AMCI

by

IDEC CORPORATION

IANG1 and IANG1E
Drive + Controller
User's Manual

AMCI Motion Control for IDEC FC6A Series MicroSmart

940-0I013

TABLE OF CONTENTS

About this Manual

Reference: IANG1(E) Specifications

Reference: Move Profiles

Reference: Calculating Move Profiles

Reference: Homing an IANG1(E)

Task: Installing the IANG1(E)

Task: Set the IP Address

Task: WindLDR 8.5 Project Setup

Task: Installing Custom Macros

Task: Using a Custom Macro in WindLDR

Task: Additional Logic and Input Data Formats

Reference: Config. Mode Output Data Format

Reference: Command Mode Output Data Format

Reference: Assembled Moves

Optional Task: Configure Your Network Interfaces

Notes

ABOUT THIS MANUAL

Introduction

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 audiences.

Audience

This manual explains the set-up, installation, and operation of the IANG1 and IANG1E AnyNET-I/O Stepper Motor Drive + Controller from AMCI. These devices are designed by Advanced Micro Controls Inc. for IDEC corporation. This manual is written for the engineer responsible for incorporating these modules into a design, as well as the engineer or technician responsible for their actual installation.

Applicable Units

This manual applies to the IANG1 and IANG1E modules. The IANG1E contains an integral network connection that allows the IANG1E to connect itself, and up to five other modules, to an IDEC MicroSmart FC6A controller over a Modbus TCP network. When information in this manual applies equally to both units the term "IANG1(E)" is used.

Navigating this Manual

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

Manual Conventions

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

Note► Notes highlight important concepts, decisions you must make, or the implications of those decisions.

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

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

The following table shows the text formatting conventions:

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

Trademark Notices

The AMCI logo and "AnyNET-I/O" are trademarks of Advanced Micro Controls Inc. "FC6A Series MicroSmart" is a trademark of IDEC Corporation. "Adobe" and "Acrobat" are registered trademarks of Adobe Systems Incorporated.

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

Revision Record

This manual, 940-0I013, is the first revision of this manual.

Revision History

940-0I013: Initial Release.

Manual Layout

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

- If you are curious about the IANG1 or IANG1E products, this manual contains the information you need to determine if these products are the right products for your application. The first chapter, *IANG1(E) Specifications* contains all of the information you will need to fully specify the right product for your application.
- If you need to install and use an IANG1 or IANG1E product, then the rest of the manual is written for you. To simplify installation and configuration, the rest of the manual is broken down into *references* and *tasks*. Using an IANG1 or IANG1E product requires you to complete multiple tasks, and the manual is broken down into sections that give the background information you need to understand how to use IANG1(E) (references), and instructions to complete operations (tasks).

Section Title	Starting Page #	Section Description
IANG1(E) Specifications	9	Complete specifications for the IANG1 and IANG1E products.
Move Profiles	19	Reference information on how the IANG1(E) can be used to control motion in your application.
Calculating Move Profiles	29	Reference information on calculating detailed move profiles.
Homing an IANG1(E)	37	Reference information on how to set the home position of the IANG1(E).
Installing the IANG1(E)	41	Task instructions covering how to install an IANG1(E) on a machine. Includes information on mounting, grounding, and wiring specific to the units.
Set the IP Address	51	Task instructions that covers the options for setting the IP address on an IANG1(E).
WindLDR 8.5 Project Setup	53	Task instructions for adding an IANG1(E) to a WindLDR 8.4+ project.
Installing Custom Macros	57	Task instructions for downloading and installing custom macros into a new or existing WindLDR 8.4+ project.
Using a Custom Macro in WindLDR	59	Task instructions for adding a custom macro to a ladder logic program.
Additional Logic and Input Data Formats	71	Reference information on the format of the input data from the IANG1(E). Data formats for configuration and command responses are given.
Config. Mode Output Data Format	79	Reference information on the format of configuration data written down to the IANG1(E). Custom macros simplify the programming of the IANG1(E), and should be used in most cases. This reference is provided as a troubleshooting aid should the need arise.
Command Mode Output Data Format	83	Reference information on the format of command data written down to the IANG1(E). Custom macros simplify the programming of the IANG1(E), and should be used in most cases. This reference is provided as a troubleshooting aid should the need arise.
Assembled Moves	95	Instructions for the optional task of programming the IANG1(E) to perform assembled moves.
Configure Your Network Interfaces	99	Instructions for the optional task of configuring your computer to communicate with an IANG1(E) before setting its IP address.

REFERENCE 1: IANG1(E) SPECIFICATIONS

Introduction

This chapter contains all of the information needed to understand how the FC6A MicroSmart controller interfaces with AMCI by IDEC Corporation products and specify IANG1 or IANG1E devices for your project.

AMCI by IDEC Corporation Motion Control Products

AMCI by IDEC Corporation is a partnership between the two companies to bring AMCI motion control products to the FC6A MicroSmart product family. Products that have been designed and manufactured by AMCI are factory configured to work immediately with the FC6A controllers. Setting the IP address is the only step needed before incorporating these products into your system. The IP address can be set with a freely available utility from the IDEC website.

In addition to the simplified hardware configuration, IDEC Corporation has released a collection of macros that simplify programming the FC6A MicroSmart controllers. In addition to these macros, IDEC has defined a set of Data Registers that are used by all sample programs for the motion control products. These Data Registers predefine twelve motion axes. With the exception of the IANF2(E), each AMCI by IDEC Corporation product controls one axis. This is true for the IANG1(E) as well as the stepper motors with integrated controllers. The IANF2(E) products control two axes. These modules use two consecutive sets of Data Registers to control their two axes.

AnyNET-I/O

The IANG1 and IANG1E mount on a DIN rail and can be connected together through an edge card connector that snaps into the DIN rail. Up to six modules can be connected together in an *AnyNET-I/O Stack*. The IANG1E contains an Ethernet port that is used to connect the stack to the Modbus TCP network. The IANG1 does not contain an Ethernet port, and can only function as a member of an *AnyNET-I/O Stack*.

Module types can be mixed within an AnyNET-I/O Stack. For example, a stack can contain an IANG1E module, two IANG1 modules and two IANF2 modules for a total of five modules controlling a total of seven motion axes. When the FC6A is configured, the IP address assigned to the IANG1E would be given enough data registers to communicate with the seven axes.

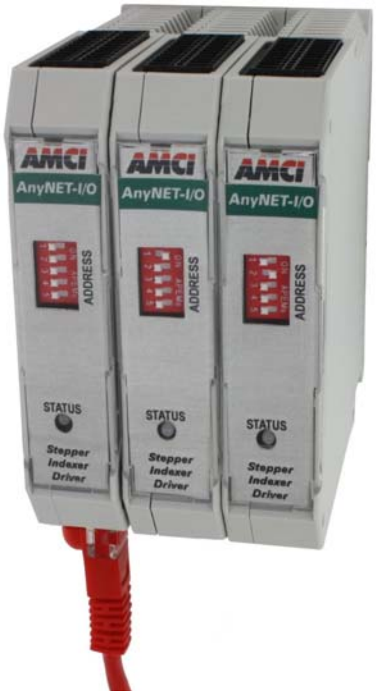


Figure R1.1 AnyNET-I/O Module Stack

The IANG1 and IANG1E

General Functionality

The IANG1(E) is a 4.0 Arms micro-stepping driver that accepts 24 to 48 Vdc as its input power source. What makes the IANG1(E) unique is its built-in controller that accepts configuration and command data from a host FC6A MicroSmart controller over a Modbus TCP network. This combination of host and driver gives you several advantages:

- Sophisticated I/O processing can be performed in the FC6A before sending commands to the IANG1(E)
- All motion logic is programmed in the FC6A, eliminating the need to learn a separate motion control language
- Eliminating the separate controller lowers Total System Cost

The IANG1(E) is powered by a nominal 24 to 48 Vdc power source, and can accept surge voltages of up to 60Vdc without damage. The output motor current is fully programmable from 0.1 Arms to 4.0 Arms which makes the IANG1(E) compatible with the complete line of AMCI stepper motors available directly from IDEC. In addition to the Motor Current setting, the Motor Steps per Turn, Idle Current Reduction, and Anti-Resonance Circuit features are also fully programmable.

The IANG1(E) 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 IANG1(E) 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 IANG1(E) has three DC inputs and one DC output that are used by the controller. Configuration data from the FC6A sets the function of these points. The output can be configured to be a Fault Output or a general purpose output. Inputs accept 5 to 24 Vdc signals and they can be individually configured as a:

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

Encoder Functionality

In addition to the discrete I/O points, the IANG1(E) has three inputs for a 5Vdc differential quadrature encoder. The inputs will also accept 12 to 24Vdc single ended encoder inputs with current limiting resistors.

Using the encoder inputs gives you the ability to:

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

The encoder input also allows you to drive the motor through a feature called *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.

Modes of Operation

The ANG1(E) has two basic modes of operation. A single bit in the output data written to the module sets the mode of operation.

Configuration Mode

Data written to the IANG1(E) in this mode configures the module for your application. Parameters set in this mode include:

- Input Functionality (IN1, 2 & 3)
- Input Active State (IN1, 2 & 3)
- Encoder Inputs Enable/Disable
- Home to Encoder Z-pulse Enable/Disable
- Backplane_Proximity bit Enable/Disable
- Stall Detection Enable/Disable
- Antiresonance Enable/Disable
- Output Functionality
- Output State on Network Loss
- Move Starting Speed
- Motor Steps per Turn
- Encoder Pulses per Turn
- Idle Current Percentage
- Motor Current
- Current Loop Gain

Command Mode

Data written to the IANG1(E) in this mode controls motion in the axis. This includes commands that begin motion as well as commands that interrupt running motion profiles and brings them to a controlled or immediate stop. Data written to the module in this mode also allows you to:

- Preset motor and encoder positions
- Reset command errors
- Control the state of the module's DC output
- Enable/Disable motor current

Power Up Behavior

The ANG1(E) will always power up in Command Mode and show a configuration error. Configuration data must be written to the ANG1(E) before commands can be issued to the module.

Note➤ The ANG1(E) will not supply power to the motor as long as there is a configuration error. (The configuration data sets, among other things, the motor current value. Without the information, the ANG1(E) does not know what current to apply to the motor.) Therefore, the motor will have no holding torque while a configuration error exists.

Specifications

Driver Type

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

Physical Dimensions

Width: 0.9 inches max.

Depth: 4.5 inches max.

Height: 3.9 inches

5.0 inches min. with mating connectors

Weight

0.38 lb. (0.17 kg) with mating connectors

Inputs

Electrical Characteristics: .... Differential. 1500 Vac/dc opto-isolated. Can be wired as single ended inputs.

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

Encoder Inputs are designed for 5 Vdc differential and require external current limiting resistor for 12 to 24 Vdc operation

Output

Electrical Characteristics: .... Open Collector/Emitter. 5 to 24 Vdc typical. 30 Vdc, 20 mA max. Opto-isolated to 560 Vac/dc.

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, and can be cleared through, the Data Registers assigned to the module.

Motor Current

Programmable from 0.1 to 4.0 Arms 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 48 Vdc, surge to 60 Vdc without damage to module.

Ambient Operating Temperature

.... -4° to 122° F (-20° to 50° C)

Storage Temperature

.... -40° to 185° F (-40° to 85° C)

Humidity .... 0 to 95%, non-condensing

Motor Specifications

Type .... 2 phase hybrid. 4, 6, or 8 lead motor

Inductance ... 0.3 mH minimum. 2.5 to 45 mH recommended

Status LEDs

See *Status LED* found on page 15 and *Network Status LED's* found on page 16.

Connectors

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

Connector	Part #	Wire	Strip Length	Min. Tightening Torque
I/O	IMS-2x11	28 - 16 AWG	0.275 inches	Spring Cage Connector
Motor	IMS-4M	28 - 12 AWG	0.394 inches	4.43lb-in (0.5 Nm)
Backplane	IIC-5			

Controller Functionality

The table below lists the functionality offered by the controller built into the IANG1(E).

Feature	Description
Programmable Inputs	Each of the three inputs can be programmed as a Home Limit, Over Travel Limit, Capture Input, Stop Jog or Registration Move, E-Stop Input, or a General Purpose Input.
Programmable Output	The single output on the IANG1(E) can be programmed as a Fault Output or as a general purpose DC output point.
Encoder Inputs	Allows the IANG1(E) to used a quadrature encoder for position verification or to drive the motor in response to changes on the encoder inputs. (Encoder Follower Mode)
Programmable Parameters	Starting Speed, Programmed Speed, Acceleration, Deceleration, Distance to Move, and Accel/Decel Types are fully programmable.
Homing	Allows you to set the machine to a known position. The IANG1(E) 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 received, the move will output a programmable number of steps before coming to a stop.
Relative Move	Allows you to drive the motor a specific number of steps in either direction from the current location.
Absolute Move	Allows you to drive the motor from one known location to another known location.
Blend Move	Allows you to perform a sequence of relative moves without stopping between them.
Dwell Move	Allows you to perform a sequence of relative moves with a stop between each move that has a programmable length of time. Used to create highly accurate move profiles that avoid network latency issues.
Indexer Move	Allows you to program a move that is not run until one of the programmable inputs makes a transition.
Hold Move	Allows you to suspend a move and restart it without losing your position value.
Resume Move	Allows you to restart a previously held move operation.
Immediate Stop	Allows you to immediately stop all motion if an error condition is detected by your host controller.
Stall Detection	When the IANG1(E) uses an encoder, the encoder can be used to verify motion when a move command is issued.
Encoder Follower	The IANG1(E) 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.

Table R1.1 Controller Functionality

Additional Notes on Stall Detection

When Stall Detection is enabled, the IANG1(E) monitors the encoder for position changes, regardless of whether or not a move is in progress. If the error between the encoder position and the motor position exceeds forty-five degrees, the IANG1(E) responds in the following manner:

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

Note that a move does not have to be in progress for stall detection to be useful. If Idle Current Reduction is set to 0%, or if the motor driver section is disabled over the network, power is removed from the motor. By enabling stall detection, the IANG1(E) can notify the system if the motor shaft moves more than forty-five degrees while power is removed from the motor.

Driver Functionality

This table summarizes the features of the stepper motor driver portion of the IANG1(E).

Feature	Benefits
RMS Current Control	RMS current control give the IANG1(E) the ability to drive the motor at its fully rated power when microstepping. Peak current controllers typically experience a 30% drop in power when microstepping a motor.
Programmable Motor Current	RMS current supplied to the motor can be programmed from 0.1 to 4.0 amps in 0.1 amp increments. This allows you to use the driver with the full line of stepper motors available through the AMCI by IDEC Corporation partnership.
Programmable Idle Current Reduction	Extends motor life by reducing the motor current when not running. This extends the life of the motor by reducing its operating temperature.
Programmable Current Loop Gain	Allows you to tailor the driver circuitry to the motor's impedance, thereby maximizing your motor's performance.
Programmable Motor Steps/Turn	Allows you to scale your motor count to a real world value. (counts per inch, counts per degree, etc.)
Anti-Resonance Circuitry	This circuitry gives the IANG1(E) 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 IANG1(E) sets a warning bit in the FC6A Data Registers when the temperature of the module approaches its safe operating threshold.
Over Temperature Protection	Protects the IANG1(E) from damage by removing power from the motor if the internal temperature of the driver exceed a safe operating threshold.

Table R1.2 Driver Functionality

Idle Current Reduction

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

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

Note► Note that the reduction values are "to" values, not "by" values. Setting a motor current to 2 Arms and the current reduction to 25% will result in an idle current of 0.5 Apk. (The IANG1(E) always switches from RMS to peak current control when the motor is idle to prevent motor damage due to excessive heating.)

Available Discrete Inputs

The IANG1(E) has three discrete DC inputs that accept 3.5 to 27 Vdc signals. (5 to 24 Vdc nominal) They can be wired as differential, sinking, or sourcing inputs. How the IANG1(E) uses these inputs is fully programmable as is their active states. (Inputs can be programmed as Normally Open (NO) or Normally Closed (NC) inputs.)

Home Input

Many applications require that the machine be brought to a known position before normal operation can begin. This is commonly called "homing" the machine or bringing the machine to its "home" position. The IANG1(E) 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 IANG1(E) that will cause the unit to seek this sensor. The third option is homing to the Z pulse of a quadrature encoder. When using the Z pulse, you can use one of the inputs or a network data bit as a home proximity sensor. How the IANG1(E) actually finds the Home sensor is described in reference chapter 3, *Homing an IANG1(E)*, starting on page 37.

CW Limit Switch or CCW Limit Switch

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

Start Indexer Move Input

Indexer Moves are programmed through the FC6A Data registers like every other move. The only difference is that Indexer Moves are not run until an input that is configured as a Start Indexer Move Input makes a inactive-to-active state transition. This allows the IANG1(E) 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 Manual 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 IANG1(E) can be configured to capture the encoder position value on a transition of a discrete DC input.

General Purpose Input

If your application does not require all three inputs, you can configure the unused inputs as General Purpose Inputs. The inputs are not used by the IANG1(E), but their state is reported in the network data.

Encoder Feedback Inputs

The IANG1(E) has three 5 Vdc differential inputs that accept quadrature encoder signals. These inputs can also accept single ended signals of 5 to 24 Vdc with the addition of external current limiting resistors.

An encoder is used by the IANG1(E) in multiple ways.

- When the encoder is mounted on the back of the motor controlled the IANG1(E), 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 reference chapter 3, *Homing an IANG1(E)*, starting on page 37. There are two options that can be used to determine which occurrence of the Z-pulse defines the home of the machine. A home proximity sensor can be wired into the IANG1(E) or the Backplane_Proximity_Bit in the network data can be used.
- Finally, it is also possible to use an encoder that is not mechanically coupled to the motor controlled by the IANG1(E). This configuration allows you to monitor the encoder data, or use a feature called *Encoder Follower.* When this feature is active, the IANG1(E) 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 IANG1(E) has a single DC output that has a maximum rating of 30 Vdc at 20 mA. It is typically used in a 5 to 24 Vdc circuit. 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 input data registers and are cleared by removing the fault and cycling power.

Front Panel

The front panels of three IANG1(E) modules are shown in figure R1.2. The front cover is hinged on the bottom, and swings down to allow you to change the module's address in the AnyNET-I/O Stack with the DIP switches. The front panel also has the Status LED, which give you information on the state of the module.

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 R1.2 shows the correct addressing for three modules. The module on the left is an IANG1E and has its address set to zero. The remaining modules can be IANG1 or IANG1E modules and their addresses are set to one and two.

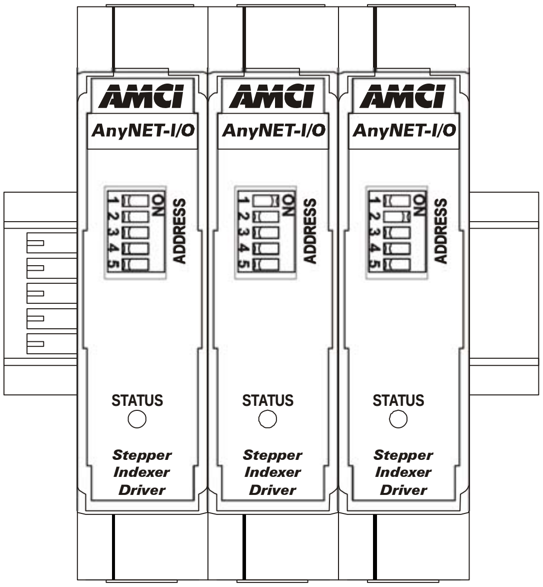


Figure R1.2 IANG1(E) Front Panel

Note➤ If an IANG1E has its address set to any value other than zero, its network interface is disabled. This allows you to use multiple IANG1E modules in a single AnyNET-I/O Stack.

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 IANG1(E) will only detect short circuit faults when the motor current is enabled.
- **Blinking Green:** Successful write to flash memory. Power must be cycled to the module before additional commands can be written to it.
- **Blinking Red:** Failed write to flash memory. You must cycle power to the module to clear this fault.
- **Alternating Red/Green:** Communications failure. This is either a communications error between the main processor and the Ethernet co-processor within the module or a communications error between modules in the AnyNET-I/O Stack. You must cycle power to the module to attempt to clear this fault.

I/O Connector

As shown in figure R1.3, the I/O connector is located on the top of the module. All digital I/O connections are made at this connector as well as the power supply connections. The mating connector is supplied with the IANG1(E) and is also available from IDEC under the part number IMS-2X11. It is also available from Phoenix Contact under their part number: 173 88 98.

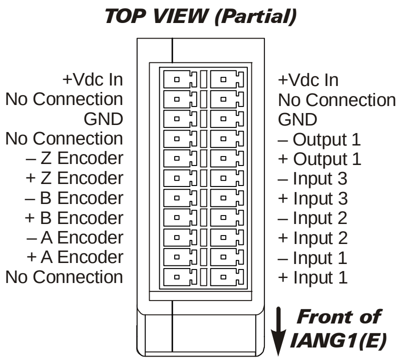


Figure R1.3 I/O Connector

Network Port

The IANG1E includes an Ethernet port. Physical connection is made through an industry standard RJ-45 port. Figure R1.4 shows the location of the network connector. An IANG1E can act as the network connection for up to five additional AnyNET-I/O modules.

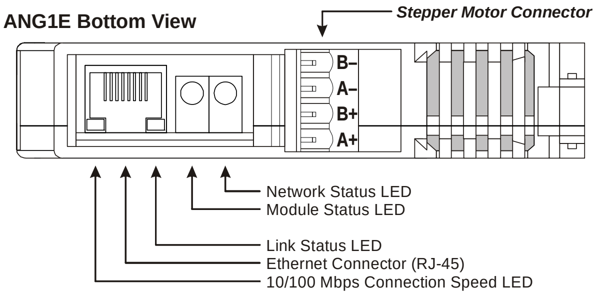


Figure R1.4 Ethernet Port and Motor Connector Locations

Network Status LED’s

The network status LED’s indicate the health of the network connection between the AnyNET-I/O module and its host.

- **Network Status –** Indicates the number of TCP connections to the AnyNET-I/O module. This LED will briefly flash red-green on power up while the module is initializing. It will flash green when a TCP connection is made to the driver. The number of flashes indicates the number of active connections. ANG1E modules support a maximum of three concurrent connections. There is a two second pause between flashes to make it easy to count the number of connections. If the LED is off, then there are no TCP connections to the module.
- **Module Status –** Indicates the health of the Ethernet co-processor on the module. This LED will briefly flash red once on power up and then flash green while the unit is initializing. It will be on steady green when the Ethernet co-processor is functioning correctly. It will turn red on an error. If this occurs, cycle power to the module.
- **Link Status –** On when there is a physical link between the Ethernet port of the AnyNET-I/O module and the Ethernet port of the device the module is plugged into. This LED will flash when data is being transmitted over the Ethernet link.
- **Connection Speed –** On when the Ethernet connection speed is 100Mbps. Off when the connection speed is 10 Mbps.

Motor Connector

Figure R1.4 also shows the location of the Stepper Motor Connector. The mate to this connector is included with the IANG1(E) and is also available from IDEC under the part number IMS-4M. It is also available from Phoenix Contact under their part number 187 80 37.

Torque and Power Curves

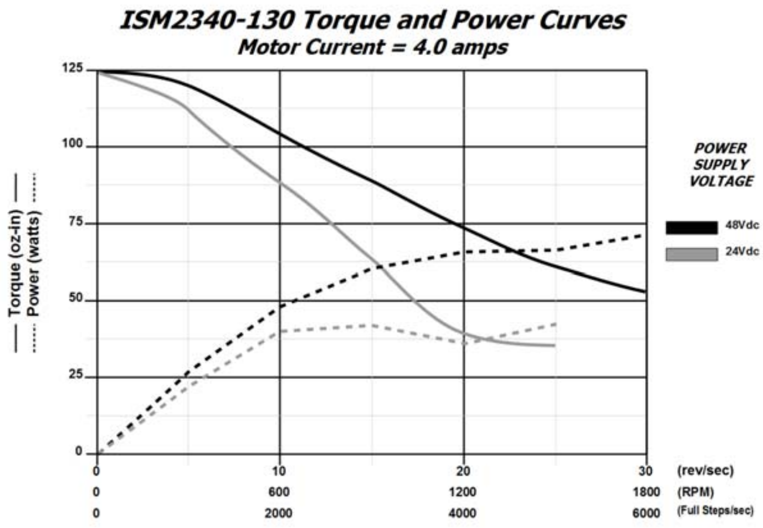


Figure R1.5 ISM2340-130 Torque and Power Curves

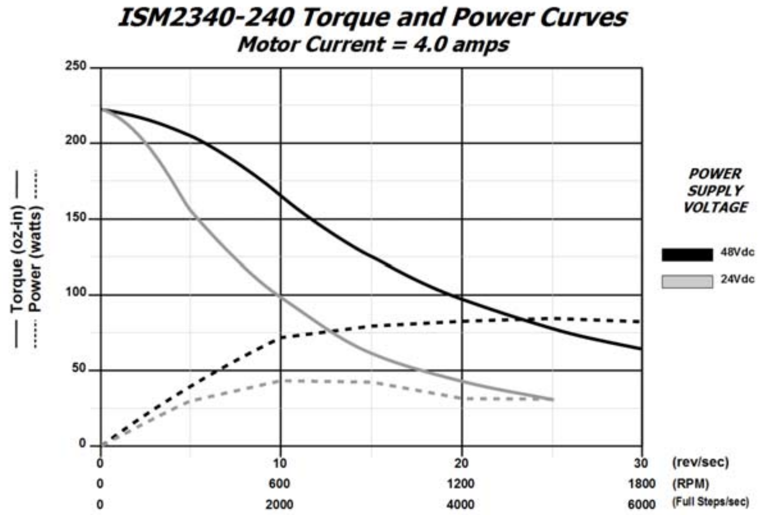


Figure R1.6 ISMD23-240 Torque and Power Curves

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 the motors available from IDEC are presented in figures R1.5 and R1.6. Torque curves show the performance of the motor at 4 Arms, which is the maximum setting for the IANG1(E).

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 IANG1(E). You cannot use these curves to accurately determine the amount of torque from an AMCI by IDEC motor when it is attached to a different drive. Nor can you accurately determine the amount of torque from a motor when attached to an IANG1(E) if its torque curves were generated using a different drive. In general, if an output bus of the foreign drive is not the same as the voltage supplied to the IANG1(E), 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 IANG1(E) 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 IANG1(E) drive. This lowers the I^2R losses, and the operating temperature of the motor, which increases the motor's life.

A Note on Microstepping

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

However, the IANG1(E) controls the RMS current through the motor. Therefore, the current supplied by the IANG1(E) 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 IANG1(E) automatically switches to peak current control to prevent motor damage.

Power Supply Sizing

The power supply is connected to the pins marked "+Vdc In" and "GND". The 24 to 48 Vdc external power supply also powers the stepper motor, so it must be rated to supply current to it. As a general guideline, your supply should be able to produce 150% to 175% of the power the motor can produce. The power and torque curves above can be used to determine the maximum power the motor can generate over its speed range.

