GE Fanuc Automation

Programmable Control Products

Stepping Motor Cube with DeviceNet

Reference Guide

GFK-2207 *August 2002*

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

The following are trademarks of GE Fanuc Automation North America, Inc.

Alarm Master	Genius	PROMACRO	Series Six
CIMPLICITY	Helpmate	PowerMotion	Series Three
CIMPLICITY 90–ADS	Logicmaster	PowerTRAC	VersaMax
CIMSTAR	Modelmaster	Series 90	VersaPro
Field Control	Motion Mate	Series Five	VuMaster
GEnet	ProLoop	Series One	Workmaster

Content of this Manual

This manual provides specifications, installation, and operation information for the IC800MCUB12xx0XD Stepping Motor Cube™ with DeviceNet capability.

Related Publications

The following publications are available at http://www.gefanuc.com/support/plc/m-MotionSolutions.htm.

Generation D RTOS Programming Manual, GFK-2205

DeviceNet Reference Guide, GFK-2208

IMC Hardware Manual, GFK-2201

Target® ARS Field Service Manual, GFK-2200

IMCjr Hardware Manual, Pub 330

DeviceNet Reference Guide (for Early Firmware Revisions), Pub 305

For an in-depth DeviceNet resource, please consult the *DeviceNet Specification*, release 2.0, Errata 3, published by the Open DeviceNet Vendor Association (www.odva.org).

Contents

Contents

Chapter 1

Specifications and Installation

Specifications

Figure 1.1: Specifications for Stepping Motor Cube with DeviceNet

DC Input Power Requirements		
Drive Input Voltage Operating Range	24 Vdc +/- 20% @ 2.4 Amps max [(a)] 48 Vdc +/- 10% @ 2.4 Amps max [(a)]	
Overvoltage Threshold	54 Vdc +/- 2 Vdc	
Undervoltage Threshold	18 Vdc +/- 2 Vdc	
Output Power		
Voltage range	17 to 38 Vrms 2 phase	
Frequency	0 - 8 KHz fundamental (16.4 KHz PWM)	
Current[(b)]	3 A rms per phase	
Environmental Specifications		
Operating Temperature, Free Air Ambient	0 to 50 °C	
Storage and Shipping Temperature	-40 to 80 °C	
Enclosure Type	open	
DeviceNet Communication Specifications		
Number Available	1 per unit	
Functions Supported	I/O slave poll messaging, UCMM, explicit messaging	
Number of Nodes	64 maximum	
Input Power Requirements	11-25 Vdc @ 120 mA maximum	
Communication Rate	125, 250, or 500 Kbaud	
Length of Drop Line	20 feet maximum	
Length of Trunk Line	thin cable:	328 feet maximum
	thick cable:	328 feet maximum @ 500 Kbaud
		820 feet maximum @ 250 Kbaud
		1,640 feet maximum @ 125 Kbaud
Steps per Revolution		
Step Size	50,000 steps/rev	

Notes for figure 1.1:
(a) DC input power has undervoltage and overvoltage detection.
(b) The outputs are provided with internal overload protection. IC800MCUB12160XD is 2.5A rms per phase.

Installation

Location

Location of the Stepping Motor Cube is important to achieve proper performance and operating life. The unit is designed with "open" construction. The unit must be installed in an enclosure that protects personnel from contact with wiring terminals and provides a pollution degree 2 environment that protects the unit from:

- Corrosive gases or liquids
- Vibration
- Conductive pollution including extreme or condensing humidity and airborne metallic particles
- Accidental contact by persons using the equipment
- Temperature extremes beyond the equipment ratings.

Overtemperature

The Stepping Motor Cube has an overtemperature detection circuit. When the temperature of the logic electronics (located inside the cube enclosure) reaches 80°C +/- 5%, the drive will fault and become disabled. Built-in 5 to 10 degree Celsius hysteresis requires that the cube must cool to below 70°C before it can be enabled following an overtemperature condition.

Wiring

Wiring diagrams for Stepping Motor Cube models are included in Chapter 5, *User Connections*.

General Wiring Considerations

All power must be in accordance with Class I, Division 2 wiring methods as defined in Article 501-4(b) of the National Electrical Code, NFPA 70 for installations within the United States, or as specified in Section 18-152 of the Canadian Electrical Code for installation within Canada.

Attach wiring connections for the main circuit according to Table 1.2 while observing the following **cautions:**

- ! **Never** connect AC mains power to any terminals.
- ! **Never** allow wire leads to contact the enclosure.
- ! **Never** operate the unit without an earth ground.
- ! **WARNING--**When using this equipment in a Hazardous (classified) location:
  - A. **WARNING--*Explosion hazard*--**substitution of components may impair suitability for Class I, Division 2;
  - B. **WARNING--*Explosion hazard*--**when in hazardous locations, turn off power before replacing or wiring modules;
  - C. **WARNING--*Explosion hazard*--**do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.

Figure 1.2: Motor Cube Motor Power Wiring

Pin	Description	Connect to	Wire Size AWG[(b)]
1	Vmotor +	24 or 48 Vdc Motor Power[(a)]	22
2	Frame	Earth Ground	22
3	Vmotor – (Common)	24 or 48 Motor Power Common[(a)]	22

Notes for figure 1.2:

(a) DC input power has undervoltage and overvoltage detection. Overvoltage threshold is 54 Vdc +/- 2 Vdc. Undervoltage threshold is 18 Vdc +/- 2 Vdc.

(b) Suggested maximum AWG size (i.e., minimum wire diameter) for stranded copper wire. Consult *National Electrical Code Handbook* ampacities tables for proper wire size.

