b><i>NETWORK CONFIGURATION</i></b>	Anyone that must set the communication parameters of the ANGI(M) using the Configurator utility from AMCI.
7	<b><i>Modbus INTERFACE</i></b>	Anyone that needs information on the memory mapping from Modbus to the ANGI(M) and supported Modbus Functions and Exceptions.
8	<b><i>CONFIGURATION MODE DATA FORMAT</i></b>	Anyone that needs detailed information on the format of the network data to and from the ANGI(M) used to configure the indexer/driver.
9	<b><i>COMMAND MODE DATA FORMAT</i></b>	Anyone that needs detailed information on the format of the network data to and from the ANGI(M) used to command the indexer/driver.
B	<b><i>CHOOSING YOUR MOTOR</i></b>	Anyone that needs information on sizing an AMCI motor for your application.

## INTRODUCTION TO THE ANG1(M)

This manual is designed to get you quickly up and running with the ANG1(M) stepper indexer / driver. As such, it assumes you have some basic knowledge of stepper systems, such as the resolution you want run your motor at, and the reasons why you'd want to use Idle Current Reduction and the reasons why you wouldn't. If these terms or ideas are new to you, we're here to help. AMCI has a great deal of information on our website and we are adding more all the time. If you can't find what you're looking for at <http://www.amci.com>, send us an e-mail or call us. We're here to support you with all of our knowledge and experience.

### **AnyNET-I/O**

The ANG1 and ANG1M are two modules in a new product line from AMCI. The concept of this product line is simple: low-cost, specialty and/or high speed I/O, that can be attached to any popular industrial network; hence the name *AnyNET-I/O*.

AnyNet-I/O is designed for a broad range of applications, from small machines with a single control enclosure, to large machines that extensively use distributed I/O to minimize wiring costs.

What makes the AnyNET-I/O line different is that all of the modules are available with or without a network interface. Eliminating the need for a separate networking module lowers the total cost of ownership for all applications, but especially for the cost sensitive small machines that only require one or two sophisticated functions.

Like many low cost controllers, AnyNet-I/O modules are designed to be DIN rail mounted. Up to six AnyNET-I/O modules can be stacked together and accessed over a single network interface. "Stacking" is accomplished through a small back-plane connector that snaps into the DIN rail before the AnyNET-I/O modules are installed. These connectors allow the AnyNET-I/O modules to communicate with each other. To the network, the stack of modules appear as one continuous block of I/O words.



Figure 1.1 AnyNET-I/O Module Stack

### **The ANG1 and ANG1M**

#### **General Functionality**

The ANG1(M) is a 4.0Arms micro-stepping driver that accepts 24 to 48Vdc as its input power source. What makes the ANG1M unique is its built-in indexer that accepts configuration and command data from a host system over its internal RS-485 serial port. Data to an ANG1, or any other module in the AnyNet-I/O line, is automatically transferred to it based on the data addressing. The ANG1M is configured to use the Modbus-RTU protocol, which makes the ANG1M 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 ANG1(M)
- All motion logic is programmed in the host, eliminating the need to learn a separate motion control language
- Eliminating the separate indexer lowers Total System Cost



## The ANG1 and ANG1M (continued)

### General Functionality (continued)

The ANG1(M) 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 0.1Arms to 4.0Arms which makes the ANG1(M) compatible with AMCI's complete line of size 23 and size 34 stepper motors. In addition to the Motor Current setting, the Motor Steps per Turn, Idle Current Reduction, and Anti-Resonance Circuit features are also fully programmable. If you have used other stepper indexer products from AMCI you will find programming to be very similar to these products.

The ANG1(M) is a true RMS motor current control driver. This means that you will always receive the motor's rated torque regardless of the *Motor Steps/Turn* setting. (Drivers that control the peak current to the motor experience a 30% decrease in motor torque when microstepping a motor.) The ANG1(M) automatically switches from RMS to peak current control when the motor is idle to prevent overheating the motor.

In addition to power and motor hookups, the ANG1(M) has three DC inputs and one DC output that are used by the indexer. Configuration data from the host sets the function of these points. The output can be configured to be a Fault Output or a general purpose output. Each input can be individually configured as a:

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

### Encoder Functionality

In addition to the discrete I/O points, the ANG1(M) has three inputs for a 5Vdc differential quadrature encoder. The inputs will also accept 12 to 24Vdc single ended encoder inputs with current limiting resistors.

Using the encoder inputs gives you the ability to:

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

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

Finally, the encoder allows you to use a feature called *Hybrid Control*. When enabled, this feature gives you servo-like performance without the need to tune servo PID loop parameters. When Hybrid Control is enabled, the ANG1(M) uses a motor mounted encoder to feed rotor position information back to the driver. The ANG1(M) uses this information during a move to dynamically increase or decrease the current through the motor's windings to the lowest values needed to achieve motion. This feature eliminates the possibility of resonance related motor stalls and can dramatically lower the motor's operating temperature, which prolongs the life of the motor.

When using this feature, a stepper motor will act in a manner very similar to a servo motor. This includes the possibility of a one count dither in position at zero speed. In order to overcome this condition, the ANG1(M) allows you to enable and disable the Hybrid Control feature on the fly. When the motor is at rest, a single message disables the Hybrid Control feature, and the motor acts as a stepper system, holding its position without dither. You can enable the Hybrid Control feature using the same message that initiates a move, minimizing the amount of motor current needed to complete it.



## ***Indexer Functionality***

The table below lists the functionality offered by the indexer built into the ANG1(M)

Feature	Description
Modbus-RTU	Allows easy setup and communication with a wide range of host controllers.
Programmable Inputs	Each of the three inputs can be programmed as a Home Limit, Over Travel Limit, Capture Input, Manual Jog Stop, E-Stop Input, or a General Purpose Input.
Programmable Output	The single output on the ANG1(M) can be programmed as a Fault Output or as a general purpose DC output point.
Encoder Inputs	Allows the ANG1(M) to use a quadrature encoder for position verification or Encoder Following.
Programmable Parameters	Starting Speed, Running Speed, Acceleration, Deceleration, and Accel/Decel Types are fully programmable.
Homing	Allows you to set the machine to a known position. The ANG1(M) can home to a discrete input or to an encoder marker pulse.
Jog Move	Allows you to jog the motor in either direction based on an input bit from your host controller.
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.
Relative Move	Allows you to drive the motor a specific number of steps in either direction from the current location.
Absolute Move	Allows you to drive the motor from one known location to another known location.
Blend Move	Allows you to perform a sequence of relative moves without stopping between them.
Dwell Move	Allows you to perform a sequence of relative moves with a stop between each move that has a programmable length of time. Used to create highly accurate move profiles that avoid network latency issues.
Indexer Move	Allows you to program a move that 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 restart it without losing your position value.
Resume Move	Allows you to restart a previously held move operation.
Immediate Stop	Allows you to immediately stop all motion if an error condition is detected by your host controller.
Stall Detection	When the ANG1(M) uses an encoder, the encoder can be used to verify motion when a move command is issued.
Encoder Follower	The ANG1(M) can be configured to control the position of a motor based on feedback from an external encoder. The ratio of encoder pulses to motor pulses is fully programmable and can be changed on-the-fly.
Hybrid Control	The ANG1(M) can be configured to use a motor mounted encoder for rotor position feedback and use this data to dynamically alter motor current to the minimum value needed to achieve motion. This operating mode eliminates the possibility of resonance induced motor stalls and can significantly reduce the operating temperature of the motor.

Table 1.1 Indexer Functionality



## Driver Functionality

This table summarizes the features of the stepper motor driver portion of the ANG1(M).

Feature	Benefits
RMS Current Control	RMS current control give the ANG1(M) 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 4.0 amps in 0.1 amp increments. This allows you to use the driver with the full line of AMCI stepper motors.
Programmable Idle Current Reduction	Extends motor life by reducing the motor current when not running. This extends the life of the motor by reducing its operating temperature.
Programmable Current Loop Gain	Allows you to tailor the driver circuitry to the motor's impedances, thereby maximizing your motor's performance.
Programmable Motor Steps/Turn	Allows you to scale your motor count to a real world value. (counts per inch, counts per degree, etc.)
Anti-Resonance Circuitry	This circuitry gives the ANG1(M) the ability to modify motor current waveforms to compensate for mechanical resonance in your system. This will give you smooth performance over the entire speed range of the motor.
Wiring Short Detection	Safety feature that removes power from the motor if a short is detected in one of the windings of the motor.
Over Temperature Detection	The ANG1(M) sets a warning bit in the network data when the temperature of the module approaches its safe operating threshold.
Over Temperature Protection	Protects the ANG1(M) from damage by removing power from the motor if the internal temperature of the driver exceed a safe operating threshold.

Table 1.2 Driver Functionality

## I/O Connector

As shown in figure 1.2, the I/O connector is located on the top of the module. All digital I/O connections are made at this connector as well as the power supply connections. The mating connector is not supplied with the ANG1(M) and is available from AMCI under the part number MS-2X11. It is also available from Phoenix Contact under their part number 173 88 98.

### Power Supply Sizing

The power supply is connected to the pins marked "+Vdc In" and "GND". The 24 to 48Vdc external power supply also powers the stepper motor, so it must be rated to supply current to it. The general rule of thumb is to specify a supply that can output the maximum current for your motor. (As stated previously, the maximum output current for the ANG1(M) is 4.0Arms.)

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.

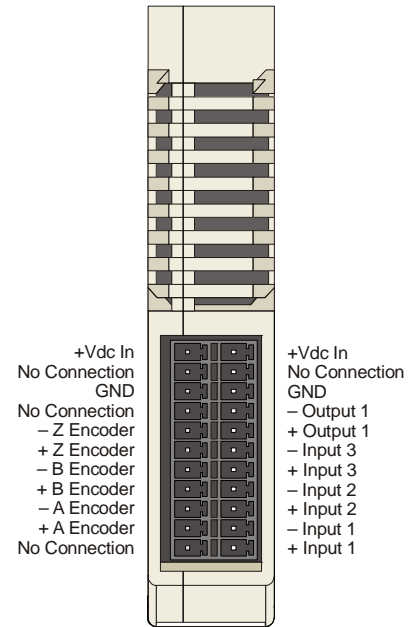


Figure 1.2 I/O Connector

## **Available Discrete Inputs**

The ANG1(M) has three discrete DC inputs that accept 3.5 to 27Vdc signals. (5 to 24Vdc nominal) They can be wired as differential, sinking, or sourcing inputs. How the ANG1(M) uses these inputs is fully programmable as is their active states. (Normally Open(NO) or Normally Closed(NC) inputs.)

### **Home Input**

Many applications require that the machine be brought to a known position before normal operation can begin. This is commonly called “homing” the machine or bringing the machine to its “home” position. The ANG1(M) allows you to define this starting position in three ways. The first is with a Position Preset Command. The second is with a sensor mounted on the machine. When you define one of the inputs as the Home Input, you can issue commands to the ANG1(M) that will cause the unit to seek this sensor. The third option is homing to the Z pulse of a quadrature encoder. When using the Z pulse, you can use one of the inputs or a backplane as a home proximity sensor. How the ANG1(M) actually finds the Home sensor is described in the [HOMING THE ANG1\(M\)](#) chapter starting on page 39.

### **CW Limit Switch or CCW Limit Switch**

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

### **Start Indexer Move Input**

Indexer Moves are programmed through the Network Data like every other move. The only difference is that Indexer Moves are not run until a Start Indexer Move Input makes an inactive-to-active state transition. This allows the ANG1(M) to run critically timed moves that cannot be reliably started from the network due to data transfer lags.

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

### **Emergency Stop Input**

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

### **Stop Jog or Registration Move Input**

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

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

### **Capture Encoder Position Input**

As described in the [Start Indexer Move Input](#) and [Stop Jog or Registration Move Input](#) sections above, the ANG1(M) can be configured to capture the encoder position value on a transition on the discrete DC input.

### **General Purpose Input**

If your application does not require all three inputs, you can configure the unused inputs as General Purpose Inputs. The inputs are not used by the ANG1(M), but the input state is reported in the network data.



## **Encoder Feedback Inputs**

The ANG1(M) has three 5Vdc differential inputs that accept quadrature encoder signals.

An encoder is used by the ANG1(M) in multiple ways.

- When the encoder is mounted on the back of the motor controlled the ANG1(M), the encoder can be used for position feedback or stall detection. The position data of the encoder can be preset to any value within its range, it is reported in the network data, and it can be captured during a move.
- The Z pulse can be used to home the machine as described in the [HOMING THE ANG1\(M\)](#) chapter starting on page 39. A home proximity sensor can be wired into the ANG1(M) or the Home Proximity Bit in the network data can be used to determine which occurrence of the Z-pulse defines the home of the machine.
- The addition of an encoder also allows you to perform an *Encoder Move*, which is described in the [Encoder Moves](#) section, starting on page 28.
- The ANG1(M) must use an encoder with a feature called *Hybrid Control*. When using this feature, the ANG1(M) uses feedback from the motor mounted encoder to dynamically adjust motor current to the minimum value needed to achieve motion. Hybrid Control will also eliminate resonance induced stalls during the move.
- Finally, it is also possible to use an encoder that is not mechanically coupled to the motor controlled by the ANG1(M). This configuration allows you to monitor the encoder data, or use a feature called *Encoder Follower*. When this feature is active, the ANG1(M) will change the position of the motor in response to a change in encoder position. The ratio of encoder turns to motor turns is fully programmable.

## **Available Discrete Output**

The ANG1(M) has a single DC output that has a maximum rating of 30Vdc at 20mA. The output can be configured to be a general purpose output or a Fault Output. When configured as a Fault Output, the output will conduct under normal conditions and will switch off when a fault occurs. The following faults affect the Fault Output:

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

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

## Modbus Network Connector

The Modbus Network Connector is located on the bottom of the ANG1M. Connections to the Common, D0, and D1 signals are paralleled on two pins to simplify a Trunk Interface (ITr) connection. The ANG1M can also be used with a Passive Tap (IDv) connection.

The +5Vdc is a 50 mA supply that is available for Line Polarization resistors if the system requires them. Note that the ANG1M does not require Line Polarization for proper operation.

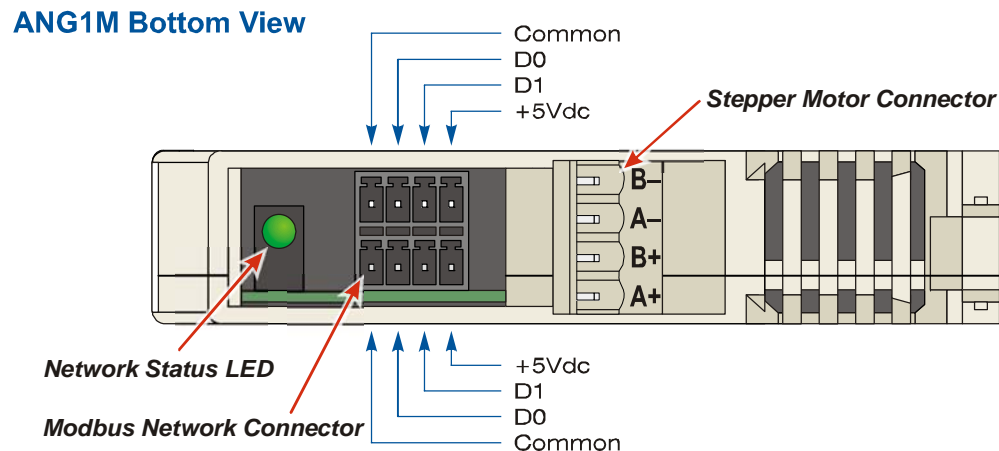


Figure 1.3 Modbus Network and Motor Connector Locations

## Mating Connector

The mating connector is sold separately by AMCI under the part number MS-2X4. It is also available from Phoenix Contact under their part number 1738827.

## Network Status LED

The Network Status LED shown in figure 1.3 indicate the health of the serial connection between the ANG1M and its host.

- ▶ **Steady Green** - The ANG1M is powered, but there is no active communication between the unit and its host.
- ▶ **Blinking Green** - The ANG1M is successfully communicating with its host over the Modbus Serial Line network
- ▶ **Blinking Red** - The ANG1M received a malformed packet over the Modbus Serial Line network. This is typically caused by noise on the RS-485 line, an improperly terminated RS-485 trunk cable, or problems with one of the devices on the trunk.

The Status LED on the front of the unit indicates the state of the ANG1M itself.

## Motor Connector

Figure 1.3 also shows the location of the Stepper Motor Connector. The mate to this connector is not included with the ANG1(M) and is available from AMCI under the part number MS-4M. It is also available from Phoenix Contact under their part number 187 80 37.

## Front Panel

The front panels of three ANG1(M) modules are shown in figure 1.4. The front cover is hinged on the bottom, and swings down to allow you to change the DIP switch address settings. The front panel also has the Status LED, which give you information on the state of the module.

Figure 1.4 shows the ANG1M with the front cover removed. It shows the location of the two rotary switches that set the address of the ANG1M on the Modbus Serial Line network as well as the USB port that is used to configure the ANG1M.

### AnyNet-I/O Address Settings

The AnyNET-I/O platform allows you to connect up to six modules to a single network connection in what we call an AnyNET-I/O Stack. The DIP switch behind the front panel cover is used to set the address of the module within the AnyNET-I/O Stack. The first module acts as the network interface and must have an address of zero. This address is set by having all of the DIP switches in their OFF position. (If you are using a single module, then it must have an address of zero.) The remaining modules in the Stack should have their addresses set to their position in the stack by setting the corresponding DIP switch to its ON position. Figure 1.4 shows the correct addressing for three modules. The module on the left is an ANG1M and has its address set to zero. The remaining modules can be ANG1, ANG1M, or even ANG1E modules and their addresses are set to one and two.



**NOTE** If an ANG1M has its address set to any value other than zero, its network interface is disabled. This is true for all AnyNet-I/O modules that have a network interface. This allows you to use multiple ANG1M modules in a single AnyNET-I/O Stack.

### Modbus Network Address Settings

The address of the AnyNet-I/O Stack on the Modbus network is set by two rotary switches. Their placement in the module is shown in figure 1.4. Figure 1.5 shows a three dimensional view of the switches to clearly show their proper settings. Each switch has a cutout on the shaft which shows the number selected by the switch. The top switch sets the first digit of the address, and the bottom switch sets the second digit. The figure shows a Modbus address of 02.

