

# ABOUT THIS GUIDE

## Chapter 1. Implementing 6K DeviceNet..... 1

DeviceNet Overview .....	2
6Kn-DN.....	2
Technical Assistance .....	2
Implementation Process.....	2
EDS File .....	2
Hardware Interface .....	3
LED Status Indicators .....	3
DeviceNet Connector Pin Out .....	4
Termination.....	4
Node Address.....	4
Baud Rate.....	5
Programming Notes.....	6
Data Format.....	6
Implementing Data Exchange .....	6
Handling a DeviceNet Fault .....	7
Affected Commands and Features.....	7
DeviceNet Objects .....	8
Identity Object, Class 0x01.....	8
Message Router Object, Class 0x02 .....	8
DeviceNet Object, Class 0x03.....	9
Assembly Object, Class 0x04.....	9
Connection Object, Class 0x05 .....	10
Acknowledge Handler Object, Class 0x2B .....	12
Command Descriptions.....	13
ERROR     Error Checking Enable .....	13
FBADDR    Fieldbus Address .....	14
FBBAUD    Fieldbus Baud Rate .....	14
[ FBS ]    Fieldbus Status.....	15
FBSIZE    Fieldbus Data Packet Size .....	16
OPTEN     Option Card Enable/Disable .....	16
OUTFNC    Output Function .....	17
TFBS     Transfer Fieldbus Status.....	17
TFBSF    Fieldbus Status Full Text .....	18
TOPSTS    Option Card Status Full Text .....	18
Programming Scenario .....	19

## Chapter 2. Implementing Gemini DeviceNet..... 21

DeviceNet Overview .....	22
Gemini GT6-DN and GV6-DN.....	22
Technical Assistance .....	22
Implementation Process.....	22
EDS File .....	22
Data Types .....	23
Hardware Interface .....	24
LED Status Indicators .....	24
DeviceNet Connector Pin Out .....	25
Termination.....	25
Node Address.....	26
Baud Rate.....	26
Configuration and Programming .....	27
Basic Requirements.....	27
Hardware Requirements .....	27
Object Model.....	27
Explicit Messaging .....	28
Position Controller Supervisor Object Class (#164)(A4hex) and Position Controller Object Class (#165)(A5hex).....	28
Explicit Message Command.....	28

Explicit Message Response.....	28
Services.....	29
Identity Object, Class 0x01 .....	30
DeviceNet Object, Class 0x03.....	31
Assembly Object, Class 0x04.....	31
Connection Object, Class 0x05 .....	32
Acknowledge Handler Object, Class 0x2B .....	34
Position Controller Object Class (0xA5) .....	37
Polled I/O.....	40
Command Assemblies .....	40
Response Assemblies .....	41
Running a Stored Program through DeviceNet .....	42
Stopping the Program through DeviceNet.....	42
Command Assembly Codes.....	43
No Operation 0x00.....	43
Target Position 0x01 .....	43
Target Velocity 0x02 .....	43
Acceleration 0x03 .....	44
Deceleration 0x04 .....	44
Variable Access 0x0C.....	44
Continuous Velocity 0x11.....	44
Homing 0x12.....	45
Position Controller Supervisor Attribute 0x1A.....	45
Position Controller Attribute 0x1B .....	45
Alarm Clear 0x1E.....	46
Response Assembly Codes.....	47
Actual Position 0x01 - Servo Only.....	47
Commanded Position 0x02 .....	47
Actual Velocity 0x03 - Servo Only .....	48
Commanded Velocity 0x04 .....	48
Variable access 0x0C.....	48
Command/Response Error 0x14.....	49
Position Controller Supervisor Attribute 0x1A.....	50
Position Controller Attribute 0x1B .....	50
Status Code 0x1E.....	51
Alarm Code 0x1F.....	51
Operating Modes .....	52
Positioning.....	52
Continuous Velocity.....	54
Homing.....	56
Command Descriptions.....	60
ERROR     Error Checking Enable .....	60
FBADDR    Fieldbus Address .....	60
FBBAUD    Fieldbus Baudrate Setting.....	61
FBMASK    Fieldbus I/O Mask.....	61
[ FBS ]    Fieldbus Status.....	62
OUTFNC    Output Function .....	63
TASX     Transfer Extended Axis Status .....	63
TFBS     Transfer Fieldbus Status.....	63

## Appendix A. Gemini DeviceNet CE Compliance..... 65

CE Compliance.....	66
Installation Instructions .....	66

## Appendix B. Gemini Drive Dimensions..... 67

Gemini Drive Dimensions.....	68
Gemini Panel Layout Dimensions .....	71

# Change Summary – Revision B

August 1, 2002

This document, 88-019049-01B, supersedes 88-019049-01A. Changes and corrections are noted below.

---

Topic	Description
Document clarifications and corrections	<ul style="list-style-type: none"><li>• DeviceNet Overview: Added ODVA software self-certification notice – see page <a href="#">22</a>.</li><li>• LED Status indicators: Enhanced description of LEDs – see page <a href="#">24</a>.</li><li>• Node Address: Rearranged table for clarity – see page <a href="#">26</a>.</li><li>• Object Model: added Acknowledge handler and renumbered position controller and position controller supervisor objects – see page <a href="#">27</a>.</li><li>• Added Identity object, DeviceNet Object, Assembly Object, Connection Object and Acknowledge Handler Object descriptions. See pages <a href="#">30</a> – <a href="#">34</a>.</li><li>• Position Controller Supervisor object: Added class attribute description, added Object Instance Attributes 100-107 (see page <a href="#">35</a>).</li><li>• Position Controller Object: Removed attribute 1, added attributes 106-112. Renamed attribute 26 to 113. Modified attributes 13, 50, 51; – see page <a href="#">37</a>.</li><li>• Polled I/O: Replaced Valid Data with Reg Arm. – see page <a href="#">40</a>.</li><li>• Polled I/O: Added more explanation how to run stored program – see page <a href="#">42</a>.</li><li>• Polled I/O: Added command assembly 0x0C – see page <a href="#">44</a>.</li><li>• Polled I/O: Removed Response assembly 0x00 – see page <a href="#">47</a>.</li><li>• Added Command/Response Error 0x14 – see page <a href="#">49</a>.</li><li>• Operating Modes: Enhanced examples - see pages <a href="#">52</a> – <a href="#">58</a>.</li><li>• Added more description to [ FBS ] command - see page <a href="#">62</a>.</li><li>• Added CE Compliance information for Gemini products – see page <a href="#">66</a>.</li><li>• Added dimensions for Gemini drives (page <a href="#">68</a>) and panel layout dimensions for Gemini drives (page <a href="#">71</a>).</li></ul>

---

CHAPTER TWO

# Implementing Gemini DeviceNet

## IN THIS CHAPTER

- DeviceNet ..... 22
- Hardware Interface ..... 24
- Configuration and Programming ..... 27
- Command Assembly Codes ..... 43
- Response Assembly Codes ..... 47
- Command Description ..... 60

## Gemini GT6-DN and GV6-DN

The Gemini GT6-DN and GV6-DN are high performance stepper and servo drives designed to support the Open DeviceNet Vendor's Association (ODVA) specification for DeviceNet. These products have been self-tested by the vendor and found to comply with ODVA Protocol Conformance Test Software Version A-16.

These compact digital drives will offer enhancements over today's drive offering with features such as digital tuning and notch filters for our servos, as well as sensorless stall detect and ABS damping for our steppers. These drive/controllers are configured over RS232/485 with Motion Planner on a full size PC or laptop.

Cabling is not provided by Compumotor.

## Technical Assistance

Technical questions regarding DeviceNet should be addressed to your local DeviceNet User Group. An address list is available on the DeviceNet Internet site at [www.odva.org](http://www.odva.org).

For support with Gemini specific questions, contact Applications at 800-358-9070 or e-mail us at [tech\\_help@cmotor.com](mailto:tech_help@cmotor.com).

## Implementation Process

DeviceNet Master (user defined):

1. Use the provided file, CMTR2.EDS. Do not modify.
2. Configure communication baudrate.

Gemini Drive:

1. Enable/disable terminating resistors as needed (see page 25).
2. Launch Motion Planner (CD-ROM is provided in your ship kit).
3. Establish a direct communication link (serial) with the Gemini. Refer to your hardware installation documentation for connection instructions.
4. Configure node address (see FBADDR, or use hardware method on page 26).
5. Configure baud rate (see FBBAUD on page 61 or use the hardware method on page 26). Setting the baud rate via hardware overrides the software setting.
6. Reset the Gemini drive to initialize the DeviceNet card.

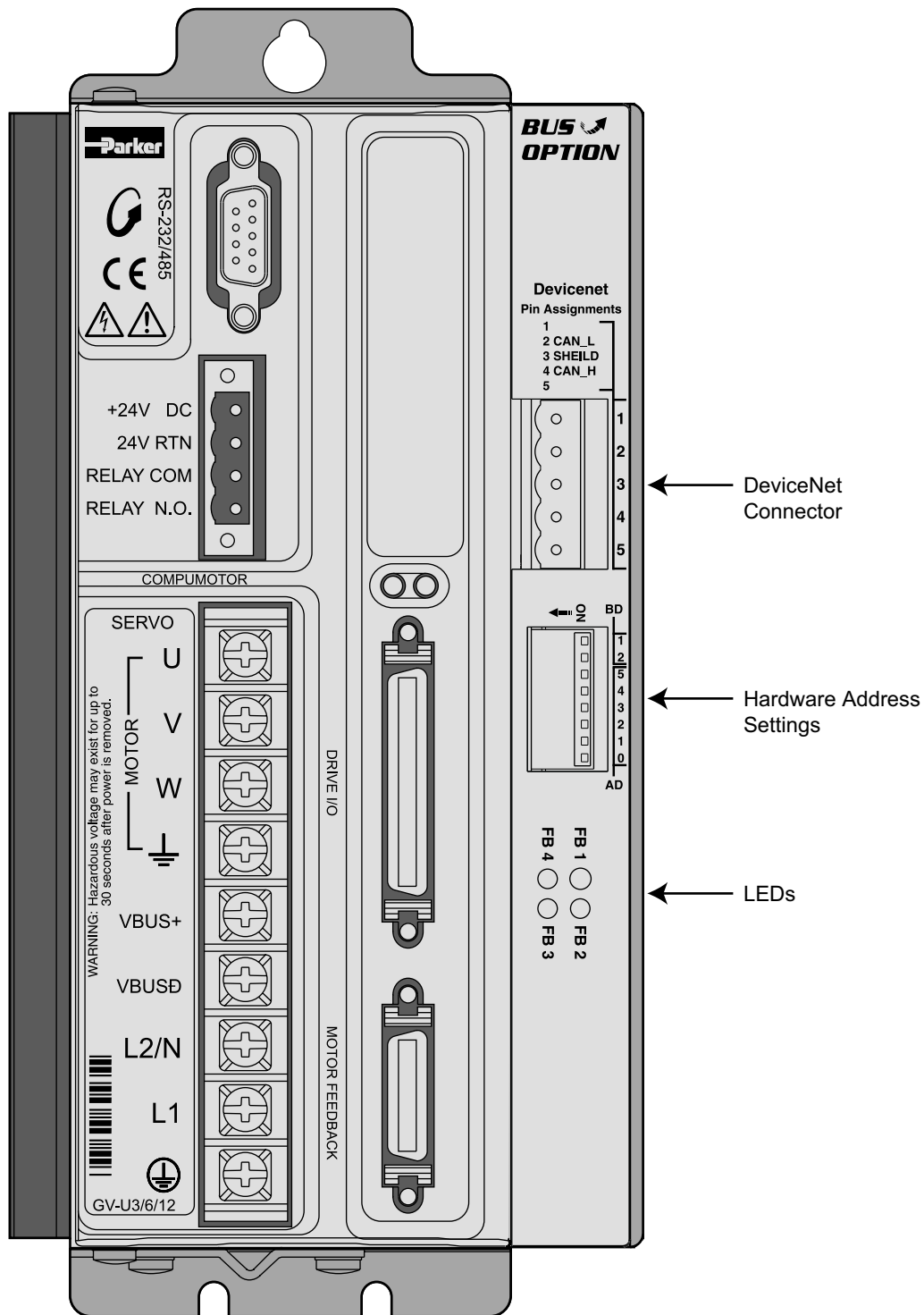
## EDS File

Each device in a DeviceNet network is associated with an EDS file, containing all necessary information about the device. The latest version of the Gemini EDS file (CMTR2.EDS) can be downloaded from [www.compumotor.com](http://www.compumotor.com).