Input Mains Wiring and Grounding

The DC power input connections are made to the connector located on the top of the Stepping Motor Cube. The unit is designed to operate with motor power voltages of 24 or 48 Vdc.

Motor and Drive Wiring

The motor and drive components of the Stepping Motor Cube are factory wired and must not be disconnected. Do not attempt to remove the motor connector or to connect the drive electronics to any external motor.

Chapter 2

Basic Setup

Getting Started

This section describes the basic setup procedure for the Stepping Motor Cube. Follow this procedure to connect and configure the Stepping Motor Cube.

Configure Motor Cube for DeviceNet

Set Node Address. Ensure that power to the unit is off. Point the arrow labeled *MSD* to the first digit of the node address (sometimes called MAC ID). Point the arrow labeled *LSD* to the second digit of the node address. Figure 2.1 shows a node address set to 33. Node addresses may be any number from 00 through 63 that is not used by another device on your network.

Set Data Rate. DeviceNet can run a network data rate, or baud rate, of 125, 250, or 500 Kbaud. Point the arrow labeled *DATA RATE (KB)* to the desired data rate. All nodes on the network must be at the same network data rate.

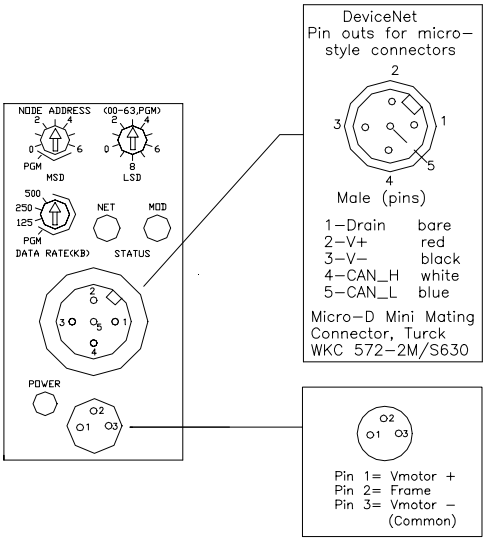


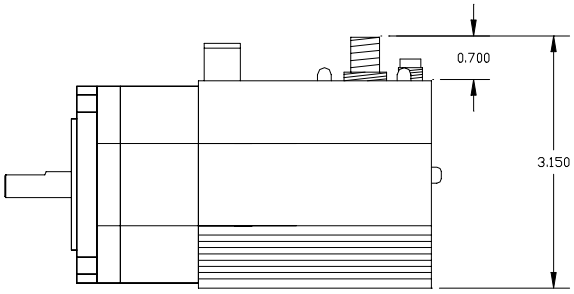
Figure 2.1: DeviceNet Connections on the Stepping Motor Cube

Connect DeviceNet Hardware

Follow the instructions provided with your non-GE Fanuc DeviceNet hardware to connect your trunk line, drop lines, Ts, terminating resistors, and power supplies.

Connect the Stepping Motor Cube to DeviceNet

Each Stepping Motor Cube requires its own dropline. Stepping Motor Cubes with DeviceNet communication feature a standard microconnector on the top of the unit to join the Stepping Motor Cube to the DeviceNet trunkline. Connect the microconnector end of the DeviceNet dropline to the DeviceNet port on the top of the unit. See figure 2.2 for the connector location.



Connections on the Stepping Motor Cube

Connect and Apply DC Power

The Stepping Motor Cube is designed to operate with motor power voltages of 24 or 48 Vdc. Connect the female, three-pin power cable to the power connector on the top of the unit. Connect the opposite end of the cable to your 24 or 48 Vdc power supply. Internal logic power is supplied on the DeviceNet cable [V + (pin 2, red) and V – (pin 3, black)].

Configure the System

The Stepping Motor Cube ships from the factory with default values for several parameters. The drive's continuous current output, for example, is set to the default of 60% of the maximum continuous current rating of the drive. Use attribute 101 of the Position Controller Object to set continuous current to the desired setting.

The power save current also defaults to 60% and may be adjusted to the desired setting using attribute 102 of the Position Controller Object. See figure 2.21 for a complete list of attributes supported by the Stepping Motor Cube.

Using Stall Detection with the DeviceNet Stepping Motor Cube

Setting Up Stall Detection

The Stepping Motor Cube powers up with stall detection inactive. The stall velocity threshold is 4 RPS for all models. Users can activate stall detection by setting the five Position Controller attributes described in the following table:

Figure 2.3: Position Controller Attributes for Stall Detection

Attribute ID	Attribute Description	Attribute Type	Attribute Values	Default	Units
47	Following error	Boolean	0, 1	0	--
100	Fault code	DINT	0 – $FFFFFFFF_{16}$	--	--
111	Stall velocity threshold	DINT	50,000 – 2,500,000	2,500,000	pulses/sec
112	Bus voltage	BYTE	0, 24 - 50	0	Volts
113*	Stall sensitivity	INT	1 - 2,000	200	ms

* Added in revision 1.1 on above firmware.

Complete the following steps to activate stall detection:

1. Determine the DC motor voltage applied to the Stepping Motor Cube. It may be necessary to measure the voltage with a multimeter if there is some uncertainty about the accuracy of the supply rating. Measure the motor voltage while the Stepping Motor Cube is enabled for the most accurate data.

2. Round the DC voltage to the nearest integer. During the initialization section of your code (or elsewhere if preferred) Set Position Controller Attribute 112 with DC motor voltage. For tips on sending this message or others, please refer to the *Generation D Real-Time Operating System DeviceNet Reference Guide*, GFK-2208.