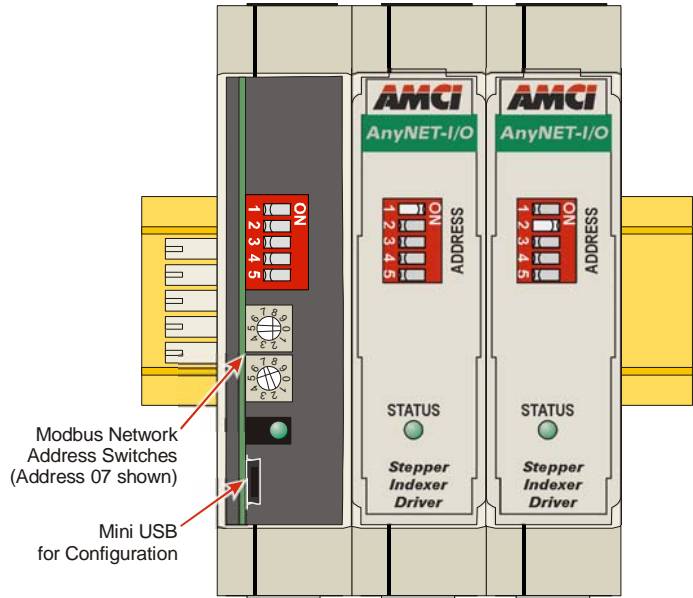


Figure 1.4 ANG1(M) Front Panel

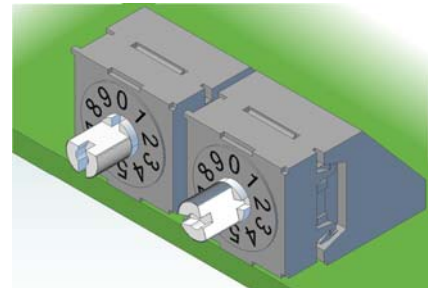


Figure 1.5 Modbus Address Switches



## ***Front Panel (continued)***

### **Status LED**

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

- **Steady Green:** Module OK
- **Steady Red:** An Overtemperature Fault or Motor Short Circuit Fault exists. Note that the ANG1(M) will only detect short circuit faults when the motor current is enabled.
- **Blinking Green:** Successfully write to flash memory. Power must be cycled to the module before additional commands can be written to it.
- **Blinking Red:** Failed write to flash memory. You must cycle power to the module to clear this fault.
- **Alternating Red/Green:** Communications failure. There is a communications error between modules in the AnyNET-I/O Stack. You must cycle power to the stack to attempt to clear this fault.

### **Micro USB Connector**

The micro USB connector allows you to easily configure the ANG1M with the latest Configurator software from AMCI. This software can be downloaded from the *Tech Library* section of our site at <http://www.amci.com/product-software.asp>. It may be necessary to download and install a Microsoft Windows driver file before you can communicate with the ANG1M. This driver file is located at the same web-site address as the Configurator software.

The micro USB connection will appear as a serial port to your computer that supports a limited subset of the Modbus protocol. Even though it is not recommended for a production environment, it is possible to control the AnyNet-I/O Stack over this connection.



## Specifications

### Driver Type

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

### Physical Dimensions

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

### Weight

0.38 lbs. (0.17 kg.) with mating connectors

### Inputs

Electrical Characteristics: .....  
Differential. 560 Vac/dc opto-isolated. Can be wired as single ended inputs.  
DC Inputs accept 3.5 to 27Vdc without the need for an external current limiting resistor.  
Encoder Inputs are designed for 5Vdc differential and require external current limiting resistor for 12 to 24Vdc operation

### Output

Electrical Characteristics:  
Open Collector/Emitter. 2500 Vac/dc opto-isolated. 30Vdc, 20 mA max.  
The Output can be programmed to be a general purpose output or a Fault Output.  
The Fault Output is normally on. Turns off under the following conditions:  
Reset ..... The driver initialization is not yet complete on power up.  
Short Circuit ..... Motor Phase to Phase or Phase to Case  
Over Temp ..... Heat Sink temperature exceeds 90° C (195° F)  
Faults are reported in the Network Input Data and can be cleared through the Network Output Data.

### Motor Current

Programmable from 0.1 to 4.0Arms in 0.1 Amp steps.

### Resolution

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

### Idle Current Reduction

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

### Internal Power Fuse

7 Amp Fast Blow. Fuse is not user replaceable.

### Environmental Specifications

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

### Motor Specifications

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

### Modbus Parameters

Address Range ..... 1 to 99  
RS485 Unit Load ... 1/8 (256 devices max.)  
Protocol ..... RTU  
Baud Rate ..... 2,400, 9,600, 14,400, 19,200, 38,400, 57,600, 115,200. (Default of 19,200)  
Parity Checking ..... Even, Odd, or None  
Interface IC ..... ADM2485 (2.5kV galvanic isolation)

### Connectors

Mating connectors are available separately under the following AMCI part numbers.

Connector	AMCI Part #	Phoenix Contact #	Wire	Strip Length	Min. Tightening Torque
Network	MS-2X4	173 88 27	28 - 16 AWG	0.275 inches	Spring Cage Connector
I/O	MS-2x11	173 88 98	28 - 16 AWG	0.275 inches	Spring Cage Connector
Motor	MS-4M	187 80 37	28 - 12 AWG	0.394 inches	4.43lb-in (0.5 Nm)

# CHAPTER 2

## MOVE PROFILES

When a move command is sent to the ANG1(M), the module calculates the entire profile before starting the move or issuing an error message. This chapter explains how the profiles are calculated and the different available moves.

### **Units of Measure**

**Distance:** Every distance is measured in steps. Your driver and motor combination 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 driver and motor combination, 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 the ANG1(M), 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 the ANG1(M).
- ▶ 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 the ANG1(M) 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 output pulses for a clockwise move.

**NOTE**  The actual direction of rotation of a stepper motor can be reversed by switching one of the motor pairs. Therefore, you can easily have positive commands result in counter-clockwise moves if your application requires it.

### **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 ±Find Home commands. If you use the module's ±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 a servo or stepper for speed control, don't require position data at all.

### **Definition of Starting Speed**

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

### **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 define the Target Position as an offset from the present position of the motor. Most ANG1(M) 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.

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 if the Target Position is greater than the Current Position and 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  $\pm 1,073,741,823$  counts with an Absolute Move. The only way to move beyond  $\pm 8,388,607$  counts is with multiple relative moves.

### **Definition of Acceleration Types**

With the exception of all homing operations, most of the move commands allow you to define the acceleration type used during the move. The ANG1(M) supports three types of accelerations and decelerations. The type of acceleration used is controlled by the *Acceleration Jerk* parameter.

#### **Linear Acceleration**

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

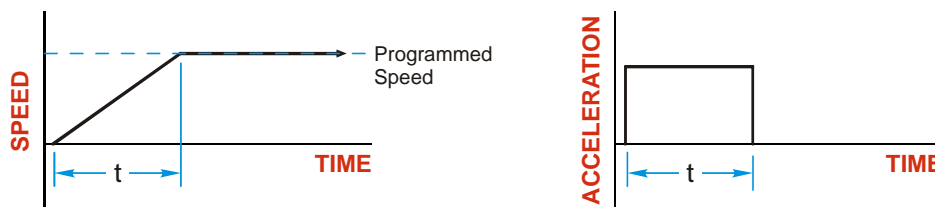


Figure 2.1 Linear Acceleration

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

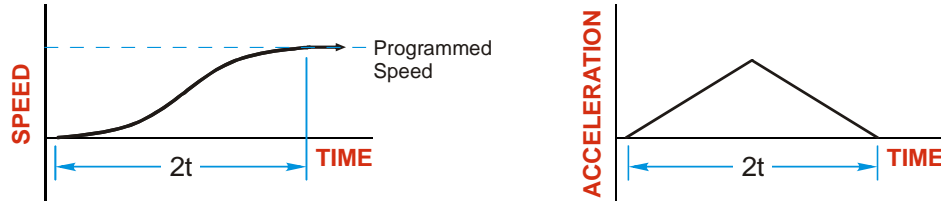


Figure 2.2 Triangular S-Curve Acceleration

### 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.

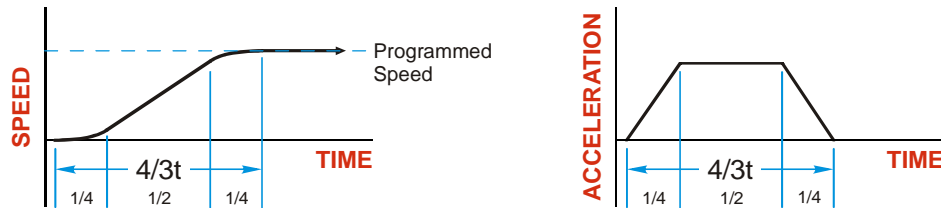


Figure 2.3 Trapezoidal S-Curve Acceleration



## A Simple Move

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

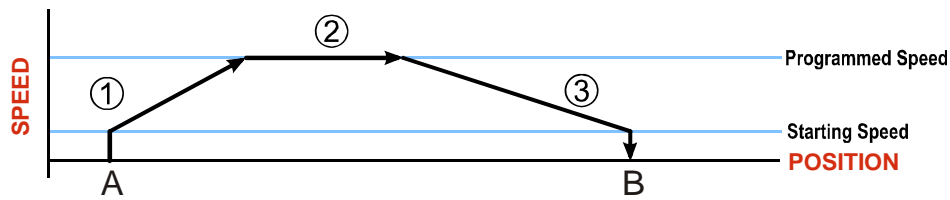


Figure 2.4 A Trapezoidal Profile

- 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*. Both the *Acceleration Value* and the *Programmed Speed* are programmed when the move command is sent to the ANG1(M).
- 2) The motor continues to run at the *Programmed Speed* until it reaches the point where it must decelerate before reaching point B.
- 3) The motor decelerates at the *Deceleration Value*, which is also programmed by the move command, until the speed reaches the *Starting Speed*, which occurs at the *Target Position (B)*. The motor stops at this point. Note that the acceleration and deceleration values can be different in the move.

Figure 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 decelerating and the *Programmed Speed* is never reached.

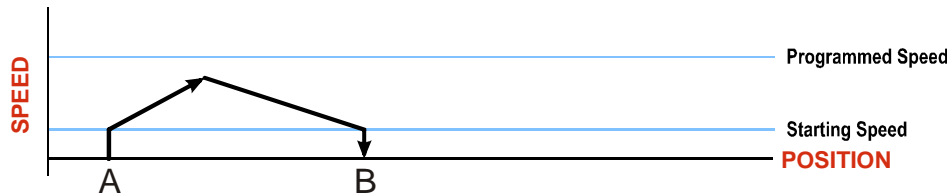


Figure 2.5 A Triangular Profile

## 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 again before Absolute Moves can be run.

### Host Control

**Hold Move Command:** This command can be used with some moves to bring the axis to a Controlled Stop. The move can be resumed and finished, or it can be aborted. Not all moves are affected by this command. The section *Basic Move Types*, starting on page 23, describes each move type in detail, including if the move is affected by this command.

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

### Hardware Control

**CW Limit and CCW Limit Inputs:** In most cases, activating these inputs during a move will bring the axis to an Immediate Stop. The exceptions are the  $\pm$ 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 ANGI(M)*, which starts on page 39. The  $\pm$ Jog Move commands are fully explained on page 25, and the  $\pm$ Registration Move commands are fully explained on page 27.

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

## Basic Move Types

### Relative Move

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

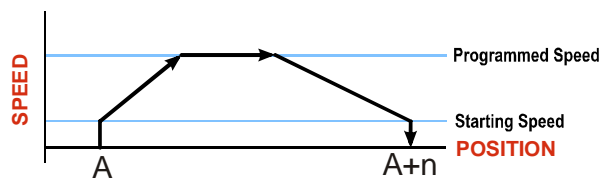


Figure 2.6 Relative Move

#### 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 the network data. When the command is accepted, the axis will immediately decelerate at the programmed rate and stop. When stopped successfully, the ANG1(M) will set a *Hold State* bit in the input data table. The Relative Move can be restarted with the Resume Move command from the network data or the move can be aborted. The Resume Move command allows you to change the move's Programmed Speed, Acceleration Value and Type, and the Deceleration Value and Type. The Target Position cannot be changed with the Resume Move Command.

#### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Relative Move by using the Resume Move command. The use of the Hold Move and Resume Move bits is further explained in the *Controlling Moves In Progress* section starting on page 34.

#### 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 ANG1(M) calculates the number of steps needed to move to the given position and moves that number of steps.) A trapezoidal profile is shown to the right, but Absolute Moves can also generate triangular profiles. The command's Target Position can be in the range of  $\pm 8,388,607$  counts. The move will be clockwise if the Target Position is greater than the Current Position and counter-clockwise if the Target Position is less than the Current Position.

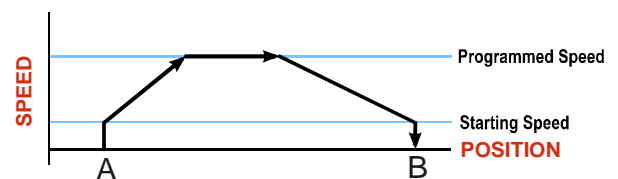


Figure 2.7 Absolute Move

#### NOTE

- 1) The *Home Position* of the machine must be set before running an Absolute Move. See chapter 3, *HOMING THE ANG1(M)*, which starts on page 39, for information on homing the machine.
- 2) The Motor Position must be valid before you can use an Absolute Move. The Motor Position becomes valid when you preset the position or home the machine.
- 3) Absolute Moves allow you to move your machine without having to calculate relative positions. If you are controlling a rotary table, you can drive the table to any angle without having to calculate the distance to travel. For example an Absolute Move to 180° will move the table to the correct position regardless of where the move starts from.

## Basic Move Types (continued)

### Absolute Move (continued)

#### Controlled Stop Conditions


- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Absolute Move by using the Resume Move bit. The use of the Hold Move and Resume Move bits is explained in the *Controlling Moves In Progress* section starting on page 34.

#### 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 output CW steps while the –Jog Move will output CCW steps. These commands are often used to give the operator manual control over the axis.

**NOTE**  The Target Position value must be zero during a Jog Move. If it is not zero, the move will actually be 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 that is the same as the direction of travel, for example activating the CW Limit during a +Jog Move, will bring the move to an Immediate Stop. Unlike the other moves, activating a limit that is opposite to the direction of travel, for example activating the CCW Limit during a +Jog Move, has no effect. This allows you to jog off of the activated limit switch.

As shown below, a Jog Moves begins at the programmed Starting Speed, accelerates at the programmed rate to the Programmed Speed and continues until a stop condition occurs. If it is a *Controlled Stop Condition*, the ANG1(M) 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.

**Basic Move Types (continued)****± Jog Move (continued)**

It is possible to change the speed of a Jog Move without stopping the motion. The Programmed Speed, Acceleration, and Deceleration parameters can be changed during a Jog Move. When the Programmed Speed is changed, the motor will accelerate or decelerate to the new Programmed Speed using the new accelerate/ decelerate parameter values. If you write a Programmed Speed to the ANG1(M) that is less than the starting speed, the Jog Move will continue at the Starting Speed.

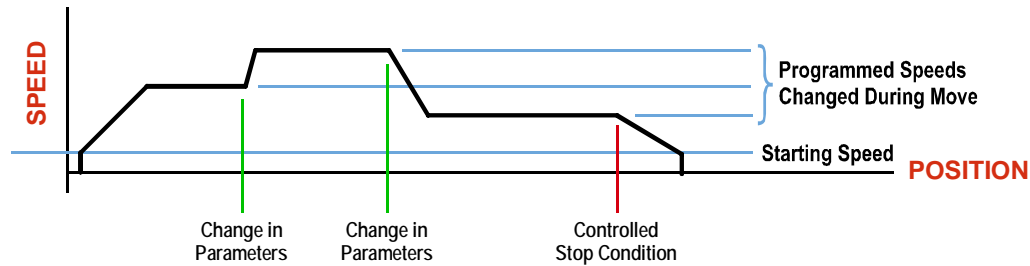


Figure 2.8 Jog Move

**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 34.

**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 CW output pulses, -Registration Moves result in CCW output pulses. When the command terminates under Controlled Stop conditions, the ANG1(M) 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.

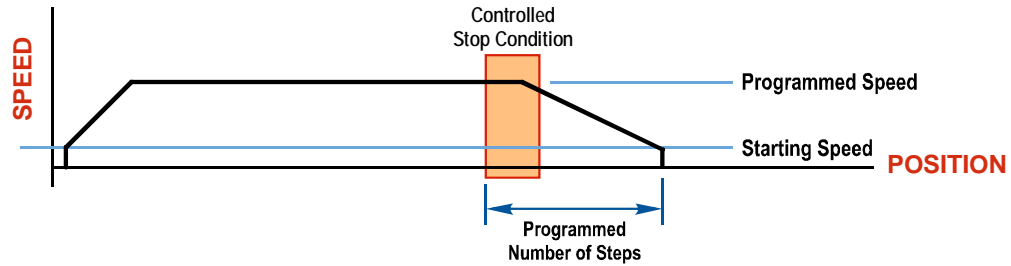


Figure 2.9 Registration Move

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 ANG1(M) will decelerate at the programmed Deceleration value until it has output the Programmed Number of Steps and then stop the move without further deceleration.

Like the ±Jog Moves, activating a limit that is opposite to the direction of travel, for example activating the CCW Limit during a +Registration Move, has no effect.

An additional feature of the ANG1(M) is the ability to program the module to ignore the Controlled Stop conditions until a minimum number of steps have occurred. This value is programmed through the Minimum Registration Move Distance parameter, which is set when you command the Registration Move. The figure below shows how the Minimum Registration Move Distance parameter affects when the Stop Condition is applied to the move. As shown in the second diagram, Controlled Stop conditions are level triggered, not edge triggered. If a Controlled Stop Condition occurs before the Minimum Registration Move Distance is reached and stays active, the move will begin its controlled stop once the Minimum Registration Move Distance is reached.