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

Example 1: An ISMD23E2-130 will be running at a maximum of 20 RPS and a 48 Vdc supply will be used. Based on the power curve in figure R1.5 on page 16, the power at this speed is approximately 65 watts, which is the maximum power over the entire speed range. Therefore, the 48 Vdc supply should be able to supply 100 to 115 watts of power.

Example 2: An ISMD23E2-240 will be running at a maximum of 20 RPS and a 24 Vdc supply will be used. Based on the power curve in figure R1.6 on page 17, the power at this speed is approximately 30 watts, but the maximum power over the entire speed range is 45 watts, which occurs at 10 RPS. Therefore, the 45 watt value should be used, and the 24 Vdc supply should be able to supply 68 to 79 watts of power.

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

		ISMD23E2-130			ISMD23E2-240		
		Motor Power	150% Supply	175% Supply	Motor Power	150% Supply	175% Supply
Supply Voltage	24 Vdc	45 W	68 W	79 W	45 W	68 W	79 W
	48 Vdc	70 W	105 W	123 W	85 W	128 W	149 W

Table R1.3 Suggested Power Supply Ratings

Regeneration (Back EMF) Effects

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

The ISMD2340-130 and ISMD2340-240 motors are low inductance motors. Regeneration effects are not an issue in most applications when using these motors. The one exception is when a gearhead is placed on the motor and the system allows the gearhead shaft to be manually rotated. In these applications, the motor may rotate at high speeds and act as a generator.

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

⚠ **Caution** Regeneration events can raise the Vdc supply voltage to an unexpected value. Components, including supplies not rated for this voltage may be damaged.

REFERENCE 2: MOVE PROFILES

Introduction

When a move command is sent to an IANG1(E), the unit calculates the entire profile before starting the move or issuing an error message. This chapter explains how the profiles are calculated and the different available moves.

Definitions

Units of Measure

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

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

Acceleration: The typical unit of measure for acceleration and deceleration is steps/second/second, or steps/second^2. However, when programming an IANG1(E), all acceleration and deceleration values must be programmed in the unit of measure of steps/millisecond/second.

- To convert from steps/second^2 to steps/millisecond/second, divide the value by 1000. This must be done when converting from a value used in the equations to a value programmed into an IANG1(E).
- To convert from steps/millisecond/second to steps/second^2, multiply the value by 1000. This must be done when converting from the value programmed into an IANG1(E) to the value used in the equations.

Motor Position

Motor Position is defined in counts, and its range is -2,147,483,648 to +2,147,483,647 counts. The optional encoder has the same range.

Home Position

The Home Position is any position on your machine that you can sense and stop at. There are two ways to defining the Home Position. The first is using the Preset Position command to set the Motor Position register to a known value. The second method is using one of the *Find Home* commands. If you use the unit's *Find Home* commands, the motor position and encoder position registers will automatically be set to zero once the home position is reached. Defining a Home Position is completely optional. Some applications, such as those that use the IANG1(E) for speed control, don't require position data at all.

Valid and Invalid Positions

The Motor Position is considered *Valid* once the Home Position has been defined. Until that time, the motor position is considered *Invalid*. Absolute moves cannot be run unless the position is Valid. Uncontrolled, immediate stops will force the position to become Invalid.

Count Direction

Clockwise moves will always increase the motor position register reported back to the host. Some of the moves, such as the Jog Move, have a positive and negative command. A positive command, such as the +Jog Move command, will result in a clockwise rotation of the shaft.

Starting Speed

The Starting Speed is the speed that most moves will begin and end at. This value is set while configuring the unit and it has a valid range of 1 to 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. A default value is not specified in this manual because it is very dependent on motor size and attached load. While bench testing the IANG1(E), a starting speed between 0.25 and 0.5 RPS is generally a safe value to begin with.

Target Position

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

Relative Coordinates

Relative coordinates define the Target Position as an offset from the present position of the motor. Most IANG1(E) moves use relative coordinates.

- The range of values for the Target Position when it is treated as an offset is ±8,388,607 counts. Positive offsets will result in clockwise moves, while negative offsets result in counter-clockwise moves.
- The Motor Position value reported back to the host can exceed ±8,388,607 counts. The only way to move beyond ±8,388,607 counts is with multiple relative moves or jog commands.

Absolute Coordinates

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

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

Definition of Acceleration Types

Most of the move commands allow you to define the acceleration type used during the move. The IANG1(E) supports three types of accelerations and decelerations. The type of acceleration used is controlled by the *Acceleration Jerk* parameter.

What is jerk?

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

The IANG1(E) uses the jerk property to smoothly change the acceleration applied during the move. In this case, the speed of the move does not increase linearly, but exponentially, resulting in an “S” shaped curve. This limits mechanical shocks to the system as the load accelerates.

In order to keep the Acceleration Jerk parameter value that is programmed into the IANG1(E) below sixteen bits, the Acceleration Jerk parameter that is programmed into the IANG1(E) does not have units of steps/sec^3. The Acceleration Jerk parameter equals ({100 * jerk in steps/sec^3} / acceleration in steps/sec^2). This translates to the jerk property in steps/sec^3 equalling ({Acceleration Jerk parameter/100} * acceleration in steps/sec^2). When the Acceleration Jerk parameter is set to zero, the resulting jerk value is zero. Within the remaining range of values for the Acceleration Jerk parameter being 1 to 5,000, the jerk value ranges from 0.01*a* to 50*a* where “*a*” is the acceleration value in steps/sec^2. For example, if the acceleration is programmed to 20,000 steps/sec^2, then the jerk value used by the module can be programmed to be between 200 steps/sec^3 (0.01*20,000) and 1,000,000 steps/sec^3 (50*20,000).

Constant Acceleration

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

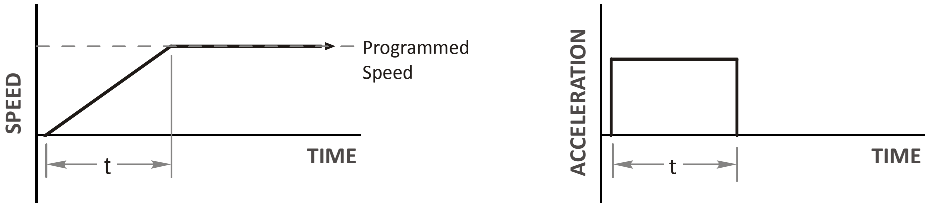


Figure R2.1 Constant Acceleration

Additional information, including example move calculations, can be found in reference chapter 8, *Calculating Move Profiles* starting on page 29.

S-Curve Accelerations

When the Acceleration Jerk parameter value is in the range of 1 to 5,000, the IANG1(E) uses this value to accelerate and decelerate the rate of acceleration. This is known as S-Curve acceleration because of the shape of the speed curve that results from the variable acceleration.

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

Triangular S-Curve Acceleration

When the Acceleration Jerk parameter is set low, Triangular S-Curve acceleration usually results. This occurs because the programmed maximum acceleration value is not reached before the IANG1(E) must start decreasing the acceleration value as the move's speed approaches its programmed maximum value. Triangular S-Curve is the smoothest form of acceleration, but the time needed to reached the move's programmed speed is increased. An example is shown in figure R2.2 where the acceleration and jerk settings results in a move that takes twice as long as a Constant Acceleration move to achieve the same velocity.

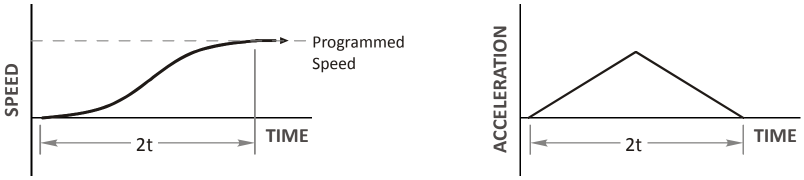


Figure R2.2 Triangular S-Curve Acceleration

Trapezoidal S-Curve Acceleration

When the Acceleration Jerk parameter is set high, Trapezoidal S-Curve acceleration usually results. The acceleration value quickly increases (accelerates) until it reaches the value of the Acceleration Parameter. At this point, the acceleration remains constant until the IANG1(E) begins to apply the jerk property value to decrease the acceleration value until it equals zero when the programmed maximum speed is reached. Figure R2.3 shows a trapezoidal curve when the Acceleration Jerk setting results in the acceleration being constant for half of the acceleration time. With this setting, the Trapezoidal S-Curve acceleration only requires 33% more time to achieve the same velocity as a Constant Acceleration move.

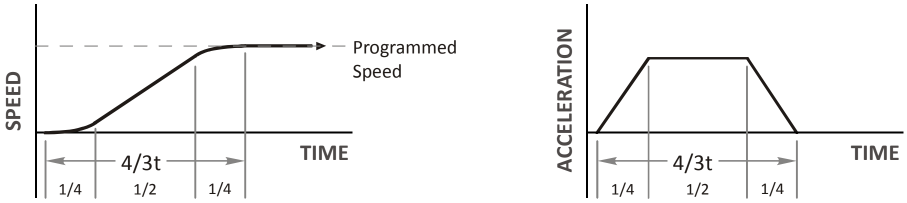


Figure R2.3 Trapezoidal S-Curve Acceleration

Additional information, including example move calculations, can be found in reference chapter 8, *Calculating Move Profiles* starting on page 29.

A Simple Move

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

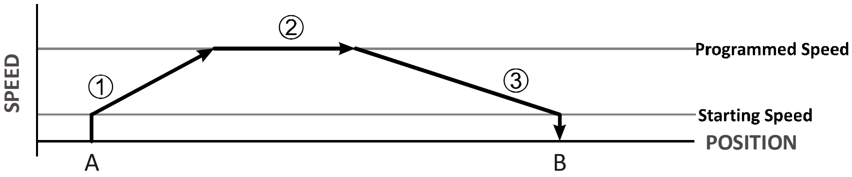


Figure R2.4 A Trapezoidal Profile

1) The move begins at point A, where the motor jumps from rest to the configured *Starting Speed*. The motor then accelerates at the programmed *Acceleration Value* until the speed of the motor reaches the *Programmed Speed*. Both the Acceleration Value and the Programmed Speed are programmed when the move command is sent to the IANG1.
2) The motor continues to run at the Programmed Speed until it reaches the point where it must decelerate before reaching point B.
3) The motor decelerates at the *Deceleration Value*, which is also programmed by the move command, until the speed reaches the Starting Speed, which occurs at the Target Position (B). The motor stops at this point. Note that the acceleration and deceleration values can be different in the move.

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

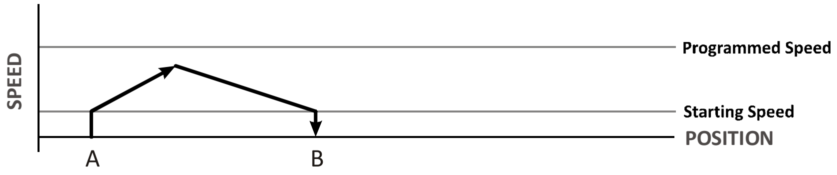


Figure R2.5 A Triangular Profile

Profile Equations

If your application requires very precise profiles, refer to reference chapter 8, *Calculating Move Profiles* for information on time and distance formulas.

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 22, 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 ±Find Home commands, the ±Jog Move commands, and the ±Registration Move commands. The ±Find Home commands are explained in reference chapter 3, *Homing an IANG1(E)*, which starts on page 37. The *CW/CCW Jog Move* commands are fully explained on page 24, and the *CW/CCW Registration Move* commands are fully explained on page 25.

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

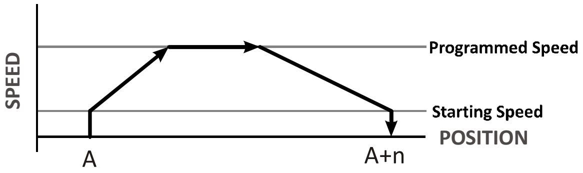


Figure R2.6 Relative Move

Note► 1) You do not have to preset the position or home the machine before you can use 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 perform a relative move of 30° multiple times without recalculating new target positions in your controller. If you perform the same action with Absolute Moves, you would have to calculate your 30° position followed by your 60° position, followed by your 90° position, etc.

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 IANG1(E) will set a *Hold State* bit in the input data table. The Relative Move can be restarted with the Resume Move command from the network data or the move can be aborted. The Resume Move command allows you to change the move's Programmed Speed, Acceleration Value and Type, and the Deceleration Value and Type. The Target Position cannot be changed with the Resume Move Command.

Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Relative Move by using the Resume Move command or abandon the held move by starting a new one. The use of the Hold Move and Resume Move bits is further explained in the *Controlling Moves In Progress* section starting on page 27.

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 IANG1(E) 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 ±8,388,607 counts. The move will be clockwise if the Target Position is greater than the Current Position and counterclockwise if the Target Position is less than the Current Position.

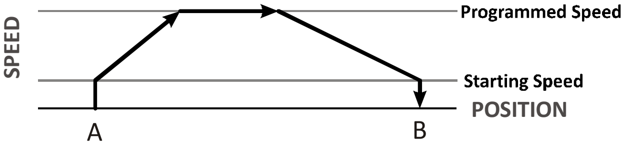


Figure R2.7 Absolute Move

Note► 1) 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. See the reference chapter 3, *Homing an IANG1(E)*, which starts on page 37, for information on homing the machine.

2) Absolute Moves allow you to move your machine without having to calculate relative positions. If you are controlling a rotary table, you can drive the table to any angle without having to calculate the distance to travel. For example an Absolute Move to 180° will move the table to the correct position regardless of where the move starts from.

Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold Move control bit in the Network Output Data. Note that your holding position will most likely not be the final position you commanded. You can resume a held Absolute Move by using the Resume Move bit or abandon the held move by starting a new one. The use of the Hold Move and Resume Move bits is explained in the *Controlling Moves In Progress* section starting on page 27.

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.

CW/CCW Jog Move

Jog Moves move in the programmed direction as long as the command is active. Two commands are available. The CW Jog Move will increase the motor position count while the CCW Jog Move will decrease the motor position count. These commands are often used to give the operator manual control over the axis.

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

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

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 IANG1(E) will decelerate the motor to the starting speed and stop without losing position. If it is an *Immediate Stop Condition*, the motion stops immediately and the position becomes invalid.

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

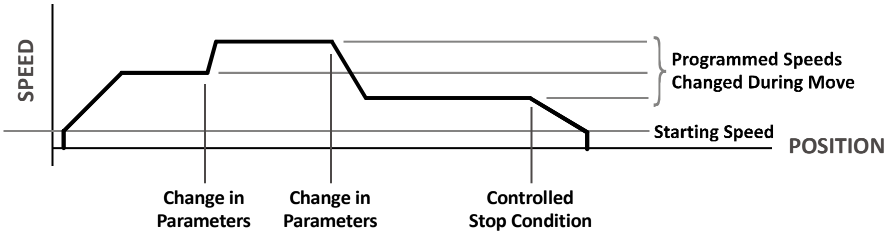


Figure R2.8 Jog Move

Controlled Stop Conditions

- The Jog Move Command bit is reset to “0”.
- An inactive-to-active transition on an input configured as a *Stop Jog or Registration Move* Input.
- You toggle the Hold_Move control bit in the Network Output Data. The use of the Hold_Move and Resume_Move bits is explained in the *Controlling Moves In Progress* section starting on page 27.

Immediate Stop Conditions

- The Immediate_Stop bit makes a 0→1 transition in the Network Output Data.
- A inactive-to-active transition on an input configured as an E-Stop Input.
- A CW or 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➤ Note that it is possible to *start* a move while a CW or CCW Limit Switch is active as long as the direction of travel is *opposite* that of the activated Limit Switch. For example, it is possible to start a CW Jog Move while the CCW Limit Switch is active

CW/CCW Registration Move

Similar to a Jog Move, a Registration Move will travel in the programmed direction as long as the command is active. CW Registration Moves increase the motor position count while the CCW Registration Moves decrease the motor position count. When the command terminates under Controlled Stop conditions, the IANG1(E) will output a programmed number of steps as part of bringing the move to a stop. Note that all position values programmed with a Registration Move are relative values, not absolute machine positions.

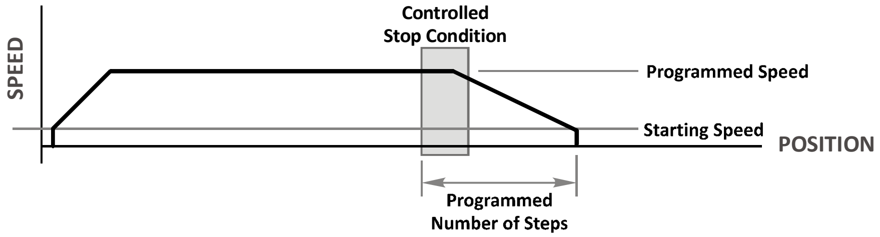


Figure R2.9 Registration Move

> **Note➤** If the Programmed Number of Steps are less than the number of steps needed to bring the axis to a stop based on the Programmed Speed and Deceleration values set with the command, the IANG1(E) will decelerate at the programmed Deceleration value until it has output the Programmed Number of Steps and then stop the move without further deceleration.

An additional feature of the Registration Moves is the ability to program the drive 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 the condition remains active, the move will begin its controlled stop once the Minimum Registration Move Distance is reached.

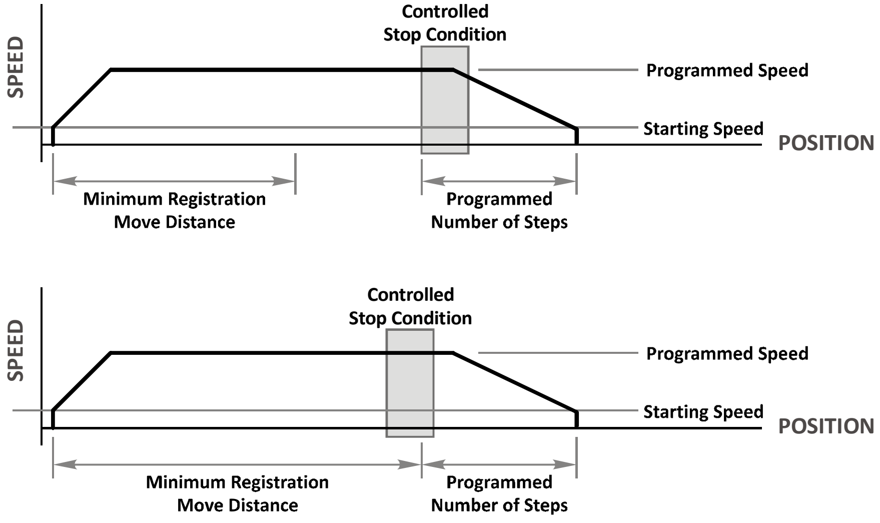


Figure R2.10 Min. Registration Move Distance

Controlled Stop Conditions

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

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

- You toggle the Hold_Move control bit in the Network Output Data. The ISMD23E2 responds by using the programmed Deceleration value to bring the move to a stop, without using the value of the Programmed Number of Steps parameter. A Registration Move does not go into the Hold State if the Hold_Move control bit is used to stop the move and it cannot be restarted with the Resume Move command.

Immediate Stop Conditions

- The Immediate_Stop bit makes a 0→1 transition in the Network Output Data.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- A CW or 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➤** Note that it is possible to start a move while a CW or CCW Limit Switch is active as long as the direction of travel is opposite that of the activated Limit Switch. For example, it is possible to start a CW Registration Move while the CCW Limit Switch is active.

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 IANG1(E) 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 IANG1(E) until the triggered move is finished.
- As started above, a move can be run multiple times as long at the move command data remains unchanged. If you wish to program a second move and run it as an Indexed Move type, then you must have a 0→1 transition on the move command bit before the new parameters are accepted. The easiest way to accomplish this is by writing a value of zero to the command word between issuing move commands.
- A Jog Move that is started as an Indexed Move will come to a controlled stop when the command bit in the Network Output Data is reset to zero.
- It is possible to perform an Indexed Registration Move by configuring two inputs for their respective functions. The first input, configured as a *Start Indexed Move Input*, starts the move and the second, configured as a *Stop Jog or Registration Move Input* causes the registration function to occur.
- You cannot issue a Hold Command with the Indexed Bit set and have the Hold Command trigger on the inactive-to-active transition of a physical input. Hold Commands are always acted upon as soon as they are accepted from the Network Output Data.
- You cannot issue an Immediate Stop Command with the Indexed Bit set and have the Immediate Stop Command trigger on the inactive-to-active transition of a physical input. Immediate Stop Commands are always acted upon as soon as they are accepted from the Network Output Data. If you need this functionality, consider programming the physical input as an E-Stop Input.
- You cannot issue a Clear Error Command with the Indexed Bit set and have the Clear Error Command trigger on the inactive-to-active transition of a physical input. Clear Error Commands are always acted upon as soon as they are accepted from the Network Output Data.

Controlling Moves In Progress

The IANG1(E) 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.

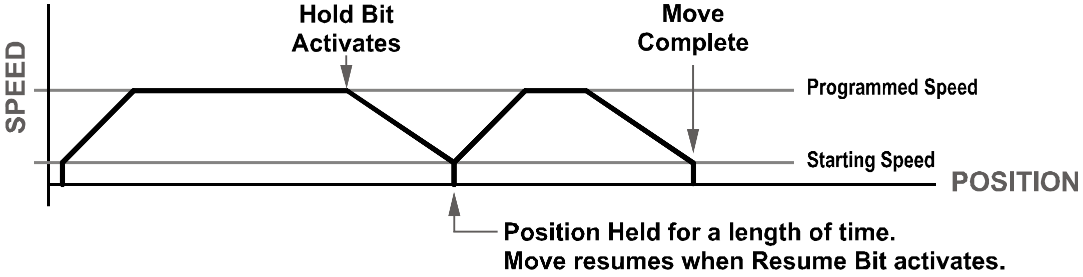


Figure R0.1 Hold/Resume a Move Profile

Find Home Moves

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

Jog Moves

Jog Moves can be placed in a Hold state and resumed if error conditions, such as programming errors, have not occurred. New Acceleration, Deceleration, and Programmed Speed parameters can be written to the IANG1(E) 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 Registration Moves

Absolute, Relative and Registration Moves can be placed in a Hold state and resumed if error conditions, such as programming errors, have not occurred. New Acceleration, Deceleration, and Programmed Speed parameters can be written to the IANG1(E) while these moves are in their hold states. If the parameters are accepted without error, the move can be resumed and it will use the new parameter values. Note that a change to the Target Position is ignored.

Assembled Moves

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

Encoder Follower

The final form of motion control available with the IANG1(E) is Encoder Follower mode. A quadrature encoder is required but it is not mounted on the motor controlled by the IANG1(E). The encoder is typically mounted on a second motor, but it can be mounted anywhere, including on something as simple as a hand crank.

This mode is sometimes referred to as *electronic gearing*, because the motor will change position in response to a change in position of the encoder. The IANG1(E) 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 IANG1(E). 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.

Encoder Follower Multiplier and Divisor

The ratio of these two parameters is applied to the number of encoder pulses read by the IANG1(E) 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 IANG1(E) always uses 4X decoding when counting pulses from the encoder. If you set both of your Encoder Follower 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 IANG1(E) monitors the Jog Move command bits in the output registers assigned to the module. When either of these bits are set, the encoder inputs are monitored for a change in position. When a change is sensed, the IANG1(E) will begin to turn the motor within 50 microseconds. An increase in encoder counts will result in an increase in motor position. A decrease in encoder counts will result in a decrease in motor position.

The values of the Encoder Follower Multiplier and Divisor can be changed while encoder follower motion is occurring. The IANG1(E) 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 output registers assigned to the module.
- A positive transition on an input configured as an E-Stop Input.
- A CW or CWW Limit Switch is reached.

Advanced Ratio Control

The Encoder Follower 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 Encoder Follower 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 Encoder Multiplier and Divisor parameters.

REFERENCE 3: CALCULATING MOVE PROFILES

Introduction

This reference is for customers that must program very precise profiles. Understanding this section is not necessary before programming the IANG1(E) and it can be considered optional. Two different approaches are presented here. The constant acceleration example takes given parameters and calculates the resulting profile. The variable acceleration example starts with a desired speed profile and calculates the required parameters.

Units of Measure

The equations in this appendix use a unit of measure of steps/second/second (steps/second^2) for acceleration and deceleration. However, when programming the IANG1(E), all acceleration and deceleration values must be programmed in the unit of measure of steps/second/millisecond.

- To convert from steps/second^2 to steps/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 IANG1(E).
- To convert from steps/millisecond/second to steps/second^2, multiply the value by 1000. This must be done when converting from the value programmed into the IANG1(E) to the value used in the equations.

Constant Acceleration Equations

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

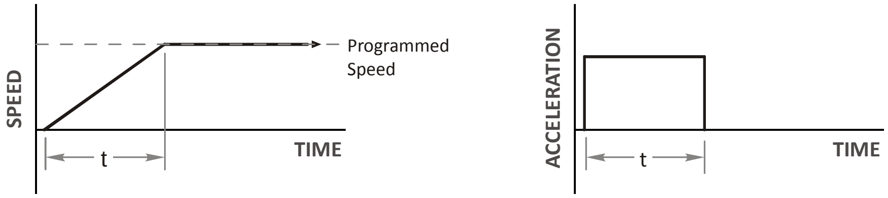


Figure R3.1 Constant Acceleration Curves

Variable Definitions

The following variables are used in these equations:

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

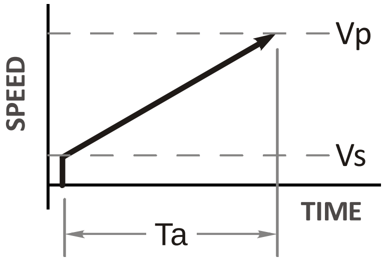


Figure R3.1 gives the equations to calculate Time, Distance, and Acceleration values for a constant acceleration move.

Acceleration Type	T_A or T_D (Time to Accelerate or Decelerate)	D_A or D_D (Distance to Accelerate or Decelerate)	a (Average Acceleration)
Linear	$T_A = (V_P - V_S)/a$	$D_A = T_A*(V_P + V_S)/2$	$a = (V_P^2 - V_S^2)/2D_A$

Table R3.1 Acceleration Equations

If the sum of the D_A and D_D values of the move is *less than* the total number of steps in the move, your move will have a Trapezoidal profile.

If the sum of the D_A and D_D values of the move is *equal to* the total number of steps in the move, your move will have a Triangular profile and your move will reach the Programmed Speed before it begins to decelerate.

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

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

Name	Value	IANG1(E) Parameter Values
Acceleration (a)	20,000 steps/sec^2	20
Deceleration (d)	25,000 steps/sec^2	25
Starting Speed (V_S)	141 steps/sec	141
Programmed Speed (V_P)	100,000 steps/sec	100,000

Table R3.2 Sample Values

From table R3.1:

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

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

If a move with the above acceleration, deceleration, starting speed, and programmed speed has a length greater than 449,984 steps, the IANG1(E) will generate a Trapezoidal profile. If the move is equal to 449,984 steps, the IANG1(E) will generate a Triangular profile and the unit will output one pulse at the programmed speed. If the move is less than 449,984 steps, the IANG1(E) will generate a Triangular profile and the programmed speed will not be reached.