## Data Types

The table below describes the various data types associated with the DeviceNet object attributes, number of bits, and min/max values.

<b>Data Type</b>	<b># of Bits</b>	<b>Min Value</b>	<b>Max Value</b>
BOOL (Boolean)	1	0 (False)	1 (True)
SINT (Short Integer)	8	-128	127
USINT (Unsigned Short Integer)	8	0	255
INT (Integer)	16	-32768	32767
UINT (Unsigned Integer)	16	0	65535
DINT (Double Integer)	32	$-2^{31}$	$2^{31}-1$
UDINT (Unsigned Double Integer)	32	0	$2^{32}-1$



## LED Status Indicators

Bicolor flashing LED indicators are provided on the DeviceNet option card. Refer to the following table for troubleshooting information provided by these LEDs.

LED	Steady	Flash	Function	Status *
FB1	--	--	Not used	--
FB2	Off		Not powered/not online	--
	Green		Network link ok	FBS bit #4 = 1
	Red		Network critical link failure	FBS bit #5 = 1
		Green	1 flash/second – Network link not connected	FBS bit #6 = 1
		Red	1 flash/second – Network connection timeout	FBS bit #7 = 1
FB3	Off		No power	--
	Green		Module device operational	FBS bit #8 = 1
	Red		Module unrecoverable fault	FBS bit #9 = 1
		Green	1 flash/second – Module in standby	FBS bit #10 = 1
		Red	1 flash/second – Module minor fault	FBS bit #11 = 1
FB4	--	--	Not used	--

\* To check status, execute the `TFBS` command (bit status report) or the `TFBSF` command (full text status report) in the terminal emulator. You can also use the `FBS` operator to assign or compare one or more status bits (e.g., use in an `IF` expression, etc.). Refer to the `TFBS` command description on page 63.

## DeviceNet Connector Pin Out

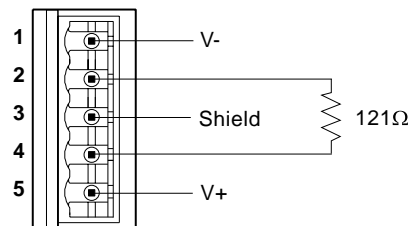
The following table gives the pin out for the DeviceNet connector. Industry standard DeviceNet connections are used.

Pin	Name	Function
1	V-	DC Return
2	CAN_L	CANBUS LOW
3	SHIELD	Protective Earth
4	CAN_H	CANBUS HIGH
5	V+	+24 VDC Power *

\* 30mA in standby and 100mA in rush

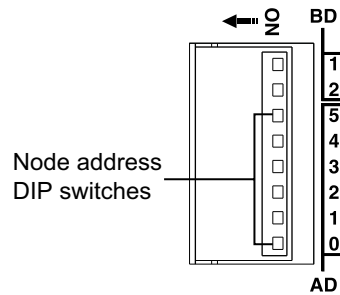
## Termination

If the Gemini drive/controller is used as the first or last node in a network, a termination resistor must be used. Compumotor recommends a resistor value of 121 ohms. Connect the resistor as shown below.



## Node Address

To configure node address via hardware, dip switches are provided to set a node address of 0-63. The switches AD0-AD5 give a binary representation of the node address. Setting the dip switches to 0xFF (all ON), enables software configuration of node address (see FBADDR on page 60).



Address	AD5 (32)	AD4 (16)	AD3 (8)	AD2 (4)	AD1 (2)	AD0 (1)
0	OFF	OFF	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	OFF	ON	OFF
3	OFF	OFF	OFF	OFF	ON	ON
...						
62	OFF	ON	ON	ON	ON	ON
63	ON	ON	ON	ON	ON	ON

Example:

Switch AD0-AD5 = OFF-ON-OFF-ON-ON-ON, node address is 23 (16+4+2+1).  
 Switch AD0-AD5 and BD1-BD2 = all ON, node address determined by FBADDR

## Baud Rate

To configure baud rate via hardware, dip switches are provided to set a baud rate of 125, 250, or 500kb. Setting the dip switches to 0xFF (all ON), enables software configuration of baud rate (see FBBAUD on page 61).



Baudrate (Bit/sec)	BD1	BD2
125k	OFF	OFF
250k	OFF	ON
500k	ON	OFF
Reserved	ON	ON

Example:

Switch BD1-BD2 = ON-OFF, baud rate 500kbit/s  
 Switch AD0-AD5 and BD1-BD2 = all ON, baud rate determined by FBBAUD

# Configuration and Programming

All parameters on the drives are set over RS232/485 except with regard to the DeviceNet card. This card provides dip switches for device number and baud rate.

## Basic Requirements

The Gemini DeviceNet options are based on the Gemini GV6 and GT6 platforms. All commands supported by these platforms, with the exception of RS232/485 commands, will also be supported by the DeviceNet options. See the *Gemini Series Programmer's Reference* for a complete listing of these commands. Additional DeviceNet specific commands are listed on page 60 of this manual.

## Hardware Requirements

Devices are expected to operate in the following environment:

Explicit Peer-to-Peer Messaging	No
I/O Peer to Peer Messaging	No
Baud Rate	500 kB
Polled Response Time	< 10msec
Explicit Response Time	< 50msec
Master/Scanner	No
I/O Slave Messaging	
Bit Strobe	No
Polling	Yes
Cyclic	No
Change-of-State	No

## Object Model

Object Class	Class code	Instance #	Description
Identity	0x01	1	Provides identification of and general information about the device. The Identity Object MUST be present in all DeviceNet products.
Message Router	0x02	1	Provides a messaging connection point through which a Client may address a service to any object class or instance residing in the physical device
DeviceNet	0x03	1	Provides the configuration and status of a DeviceNet port. Each DeviceNet product must support one (and only one) DeviceNet object per physical connection to the DeviceNet communication link.
Assembly	0x04	1	Binds attributes of multiple objects, which allows data to or from each object to be sent or received over a single connection. Stores I/O output Message Data.
Assembly	0x04	2	Stores I/O Input Message Data
Explicit Connection	0x05	1	Manages the explicit messages
I/O Connection	0x05	2	Manages the I/O messages
Acknowledge Handler	0x2B	1	Handles/configures acknowledgement of data.
Position Controller Supervisor	0xA4	1	Handles errors for the position controller as well as Home and Registration inputs
Position Controller	0xA5	Axis	Performs the control output velocity profiling and handles input and output to and from the motor drive unit, limit switches registration etc.

# Explicit Messaging

## Position Controller Supervisor Object Class (#164)(A4hex) and Position Controller Object Class (#165)(A5hex)

Explicit Messaging Connections provide generic, multi-purpose communication paths between two devices. Explicit Messages are exchanged across Explicit Messaging Connections. Explicit Messages are used to command the performance of a particular task and to report the results of performing the task. The meaning/intended use of an Explicit Message is stated within the CAN Data Field. Explicit Messaging provides the means by which typical request/response oriented functions are performed (e.g. module configuration). DeviceNet defines an Explicit Messaging protocol that states the meaning of the message.

An Explicit Message consists of a Connection ID and associated messaging protocol information.

### Explicit Message Command

TXID	COMMAND	Word 0
PORT	SIZE	Word 1
SERCE	MAC ID	Word 2
CLASS		Word 3
AXIS INSTANCE		Word 4
ATTRIBUTE		Word 5
LOWER DATA WORD		Word 6
UPPPER DATA WORD		Word 7

TXID	Transmit ID This value must be unique for each explicit message sent.
COMMAND	The command specified for this block of data. 01 - Send Explicit message 04 - Clear response buffer
PORT	0 - Channel A 1 - Channel B
SIZE	Size in bytes of all data after MAC ID.
SERVICE	0E(hex) - Get_Attribute_Single 10(hex) - Set_Attribute_Single
MAC ID	The DeviceNet ID of the amplifier.
CLASS	Position Controller Supervisor – 164(A4 hex) Position Controller Object - 165(A5 hex)
AXIS INSTANCE	Always 1 (01 hex)
ATTRIBUTE	The attribute number of the attribute being accessed (set or get).

### Explicit Message Response

TXID	STATUS	Word 0
PORT	SIZE	Word 1
SERCE	MAC ID	Word 2
DATA		Word 3 - 31

TXID	Transmit ID This value must be unique for each explicit message sent.
STATUS	The Status specified for this block of data. 01 - Transaction Successful XX - For all other response definitions, refer to the DeviceNet Master documentation.
PORT	0 - Channel A 1 - Channel B
SIZE	Size in bytes of all data after MAC ID.
SERVICE	0E(hex) - Get_Attribute_Single 10(hex) - Set_Attribute_Single
MACID	The DeviceNet ID of the amplifier.

## Services

Service Code	Name	Description
0x0E	Get_Attribute_Single	Returns the specified attribute.
0x10	Set_Attribute_Single	Sets the specified attribute.

Example:

In this example, we will use the Set\_Attribute\_Single service (10 hex) to set the Target Position attribute (06 hex) to 65536 (00010000 hex) of the Position Controller Class 165 (A5 hex) of MACID 25 (19 hex). In all cases with Gemini motion controllers, the axis instance is "1" (01 hex).