3. Determine the minimum velocity where stall detection is desired. The controller will be sensing for a stall condition only while the actual velocity is above the Stall Velocity Threshold. To detect stall, the profile must have an actual velocity above this threshold for a time greater than the time set in the Stall Sensitivity attribute (default 200ms). For example, if the profile ramps to a velocity of 10 RPS, and is over 8 RPS for a minimum of 500ms, selecting a stall velocity threshold around 5 RPS would be good starting point. Of course, this parameter can be tweaked in for the application. Set Position Controller attribute 111 accordingly.

4. Please note that stall detection can be used in both position and velocity modes. Also, it is recommended that the continuous current setting be at least 25.0% for stall to operate, as lower settings tend to add errors.

How to Determine if a Stall Occurred

When the Stepping Motor Cube detects a valid stall condition (Attributes 111-113 are set correctly), the controller will fault. The fault status for stall can be retrieved implicitly (I/O), or explicitly. Implicitly, bit 5 of byte 2 in any response message will have the FE (Following Error)

fault bit. If FE Fault=0, then a stall has not been detected; if FE Fault=1, then a stall occurred. Explicitly, stall can be check in two ways. First, by accessing Position Controller Attribute 47, which is the attribute for Following Error. Again, the fault flag is 1=stalled, 0=no stall. Also, this flag is present in the Fault Code attribute (Position Controller Attribute 100). In this attribute, the FE Fault is bit number 21.

Tips on Using Stall Detection

The goal of using stall detection should be to have 100% accurate detection. This is possible for most applications. Of course, there are always tricks to a trade, and following are some suggestions to optimize your use of stall detection.

Tweaking the Lowest Possible Velocity Threshold

In some applications, the profile never achieves a high velocity. In general, stall detection becomes less accurate under 4 RPS. It is possible, however, to tweak in stall detection to perform well at 1 RPS. For the best accuracy at lower velocities, increase the stall sensitivity time. This can be a trade-off if the motor performs very short position moves while at low velocities. It is important to determine how long the motor will be above the stall detection threshold (look at velocity vs. time curve for the profile). Ensure that the motor achieves a velocity above the stall velocity threshold for at least twice the time set in the Stall Sensitivity parameter. On a different note, motors with a higher torque constant will provide better results at lower velocities. For example, the Whedco 1231 motor will produce more accurate results at lower velocities than a Whedco 1216 motor.

Fastest Possible Detection Time

You can achieve the fastest possible detection time in two ways: 1) reduce the Stall Sensitivity time so that the controller will react more quickly to a stall. Note that using too low a Stall Sensitivity time value may cause the Stepping Motor Cube to falsely detect stalls. 2) use a higher Target Velocity with a higher Stall Velocity Threshold. This works together with the Stall Sensitivity because velocities over 4 RPS provide accurate results, and thus the Stall Sensitivity time can be lowered.

Selective Use

If there is only one particular move in which you are interested in detecting stall, you can selectively activate the stall feature by using Motor Voltage attribute 112 as an enable for stall detect. By setting this attribute to 0, stall detection is disabled. In other words, you can set up the Stall Velocity Threshold and the Stall Sensitivity time for the application at hand. Then, only before certain moves, the Motor Voltage can be set to its correct value. When the move is complete, set the Motor Voltage to 0. This will not change the Stall Velocity or the Stall Sensitivity.

Use On-board LEDs to Diagnose System Faults

Status LEDs labeled *NET* and *MOD* are located on the top of the Stepping Motor Cube, just above the DeviceNet connector. Figure 2.4 details the LED diagnostics.

Figure 2.4: LED States

NET = Network LED		
Off	Not Powered/ Not Online	Device is not online.
Flashing Green	Online, Not Connected	Device is online but has no connections in the established state.
Green	Link OK; Online, Connected	The device is online and has connections in the established state.
Flashing Red	Connection Time-Out	The I/O Connection is in the Timed-Out state.
Red	Critical Link Failure	Failed communication device. The device has detected an error that has rendered it incapable of communication on the network. Check for another device with the same node address and ensure that all devices are set to the same data rate.
MOD = Module Status LED		
Off	Power Off	No power is applied to device.
Flashing Green-Red	Device Self-Test	Device is in self-test.
Green	Device Operational	Device is operational.
Flashing Red	Recoverable Fault	Device has detected a recoverable fault.
Red	Unrecoverable Fault	Device has detected an unrecoverable fault; may need to be replaced.

DeviceNet Command Messages Supported by Stepping Motor Cube

See page 16 for the complete DeviceNet I/O handshaking between command and response messages for each of the following command messages types.

Figure 2.5: Type 01 hex Target Position

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	Hard Stop	Smooth Stop	Direction (V. Mode)	Incremental	n/a	Load Data/ Start Profile
1	n/a							
2	Command Axis Number = 001_2			Command Message Type = 01				
3	Response Axis Number = 001_2			Response Message Type				
4	Target Position Low Byte							
5	Target Position Low Middle Byte							
6	Target Position High Middle Byte							
7	Target Position High Byte							

Target Position - Command Message Type 01 hex

This double word defines the profile move's Target Position in pulses when the Load Data/Start Profile bit transitions from zero to one.