In the case of a Triangular profile where the programmed speed is not reached, it is fairly easy to calculate the maximum speed (V_M) attained during the move. Because the move is always accelerating or decelerating, the total distance traveled is equal to the sum of D_A and D_D.

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

Therefore, total distance traveled =
$D_A + D_D = (V_M^2 - V_S^2)/2a + (V_M^2 - V_S^2)/2d$.

In the case where the acceleration and deceleration values are equal, this formula reduces to:
$D_A + D_D = (V_M^2 - V_S^2)/a$

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

$$\begin{aligned}
D_A + D_D &= \frac{V_M^2 - V_S^2}{2a} + \frac{V_M^2 - V_S^2}{2d} \\
300{,}000 \text{ steps} &= \frac{V_M^2 - 141^2}{2(20{,}000)} + \frac{V_M^2 - 141^2}{2(25{,}000)} \\
300{,}000 \text{ steps} &= \frac{V_M^2 - 20{,}000}{40{,}000} + \frac{V_M^2 - 20{,}000}{50{,}000} \\
300{,}000 \text{ steps} &= \frac{5}{5}\left(\frac{V_M^2 - 20{,}000}{40{,}000}\right) + \frac{4}{4}\left(\frac{V_M^2 - 20{,}000}{50{,}000}\right) \\
300{,}000 \text{ steps} &= \frac{5V_M^2 - 100{,}000}{200{,}000} + \frac{4V_M^2 - 80{,}000}{200{,}000} \\
300{,}000\ (200{,}000) &= 9V_M^2 - 180{,}000 \\
\frac{60{,}000.18 \times 10^6}{9} &= V_M^2 \\
V_M &= 81{,}650 \text{ steps/sec}
\end{aligned}$$

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

Total Time Equations

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

$$\mathbf{D_P = (Total\ Number\ of\ Steps) – (D_A + D_D)}$$
$$\mathbf{T_T = T_A + T_D + D_P/V_P}$$

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

$$\mathbf{T_T = T_A + T_D}$$

S-Curve Acceleration Equations

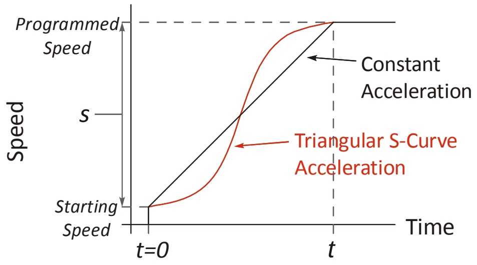
When the Acceleration Jerk parameter value is in the range of 1 to 5,000, the IANG1(E) uses this value to smoothly change the acceleration value applied during the move. In this case, the speed of the move does not increase linearly, but exponentially, resulting in an “S” shaped curve. This limits mechanical shocks to the system as the load accelerates. Just as constant acceleration will result in a trapezoidal or triangular speed profile, the Acceleration Jerk parameter will result in a trapezoidal or triangular acceleration phase.

In order to keep the Acceleration Jerk parameter value that is programmed into the IANG1(E) below sixteen bits, the Acceleration Jerk parameter programmed into the drive does not have units of steps/sec^3. The Acceleration Jerk parameter equals ({100 * jerk in steps/sec^3} / acceleration in steps/sec^2). This translates to the jerk property in steps/sec^3 equalling ({Acceleration Jerk parameter/ 100} * acceleration in steps/sec^2). With the range of values for the Acceleration Jerk parameter being 1 to 5,000, the jerk value ranges from 0.01*a* to 50*a* where “*a*” is the acceleration value in steps/sec^2. For example, if the acceleration is programmed to 20,000 steps/sec^2, then the value of the jerk property used by the unit can be programmed to be between 200 steps/sec^3 (0.01*20,000) and 1,000,000 steps/sec^3 (50*20,000). This statement applies to the Deceleration Parameter as well. If the Acceleration and Deceleration parameters are different, the calculated jerk values will also differ.

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

Triangular S-Curve Acceleration

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



s = Programmed Speed – Starting Speed

$$\text{Acceleration} = \frac{\text{speed}}{\text{time}} \qquad \text{jerk} = \frac{\text{acceleration}}{\text{time}} \qquad \begin{array}{c}\text{IANG1(E)}\\ \text{Acceleration}\\ \text{Jerk Parameter (J)}\end{array} = \frac{100j}{a}$$

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

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

Figure R3.2 Move Profile Example

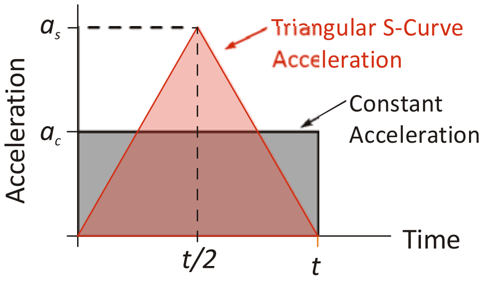


Figure R3.3 Triangular Acceleration

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

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

$$a_s = 2a_c$$

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

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

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

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

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

$$Ja_st = 200a_s$$

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

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

When $a_S = a_C$

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

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

Trapezoidal S-Curve Acceleration

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

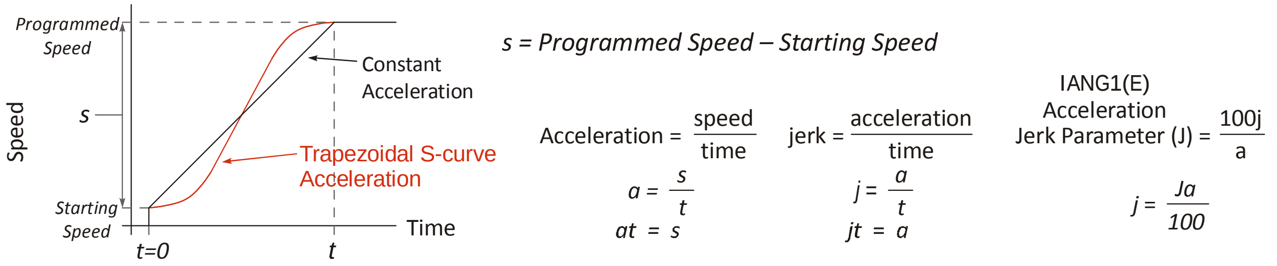

Programmed Speed
Speed
s
Starting Speed
t=0
t
Time
Constant Acceleration
Trapezoidal S-curve Acceleration
s = Programmed Speed – Starting Speed
Acceleration = speed / time
a = s / t
at = s
jerk = acceleration / time
j = a / t
jt = a
IANG1(E) Acceleration Jerk Parameter (J) = 100j / a
j = Ja / 100

Figure R3.4 Move Profile Example

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

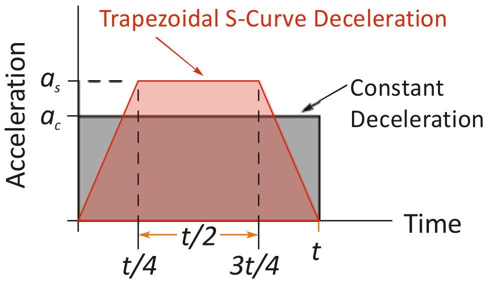


Figure R3.5 Trapezoidal Acceleration

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

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

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

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

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

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

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

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

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

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

$$Ja_s t = 400a_s$$

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

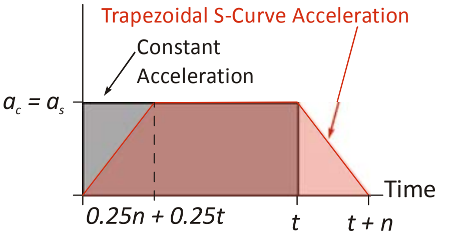
This value represents the ideal Acceleration Jerk parameter value for a trapezoidal S-curve acceleration with a constant acceleration for half of the phase. Setting the value lower than this will result in a shorter constant period, while setting the value greater than this will result in a longer constant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

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

When $a_S = a_C$

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

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



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

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

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

$$0.25t = .75n$$

$$t = 3n$$

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

Figure R3.6 Trapezoidal S-curve Time Increase Example

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

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

Determining Waveforms by Values

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

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

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

Example 1, Jerk = 20

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

S_m = midpoint of change in speed

J = Acceleration Jerk parameter

j = physical jerk property

a_f = calculated final acceleration

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

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

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

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

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

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

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

Just as speed = at, acceleration = jt

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

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

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

Example 2, Jerk = 400

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

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

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

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

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

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

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

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

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

Just as speed = at, acceleration = jt

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

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

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

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

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

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

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

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

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

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

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

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

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

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

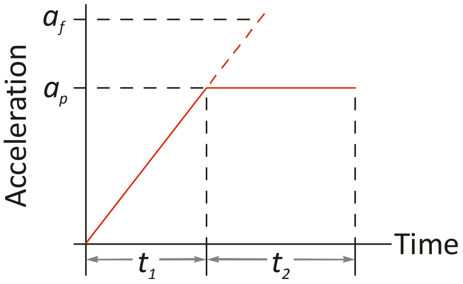


Figure R3.7 Calculating Trapezoidal S-Curve

The time for this acceleration phase is 2(t1 + t2), which equals 2(0.2500 sec + 0.1336 sec) or 0.7672 seconds. Time spent in the constant acceleration period is (2(0.1336))/0.7672) or 34.8% of the entire phase.

REFERENCE 4: HOMING AN IANG1(E)

Introduction

This chapter explains the various ways of homing an IANG1(E). Inputs used to home the unit are introduced and diagrams that show how the unit responds to a homing command are given.

Definition of Home Position

The Home Position is any position on your machine that you can sense and stop at. Once at the Home Position, the motor position register of the IANG1(E) must be set to an appropriate value. If you use the module's *CW/CCW Find Home* commands, the motor position register will automatically be set to zero once the home position is reached. *The Encoder Position register will also be reset to zero if the encoder is available and enabled.*

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

With the exception of Absolute Moves, the IANG1(E) can still perform all of its move commands if the Home Position is not defined.

Preset Position

One of the ways to define the Home Position is to issue the Preset Position command to the IANG1(E). Before doing this, you will need a way of sensing position outside the IANG1(E) module. The machine position data must be brought into the F6CA, the correct preset value calculated, and this value written to the IANG1(E) with the Preset Position command. The motor and encoder position values can be preset anywhere in the range of -8,388,608 to +8,388,607.

CW/CCW Find Home Commands

The other choice is to use the driver's Find Home commands to order the IANG1(E) to find the Home Position based on sensors brought into the unit. The CW Find Home command begins searching by rotating the motor shaft in the clockwise direction and ends when the home sensor triggers while the IANG1(E) is rotating in the clockwise direction *at the starting speed*. The CCW Find Home command operates in the same way but starts and ends with motion in the counter-clockwise direction.

Homing Inputs

Five inputs can be used when homing the module. Four are physical inputs attached to the module and one is a bit in the 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 IANG1(E) 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

- **Backplane_Proximity_Bit:** The IANG1(E) can be configured to ignore changes on the physical homing input until the Backplane_Proximity_Bit makes a 0→1 transition. The IANG1(E) will home on the next inactive-to-active change on the physical input once this transition occurs. You must program your F6CA to control the state of this bit.

> **Note➤** This bit is not used by default, and must be activated when you configure the IANG1(E) before is can be used. If you decide to activate this bit when you configure the unit and then never set the Backplane_Proximity_Bit to a "1" while homing the IANG1(E), the unit will never act on the physical homing input and the homing routine will fail.

Homing Configurations

The IANG1(E) axis must be correctly configured before one of the homing commands will be accepted. One of the following must be part of the module's configuration before you can run the homing commands.

1) Configure one of the DC inputs as a Home Input
2) Configure the IANG1(E) 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 unit this way, then the IANG1(E) has no way to automatically prevent over travel during a homing operation. In linear applications, you must prevent over travel by some external means, or ensure that the homing command is issued in the direction that will result in reaching the homing input directly.
2) You can use a bit in the Network Output Data (the Backplane_Proximity_Bit) as a home proximity input. Using this bit is completely optional and prevents the Home Input from being acted upon until the Backplane_Proximity_Bit makes a 0→1 transition.
3) When using an encoder's Z-pulse as the homing sensor, any DC input configured as a Home Input will function as a hardware home proximity sensor. Using this feature is completely optional.

Homing Profiles

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

Home Input Only Profile

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

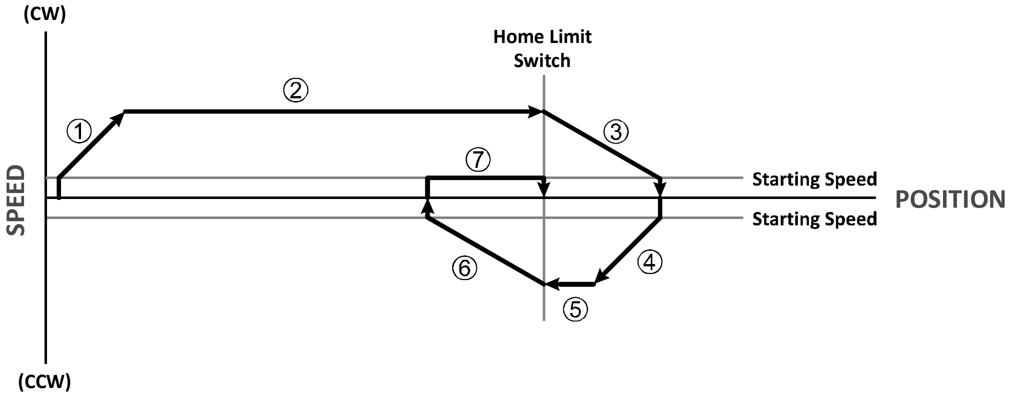


Figure R4.1 Home Input Profile

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

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

Profile with Proximity Input

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

- the Home Input with the Backplane_Proximity_Bit bit
- the Marker Pulse home with the Home Input as the proximity sensor
- the Marker Pulse home with the Backplane_Proximity_Bit bit

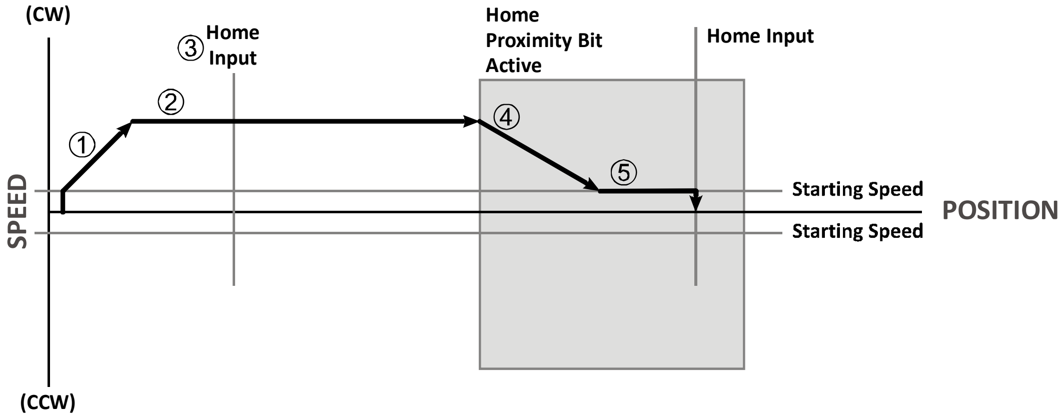


Figure R4.2 Homing with Proximity

1) Acceleration from the configured Starting Speed to the Programmed Speed
2) Run at the Programmed Speed
3) Ignores the Home Input because Backplane_Proximity_Bit has not made a 0→1 transition.
4) Deceleration towards the Starting Speed when the Backplane_Proximity_Bit transitions from 0 to 1. The axis will stop as soon as the Home Input becomes active.
5) The Starting Speed is the minimum speed the profile will run at. If the axis decelerates to the Starting Speed before reaching the Home Input, it will continue at this speed.

Note➤ Figure R3.2 shows the Backplane_Proximity_Bit staying active until the IANG1(E) reaches its home position. This is valid, but does not have to occur. As stated in step 4, the IANG1(E) starts to hunt for the home position as soon as the Backplane_Proximity_Bit makes a 0→1 transition.

Profile with Overtravel Limit

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

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

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

The IANG1(E) will stop the axis with an error if both overtravel limits are activated while the unit is trying to find the home position.

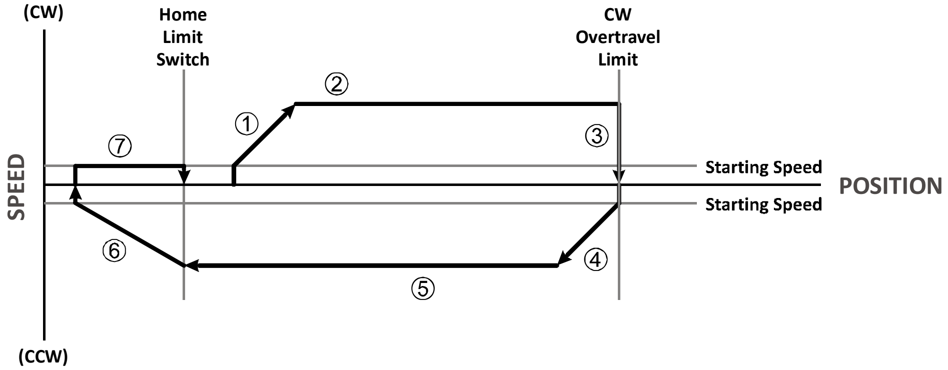


Figure R4.3 Profile with Overtravel Limit

1) Acceleration from the configured Starting Speed to the Programmed Speed
2) Run at the Programmed Speed

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

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

Controlling Find Home Commands In Progress

Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold_Move control bit in the Network Output Data. This will abort the command and the axis will decelerate at the programmed rate until it reaches the Starting Speed. At this point, the motor will stop. Note that Find Home commands cannot be restarted once held.

Immediate Stop Conditions

- The Immediate Stop bit makes a 0→1 transition in the Network Output Data.
- An inactive-to-active transition on an input configured as an E-Stop Input.
- The overtravel limit associated with travel in the opposite direction is activated. i.e. Activating the CCW limit during a CW Find Home command. This can occur if the overtravel limits are not wired to the IANG1(E) correctly, or if both overtravel limits are activated while the unit is trying to find the home position.

TASK 1: INSTALLING THE IANG1(E)

Introduction

The IANG1E can be installed as a single device. An IANG1 module must be installed as part of an AnyNET-I/O Stack, which includes an IANG1E or IANF2E as the network connection for the AnyNET-I/O Stack.

1.1 Safe Handling Guidelines

1.1.1 Prevent Electrostatic Damage

⚠ **Caution** Electrostatic discharge can damage the IANG1(E). 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 unit in its anti-static bag and shipping box when it is not in use.

1.1.2 Prevent Debris From Entering the Unit

Warning While mounting 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 unit or unintended machine operation with possible personal injury.

1.1.3 Remove Power Before Servicing

Warning Remove power before removing or installing any IANG1(E) module. The module interconnect bus is not power limited.

1.2 Mounting

1.2.1 Dimensions

Figure T1.1 shows the dimensions of an IANG1(E) module.

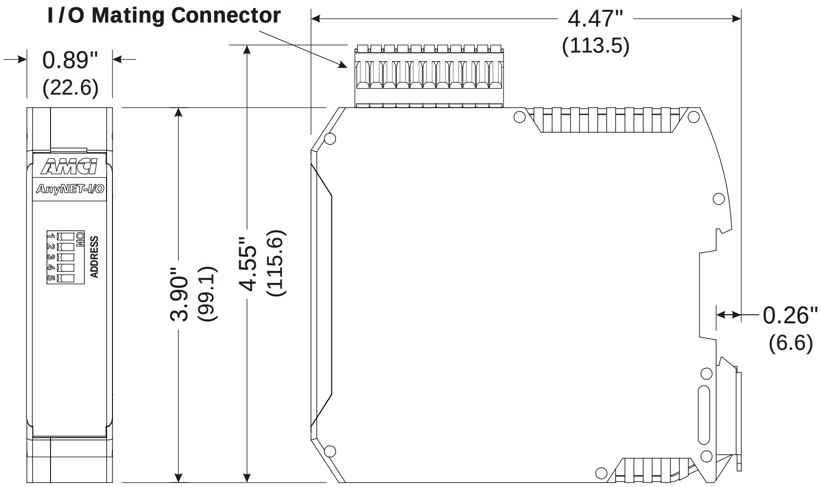


Figure T1.1 AnyNET-I/O Outline

1.2.2 Minimum Spacing

As shown in figure T1.2, you must maintain a minimum spacing of 2 inches (50.8 millimeters) from enclosure walls, wireways, adjacent equipment, etc. for adequate system ventilation.

Also note that the 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 IANG1(E) modules that are all running at full 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 IIC-5 connector between each module to space them out. Information on installing the IIC-5 connectors can be found below.

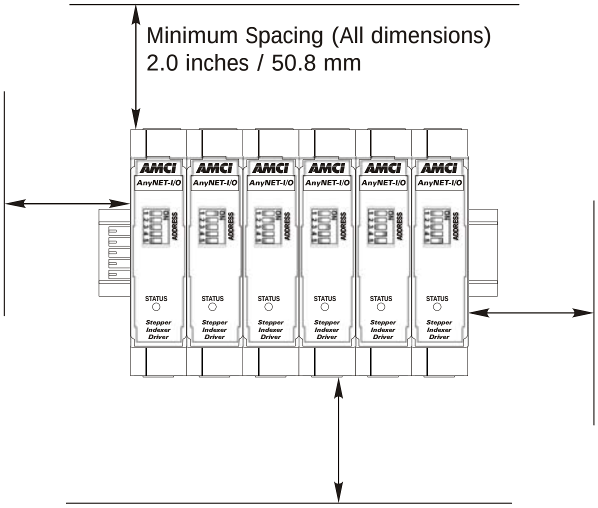


Figure T1.2 Ventilation Spacing

The IANG1(E) has a bit in the network data that signals when the module is close to overheating. This bit is the Temperature_Above_90°C bit, and is available in Status Word 1 of the network input data while in Command Mode. This bit is further explained in the *Status Word 1 Format* section starting on page 74.

1.2.3 DIN Rail Installation

The ANG1(E) module is designed to be mounted on DIN rail. EN 05 022 - 35 x 7.5 DIN rail is preferred. EN 05 022 - 35 x 15 DIN rail can be used if mounting a single ANG1E module. The DIN rail for the modules should be securely installed and grounded before the modules are mounted on it.

1.2.4 Installing IC-5 Connectors (as needed)

If you are installing a stack of modules instead of a single IANG1E, then you need to install the IIC-5 connectors on the DIN rail to allow the modules in the stack to communicate. Figure T1.3 shows how to install the IIC-5 connectors in the DIN rail.

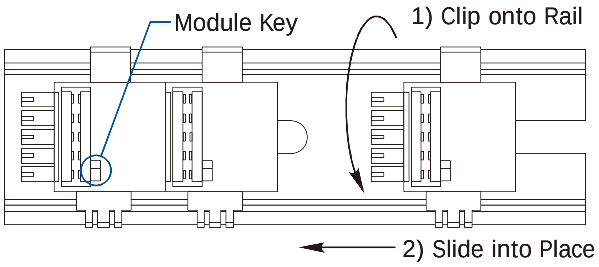


Figure T1.3 IIC-5 Connector Installation

Note► 1) EN 05 022 - 35 x 7.5 DIN rail must be used. The IIC-5 connectors are not properly supported in EN 05 022 - 35 x 15 DIN rail.

2) Note the orientation of the IIC-5 connectors when installing them. The module key goes towards the bottom of the DIN rail.

3) The IIC-5 connector is included with the IANG1(E). Additional connectors can be ordered from IDEC or directly from Phoenix Contact. Their Phoenix Contact part number is 271 37 22. (ME 22,5 TBUS 1,5/ 5-ST-3,81 KM)

1.2.5 Mounting the IANG1(E) Module

Mounting an AnyNET-I/O module is a very simple process thanks to the design of the enclosure.

1) Partially engage the connector into the enclosure.
2) Engage the top clip in the enclosure with the top of the DIN rail and rotate the module down 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 module from sliding along the DIN rail if it is subjected to shock or vibration during machine operation.

1.3 Addressing

Each module needs to be given an address within the stack 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 IANG1E modules to work in a single stack.

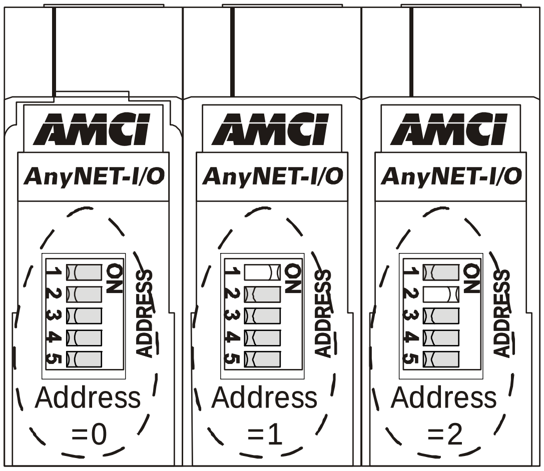


Figure T1.4 Addressing Example

Figure T1.4 is a close up of three modules in an AnyNET-I/O Stack. The module on the left is a module with a network interface, 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 *may* have network interfaces. If they do, their network interfaces are disabled.

1.4 Ethernet Connections

The Ethernet port is located on the bottom of the ANG1(E) module. The connector is a standard RJ-45 jack that will accept any standard or shielded 100baseT cable. (CAT5, CAT5e, CAT6, etc.) The Ethernet port on the module has an "auto switch" capability which eliminates the need for a crossover cable when directly connecting the module to a PC. A standard Ethernet cable can be used when connecting the ANG1(E) to any device.

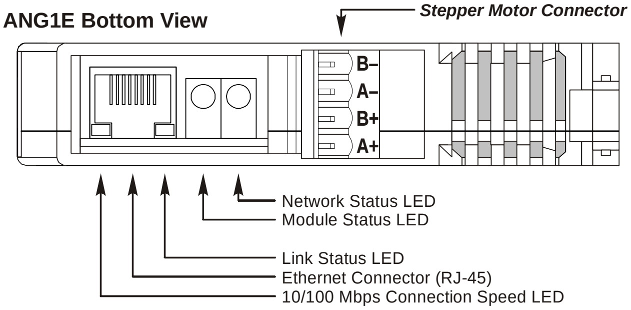


Figure R4.4 Ethernet Port Location

1.5 Power Wiring

The IANG1(E) accepts 24 to 48 Vdc as its input power. Each module has two connections on the IMS-2X11 I/O connector for the power supply. These connections are internally connected together. The suggested wire gauge for power supply connections is 16 to 18 AWG, which is equivalent to 1.31 to 0.82 mm^2. The maximum wire size that is accepted by the IMS-2X11 is 1.5 mm^2. 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 IMS-2X11 if you daisy chain connections. If this possibility exists in your installation, run separate wires from your power supply to each module.