TXID	COMMAND	0101
PORT	SIZE(IN BYTES)	000A
SERVICE	MAC ID	1019
CLASS		00A5
AXIS INSTANCE		0001
ATTRIBUTE		0006
LOWER DATA WORD		0000
UPPER DATA WORD		0001

## Identity Object, Class 0x01

### Class Attributes (0)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	Revision	Get_Attribute_Single	Revision of identity object	1,1,65535	UINT

### Instance Attributes (1)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	Vendor Id	Get_Attribute_Single	Parker Hannifin/Compumotor Division	620	UINT
2	Device Type	Get_Attribute_Single	Communications Adapter	12	UINT
3	Product Code	Get_Attribute_Single	Gemini Drive	2	UINT
4	Revision	Get_Attribute_Single	Consists of major.minor	{1,1}, {1,1}, {1,1}	Array of USINT
5	Status	Get_Attribute_Single	Represents the current status of the entire device.	0,0,255	WORD
6	Serial Number	Get_Attribute_Single	Used in conjunction with the Vendor ID to form a unique identifier for each device on DeviceNet.	n/a	UDINT
7	Product Name	Get_Attribute_Single	Short description of the product represented by the Product Code.	"Gemini Drive"	SHORT_STRING
9	Config. Consist. Value	Get_Attribute_Single	Contents identify configuration of device.	n/a	UINT

## DeviceNet Object, Class 0x03

### Class Attributes (0)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	Revision	Get_Attribute_Single	Revision of identity object.	2,1,65535	UINT
32	Consumed Command Message	Get_Attribute_Single	Contains the content of the Command Message	Contains the content of the Command Message	
33	Produced Response Message	Get_Attribute_Single	Contains the content of the Response Message	Contains the content of the Response Message	

### Instance Attributes (1)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	MAC_ID	Get_Attribute_Single	Node Address	1*,0,63	USINT
2	Baud Rate	Get_Attribute_Single	Indicates the selected baud rate. Values are: 00 - 125k 01 - 250k 10 - 500k	2*,0,2	USINT
3	BOI	Get_Attribute_Single Set_Attribute_Single	Bus off interrupt	n/a	BOOL
4	Bus Off Counter	Get_Attribute_Single Set_Attribute_Single	Number of times device went to bus off state.	0,0,255	USINT
5	Allocation Information	Get_Attribute_Single	Struct of: BYTE: Allocation choice USINT: Master's MAC ID	n/a	Struct of: BYTE USINT

\* Assumes software configuration. If using dip switches, then the dip switches determines the default value.

## Assembly Object, Class 0x04

### Class Attributes (0)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	Revision	Get_Attribute_Single	Revision of assembly object.	2,1,65535	UINT
2	Max Instance	Get_Attribute_Single	Maximum instance number of an object currently created in this class level of the device.		UINT

### Static Input Instance Attributes (100)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
3	DATA	Get_Attribute_Single	Data put into the network from the controller	n/a	Array of BYTE

## Static Output Instance Attributes (150)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
3	DATA	Get_Attribute_Single Set_Attribute_Single	Data from the network and into the controller	n/a	Array of BYTE

## Connection Object, Class 0x05

### Class Attributes (0)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	Revision	Get_Attribute_Single	Revision of connection object.	1,1,65535	UINT

### Explicit Connecting Instance (1)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	State	Get_Attribute_Single	State of the object: 0=nonexistent, 1=configuring 3=established, 4=timed out	1,0,4	USINT
2	Instance Type	Get_Attribute_Single	Indicates either IO or messaging connection 0=Explicit, 1=I/O	0,0,1	USINT
3	Transport Class Trigger	Get_Attribute_Single	Defines behavior of connection	0x83	BYTE
4	Produced Conn Id	Get_Attribute_Single	Placed in CAN identifier field when the connection transmits	n/a	UINT
5	Consumed Conn Id	Get_Attribute_Single	CAN identifier field value that denotes message to be received	n/a	UINT
6	Initial Comm Characteristics	Get_Attribute_Single	Defines the Message Group(s) across which productions and consumptions associated with this connection	n/a	BYTE
7	Produced Conn Size	Get_Attribute_Single	Maximum number of bytes transmitted across this connection	516,4,516	UINT
8	Consumed Conn Size	Get_Attribute_Single	Maximum number of bytes received across this connection	516,4,516	UINT
9	Expected Packet Rate	Get_Attribute_Single Set_Attribute_Single	Defines timing associated with this connection. Resolution 10 ms	n/a	UINT
12	Watchdog Timeout Action	Get_Attribute_Single Set_Attribute_Single	Defines how to handle inactivity/watchdog timeouts	1 – Auto Delete 3 – Deferred Delete	USINT
13	Produced Connection Path Length	Get_Attribute_Single	Number of bytes in Produced_Connection_Path length attribute	0	UINT
14	Produced Connection Path	Get_Attribute_Single Set_Attribute_Single	Application object producing data on this connection	0	Array of: USINT
15	Consumed Connection Path Length	Get_Attribute_Single Set_Attribute_Single	Number of bytes in the Consumed_Connection_Path length attribute	0	UINT
16	Consumed Connection Path	Get_Attribute_Single	Specifies the application object(s) that are to receive data consumed by this connection object	n/a	Array of: 01 UINT
17	Production Inhibit Time	Get_Attribute_Single	Minimum delay time between new data production	n/a	UINT

## Polled IO Connection Instance (2)

#	Attribute Name	Services	Description	Default,Min,Max	Data Type
1	State	Get_Attribute_Single	State of the object: 0=nonexistent, 1=configuring 3=established, 4=timed out	1,0,4	USINT
2	Instance Type	Get_Attribute_Single	Indicates either IO or messaging connection 0=Explicit, 1=I/O	0,0,1	USINT
3	Transport Class Trigger	Get_Attribute_Single	Defines behavior of connection	n/a	BYTE
4	Produced Conn Id	Get_Attribute_Single	Placed in CAN identifier field when the connection transmits	n/a	UINT
5	Consumed Conn Id	Get_Attribute_Single	CAN identifier field value that denotes message to be received	n/a	UINT
6	Initial Comm Characteristics	Get_Attribute_Single	Defines the Message Group(s) across which productions and consumptions associated with this connection	n/a	BYTE
7	Produced Conn Size	Get_Attribute_Single	Maximum number of bytes transmitted across this connection	32,4,32	UINT
8	Consumed Conn Size	Get_Attribute_Single	Maximum number of bytes received across this connection	32,4,32	UINT
9	Expected Packet Rate	Get_Attribute_Single Set_Attribute_Single	Defines timing associated with this connection	n/a	UINT
12	Watchdog Timeout Action	Get_Attribute_Single	Defines how to handle inactivity/watchdog timeouts	n/a	USINT
13	Produced Connection Path Length	Get_Attribute_Single	Number of bytes in produced_connection_path length attribute	6	UINT
14	Produced Connection Path	Get_Attribute_Single Set_Attribute_Single	Application object producing data on this connection	20 04 24 64 30 03, n/a, n/a	ARRAY OF: USINT
15	Consumed Connection Path Length	Get_Attribute_Single	Number of bytes in the Consumed_Connection_Path length attribute	6	UINT
16	Consumed Connection Path	Get_Attribute_Single Set_Attribute_Single	Specifies the application object(s) that are to receive data consumed by this connection object	20 04 24 96 30 03, n/a, n/a	ARRAY OF: 01 UINT
17	Production Inhibit Time	Get_Attribute_Single	Minimum delay time between new data production.	n/a	UINT

NB: Attributes 14,15 and 16 can only be set when the connection is configuring.

## Acknowledge Handler Object, Class 0x2B

### Class Attributes (0)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
1	Revision	Get_Attribute_Single	Revision of assembly object	1,1,65535	UINT
2	Max Instance	Get_Attribute_Single	Maximum instance number of an object currently created in this class level of the device		UINT

### Instance Attributes (1)

#	Attribute Name	Services	Description	Default,Min,Max	Data Type
1	Acknowledge Timer	Get_Attribute_Single Set_Attribute_Single	Time to wait for acknowledge before resending	20,1, 65535	UINT
2	Retry Limit	Get_Attribute_Single Set_Attribute_Single	Number of Ack timeouts to wait before informing the producing application of a RetryLimit_Reached event	1,0,255	USINT
3	COS Producing Connection Instance	Get_Attribute_Single Set_Attribute_Single	Connection instance which contains the path of the producing IO application object which will be notified of Ack handle error event	n/a	UINT
4	Ack List Size	Get_Attribute_Single	Maximum number of member in Ack list	n/a	BYTE
5	Ack List	Get_Attribute_Single	List of active connection instances, which are receiving Acks	n/a	BYTE Array of USINT
6	Data with Ack Path List Size	Get_Attribute_Single	Maximum number of members in data with Ack path list	n/a	BYTE
7	Data with Ack Path List	Get_Attribute_Single	List of connection instance/consuming application object pairs. This attribute is used to forward data received with acknowledgment.	n/a	BYTE Array of UINT USINT Array of USINT

## Position Controller Supervisor Object (0xA4)

This section describes the required behavior of the Position Controller Supervisor (0xA4).

### Class Attributes (0)

#	Attribute Name	Service Name	Description	Default,Min,Max	Data Type
32	Consumed Command Message	Get_Attribute_Single	Contains the content of the Command Message		Contains the content of the Command Message
33	Produced Response Message	Get_Attribute_Single	Contains the content of the Response Message		Contains the content of the Response Message

### Object Instance Attributes(1)

#	Attribute Name	Service Name	Description	Default	Range	Data Factor	Data Type
2	Attribute List	Get_Attribute_Single	Returns an array with a list of the attributes supported by this object in this device.	1,2,3,5,6,7,8,11,16,100,101,102,103,104,105,106,107	See Default column	1	ARRAY OF UNSIGNED SHORT INTEGERS
3	Axis Number	Get_Attribute_Single	Returns the axis number which is the same as the instance for this object. The value can be in the range from 0 to 7, but will always be 1 for the Gemini single axis controllers	1	0-7	1	UNSIGNED SHORT INTEGER
5	General Fault	Get_Attribute_Single	The logical OR of all fault condition attribute flags in the device. This bit is reset when the fault condition is removed. When active, this indicates a drive failure has occurred (bridge fault, overcurrent, etc.).	0	0 – No Fault 1 – Fault	1	BOOLEAN
6	Input Command Message Type	Get_Attribute_Single Set_Attribute_Single	Specifies which Command Message Type is used during polled I/O commands.	0	Dependent on Product 1 to 1F is Valid 0 is 'no I/O command'	1	UNSIGNED SHORT INTEGER
7	Output Command Message Type	Get_Attribute_Single Set_Attribute_Single	Specifies which Response Message Type is used during polled I/O commands.	1	Dependant on Product 1 to 1F is Valid 0 is 'no I/O command returned'	1	UNSIGNED SHORT INTEGER
8	Fault Input	Get_Attribute_Single	Displays the state of the Hardware Fault Input.	0	0 - Inactive 1 - Active	1	BOOLEAN
11	Home Active Level	Get_Attribute_Single Set_Attribute_Single	Sets the active level of the Hardware Home Input.	0	0 – Low 1 – High	1	BOOLEAN
16	Home Input Level	Get_Attribute_Single	Actual level of the Home Input.	0	0 – Off 1 – ON	1	BOOLEAN
100	Go Home (HOM)	Set_Attribute_Single	Used to initiate homing function in specified direction.	0	0 – Negative direction	1	BOOLEAN

					1 – Positive Direction		
101	Home Velocity (HOMV)	Get_Attribute_Single Set_Attribute_Single	Specifies initial homing velocity.	1.0000	0.0000 – 200.0000	0.0001	DOUBLE INTEGER
102	Home Final Velocity (HOMVF)	Get_Attribute_Single Set_Attribute_Single	Specifies final approach homing velocity	0.1000	0.0000 – 200.0000	0.0001	DOUBLE INTEGER
103	Home Acceleration (HOMA)	Get_Attribute_Single Set_Attribute_Single	Specifies accel/decel rate for next Go Home command	10.0000	0.0001-9999.9999	0.0001	DOUBLE INTEGER
104	Home Backup Enable (HOMBAC)	Get_Attribute_Single Set_Attribute_Single	Enables/disables backup to home switch function	0	0 – Disable 1 – Enable	1	BOOLEAN
105	Home Final Direction (HOMDF)	Get_Attribute_Single Set_Attribute_Single	Specifies final approach direction	0	0 – Positive Direction 1 – Negative Direction	1	BOOLEAN
106	Home Reference Edge (HOMEDG)	Get_Attribute_Single Set_Attribute_Single	Specifies which edge of the home switch the homing operation considers as its final destination.	0	0 – Positive Direction Edge 1 – Negative Direction Edge	1	BOOLEAN
107	Home to Encoder Z-Channel Enable (HOMZ)	Get_Attribute_Single Set_Attribute_Single	Enables homing to an encoder z-channel after initial home input has gone active.	0	0 – Disable 1 – Enable	1	BOOLEAN

## Position Controller Object Class (0xA5)

The position controller object performs the control output velocity profiling and handles input and output to and from the motor drive unit, limit switches, registration, etc.