Figure 2.6: Type 02 hex Target Velocity

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	Hard Stop	Smooth Stop	Direction (V. Mode)	Incremental	n/a	Load Data/ Start Profile
1	n/a							
2	Command Axis Number = 001_2			Command Message Type = 02				
3	Response Axis Number = 001_2			Response Message Type				
4	Target Velocity Low Byte							
5	Target Velocity Low Middle Byte							
6	Target Velocity High Middle Byte							
7	Target Velocity High Byte							

Target Velocity - Command Message Type 02 hex

This double word defines the Profile Move's Target Velocity in pulses/sec, when the Load Data /Start Profile bit transitions from zero to one

Figure 2.7: Type 03 hex Acceleration

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	Hard Stop	Smooth Stop	Direction (V. Mode)	Incremental	n/a	Load Data/ Start Profile
1	n/a							
2	Command Axis Number = 001_2			Command Message Type = 03				
3	Response Axis Number = 001_2			Response Message Type				
4	Acceleration Low Byte							
5	Acceleration Low Middle Byte							
6	Acceleration High Middle Byte							
7	Acceleration High Byte							

Acceleration - Command Message Type 03 hex

This double word defines the Profile Move's Acceleration in pulses/sec^2, when the Load Data /Start Profile bit transitions from zero to one.

Figure 2.8: Type 04 hex Deceleration

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	Hard Stop	Smooth Stop	Direction (V. Mode)	Incremental	n/a	Load Data/ Start Profile
1	n/a							
2	Command Axis Number = 001_2			Command Message Type = 04				
3	Response Axis Number = 001_2			Response Message Type				
4	Deceleration Low Byte							
5	Deceleration Low Middle Byte							
6	Deceleration High Middle Byte							
7	Deceleration High Byte							

Deceleration - Command Message Type 04 hex

This double word defines the Profile Move's Deceleration in pulses/sec^2, when the Load Data/Start Profile bit transitions from zero to one.

Figure 2.9: Type 1B hex Position Controller Attribute

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	Hard Stop	Smooth Stop	Direction (V. Mode)	Incremental	n/a	Load Data/ Start Profile
1	Position Controller Attribute to Get							
2	Command Axis Number = 001_2			Command Message Type = 1B				
3	Position Controller Attribute to Set							
4	Position Controller Attribute Value Low Byte							
5	Position Controller Attribute Value Low Middle Byte							
6	Position Controller Attribute Value High Middle Byte							
7	Position Controller Attribute Value High Byte							

Attribute Value - Command Message Type 1B hex

This double word defines the new value of the attribute to set when the Load Data/Start Profile bit transitions from zero to one.

Object Attribute to Get - Command Message Type 1B hex

This byte specifies the Position Controller object attribute from which to get the value and return in the response message.

Object Attribute to Set - Command Message Type 1B hex

This byte specifies the Position Controller object attribute to set to the new value defined by the attribute value when the Load Data/Start Profile bit transitions from zero to one.

Figure 2.10: Descriptions of Command Message Format Bits for the Stepping Motor Cube

Byte	Bit	Name	Action
0	0	Load Data/ Start Profile	Set from zero to one to load command data. The transition of this bit from zero to one will also start a Profile Move when the command message type contained in the command message field is the message type that starts a Profile Move for the mode selected.
0	2	Incremental	This bit is used to define the position value as either absolute or incremental. 0 = absolute position value and 1 = incremental position value.
0	3	Direction (V. Mode)	This bit is used to control the direction of the motor in Velocity mode. 1 = forward, positive and a 0 = reverse, negative.
0	4	Smooth Stop	This bit is used to bring the motor to a controlled stop at the currently implemented deceleration rate.
0	5	Hard Stop	This bit is used to bring the motor to an immediate stop.
0	7	Enable	This bit is used to control the drive enable. Clearing this bit will disable the drive. If there is a motion Profile in Progress when the drive is disabled, the currently executing motion profile will be aborted.
2	0-4	Command Message Type	This field defines the Command Message Type.
2	5-7	Command Axis Number	These three bits will always be set to 001_2 to indicate axis 1.
3	0-4	Response Message Type	This field defines the Response Message Type.
3	5-7	Response Axis Number	These three bits will always be set to 001_2 to indicate axis 1.

DeviceNet Response Messages Supported by Stepping Motor Cube

Figure 2.11: Type 01 hex Actual Position

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	n/a	Current Direction	General Fault	On Target Position	n/a	Profile in Progress
1	n/a							
2	Load Complete	n/a	FE Fault	n/a	n/a	n/a	n/a	n/a
3	Response Axis Number = 001_2			Response Message Type = 01				
4	Actual Position Low Byte							
5	Actual Position Low Middle Byte							
6	Actual Position High Middle Byte							
7	Actual Position High Byte							

Actual Position - Response Message Type 01 hex

This double word reflects the actual position in pulses. If position feedback is not used, this word will report the commanded position.

Figure 2.12: Type 02 hex Commanded Position

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	n/a	Current Direction	General Fault	On Target Position	n/a	Profile in Progress
1	n/a							
2	Load Complete	n/a	FE Fault	n/a	n/a	n/a	n/a	n/a
3	Response Axis Number = 001_2			Response Message Type = 02				
4	Commanded Position Low Byte							
5	Commanded Position Low Middle Byte							
6	Commanded Position High Middle Byte							
7	Commanded Position High Byte							

Commanded Position - Response Message Type 02 hex

This double word reflects the commanded position in pulses.

Figure 2.13: Type 03 hex Actual Velocity

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	n/a	Current Direction	General Fault	On Target Position	n/a	Profile in Progress
1	n/a							
2	Load Complete	n/a	FE Fault	n/a	n/a	n/a	n/a	n/a
3	Response Axis Number = 001_2			Response Message Type = 03				
4	Actual Velocity Low Byte							
5	Actual Velocity Low Middle Byte							
6	Actual Velocity High Middle Byte							
7	Actual Velocity High Byte							

Actual Velocity - Response Message Type 03 hex

This double word reflects the actual velocity in pulses/second.