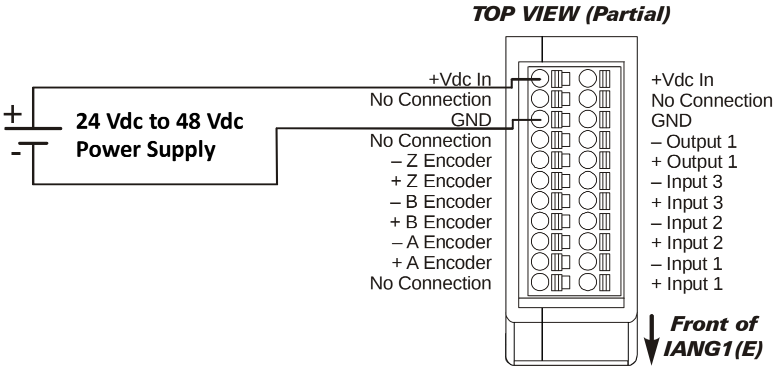


Figure T1.5 Power Supply Wiring

1.6 Input Wiring

Inputs 1, 2, and 3 are optoisolated differential inputs that can be wired directly to a sinking or sourcing sensor without the need of a pull up resistor. They accept 3.5 to 27 Vdc without the need for an external current limiting resistor. Figure T1.6 below shows how to wire discrete DC sourcing or sinking sensors to any of the three inputs of the IANG1(E).

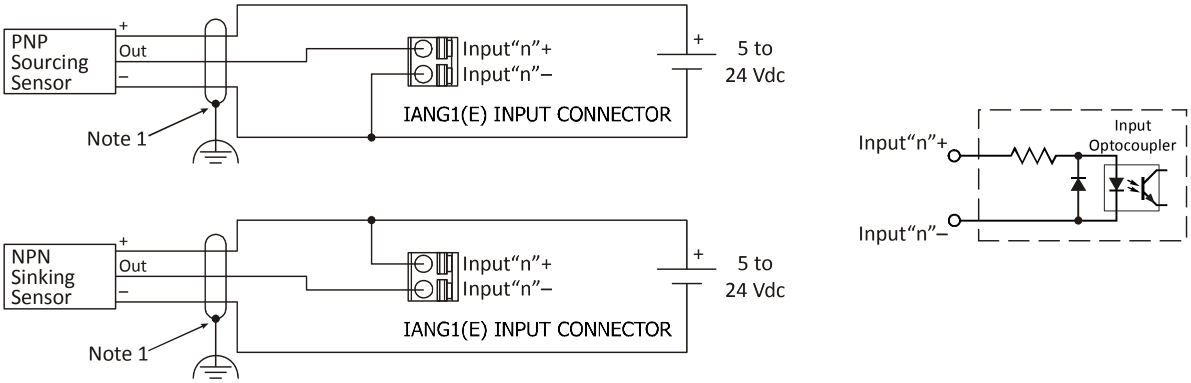


Figure T1.6 IANG1(E) Input Wiring and Schematic

Note➤ 1) Because they are low power signals, cabling from a sensor to the IANG1(E) should be made 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 IANG1(E). The shield should not be grounded at both ends of the cable to avoid the potential of forming a ground loop.

1.7 Output Wiring

The output on the ANG1(E) 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.

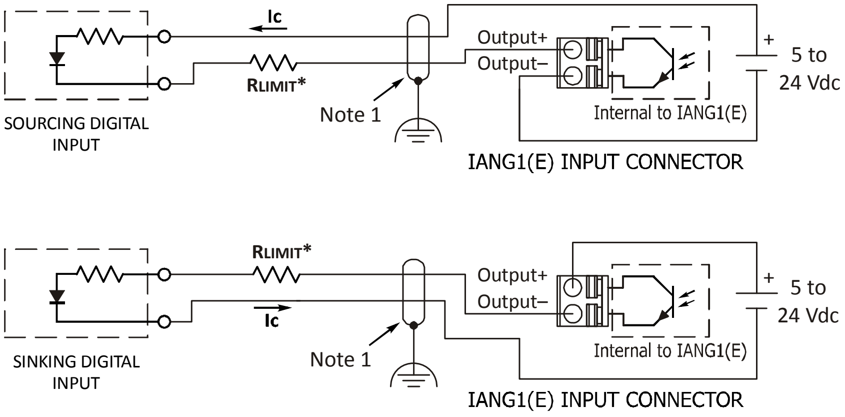


Figure T1.7 IANG1(E) Input Wiring and Schematic

Note► 1) Because they are low power signals, cabling from a sensor to the IANG1(E) should be made 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 IANG1(E). The shield should not be grounded at both ends of the cable to avoid the potential of forming a ground loop.

1.7.1 Electrical Specifications

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

Figure T1.8 Output Electrical Specifications

*1.7.2 RLIMIT**

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

1.8 Encoder Wiring

1.8.1 Differential Wiring

The figure below shows how to wire a 5 Vdc differential encoder to the IANG1. 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) that lists the color codes of encoders used by AMCI. It is available in the ‘PDF Documents’ section of the website.

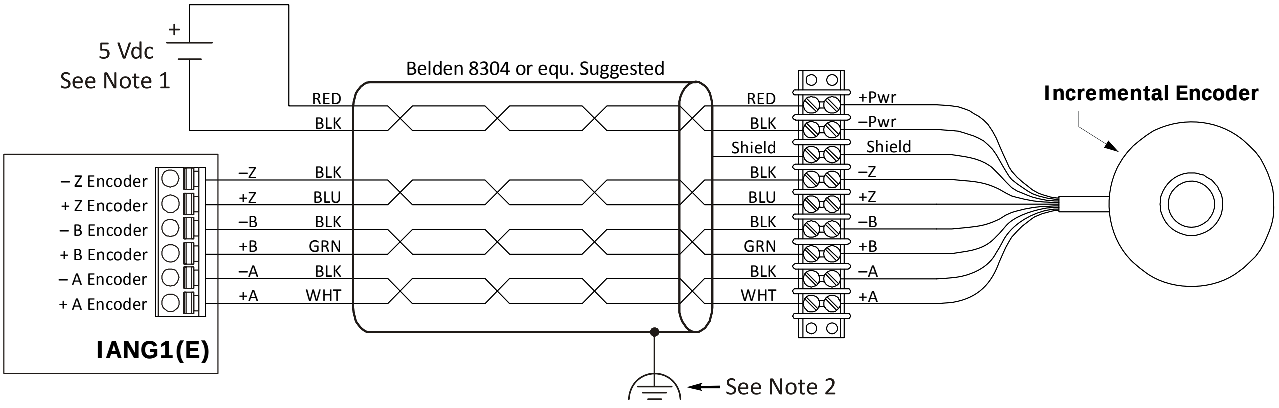


Figure T1.9 Sample Differential Encoder Wiring

Note➤ 1) A 5 Vdc supply is shown because the encoder inputs on the IANG1(E) cannot exceed 5 Vdc. If the encoder regulates its outputs to 5Vdc, any supply can be used to power the encoder.

2) Because they are low power signals, cabling from an encoder to the IANG1(E) should be made using a twisted pair cable with an overall shield. The shield should be grounded at the end when the signal is generated, which is at the encoder. If this is not practical, the shield should be grounded to the same ground bus as the IANG1(E). The shield should not be grounded at both ends of the cable to avoid the potential of forming a ground loop.

1.8.2 Single Ended Wiring

Figure T1.10 below shows how to wire the encoder inputs to both a single ended sourcing and single ended sinking encoder outputs.

⚠ **Caution** The encoder inputs on the IANG1(E) are rated for 5 Vdc only. You must use a current limiting resistor on each input if the outputs of your encoder are greater than 5 Vdc. Failure to use these resistors will result in damage to the IANG1(E). Appropriate current limiting resistors are shown below.

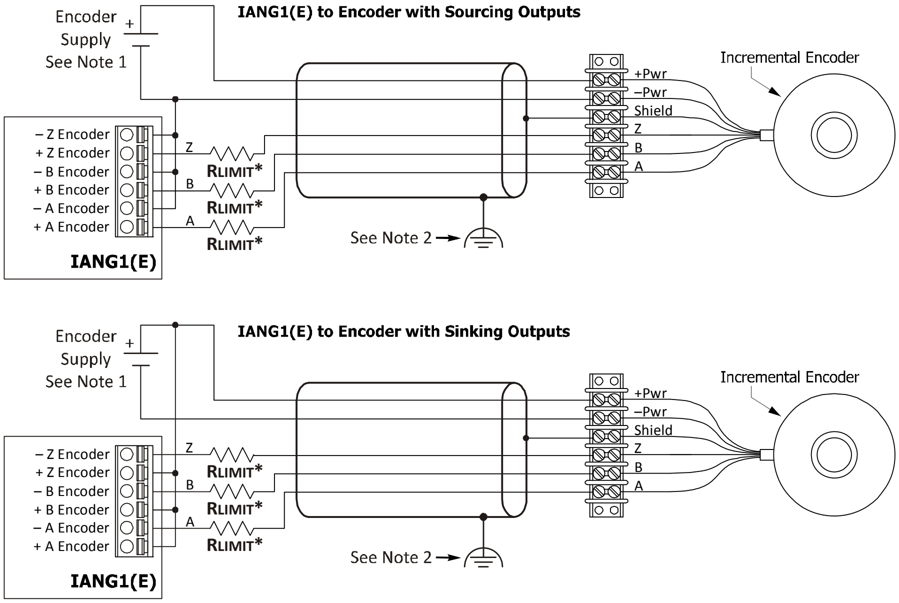


Figure T1.10 Single Ended Encoder Wiring

RLIMIT Values

The following table lists the resistor values that are required when the voltage on the output of the encoder exceed 5 Vdc.

Encoder Output Voltage	RLIMIT Value	RLIMIT Power
5 Vdc	None	–
12 Vdc	2.0 kohm	0.25 Watt
15 Vdc	2.4 kohm	0.25 Watt
24 Vdc	4.3 kohm	0.25 Watt

1.9 Installing the Stepper Motor

1.9.1 Outline Drawings

At the time this manual was released, outline drawings for all AMCI by IDEC Corporation motors could be found at the following website address:

http://us.idec.com/Catalog/ProductSeries.aspx?SeriesName=Stepper_Motor&FamilyName=QT_AMCI_Motion_Controls

Both 2D DWG files and 3D STEP files are available.

1.9.2 Mounting the Motor

All IDEC motors 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. IDEC motors can produce high torques and accelerations that may weaken and shear inadequate mounting hardware.

Note➤ 1) The motor case must be grounded for proper operation. This is usually accomplished through its mounting hardware. If you suspect a problem with your installation, such as mounting the motor to a painted surface, then run a bonding wire from the motor to a solid earth ground point near it. Use a minimum #8 gauge stranded wire or 1/2" wire braid as the grounding wire.

2) Do not disassemble *any* stepper motor. A significant reduction in motor performance will result.

1.9.3 Connecting the Load

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

1.9.4 Extending the Motor Cable

Even though it is possible to extend the cable length an additional forty feet, IDEC recommends installing the IANG1(E) 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.

1.9.5 Installing the Motor Cable

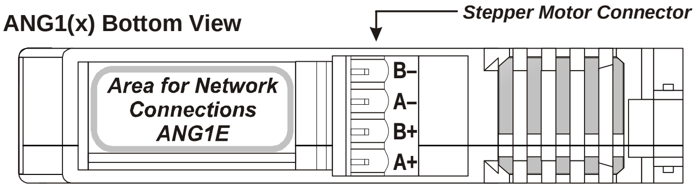
Note➤ 1) All of the motor connections are high power, high voltage signals. Cable from the motor can be installed in conduit along with ac/dc power lines or high power ac/dc I/O. It cannot be installed in conduit with low power cabling such as I/O cabling or Ethernet cabling attached to the IANG1(E).

2) If you decide to extend the motor cable, treat the shield as a signal carrying conductor when installing the motor cable. Do not connect the shield to earth ground at any junction box.

1.10 Connecting the Motor

1.10.1 Motor Connector

The motor connector is included with the IANG1(E). Spares are available from IDEC under the part number IMS-4M as well as directly from Phoenix Contact under their part number 187 80 37. Motor connections should be tight, as loose connections may lead to arcing which will heat the connector. Phoenix Contact specifies a tightening torque of 4.4 to 5.4 lb-in (0.5 to 0.6 Nm)



Warning When powered, the full DC input voltage may be present on the terminals of the motor connector. This voltage may represent a shock hazard. Always remove power from the IANG1(E) 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 at all possible. If the connection from the DIN rail to the Grounding Bus fails over time, then you will eventually have a condition where electrical noise will be injected into the AnyNET-I/O modules, which may result in future system errors.

1.10.2 Motor Wiring

The IANG1(E) will work with many different motors, including those not sold by IDEC. 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’s data sheets for this information.

The following three figures show how to wire a motor to the IANG1(E) in series, parallel, and center-tap configurations. Refer to the torque vs. speed curves on your motor’s specifications sheet to determine how you should wire your motor to the IANG1(E).

Note► All IDEC ISMD23 motors are eight lead motors. Torque curves and current settings are for parallel connected motors.

Note► You can reverse the direction of rotation in a stepper motor by reversing the ±B wire connections. If this is done, moves that are listed as “CW” in this manual and the IDEC custom macros will generate counter-clockwise rotation in the motor shaft.

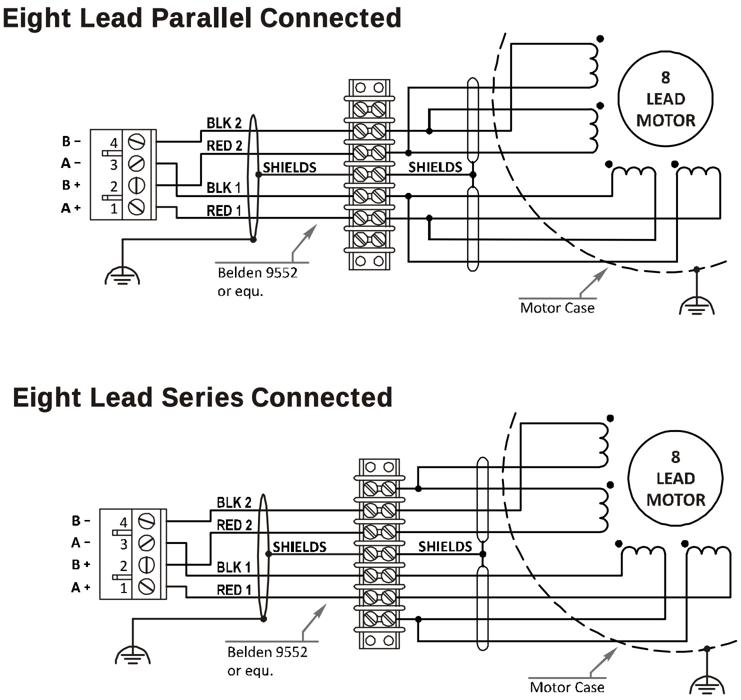
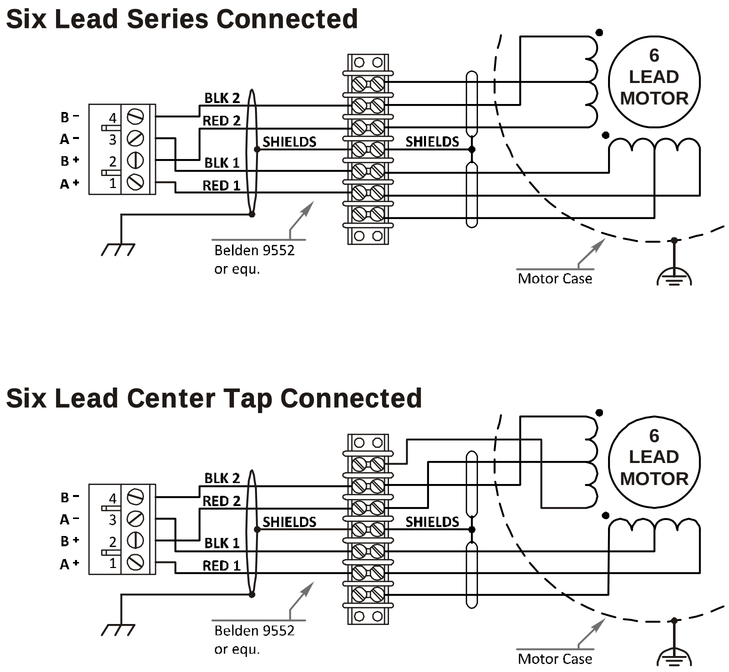


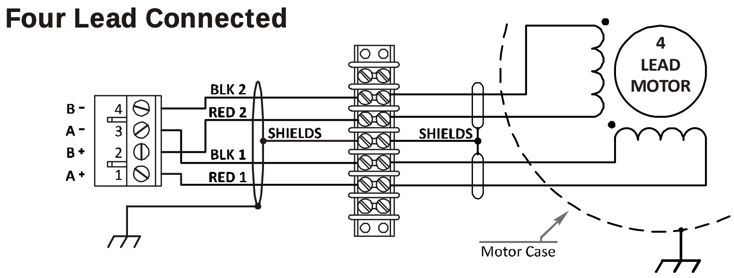
Figure T1.11 Eight Lead Motor Connections



Six Lead Series Connected

Six Lead Center Tap Connected

Figure T1.12 Six Lead Motor Connections



Four Lead Connected

Figure T1.13 Four Lead Motor Connection

Notes

TASK 2: SET THE IP ADDRESS

Introduction

This section is intended for the engineer or technician responsible for setting the IP address of an IANG1E.

2.1 Determine the Best Method for Setting the IP Address

There are two methods for setting the IP address on an IANG1E. Table T2.1 below outlines the available methods and when you can use them.

Method	Restrictions	Starting page
Use Factory Default Settings	1) The machine must use 192.168.1.xxx subnet. 2) The 192.168.1.50 address must be available.	51
Use the AMCI by IDEC Ethernet Tool	No restrictions on use. The software can be used to set the IANG1E to any IPv4 address. The IP address will be stored in nonvolatile memory and used on subsequent power-ups.	51

Table T2.1 Methods for Setting the IP Address

> **Note➤** There is a MAC address label on each IANG1E which has a writable surface. There is room on the label for writing the programmed IP address of the unit. It is a best practice to use this label to document the IP address of the unit in case it is ever repurposed.

2.2a Use Factory Default Settings

The factory default address for the IANG1E is 192.168.1.50 with a subnet mask of 255.255.255.0. The easiest way to verify this address is with the ping command as described in steps A.3 and A.4 of the Optional Task *Assembled Moves*. The instructions for this optional task starts on page 95.

If the IANG1E does not respond to this address then it may take some effort to determine the correct address. There is a label on the drive that lists the MAC address of the device. There is space on the label for noting the IP address of the device if it is changed. If the address was not documented, a program called Wireshark (*https://www.wireshark.org/*) can be used to determine the address of the IANG1E.

Task Complete

The remainder of this chapter can be skipped, as it contains alternate methods of setting the IP address of an IANG1E.

2.2b Use the AMCI by IDEC Ethernet Tool

PREREQUISITE: You must know the present IP address. The factory default address is 192.168.1.50.

PREREQUISITE: Task 1.4: *Ethernet Connections* found on page 43. You must be able to attach the IANG1E to your computer.

PREREQUISITE: Task 1.5: *Power Wiring* found on page 43. You must be able to power the IANG1E.

PREREQUISITE: Optional Task A: *Assembled Moves*. (page 95) The network interfaces on your computer must be correctly configured before you can communicate with an IANG1(E).

2.2b.1 If need be, download the AMCI by IDEC Ethernet Tool

1) Download link: *http://us.idec.com/productdocuments/english/Other/AMCI-IDEC-Configurator_1_2.zip*
2) Unzip the file to any location on your computer.

2.2b.2 If need be, install the AMCI by IDEC Ethernet Tool

Once downloaded, simply extract the program from the ZIP file and run the program to install the AMCI by IDEC Ethernet software tool on your computer. The software installs as most products do, giving you the option to change the file locations before installing the utility. Once the install is complete, a link to the utility is available on the Start Menu. The install process only copies the utility to the designated location and creates links to the Start Menu. No changes are made to the registry settings of the computer.

2.2b.3 Verify that Your IDEC Controller is Disconnected from the IANG1E

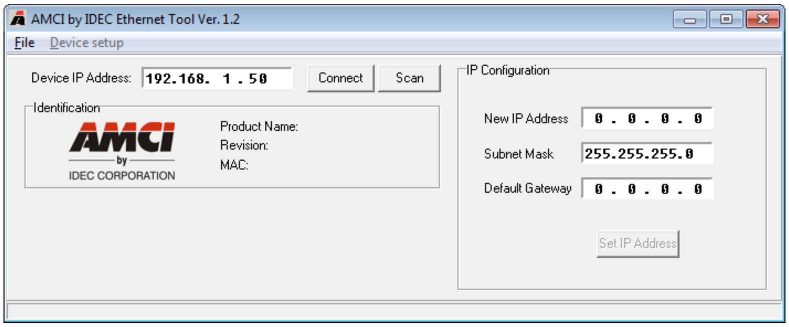
Successfully changing the IP address of the IANG1E is fundamental to the operation of the unit. While performing this operation, the computer that is running the Ethernet tool should be the only device that can establish communications with the IANG1E.

2.2b.4 Apply or Cycle Power to the IANG1E

Cycling power to the IANG1E will reset any connections it may have had with the IDEC host controller.

2.2b.5 Start the AMCI by IDEC Ethernet Tool

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



Device IP Address: 192.168. 1 . 50

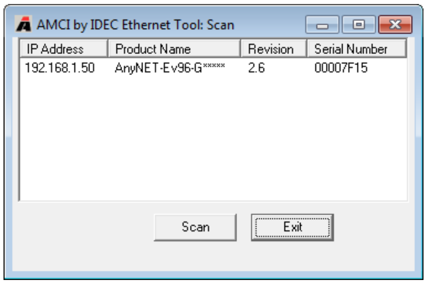
Product Name:
Revision:
MAC:

New IP Address: 0 . 0 . 0 . 0
Subnet Mask: 255.255.255.0
Default Gateway: 0 . 0 . 0 . 0

Figure T2.1 Ethernet Tool Welcome Screen

2.2b.6 Press the [SCAN] button and Connect to the IANG1E

Pressing the [Scan] button will open the window shown in figure T2.2. The IANG1E will appear in the scan list after a few seconds only if the unit and your network interface are on the same subnet.



IP Address	Product Name	Revision	Serial Number
192.168.1.50	AnyNET-Ev96-G*****	2.6	00007F15

Figure T2.2 Scan for IANG1E

Once the IANG1E appears in the list, double click on the IP address. The Ethernet Tool will connect to the unit.

Optionally, from the main screen, you can enter the IP address of the IANG1E and press the [Connect] button. If the unit is found, the Ethernet Tool will directly connect with the unit.

2.2b.7 Set the IP Address, Subnet Mask, and Default Gateway

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

Note► The Default Gateway setting is not optional! It must be set to a valid address on the chosen subnet. If you do not have a required value for it, set the Default Gateway to the IP address of your FC6A controller.

2.2b.8 Write the New IP Address to the IANG1E

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

2.2b.9 Remove Power from the IANG1E

The new IP address will not be used until power to the IANG1E has been cycled.

Task Complete

TASK 3: WINDLDR 8.5 PROJECT SETUP

Introduction

The following task adds a module stack to a WindLDR 8.5+ project. (WindLDR versions 8.5+ is required.) A module stack consists of an IANG1E and zero to five additional IANG1 modules. This involves adding the IANG1E to the Remote Host List and then configure Modbus TCP connection requests.

> **Note➤** Before adding an IANG1E to a new project, the controller must be defined. Refer to IDEC manuals and help files for information on accomplishing this task.

3.1 Add the IANG1E to the Remote Host List

3.1.1 Bring up the project window in the Work Space area on the left of the window.

If need be, select the View Tab, and verify that the Project Window icon is enabled. (Property Sheet and Cross Reference are disabled in figure T3.1 below)

Select the Project Window tab in the bottom of the Work Space area.

3.1.2 Double click on the Remote Host List in the Project Window to open the Remote Host List screen.

3.1.3 Click on the [New] button to open the Remote Host screen.

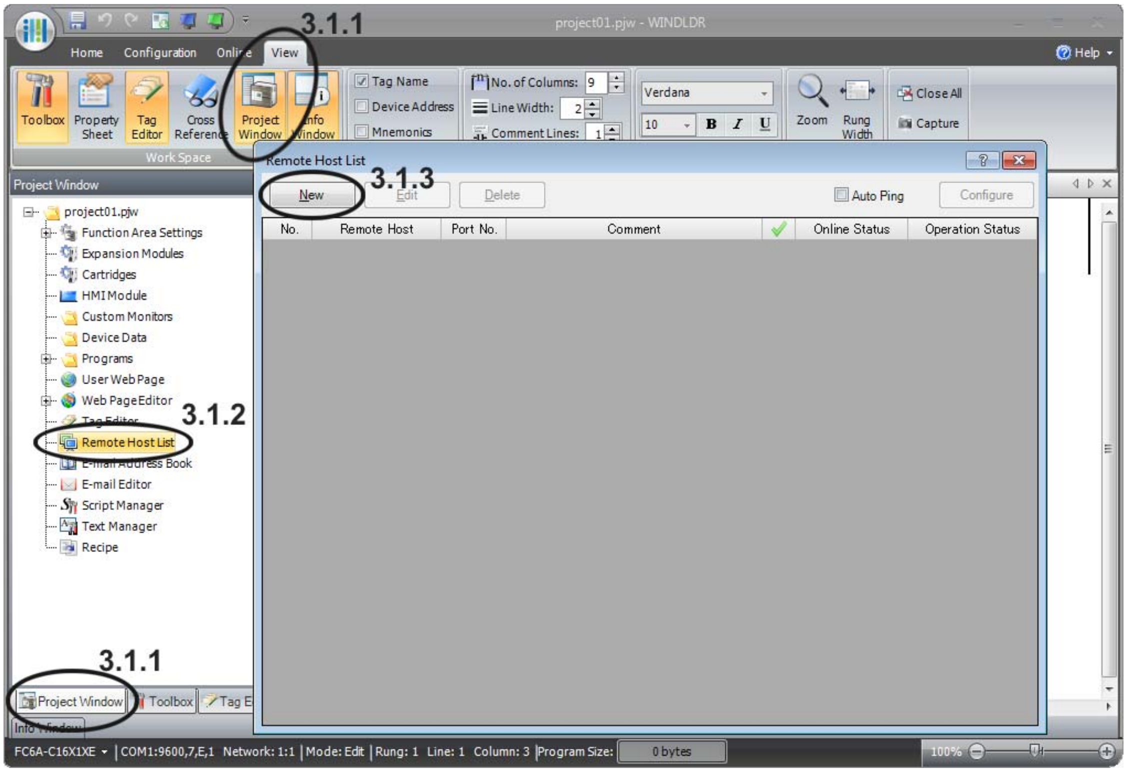


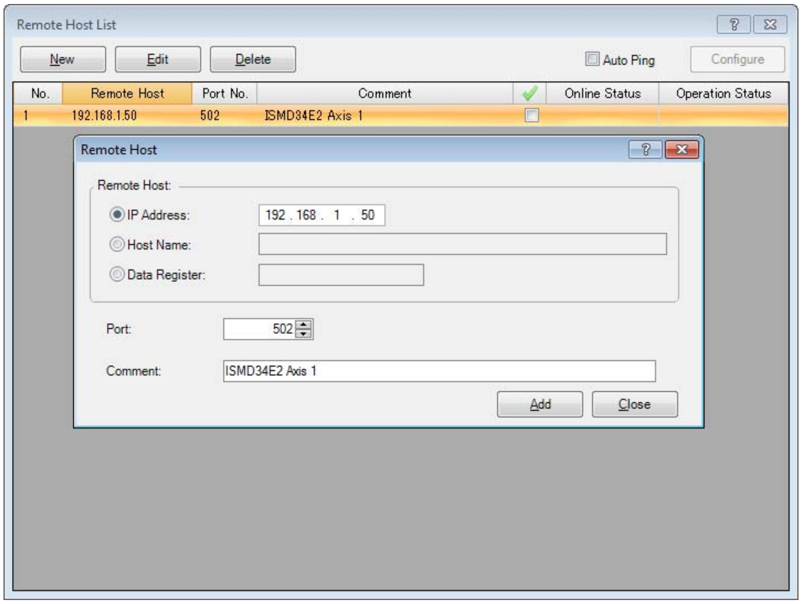
Figure T3.1 WindLDR Remote Host List

3.1.4 Enter the IP Address Information.

Enter the IP Address of the IANG1E. The default address is 192.168.1.50. You can enter a Host Name if you have configured a DNS server for the IDEC controller, which is beyond the scope of this manual.