### Object Instance Attributes

#	Attribute Name	Service Name	Description	Default	Range	Data Factor	Data Type
2	Attribute List	Get_Attribute_Single	Returns an array with a list of the attributes supported by this object in this device.	1,2,5,6,7,8,9,10,11,13,14,15,16,17,20,21,23,26,45,48,52,54,55, 58,100,101,102,103, 104,105, 106,107,108,109,110,111	Defined by attribute 1	1	ARRAY OF UNSIGNED SHORT INTEGERS
5	Profile Units	Get_Attribute_Single Set_Attribute_Single	Sets/returns the number of motor/feedback counts per second for velocity and acceleration/deceleration. This is the ERES command for servos and the DRES command for steppers.	1	$-2^{31}$ to $2^{31}$	1	DOUBLE INTEGER
6	Target Position (D)	Get_Attribute_Single Set_Attribute_Single	Sets/returns the target position in counts.	0	$-2^{31}$ to $2^{31}$	1	DOUBLE INTEGER
7	Target Velocity (V)	Get_Attribute_Single Set_Attribute_Single	Sets/returns the target velocity. The units are set by the PROFILE UNIT(5).	1	0.0000 - 200.0000	0.0001	DOUBLE INTEGER
8	Target Acceleration (A)	Get_Attribute_Single Set_Attribute_Single	Sets/returns the target acceleration. The units are set by the PROFILE UNIT(5).	10	0.0001 - 9999.9999	0.0001	DOUBLE INTEGER
9	Target Deceleration (AD)	Get_Attribute_Single Set_Attribute_Single	Sets/returns the target deceleration. The units are set by the PROFILE UNIT(5).	10	0.0001 - 9999.9999	0.0001	DOUBLE INTEGER
10	Incremental Position Flag (MA)	Get_Attribute_Single Set_Attribute_Single	Sets/returns incremental VS absolute mode.	0	0 - Absolute 1 - Incremental	1	BOOLEAN
11	Start a Profile Move (GO)	Get_Attribute_Single Set_Attribute_Single	Used to load command data, start a profile move, and indicate that a profile move is in progress. This will remain set until the motion is complete (or a hard/soft stop is issued).	0	0 - No Action 1 - Start Move	1	BOOLEAN
13	Actual Position (TPE/PSET)	Get_Attribute_Single Set_Attribute_Single	Reports actual position/Sets a new position.	0	$-2^{31}$ to $2^{31}$	1	DOUBLE INTEGER
14	Actual Velocity (TVELA)	Get_Attribute_Single	Gemini GV6-DN only. Reports the actual velocity.	n/a	-200.000000 to 200.000000	0.0001	DOUBLE INTEGER
15	Commanded Position (TPC)	Get_Attribute_Single	Reports position.	n/a	$-2^{31}$ to $2^{31}$	1	DOUBLE INTEGER
16	Commanded Velocity(TVEL)	Get_Attribute_Single	Reports velocity	n/a	-200.000000 to 200.000000	0.0001	DOUBLE INTEGER
17	Enable (DRIVE)	Get_Attribute_Single Set_Attribute_Single	Enables/disables motor current.	0	0 - Disable 1 - Enable	1	BOOLEAN
20	Smooth Stop (S1)	Get_Attribute_Single Set_Attribute_Single	Causes a controlled deceleration to a stop. This is the !S1 command.	0	0 - No action 1 - Start	1	BOOLEAN
21	Hard Stop (K)	Get_Attribute_Single Set_Attribute_Single	Causes a fast deceleration. This is the !K1 command.	0	0 - No action 1 - Kill	1	BOOLEAN

#	Attribute Name	Service Name	Description	Default	Range	Data Factor	Data Type
23	Direction (D- /D+)	Get_Attribute_Single Set_Attribute_Single	Specifies the direction of travel.	1	0 - Negative 1 - Positive	1	BOOLEAN
45	Maximum Dynamic Following Error (SMPER)	Get_Attribute_Single Set_Attribute_Single	Gemini GV6-DN only. This specifies the maximum following error allowed before a Following Error Fault.	4000	0 to 2 <sup>31</sup>	1	DOUBLE INTEGER
48	Actual Following Error (TPER)	Get_Attribute_Single	Gemini GV6-DN only. Reports the actual following error in counts.	n/a	0 to 2 <sup>31</sup>	1	DOUBLE INTEGER
50	Positive Hardware Limit (TAS . 15)	Get_Attribute_Single	Motion is not allowed in the positive direction when active. Set when the positive limit is active.	0	0 - Positive limit not active 1 - Positive Limit Active	1	BOOLEAN
51	Negative Hardware Limit (TAS . 16)	Get_Attribute_Single	Motion is not allowed in the negative direction when active. Set when the negative limit is active.	0	0 - Negative limit not active 1 - Negative limit active	1	BOOLEAN
52	Soft Limit Enable (LS)	Get_Attribute_Single Set_Attribute_Single	Enables Soft Limit checking.	0	0 - Disable 1 - Enable	1	BOOLEAN
54	Positive Soft Limit Position (LSPOS)	Get_Attribute_Single Set_Attribute_Single	Especifies the positive soft limit position in Position Units.	0	-2 <sup>31</sup> to 2 <sup>31</sup>	1	DOUBLE INTEGER
55	Negative Soft Limit Position (LSNEG)	Get_Attribute_Single Set_Attribute_Single	Specifies the negative soft limit position in Position Units.	0	-2 <sup>31</sup> to 2 <sup>31</sup>	1	DOUBLE INTEGER
58	Load Data Complete	Get_Attribute_Single	Indicates that valid data for a valid I/O command message type has been loaded into the position controller device.	n/a	0 - Invalid Data 1 - Valid Data	1	BOOLEAN
100	Position Bandwidth (DPBW)	Get_Attribute_Single Set_Attribute_Single	Gemini GV6-DN only. Specifies Position loop bandwidth (Hz)	5.0	1.00 - 100.00	Normalized	INTEGER
101	Position Damping Ratio (SGPRAT)	Get_Attribute_Single Set_Attribute_Single	Gemini GV6-DN only SGPRAT command	1	0.500 - 2.000	Normalized	INTEGER
102	Load to Rotor Inertia Ratio (LJRAT)	Get_Attribute_Single Set_Attribute_Single	LJRAT command	0.0	0.0 - 100.00	0.1	INTEGER
103	Load Damping (LDAMP)	Get_Attribute_Single Set_Attribute_Single	Gemini GV6-DN only LDAMP command	0.0000	0.0000 - 1.0000	Normalized	INTEGER
104	Integrator Enable (SGINTE)	Get_Attribute_Single Set_Attribute_Single	Gemini GV6-DN only SGINTE command	1	0 - Disable 1 - Enable	1	BOOLEAN
105	Mode Continuous (MC)	Get_Attribute_Single Set_Attribute_Single	MC command	1	0 - Normal 1 - Mode Continuous	1	BOOLEAN
106	Input Status (TIN)	Get_Attribute_Single		n/a		1	INTEGER
107	Get/Set Outputs (TOUT/OUT)	Get_Attribute_Single Set_Attribute_Single		0		1	INTEGER
108	Variable Access - Variable number	Get_Attribute_Single Set_Attribute_Single	Allows access to VARI 1-99	1	1-99	1	INTEGER
109	Variable Access - Variable value	Get_Attribute_Single Set_Attribute_Single	Accesses contents of variable defined in #109	0	-2 <sup>31</sup> to 2 <sup>31</sup> -1	1	DOUBLE INTEGER
110	Hardware Limit (LH)	Get_Attribute_Single Set_Attribute_Single	LH command	0	0-3	1	INTEGER
111	Drive Status (TAS)	Get_Attribute_Single	Reports the drive status of the drive. See TAS command	n/a	0000_0000_ 0000_0000 to 1111_1111_	1	DOUBLE INTEGER

#	Attribute Name	Service Name	Description	Default	Range	Data Factor	Data Type
112	Error Status (TASX)	Get_Attribute_Single	Reports the error status of the drive. See TASX command	n/a	1111_1111 0000_0000_ 0000_0000 to 1111_1111_ 1111_1111	1	DOUBLE INTEGER
113	Positive Torque Limit (DMTIP)	Get_Attribute_Single Set_Attribute_Single	Gemini GV6-DN only. Specifies the maximum torque the motor can achieve.	depends on motor	0.00 - 128.00	Normalized	INTEGER

## Command Assemblies

This section defines the command assembly types. This table shows the format for a Command Assembly I/O message.