Figure 2.14: Type 14 hex Command/Response Error

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	n/a	Current Direction	General Fault	On Target Position	n/a	Profile in Progress
1	Reserved = 0							
2	Load Complete	n/a	FE Fault	n/a	n/a	n/a	n/a	n/a
3	Response Axis Number = 001_2			Response Message Type = 14				
4	General Error Code							
5	Additional Code							
6	Copy of Command Message Byte 2							
7	Copy of Command Message Byte 3							

General Error Code- Response Message Type 14 hex

This byte (byte 4) identifies an error has been encountered. Figure 2.15 summarizes specific behavior for the position controller profile.

Additional Code – Response Message Type 14 hex

This byte (byte 5) contains an object/service-specific value that further describes the error condition. If the responding object has no additional information to specify, then the value FF_{hex} is placed within this field.

Figure 2.15: DeviceNet Error Codes Reported in Message Type 14 hex

General Code	Additional Code	Response	Semantics
08_{hex}	01_{hex}	Service Not Supported	Command Message type not supported. Additional code 01 takes precedence over additional code 02.
	02_{hex}	Service Not Supported	Response message type not supported.
05_{hex}	01_{hex}	Path Destination Unknown	A consumed axis number was requested that does not exist in the drive.
	02_{hex}	Path Destination Unknown	A produced axis number was requested that does not exist in the drive.
09_{hex}	FF_{hex}	Invalid Attribute Value	Load value is out of range.
$0E_{hex}$	FF_{hex}	Attribute not Settable	A request to modify a non-modifiable attribute was received.
11_{hex}	FF_{hex}	Reply Data too Large	The data requested is more than four bytes.
13_{hex}	FF_{hex}	Not Enough Data	I/O command message contains fewer than 8 bytes.
14_{hex}	01	Attribute Not Supported	Attribute to set specified in request is not supported.
	02	Attribute Not Supported	Attribute to get specified in request is not supported.

Figure 2.16: Type 1B hex Position Controller Attribute

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	n/a	n/a	Current Direction	General Fault	On Target Position	n/a	Profile in Progress
1	Position Controller Attribute to Get							
2	Load Complete	n/a	FE Fault	n/a	n/a	n/a	n/a	n/a
3	Response Axis Number = 001_2			Response Message Type = 1B				
4	Position Controller Attribute Value Low Byte							
5	Position Controller Attribute Value Low Middle Byte							
6	Position Controller Attribute Value High Middle Byte							
7	Position Controller Attribute Value High Byte							

Figure 2.17 Descriptions of Response Message Format Bits for the Stepping Motor Cube

Byte	Bit	Name	Action
0	0	Profile in Progress	This bit indicates that a profile move is in progress. 1 = profile in progress; 0 = Profile not in progress.
0	2	On Target Position	This bit indicates whether the motor is on the last targeted position. (1 = Current position equals the last target position.)
0	3	General Fault	This bit indicates the logical "or" of all fault conditions. 1 = general fault has occurred. General faults include lost enable, overcurrent, undervoltage, and invalid motion.
0	4	Current Direction	This bit shows the current direction of the motor. If the motor is not moving the bit will indicate the direction of the last commanded move. 0 = reverse or negative direction and 1 = forward or positive direction.
0	7	Enable	This bit indicates the state of the drive enable. A 1 indicates the drive is active.
2	7	Load Complete	A 1 in this bit indicates that the command data contained in the command message has been successfully loaded into the device.
2	5	FE Fault	A 1 in this bit indicates that the motor has stalled.
3	0-4	Response Message Type	This byte defines the Response Message Type.
3	5-7	Response Axis Number	These three bits will always be set to 001_2 to indicate axis 1.

DeviceNet I/O Handshaking Procedure

Two bits are used to provide handshaking between command and response messages: the Load Data/Start Profile bit (i.e., bit 0 of byte 0) and the Load Complete bit (i.e., bit 7 of byte 2). The Load Data/Start Profile bit is used to synchronize data transfers between the master and the Stepping Motor Cube. The Stepping Motor Cube acts on the data contained in an input message only when the Load Data/Start Profile bit is true. In each data transaction example, reset the Load Data/Start Profile bit before writing new data into the scanner output table. Use the Load Data bit to ensure that a complete data set is written into the scanner output table before the Load Data/Start Profile bit is set true. Figures 2.18 and 2.19 flowchart the procedures for Client Data Loading and Client Profile Move.