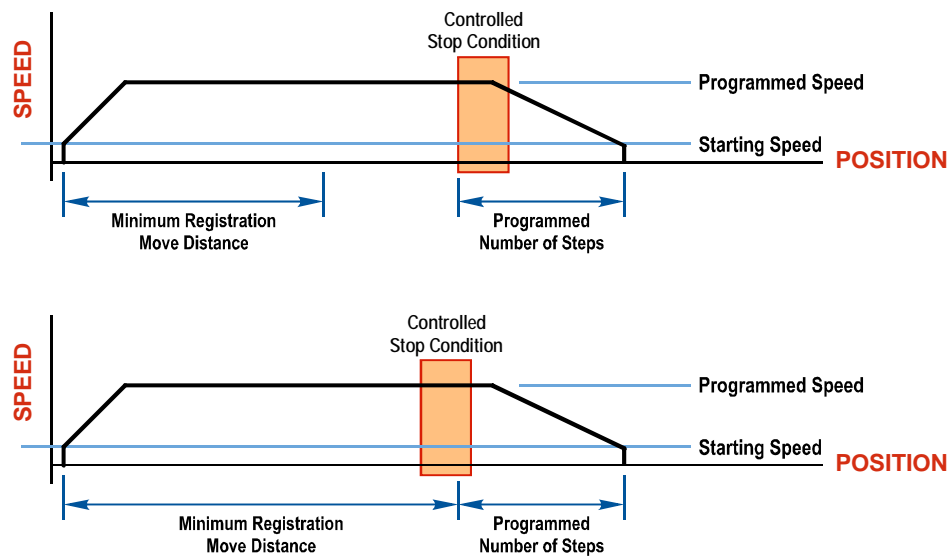


Figure 2.10 Min. Registration Move Distance

**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 ANGI(M) responds by using the programmed Deceleration value to bring the move to a stop, without using the programmed Programmed Number of Steps. A Registration Move does not go into the Hold State if the Hold Move control bit is used to stop the move and it cannot be restarted.

*Immediate Stop Conditions*

- The Immediate Stop bit makes a 0 → 1 transition in the Network Input Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW/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 the ANGI(M) is configured to use a quadrature encoder, 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 you program in the move defines the point at which the motor begins to decelerate and stop. *It does not define the stopping position as it does in other move types.* The endpoint of the move depends on the speed of the motor when the programmed encoder position is reached and the deceleration values. This behavior is different from Absolute and Relative Moves where the position you program into the move is the end point of the move.

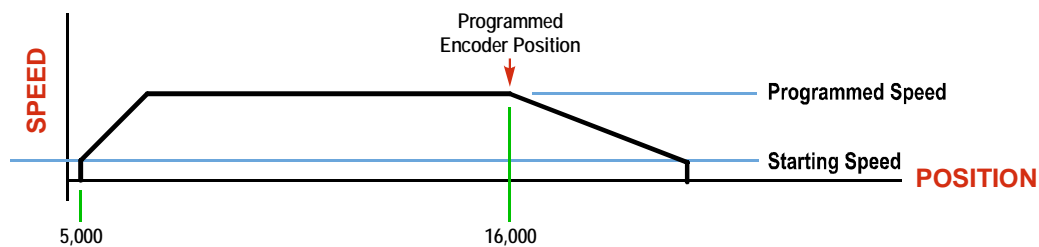


Figure 2.11 Encoder Move

## 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. The use of the Hold Move and Resume Move bits is explained in the [Controlling Moves In Progress](#) section starting on page 34.

### Immediate Stop Conditions

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

## Assembled Moves

All of the moves explained so far must be run individually to their completion or be stopped before another move can begin. The ANG1(M) gives you the ability to assemble more complex profiles from a series of relative moves that are stored in memory and then run with a single command. Each Assembled Move can consist of 2 to 16 segments. Two types of Assembled Moves exist in the ANG1(M):

- **Blend Move** - A Blend Move gives you the ability to string multiple relative moves together and run all of them sequentially without stopping the shaft between moves. All of the moves are in the same direction.
- **Dwell Move** - A Dwell Move gives you the ability to string multiple relative moves together, and the ANG1(M) 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 ANG1(M) before the move can be initiated. The ANG1(M) 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 so only the magnitude of the target position is used. (The sign of the target position is ignored.) 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](#) that is explained on page 31.
- 4) The Programmed Speed of each segment must be greater than or equal to the Starting Speed. The Programmed Speed must also be different from the previous segment. If you are accelerating between segments then the acceleration value of the new segment is used. If you are decelerating between segments, the deceleration value of the old segment is used.

## 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.

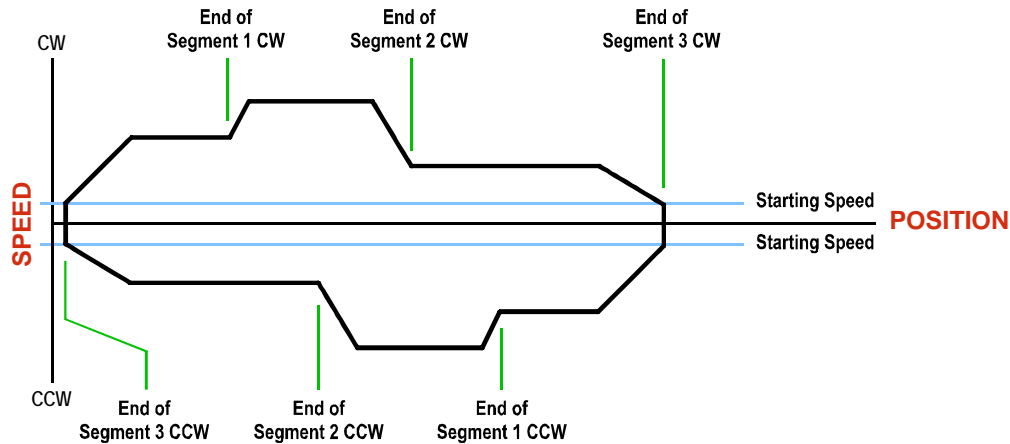


Figure 2.12 Blend Move

#### 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 ANG1(M) 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 ANG1(M). As described in [Saving an Assembled Move in Flash](#) on page 32, it is also possible to save a Blend Move to flash memory. This move is restored on power up and can be run as soon as you configure the ANG1(M) and enter Command Mode.
- 3) There are two control bits used to specify which direction the Blend Move is run in. This gives you the ability to run the Blend Move in either direction.

#### Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. When this occurs, the ANG1(M) decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. The use of the Hold Move bit is explained in the [Controlling Moves In Progress](#) section starting on page 34.

#### 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 the ANG1(M) as a series of relative moves before the move is started.

Unlike a Blend Move, the motor is stopped between each segment of the Dwell Move for a programmed *Dwell Time*. The Dwell Time is programmed as part of 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 moves while a negative segment will result in a counter-clockwise move. The following figure shows a drilling profile that enters the part in stages and reverses direction during the drilling operation so chips can be relieved from the bit.

You can accomplish this Dwell Move with a series of six relative moves that are sent down to the ANG1(M) sequentially. The two advantages of a Dwell Move in this case is that the ANG1(M) will be more accurate with the Dwell Time than you could accomplish in the PLC, and Dwell Moves simplifies your program's logic.

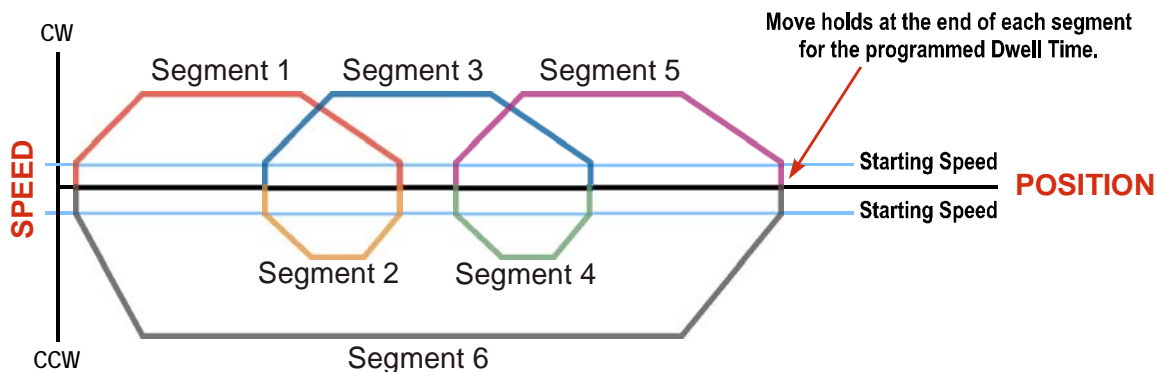


Figure 2.13 Dwell Move

#### 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 the ANG1(M) 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 ANG1(M). As described in [Saving an Assembled Move in Flash](#) on page 32, it is also possible to save a Dwell Move to flash memory. This move is restored on power up and can be run as soon as you configure the ANG1(M) 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 ANG1(M) decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. The use of the Hold Move bit is explained in the [Controlling Moves In Progress](#) section starting on page 34.

### 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 ANG1(M) 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, there is a bit in the command data that 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 and the programmed Dwell Time is written to the ANG1(M) with the command.

### **Control Bits – Output Data**

- **Program Move bit** – Set this bit to tell the ANG1(M) that you want to program a Blend or Dwell Move Profile. The ANG1(M) will respond by setting the *Move Program Mode* bit in the Network Input Data. At the beginning of the programming cycle, the ANG1(M) 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 ANG1(M) that the data for the next segment is available in the remaining data words.

### **Control Bits – Input Data**

- **Move Program Mode bit** – The ANG1(M) sets this bit to tell you that it is ready to accept segment programming data in the remaining output data words. The actual transfer of segment data is controlled by the *Program Move Segment* and *Transmit Move Segment* bits.
- **Transmit Move Segment bit** – The ANG1(M) 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 ANG1(M) 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 ANG1(M) 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 ANG1(M) 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 from 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 ANG1(M) resets the *Move Program Mode* bit and the *Transmit Blend Move Segment* bit.

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

The ANG1(M) 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 ANG1(M) 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 ANG1(M) before you can continue. With a limit of 10,000 write cycles, the design decision that requires you to cycle power to the ANG1(M) was made to prevent an application from damaging the module by continuously writing to it.

### ***Indexed Moves***

All of the moves that have been explained in the chapter up to this point can be started by a transition on one of the three inputs instead of a command from the network. If the *Indexed Move* bit is set when the command is issued, the ANG1(M) 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 ANG1(M) until the triggered move is finished.
- ▶ As started above, a move can be run multiple times as long as the move command data remains unchanged. If you wish to program a second move and run it as an Indexed Move type, then you must have a 0→1 transition on the move command bit before the new parameters are accepted. The easiest way to accomplish this is by writing a value of 0x0000 to the command word between issuing move commands.
- ▶ A Jog Move that is started as an Indexed Move will come to a controlled stop when the command bit in the Network Output Data is reset to zero.
- ▶ It is possible to perform an Indexed Registration Move by configuring two inputs for their respective functions. The first input, configured as a *Start Indexed Move Input*, starts the move and the second, configured as a *Stop 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**

The ANG1(M) 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.

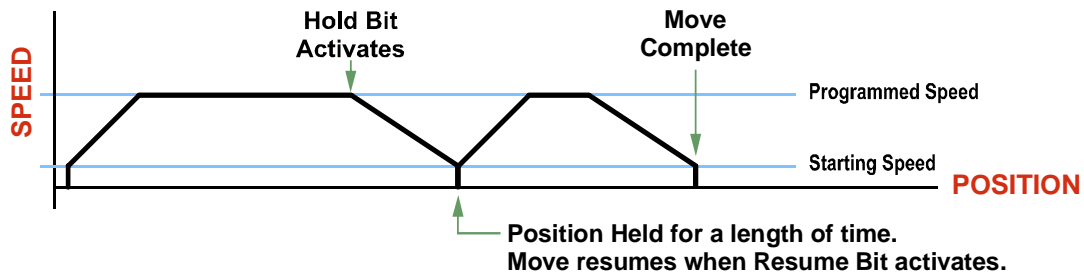


Figure 2.14 Hold/Resume a Move Profile

### **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 ANG1(M) 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 ANG1(M) 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.

## **Encoder Follower**

The final form of motion control available with the ANG1(M) is called Encoder Follower. A quadrature encoder is required but it is not mounted on the motor controlled by the ANG1(M). The encoder is typically mounted on a second motor, but it can be mounted anywhere, including on something as simple as a hand crank.

AMCI also refers to it as *Electronic Gearing* because the ANG1(M) has three parameters that allow you to set any turns ratio you want between the encoder and the motor.

### **Motor Steps/Turn**

This is the same parameter explained at the beginning of this chapter. In Encoder Follower mode, this parameter sets the number of encoder counts required to complete one rotation of the shaft of the motor driven by the ANG1(M). It has a range of 200 to 32,767. This parameter is programmed when you configure the module and cannot be adjusted while a move is in progress.

### **ELGearing Multiplier and Divisor**

The ratio of these two parameters is applied to the number of encoder pulses read by the ANG1(M) before it determines the number of motor steps to move. Each parameter has a range of 1 to 255. These two parameters can be adjusted while a move is in progress which allows you to adjust the tracking speed and position of the motor.

### **How It Works**

The ANG1(M) always uses 4X decoding when counting pulses from the encoder. If you set both of your ELGearing Multiplier and Divisors to 1 and set the Motor Steps/Turn to four times the number of encoder lines, then the motor will complete one rotation for every rotation of the encoder's shaft.

Once placed in Encoder Follower mode, the ANG1(M) monitors the Jog Move command bits in the Network Output Registers. When either of these bits is set, the encoder inputs are monitored for a change in position. When a change is sensed, the ANG1(M) will begin to turn the motor within 50 microseconds. An increase in encoder counts will result in clockwise rotation. A decrease in encoder counts will result in counter-clockwise rotation.

The values of the ELGearing Multiplier and Divisor can be changed while encoder follower motion is occurring. The ANG1(M) will accelerate or decelerate the motor to match the new ratio.

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

### **Controlled Stop Conditions**

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

### **Immediate Stop Conditions**

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



## **Encoder Follower (continued)**

### **Advanced Ratio Control**

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

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

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

### **Stall Detection**

Another feature available when using an encoder is stall detection. The encoder must be mounted on the motor controlled by the ANG1(M), which means that you cannot use Stall Detection when using Encoder Follower mode. When Stall Detection is enabled, the ANG1(M) 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 its correct value in the Configuration Data of the ANG1(M). The ANG1(M) always uses X4 decoding when determining the encoder position value, so the *Encoder Pulses/Turn* parameter must be set to four times the number of encoder lines. (When using a 1,024 line encoder, the *Encoder Pulses/Turn* parameter must equal 4,096 for stall detection to work correctly.)

### **Hybrid Control**

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

In order for the Hybrid Control feature to work correctly, the *Encoder Pulses/Turn* parameter must be set to four times the number of encoder lines. (When using a 1,024 line encoder, the *Encoder Pulses/Turn* parameter must equal 4,096.) The *Motor Steps/Turn* parameter must also be set to this value.

The ANG1(M) uses the encoder feedback during a move to dynamically increase or decrease the current through the motor's windings to the lowest values needed to achieve motion. This feature eliminates the possibility of resonance induced motor stalls and can dramatically lower the motor's operating temperature.

Like a servo system, there is the possibility of a one count dither in position at zero speed and a load induced error in the final position if this feature is always enabled. The "stiffness" of the shaft, (how quickly the system compensates for shaft position error), can be adjusted with a single gain setting. The ANG1(M) also allows you to enable and disable the Hybrid Control feature on the fly. When the motor is at rest, a single command will disable the Hybrid Control feature, and the motor will again act as a stepper system, holding its position without dither. You can then re-enable the Hybrid Control feature using the same command that initiates the next move, minimizing the amount of motor current needed to complete the move.

When using Hybrid Control, the ANG1(M) has an additional status bit, the *Motion Lag* bit, that turns on if the position error becomes larger than 360°. This status bit indicates that the speed of the move is not the one that is expected, but the move is still in progress. This can be an indication that something has changed mechanically in the system that has placed an additional load on the motor. Using this bit is the preferred method of performing stall detection when using Hybrid Control. When this bit turns on, monitor the encoder position value for change. If the encoder value is changing, then the motor is heavily loaded but still moving. If the encoder value is not changing, then the motor cannot rotate under load and the move can be stopped by your program.

### **Hybrid Control (continued)**


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

The stall detection feature that is built into the ANG1(M) can be used with the Hybrid Control feature. If the rotor position becomes more than 45° behind the commanded position, the motor will be considered stalled and the move will be terminated. Error bits will be set in the network data to inform you of the stalled condition. Note that the *Motion Lag* bit will never be set when using the stall detection feature of the ANG1(M) because it will trip before the 360° position error that will set the *Motion Lag* bit.

### **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 4Arms and the current reduction to 25% will result in an idle current of 1Apk. (The ANG1(M) always switches from RMS to peak current control when the motor is idle to prevent motor damage due to excessive heating.)

### **Current Loop Gain**

This feature gives you the ability to adjust the gain of the power amplifiers in the ANG1(M) to match the electrical characteristics of your motor. The value of this parameter can range from 1 to 40 with 40 representing the largest gain increase. In general, using a larger gain will increase high speed torque but the motor will run louder. A lower gain will offer quieter low speed operation at the cost of some high speed torque.

This parameter must be set, and a gain setting of “5” will work for all AMCI motors. Your system may benefit from increasing or decreasing these settings. In general, increase the setting by one or two counts to improve high speed performance or decrease the settings by one or two for quieter low speed operation.



*Notes*

# HOMING THE ANG1(M)

This chapter explains the various ways of homing an axis on the ANG1(M). Inputs used to home the module are introduced and diagrams that show how the module responds to a homing command are given.

### Definition of Home Position

The Home Position is any position on your machine that you can sense and stop at. Once at the Home Position, the motor position register of the ANG1(M) 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 enabled for the axis.



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

With the exception of Absolute Moves, the ANG1(M) 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 ANG1(M). Before doing this, you will need a way of sensing position outside the ANG1(M) module. The machine position data must be brought into the host, the correct preset value calculated, and this value written to the ANG1(M) axis 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 ANG1(M) to find the Home Position based on sensors brought into the unit. The +Find Home command begins searching by outputting CW pulses to the motor's driver and ends when the home sensor triggers while the ANG1(M) is outputting CW pulses at a low rate. The -Find Home command operates in the same way but starts and ends with CCW pulses.

### Homing Inputs

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

#### Physical Inputs

- **Home Input:** This input is used in one of two ways: 1) This input is used to define the actual home position of the machine. 2) The input is used as a home proximity input when using the encoder marker pulse to home the machine.
- **Encoder Marker (Z) Pulse:** If you configure the ANG1(M) to use an encoder, you have the option of using the encoder's marker pulse to home the machine.
- **CW Limit Switch Input:** This input is used to prevent overtravel in the clockwise direction.
- **CCW Limit Switch Input:** This input is used to prevent overtravel in the counter-clockwise direction.