### Command Assembly

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Reg Arm	Hard Stop	Smooth Stop	Direction (V Mode)	Absolute / Incremental	Start Program	Start Trajectory
1	RUN (0) / PRUN (1)	Program #						
2	Axis Instance (001)			Command Assembly Code				
3	Axis Instance (001)			Response Assembly Code				
4	Data Low Byte							
5	Data Low Middle Byte							
6	Data High Middle Byte							
7	Data High Byte							

Command	Description
Start Trajectory (Edge triggered)	Starts a move. Setting this bit to 1 in conjunction with command assembly 0x01 (positioning). For all other command assemblies, the transition of this bit sets the data value(i.e. velocity, acceleration, etc.).
Start Program (Edge triggered)	By setting this bit high (1) , the command will execute programs and profiles previously generated by Motion Planner and stored in the drive. The program executed is defined by the second byte of the Command Assembly. To stop program execution, set Start Program high (1) and Program Number to 0. Programs can be executed with any Command Assembly.
Absolute/ Incremental	Defines the position value as either absolute or incremental. This data is valid when the Start Trajectory bit is set. 0 = absolute position value; 1 = incremental or relative position value.
Direction (Velocity Mode)	Controls the direction of the motor in Profiled Velocity mode. This bit is only valid in 0x11 (velocity mode) command assembly. 1 = forward, positive or clockwise; 0 = reverse, negative or counter clockwise.
Smooth Stop (Level triggered)	Brings the motor to a controlled stop at the currently implemented deceleration rate.
Hard Stop (Level Triggered)	Brings the motor to an immediate stop.
Reg Arm (Level Triggered)	By setting this bit, the registration input will be armed. If a registration input is then triggered the registration action will be executed if a profile is currently being executed. Clearing this bit will turn off registration.
Enable (Level triggered)	Enables/disables the drive. Clearing this bit will disable the drive and the currently executing motion profile will be aborted. This attribute is level, not edge, sensitive.
Program #	Used in conjunction with Start Program to run a program or profile previously defined in the drive. With Start Program set to high (1) , the drive will execute the program indicated by the second byte of the Assembly. To stop a program, set Start Program high (1) and Program Number to zero (0). Programs can be executed with any Command Assembly.
Axis Instance	Always set to 1. Any other values should cause a command error.
Command Assembly Code	Defines the Command Assembly Code. The first four bytes of the command always have the same meaning. The last four bytes are defined by the command assembly code.
Response Assembly Code	Defines the Response Assembly Code. The first four bytes of the response always have the same meaning. The last four bytes are defined by the response assembly code.

## Response Assemblies

This section defines the response assembly types. Here is the format for a Response Assembly I/O message. Each response assembly should echo the trajectory start bit from the previous command assembly.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable State	Reg Level	Home Level	Current Direction	General Fault (Alarm)	On Target Position	Program In Execution	Trajectory In Progress
1	Executing Program # / Attribute number							
2	Load Complete	Program Fault	Trajectory Start Echo	Negative Software Limit	Positive Software Limit	CCW Hardware Limit	CW Hardware Limit	Fault Input Fault
3	Axis Instance (001)			Response Assembly Code				
4	Data Low Byte							
5	Data Low Middle Byte							
6	Data High Middle Byte							
7	Data High Byte							

Command	Description
Trajectory in Progress	Indicates whether or not a trajectory move is commanded or has completed. "0" indicates a move has completed. "1" indicates a move has been commanded. For Home routines, the trajectory in progress bit is used to determine home complete. The "Trajectory in Progress" bit must be set high during the data valid response to a successful command.
Program in Execution	Indicates that a Program is in execution. The Program that is currently being executed is returned in byte 1 of the response assembly.
On Target Position	Indicates whether or not the motor is within an end of move condition. The end of move condition can be position, velocity and/or time based. This bit is set to 1 indicates 'In Position'. The attribute used to define the end of move position condition is specified in Object 0x25, attribute 38, position deadband.
General Fault (Alarm)	Set if a device alarm occurs; 1 = alarm, 0 = no alarms.
Current Direction	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.
Home Level	The Home Flag; 1 = off home, 0 = on home.
Reg Level	Reflects the level of the registration input.
Enable State	Indicates whether the drive is enabled or disabled. A 1 indicates the device is enabled.
Fault Input Fault	Indicates the fault input is active.
CW/CCW Hardware Limit	Indicates that the hardware limits are active; 1 = active.
Negative/Positive Software Limit	Indicate that the motor has attempted a move past the programmed limit position; 1 = active.
Trajectory Start Echo	This is a copy of the Start Trajectory bit in the Command Assembly (bit 0 of byte 1).
Program Fault	Indicates that a program execution fault occurred. When this happens, the program/profile move will stop. 1 = Program Fault.
Load Complete	Indicates that the command data contained in the command message has been successfully loaded into the device.

## Running a Stored Program through DeviceNet

1. Create and store a program in the drive with Motion Planner
2. Convert the program number from decimal (a.k.a. Prog32) to binary.  
For example, if you store a program as PROG5, this is converted into binary as 00000101 (1+4).
3. Save the binary value into byte 1 of the command assembly for standard programs.  
Add 128 to the binary value (ie set bit 7) for compiled programs, and save into byte 1 again.
4. Using any of the command assemblies, send out Start Program = 1.

### Example:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	X	X	X	X	X	1	X
1	0	0	1	0	0	0	0	0
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X

This would run program 32. Bit 1 of byte 0 must transition from a “0” to a “1” for the drive to begin running the program.

## Stopping the Program through DeviceNet

To stop the program, the user must send out a “0” in the Program Number with Start Program = 1.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	X	X	X	X	X	X	1	X
1	0	0	0	0	0	0	0	0
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X

# Command Assembly Codes

Available command assembly codes:

Code	Description	Data Type
0x00	No Operation	-
0x01	Target Position	DINT
0x02	Target Velocity	DINT
0x03	Acceleration	DINT
0x04	Deceleration	DINT
0x05 to 0x0B	Reserved	
0x0C	Variable Access	
0x0D to 0x10	Reserved	
0x11	Continuous Velocity	DINT
0x12	Home Type	USINT
0x13 to 19	Reserved	
0x1A	Position Controller Supervisor Attribute	DINT
0x1B	Position Controller Attribute	DINT
0x1C to 1D	Reserved	
0x1E	Alarm Clear	BOOL
0x1F	Reserved	

## No Operation 0x00

This code does not have any meaning. The command data is ignored.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								Data Low Byte
5								Data Low Middle Byte
6								Data High Middle Byte
7								Data High Byte

## Target Position 0x01

The data bytes define the target position in units of steps (1 LSB = 1 step). This data is valid when the Valid Data bit is set.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								Target Position Low Byte
5								Target Position Low Middle Byte
6								Target Position High Middle Byte
7								Target Position High Byte

## Target Velocity 0x02

The data bytes define the target velocity in units of steps/sec (1 LSB = 1 step/sec). This data is valid when the Valid Data bit is set. The data is always a positive number.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								Target Velocity Low Byte
5								Target Velocity Low Middle Byte
6								Target Velocity High Middle Byte
7								Target Velocity High Byte

## Acceleration 0x03

The data bytes define the acceleration in units of steps/sec<sup>2</sup> (1 LSB = 1 step/sec<sup>2</sup>). This data is valid when the Valid Data bit is set. The data is always a positive number. The data is used for positioning, velocity and home modes.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Acceleration Low Byte							
5	Acceleration Low Middle Byte							
6	Acceleration High Middle Byte							
7	Acceleration High Byte							

## Deceleration 0x04

The data bytes define the acceleration in units of steps/sec<sup>2</sup> (1 LSB = 1 step/sec<sup>2</sup>). This data is valid when the Valid Data bit is set. The data is always a positive number. The data is used for positioning, velocity and home modes.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Deceleration Low Byte							
5	Deceleration Low Middle Byte							
6	Deceleration High Middle Byte							
7	Deceleration High Byte							

## Variable Access 0x0C

Gemini variables can be stored via this command. The value in data byte 1 (range 1-99) determines which variable is modified. Data bytes 4-7 contain the double integer value to be stored in the variable. This data is valid when the Valid Data bit is set.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Reg Arm	Hard Stop	Smooth Stop	Direction (V Mode)	Absolute / Incremental	Start Program	Start Trajectory
1	DeviceNet variable 1 through 99							
2	0	0	1	0	0x0C			
3	0	0	1	0	Response Assembly Code			
4	DeviceNet variable Value Low Byte							
5	DeviceNet variable Value Low Middle Byte							
6	DeviceNet Variable Value High Middle Byte							
7	DeviceNet variable Value High Byte							

## Continuous Velocity 0x11

The data bytes define the continuous velocity in units of steps/sec (1 LSB = 1 step/sec). The direction of the velocity is defined by the Direction (Velocity Mode) bit. This data is valid when the Valid Data bit is set. The data is always a positive number.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Continuous Velocity Low Byte							
5	Continuous Velocity Low Middle Byte							
6	Continuous Velocity High Middle Byte							
7	Continuous Velocity High Byte							

## Homing 0x12

The data byte defines the Homing direction; 0 is positive, 1 is negative. The data is valid when the Valid Data bit is set. Home starts after setting the Start Trajectory bit. Bytes 5, 6 and 7 should be set to 0x00.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	Homing Direction							
5	0x00 (fixed)							
6	0x00 (fixed)							
7	0x00 (fixed)							

## Position Controller Supervisor Attribute 0x1A

This assembly allows any attribute in the Position Controller Supervisor group to be set or accessed. The command assembly allows a set and a get function to be performed simultaneously.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Reg Arm	Hard Stop	Smooth Stop	Direction (V Mode)	Absolute / Incremental	Start Program	Start Trajectory
1	Attribute to Get							
2	0	0	1	1	0xA			
3	Attribute to Set							
4	Position Controller Supervisor attribute value Low Byte							
5	Position Controller Supervisor attribute value Low Middle Byte							
6	Position Controller Supervisor attribute value High Middle Byte							
7	Position Controller Supervisor attribute value High Byte							

## Position Controller Attribute 0x1B

This assembly allows any attribute in the Position Controller group to be set or accessed. The command assembly allows a set and a get function to be performed simultaneously.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable	Reg Arm	Hard Stop	Smooth Stop	Direction (V Mode)	Absolute / Incremental	Start Program	Start Trajectory
1	Attribute to Get							
2	0	0	1	1	0xB			
3	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							

## Alarm Clear 0x1E

The data bit clears the alarm occurring on the drive. Setting the Alarm Clear to 1 clears the alarm condition, on the rising edge of the “Start Trajectory” bit, with Data valid set high. Note, some alarm states can not be cleared using Alarm Clear 0x1E. In this case, the drive must be power cycled. This bit does not clear alarms due to communications errors. Sending Alarm Clear = 0 is equivalent to Command Assembly 0x00 (No Operation).

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4	0 (fixed)	0 (fixed)	0 (fixed)	0 (fixed)	0 (fixed)	0 (fixed)	0 (fixed)	Alarm Clear
5					0x00 (fixed)			
6					0x00 (fixed)			
7					0x00 (fixed)			

# Response Assembly Codes

Code	Description	Data Type
0x01	Actual Position - servo only	DINT
0x02	Commanded Position	DINT
0x03	Actual Velocity - servo only	DINT
0x04	Command Velocity	DINT
0x04 to 0x0B	Reserved	
0x0C	Variable access	DINT
0x0D to 0x13	Reserved	
0x14	Command/Response Error	DINT
0x15 to 0x19	Reserved	
0x1A	Position Controller Supervisor Attribute	DINT
0x1B	Position Controller Attribute	DINT
0x1E	Alarm Code	DINT
0x1F	Reserved	

## Actual Position 0x01 - Servo Only

This double word reflects the actual encoder position in units of steps (1 LSB = 1 step).

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								Actual Position Low Byte
5								Actual Position Low Middle Byte
6								Actual Position High Middle Byte
7								Actual Position High Byte

## Commanded Position 0x02

This double word reflects the commanded or calculated position in units of steps (1 LSB = 1 step).