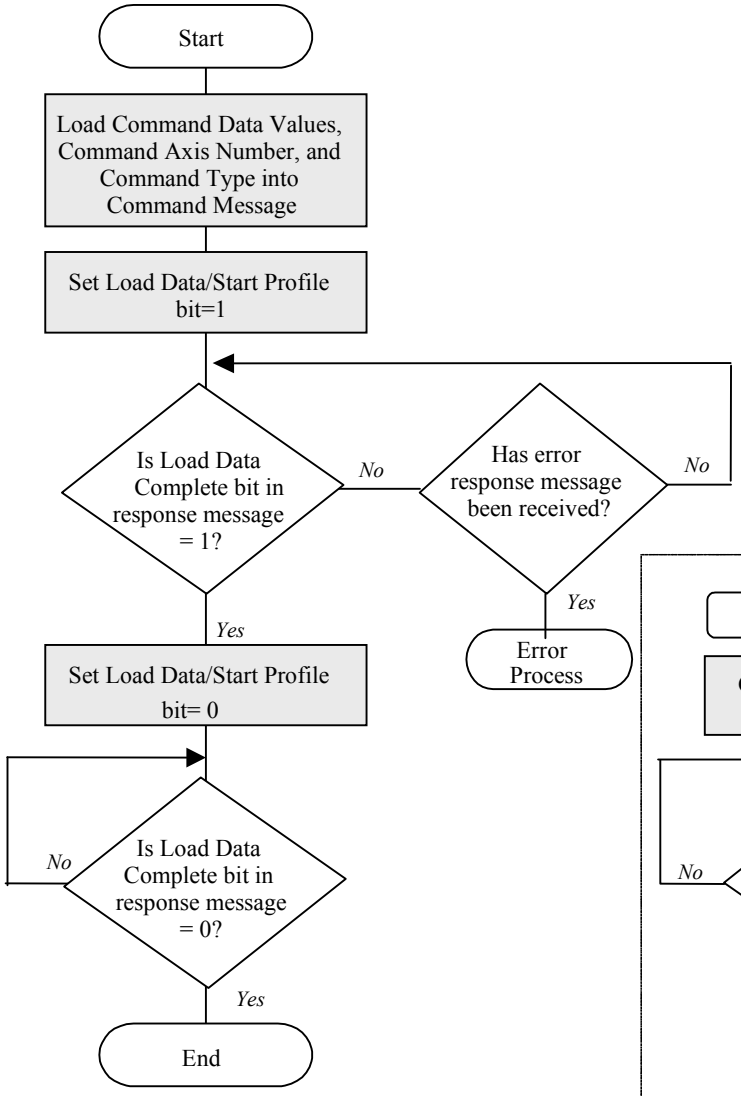


Figure 2.18: Client Data Loading Procedure

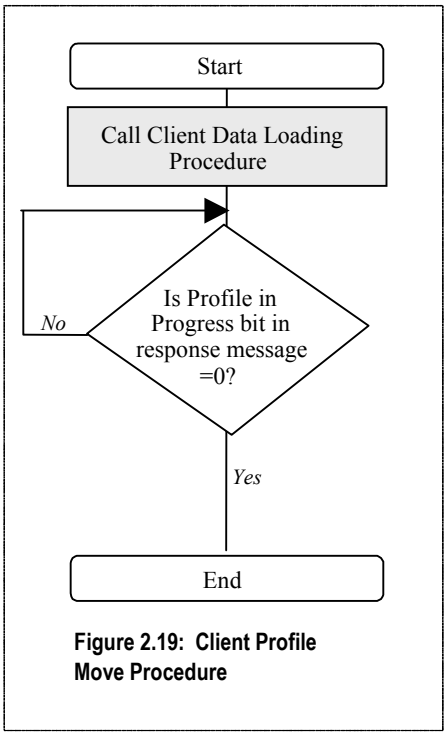


Figure 2.19: Client Profile Move Procedure

Sending Explicit Messages

The *Explicit Messaging Connection* is a generic, multipurpose communication path between two *nodes*. Explicit messages travel from client to server. The *client* originates a message, or request; the *server* reacts to the message with a response. The client's DeviceNet services usually generate message identifiers and headers automatically.

Explicit messages use the DeviceNet *objects* that reside within the Stepping Motor Cube. Those objects support *services* that allow you to get/set attributes to and from your Stepping Motor Cube over DeviceNet. The services perform as described below:

Set Attribute Single Service Modifies an attribute value within the DspMotion controller. The service validates any attribute data before it accepts the modification. If an error is detected, an error response is returned; otherwise, a successful Set Attribute Single response is returned.

Get Attribute Single Service Causes the object to return the contents of the specified attribute to the requester. If an error is detected, an error response is returned; otherwise, a successful Get Attribute Single response is returned along with the requested attribute data.

Figure 2.20 lists the parameters that are specified within the service data field of service requests and successful service responses.

Figure 2.20: Service Data Field Parameters

Set Attribute Single Service	*Data for Request*	Attribute ID	USINT	Identifies the attribute to be set.
		Attribute Data	Object/class Attribute-specific Struct	Contains the value to which the specified attribute is to be modified.
Get Attribute Single Service	*Data for Request*	Attribute ID	USINT	Identifies the attribute to be read/returned.
	Service Data for Success Response	Attribute Data	Object/class Attribute-specific Struct	Contains the requested attribute data.

DeviceNet Objects for Explicit Messaging

This section identifies the *objects* in the Stepping Motor Cube and the services and attributes that they provide.

Figure 2.21: Motor Cube DeviceNet Objects

Object: Identity
Class ID: 01

Instance*	Service	Service Code (Hex)	Attr. ID	Attribute Description	Attribute Type	Attribute Values	Motor Cube Default Value	Units
1	Reset	05	—	Type of reset	USINT	0, 1	n/a	n/a
1	Get Attr Single	0E	1	Vendor ID	UINT	38	n/a	n/a
1	Get Attr Single	0E	2	Device type	UINT	16	n/a	n/a
1	Get Attr Single	0E	3	Product code	UINT	0-65,535	n/a	n/a
1	Get Attr Single	0E	4	Revision	UINT, UINT	1-4, 10-99	n/a	n/a
1	Get Attr Single	0E	5	Status	WORD	0-65,535	n/a	n/a
1	Get Attr Single	0E	6	Serial number	UDINT	$0-(2^{32}-1)$	n/a	n/a
1	Get Attr Single	0E	7	Product name	SHORT_STRING	Model number	n/a	n/a