#### Backplane Inputs

- **Home Proximity Bit:** The ANG1(M) can be configured to ignore changes on the physical homing input until the Home Proximity Bit makes a 0→1 transition. The ANG1(M) 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

The ANG1(M) must be correctly configured before one of the homing commands will be accepted. One of the following must be part of the module configuration before you can run the homing commands.

- 1) Configure one of the DC inputs as a Home Input
- 2) Configure the ANG1(M) to use an encoder and home to the encoder Z-pulse

### NOTE

- 1) You do not have to configure and use CW or CCW Limits. If you choose to configure the module this way, then the ANG1(M) has no way to automatically prevent overtravel during a homing operation. You must prevent overtravel by some external means, or ensure that the homing command is issued in the direction that will result in reaching the homing input directly.
- 2) When using one of the DC inputs as a Home Input, you can use a bit in the network data as a home proximity input. The use of this bit is completely optional.
- 3) When using an encoder's Z-pulse as the homing sensor, any DC input you configure as a Home Input will function as a hardware home proximity sensor. The use of this feature 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.

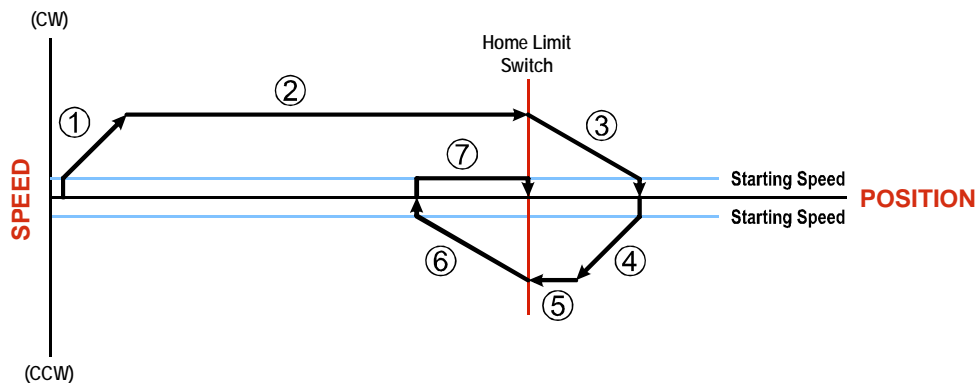


Figure 3.1 Home Input Profile

- 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 Proximity Input

Figure 3.2 below shows the move profile generated by a +Find Home command when you use:

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

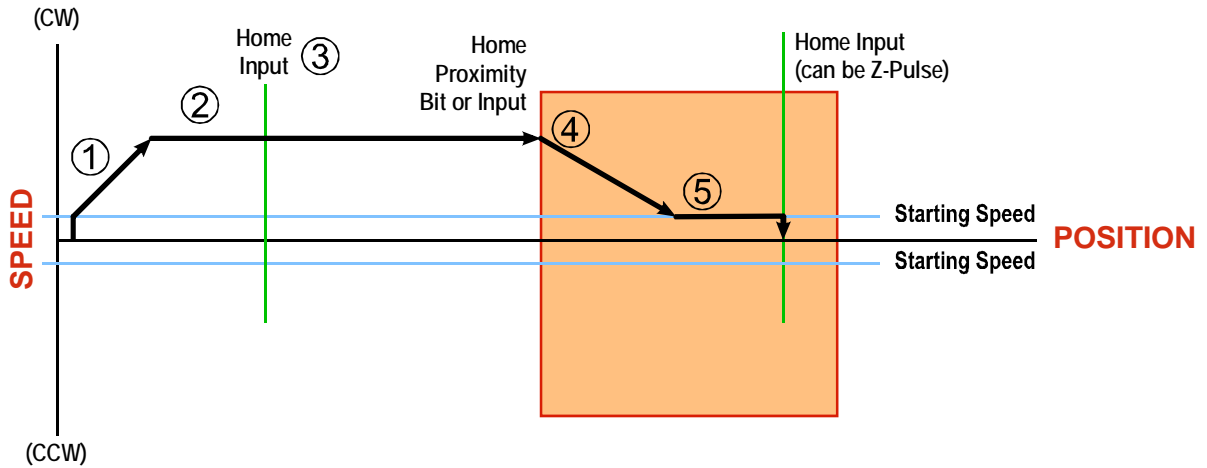


Figure 3.2 Homing with Proximity

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

**NOTE** Figure 3.2 shows the Backplane Home Proximity Bit staying active until the ANG1(M) reaches its home position. This is valid, but does not have to occur. As stated in step 4, the ANG1(M) 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 axis will stop all motion and issue a *Home Invalid* error to your host.

The ANG1(M) will stop the axis with an error if both overtravel limits are activated while the unit is trying to find the home position.

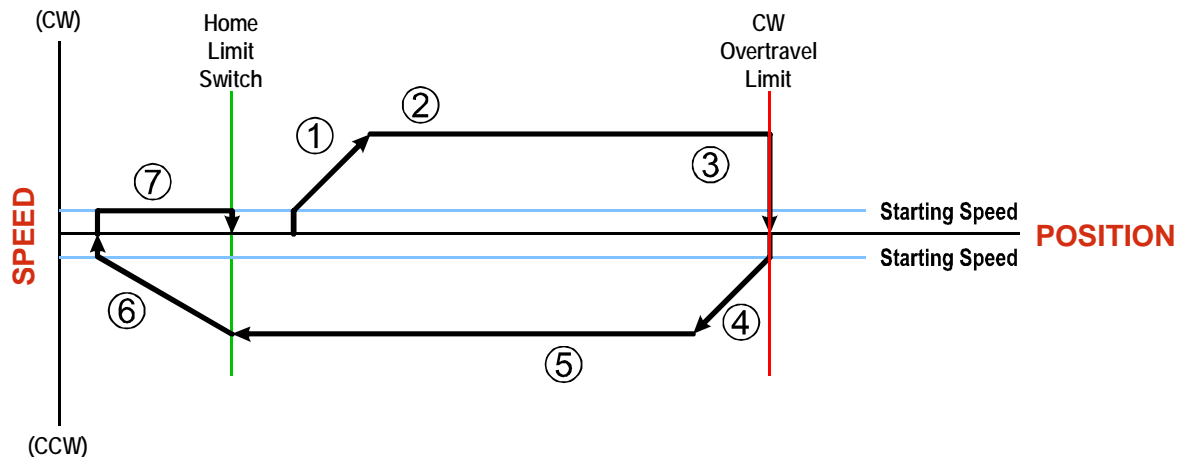


Figure 3.3 Profile with Overtravel Limit

- 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.

## INSTALLING THE ANG1(M)

### Location

AnyNET-I/O modules, including the ANG1 and ANG1M, are suitable for use in an industrial environment that meet the following criteria:

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

Note that these criteria apply to the system as a whole, not just to the ANG1 and ANG1M.

These criteria are equivalent to the *Pollution Degree 2* and *Over Voltage Category II* designations of the International Electrotechnical Commission (IEC).

### Safe Handling Guidelines

#### Prevent Electrostatic Damage



Electrostatic discharge can damage the ANG1 or ANG1M if you touch the rear bus connector pins. Follow these guidelines when handling the module.

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

#### Prevent Debris From Entering the Module



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

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



The AnyNET-I/O InterConnect bus is hot-swappable, but remove power before removing or installing any modules in a hazardous environment.

## Mounting

### Minimum Spacing

As shown in figure 4.1, you must maintain a minimum spacing of 2 inches (50.8 millimeters) from enclosure walls, wireways, adjacent equipment, etc. for adequate system ventilation.

Also note that the AnyNET-I/O modules must be mounted in the orientation shown in the figure. Mounting the system in any other orientation will decrease the efficiency of the ventilation slots on the top and bottom of each module which may lead to system overheating and malfunction.

When you install a stack of ANG1(M) modules that are all running at the full 4.0A current with a duty cycle near 100%, it is possible for the modules to overheat. It is difficult to calculate when this can occur because it is based on not only the current and duty cycle of the motor, but also such variables as enclosure size and ambient temperature. If overheating does occur, you have two choices. You can install a cooling fan beneath the stack to force additional air up through the modules or you can install an addition IC-5 connector between each module to space them out. Information on installing the IC-5 connectors can be found in the [Mounting](#) section of this manual starting on page 45.

The ANG1(M) has a bit in the network data that signals when the module is close to overheating. This bit is the Temperature Warning bit, and is available in Status Word 2 of the network input data while in Command Mode. This bit is further explained in the [Status Word 2 Format](#) section starting on page 82

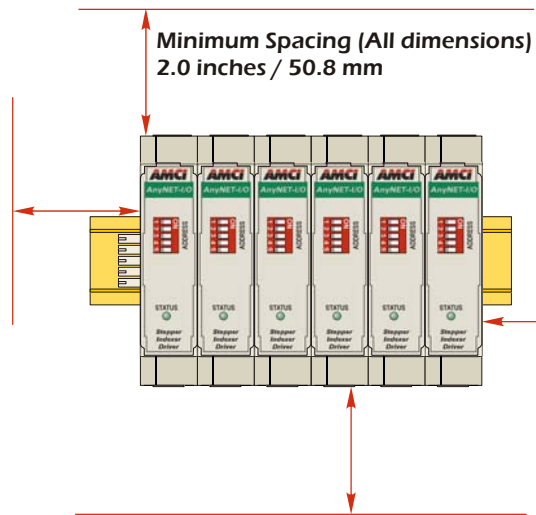


Figure 4.1 Ventilation Spacing

### DIN Rail Installation

AnyNET-I/O modules can be mounted on following DIN Rail

- EN 05 022 - 35 x 7.5 (35 x 7.5 mm)

If you are only installing one ANG1M module instead of an inter-connected stack, then you can also use the following DIN rail.

- EN 05 022 - 35 x 15 (35 x 15 mm)

**NOTE** The EN 05 022 - 35 x 15 DIN rail is unacceptable when installing multiple AnyNET-I/O modules as a stack. The IC-5 connector that is used to interconnect the modules requires the EN 05 022 - 35 x 7.5 DIN rail for proper operation.

The DIN Rail must be securely mounted to a panel and solidly grounded before the modules are installed. Grounding is usually accomplished through the mounting hardware, by first removing any paint or other material from all surfaces that may interfere with proper grounding. Another option is to install a heavy gauge wire from the DIN rail to your system's ground bus.

Figure 4.2 shows the amount of clearance space available for either a bolt head or a washer and nut when installing the 7.5mm deep DIN rail. The left side of the figure shows the clearance when installing a single ANG1M and the right side of the figure shows the clearance when using the IC-5 connector to interconnect multiple modules in an AnyNET-I/O Stack. When installing a single ANG1M module on 15mm deep rail, the clearance is 0.70 inches.

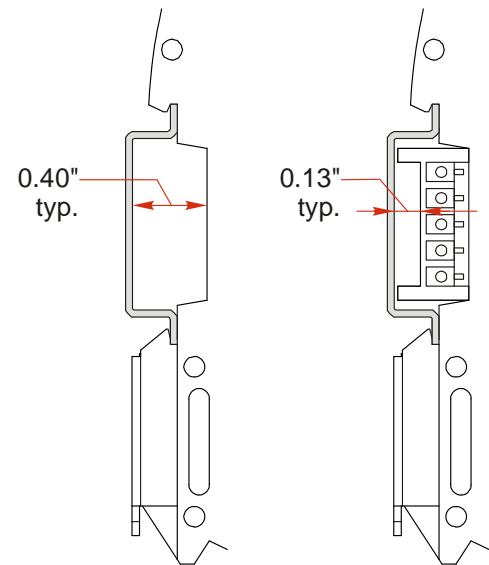


Figure 4.2 7.5 mm DIN Rail Installation

# INSTALLING THE ANG1(M)

## Mounting (continued)

### Installing IC-5 Connectors (as needed)

If you are installing a stack of modules instead of a single ANG1M, then you need to install the IC-5 connectors on the DIN rail to allow the modules in the stack to communicate. Figure 4.3 shows how to install the IC-5 connectors in the DIN rail.

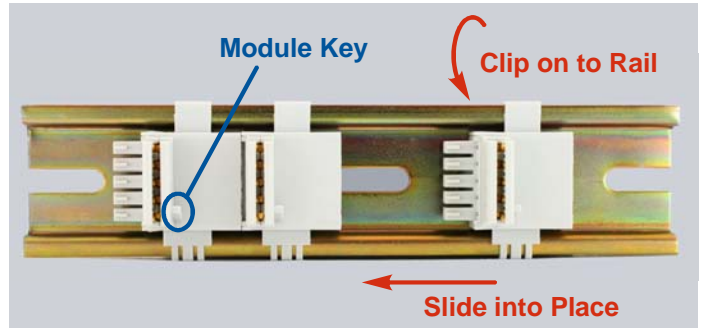


Figure 4.3 IC-5 Connector Installation

- NOTE**
- 1) EN 05 022 - 35 x 7.5 DIN rail must be used. The IC-5 connectors are not properly supported in EN 05 022 - 35 x 15 DIN Rail
  - 2) Note the orientation of the IC-5 connectors when installing them. The module key goes towards the bottom of the DIN rail.

### Mounting the ANG1(M) Module

Mounting an AnyNET-I/O module is a very simple process thanks to the design of the enclosure. If you are using the IC-5 connectors, then you will have to partially engage the connector in to the enclosure. Once that is done, engage the top clip in the enclosure with the top of the DIN rail and rotate the module down on to the DIN rail until the metal bracket snaps on to the DIN Rail.

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

## Addressing

### AnyNet-I/O Stack Addressing

Each module needs to be given an address before the system will operate correctly. The address is set with the five position DIP switch on the front of the module.

- NOTE**
- 1) Only a single switch should be in the "ON" position when setting the address.
  - 2) The module that has an address of zero must have a network interface and it is the only module in the stack that can have a direct connection to the network.
  - 3) If a module with a network interface has a non-zero address, then its network interface is disabled. This allows two ANG1M modules to work in a single stack.

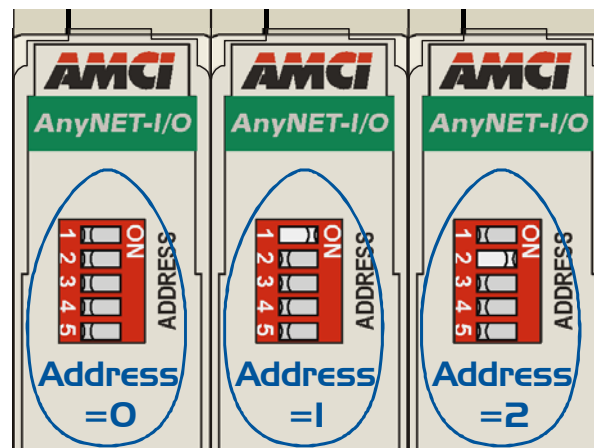


Figure 4.4 Addressing Example

## Addressing (continued)

### AnyNet-I/O Stack Addressing (continued)

Figure 4.4 on the previous page is a close up of three modules in an AnyNET-I/O Stack. The module on the left is an ANG1M, and has an address of zero (All DIP switches off.) This module has the active network interface and connects the stack to the network. Reading left to right, the remaining modules have addresses of one and two respectively. These modules can be either ANG1 or ANG1M modules. If they are ANG1M's, their network interfaces are disabled.

### Modbus Network Addressing

The address of the ANG1M unit in the Modbus network is set with two rotary switches that are behind the front panel door. The address can be set to any value between 1 and 99. The top switch sets the first digit of the address, and the bottom switch sets the second digit. As shown in figure 4.5, the rotary knob has a slot for a small flat head screwdriver and part of the shaft is cut away to indicate the number selected by the switch. The figure shows a Modbus network address of 08. Note that every device on the Modbus network must have a different address.

The figure also shows the AnyNet-I/O module address of zero. This is the proper setting for an ANG1M module that acts as the stack's network interface.

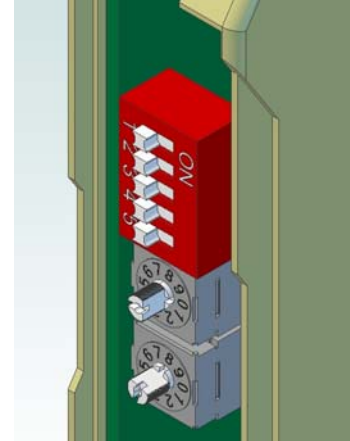


Figure 4.5 Modbus Addressing

### Modbus Network Connector

The Modbus network connector is located on the bottom of the ANG1M. The mating connector is sold separately by AMCI under the part number MS-2X4. It is also available from Phoenix Contact under their part number 173 88 27. The connector is laid out so that you can easily wire the ANG1M into the trunk cable. This is called a Trunk Interface (ITr) in the Modbus over Serial Line specification and is the preferred connection for the ANG1M. The three required interface lines are each brought out on two pins which makes it easy to daisy-chain the ANG1M in the trunk, or add the required Line Termination resistors if the ANG1E is the last device on the trunk.

A +5 Vdc, 50 mA supply is also available on the connector. This supply is made available as a convenient location to place Line Polarization resistors if your system requires them.

**NOTE** The ANG1M does not require Line Polarization resistors for proper operation.

**CAUTION** The +5 Vdc, 50 mA supply is for Line Polarization resistors only. Do not power digital inputs from this supply. Drawing excessive current from this supply may force the ANG1M into a reset state or damage it.