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								Commanded Position Low Byte
5								Commanded Position Low Middle Byte
6								Commanded Position High Middle Byte
7								Commanded Position High Byte

## Actual Velocity 0x03 - Servo Only

This double word reflects the actual velocity in units of steps/sec (1 LSB = 1 step/sec). The data is always a positive number.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								Actual Velocity Low Byte
5								Actual Velocity Low Middle Byte
6								Actual Velocity High Middle Byte
7								Actual Velocity High Byte

## Commanded Velocity 0x04

This double word reflects the commanded velocity in units of steps/sec (1 LSB = 1 step/sec). The data is always a positive number.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								Commanded Velocity Low Byte
5								Commanded Velocity Low Middle Byte
6								Commanded Velocity High Middle Byte
7								Commanded Velocity High Byte

## Variable access 0x0C

Gemini variables can be read via this command. The value in command assembly data byte 1 (range 1-99) determines which variable is read. Response assembly bytes 4-7 contain the double integer variable value.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4								DeviceNet Variable Value Low Byte
5								DeviceNet variable Value Low Middle Byte
6								DeviceNet variable Value High Middle Byte
7								DeviceNet Variable Value High Byte

## Command/Response Error 0x14

This assembly indicates whether there was an error in the command or response assembly, along with an error code describing the problem.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable State	Reg Level	Home Level	Current Direction	General Fault (Alarm)	On Target Position	Program In Execution	Trajectory In Progress
1	Executing Program #							
2	Load Complete	Program Fault	Trajectory Start Echo	Negative Software Limit	Positive Software Limit	CCW Hardware Limit	CW Hardware Limit	Fault Input Fault
3	Axis Instance (001)			Response Message Code (0x14)				
4	General Error code							
5	Additional Code							
6	Copy of Command Message Byte 2							
7	Copy of Command Message Byte 3							

The following is an explanation of the various error codes reported by this command:

General Error Code	Additional Code	Response	Semantics
0x08	0x01	Service Not Supported	Command Message type not supported. Additional code 0x01 takes precedence over code 0x02
	0x02	Service Not Supported	Response type not supported
0x05	0x01	Path Destination Unknown	Command message axis number invalid
	0x02	Path Destination Unknown	Response message axis number invalid
0x09	0xFF	Invalid Attribute Value	Load value out of range
0x0E	0xFF	Attribute not Settable	A request to modify a non-modifiable attribute was received
0x13	0xFF	Not Enough Data	I/O command message contained fewer than 8 bytes.
0x14	0xFF	Attribute Not Supported	Attribute specified in request was not supported

## Position Controller Supervisor Attribute 0x1A

This assembly reflects the value of the attribute in command assembly data byte 1 when accessing the Position Controller Supervisor.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable State	Reg Level	Home Level	Current Direction	General Fault (Alarm)	On Target Position	Program In Execution	Trajectory In Progress
1	Attribute number							
2	Load Complete	Program Fault	Trajectory Start Echo	Negative Software Limit	Positive Software Limit	CCW Hardware Limit	CW Hardware Limit	Fault Input Fault
3	Axis Instance (001)			Response Message Code (0x1A)				
4	Position Controller Supervisor Low Byte							
5	Position Controller Supervisor Low Middle Byte							
6	Position Controller Supervisor High Middle Byte							
7	Position Controller Supervisor High Byte							

## Position Controller Attribute 0x1B

This assembly reflects the value of the attribute in command assembly data byte 1 when accessing the Position Controller.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Enable State	Reg Level	Home Level	Current Direction	General Fault (Alarm)	On Target Position	Program In Execution	Trajectory In Progress
1	Attribute number							
2	Load Complete	Program Fault	Trajectory Start Echo	Negative Software Limit	Positive Software Limit	CCW Hardware Limit	CW Hardware Limit	Fault Input Fault
3	Axis Instance (001)			Response Message Code (0x1B)				
4	Position Controller Low Byte							
5	Position Controller Low Middle Byte							
6	Position Controller High Middle Byte							
7	Position Controller High Byte							

## Status Code 0x1E

Indicates the status code. The functional mapping should follow the TAS command in the Gemini controller.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4					Status Code			
5					Status Code			
6					Status Code			
7					Status Code			

## Alarm Code 0x1F

Indicates the alarm code. A 0x00000000 alarm code is when no alarms have occurred. The specification doesn't require a specific mapping between alarms to alarm codes; however, a list of the alarm conditions are in the table below.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4					Alarm Code			
5					Alarm Code			
6					Alarm Code			
7					Alarm Code7			

### Alarm Codes

Alarm Code (low byte)	Alarm Description	Cleared by Alarm Clear
0x01	Motor temperature fault	
0x02	Low voltage fault	
0x03	Drive over-temperature fault	
0x04	Duplicate MAC ID Error	
0x05	Encoder failed	
0x06	Dual Port Ram Error	
0x07	Motor configuration error	
0x08	Incoming steps during startup	
0x09	Velocity Error limit	
0x10	Bridge Fault	
0x11	Bridge temperature fault	
0x12	Over-voltage	
0x13	Bus Off Error	
0x14	EEPROM Error	
0x15	Memory Error	
0x16	Over Current	
0x17	Stall Detected	
0x18	Override mode/Commanded velocity error	
0x19	Bridge is in foldback mode	
0x20	Power Disipation circuit has been active	
0x21	Bad hall state detected	

Alarm Code (low byte)	Alarm Description	Cleared by Alarm Clear
0x22	Unrecognized hardware	
0x23	User fault	
0x24	Keep alive active	
0x25	Power Disipation circuit fault	
0x26	Reserved	
0x27	Reserved	
0x28	Motor configuration warning	
0x29	ORES failure	
0x30	Motor thermal model fault	
0x31	Commanded torque/force is at limit	
0x32	Reserved	

# Operating Modes

This specification defines the following operating modes:

Positioning .....	See page 52
Continuous Velocity .....	See page 54
Homing .....	See page 56

## Positioning

Figure 1 describes the positioning mode. Note, all messages and I/O states are from the device perspective. The general positioning procedure is as follows:

1. Specify the relevant move parameters to the driver (target velocity 0x02, acceleration 0x03, and deceleration 0x04).
2. Specify the Target Position (0x01), either Absolute/Incremental positioning, and set the Start Trajectory bit.

The following is an example Command Assembly for positioning mode. For this example, the position data is 128,000 steps or 0x0001F400 and the move is absolute. The device is asked to respond with Actual Velocity 0x03.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1 Enable	0 Reg Arm	0 Hard Stop	0 Smooth Stop	0 Direction (V Mode)	0 Absolute / Incremental	0 Start Block	1 Start Trajectory
1	0x00							
2	001 Axis Instance			0x01 Command Assembly Code				
3	001 Axis Instance			0x03 Response Assembly Code				
4	0x00 Target Position Low Byte							
5	0xF4 Target Position Low Middle Byte							
6	0x01 Target Position High Middle Byte							
7	0x00 Target Position High Byte							

The Trajectory In Progress bit is set when the move begins. The bit is cleared when the pre-programmed in-position conditions have been met. The positioning mode is disabled when an alarm is reported.

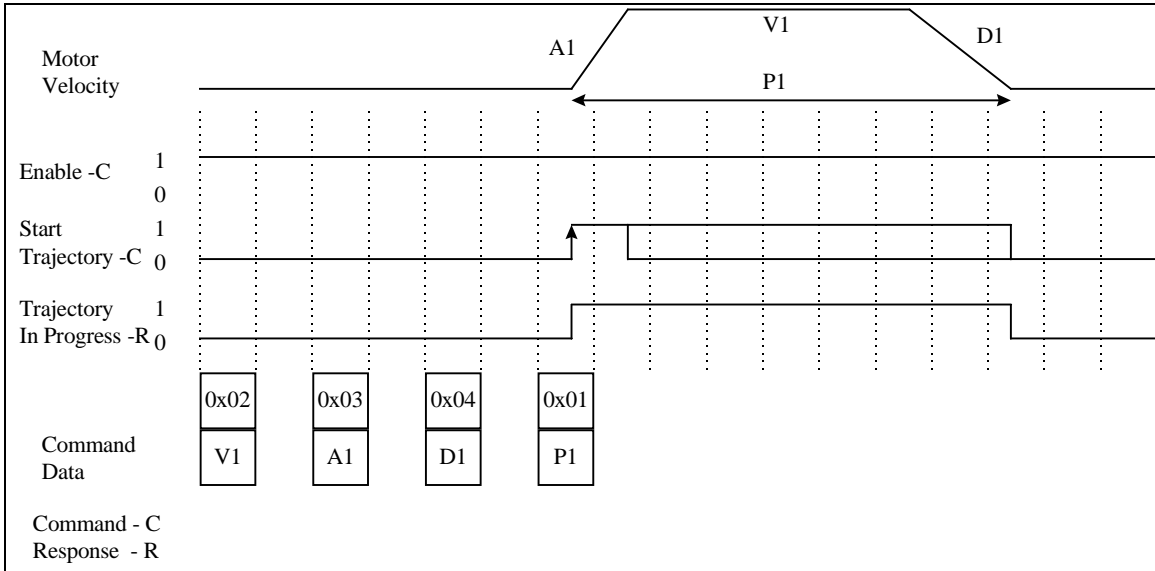


Figure 1. Positioning Mode

## Continuous Velocity

Figure 2 describes the continuous velocity mode. The general method for defining a continuous velocity move is as follows:

1. Specify the relevant move parameters to the driver (acceleration 0x03, and deceleration 0x04)

Specify the Continuous Velocity (0x11), the direction bit, and set the Start Trajectory bit.

To stop the continuous velocity move, send a zero velocity move, set the smooth stop or hard stop bits, or toggle the direction bit (this will perform a hard stop).

The Trajectory In Progress bit is set when the move begins. The bit is cleared when the pre-programmed in-position conditions have been met. In Velocity Mode, the velocity and direction data can be modified while the move is occurring.

The following is an example Command Assembly for the Continuous Velocity mode. For this example, the direction is positive and the target velocity is 0x06666666 or 10rps (0x7FFFFFFF is 200rps). The device is asked to respond with Actual Velocity 0x03.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1 Enable	0 Reg Arm	0 Hard Stop	0 Smooth Stop	1 Direction (V Mode)	0 Absolute / Incremental	0 Start Block	1 Start Trajectory
1	0x00							
2	001 Axis Instance			0x11 Command Assembly Code				
3	001 Axis Instance			0x03 Response Assembly Code				
4	0x66 Target Velocity Low Byte							
5	0x66 Target Velocity Low Middle Byte							
6	0x66 Target Velocity High Middle Byte							
7	0x06 Velocity Position High Byte							

The Trajectory In Progress bit is set when the move begins. The bit is cleared when the motor is stopped by setting the Smooth Stop bit. The continuous velocity mode is disabled when an alarm is reported.

By sending another continuous velocity command with another target velocity the motor will accelerate or decelerate to the new velocity. In order to change direction a soft or hard stop must be issued and the motor must come to rest before the new direction ("0" in this case) is issued and the 'Start Trajectory' bit is again set to "1" to initiate the move.