Set the *Port:* field to 502, which is the default port used by the Modbus TCP protocol

You can add a *Comment:*, such as the name of the IANG1E stack in the project. This step is optional.



No.	Remote Host	Port No.	Comment	✓	Online Status	Operation Status
1	192.168.1.50	502	ISMD34E2 Axis 1	□		

● IP Address: 192 . 168 . 1 . 50

Port: 502

Comment: ISMD34E2 Axis 1

Figure T3.2 Remote Host List Entry Screen

3.1.5 Click on the [Add] button to add the IANG1E to the Remote Host List.

3.1.6 Repeat steps 3.1.3 and 3.1.4 for each IANG1E stack that must be added to the project. Each stack can consist of up to six modules.

3.1.7 Click on the [Close] button to close the Remote Host screen.

3.1.8 Close the Remote Host List screen.

3.2 Configure the Connection Settings for the IANG1E Stack

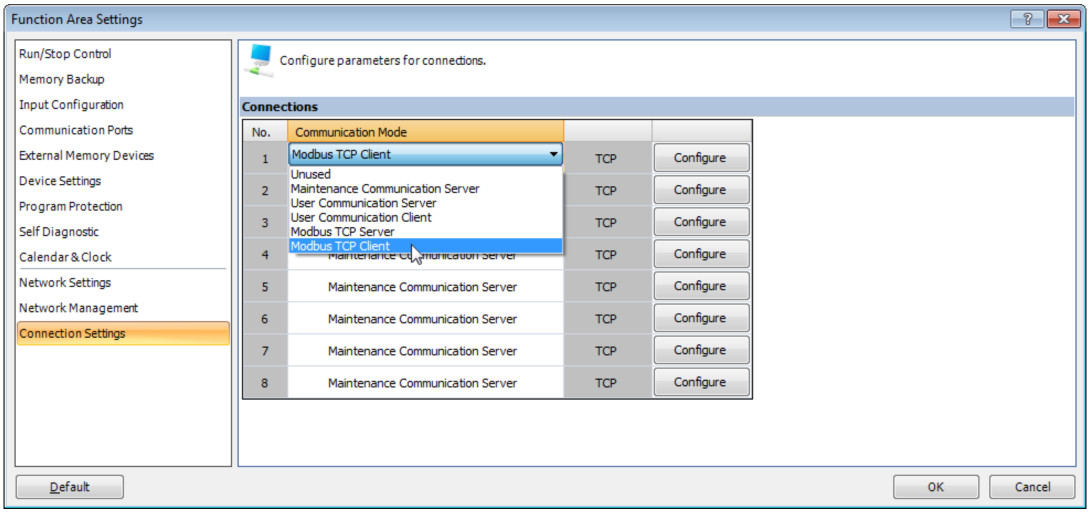
FC6A processors support a maximum of eight communication channels on the Ethernet port. At least one of the communications channels must be configured to service “Modbus TCP Client” communications.

3.2.1 From the WindLDR menu, select Configuration > Connection Settings.

The Function Area Settings dialog box will open, with the Connection Settings view selected. The eight communication channels are shown. Note that the default Communication Mode for all channels is “Maintenance Communication Server”.

3.2.2 Set the Communications Mode for the selected channel to "Modbus TCP Client"

Click on the Communications Mode cell of the selected port. A drop down menu will appear. (See figure T3.3 below.) In this menu, select "Modbus TCP Client". The Modbus TCP Client dialog box will appear.



Function Area Settings

Configure parameters for connections.

Connections

No.	Communication Mode		
1	Modbus TCP Client	TCP	Configure
2		TCP	Configure
3		TCP	Configure
4	Maintenance Communication Server	TCP	Configure
5	Maintenance Communication Server	TCP	Configure
6	Maintenance Communication Server	TCP	Configure
7	Maintenance Communication Server	TCP	Configure
8	Maintenance Communication Server	TCP	Configure

Figure T3.3 Connection Settings

3.2.3 Configure the Modbus TCP Client Requests

There are three setting areas that affect the communication channel as a whole.

Request Execution Settings: The "Request Execution Device" checkbox is typically left unchecked so that the FC6A communicates with the IANG1E stack continuously. If checked, communications is controlled by internal relays or data register bits. If your application requires you to enable this option, refer to the FC6A help file or the FC6A Series Communications Manual for additional information. The "Synchronize with auto ping" checkbox is typically left unchecked. If your application requires you to enable this option, refer to the FC6A help file or the FC6A Series Communications Manual for additional information.

Error Status: Error status information should be collected on each communication request. Select the "Use" radio button and set a data register address in the text box. All IDEC sample programs use register D49240 as the first error status register. The "Use a single DR for all communication requests" checkbox should be left unchecked so that error data on each request is collected individually. The number of data registers consumed will be equal to the number of requests on the channel, starting at the specified address. The "Update error status only when communication fails" checkbox is left unchecked in all IDEC sample programs. This setting allows the sample programs to act when communications is established.

Communication Settings: If needed, click on the [Communication Settings] button. A dialog box will open that allows you to set the Receive Timeout and Transmission Wait Time values. These setting are application specific and depend on network loading. Default values have been tested by AMCI and IDEC. Click on the [OK] button to close the dialog box.

In addition to these settings, two requests are required to communicate with each IANG1E stack. One request reads data from the stack and the other writes data to the stack. Each stack, which also can be considered to be each IP address, requires these two requests. For example, if you have two stacks on your machine, you will be configuring four requests.

Read Request: Set the following values to read data from the IANG1E stack.

Function Code: Click inside the cell and select "04 Read Input Registers".

Master Device Address: Depends on the first axis being controlled by the stack, which is the motor controlled by the module that connects the stack to the network. (IANG1E or IANF2E) See table T3.1 below to determine the correct starting address.

Machine Axis	Starting Data Register Address	Machine Axis	Starting Data Register Address
1	D49000	7	D49060
2	D49010	8	D49070
3	D49020	9	D49080
4	D49030	10	D49090
5	D49040	11	D49100
6	D49050	12	D49110

Table T3.1 Master Device Addresses: Read Input Registers

Data Size: Depends on the number of axes controlled by the AnyNET-I/O stack. Each IANG1(E) module controls a single axis. Each IANF2(E) module controls two axes. The Data Size must be set to (10 * number of axes in the AnyNET-I/O Stack.) For example, a stack of an IANG1E and three IANG1 modules controls a total of four axes. In this example the Data Size must be set to 40.

Remote Host No.: Click inside the cell to change it into a drop down menu control. Click on the down arrow and select the proper device that was configured in step 3.1, *Add the IANG1E to the Remote Host List*, above.

Slave Number: Not used by the AnyNET-I/O Stack. Set to "1".

Modbus Slave Address: 300001

The remaining fields are filled in automatically

Write Request: Set the following values to write values to the AnyNET-I/O Stack.

Function Code: Click inside the cell and select "16 Preset Multiple Registers".

Master Device Address: Depends on the first axis being controlled by the stack, which is the motor controlled by the module that connects the stack to the network. (IANG1E or IANF2E) See table T3.1 below to determine the correct address.

Machine Axis	Starting Data Register Address	Machine Axis	Starting Data Register Address
1	D49120	7	D49180
2	D49130	8	D49190
3	D49140	9	D49200
4	D49150	10	D49210
5	D49160	11	D49220
6	D49170	12	D49230

Table T3.2 Master Device Addresses: Read Input Registers

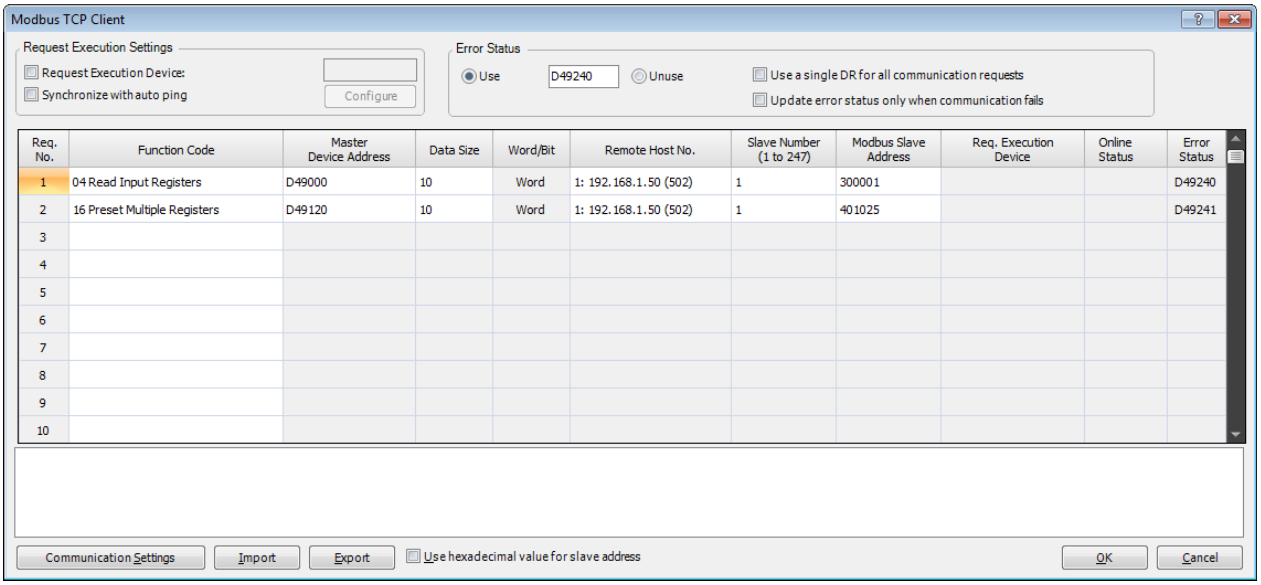
Data Size: Depends on the number of axes controlled by the AnyNET-I/O stack. Each IANG1(E) module controls a single axis. IANF2(E) modules control two axes. The Data Size must be set to (10 * number of axes in the AnyNET-I/O Stack.) For example, a stack of an IANG1E and three IANG1 modules controls a total of four axes. In this example the Data Size must be set to 40.

Remote Host No.: Click inside the cell to change it into a drop down menu control. Click on the down arrow to open the list and select the proper device that was configured in step 3.1, *Add the IANG1E to the Remote Host List*, above.

Slave Number: Not used by the AnyNET-I/O Stack. Set to "1".

Modbus Slave Address: 401025

The remaining fields are filled in automatically. Figure T3.4 below shows the Modbus TCP Client dialog box populated with a single ANG1E module as axis one of the machine.


Modbus TCP Client

Request Execution Settings
☐ Request Execution Device:
☐ Synchronize with auto ping

Error Status
● Use D49240 ○ Unuse
☐ Use a single DR for all communication requests
☐ Update error status only when communication fails

Req. No.	Function Code	Master Device Address	Data Size	Word/Bit	Remote Host No.	Slave Number (1 to 247)	Modbus Slave Address	Req. Execution Device	Online Status	Error Status
1	04 Read Input Registers	D49000	10	Word	1: 192.168.1.50 (502)	1	300001			D49240
2	16 Preset Multiple Registers	D49120	10	Word	1: 192.168.1.50 (502)	1	401025			D49241
3										
4										
5										
6										
7										
8										
9										
10										

☐ Use hexadecimal value for slave address

Figure T3.4 Modbus TCP Client Configuration Screen

Task 4: Installing Custom Macros

Introduction

IDEC has created User-defined Macros for configuring the IANG1(E) and most of its commands. These macros simplify the configuration and control of an IANG1(E). This task covers how to add these macros to a WindLDR 8.5+ project.

4.1 Download the Custom Macro File

1) Visit *http://us.idec.com/Home.aspx*
2) Click Products -> Automation -> AMCI Motion Controls
3) Select the *Drive+Controller* category
4) Click on the *Download* tab
5) Select the Download link for the *User Defined Macros - WindLDR* file.
6) Save the file to any location on your computer.

4.2 Add the Macros to a New or Existing Project

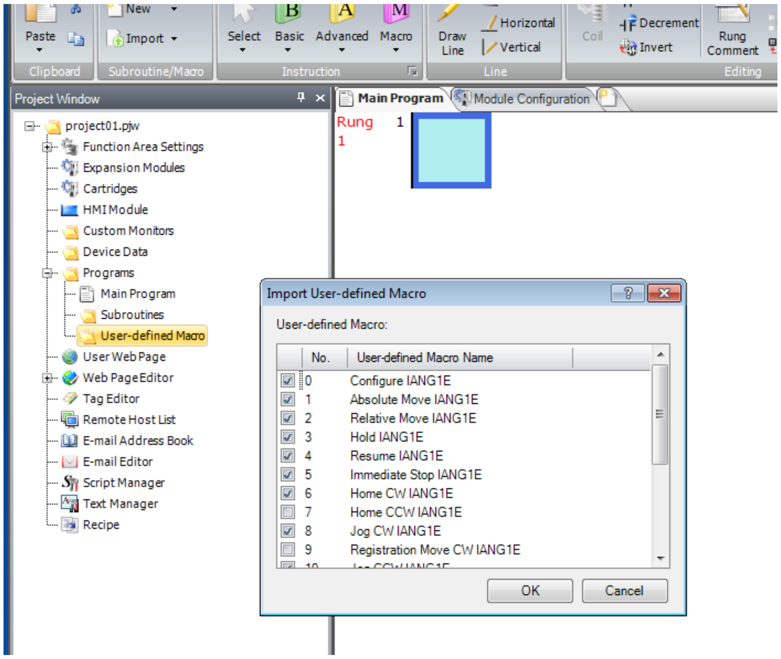
4.2.1 Open the Import User-defined Macro dialog box.

In the Project Window, if necessary, double click on Programs in order to make User-defined Macro visible. Right click on User-defined Macro and select Import from the resulting menu. In the Open dialog box, browse to the folder that contains the file you downloaded in step 4.1 above. Double click on the file name to open the Import User-defined Macro dialog box.

4.2.2 Import the macros into your program.

Click on the checkboxes next to the macros required for your project. You can import all of them if you so choose. Once the macros are selected, click on the [OK] button to import them.

> **Note➤** You can repeat the steps in 4.2 to import additional macros at any time.



User-defined Macro:

	No.	User-defined Macro Name
☑	0	Configure IANG1E
☑	1	Absolute Move IANG1E
☑	2	Relative Move IANG1E
☑	3	Hold IANG1E
☑	4	Resume IANG1E
☑	5	Immediate Stop IANG1E
☑	6	Home CW IANG1E
□	7	Home CCW IANG1E
☑	8	Jog CW IANG1E
□	9	Registration Move CW IANG1E
☑	10	Jog CCW IANG1E

Figure T4.1 Importing User-defined Macros

Notes

Task 5: Using a Custom Macro in WindLDR

Introduction

Once a User-defined Macro has been added to a project, it can be used as an instruction in a ladder logic program. This task covers how to add and configure a User-defined Macro in a ladder logic program.

5.1 Verify Memory Usage

The following address ranges are used by the IDEC supplied macros. You should check the registers currently being used in your project against this list and make adjustments as necessary. Note that registers D49000 through D49239 are the same addresses listed in the *WindLDR 8.5 Project Setup* Task starting on page 55.

Address Range			Use	Comments
D48000	to	D48999	Internal to Macros	These registers are used internally by the macros. Using registers in this range for any other purpose may result in unexpected operation.
D49000	to	D49009	Axis 1	Read Registers (Input data from the IANG1(E) modules. Updated through Modbus Function Code 04, Read Registers.) Data registers assigned to unused axes can be used for other purposes in the program.
D49010	to	D49019	Axis 2	
D49020	to	D49029	Axis 3	
D49030	to	D49039	Axis 4	
D49040	to	D49049	Axis 5	
D49050	to	D49059	Axis 6	
D49060	to	D49069	Axis 7	
D49070	to	D49079	Axis 8	
D49080	to	D49089	Axis 9	
D49090	to	D49099	Axis 10	
D49100	to	D49109	Axis 11	
D49110	to	D49119	Axis 12	
D49120	to	D49129	Axis 1	Preset Multiple Registers (Output data to the IANG1(E) modules. Updated through Modbus Function Code 16, Preset Multiple Registers.) Data registers assigned to unused axes can be used for other purposes in the program.
D49130	to	D49139	Axis 2	
D49140	to	D49149	Axis 3	
D49150	to	D49159	Axis 4	
D49160	to	D49169	Axis 5	
D49170	to	D49179	Axis 6	
D49180	to	D49189	Axis 7	
D49190	to	D49199	Axis 8	
D49200	to	D49209	Axis 9	
D49210	to	D49219	Axis 10	
D49220	to	D49229	Axis 11	
D49230	to	D49239	Axis 12	
D49240	to	D49499	Modbus TCP Communications	These registers used when monitoring and controlling Modbus TCP communications with the IANG1(E).
D49500	to	D49619	Read Registers Buffer	Memory space used to buffer the data read from the IANG1(E) modules. Any registers can be used, but all sample programs use addresses in this range. Note that this area is enough to buffer 12 axes. Unused registers can be used to store macro user data.
D49620	to	D49999	Macro User Data	Used to store data that is sent to an IANG1(E) through the macros. Any registers can be used, but all sample programs use addresses in this range.
M14500	to	M14747	Macro User Data	Addresses used to store data that is sent to an IANG1(E) through the macros. Any internal relays can be used, but all sample programs use internal relays in this range.
M14750	to	M14997	Program Control Bits	Internal relays used to store move states and control program flow. Any internal relays can be used, but all sample programs use internal relays in this range.

Table T5.1 Reserved Memory Addresses

5.2 Verify/Alter Internal Relay Keep Settings

By default, the M internal relays are cleared at startup. The M14500 to M14747 internal relays listed above are used to store configuration and command bit settings and must be maintained between power cycles. Use the Configuration > Memory Backup screen to configure the FC6A to keep previous settings at startup.

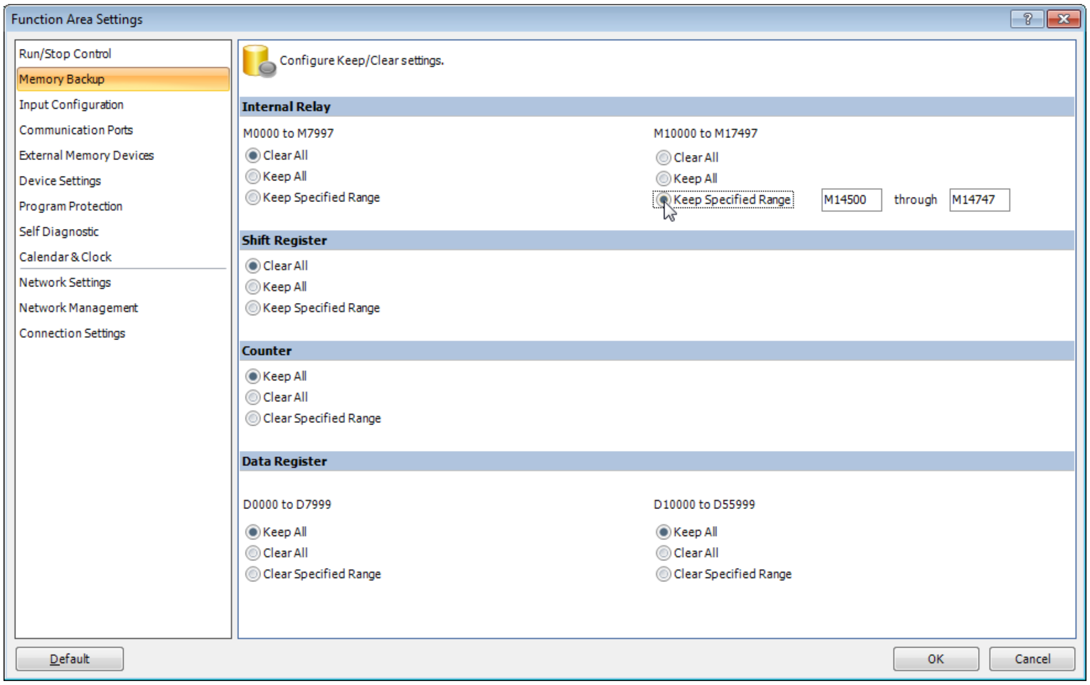
5.2.1 Open the Function Area Settings Dialog Box.

From the WindLDR menu bar, select **Configuration > Memory Backup**. The **Function Area Settings** dialog box for Configure Keep/Clear Settings appears.

5.2.2 Verify or Alter the Keep Settings.

Under the ‘M10000 to M17497’ heading of the Internal Relay section, select either the ‘Keep All’ or ‘Keep Specified Range’ radio button. If your program does not rely on internal relays being reset on startup, then you can select ‘Keep All’. If your program does rely on this behavior, select the ‘Keep Specified Range’ radio button and specify a range of M14500 to M14747.

> **Note➤** You can only specify one range when using the ‘Keep Specified Range’ feature. If you are already using this feature, adjust this setting as necessary.



Configure Keep/Clear settings.

Internal Relay

M0000 to M7997
● Clear All
○ Keep All
○ Keep Specified Range

M10000 to M17497
○ Clear All
○ Keep All
● Keep Specified Range M14500 through M14747

Shift Register
● Clear All
○ Keep All
○ Keep Specified Range

Counter
● Keep All
○ Clear All
○ Clear Specified Range

Data Register

D0000 to D7999
● Keep All
○ Clear All
○ Clear Specified Range

D10000 to D55999
● Keep All
○ Clear All
○ Clear Specified Range

Figure T5.1 Internal Relay Keep Settings

5.3 Define the Memory Map to the Macro’s Argument Devices.

Macros contain a list of arguments that are used to pass data into, and out of, the macro. While adding the macro to your ladder logic, the list of arguments must be mapped to actual Device Addresses in the processor’s memory. Sample programs available from IDEC use Device Addresses starting at register D49620 and internal relay M14500 for these purposes.

Even though it is possible to map from one register or relay to multiple macros, sample programs from IDEC use separate blocks of registers for each instance of a macro in the ladder logic program. This aids in debugging by preventing a change in value for one macro from inadvertently affecting another.

Before mapping registers and relays to their macros, tag name and comment fields can be added through the Tag Editor dialog box. This simplifies register lookup when adding them to the macro. The Tag Editor in the WindLDR program is used to add tag names and comments to the Device Addresses. Consult IDEC manuals and help files for information on using the Tag Editor.

5.3.1 Arguments to the "Configure IANG1" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	D (Data Register)	Word	0 to 6	Input 1 Function. See Table T5.3 for valid settings.
A3	D (Data Register)	Word	0 to 6	Input 2 Function. See Table T5.3 for valid settings.
A4	D (Data Register)	Word	0 to 6	Input 3 Function. See Table T5.3 for valid settings.
A5	M (Internal Relay)	Bit	0/1	Input 1 Active State 0 = NC Input, 1 = NO Input
A6	M (Internal Relay)	Bit	0/1	Input 2 Active State 0 = NC Input, 1 = NO Input
A7	M (Internal Relay)	Bit	0/1	Input 3 Active State 0 = NC Input, 1 = NO Input
A8	M (Internal Relay)	Bit	0/1	Home to Encoder Z Pulse (Encoder must be enabled, Argument A9 = 1) 0 = Disable home to Z pulse, 1 = Enable home to Z pulse
A9	M (Internal Relay)	Bit	0/1	Encoder Enable 0 = Encoder Disabled, 1 = Encoder Enabled
A10	M (Internal Relay)	Bit	0/1	Backplane_Proximity_Bit Enable 0 = Backplane Home Proximity Bit is not used 1 = Backplane Home Proximity Bit is used See *Backplane Inputs* found on page 37 for a description of this bit. This bit is reset in most applications.
A11	M (Internal Relay)	Bit	0/1	Stall_Detect_Enable (Encoder must be enabled, Argument A9 = 1) 0 = Stall detection disabled, 1 = Stall detection enabled
A12	M (Internal Relay)	Bit	0/1	Antiresonance_Disable 0 = Antiresonance enabled, 1 = Antiresonance disabled
A13	M (Internal Relay)	Bit	0/1	Output State on Network Loss 0 = Output on, 1 = Output off Only used if A14 = "1"
A14	M (Internal Relay)	Bit	0/1	Output State Control on Network Loss 0 = Output keeps last state, 1 = Output assumes state set by A13
A15	M (Internal Relay)	Bit	0/1	Output_Functionality 0 = Output is a fault output, 1 = Output is a general purpose output.
A16	M (Internal Relay)	Bit	0/1	Drive_Enable 0 = Drive section disabled. Not current to motor. 1 = Drive section enabled.
A17	D (Data Register)	Word	0 to 999	Starting Speed. All moves begin and end at this speed
A18	D (Data Register)	Word	200 to 32,767	Motor Steps per Turn
A19	D (Data Register)	Word	0 to 32,767	Encoder Counts per Turn
A20	D (Data Register)	Word	0 to 100	Idle Current, as a percentage of Motor Current 0 = Current removed when idle 100 = No current reduction when idle
A21	D (Data Register)	Word	1 to 40	Motor Current in steps of 0.1 amps. 1 = 0.1 amp motor current 40 = 4.0 amp motor current
A22	D (Data Register)	Word	1 to 80	Current Loop Gain "5" is suggested value for all AMCI by IDEC Corporation motors. This value can be used as a reasonable value for motors from other manufacturers.