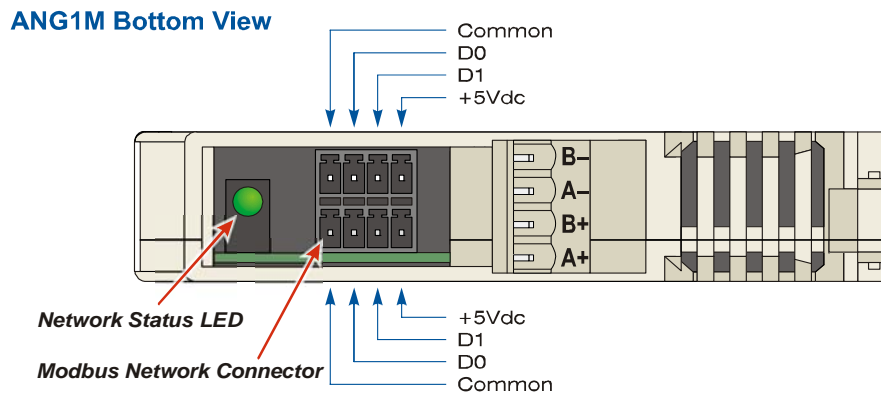


Figure 4.6 Modbus Connector Location

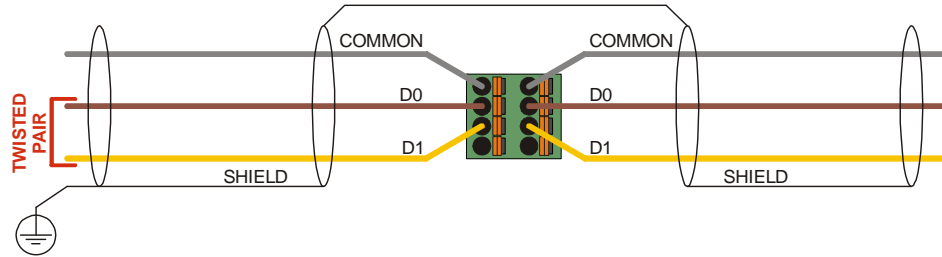
# INSTALLING THE ANG1(M)

## Modbus Network Wiring

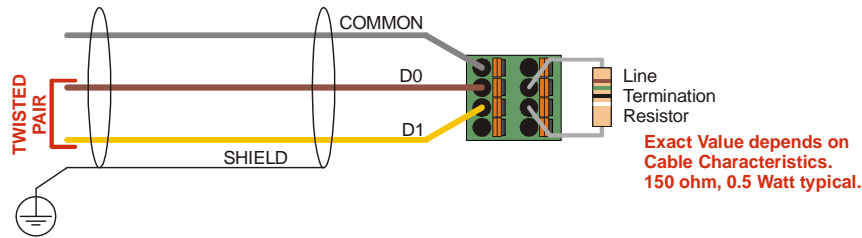
The ANG1M is designed to be wired as a Trunk Interface (ITr) device on a 2-wire serial bus. Figure 4.7 shows how to wire the ANG1M to the Trunk Cable as well as how to install the Line Termination Resistor if the ANG1M is the last device on the cable.

The final part of the figure shows how to wire the ANG1M to a Modbus serial line using a single twisted pair cable. This practice does not follow the Modbus over serial line specification, but it is commonly seen in practice. AMCI strongly recommends a three wire cable for a trunk cable.

### Cabling using 1.5 pair cable with Modbus recommended colors



### Line Termination Resistor



### Common Practice of using single twisted pair and shield as Common connection

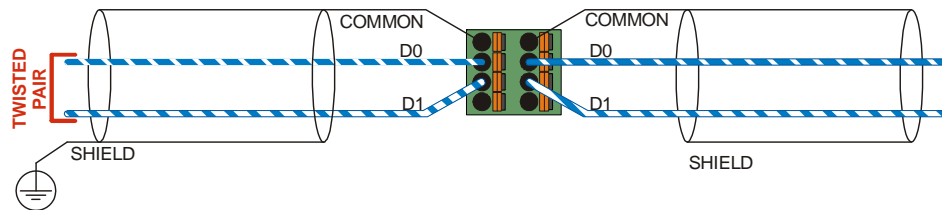


Figure 4.7 Wiring of Modbus over Serial Line

## 4-Wire Bus

The Modbus over serial line specification allows a four wire interface for the trunk cable. The ANG1M does not implement a 4-wire interface. Refer to the latest version of the *MODBUS over Serial Line Specification and Implementation Guide* found on the [www.modbus.org](http://www.modbus.org) website for information on connecting a 2-wire bus device to a 4-wire bus.

### Network Status LED

The Network Status LED shown in figure 4.6 indicate the health of the serial connection between the ANG1M and its host.

- **Steady Green** - The ANG1M is powered, but there is no active communication between the unit and its host.
- **Blinking Green** - The ANG1M is successfully communicating with its host over the Modbus Serial Line network
- **Blinking Red** - The ANG1M received a malformed packet over the Modbus Serial Line network. This is typically caused by noise on the RS-485 line, an improperly terminated RS-485 trunk cable, or problems with one of the devices on the trunk.

The Status LED on the front of the unit indicates the state of the ANG1M itself.

### I/O Connector Pin Out

The I/O Connector is located on the top of the module. The mate for this connector is not included with the ANG1(M). It is available from AMCI under the part number MS-2X11 and is also available from Phoenix Contact under their part number 173 88 98. Figure 4.8 shows the pin out for the I/O connector.

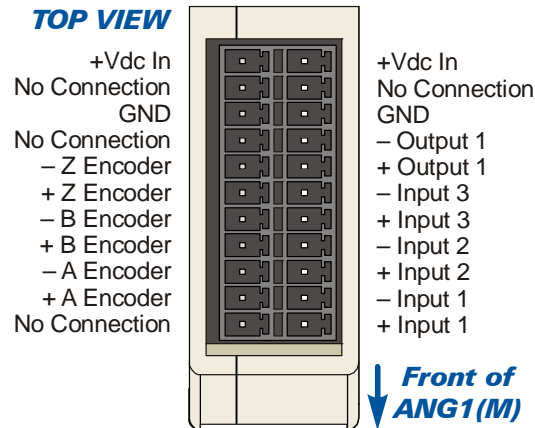


Figure 4.8 I/O Connector

### Power Wiring

The ANG1(M) accepts 24 to 48Vdc as its input power. Each module has two connections on the MS-2X11 I/O connector for the power supply that are internally connected together. This is typically used to power the digital I/O, but it is possible to daisy chain the power supply connections from one module to the next to simplify wiring if you do so with caution.

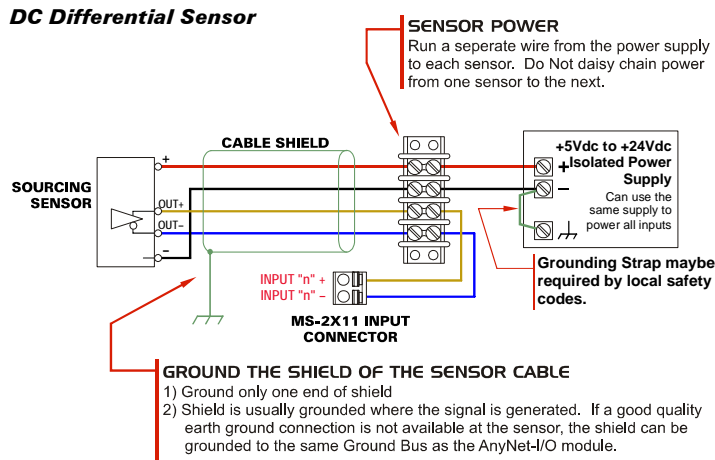
#### **CAUTION**

The I/O connector is rated for a maximum current of 8 amps per pin. If you daisy chain the power supply connections of the modules note that the current for *all* of the modules will flow through the first one. With a maximum current of 4 amps per module, you can easily exceed the current carrying capacity of the MS-2X11 if you daisy chain connections. If this possibility exists in your installation, run separate wires from your power supply to each module.

# INSTALLING THE ANG1(M)

## Input Wiring

Figure 4.9 below shows how to wire discrete DC differential, sourcing, and sinking sensors to Inputs 1, 2, and 3 of the ANG1(M).



### Input Specifications:

Differential. 560 Vac/dc opto-isolated. Will withstand 6000 Vac/dc for 60 seconds. Can be wired as single ended inputs. Accepts 3.5 to 27Vdc without the need for an external current limiting resistor.

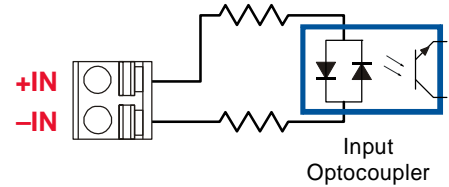


Figure 4.10 Simplified Input Schematic

Because they are low power signals, cabling from the sensor to the ANG1(M) 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 ANG1(M).

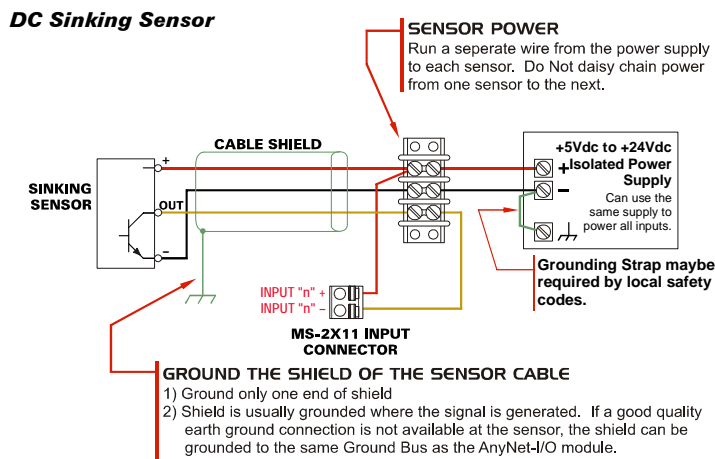
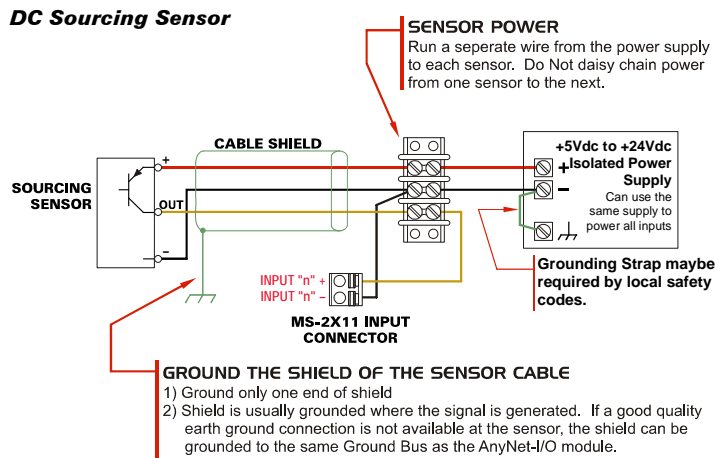
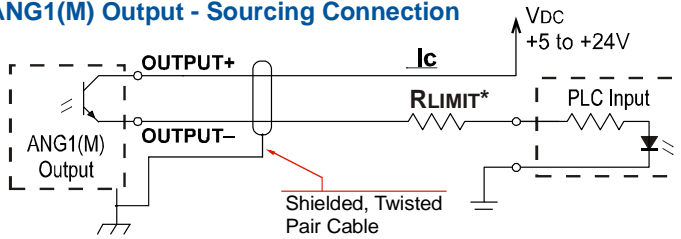


Figure 4.9 Input Wiring

## Output Wiring

### ANG1(M) Output - Sourcing Connection

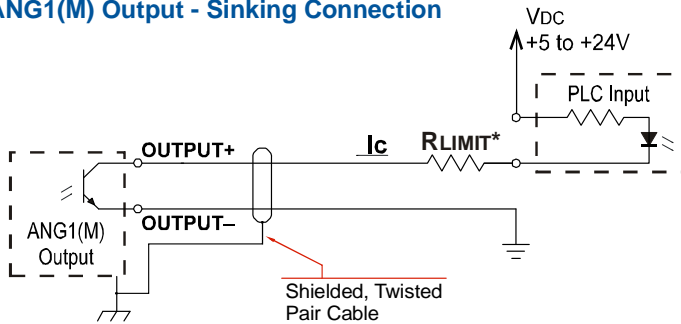


The ANG1(M) output is an optically isolated transistor that is capable of driving a typical PLC input. Both ends are uncommitted, so it can be wired as a sinking or sourcing output.

### Electrical Specifications

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

### ANG1(M) Output - Sinking Connection



### RLIMIT

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

Figure 4.11 Output Wiring

## Encoder Wiring

### Differential Wiring

The figure below shows how to wire a 5Vdc differential encoder to the ANG1(M). There is no standard when it comes to the color code of the encoder's wires. A document named 'encoder specs', is available on the AMCI website ([www.amci.com](http://www.amci.com)) that lists the color codes of encoders used by AMCI. It is available in the 'PDF Documents' section of the website.

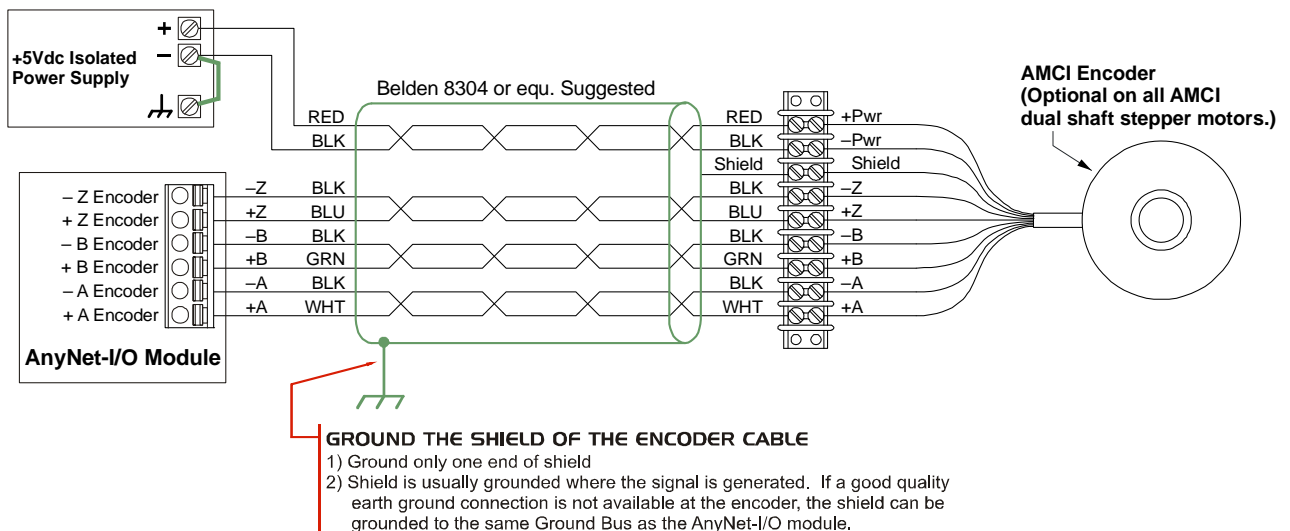


Figure 4.12 Sample Differential Encoder Wiring

# INSTALLING THE ANG1(M)

## Encoder Wiring (continued)

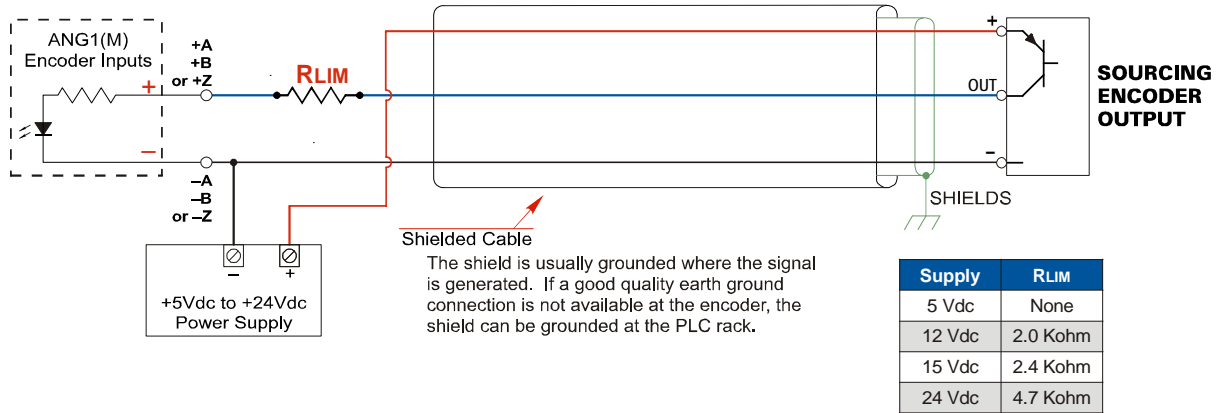
### Single Ended Wiring

Figure 4.13 below shows how to wire the encoder inputs to both a single ended sourcing and single ended sinking encoder outputs.



The encoder inputs on the ANG1(M) are rated for 5Vdc only. You must use a current limiting resistor on each input if the outputs of your encoder are greater than 5Vdc. Appropriate current limiting resistors are shown in the figure below.

#### ANG1(M) Encoder Input Connection to Sourcing Encoder Output



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

#### ANG1(M) Encoder Input Connection to Sinking Encoder Output

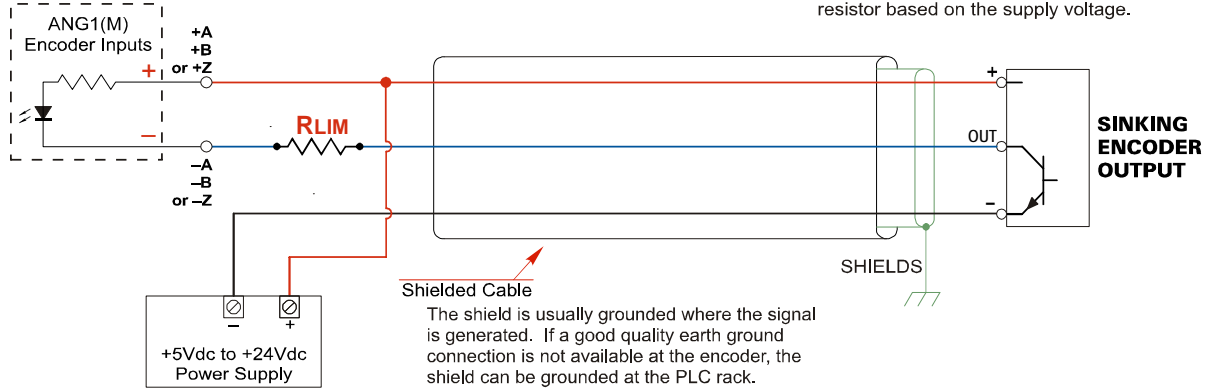


Figure 4.13 Single Ended Encoder Wiring

## Installing the Stepper Motor

### Outline Drawings

Outline drawings for all of our motors can be found on our website, [www.amci.com](http://www.amci.com), in the *PDF Documents* section. They're available as Adobe Acrobat pdf files. A document that is simply called *wiring* lists all of the wiring color codes for all AMCI motors. If you do not have internet access contact AMCI and we will fax the information to you.