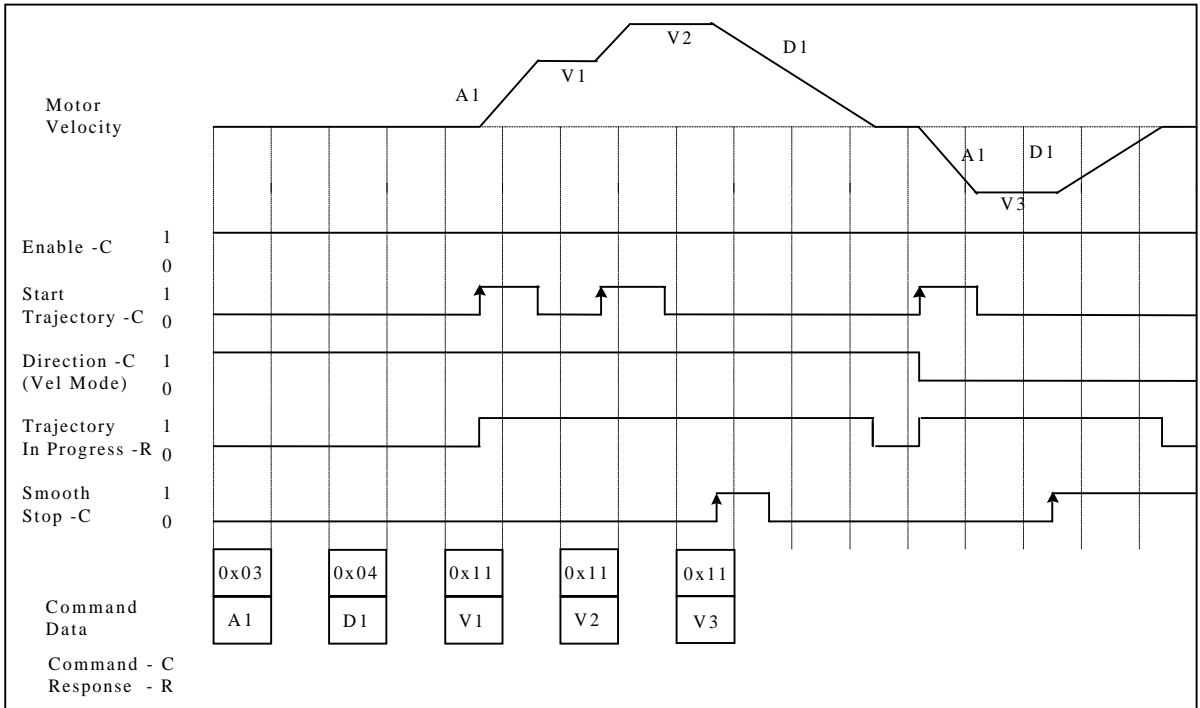


Figure 2. Continuous Velocity

# Homing

Homing is done by sending the Home Type Command Assembly (0x12), setting the home direction and start trajectory bit.

## Example 1

Figure 3 presents example 1. This Home method uses the home flag and the index or Z phase of the encoder to define home.

Object 0x24, Object instance attributes 100-107 will define the velocity and direction attributes to perform the home. Figure 3 shows the attributes set-up in the following fashion: HOMZ1 (use encoder z-pulse) HOMBAC1 (enable back-up to home flag), HOMDF1 (final direction negative) and HOMDGO (stop on positive travel side of active region). and.

The general procedure is as follows:

1. Move in a positive direction towards the home flag direction.
2. When the home flag changes state, reverse direction until the home flag returns to the original state.
3. Once the motor has moved past the home flag, continue moving until the first index pulse of the encoder.
4. Bring motor to a stop, perform a residual move back to the home indicator. This is HOME; Position = 0. Other index pulse may occur during deceleration, these should not be used as home.

Note: The Trajectory In Progress bit reports when Home is complete.

The following is an example Command Assembly for the Homing command. This example assumes the homing parameters have been set up as above. The drive is asked to Home in a positive direction. The device is asked to respond with Actual Velocity 0x03. Homing is complete when the Trajectory in Progress bit returns to "0"

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1 Enable	0 Reg Arm	0 Hard Stop	0 Smooth Stop	0 Direction (V Mode)	0 Absolute / Incremental	0 Start Block	1 Start Trajectory
1	0x00							
2	001 Axis Instance			0x12 Command Assembly Code				
3	001 Axis Instance			0x03 Response Assembly Code				
4	0x00 Home Direction							
5	0x00							
6	0x00							
7	0x00							

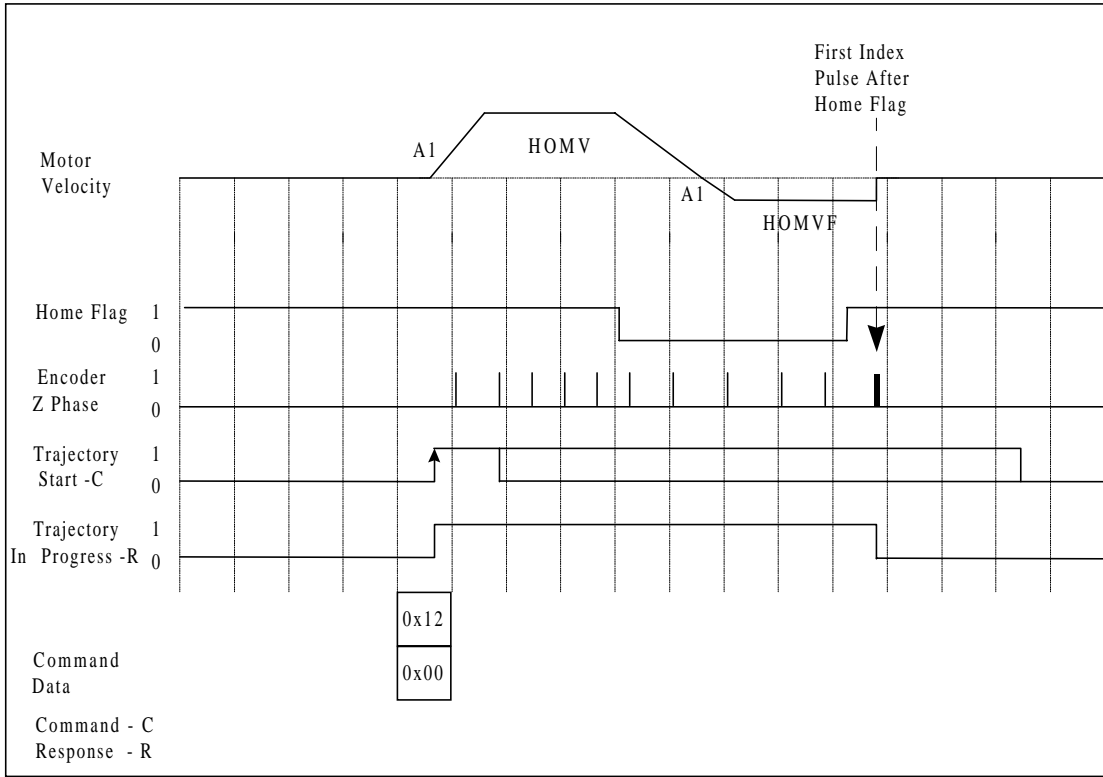


Figure 3. Home Example 1

## Example 2

Figure 4 presents example 2. This Home method uses the home flag to define home.

Object 0x24, Object instance attributes 100-107 will define the velocity and direction attributes to perform the home. Figure 3 shows the attributes set-up in the following fashion: HOMZ0 (do not use encoder z-pulse) HOMBAC1 (enable back-up to home flag), HOMDF0 (final direction positive) and HOMDG1 (stop on negative travel side of active region).

The general procedure is as follows:

1. Move towards the home flag transition.
2. When the home flag changes state, reverse direction until the home flag returns to the original state.
3. Move forward again until the home flag transition.
4. This is HOME; Position = 0.

Note: The Trajectory In Progress bit reports when home is complete.

The following is an example Command Assembly for the Homing command. This example assumes the homing parameters have been set up as above. The drive is asked to Home in a negative direction. The device is asked to respond with Actual Velocity 0x03. Homing is complete when the Trajectory in Progress bit returns to "0"

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1 Enable	0 Reg Arm	0 Hard Stop	0 Smooth Stop	0 Direction (V Mode)	0 Absolute / Incremental	0 Start Block	1 Start Trajectory
1	0x00							
2	001 Axis Instance			0x12 Command Assembly Code				
3	001 Axis Instance			0x03 Response Assembly Code				
4	0x01 Home Direction							
5	0x00							
6	0x00							
7	0x00							

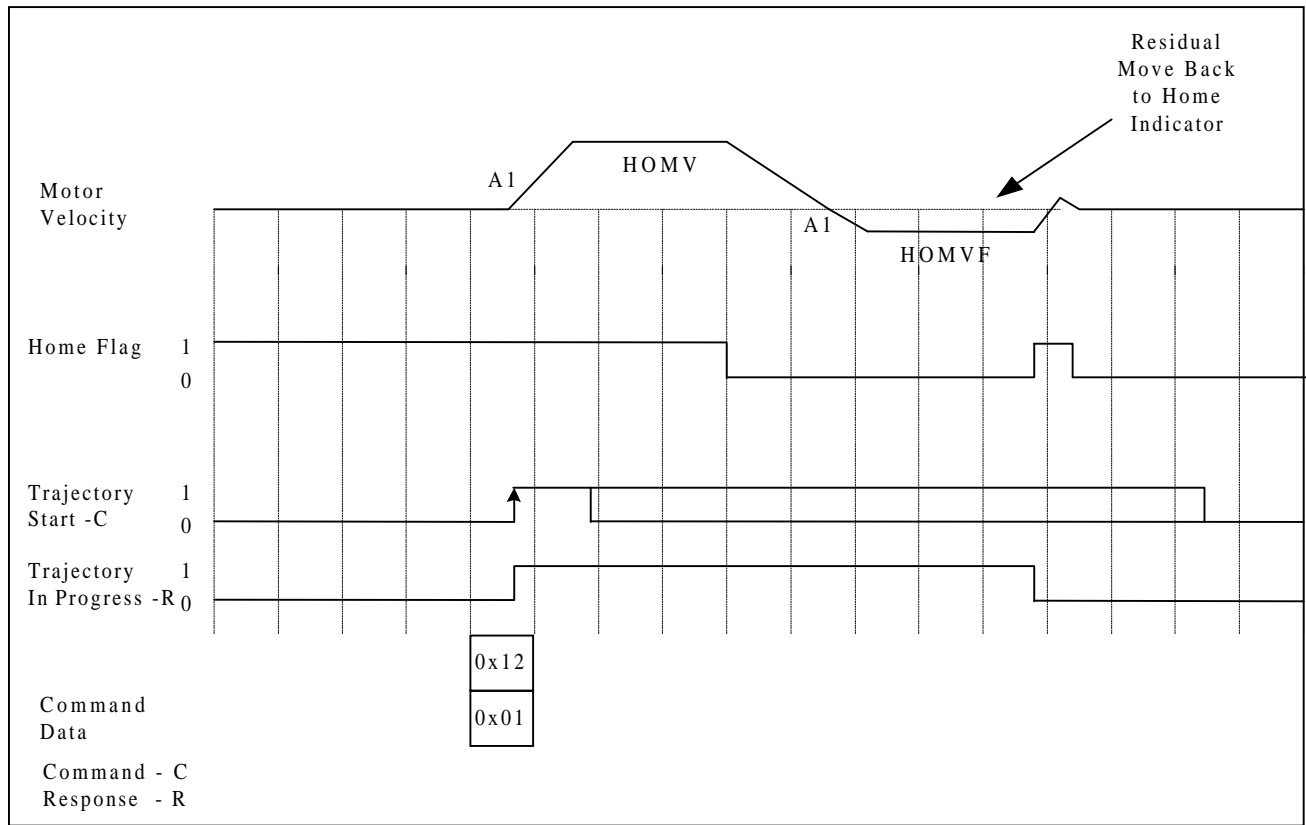


Figure 4. Home Example 2

# Command Descriptions

The following is a list of all the DeviceNet specific commands. For a complete listing of Gemini commands see the *Gemini Series Programmer's Guide*.