Table T5.2 Arguments to the "Configure IANG1E" Macro

Value	Function	Description
0	General Purpose Input	The input is not used in any of the functions of the IANG1(E), but it's status is reported in the Input data registers. This allows the input to be used as a discrete DC input to the host controller.
1	CW Limit	Input defines the mechanical end point for CW motion.
2	CCW Limit	Input defines the mechanical end point for CCW motion.
3	Start Indexed Move	Starts the move that is currently located in the output registers.
3	Start Indexed Move / Capture Encoder Value	When the encoder is enabled on an IANG1(E), the encoder position value is captured whenever this input transitions. An inactive-to-active state transition will also trigger an Indexed Move if one is pending in the IANG1(E).
4	Stop Jog or Registration Move	Brings a Jog or Registration Move to a controlled stop.
4	Stop Jog or Registration Move & Capture Encoder Value	When the encoder is enabled on an IANG1(E), the encoder position value is captured when the input triggers a controlled stop to a Jog or Registration move.
5	Emergency Stop	All motion is immediately stopped when this input makes an inactive-to-active transition.
6	Home	Used to define the home position of the machine.

Table T5.3 Input Function Definitions

5.3.2 Arguments to the "Absolute Move IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	M (Internal Relay)	Bit	0/1	Move Trigger Type 0 = Normal Move. Move begins as soon as data is accepted. 1 = Indexed Move. Move triggered by discrete input. See *Indexed Moves* found on page 26 for additional information.
A3	D (Data Register)	Long†	–8,388,607 to +8,388,607	Target Position
A4	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the "Configure IANG1" Macro* found on page 61.)
A5	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A6	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A7	D (Data Register)	Word	0 to 5000	Acceleration/Deceleration Jerk 0 = Linear Acceleration 1 = Triangular Acceleration 2 to 5,000 = Trapezoidal Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.4 Arguments to the "Absolute Move IANG1E" Macro

5.3.3 Arguments to the “Relative Move IANG1E” Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	M (Internal Relay)	Bit	0/1	Move Trigger Type 0 = Normal Move. Move begins as soon as data is accepted. 1 = Indexed Move. Move triggered by discrete input. See *Indexed Moves* found on page 26 for additional information.
A3	D (Data Register)	Long†	–8,388,607 to +8,388,607	Target Distance
A4	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the “Configure IANG1” Macro* found on page 61.)
A5	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A6	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A7	D (Data Register)	Word	0 to 5000	Acceleration/Deceleration Jerk 0 = Linear Acceleration 1 = Triangular Acceleration 2 to 5,000 = Trapezoidal Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.5 Arguments to the “Relative Move IANG1E” Macro

5.3.4 Arguments to the “Hold IANG1E” Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)

Table T5.6 Arguments to the “Hold IANG1E” Macro

5.3.5 Arguments to the “Resume IANG1E” Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)

Table T5.7 Arguments to the “Resume IANG1E” Macro

5.3.6 Arguments to the “Immediate Stop IANG1E” Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)

Table T5.8 Arguments to the “Immediate Stop IANG1E” Macro

5.3.7 Arguments to the “Home CW IANG1E” Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	M (Internal Relay)	Bit	0/1	Move Trigger Type 0 = Normal Move. Move begins as soon as data is accepted. 1 = Indexed Move. Move triggered by discrete input. See *Indexed Moves* found on page 26 for additional information.
A3	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the “Configure IANG1” Macro* found on page 61.)
A4	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A6	D (Data Register)	Word	0 to 5000	Acceleration/Deceleration Jerk 0 = Linear Acceleration 1 = Triangular Acceleration 2 to 5,000 = Trapezoidal Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.9 Arguments to the “Home CW IANG1E” Macro

5.3.8 Arguments to the “Home CCW IANG1E” Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	M (Internal Relay)	Bit	0/1	Move Trigger Type 0 = Normal Move. Move begins as soon as data is accepted. 1 = Indexed Move. Move triggered by discrete input. See *Indexed Moves* found on page 26 for additional information.
A3	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the “Configure IANG1” Macro* found on page 61.)
A4	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A5	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A6	D (Data Register)	Word	0 to 5000	Acceleration/Deceleration Jerk 0 = Linear Acceleration 1 = Triangular Acceleration 2 to 5,000 = Trapezoidal Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.10 Arguments to the “Home CCW IANG1E” Macro

5.3.9 Arguments to the "Jog CW IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the "Configure IANG1" Macro* found on page 61.)
A3	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A4	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A5	D (Data Register)	Word	0 to 5000	Acceleration/Deceleration Jerk 0 = Linear Acceleration 1 = Triangular Acceleration 2 to 5,000 = Trapezoidal Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.11 Arguments to the "Jog CW IANG1E" Macro

5.3.10 Arguments to the "Registration Move CW IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	M (Internal Relay)	Bit	0/1	Move Trigger Type 0 = Normal Move. Move begins as soon as data is accepted. 1 = Indexed Move. Move triggered by discrete input. See *Indexed Moves* found on page 26 for additional information.
A3	D (Data Register)	Long†	0 to 8,388,607	Stopping Distance.
A4	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the "Configure IANG1" Macro* found on page 61.)
A5	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A6	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A7	D (Data Register)	Long†	0 to 8,388,607	Minimum Registration Move Distance.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.12 Arguments to the "Registration Move CW IANG1E" Macro

5.3.11 Arguments to the "Jog CCW IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the "Configure IANG1" Macro* found on page 61.)
A3	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A4	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A5	D (Data Register)	Word	0 to 5000	Acceleration/Deceleration Jerk 0 = Linear Acceleration 1 = Triangular Acceleration 2 to 5,000 = Trapezoidal Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.13 Arguments to the "Jog CCW IANG1E" Macro

5.3.12 Arguments to the "Registration Move CCW IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	M (Internal Relay)	Bit	0/1	Move Trigger Type 0 = Normal Move. Move begins as soon as data is accepted. 1 = Indexed Move. Move triggered by discrete input. See *Indexed Moves* found on page 26 for additional information.
A3	D (Data Register)	Long†	0 to 8,388,607	Stopping Distance.
A4	D (Data Register)	Long†	Starting Speed to 2,999,999	Programmed Speed. (Starting Speed is one of the *Arguments to the "Configure IANG1" Macro* found on page 61.)
A5	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A6	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A7	D (Data Register)	Long†	0 to 8,388,607	Minimum Registration Move Distance.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A3 begins at address D49622, the next available address would be D49624. (A3 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A3 could not start with register D49621).

Table T5.14 Arguments to the "Registration Move CCW IANG1E" Macro

5.3.13 Arguments to the "Preset Position IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	D (Data Register)	Long†	–8,388,607 to +8,388,607	Desired Motor Position value.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.15 Arguments to the "Preset Position IANG1E" Macro

5.3.14 Arguments to the "Clear Errors IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)

Table T5.16 Arguments to the "Clear Errors IANG1E" Macro

5.3.15 Arguments to the "Preset Encoder IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	D (Data Register)	Long†	–8,388,607 to +8,388,607	Desired Encoder Position value.

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.17 Arguments to the "Preset Encoder IANG1E" Macro

5.3.16 Arguments to the "Drive Enable IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)

Table T5.18 Arguments to the "Drive Enable IANG1E" Macro

5.3.17 Arguments to the "Drive Disable IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)

Table T5.19 Arguments to the "Drive Disable IANG1E" Macro

5.3.18 Arguments to the "Encoder Follower IANG1E" Macro

Argument	Device Type	Device Size	Range of Values	Description
A1	D (Data Register)	Word		Starting Address of the output registers assigned to the axis controlled by the IANG1(E). (D49120 to D49230 in the sample programs.)
A2	D (Data Register)	Word	1 to 255	Encoder Follower Multiplier
A3	D (Data Register)	Word	1 to 255	Encoder Follower Divisor
A3	D (Data Register)	Long†	Starting Speed to 2,999,999	Motor Speed. (Starting Speed is one of the *Arguments to the "Configure IANG1" Macro* found on page 61.)
A3	D (Data Register)	Word	1 to 5000	Acceleration in steps/millisecond/second.
A4	D (Data Register)	Word	1 to 5000	Deceleration in steps/millisecond/second.
A5	D (Data Register)	Word	0 to 5000	Acceleration/Deceleration Jerk 0 = Linear Acceleration 1 = Triangular Acceleration 2 to 5,000 = Trapezoidal Acceleration

† Long integer values consume two consecutive data registers and the starting address must be even. For example, if argument A2 begins at address D49622, the next available address would be D49624. (A2 would consume both D49622 and D49623.) The starting register address of a long value must be even. (For example, argument A2 could not start with register D49621).

Table T5.20 Arguments to the "Encoder Follower IANG1E" Macro

5.4 Add a User-defined Macro to Ladder Logic

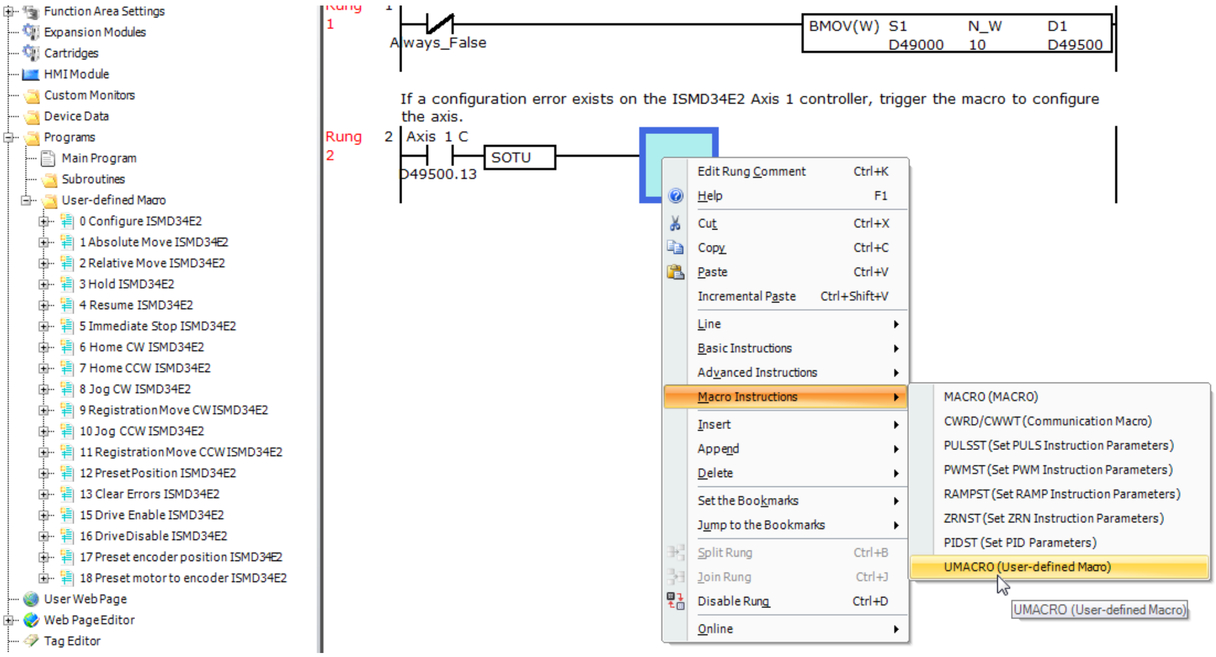
Adding a User-defined Macro to your ladder logic requires you to set the initial conditions for the rung, select the macro, and map the macro's arguments to the Device Addresses chosen in the previous step.

5.4.1 Set initial conditions

Most macros should be triggered once to initiate a data transfer to the IANG1(E). Therefore, a **Single Output Up** instruction, (SOTU), should be added to the rung as part of the condition. The exceptions to this guideline are the Jog and Registration Moves.

5.4.2 Choose the desired macro to add to the ladder.

To add a User-defined Macro to the ladder, first, right-click where you wish to add the macro. In the popup menu that appears, hover over the "**Macro Instructions ▶**" item until a second menu appears. From that menu, select **UMACRO (User-defined Macro)**. The UMACRO (User-defined Macro) dialog box will appear.



If a configuration error exists on the ISMD34E2 Axis 1 controller, trigger the macro to configure the axis.

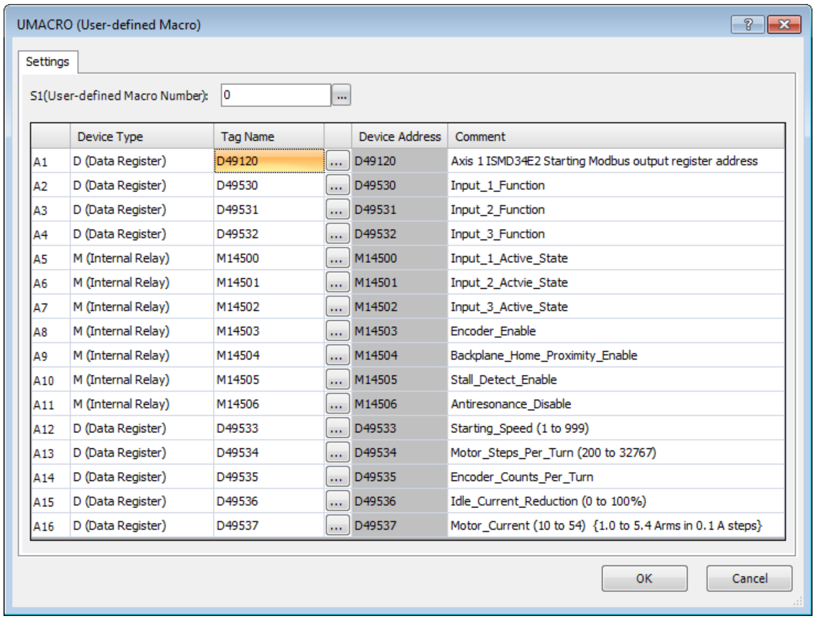
Figure T5.2 Selecting a User-defined Macro

5.4.3 Select the Proper Macro by Macro Number

Each macro is assigned a number between 0 and 255. When user-defined macros are viewed in the Project Window, these numbers are shown before the name of the macro. (See figure T5.2 above for an example.) To select the proper macro, enter the correct number in the "S1 (User-defined Macro Number):" field and press the [Tab] key. The program will open a confirmation window stating that the current settings will be lost and the argument list will be updated. Click on the [Yes] button to confirm the change.

5.4.4 Map Arguments to Device Addresses

(Continuing from the previous step.) The **UMACRO (User-defined Macro)** dialog box now displays a list of the macro’s arguments along with their required Device Type. The Tag Name fields are used to map the Device Addresses to the macro’s arguments. If a tag name has been defined, this name can be typed directly into the field. Optionally, the actual Device Address can be entered into the field. Finally the ellipsis button […] can be clicked to open the **Tag Name Editor** dialog box. Once opened, it can be used to chose the desired Device Address. Figure T5.3 below shows a **UMACRO (User-defined Macro)** dialog box for the “Configure IANG1E” macro after Device Addresses have been mapped to the macro’s arguments.



S1(User-defined Macro Number): 0

	Device Type	Tag Name		Device Address	Comment
A1	D (Data Register)	D49120	...	D49120	Axis 1 ISMD34E2 Starting Modbus output register address
A2	D (Data Register)	D49530	...	D49530	Input_1_Function
A3	D (Data Register)	D49531	...	D49531	Input_2_Function
A4	D (Data Register)	D49532	...	D49532	Input_3_Function
A5	M (Internal Relay)	M14500	...	M14500	Input_1_Active_State
A6	M (Internal Relay)	M14501	...	M14501	Input_2_Actvie_State
A7	M (Internal Relay)	M14502	...	M14502	Input_3_Active_State
A8	M (Internal Relay)	M14503	...	M14503	Encoder_Enable
A9	M (Internal Relay)	M14504	...	M14504	Backplane_Home_Proximity_Enable
A10	M (Internal Relay)	M14505	...	M14505	Stall_Detect_Enable
A11	M (Internal Relay)	M14506	...	M14506	Antiresonance_Disable
A12	D (Data Register)	D49533	...	D49533	Starting_Speed (1 to 999)
A13	D (Data Register)	D49534	...	D49534	Motor_Steps_Per_Turn (200 to 32767)
A14	D (Data Register)	D49535	...	D49535	Encoder_Counts_Per_Turn
A15	D (Data Register)	D49536	...	D49536	Idle_Current_Reduction (0 to 100%)
A16	D (Data Register)	D49537	...	D49537	Motor_Current (10 to 54) {1.0 to 5.4 Arms in 0.1 A steps}

Figure T5.3 Argument Assignments

Notes

TASK 6: ADDITIONAL LOGIC AND INPUT DATA FORMATS

Introduction

The IDEC User-defined Macros simplify issuing commands and configuration data to the IANG1(E). Once a command, or configuration data, is issued, monitoring the state of the IANG1(E) is application specific and must be written. The IANG1E sample program available on the IDEC website shows one method of monitoring an IANG1(E) and controlling when commands are issued.

The following sections give guidelines for writing the additional logic needed to control an IANG1(E) as well as the format of the status data that is read from the IANG1(E) in both Command and Configuration modes.

6.1 Power-Up Behavior

The ANG1(E) will always power up in Command Mode and show a configuration error. (The first data register read from the IANG1(E) axis will have a hexadecimal value of 6408h.) Configuration data must be written down to the ANG1(E) before commands can be issued to the module.

Note➤ The ANG1(E) will not supply power to the motor as long as there is a configuration error. (The configuration data sets, among other things, the motor current value. Without the information, the ANG1(E) does not know what current to apply to the motor.) Therefore, the motor will have no holding torque while a configuration error exists.

6.2 Buffer Input Data

It is common practice to buffer the input data from the IANG1(E) to guarantee that it does not change during a program scan. IDEC sample programs use a BMOV instruction to copy data from the Modbus TCP registers to a buffer in the D49500 to D49999 range. This instruction is conditioned by an NC relay tied to a bit defined as Always_False. This buffers the data on every program scan and must occur before the data is used in the program.

6.3 Command Bits Must Transition

Note➤ Commands are only accepted when the command bit makes a 0→1 transition. The easiest way to meet this requirement is to write a value of zero into the first of the output registers assigned to the IANG1(E) before writing the next command. These registers are in the range of D49120 to D49239 in the IDEC sample program. See T5.1, *Reserved Memory Addresses* on page 59 for the memory layout used by the IDEC sample program.

This condition also applies when switching from Configuration Mode to Command Mode. Assume a bit is set in the first of the registers assigned to the IANG1(E) while in Configuration Mode. If a switch is made to Command Mode with the same bit set, the command will not occur because the bit must transition between writes to the unit.

6.3.1 Commands that do not cause motion: Clear Errors, preset commands, etc.

In the unlikely event that a command in this group is issued successively, its command bit must be reset and held for at least one Modbus TCP update. The time it must be held is dependent on the amount of TCP traffic in your application.

6.3.2 Jog and Registration Moves

Unlike other motion commands, the Jog and Registration Moves only occur while the command bit is set. Your program must monitor machine conditions to determine when to reset the command bit to zero. Once the command bit is off, your program should monitor the Moving_CW, Moving_CCW, or Stopped status bits to verify that the axis has stopped moving before issuing another command.

6.3.3 Absolute Move, Relative Move, and Find Home

Once issued, these commands will run to their completion regardless of the state of their command bit. Once the command has been issued, your program can monitor the Moving_CW, Moving_CCW, or Stopped status bits to verify the axis is in motion and reset the command word at that point. Unless the commanded move is very short, a network update will occur while motion is occurring and the IANG1(E) will see the reset command bit. As with the Jog and Registration Moves, the program should monitor the Moving_CW, Moving_CCW, or Stopped status bits to verify that the axis has stopped moving before issuing another command.

6.4 Command Mode Input Data Format

The format for the Network Input Data when the IANG1(E) is in Command Mode is shown below.

Modbus TCP Register	Command Mode Input Data
0	Status Word 0
1	Status Word 1
2	32 bit Motor Position: Upper 16 bits
3	32 bit Motor Position: Lower 16 bits
4	32 bit Encoder Position: Upper 16 bits
5	32 bit Encoder Position: Lower 16 bits
6	32 bit Trapped Encoder Position: Upper 16 bits
7	32 bit Trapped Encoder Position: Lower 16 bits
8	Programmed Motor Current (X10)
9	Value of Acceleration Jerk Parameter

Table T6.1 Network Input Data Format: Command Mode

6.4.1 Status Word 0 Format

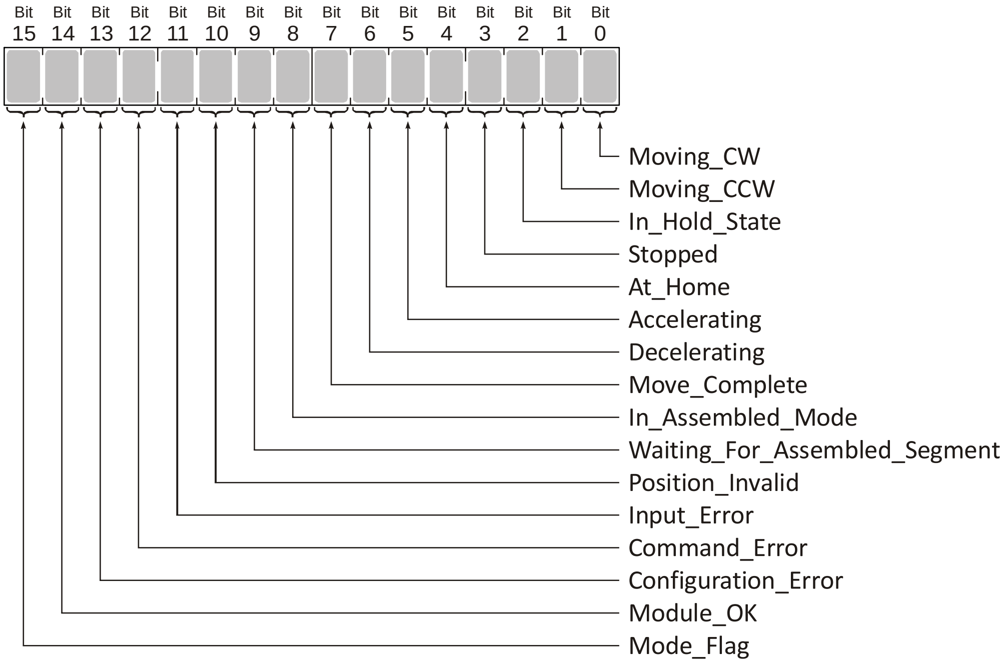


Figure T6.1 Command Mode: Status Word 0 Format

Bit 0: Moving_CW – Set to “1” when the motor is rotating in a clockwise direction, otherwise “0”.

Bit 1: Moving_CCW – Set to “1” when the motor is rotating in a counter-clockwise direction, otherwise “0”.

Bit 2: In_Hold_State – Set to “1” when a move command has been successfully brought into a Hold State. Hold States are explained in the *Controlling Moves In Progress* section starting on page 27.

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

Bit 4: At_Home – Set to “1” when a homing command has completed successfully, “0” at all other times. This bit is only set at the successful completion of a Find Home command. This bit is not set if the home position is reached by any other move.

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

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

Bit 7: Move_Complete – Set to “1” when the present move command completes without error. This bit is reset to “0” when the next move command is written to the IANG1(E), when the position is preset, or a *Clear 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 8: In_Assembled_Mode – The IANG1(E) sets this bit to signal the host that it is ready to accept assembled move profile programming data. Assembled Moves do not have User-defined Macros at this point in time. The use of this bit is explained in the *Assembled Move Programming* section of this manual starting on page 97.

Bit 9: Waiting_For_Assembled_Segment – The IANG1(E) sets this bit to tell the host that it is ready to accept the data for the next segment of your assembled move profile. Assembled Moves do not have User-defined Macros at this point in time. The use of this bit is explained in the *Assembled Move Programming* section of this manual starting on page 97.

Bit 10: Position_Invalid – “1” when:

- A configuration is written to the IANG1(E)
- The motor position has not been preset or 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 11: Input_Error – “1” when:

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

This bit is reset by a *Clear Errors* command. This command can be issued with a Clear Errors IANG1(E) User-defined Macro available from IDEC. The format of the *Arguments to the “Clear Errors IANG1E” Macro* is given on page 66.

Bit 12: Command_Error – “1” when an invalid command has been written to the IANG1(E). This bit can only be reset by the *Reset_Errors* bit, Command Word 0, Bit 10. See 5.3.14, *Arguments to the “Clear Errors IANG1E” Macro* on page 66 for the macro that can be used to clear the error.

Bit 13: Configuration_Error – “0” when the IANG1(E) has a valid configuration in memory. “1” on power up or when the ANG1(E) experiences a power on reset. Also “1” after any invalid configuration has been written to the unit.

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

Bit 14: Module_OK – “1” when the IANG1(E) is operating without a fault, “0” when an internal fault condition exists.

Bit 15: Mode_Flag – Set to “1” if in Configuration Mode, and set to “0” if in Command Mode.

6.4.2 Status Word 1 Format

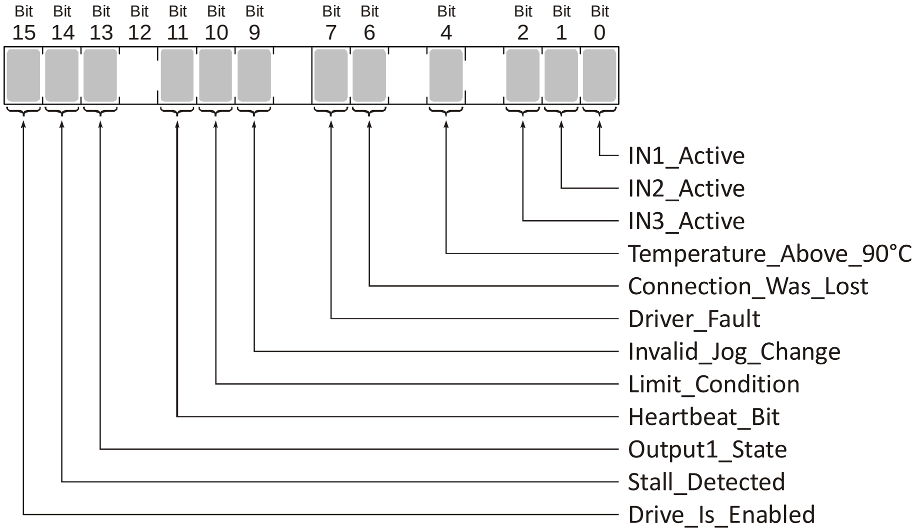


Figure R4.5 Command Mode: Status Word 1 Format

Bit 0: IN1_Active – "1" when Input 1 is in its active state. The active state of the input is programmed with argument A5 of the "Configure IANG1E" macro. *Arguments to the "Configure IANG1" Macro* can be found on page 61.

Bit 1: IN2_Active – "1" when Input 2 is in its active state. The active state of the input is programmed with argument A6 of the "Configure IANG1E" macro. *Arguments to the "Configure IANG1" Macro* can be found on page 61.

Bit 2: IN3_Active – "1" when Input 3 is in its active state. The active state of the input is programmed with argument A7 of the "Configure IANG1E" macro. *Arguments to the "Configure IANG1" Macro* can be found on page 61.

Bit 3: Reserved – This bit will always equal zero.