### Mounting the Motor

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

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

**NOTE**

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

### Connecting the Load

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

### Extending the Motor Cable

Even though it is possible to extend the cable length an additional forty feet, AMCI recommends installing the ANG1(M) as close to the motor as possible. This will decrease the chances of forming a ground loop, and has the added benefit of limiting the amount of power loss in the motor cable. If you must extend the cable, you should use a cable with twisted pairs 18 AWG or larger and an overall shield. The exact gauge that you use will depend on the length of the run and expected temperature rise in your application. Belden 9552 is a suggested 18 AWG cable.

### Installing the Motor Cable

**NOTE**

- 1) All of the motor connections are high power, high voltage signals. Cable from the motor can be installed in conduit along with ac/dc power lines or high power ac/dc I/O. It cannot be installed in conduit with low power cabling such as I/O cabling or network cabling attached to the ANG1(M).
- 2) If you decide to extend the motor cable, treat the shield as a signal carrying conductor when installing the motor cable. Do not connect the shield to earth ground at any junction box.

# INSTALLING THE ANG1(M)

## Connecting the Motor

### Motor Connector

#### ANG1M Bottom View (Partial)

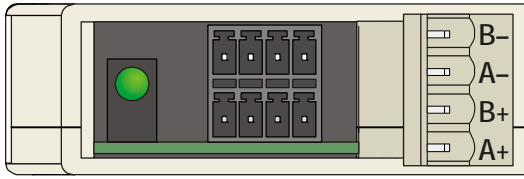


Figure 4.14 Motor Connector

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

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

**WARNING** Always remove power from the ANG1(M) before connecting or disconnecting the motor.

**NOTE**

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

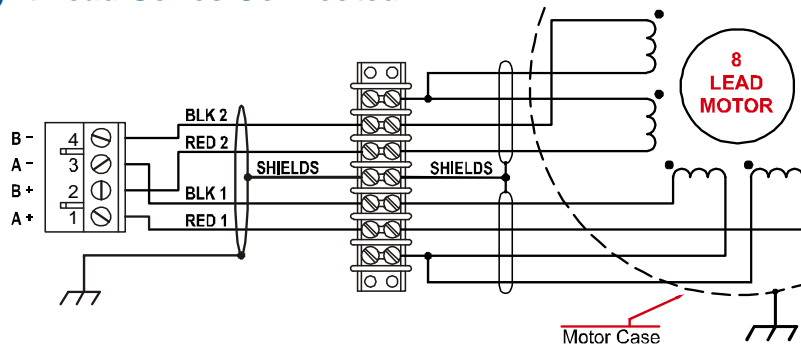
### Motor Wiring

The ANG1(M) will work with many different motors, including those not sold by AMCI. This section assumes that you have already chosen your motor and you are looking for wiring information. No wire colors are given in the figures below because there is no single industry wide color coding scheme for stepper motors. You must refer to your motor data sheets for this information.

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

Figure 4.15, which is continued on the following page, shows how to wire a motor to the ANG1(M) in series, parallel, or center-tap configurations. Refer to the torque vs. speed curves on your motor's specifications sheet to determine how you should wire the motor to the ANG1(M). Please note that you can reverse the direction of motor rotation by reversing the connections of the ±B pins on the ANG1(M).

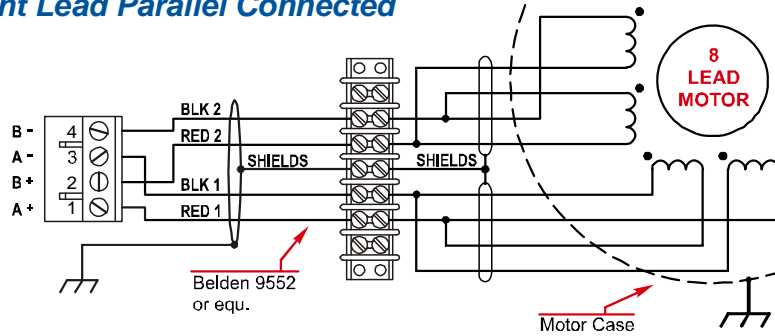
#### Eight Lead Series Connected



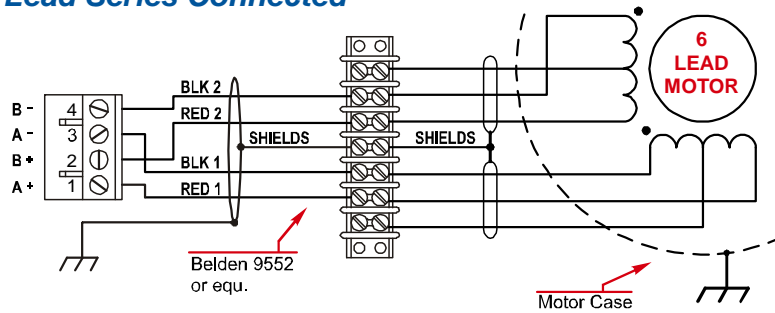
**Connecting the Motor (continued)**

**Motor Wiring (continued)**

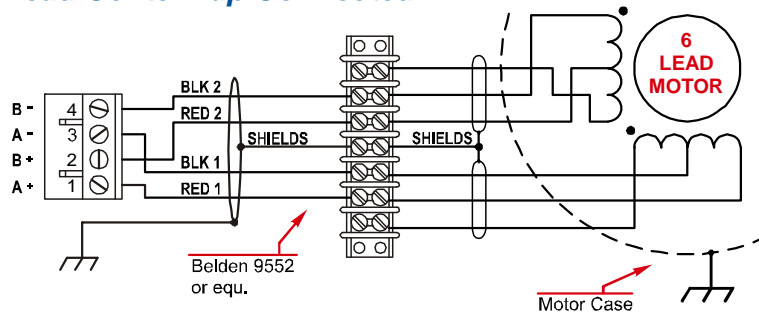
**Eight Lead Parallel Connected**



**Six Lead Series Connected**



**Six Lead Center Tap Connected**



**Four Lead Connected**

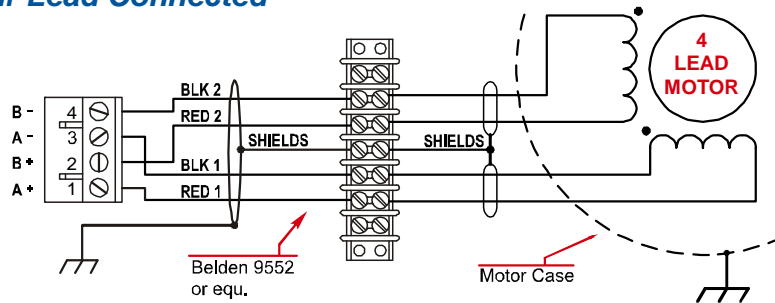


Figure 4.15 Motor Wiring

# CHAPTER 5

## NETWORK CONFIGURATION

This chapter covers how to configure the baud rate and parity of the Modbus serial line interface using the AMCI NET software. With this software, it is also possible to configure the modules in the AnyNet-I/O stack. The chapter only gives a brief introduction to the AMCI NET software. Check our website for additional information on the software that may be available.

### USB Connection

The ANG1M has a micro USB connector behind the front panel door. When connected to a computer, this USB connection acts as a serial port and allows you to connect to every module in your AnyNet-I/O stack through the AMCI NET software. You may require a windows driver when you use the USB connection for the first time. The software and driver are both available as free downloads from the AMCI website using the following address: <http://www.amci.com/product-software.asp>. Once on the page, click on the appropriate links to download the software.

### Installing the AMCI NET Software

Because many corporate firewalls prevent their users from downloading executable files, the install program exists as a ZIP file from our website. Installation is straight forward.

- 1) Download the software from <http://www.amci.com/product-software.asp>.
- 2) Extract the ZIP file to your computer. The resulting file is a simple install file.
- 3) Run the install program. When you do this, the following changes are made to your computer:
  - An AMCI folder is created in your programs folder. (Typically \Program Files\AMCI.)
  - The AMCI NET program and the USB driver files are placed in this directory
  - A shortcut to the AMCI NET program is created on the desktop and a program listing is created that will display the program off of the Start Menu.



No changes are made to the registry. The AMCI NET program can be easily un-installed by deleting the \Program Files\AMCI directory and shortcuts.

### USB Driver Installation

After the AMCI NET program is installed, apply power to the ANG1M module and plug in a USB cable between the ANG1M and you computer.

At this point, your system may notify you that an unknown device has been plugged into the computer and the system will start to look for the required driver. If the “Found New Hardware Wizard” opens on your XP system, and it asks “Can Windows connect to Windows Update to search for software?” you can click on the radio button that selects “Yes, this time only” and click on the [Next >] button. Your system should be able to locate the required driver.

Alternately, you can click on the “No, not at this time” radio button and click the [Next >] button. On the next screen, select the “Install from a list or specific location” radio button and click the [Next >] button. Now select the “Search for the best driver in these locations” radio button, un-check the “Search removable media” check box, check the “Include this location in the search” check box, and click on the [Browse] button. Browse to the \Program Files\AMCI folder and click the [OK] button. Click the [Next >] button, and your system should install the required driver.

The process may repeat to install a USB serial port.

The ANG1M uses a USB to serial conversion IC that is popular in embedded devices. It is possible that the device driver is already installed on your system. If this is the case, your system will automatically connect to the ANG1M.

## Using the AMCI NET Software

To start the program, click on the desktop icon or navigate from the start button to the program listing and click on the icon there. The program will present a screen similar to figure 5.1 below.

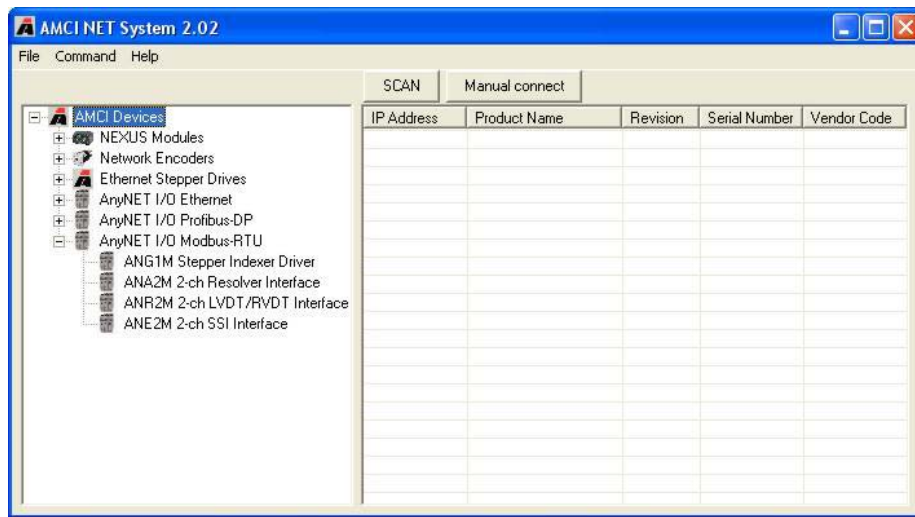


Figure 5.1 AMCI NET System software

## Connect to your ANG1E

The first step in configuring your ANG1M is to connect to it. Scroll the left pane of the window until you can see the ANG1M Stepper Indexer Driver listing. Double click on the listing to select the device.

A “Device Properties” window will open that allows you to connect to the ANG1M. Click on the [Connect] button.

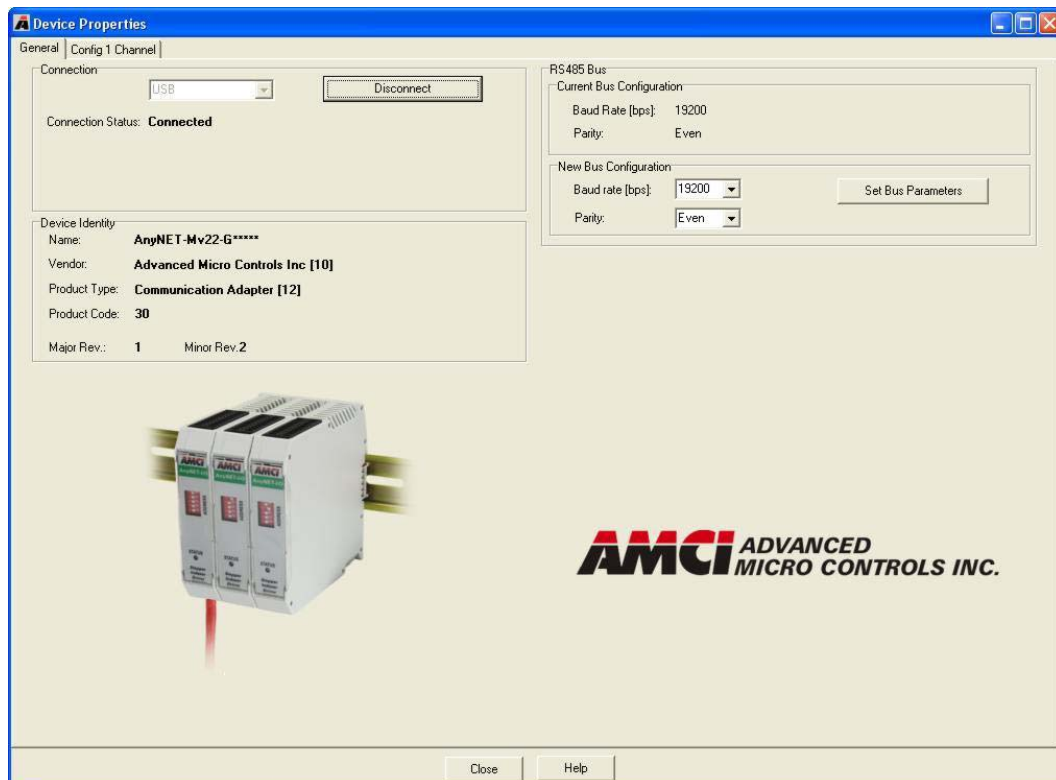


Figure 5.2 ANG1M Device Properties

## Using the AMCI NET Software (continued)

### The Meaning of the Product Name

Once you successfully connect to the ANG1M, the Product Name will show up in the General tab of the Device Properties windows. The Product Name gives you clues to the configuration of the ANG1M as well as other modules in the AnyNET-I/O Stack. As shown in figure 5.3 below, the Product Name is broken down into four parts.

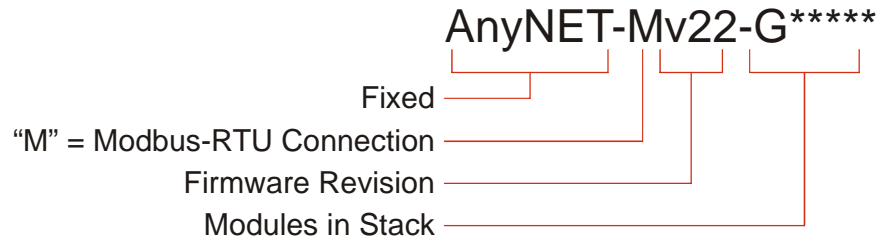


Figure 5.3 Product Name Description

The AnyNET-I/O system allows you to stack up to six AnyNET-I/O modules together and attach them to a single network connection. The last section of the product name lists all of the modules in the stack as individual letters. A "G" is the code for the ANG1 modules. A different letter is used for each module in the AnyNET-I/O system. Refer to the user manuals of other AnyNET-I/O modules to determine their identification letters. An "\*" is used as a place holder when no module is present. The modules are ordered by their address settings, not their physical location in the stack. For simplicity's sake, AMCI strongly suggests using consecutive address settings and place them in numeric order in their physical stack.

### Setting Modbus Serial Line Parameters

The General tab also shows you the current Modbus serial line baud rate and parity and allows you to change these settings. Simply click on the drop down select boxes to select the proper baud rate and parity for your application and the press the [Set Bus Parameters] button. When the save is complete in the ANG1M, a window will open that tells you that you must cycle power to the ANG1M before the parameter changes are accepted.

## Using the AMCI NET Software (continued)

### Configuring an ANG1(M) module (optional)

The remaining tab(s) in the Device Parameters window allow you to configure the modules in the AnyNet-I/O stack. In the case of ANG1(M) modules, you can configure the module and execute commands. These features are typically used for bench testing because they can help you determine the correct configuration and command data to use in your Modbus program. Figure 5.4 below shows the configuration and command window for an ANG1(M). The data layout follows the format of the configuration and command data written to the ANG1(M) using the Modbus registers. The format of these registers are explained in the remaining chapters of this manual.