Generation D RTOS Equivalent

Object: DeviceNet
Class ID: 03

Instance*	Service	Service Code (Hex)	Attr. ID	Attribute Description	Attribute Type	Attribute Values	Motor Cube Default Value	Units
0	Get Attr Single	0E	1	Revision	UINT	2	n/a	n/a
1	Get Attr Single	0E	5	Allocation info	BYTE, USINT	0-255, 0-63, 255	n/a	n/a
1	Allocate M/S Connection Set	4B	—	—	BYTE, USINT	0-255, 0-63	n/a	n/a
1	Release M/S Connection Set	4C	—	—	BYTE	0-255	n/a	n/a

Generation D RTOS Equivalent

Object: Connection
Class ID: 05

Instance*	Service	Service Code (Hex)	Attr. ID	Attribute Description	Attribute Type	Attribute Values	Motor Cube Default Value	Units
1-3	Reset	05	—	—	—	—	n/a	n/a
1-3	Get Attr Single	0E	1	State	USINT	0-5	n/a	n/a
1-3	Get Attr Single	0E	2	Instance type	USINT	0, 1	n/a	n/a
1-3	Get Attr Single	0E	3	Transport class trigger	BYTE	23_{16}, 83_{16}	n/a	n/a
1-3	Get Attr Single	0E	4	Produced connection ID	UINT	0-$7F0_{16}$, $FFFF_{16}$	n/a	n/a
1-3	Get/Set Attr Single	0E/10	5	Consumed connection ID	UINT	0-$7F0_{16}$, $FFFF_{16}$	n/a	n/a
1-3	Get Attr Single	0E	6	Initial comm. characteristics	BYTE	0, 1, 21_{16}, 33_{16}	n/a	n/a
1-3	Get Attr Single	0E	7	Produced connection size	UINT	8, 24	n/a	n/a
1-3	Get Attr Single	0E	8	Consumed connection size	UINT	8, 24	n/a	n/a
1-3	Get/Set Attr Single	0E/10	9	Expected packet rate	UINT	0-65,535	n/a	n/a
1-3	Get/Set Attr Single	0E/10	12	Watchdog timeout action	UINT	0, 1, 3	n/a	n/a
1-3	Get Attr Single	0E	13	Produced connection path length	UINT	0, 6	n/a	n/a
1-3	Get Attr Single	0E	14	Produced connection path	EPATH	Empty, 202424003021_{16}	n/a	n/a
1-3	Get Attr Single	0E	15	Consumed connection path length	UINT	0, 6	n/a	n/a
1-3	Get Attr Single	0E	16	Consumed connection path	EPATH	Empty, 202424003020_{16}	n/a	n/a

Generation D RTOS Equivalent

Object: Position Controller Supervisor
Class ID: 2416

Instance*	Service	Service Code (Hex)	Attr. ID	Attribute Description	Attribute Type	Attribute Values	Motor Cube Default Value	Units
0	Get Attr Single	0E	1	Revision	UINT	2	n/a	n/a
0	Get/Set Attr Single	0E/10	32	Consumed command message	ARRAY of BYTE	See figures 2.5 – 2.9	n/a	n/a
0	Get Attr Single	0E	33	Produced response message	ARRAY of BYTE	See figures 2.11 – 2.16	n/a	n/a
1	Get Attr Single	0E	3	Axis number	USINT	1	1	n/a
1	Get Attr Single	0E	5	General fault	BOOL	0, 1 = fault condition exists	1	n/a
1	Get Attr Single	0E	6	Command message type	USINT	1-4, 27	n/a	n/a
1	Get Attr Single	0E	7	Response message type	USINT	1-3, 20, 27	n/a	n/a

Generation D RTOS Equivalent

Object: Position Controller
Class ID: 2516

Instance*	Service	Service Code (Hex)	Attr. ID	Attribute Description	Attribute Type	Attribute Values	Motor Cube Default Value	Units
0	Get Attr Single	0E	1	Revision	UINT	2	n/a	n/a
1	Get Attr Single	0E	1	Number of attributes	USINT	26	26	n/a
1	Get Attr Single	0E	2	Attribute list	ARRAY of USINT	1-3, 6-15, 17, 20, 21, 23, 24, 47, 58, 100-102, 111-113	same as attribute values	n/a
1	Get/Set Attr Single	0E/10	3	Mode	USINT	0 = position; 1 = velocity	0	n/a
1	Get/Set Attr Single	0E/10	6	Target position	DINT	-2,147,483,648 to +2,147,483,647	0	pulses
1	Get/Set Attr Single	0E/10	7	Target velocity	DINT	1-5,000,000	1	pulse
1	Get/Set Attr Single	0E/10	8	Acceleration	DINT	100-1,000,000,000	100	pulse/sec^2
1	Get/Set Attr Single	0E/10	9	Deceleration	DINT	100-1,000,000,000	100	pulse/sec^2
1	Get/Set Attr Single	0E/10	10	Incremental position flag	BOOL	0 = absolute 1 = incremental	0	n/a
1	Get/Set Attr Single	0E/10	11	Load data/ profile handshake	BOOL	0, 1	0	n/a
1	Get Attr Single	0E	12	On target position	BOOL	0, 1	0	n/a
1	Get/Set Attr Single	0E/10	13	Actual position	DINT	+/- 2,000,000,000	0	pulses