Bit 4: Temperature_Above_90°C – 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 IIC-5 connector on each side of the IANG1(E) to allow for more cooling space, or install a fan below the stack to allow additional airflow through the stack.

Bit 5: Reserved – Will always equal zero.

Bit 6: Connection_Was_Lost – If the *physical* network connection is lost at any time, this bit will be set when the connection is re-established. The Input_Error bit will also be set. Note that this bit is not set if the communication loss is on the protocol level, not the physical level.

Bit 7: Drive_Fault – If the drive section of the IANG1(E) is enabled, this bit will be a "1" during an Over Temperature fault or Short Circuit fault. This fault is cleared by removing the source of the fault and cycling power IANG1(E).

Bit 8: Reserved – Will always equal zero.

Bit 9: Invalid_Jog_Change – Set during a Jog Move if parameters are changed to invalid values. Parameters that can be changed during a Jog Move are Programmed Speed, Acceleration, and Deceleration. Also set while in Encoder Follower mode if the Multiplier or Divisor are set outside their range of 1 to 255.

Bit 10: Limit_Condition – This bit is set if an End Limit Switch is reached during a move. This bit will be reset when the Limit Switch changes from its active to inactive state, or when a *Clear Errors* Command is issued.

Bit 11: Heartbeat_Bit – This bit will change state approximately every 500 milliseconds. Monitor this bit to verify that the unit and network connection are operating correctly. Note that this bit is only available while the ANG1(E) is in Command Mode.

Bit 12: Reserved – This bit will always equal zero.

Bit 13: Output1_State – Present state of Output1. When this bit equals "1", the output is in its conductive state.

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

Bit 15: Drive_Is_Enabled – Set to "1" when the motor drive section of the IANG1(E) is enabled and current is available to the motor. Set to "0" when the motor drive section is disabled. If this bit is set to "1", the motor current remains present when an E-Stop input is active. Motor current is removed if there is a Drive_Fault (Bit 7 above) regardless of the state of this bit. Motor current is also removed if the motor is idle and Idle Current Reduction is programmed to its *To 0%* setting.

6.5 Configuration Mode Input Data Format

The format for the Network Input Data when an IANG1(E) is in Configuration Mode is shown below.

Modbus TCP Register	Configuration Data
0	Mirror of Output Configuration Word 0
1	Mirror of Output Configuration Word 1
2	0000
3	Mirror of Starting Speed
4	Mirror of Motor Steps/Turn
5	0000
6	Mirror of Encoder_Resolution
7	Mirror of Idle Current Percentage
8	Mirror of Motor Current (X10)
9	Mirror of Current Loop Gain

Table R4.1 Network Input Data Format: Configuration Mode

6.5.1 Configuration Word 0 Mirror

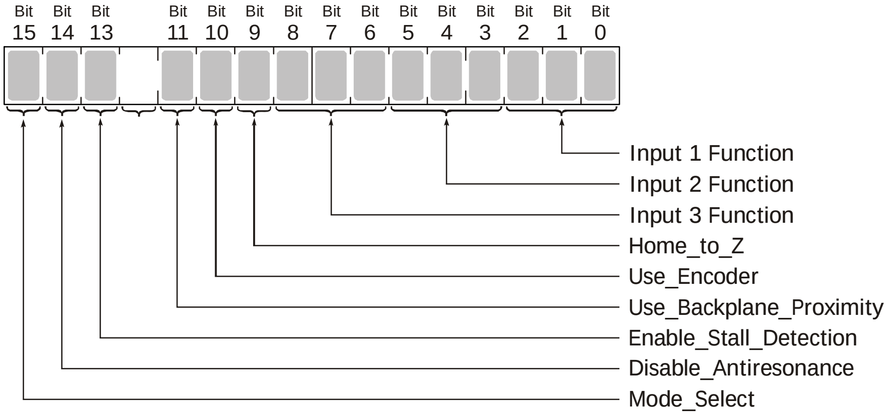


Figure R4.6 Configuration Mode: Configuration Word 0 Format

Bits 2-0: Input 1 Function – See the table on the following page.

Bits 5-3: Input 2 Function – See the table on the following page.

Bits 8-6: Input 3 Function – See the table on the following page.

Bit 9: Home_to_Z – Set to “1” to home the machine to the encoder’s Z pulse. The Use_Encoder bit, bit 10, must also be set. You must also program the Encoder_Resolution parameter in configuration word 6. If one of the three discrete inputs is programmed as a Home Input, is will act as a home proximity input. In this case, the Z-pulse will not be acted upon unless the discrete input is in its active state.

Bit 10: Use_Encoder – “0” when the quadrature encoder inputs are not used. “1” to enable the quadrature encoder inputs.You must also program the Encoder_Resolution parameter in configuration word 7. (Modbus register address 1030.)

Bit 11: Use_Backplane_Proximity – “0” when the Backplane_Proximity_Bit is not used when homing the IANG1(E). “1” when the Backplane_Proximity_Bit is used when homing the IANG1(E). Note that this bit is not the Backplane_Proximity_Bit, but enables or disables its operation. Do not use the Backplane_Proximity_Bit if you only want to home to a Home Limit Switch. (Leave this bit equal to “0”.) If you enable this bit and then never turn on the Backplane_Proximity_Bit, the IANG1(E) will ignore all transitions of the home limit switch and you will not be able to home the IANG1(E).

Bit 12: Reserved – Must equal zero.

Bit 13: Enable_Stall_Detection – “0” disables motor stall detection. “1” enables motor stall detection. The Use_Encoder bit, which is bit 10 of this word, must be also be set to “1”. You must also program the Encoder_Resolution parameter in configuration word 6.

Bit 14: Disable_Antiresonance – “1” disables the antiresonance feature. “0” enables the anti-resonance feature of the IANG1(E). 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 15: Mode_Select – “1” for Configuration Mode Programming, “0” for Command Mode Programming.

Bits				
8	7	6		
5	4	3		
2	1	0	Function	Description
0	0	0	General Purpose Input	The input is not used in any of the functions of the IANG1(E), but it's status is reported in the Network Data. This allows the input to be used as a discrete DC input to the host controller.
0	0	1	CW Limit	Input defines the mechanical end point for CW motion.
0	1	0	CCW Limit	Input defines the mechanical end point for CCW motion.
0	1	1	Start Indexed Move	Starts the move that is currently located in the output registers.
0	1	1	Start Indexed Move / Capture Encoder Value	When the quadrature encoder inputs are enabled on an IANG1(E), the encoder position value is captured whenever this input transitions. An inactive-to-active state transition will also trigger an Indexed Move if one is pending in the IANG1(E).
1	0	0	Stop Jog or Registration Move	Brings a Jog or Registration Move to a controlled stop.
1	0	0	Stop Jog or Registration Move & Capture Encoder Value	When the quadrature encoder inputs are enabled on an IANG1(E), the encoder position value is captured when the input triggers a controlled stop to a Jog or Registration move.
1	0	1	Emergency Stop	All motion is immediately stopped when this input makes an inactive-to-active transition.
1	1	0	Home	Used to define the home position of the machine. When homing to the Z-pulse of the encoder, (bit 9 of Configuration Word 0 = "1"), this input will act as a home proximity input.
1	1	1	Invalid Combination	This bit combination is reserved.

Table R4.2 Configuration Data: Input Function Selections

Note➤ When using an encoder, you must set bit 10, the Use_Encoder bit. You must also program the Encoder_Resolution parameter in configuration word 6.

Configuration Word 1 Mirror

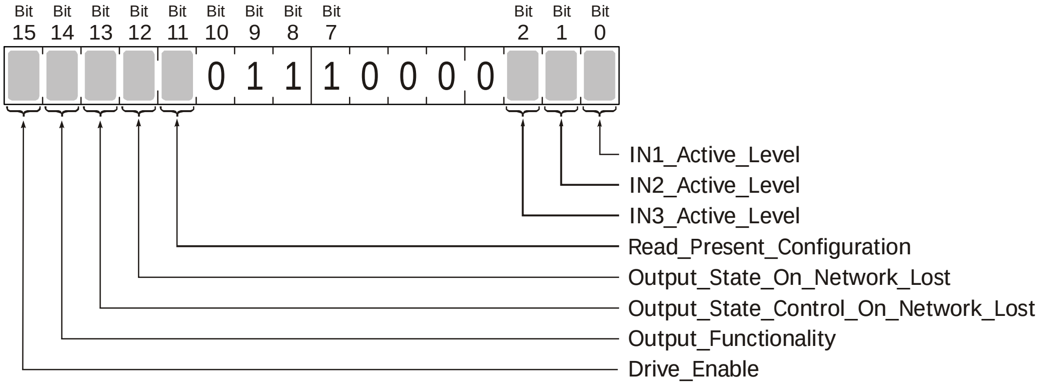


Figure R4.7 Configuration Mode: Configuration Word 1 Format

Bit 0: IN1_Active_Level – Determines the active state of Input 1. Set to "0" if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to "1" if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Bit 1: IN2_Active_Level – Determines the active state of Input 2. Set to "0" if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to "1" if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Bit 2: IN3_Active_Level – Determines the active state of Input 3. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

> **Note➤** If you are not using the input, sets its Active_Level bit to “1”. The input will then always report as inactive in the network data.

Bits 3 - 6: Reserved – Must be reset to zero.

Bits 7 - 9: Reserved – Must be set to one. These bits control the format of data accepted and transmitted by the IANG1(E). They must be set to “111” in order to be compatible with the programming macros published by IDEC.

Bit 10: Reserved – Must be reset to zero.

Bit 11: Read_Present_Configuration – If this bit is set when you enter Configuration Mode, the IANG1(E) responds by placing the present configuration data in the Network Input Data. You cannot write new configuration data to the unit while this bit is set. The format of the Configuration Data is given in the *Output Data Format* section of this chapter, starting on page 80.

Bit 12: Output_State_On_Network_Loss – 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 13: Read_Present_Configuration – “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		
13	**12**	**Description**
0	0	Output 1 stays in its last state if the network connection is lost.
0	1	Output 1 stays in its last state if the network connection is lost.
1	0	Output 1 turns off if the network connection is lost.
1	1	Output 1 turns on (conducts) if the network connection is lost.

Bit 14: Output_Functionality – “0” configures Output 1 to be a Fault Output. The output will conduct current until a fault occurs. “1” configures Output 1 to be a general purpose output whose state is determined by a bit in the Command Mode output data registers.

Bit 15: Drive_Enable – “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.

Notes on Other Configuration Words

- Changes to the Idle Current only take effect at the *end of the first move after re-configuration*.

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 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. (An error does not occur if both are configured as General Purpose Inputs.)
4) Setting the *Home to Encoder Z-Pluse* Bit without configuring theIANG1(E) to use an encoder.
5) Setting the *Stall_Detect_Enable* Bit without configuring the IANG1(E) to use an encoder.
6) Using and encoder and not setting the Encoder Pulses/Turn parameter to a valid value.
7) Setting the Input Configuration bits for any input to “111”. See table T5.3, *Input Function Definitions* on page 62 for more information. When using the IDEC User-defined Macro for configuration, the input configuration value must be between 0 and 6.

Notes

Reference 5: Config. Mode Output Data Format

Introduction

This chapter covers the format of the Network Output Data used to configure the IANG1(E). An IANG1(E) requires ten data registers for output data to the module as well as ten data registers to hold data read from the module. The format of this data is a mix of sixteen bit and thirty-two bit integers.

Note➤ The custom macros supplied by IDEC simplify the configuration and programming of the IANG1(E) units. The data in this reference is presented as background information to aid in troubleshooting.

Power Up Behavior

The ANG1(E) will always power up in Command Mode and show a configuration error. (The first data register read from the ANG1(E) will have a hexadecimal value of 6408h.) Configuration data must be written down to the ANG1(E) before commands can be issued to the module.

Note➤ The ANG1(E) will not supply power to the motor as long as there is a configuration error. (The configuration data sets, among other things, the motor current value. Without the information, the ANG1(E) does not know what current to apply to the motor.) Therefore, the motor will have no holding torque while a configuration error exists.

Data Format

An IANG1(E) requires ten registers for output data as well as ten registers for input data. These words are broken down into a combination of sixteen and thirty-two bit integers.

Thirty-two bit values are transmitted most significant word first (big endian). This is the data format specified by the Modbus specification and used by IDEC controllers. Table R5.1 below shows the format of the data as it is stored and transmitted.

	32 bit Integer Format	
Value	**First Word**	**Second Word**
12	0x0000	0x000C
-12	0xFFFF	0xFFF4
1,234,567	0x0012	0xD687
-7,654,321	0xFF8B	0x344F

Table R5.1 Position Data Format Examples

Output Data Format

The correct format for the output data registers when the IANG1(E) is in Configuration Mode is shown below.

Register	Configuration Data	Range
1024	Configuration Word 0	See below
1025	Configuration Word 1	See below
1026	Reserved	Set to zero
1027	Starting Speed (steps/sec)	1 to 999
1028	Motor Steps/Turn	200 to 32,767
1029	Reserved	Set to zero
1030	Encoder_Resolution	0 to 32,767
1031	Idle Current Percentage	0 to 100%
1032	Motor Current (X10)	1 to 40, Represents 0.1 to 4.0 Arms
1033	Current Loop Gain	1 to 80

Table R5.2 Network Output Data Format: Configuration Mode

Configuration Word 0 Format

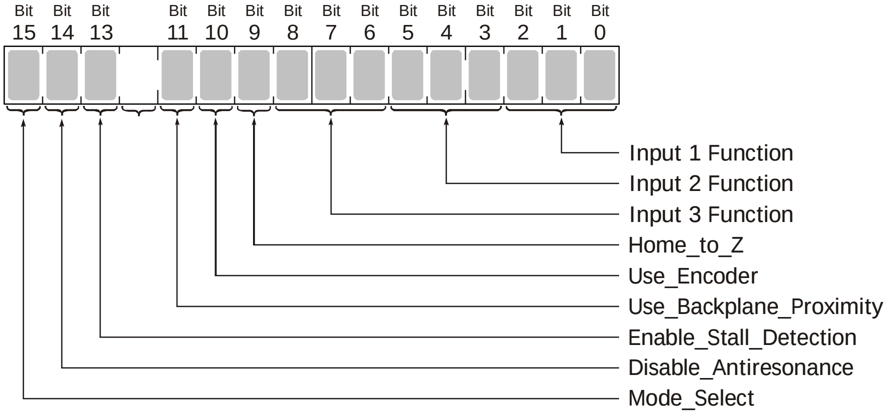


Figure R5.1 Configuration Mode: Configuration Word 0 Format

Bits 2-0: Input 1 Function – See the table on the following page.

Bits 5-3: Input 2 Function – See the table on the following page.

Bits 8-6: Input 3 Function – See the table on the following page.

Bit 9: Home_to_Z – Set to “1” to home the machine to the encoder’s Z pulse. The Use_Encoder bit, bit 10, must also be set. You must also program the Encoder_Resolution parameter in configuration word 6. If one of the three discrete inputs is programmed as a Home Input, is will act as a home proximity input. In this case, the Z-pulse will not be acted upon unless the discrete input is in its active state.

Bit 10: Use_Encoder – “0” when the quadrature encoder inputs are not used. “1” to enable the quadrature encoder inputs.You must also program the Encoder_Resolution parameter in configuration word 7. (Modbus register address 1030.)

Bit 11: Use_Backplane_Proximity – “0” when the Backplane_Proximity_Bit is not used when homing the IANG1(E). “1” when the Backplane_Proximity_Bit is used when homing the IANG1(E). Note that this bit is not the Backplane_Proximity_Bit, but enables or disables its operation. Do not use the Backplane_Proximity_Bit if you only want to home to a Home Limit Switch. (Leave this bit equal to “0”.) If you enable this bit and then never turn on the Backplane_Proximity_Bit, the IANG1(E) will ignore all transitions of the home limit switch and you will not be able to home the IANG1(E).

Bit 12: Reserved – Must equal zero.

Bit 13: Enable_Stall_Detection – “0” disables motor stall detection. “1” enables motor stall detection. The Use_Encoder bit, which is bit 10 of this word, must be also be set to “1”. You must also program the Encoder_Resolution parameter in configuration word 6.

Bit 14: Disable_Antiresonance – “1” disables the antiresonance feature. “0” enables the anti-resonance feature of the IANG1(E). 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 15: Mode_Select – “1” for Configuration Mode Programming, “0” for Command Mode Programming.

Bits				
8	7	6		
5	4	3		
2	**1**	**0**	**Function**	**Description**
0	0	0	General Purpose Input	The input is not used in any of the functions of the IANG1(E), but it's status is reported in the Network Data. This allows the input to be used as a discrete DC input to the host controller.
0	0	1	CW Limit	Input defines the mechanical end point for CW motion.
0	1	0	CCW Limit	Input defines the mechanical end point for CCW motion.
0	1	1	Start Indexed Move	Starts the move that is currently located in the output registers.
0	1	1	Start Indexed Move / Capture Encoder Value	When the quadrature encoder inputs are enabled on an IANG1(E), the encoder position value is captured whenever this input transitions. An inactive-to-active state transition will also trigger an Indexed Move if one is pending in the IANG1(E).
1	0	0	Stop Jog or Registration Move	Brings a Jog or Registration Move to a controlled stop.
1	0	0	Stop Jog or Registration Move & Capture Encoder Value	When the quadrature encoder inputs are enabled on an IANG1(E), the encoder position value is captured when the input triggers a controlled stop to a Jog or Registration move.
1	0	1	Emergency Stop	All motion is immediately stopped when this input makes an inactive-to-active transition.
1	1	0	Home	Used to define the home position of the machine. When homing to the Z-pulse of the encoder, (bit 9 of Configuration Word 0 = "1"), this input will act as a home proximity input.
1	1	1	Invalid Combination	This bit combination is reserved.

Table R5.3 Configuration Data: Input Function Selections

> **Note➤** When using an encoder, you must set bit 10, the Use_Encoder bit. You must also program the Encoder_Resolution parameter in configuration word 6.

Configuration Word 1 Format

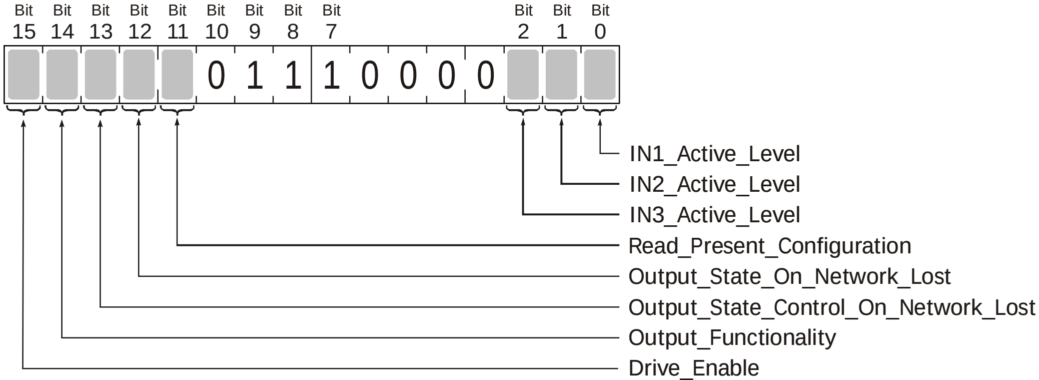


Figure R5.2 Configuration Mode: Configuration Word 1 Format

Bit 0: IN1_Active_Level – Determines the active state of Input 1. Set to "0" if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to "1" if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Bit 1: IN2_Active_Level – Determines the active state of Input 2. Set to "0" if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to "1" if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Bit 2: IN3_Active_Level – Determines the active state of Input 3. Set to “0” if your sensor has Normally Closed (NC) contacts and the input is active when there is no current flow through it. Set to “1” if your sensor has Normally Open (NO) contacts and current flows through the input when it is active.

Note➤ If you are not using the input, sets its Active_Level bit to “1”. The input will then always report as inactive in the network data.

Bits 3 - 6: Reserved – Must be reset to zero.

Bits 7 - 9: Reserved – Must be set to one. These bits control the format of data accepted and transmitted by the IANG1(E). They must be set to “111” in order to be compatible with the programming macros published by IDEC.

Bit 10: Reserved – Must be reset to zero.

Bit 11: Read_Present_Configuration – If this bit is set when you enter Configuration Mode, the IANG1(E) responds by placing the present configuration data in the Network Input Data. You cannot write new configuration data to the unit while this bit is set. The format of the Configuration Data is given in the *Output Data Format* section of this chapter, starting on page 80.

Bit 12: Output_State_On_Network_Loss – 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 13: Read_Present_Configuration – “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		
13	**12**	**Description**
0	0	Output 1 stays in its last state if the network connection is lost.
0	1	Output 1 stays in its last state if the network connection is lost.
1	0	Output 1 turns off if the network connection is lost.
1	1	Output 1 turns on (conducts) if the network connection is lost.

Bit 14: Output_Functionality – “0” configures Output 1 to be a Fault Output. The output will conduct current until a fault occurs. “1” configures Output 1 to be a general purpose output whose state is determined by a bit in the Command Mode output data registers.

Bit 15: Drive_Enable – “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.

Notes on Other Configuration Words

- Changes to the Idle Current only take effect at the *end of the first move after re-configuration*.

Reference 6: Command Mode Output Data Format

Introduction

This chapter covers the formats of the Network Output Data used to command an IANG1(E). An IANG1(E) requires ten data registers to hold output data as well as ten data registers to hold input data. The format of this data is a mix of sixteen bit and thirty-two bit integers.

Note➤ The custom macros supplied by IDEC simplify the configuration and programming of the IANG1(E) units. The data in this reference is presented as background information to aid in troubleshooting.

Data Format

Thirty-two bit values are transmitted most significant word first (big endian). This is the data format specified by the Modbus specification and used by IDEC controllers. Table R5.1 below shows the format of the data as it is stored and transmitted.

	32 bit Integer Format	
Value	**First Word**	**Second Word**
12	0x0000	0x000C
-12	0xFFFF	0xFFF4
1,234,567	0x0012	0xD687
-7,654,321	0xFF8B	0x344F

Table R6.1 Position Data Format Examples

Command Bits Must Transition

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

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

The command bits are split between 2 sixteen bit words, Command Word 0 and Command Word 1. Only one bit in Command Word 0 can make a 0→1 transition at a time.

Output Data Format

The following table shows the format of the output network data words when writing command data to the IANG1(E).

Register	Function
1024	Command Word 0
1025	Command Word 1
1026	Command Parameters: Word meaning depends on the command set to the IANG1(E)
1027	
1028	
1029	
1030	
1031	
1032	
1033	

Figure R6.1 Command Data Format

Command Word 0

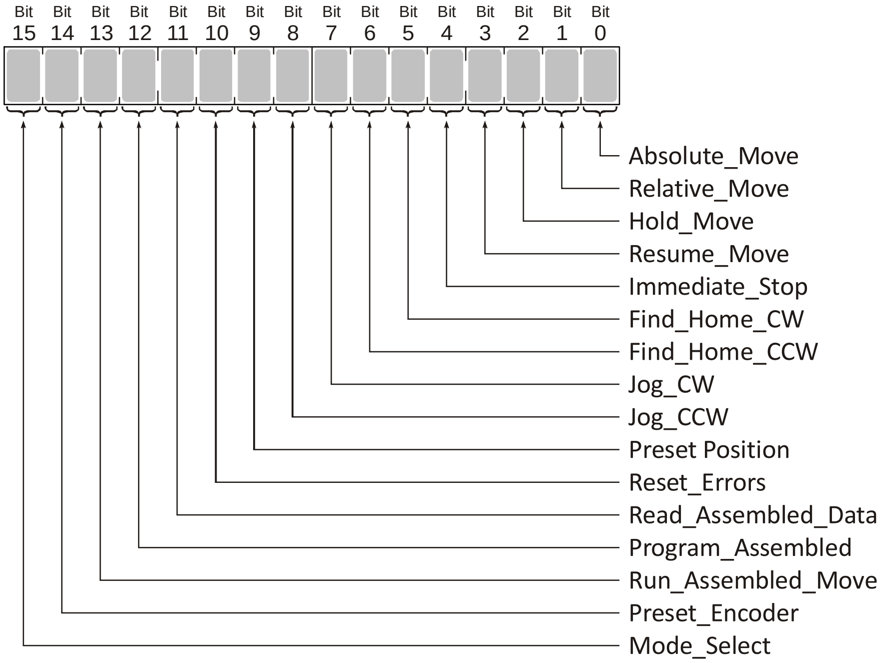


Figure R6.2 Command Word 0 Format

Bit 0: Absolute_Move – When this bit makes a 0→1 transition, the IANG1(E) will perform an Absolute Move using the values in the rest of the command data. The full explanation of an *Absolute Move* can be found starting on page 23.

Bit 1: Relative_Move – When this bit makes a 0→1 transition, the IANG1(E) will perform a Relative Move using the values in the rest of the command data. The full explanation of a *Relative Move* can be found starting on page 22.

Bit 2: Hold_Move – When this bit makes a 0→1 transition, the IANG1(E) will place a move in its hold state. 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 27.

Bit 3: Resume_Move – When this bit makes a 0→1 transition, the IANG1(E) will 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 27. Note that a move in its hold state need not be resumed. The move is automatically cancelled if another move is started in its place.

Bit 4: Immediate_Stop – When this bit makes a 0→1 transition, the IANG1(E) 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 5: Find_Home_CW – When this bit makes a 0→1 transition, the IANG1(E) will attempt to move to the Home Limit Switch in the clockwise direction. A full explanation of homing can be found in the *Homing an IANG1(E)* reference chapter starting on page 37.

Bit 6: Find_Home_CCW – When this bit makes a 0→1 transition, the IANG1(E) will attempt to move to the Home Limit Switch in the counter-clockwise direction. A full explanation of homing can be found in the *Homing an IANG1(E)* reference chapter starting on page 37.

Bit 7: Jog_CW – When this bit makes a 0→1 transition, the IANG1(E) will run a Jog Move in the clockwise direction. The full explanation of a *CW/CCW Jog Move* can be found starting on page 24.

- **Registration_Move – Command Word 1, Bit 7:** When this bit equals "0", and a Jog Move command is issued, it will run as a standard Jog Move. When this bit equals "1" and a Jog Move command is issued, the move will run as a Registration Move. The state of this bit cannot be changed while a Jog Move is in progress.

Bit 8: Jog_CCW – When this bit makes a 0→1 transition, the IANG1(E) will run a Jog Move in the counter-clockwise direction. The full explanation of a *CW/CCW Jog Move* can be found starting on page 24.

- **Registration_Move – Command Word 1, Bit 7:** When this bit equals "0", and a Jog Move command is issued, it will run as a standard Jog Move. When this bit equals "1" and a Jog Move command is issued, the move will run as a Registration Move. The state of this bit cannot be changed while a Jog Move is in progress.

Bit 9: Preset_Position – When this bit makes a 0→1 transition, the IANG1(E) will preset the Motor Position to the value stored in Output Words 2 and 3. The *Move_Complete* bit in the Network Input Data is reset to "0".