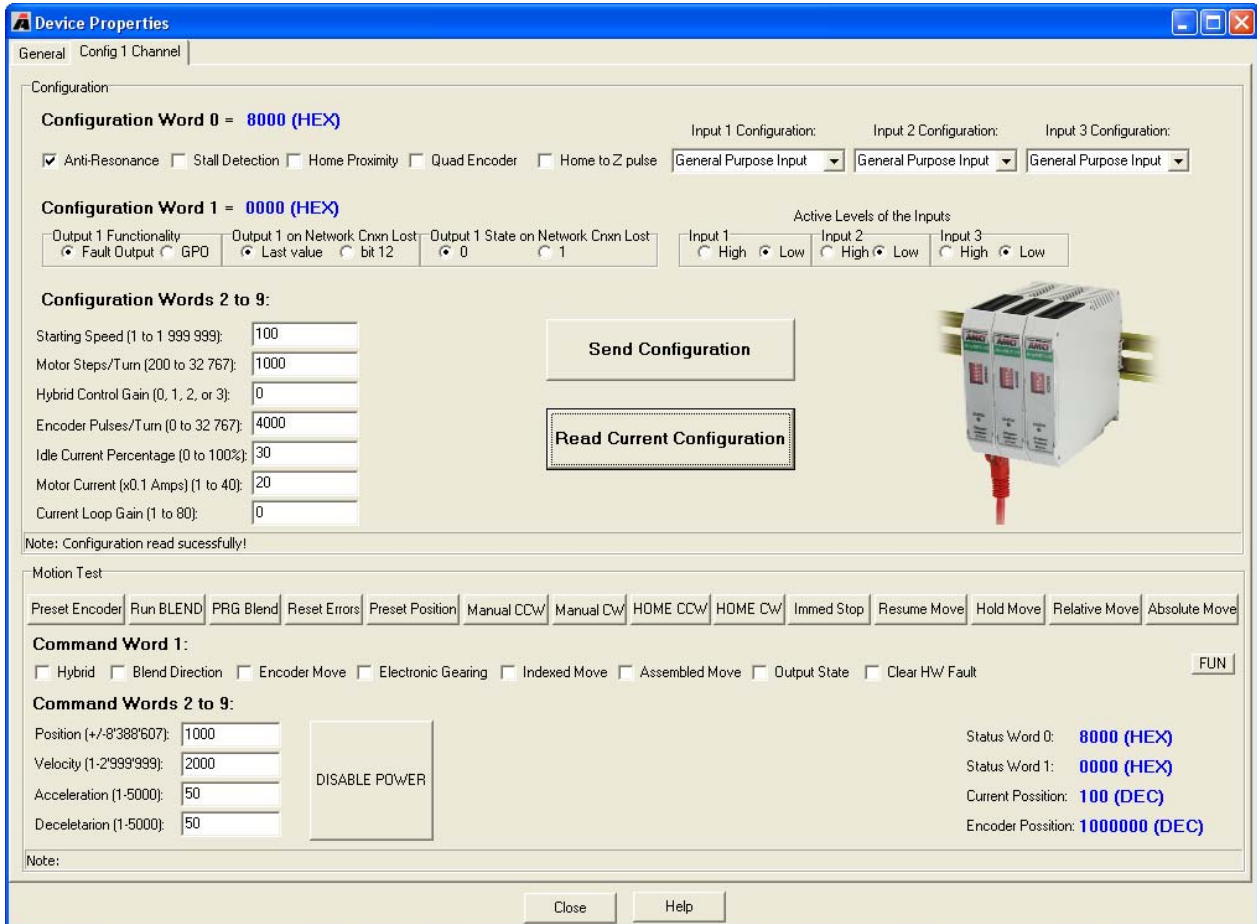


Figure 5.4 Configuration and Command Screen

## Modbus INTERFACE

An AnyNET-I/O Stack that includes the ANG1 or ANG1M requires a host controller to issue motion commands to these modules. This chapter tell you how the I/O words used by an AnyNET-I/O Stack are mapped to the Modbus I/O registers.

### **AnyNET-I/O Stack Memory Layout**

The first module in every AnyNET-I/O Stack has a starting Input Register address of 0 and a starting Output Register address of 1024. The next module in the AnyNET-I/O stack is addressed immediately after the previous module. Input Registers hold the data from the AnyNET-I/O Stack while Output Registers hold the data to be written to the stack. Figure 6.1 shows how an AnyNET-I/O Stack, which consists of an ANG1M and an ANG1, is mapped to the Modbus data reference. The complete specification for the Modbus protocol can be downloaded at <http://www.modbus.org/specs.php>.

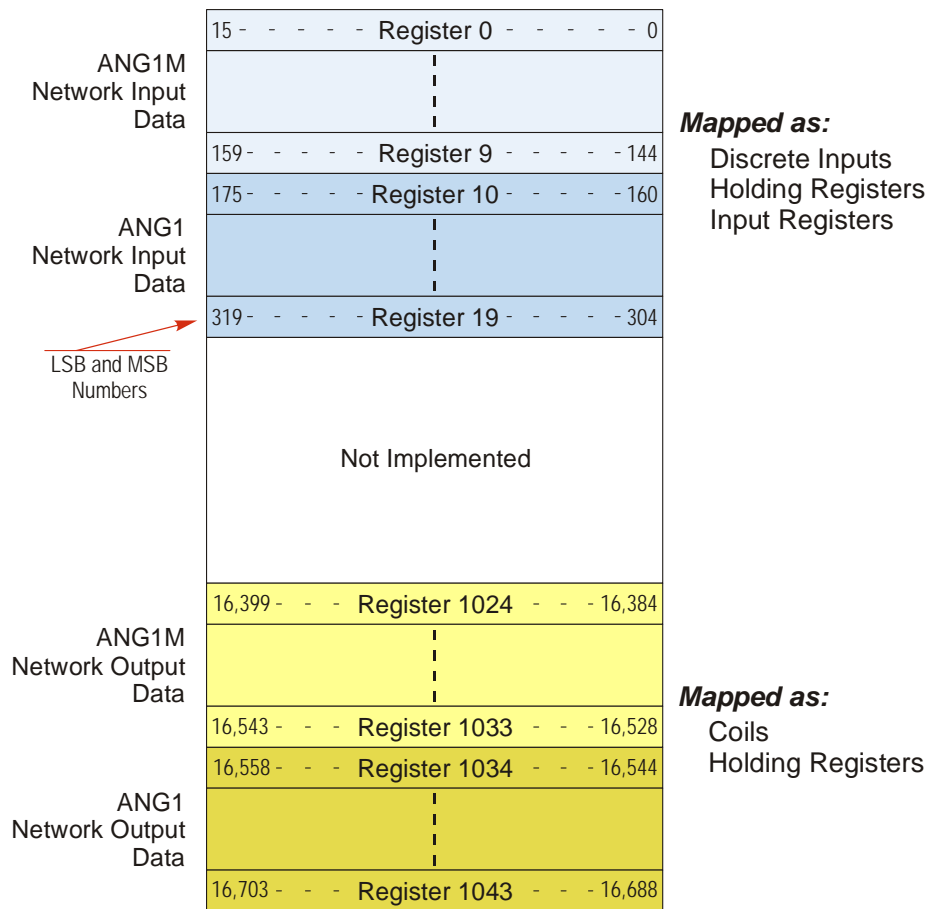


Figure 6.1 Modbus Data Reference Map: Stack of ANG1M and ANG1

### **AnyNET-I/O Stack Addressing**

There are two ways that you can address modules in your AnyNET-I/O Stack when using Modbus-RTU. You can address the stack as a whole, or you can address modules individually by selecting the correct starting address and number of registers for the Modbus-RTU transfer. If you decide to address the stack as a whole, the number of words needed is simply the sum of all the I/O words used by all AnyNET-I/O modules in the stack. Each ANG1M and ANG1 module requires ten input and ten output words. Refer to the manuals of any other modules in the AnyNET-I/O Stack to determine the number of I/O words used by those modules.

**Supported Modbus Functions**

Function Code	Function Name	ANG1(M) Register	Addressing method
1	Read Coils	OUTPUT	Bit: Addresses starting at 16,384
2	Read Discrete Inputs	INPUT	Bit: Addresses starting at 0
3	Read Holding Registers	OUTPUT & INPUT	Word: Out Regs. Starting at 1024 In Regs. Starting at 0
4	Read Input Registers	INPUT	Word: Addresses starting at 0.
5	Write Single Coil	OUTPUT	Bit: Addresses starting at 16,384
6	Write Single Register	OUTPUT	Word: Addresses starting at 1024
15	Write Multiple Coils	OUTPUT	Bit: Addresses starting at 16,384
16	Write Multiple Registers	OUTPUT	Word: Addresses starting at 1024

Table 6.1 Supported Modbus Functions

Table 6.1 above lists the Modbus functions that are supported by AnyNET-I/O modules when an ANG1M is the stack's network connection. AMCI supports all of these functions so that you can control the stack as you see fit.

**Supported Modbus Exceptions**

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

Table 6.2 Supported Modbus Exceptions

## CONFIGURATION MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to configure the ANG1(M) as well as the formats of the Network Input Data that contains the responses from the module. Each ANG1(M) requires ten 16-bit words (20 bytes) for Output Data as well as ten 16-bit words for Input Data.

### **Modes of Operation**

The ANG1(M) has two operating modes, Configuration Mode and Command Mode. You switch between these modes by changing the state of a single bit in the Network Output Data.

#### **Configuration Mode**

Configuration Mode gives you the ability to select the proper configuration for your application without having to set any switches. The ANG1(M) must be configured on every power up. Additionally, the motor will not receive power until the module is configured properly.

#### **Command Mode**

This mode gives you the ability to program and execute stepper moves, and reset errors when they occur. The ANG1(M) will always power up in this mode, but a valid configuration must be sent to the ANG1(M) before it will apply power to the motor or allow you to issue move commands. The command data formats are described in the following chapter.

### **Multi-Word Format**

Many of the hosts that can be used with the ANG1(M) 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 ANG1(M) 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 ANG1(M) is in Configuration Mode is shown below. The Modbus addresses shown are for an ANG1M module that is the network connection for the AnyNET-I/O Stack. You will have to adjust the memory addresses if your ANG1(M) module is not the first module in the AnyNET-I/O stack.

Modbus Register	Configuration Data	Range
1024	Control Word	See below
1025	Configuration Word	See below
1026	Starting Speed: Upper Word	Combined value between 1 and 1,999,999 steps/sec.
1027	Starting Speed: Lower Word	
1028	Motor Steps/Turn	200 to 32,767
1029	Hybrid Control Gain	0, 1, 2, or 3
1030	Encoder Pulses/Turn	0 to 32,767
1031	Idle Current Percentage	0 to 100%
1032	Motor Current (X10)	10 to 40, Represents 1.0 to 4.0 Arms
1033	Current Loop Gain	1 to 40

Table 7.2 Network Output Data Format: Configuration Mode

## Control Word Format

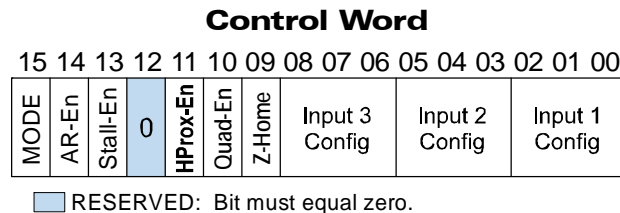


Figure 7.1 Configuration Mode: Control Word Format

- Bit 15: Mode Bit** – “1” for Configuration Mode Programming, “0” for Command Mode Programming. The ANG1(M) powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), unless a valid configuration has been written to the flash memory of the ANG1(M). The ANG1(M) will not power the motor or accept commands until a valid configuration is written to it.
- Bit 14: Anti-Resonance Enable Bit** – “0” enables the anti-resonance feature of the ANG1(M). “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 when an encoder is used and attached to the motor controlled by the ANG1(M). The encoder pulses/turn must also be programmed.
- Bit 11: Home Proximity Bit Enable Bit** – “0” when Home Proximity bit is not used when homing the ANG1(M). “1” when the Home Proximity bit is used when homing the ANG1(M). 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.



## Configuration Mode - Output Data Format (continued)

### Control Word Format (continued)

**Bit 9: Home to Encoder Z Pulse** – Set to “1” to home the machine to the encoder’s Z pulse. The Quadrature Encoder Enable Bit, bit 10, must be set. You must also program the Encoder Pulses/Turn parameter in Configuration word 6. If a Discrete DC Input is configured as a Home Input, it will act as a Home Proximity Input.

**Bits 8-6: Input 3 Configuration Bits** – See Table Below

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

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

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

Table 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	Out1Fn	Out1INCL	Out1State	RdConfig	SavConfig	0	0	0	0	0	0	0	ActSt-3	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.
- Bit 14: Output 1 Functionality Bit** – “0” configures Output 1 to be a Fault Output. The output will conduct current until a fault occurs. “1” configures Output 1 to be a general purpose output whose state is determined by a bit in the Command Mode Network Output Data. This output is in an ON state when power is applied to the ANG1(M) and it has not yet been configured.
- Bit 13: Output 1 on Network Connection Lost Bit** – “0” will keep Output 1 at its last value. “1” will set the state of Output 1 to the value specified in Bit 12 of this word.
- Bit 12: Output 1 State on Network Connection Lost Bit** – When bit 13 of this word is set, Output 1 will be set to the state of this bit if the network connection is lost.
- Bit 11: Read Present Configuration** – If this bit is set when you enter Configuration Mode, the ANG1(M) responds by placing the present configuration data in the Network Input Data. You cannot write new configuration data to the module while this bit is set. The format of the Configuration Data is given in the *Input Data Format* section of this chapter, starting on page 65.
- Bit 10: Save Configuration** – The ANG1(M) will store the configuration data to flash memory when this bit makes a 0 → 1 transition. The validity of the configuration data is checked before being written to the flash memory. If the data is not correct, the transition on this bit is ignored. Once the write to flash is completed, the ANG1(M) will write 0xAAAA into the last status word if the write was successful and the Status LED will begin flashing green, or the ANG1(M) will write 0xEEEE into the last status word if the write was unsuccessful and the Status LED will begin flashing red. Once the ANG1(M) does this, it will stop responding to commands and you must cycle power to the module. This design decision is to prevent the flash memory from constant write commands. The flash memory has a minimum of 10,000 write cycles.
- Bits 9 - 3: Reserved** – Must equal zero.
- Bit 2: Input 3 Active Level Bit** – Determines the active state of Input 3. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.
- Bit 1: 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 61.
- Hybrid Control Gain controls the “stiffness” of the shaft at zero speed when using the Hybrid Control feature. With a higher gain setting, the ANG1(M) will increase the motor current more quickly to compensate for shaft loading.
- Changes to the Idle Current only take effect at the *end of the first move after re-configuration*.
- You should start with Current Loop Gain setting of 5 for all AMCI motors and can adjust the setting from there based on your system requirements. Refer to the *Current Loop Gain* section on page 37 for additional information.

### Input Data Format

The correct format for the Network Input Data when the ANG1(M) is in Configuration Mode is shown below. The Modbus addresses shown are for an ANG1M module that is the network connection for the AnyNET-I/O Stack. You will have to adjust the memory addresses if your ANG1(M) module is not the first module in the stack.

Modbus Register	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	Mirror of Hybrid Control Gain
6	Mirror of Encoder Pulses/Turn
7	Mirror of Idle Current Percentage
8	Mirror of Motor Current (X10)
9	Mirror of Current Loop Gain 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 ANG1(M). When the ANG1(M) is not configured, or the data written to it is invalid, then this word has the same format of Status Word 1 when the ANG1(M) is in Command Mode. This format is explained in the *Status Word 1 Format* section starting on page 80. 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 ANG1(M).



### ***Invalid Configurations***

The following configurations are invalid:

- 1) Setting any of the reserved bits in the configuration words.
- 2) Setting any parameter to a value outside of its valid range. This includes setting the Lower Word of the Starting Speed to a value greater than 999.
- 3) You configure two or more inputs to have the same function, such as two CW Limit Switches.
- 4) Setting the *Home to Encoder Z-Pulse Bit* without configuring the ANG1(M) to use the encoder.
- 5) Setting the *Stall Detection Enable Bit* without configuring the ANG1(M) to use the encoder.
- 6) Setting the Input Configuration bits for any input to “111”. See table 7.3 on page 63 for more information.

## COMMAND MODE DATA FORMAT

This chapter covers the formats of the Network Output Data used to command the ANG1(M) as well as the formats of the Network Input Data that contains the responses from the driver. The ANG1(M) 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 ANG1(M) 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 ANG1(M) 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 ANG1(M). The Modbus addresses shown are for an ANG1M module that is the network connection for the AnyNET-I/O Stack. You will have to adjust the memory addresses if your ANG1(M) module is not the first module in the stack.

Modbus Register	Function
1024	Command Bits MSW
1025	Command Bits LSW
1026	Command Parameters  <i>Word meaning depends on the command set to the ANG1(M)</i>
1027	
1028	
1029	
1030	
1031	
1032	
1033	

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	ResmMv	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 ANG1(M) powers up in Command Mode and shows a configuration error, (hexadecimal value of 6408h), if a valid configuration has not been stored in flash memory. The ANG1(M) will not power the motor or accept commands until a valid configuration is written to it or read from flash memory.
- Bit 14: Preset Encoder Bit** – When set to “1” the ANG1(M) will preset the Encoder Position to the value stored in Output Words 2 and 3.
- Bit 13: Run Assembled Move** – When set to “1” the ANG1(M) 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 32.
- Bit 10: Reset Errors** – When set to “1” the ANG1(M) 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 ANG1(M) 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** – When set to “1” the ANG1(M) will run a Jog Move in the counter-clockwise direction. The full explanation of a [±Jog Move](#) can be found starting on page 25.
- **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.
  - **Enable Encoder Follower Mode – Command Bits LSW, Bit 6:** When this bit equals “1” the ANG1(M) will switch its operation to [Encoder Follower](#) mode as described on page 35. While in this mode, the two Jog Move bits are used to enable motor motion. One of these two bits must equal “1” before the motor will follow a change in encoder position.
- Bit 7: + Jog Move** – When set to “1” the ANG1(M) will run a Jog Move in the clockwise direction. The full explanation of a [±Jog Move](#) can be found starting on page 25.
- **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.
  - **Enable Encoder Follower Mode – Command Bits LSW, Bit 6:** When this bit equals “1” the ANG1(M) will switch its operation to [Encoder Follower](#) mode as described on page 35. While in this mode, the two Jog Move bits are used to enable motor motion. One of these two bits must equal “1” before the motor will follow a change in encoder position.

# COMMAND MODE DATA FORMAT

## Command Bits MSW (continued)

- Bit 6: Find Home CCW** – When set to “1” the ANG1(M) 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 ANG1\(M\)](#) chapter starting on page 39.
- Bit 5: Find Home CW** – When set to “1” the ANG1(M) 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 ANG1\(M\)](#) chapter starting on page 39.
- Bit 4: Immediate Stop Bit** – When set to “1” the ANG1(M) 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 34. 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. 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 34.
- 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 23.
- 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 24.

## Command Bits LSW

Command Bits LSW

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DriveEn	0	GPOut1	0	BkHPx	ClrDrvFlt	AMType	IndxMv	RunReg	EGMode	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 ANG1(M) before the driver can be enabled.
- Bit 14: Reserved** – Must equal “0”.
- Bit 13: General Purpose Output State Bit** – When the output is configured as a general purpose output instead of the Fault Output, this bit controls the state of the output. When this bit equals a “1”, the output is on and conducts current.
- Bit 12: Reserved** – Must equal “0”.
- Bit 11: Backplane Home Proximity Bit** – When the ANG1(M) is configured to use the Backplane Home Proximity bit, the ANG1(M) will ignore the state of the Home Input as long as this bit equals “0”. This bit must equal “1” before a transition on the Home Input can be used to home the machine. These statements also apply when the encoder Z-pulse is used to home the machine. Please note that you cannot use the Backplane Home Proximity bit and a discrete input configured as a Home Proximity Input at the same time. You must choose one or the other when configuring your machine. Further information on using the Home Proximity bit can be found in the [Homing Inputs](#) section starting on page 39.