Instance*	Service	Service Code (Hex)	Attr. ID	Attribute Description	Attribute Type	Attribute Values	Motor Cube Default Value	Units
1	Get Attr Single	0E	14	Actual velocity	DINT	+/-5,000,000	0	pulses/sec
1	Get Attr Single	0E	15	Commanded position	DINT	-2,147,483,648 to +2,147,483,647	0	pulses
1	Get/Set Attr Single	0E/10	17	Enable	BOOL	0 = disable 1 = enable drive	0	n/a
1	Get/Set Attr Single	0E/10	20	Smooth stop	BOOL	0, 1 = stop	0	n/a
1	Get/Set Attr Single	0E/10	21	Hard stop	BOOL	0, 1 = halt	0	n/a
1	Get/Set Attr Single	0E/10	23	Instantaneous direction	BOOL	0 = negative or reverse; 1 = positive or forward	0	n/a
1	Get/Set Attr Single	0E/10	24	Reference direction	BOOL	0 = forward is CW 1 = forward is CCW	0	n/a
1	Get Attr Single	0E	47	Following error fault	BOOL	0, 1	0	n/a
1	Get Attr Single	0E	58	Load data complete	BOOL	0, 1	0	n/a
1	Get Attr Single	0E	100	Fault code (see table 2.22)	UDINT	0 – $FFFFFFFF_{16}$	8 (LE)	n/a
1	Get/Set Attr Single	0E/10	101	Continuous current	INT	1-1,000	600	% * 10
1	Get/Set Attr Single	0E/10	102	Power save current	INT	0-1,000	600	% * 10
1	Get/Set Attr Single	0E/10	111	Stall velocity threshold	DINT	50,000 – 5,000,000 pulses/sec	5,000,000	pulses/sec
1	Get/Set Attr Single	0E/10	112	Bus voltage	BYTE	0, 24 – 50 Volts	0	VDC
1	Get/Set Attr Single	0E/10	113	Stall sensitivity	INT	1 - 2,000ms	200	ms

Generation D RTOS Equivalent

Figure 2.22: Fault Code Attribute Bits and Fault Messages

(Stepping Motor Cube Explicit Message: Position Controller Object, Service Code 100 Hex)

Bit	Message
0 – 2	Reserved
3	Lost Enable
4 - 15	Reserved
16	Invalid Motion
17 – 20	Reserved
21	Excessive Following Error
22 - 25	Reserved
26	4.3 Amp – Motor Power Clamp Excessive Duty Cycle 7.2 - 28 Amp: Motor Power – Under Voltage
27	Reserved
28	Motor Over-Current Fault
29	Reserved
30	Controller Over-Temperature

After a fault occurs, the drive will become disabled. To enable the drive, use one of the following methods:

- DeviceNet Command Message (implicit messaging): set byte 0, bit 7 true.
- DeviceNet Explicit Message: set Position Controller Attribute 17 to 1.

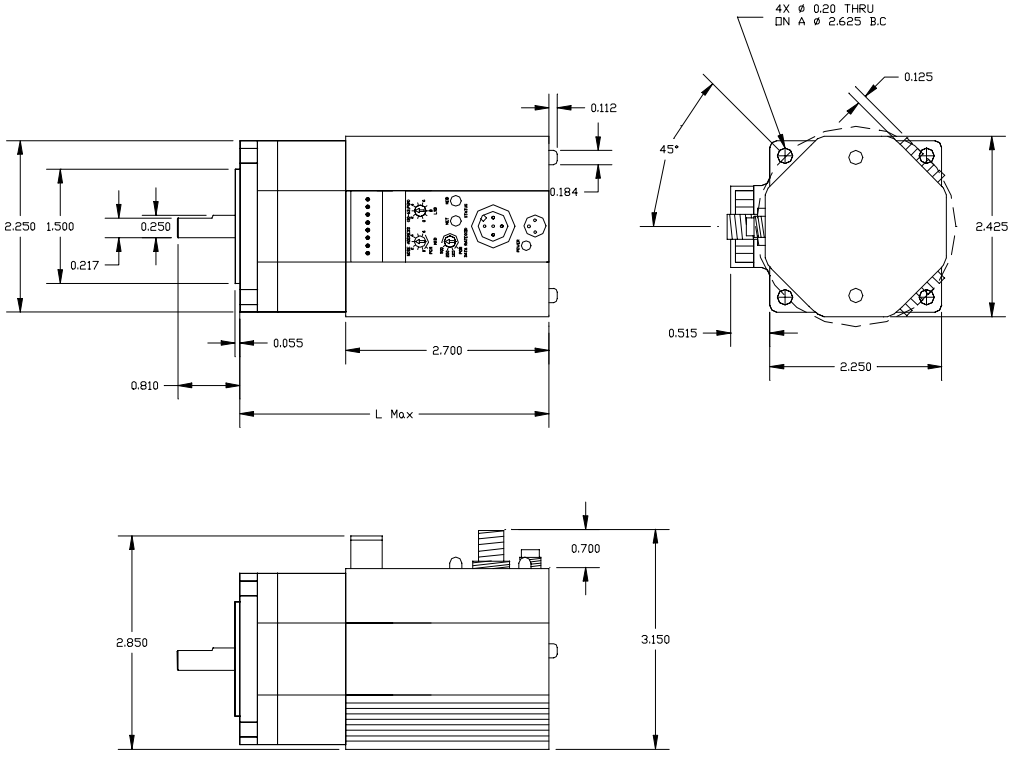
Chapter 3

Mechanical Drawing

Mechanical Specifications

Stepping Motor Cube with Integral Drive and DeviceNet

MODEL	LMax
IC800MCUB12160XD	3.700
IC800MCUB12210XD	4.200
IC800MCUB12310XD	5.200



Chapter 4

User Connections and Network Settings

Stepping Motor Cube with Integral Drive and DeviceNet

Models:

IC800MCUB12160XD
IC800MCUB12210XD
IC800MCUB12310XD