Bit 10: Reset_Errors – When this bit makes a 0→1 transition, the IANG1(E) will clear all existing command errors. The *Move_Complete* bit in the Network Input Data will be reset to "0". This bit does not clear a configuration error or the *Position_Invalid* status bit.

Bits 11 & 12: Read_Assembled_Data & Program_Assembled – 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 97.

Bit 13: Run_Assembled_Move – When this bit makes a 0→1 transition, the IANG1(E) will run the Assembled Move already stored in memory.

- **Assembled_Move_Type – Command Word 1, Bit 9:** This bit determines the type of move that is run. When this bit equals "0", a Blend Move is run. When this bit equals "1", a Dwell Move is run. When starting a Dwell Move, the Dwell Time is programmed in word 9 of the Command Data. The value is programmed in milliseconds and can range from 0 to 65,536.
- **Reverse_Blend_Direction – Command Word 1, Bit 4:** This bit is used to determine the direction that the Blend Move will be run in. When this bit equals "0", the Blend Move runs in the clockwise direction. When this bit equals "1", the Blend Move is run in the counter-clockwise direction. This bit is ignored if a Dwell Move is being run.

Bit 14: Preset_Encoder – When this bit makes a 0→1 transition, the IANG1(E) will preset the Encoder Position to the value stored in Output Words 2 and 3.

Bit 15: Mode_Select – "1" for Configuration Mode Programming "0" for Command Mode Programming.

Command Word 1

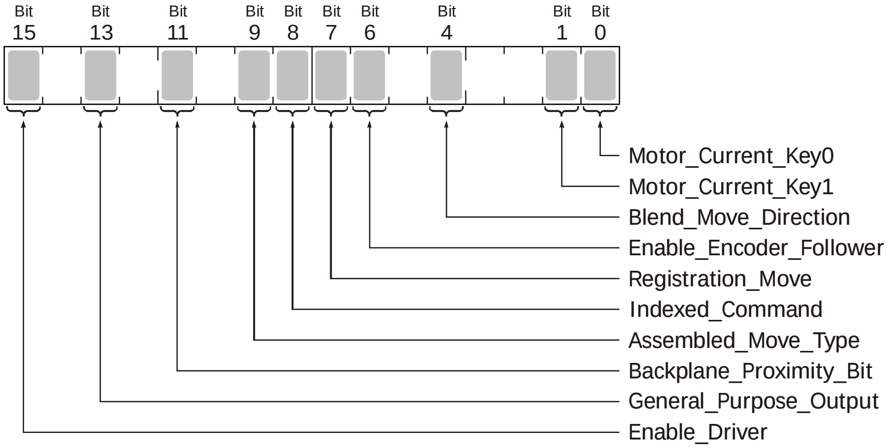


Figure R6.3 Command Word 1 Format

Bits 0, 1: Motor_Current_Key0, Motor_Current_Key1 – These bits can be used to set the motor current "on-the-fly" by setting the bits to a value of "10". When these bits are "10", the base motor current will be set to the value contained in word 8, the ninth word, of the command data.

- When these bits are set to "10", changes in output word 8 are acted upon immediately. The range of values for word 8 are 1 to 40. (0.1 to 4.0 amps) Values outside of this range are ignored and the last valid motor current setting will continue to be used.
- When these bits are set to "00", "01", or "11" the motor current is left at the last accepted value.
- Note that this procedure sets the base motor current. Any idle current reduction value set in the Configuration data will still affect the actual current delivered to the motor when the motor is idle.

Bits 2 & 3: Reserved – Must equal "0".

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 5: Reserved – Must equal "0".

Bit 6: Enable_Encoder_Follower – Set to "1" to put the IANG1(E) in Encoder Follower mode. Set to "0" for normal operation. A full description of *Encoder Follower* starts on page 27.

Bit 7: Registration_Move – When this bit equals "0", and a Jog Move command is issued, it will run as a standard Jog Move. When this bit equals "1" and a Jog Move command is issued, the move will run as a Registration Move.

Bit 8: Indexed_Command – If this bit is set when a move command is issued, the IANG1(E) will not run the move immediately, but will instead wait for an inactive-to-active transition on an input configured as a *Start Indexer Move* input. The move command data, including this bit, must remain in the network output registers while performing an Indexed Move.

Bit 9: Assembled_Move_Type – When this bit equals “0”, a Blend Move is started when the Run Assembled Move bit, (Command Word 1, Bit 13) makes a 0→1 transition. When this bit equals “1”, a Dwell Move is started on the transition. The direction of a Blend Move is controlled by the Reverse_Blend_Direction bit, (Command Word 1, Bit 4). In a Dwell Move, the Dwell Time between segments is programmed in Word 9 of the command data.

Bit 10: Reserved – Must equal “0”.

Bit 11: Backplane_Proximity_Bit – When the IANG1(E) is configured to use the Backplace_Proximity_Bit, the unit will ignore the state of the Home Input as long as this bit equals “0”. This bit must equal “1” before a transition on the Home Input can be used to home the machine. Further information on using the Backplace_Proximity_Bit can be found in the *Profile with Proximity Input* section found on page 39.

Bit 12: Reserved – Must equal “0”.

Bit 13: General_Purpose_Output – When the output is configured as a general purpose output point instead of the Fault Output, this bit controls the state of the output. When this bit equals a “1”, the output is on and conducts current.

Bit 14: Reserved – Must equal “0”.

Bit 15: Enable_Drive – “0” to disable the motor current, “1” to enable motor current. A valid configuration must exist in the IANG1(E) before the drive can be enabled.

Command Blocks

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

Absolute Move

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0001
1025	*Command Word 1*		See pg. 85
1026	Abs. Target Position: Upper 16 bits	Steps	Combined value between –8,388,607 and +8,388,607
1027	Abs. Target Position: Lower 16 bits		
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.2 Absolute Move Command Block

Relative Move

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0002
1025	*Command Word 1*		See pg. 85
1026	Rel. Target Position: Upper 16 bits	Steps	Combined value between –8,388,607 and +8,388,607
1027	Rel. Target Position: Lower 16 bits		
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.3 Relative Move Command Block

Hold Move

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0004
1025	*Command Word 1*		See pg. 85
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	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Unused		See Note Below

Table R6.4 Hold Move Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Resume Move

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0008
1025	*Command Word 1*		See pg. 85
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.5 Resume Move Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from 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.

Immediate Stop

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0010
1025	*Command Word 1*		See pg. 85
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	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Unused		See Note Below

Table R6.6 Immediate Stop Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Find Home CW

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0020
1025	*Command Word 1*		See pg. 85
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.7 Find Home CW Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Find Home CCW

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0040
1025	*Command Word 1*		See pg. 85
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.8 Find Home CCW Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Jog CW

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0080
1025	*Command Word 1*		See pg. 85 Bit 7 must equal “0”
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.9 Jog Move CW Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Registration Move CW

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0080
1025	*Command Word 1*		See pg. 85 Bit 7 must equal “1”
1026	Stopping Distance: Upper 16 bits	Steps	Combined value between 0 and +8,388,607
1027	Stopping Distance: Lower 16 bits		
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Min. Reg. Move Distance: Upper 16 bits	Steps	Combined value between 0 and +8,388,607
1033	Min. Reg. Move Distance: Lower 16 bits		

Table R6.10 Registration Move CW Command Block

Jog CCW

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0100
1025	*Command Word 1*		See pg. 85 Bits 7 must equal “0”
1026	Unused		See Note Below
1027	Unused		See Note Below
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the Configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.11 Jog CCW Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Registration Move CCW

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0100
1025	*Command Word 1*		See pg. 85 Bits 7 must equal “1”
1026	Stopping Distance: Upper 16 bits	Steps	Combined value between 0 and +8,388,607
1027	Stopping Distance: Lower 16 bits		
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Min. Reg. Move Distance: Upper 16 bits	Steps	Combined value between 0 and +8,388,607
1033	Min. Reg. Move Distance: Lower 16 bits		

Table R6.12 Registration Move CCW Command Block

Encoder Follower

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0080 or 0x0100
1025	*Command Word 1*		See pg. 85 Bits 6 must equal “1”
1026	Encoder Follower Multiplier		1 to 255
1027	Encoder Follower Divisor		1 to 255
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Acceleration Jerk		0 to 5000

Table R6.13 Encoder Follower Move Command Block

Preset Position

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0200
1025	*Command Word 1*		See pg. 85
1026	Position Preset Value: Upper 16 bits	Steps	Combined value between –8,388,607 and +8,388,607
1027	Position Preset Value: Lower 16 bits		
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 R6.14 Preset Position Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Presetting the position resets the *Position_Invalid* and *Move_Complete* status bits in the input data registers.

Clear Errors

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x0400
1025	*Command Word 1*		See pg. 85
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 R6.15 Clear Errors Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Resetting errors will also reset the *Move_Complete* status bit in the Network Input Data. Resetting errors will not reset the *Position_Invalid* or *Configuration_Error* bits.

Run Assembled Move

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x2000
1025	*Command Word 1*		See pg. 85 Blend Move: Bit 9 = “0”, Dwell Move: Bit 9 = “1” If Blend Move, CW move when Bit 4 = “0”, CCW move when Bit 4 = “1”.
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	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Unused with Blend Move Dwell Time with Dwell Move	milliseconds	0 to 65,535

Table R6.16 Run Assembled Move Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Preset Encoder Position

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x4000
1025	*Command Word 1*		See pg. 85
1026	Encoder Preset Value: Upper 16 bits	Steps	Combined value between –8,388,607 and +8,388,607
1027	Encoder Preset Value: Lower 16 bits		
1028	Unused		See Note Below
1029	Unused		See Note Below
1030	Unused		See Note Below
1031	Unused		See Note Below
1032	Motor Current	0.1 amps	1 to 40. (0.1 to 4.0 amps) Ignored if bits 1 and 0 of Command Word 1 are not set to ‘10’.
1033	Unused		See Note Below

Table R6.17 Preset Encoder Position Command Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Programming Blocks

The following blocks are used to program an Assembled Move. Both of the move types, 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 Address	Function	Units	Range
1024	*Command Word 0*		0x0800
1025	*Command Word 1*		See pg. 85
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 R6.18 Assembled Move First Programming Block

Unused words are ignored by the IANG1(E) and can be any value, including parameter values from the previous command.

Once the first block is transmitted, the IANG1(E) responds by setting bits 8 and 9 in Status Word 0. (See *Status Word 0 Format* starting on page 72.) Once these are set, you can then start transmitting Segment Blocks.

Segment Block

Modbus Address	Function	Units	Range
1024	*Command Word 0*		0x1800
1025	*Command Word 1*		See pg. 85
1026	Rel. Target Position: Upper 16 bits	Steps	Combined value between –8,388,607 and +8,388,607
1027	Rel. Target Position: Lower 16 bits		
1028	Programmed Speed: Upper 16 bits	Steps/Second	Combined value between the configured Starting Speed and 2,999,999
1029	Programmed Speed: Lower 16 bits		
1030	Acceleration	Steps/ms/sec	1 to 5000
1031	Deceleration	Steps/ms/sec	1 to 5000
1032	*Reserved*		Must equal zero for compatibility with future releases.
1033	Acceleration Jerk		0 to 5000

Table R6.19 Assembled Move Segment Programming Block

Note that each Segment Block starts with bits 11 and 12 in Command Word 0 set to “1” (0x1800). When the IANG1(E) sees bit 12 of Command Word 0 set, it will accept the block and reset bit 9 in Status Word 0. When your program sees this bit reset, it must respond by resetting bit 12 of Command Word 0. The IANG1(E) will respond to this by setting bit 9 in Status Word 0 and the next Segment Block can be written to the module. You can write a maximum of sixteen Segment Blocks for each Assembled Move.

Notes

Reference 7: Assembled Moves

Introduction

The IANG1(E) contains functionality that is not programmable through custom macros at this time. This reference section describes this additional functionality.

Assembled Moves

All of the moves explained so far must be run individually to their completion or must be stopped before another move can begin. The IANG1(E) also gives you the ability to pre-assemble more complex profiles from a series of relative moves that are then run with a single command. Each Assembled Move consists of two to sixteen segments. Assembled Moves are programmed through a hand shaking protocol that uses the input and output registers assigned to the unit. The protocol is fully described in the *Assembled Move Programming* section on page 97.

Two types of Assembled Moves exist in an IANG1(E):

- **Blend Move -** A Blend Move gives you the ability to string multiple relative moves together and run all of them sequentially without stopping the shaft between moves. A Blend Move can be run in either direction, and the direction is set when the command is issued. The direction of motion cannot be reversed within a Blend Move.
- **Dwell Move -** A Dwell Move gives you the ability to string multiple relative moves together, and the IANG1(E) will stop between each move for a programed *Dwell Time*. Because motion stops between each segment, a Dwell Move allows you to reverse direction during the move.

Blend Move

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

1) Each segment of the Blend Move must be written to the IANG1(E) before the move can be initiated.
    - The IANG1(E) supports Blend Moves with up to sixteen segments.
2) Each segment is programmed as a relative move. Blend Moves cannot be programmed with absolute coordinates.
3) All segments run in the same direction. The sign of the target position is ignored and only the magnitude of the target position is used. The move's direction is controlled by the bit pattern used to start the move. If you want to reverse direction during your move, consider using the *Dwell Move* which is explained starting on page 96.
4) The Programmed Speed of each segment must be greater than or equal to the Starting Speed.
5) The Programmed Speed can be the same between segments. This allows you to chain two segments together.
6) For all segments except for the last one, the programmed position defines the end of the segment. For the last segment, the programmed position defines the end of the move.
7) Once you enter a segment, that segment's programmed acceleration and deceleration values are used to change the speed of the motor.
8) The blend segment must be long enough for the acceleration or deceleration portions of the segment to occur. If the segment is not long enough, the motor speed will jump to the speed of the next segment without acceleration or deceleration.

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

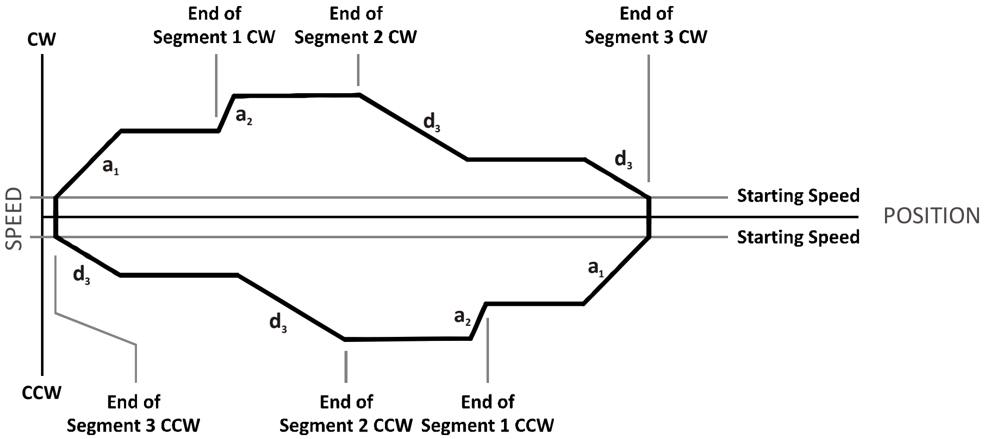


Figure R7.1 Blend Move

Note➤ 1) The deceleration value programmed with segment 3 is used twice in the segment. Once to decelerate from the Programmed Speed of segment 2 and once again to decelerate at the end of the move.

2) 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.

3) The Blend Move is stored in the internal memory of the IANG1(E) and can be run multiple times once it is written to the unit. The Blend Move data stays in memory until power is removed, the unit is sent new Configuration Data, or a new Blend or Dwell Move is written to the unit.

4) 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 IANG1(E) decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. A Blend Move that is brought to a controlled stop with the Hold_Move bit cannot be restarted. The use of the Hold_Move bit is explained in the *Controlling Moves In Progress* section starting on page 27.

Immediate Stop Conditions

- The Immediate_Stop bit makes a 0→1 transition in the Network Output Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW or 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 *Clear 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 *Clear Errors* command does not have to be issued.

Dwell Move

A Dwell Move gives you the ability to string multiple relative moves together and run all of them sequentially with a single start condition. Like a Blend Move, a Dwell Move is programmed into an IANG1(E) as a series of relative moves before the move is started.

Unlike a Blend Move, the motor is stopped between each segment of the Dwell Move for a programed *Dwell Time*. The Dwell Time is programmed as part of the command that starts the move. The Dwell Time is the same for all segments. Because the motor is stopped between segments, the motor direction can be reversed during the move. The sign of the target position for the segment determines the direction of motion for that segment. Positive segments will result in clockwise shaft rotation while a negative segment will result in a counter-clockwise shaft rotation.

The following figure shows a drilling profile that enters the part in stages and reverses direction during the drilling operation so chips can be relieved from the bit. You *could* accomplish this Dwell Move with a series of six relative moves that are sent down to the IANG1(E) sequentially. The two advantages of a Dwell Move in this case are that the IANG1(E) will be more accurate with the Dwell Time then you can be in your control program, and Dwell Moves simplify your program's logic.

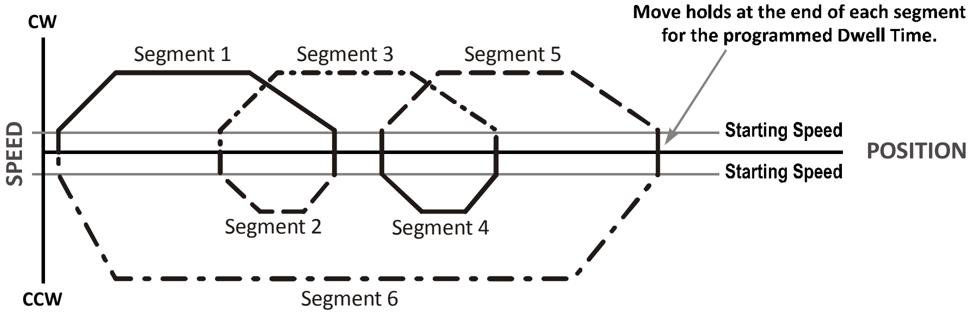


Figure R7.2 Dwell Move

Note➤ 1) You do not have to preset the position or home the machine before you using a Dwell Move. The Dwell Move is based on Relative Moves, and can be run from any location.

2) The Dwell Move is stored in the internal memory of the IANG1(E) 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 IANG1(E).

Controlled Stop Conditions

- The move completes without error.
- You toggle the Hold_Move control bit in the Network Output Data. When this occurs, the IANG1(E) decelerates the move at the deceleration rate of the present segment to the Starting Speed and ends the move. Note that your final position will most likely not be the one you commanded. A Dwell Move that is brought to a controlled stop with the Hold_Move bit cannot be restarted.

Immediate Stop Conditions

- The Immediate_Stop bit makes a 0→1 transition in the Network Output Data.
- A positive transition on an input configured as an E-Stop Input.
- A CW or 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 *Clear 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 *Clear Errors* command does not have to be issued.

Controlling an Assembled Move In Progress

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

Assembled Move Programming

All of the segments in a Blend or Dwell Move must be written to the IANG1(E) before the move can be run. Segment programming is controlled with two bits in the Network Output Data and two bits in the Network Input Data. Blend and Dwell Moves are programmed in exactly the same way. When you start the move, a bit in the command data determines which type of Assembled Move is run. In the case of a Blend Move, the sign of each segment's Target Position is ignored and all segments are run in the same direction. In the case of a Dwell Move, the sign of each segment's Target Position determines the direction of the segment. For Dwell Moves, the Dwell Time is sent to the IANG1(E) as part of the command.

Control Bits – Output Data

- **Program_Assembled bit (Command Word 0, bit 12) –** A 0→1 transition on this bit tells the IANG1(E) that you want to program a Blend or Dwell Move Profile. The IANG1(E) will respond by setting the *In_Assembled_Mode* bit in the Network Input Data. At the beginning of the programming cycle, the IANG1(E) will also set the *Waiting_For_Assembled_Segment* bit to signify that it is ready for the first segment.
- **Read_Assembled_Data bit (Command Word 0, bit 11) –** A 0→1 transition on this bit tells the IANG1(E) that the data for the next segment is available in the remaining data words.

Control Bits – Input Data

- **In_Assembled_Mode bit (Status Word 0, bit 8) –** The IANG1(E) sets this bit to signal that it is ready to accept segment programming data in the remaining output data words. The actual transfer of segment data is controlled by the *Waiting_For_Assembled_Segment* and *Read_Assembled_Data* bits.
- **Waiting_For_Assembled_Segment bit (Status Word 0, bit 9) –** A 0→1 transition on this bit from the IANG1(E) is the signal to the host that the IANG1(E) is ready to accept the data for the next segment.

Programming Routine

1) The FC6A sets the *Program_Assembled* bit in the Output Registers.
2) The IANG1(E) responds by setting both the *In_Assembled_Mode* and *Waiting_For_Assembled_Segment* bits in the Input Registers.
3) When the FC6A detects that the *Waiting_For_Assembled_Segment* bit is set, it writes the data for the first segment in the Output Registers and sets the *Read_Assembled_Data* bit.
4) The IANG1(E) checks the data, and when finished, resets the *Waiting_For_Assembled_Segment* bit. If an error is detected, it also sets the *Command_Error* bit.
5) When the FC6A detects that the *Waiting_For_Assembled_Segment* bit is reset, it resets the *Read_Assembled_Data* bit.
6) The IANG1(E) detects that the *Read_Assembled_Data* bit is reset, and sets the *Waiting_For_Assembled_Segment* bit to signal that it is ready to accept data for the next segment.
7) Steps 3 to 6 are repeated for the remaining segments until the entire move profile has been entered. The maximum number of segments per profile is sixteen.
8) After the last segment has been transferred, the FC6A exits Assembled Move Programming Mode by resetting the *Program_Assembled* bit.
9) The IANG1(E) resets the *In_Assembled_Mode* and the *Waiting_For_Assembled_Segment* bits.

Commanding an Assembled Move

Three bits in the output data are used to start an Assembled Move.

- **Run_Assembled_Move bit (Command Word 0, bit 13) –** A 0→1 transition on this bit commands an Assembled move to begin. Bits in Command Word 1 control the type of move and its direction.
- **Assembled_Move_Type bit (Command Word 1, bit 9) –** When this bit equals “0”, a Blend Move is started on the 0→1 transition when of the Run_Assembled_Move bit. 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.
- **Blend_Move_Direction bit (Command Word 1, bit 4) –** This bit determines the direction of rotation of a Blend Move. Set to “0” for a clockwise Blend Move, ‘1’ for a counter-clockwise Blend Move.

Optional Task A: Configure Your Network Interfaces

Introduction

Before communicating with an ANG1(E) using a personal computer, you may need to adjust the network setting on the computer. This task section give you information on configuring your computer's network interfaces.

A.1 Firewall Settings

Firewalls are hardware devices or software that prevent unwanted network connections from occurring. Firewall software is present in Windows XP and above and it may prevent your computer for communicating with the IANG1(E). Configuring your firewall to allow communication with the IANG1(E) is beyond the scope of this manual.

It may be necessary to temporarily disable any firewall software while using the Ethernet Tool. This is typically done from the Windows Control Panel. You should re-enable the firewall once you have finished using the tool.

A.2 Disable All Unused Network Interfaces

Routing and default gateway setting on your computer can interfere with the proper operation of the Ethernet Tool. The software uses broadcast packets to locate devices on the network, and sometimes these packets are sent out through the default gateway instead of the interface attached to the IANG1(E). The easiest way to avoid this problem is to temporarily disable all network interfaces that are not attached to the stepper drive.

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

A.3 Configure Your Network Interface

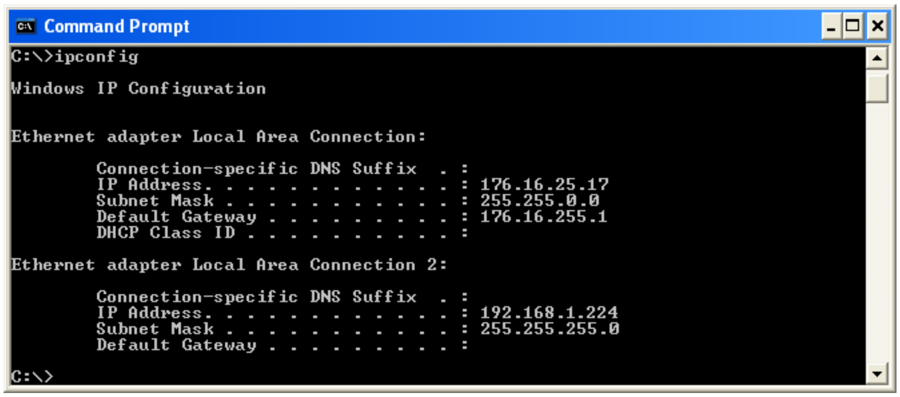
Before you can communicate with the IANG1(E), your network interface must be on the same subnet as the module.

Note➤ The rest of this procedure assumes you are using the 192.168.1.xxx subnet. If you are not, you will have to adjust the given network addresses accordingly.

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

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

A DOS like terminal will open. Type in 'ipconfig', press [Enter] on the keyboard and the computer will return the present Address, Subnet Mask, and Default Gateway for all of your network interfaces. If your present address is 192.168.1.xxx, where 'xxx' does not equal 50, and your subnet mask is 255.255.255.0, then you are ready to configure your IANG1(E). Figure A.1 shows the output of an ipconfig command that shows the "Local Area Connection 2" interface on the 192.168.1.xxx subnet.



```
C:\>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

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

Ethernet adapter Local Area Connection 2:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.224
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

C:\>
```

Figure A.1 ipconfig Command

If your present address in not in the 192.168.1.xxx range, type in 'ncpa.cpl' at the command prompt and hit [Enter] on the keyboard.

- ➤ For Vista and Windows 7, this open the *Network Connections* window. Double click on the appropriate interface. In the window that opens, select "Internet Protocol Version 4 (TCP/IP v4)" from the list and then click on the [Properties] button.
- ➤ For Windows 8 and 10, this open the *Network Connections* window. Double click on the appropriate interface. In the window that opens, select "Internet Protocol Version 4 (TCP/IP v4)" from the list and then click on the [Properties] button.

Set the address and subnet mask to appropriate values. (192.168.1.1 and 255.255.255.0 will work for an IANG1(E) that has factory default settings.) The default gateway and DNS server settings can be ignored.

A.4 Test Your Network Interface

Going back to the terminal you opened in the last step, type in ‘ping aaa.bbb.ccc.ddd’ where ‘aaa.bbb.ccc.ddd’ in the IP address of the IANG1E that is the network connection for the AnyNET-I/O Stack. The computer will ping the unit and the message “Reply from aaa.bbb.ccc.ddd: bytes=32 time<10ms TTL=255” should appear four times.

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

- You set a new IP address, but have not yet cycled power to the IANG1E that is the network connection for the AnyNET-I/O Stack.
- You did not enter the correct address in the ping command.
- The IP address of the IANG1E that is the network connection for the AnyNET-I/O Stack, is not set correctly.
- The IANG1E that is the network connection for the AnyNET-I/O Stack and the computer are not on the same subnet.

Notes

AMCI

by

IDEC CORPORATION