**Command Bits LSW (continued)**

- Bit 10: Clear Driver Fault** – If this bit is set when a Reset Errors Command is issued, (Command Bits MSW Bit 10) the ANG1(M) will attempt to clear driver errors such as an overtemperature or motor short fault. Note that the driver must be disabled (Command Bits LSW, Bit 15 = 0), when using this command.
- Bit 9: Assembled Move Type** – When this bit equals “0”, a Blend Move is started when the Run Assembled Move bit, (Command Word 1, Bit 13) makes a 0 → 1 transition. When this bit equals “1”, a Dwell Move is started on the transition. The direction of a Blend Move is controlled by the 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 ANG1(M) will not run the move immediately, but will instead wait for an inactive-to-active transition on an input configured as a *Start Indexer Move* input.
- Bit 7: 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: Enable Encoder Follower Mode** – Set to “1” to put the ANG1(M) in Encoder Follower mode. Set to “0” for normal operation. A full description of *Encoder Follower* mode starts on page 35.
- 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 ANG1(M) responds by flashing the Status LED when the writing is complete. If the LED is flashing green, the write to flash memory was successful. If it flashes red, then there was an error in writing the data. In either case, power must be cycled to the ANG1(M) before you can continue. This design decision is to prevent the flash memory from constant write commands. The flash memory has a minimum of 10,000 write cycles.
- Bit 4: 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: Hybrid Control Enable** – Set to “0” to operate the ANG1(M) as a standard stepper system. Set to “1” to enable Hybrid Control. *Hybrid Control* is explained starting on page 36.
- Bit 2: Encoder Move** – Set to “1” when you command an Absolute or Relative move to cause the move distance to be based on the encoder position, not on the motor position. Set to “0” for a standard Relative or Absolute move based on motor position. Please note that an encoder 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 28.
- 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. The Modbus addresses shown are for an ANGI1M module that is the network connection for the AnyNET-I/O Stack. You will have to adjust the memory addresses if your ANGI1(M) module is not the first module in the stack.

### Absolute Move

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0001
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Abs. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
1027	Abs. Target Position: Lower Word		
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.2 Absolute Move Command Block

### Relative Move

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0002
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Rel. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
1027	Rel. Target Position: Lower Word		
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.3 Relative Move Command Block

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

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0004
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused		See Note Below

Table 8.4 Hold Move Command Block

Unused words are ignored by the ANGI(M) and can be any value, including parameter values in the previous command.

**Resume Move**

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0008
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.5 Resume Move Command Block

Unused words are ignored by the ANGI(M) 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

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0010
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused		See Note Below

Table 8.6 Immediate Stop Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

### +Find Home (CW)

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0020
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.7 +Find Home (CW) Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

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

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0040
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.8 -Find Home (CCW) Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

**+Jog Move (CW)**

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0080
1025	<i>Command Bits LSW</i>		See pg. 69 Bits 7 & 6 must equal "00"
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.9 +Jog Move (CW) Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

## Command Blocks (continued)

### +Registration Move (CW)

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0080
1025	<i>Command Bits LSW</i>		See pg. 69 Bits 7 & 6 must equal "10"
1026	Stopping Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
1027	Stopping Distance: Lower Word		
1028	Programmed Speed: Upper Word	Steps per Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
1033	Min. Reg. Move Distance: Lower Word		

Table 8.10 +Registration Move (CW) Command Block

### -Jog Move (CCW)

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0100
1025	<i>Command Bits LSW</i>		See pg. 69 Bits 7 & 6 must equal "00"
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.11 -Jog Move (CCW) Command Block

Unused words are ignored by the ANGI(M) and can be any value, including parameter values in the previous command.

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

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0100
1025	<i>Command Bits LSW</i>		See pg. 69 Bits 7 & 6 must equal “10”
1026	Stopping Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
1027	Stopping Distance: Lower Word		
1028	Programmed Speed: Upper Word	Steps per Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Min. Reg. Move Distance: Upper Word	Steps	Combined value between 0 and +8,388,607
1033	Min. Reg. Move Distance: Lower Word		

Table 8.12 –Registration Move (CCW) Command Block

**Encoder Follower Move**

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0080 or 16#0100
1025	<i>Command Bits LSW</i>		See pg. 69 Bit 6 must equal “1”
1026	Electronic Gearing Numerator		1 to 255
1027	Electronic Gearing Denominator		1 to 255
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.13 Encoder Follower Move Command Block

## Command Blocks (continued)

### Preset Position

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0200
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Position Preset Value: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
1027	Position Preset Value: Lower Word		
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused		See Note Below

Table 8.14 Preset Position Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

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

### Reset Errors

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0400
1025	<i>Command Bits LSW</i>		See pg. 69 Set bit 10 to clear driver faults
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused		See Note Below

Table 8.15 Reset Errors Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

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

**Command Blocks (continued)****Run Assembled Move**

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#2000
1025	<i>Command Bits LSW</i>		See pg. 69 Blend Move: Bit 9 = "0" Dwell Move: Bit 9 = "1" Blend Move direction set by Bit 4.
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused with Blend Move Dwell Time with Dwell Move	milliseconds	0 to 65,535

Table 8.16 Run Assembled Move Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

**Preset Encoder Position**

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#4000
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Encoder Preset Value: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
1027	Encoder Preset Value: Lower Word		
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused		See Note Below

Table 8.17 Preset Encoder Position Command Block

Unused words are ignored by the ANG1(M) 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

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0800
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused		See Note Below

Table 8.18 Assembled Move First Programming Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values from the previous command.

Once the first block is transmitted, the ANG1(M) responds by setting bits 8 and 9 in Status Word 1. (See [Status Word 1 Format](#) starting on page 80.) Once these are set, you can then start transmitting Segment Blocks.

### Segment Block

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#1800
1025	<i>Command Bits LSW</i>		See pg. 69
1026	Rel. Target Position: Upper Word	Steps	Combined value between -8,388,607 and +8,388,607
1027	Rel. Target Position: Lower Word		
1028	Programmed Speed: Upper Word	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower Word		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	<i>Reserved</i>		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table 8.19 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 ANG1(M) sees bit 12 of Command Bits MSW set, it will accept the block and reset bit 9 in Status Word 1. When your program sees this bit reset, it must respond by resetting bit 12 of Command Bits MSW. The ANG1(M) will respond to this by setting bit 9 in Status Word 1 and the next Segment Block can be written to the ANG1(M). You can write a maximum of sixteen Segment Blocks for each Assembled Move.

### Input Data Format

The correct format for the Network Input Data when the ANG1(M) is in Command Mode is shown below. The Modbus addresses shown are for an ANG1M module that is the network connection for the AnyNET-I/O Stack. You will have to adjust the memory addresses if your ANG1(M) module is not the first module in the stack.

Modbus Register	Command Mode Input Data
0	Status Word 1
1	Status Word 2
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	Value of Acceleration Jerk Parameter

Table 8.20 Network Input Data Format: Command Mode

### Status Word 1 Format

Status Word 1															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MODE	CtrlOK	ConErr	CmdErr	InErr	PosInVld	Xmt_MS	PgmMov	MvCmp	Decel	Accel	Homed	Stopped	Hold	MvCCW	MvCW

Figure 8.4 Command Mode: Status Word 1 Format

- Bit 15: Mode Bit** – “1” for Configuration Mode Programming, “0” for Command Mode Programming. The ANG1(M) powers up in Command Mode and shows a configuration error unless a valid configuration was written to the module using the AMCI NET software or a command while in Configuration Mode. The ANG1(M) will not power the motor or accept commands until a valid configuration is written to it.
- Bit 14: Controller OK Bit** – “1” when the ANG1(M) 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 ANG1(M) or after any invalid configuration has been written to it. “0” when the ANG1(M) 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 ANG1(M). 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 77.

## ***Input Data Format (continued)***

### **Status Word 1 Format (continued)**

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

- A configuration is written to the ANG1(M)
- The motor position has not been preset and the machine has not been homed
- 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 ANG1(M) 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 32.

**Bit 8: Program Move Mode Bit** – The ANG1(M) 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 32.

**Bit 7: Move Complete Bit** – Set to “1” when the present Absolute, Relative, Manual, Registration, or Assembled Move command completes without error. This bit is reset to “0” when the next move command is written to the ANG1(M), 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 Word 2 Format**

Status Word 2															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DrvEn	Stall	OUT1St	0	HeartBt	LmtCon	InvPar	MtnLag	DrvFit	NetLoss	PLCState	TempWarn	0	IN3St	IN2St	IN1St

Figure 8.5 Command Mode: Status Word 2 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 ANG1(M) 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: Output 1 State Bit** – Present actual state of Output 1. When this bit is set to “1”, the output is in its on state and conducts current.
- Bit 12: Reserved Bit** – Will always equal zero.
- Bit 11: Heart Beat Bit** – This bit will change state approximately every 500 milliseconds. Monitor this bit to verify that the module and network connection are operating correctly. Note 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. Parameters that can be changed during a Jog Move are Programmed Speed, Acceleration, and Deceleration. Set while in Encoder Follower mode if the Numerator or Denominator are set outside their range of 1 to 255.
- Bit 8: Motion Lag** – When using the Hybrid Control feature, this bit will equal “1” if the actual rotor position is more than 360° behind the commanded position. This indicates a heavily loaded motor and the encoder position should be monitored for a stalled condition.
- Bit 7: Driver Fault Bit** – If the driver section of the ANG1(M) is enabled, this bit will be a “1” during a Overtemperature Fault, or a Short Circuit Fault. This fault can be cleared by issuing a [Reset Errors](#) programming block with the Clear Driver Fault bit, (Command Bits LSW, bit 10) set to “1”. For additional information, see [Notes on Clearing a Driver Fault](#) on page 84.
- 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: PLC State Bit** – On ControlLogix and CompactLogix platforms, this bit equals “1” when the PLC is in Program mode and “0” when in Run mode. This bit will always equal “0” on all other platforms.
- Bit 4: Temperature 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 module, consider installing an additional IC-5 connector on each side of the ANG1(M) to allow for more cooling space, or install a fan below the stack to allow additional airflow through the stack.
- Bit 3: Reserved Bit** – Will always equal zero.

## ***Input Data Format (continued)***

### **Status Word 2 Format (continued)**

- Bit 2: Input3 State Bit** – “1” when Input 3 is in its active state. The active state of the input is programmed as explained in the *Configuration Word Format* section starting on page 64.
- 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 64.
- 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 64.

### **Motor Position Data**

These two inputs words report the motor position based on the number of steps output by the ANG1(M). 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 51. Note that both words in the multi-word format will be negative if the position value is negative. Also note that a roll over in the position value will cause the position to become invalid.

### **Encoder Position Data**

These two inputs words report the encoder position based on the quadrature pulses received by the ANG1(M). 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 51. 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 encoder position based on the quadrature pulses received by the ANG1(M) when an Input 3 transitions for its inactive to its active state. Input 3 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 51. 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, or a short circuit condition in the motor. When a Driver Fault occurs, the ANG1(M) sets bit 7 of Status Word 2 in the Network Input Data. (See [Status Word 2 Format](#) on page 82 for a full description of Status Word 2.) *Once you have cleared the fault condition, you can reset the Driver Fault with the following Command Block:*

#### Reset Driver Fault

Modbus Register	Function	Units	Range
1024	<i>Command Bits MSW</i>		16#0400
1025	<i>Command Bits LSW</i>		16#0400
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Unused		See Note Below
1033	Unused		See Note Below

Table 8.21 Reset Driver Fault Command Block

Unused words are ignored by the ANG1(M) and can be any value, including parameter values in the previous command.

Once the command block is accepted by the ANG1(M), it will respond by resetting bit 7 in Status Word 2 of the Network Input Data.

**NOTE** 

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

## CHOOSING YOUR MOTOR

### Sizing Your Motor

Your motor choice is based on the output torque you need, the mounting space you have, and your budgetary constraints. Torque curves for all of AMCI's motors are available on the following pages. Torque curves show the performance of the motor at 4Arms, which is the maximum setting for the ANG1(M).

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

- 1) The torque curves in this manual are for the ANG1(M). You cannot use these curves to accurately determine the amount of torque from an AMCI motor when it is attached to a different driver. Nor can you accurately determine the amount of torque from a motor when attached to an ANG1(M) if its torque curves were generated using a different driver. In general, if a output bus of the foreign driver is not the same as the voltage supplied to the ANG1(M), then the torque curves will be less accurate at higher speeds.
- 2) Make sure that the motor can provide the needed torque over the entire speed range of your application. Available torque drops as speed increases, so evaluate the motor's torque at its highest operating speed.
- 3) All of the torque curves show when the motor's windings are attached to the ANG1(M) in parallel. Parallel attached motors have the advantage of more torque at high speeds when compared to series attached motors.

A simple guideline is to use the largest motor your mounting space and budgetary constraints allow. Because the  $I^2R$  losses in the motor's windings manifest themselves as heat, the maximum allowable motor temperature limits the motor's current. Using the largest motor possible may allow you to use a lower current setting on the ANG1(M) driver. This lowers the  $I^2R$  losses, and the operating temperature of the motor, which increases the motor's life.

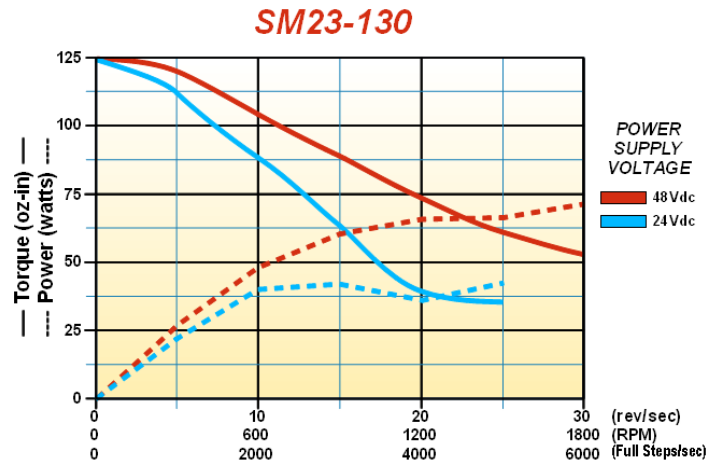


Figure A.1 SM23-130 Torque Curve (4Amp)

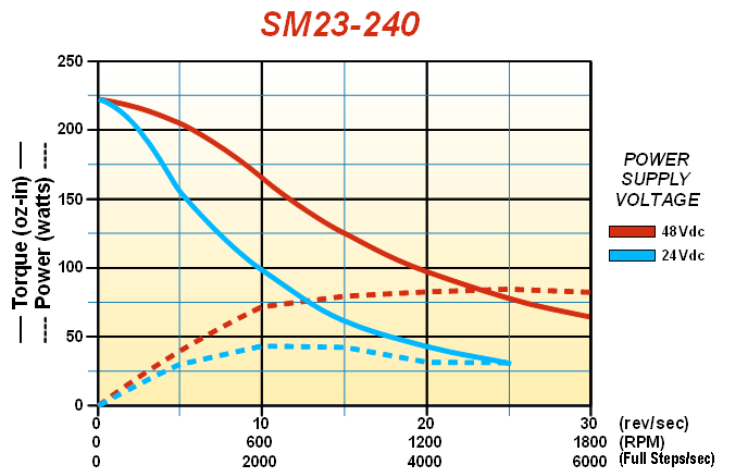


Figure A.2 SM23-240 Torque Curve (4Amp)

## Determining Your Motor Current Setting

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

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

## A Note on Microstepping

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

However, the ANG1(M) controls the RMS current through the motor. Therefore, the current supplied by the ANG1(M) when microstepping is always the power equivalent of the full step current. This means that the motors' full torque is always available. At very low speeds, the ANG1(M) automatically switches to peak current control to prevent motor damage.

**SM34-450**

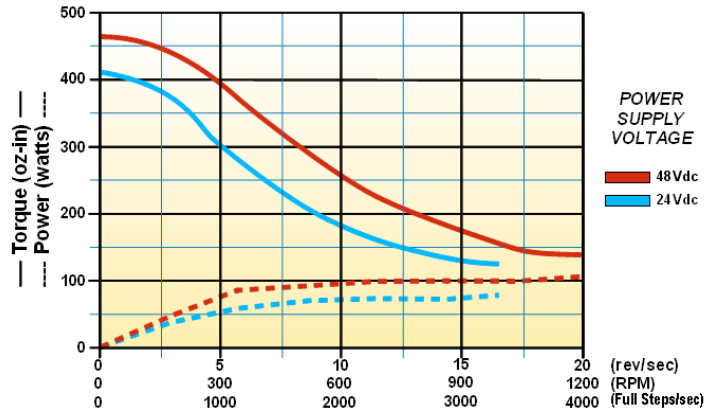


Figure A.3 SM4-450 Torque Curve (4Amp)

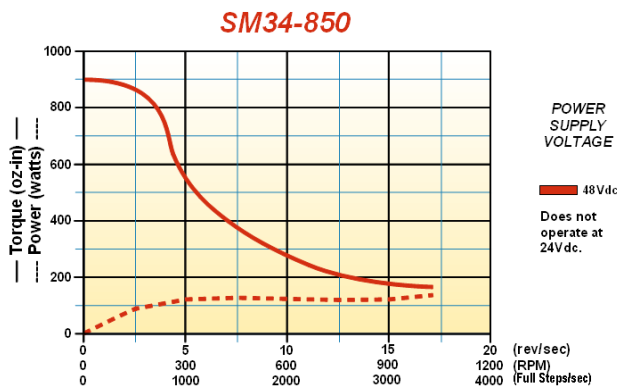


Figure A.4 SM34-850 Torque Curve (4 Amp)

**SM34-1100**

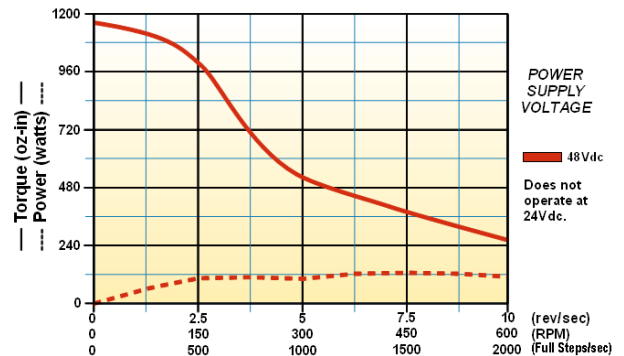


Figure A.5 SM34-1100 Torque Curve (4Amp)



*Notes*



***ADVANCED MICRO CONTROLS INC.***

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

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

***LEADERS IN ADVANCED CONTROL PRODUCTS***