- ERROR (Error Checking Enable) .....See page 60
- FBADDR (Fieldbus Address) .....See page 60
- FBBAUD (Fieldbus Baudrate Setting).....See page 61
- FBMASK (Fieldbus I/O Mask) .....See page 61
- [FBS] (Fieldbus Status).....See page 62
- OUTFNC (Output Function).....See page 63
- TASX (Transfer Extended Axis Status) .....See page 63
- TFBS (Transfer Fieldbus Status) .....See page 63

---

## ERROR Error Checking Enable

<b>Type:</b>	Communication Setup	<b>Product</b>	<b>Rev</b>
<b>Syntax:</b>	<a_><!>ERROR<b><b>...<b><b> (32bits)	GV6-DN	2.00
<b>Units:</b>	n/a	GT6-DN	
<b>Range:</b>	b=0 (disable), 1 (enable), or X (don't change)		
<b>Default:</b>	0		
<b>Response:</b>	ERROR: *ERROR0000_0000_0000_0000_0000_0000_0000_0000		
<b>See Also:</b>	[ASX], ERRORP, INFNC, LH, LS, S, TASX, TER, TRGFN		

---

A new bit assignment is added for the ERROR command, bit #19.

See FBS on page 62 for the event causing the error condition. To clear the error event, first resolve the cause, and then issue the ERROR.19-0 command followed by the ERROR.19-1 command. Error bit 19 is edge sensitive to error events.

Bit#	Function	Branch Type
19	Option card fault	GOTO

---

## FBADDR Fieldbus Address

<b>Type:</b>	Communication Setup	<b>Product</b>	<b>Rev</b>
<b>Syntax:</b>	<!>FBADDR<i>	GV6-DN	2.00
<b>Units:</b>	i = fieldbus address	GT6-DN	
<b>Range:</b>	i = 1-125		
<b>Default:</b>	1		
<b>Response:</b>	FBADDR: *FBADDR3		
<b>See Also:</b>	FBSIZE		

---

Use the FBADDR command to report the controller's current node address assignment and set the node address via software. The new value is saved into nonvolatile memory, and becomes effective after the controller is reset. Network configuration of node address is not supported. This command cannot report the hardware configuration setting. In order to set the node address via software, the hardware configuration method must be disabled (default from factory). See Node Address on page 26.

**Example:**

Assume controller was assigned node address 1 out of reset:

```
>FBADDR
*FBADDR1
>FBADDR3      Set node address to 3
>FBADDR
*FBADDR3      New network setting will take effect after unit is reset!
```

---

## FBBAUD      Fieldbus Baudrate Setting

**Type:**            Communication Setup  
**Syntax:**        <!>FBBAUD<r>  
**Units:**           n/a  
**Range:**         r = 0 (125kbaud) , 1 (250kbaud), or 2 (500kbaud)  
**Default:**        2 (500kbaud)  
**Response:**      FBBAUD:        \*FBBAUD2  
**See Also:**      FBADDR, FBSIZE

**Product**   **Rev**  
GV6-DN    2.00  
GT6-DN

---

Use the FBBAUD command to report the controller's current baudrate assignment (used during boot-up) and set the baudrate via software. This command cannot report the hardware configuration setting. In order to set the baudrate via software, the hardware configuration method must be disabled (default from factory). See Baud Rate on page 26.

When setting the baudrate via software, the node address must also be set via software, and vice versa. If the hardware configuration method is used to set the baudrate, an attempt to set the baudrate via software will be ignored. No message is reported indicating success or failure.

The new value is saved into nonvolatile memory, and becomes effective after the controller is reset.

**Example:**

```
>FBBAUD
*FBBAUD1
>FBBAUD1            ;Set baudrate to 250 kbaud
>FBBAUD
*FBBAUD1            ;New network setting will take effect after unit is reset!
```

---

## FBMASK      Fieldbus I/O Mask

**Type:**            I/O Setup  
**Syntax:**        <!>FBMASK <b>, <b>, <b>, <b>, <b>, <b>, <b>, <b>  
**Units:**           n/a  
**Range:**         b=0 (Gemini control), 1 (Fieldbus control)  
**Default:**        0  
**Response:**      FBMASK:        \*FBMASK0000\_0000  
**See Also:**      FBPIC, FBPOC

**Product**   **Rev**  
GV6-DN    2.00  
GT6-DN

---

Use FBMASK to prevent an output from being changed by fieldbus.

**Example:**

FBMASK00100000      Only output #3 can be changed by fieldbus

Response for FBMASK: \*FBMASK0010\_0000



## [ FBS ]                      Fieldbus Status

<b>Type:</b>	Communication Setup	<b>Product</b>	<b>Rev</b>
<b>Syntax:</b>	See below	GV6-DN	2.00
<b>Units:</b>	n/a	GT6-DN	
<b>Range:</b>	n/a		
<b>Default:</b>	n/a		
<b>Response:</b>	n/a		
<b>See Also:</b>	ER.19, TFBS		

Use the FBS command to assign the fieldbus status to a binary variable or for use in a comparison command.

Note: The left-most bit corresponds to OUT.1

**Example:**

```
IF(FBS.4-1)                      ;Branch based on the status of FBS bit 4
```

The Fieldbus Status register bits are defined as follows:

Bit #	Function (1=Yes, 0=No)	Description
1	TIMEOUT <sup>1,2</sup>	Watchdog timed out. Controller has lost communication with fieldbus card
2	CHECKSUM FAULT <sup>1,2</sup>	Fieldbus card failed hardware check on boot-up
3	HWD CFG MODE	0 - Configuration set via software, 1 - Configuration set via hardware
4	NETWORK LINK OK <sup>1,3</sup>	Controller is connected and data exchange is possible
5	NETWORK CRITICAL LINK FAILURE <sup>4</sup>	Cannot communicate with the fieldbus. Duplicate node address or bus-off state exists.
6	NETWORK LINK NOT CONNECTED <sup>4</sup>	Passed Dup_MAC_ID test and online, but no connection established with another node
7	NETWORK CONNECTION TIMEOUT <sup>4</sup>	Connection has timed out
8	MODULE DEVICE OPERATIONAL <sup>4</sup>	Operating in a normal condition
9	MODULE UNRECOVERABLE FAULT <sup>4</sup>	Fieldbus card may need to be replaced
10	MODULE DEVICE IN STANDBY <sup>4</sup>	Device needs commissioning due to configuration missing, incomplete, or incorrect
11	MODULE MINOR FAULT <sup>4</sup>	Fieldbus card is indicating a recoverable fault
12-31	RESERVED	
32	Incorrect Firmware	The DeviceNet card contains incorrect firmware.

<sup>1</sup> If any of these error conditions occur (bit #1 = 1, bit #2 = 1, or bit #4 = 0), the motion controller will perform a Kill (K command) on all axes. If error-checking bit #19 is enabled with the ERROR command (ERROR.19-1) the controller will also set error status bit #19 (ER, TER, and TERF) and branch to the ERRORP program.

<sup>2</sup> Error event is latched. Reset the controller to clear the error.

<sup>3</sup> Error event is recoverable, if error checking (ERROR) bit #19 is enabled and ERRORP program exists. If error bit #19 is disabled or no ERRORP program exists, the event becomes latched, and you will need to reset the controller to clear the error.

<sup>4</sup> After error checking (ERROR) bit #19 is set, it can take up to 600ms to correctly set these status bits.

---

## OUTFNC Output Function

<b>Type:</b>	Output	<b>Product</b>	<b>Rev</b>
<b>Syntax:</b>	<!>OUTFNC<i>-<c>	GV6-DN	2.00
<b>Units:</b>	i = output #, c = function identifier (letter)	GT6-DN	
<b>Range:</b>	i = 1-7, c = A-G		
<b>Default:</b>	(see programmer's reference)		
<b>Response:</b>	OUTFNC1: *OUTFNC1-A PROGRAMMABLE OUTPUT - STATUS OFF		
<b>See Also:</b>	INFNC, OUT, OUTLVL, POUTA, SMPER, TAS, TASX, TOUT		

---

A new identifier is added for the OUTFNC command. The other function identifiers can be found in the Gemini Programmer's Reference p135-136.

Identifier	Function Description
I	<b>Option Card Fault:</b> Output activates when error bit #19 is set for the option card fault. See the ERROR command for description of events. This requires ERROR.19-1 to be set, or the output will not activate.

---

**Example:**

```
OUTFNC1-i ;assign output 1 to option card fault
```

---

## TASX Transfer Extended Axis Status

<b>Type:</b>	Communication Setup	<b>Product</b>	<b>Rev</b>
<b>Syntax:</b>	<a_><!>TASX	GV6-DN	2.00
<b>Units:</b>	n/a	GT6-DN	
<b>Range:</b>	n/a		
<b>Default:</b>	n/a		
<b>Response:</b>	TASX: *TASX 0001_0000_0000_0000_0000_0000_0000_0000		
<b>See Also:</b>	DCLRLR, DRIVE, RESET, TAS, TCS, TER		

---

A new bit assignment is added for the TASX command, bit #27.

See FBS on page 62 for the event causing the condition. To clear the status bit the drive must be reset or power must be cycled.

Bit#	Function	To Clear the status Bit
27	Fieldbus Fault	RESET or cycle power

---

---

## TFBS Transfer Fieldbus Status

<b>Type:</b>	Communication Setup	<b>Product</b>	<b>Rev</b>
<b>Syntax:</b>	<!>TFBS<.i>	GV6-DN	2.00
<b>Units:</b>	i = status bit number	GT6-DN	
<b>Range:</b>	1-32		
<b>Default:</b>	n/a		
<b>Response:</b>	TFBS: TFBS.4: *1 (unit online or link ok, yes) *TFBS0001_0000_0000_0000_0000_0000_0000_0000		
<b>See Also:</b>	ER.19, [FBS]		

---

The TFBS command provides information on the 32 fieldbus status bits. The TFBS command reports a binary bit report.

Response for TFBS: \*TFBS0001\_0000\_0000\_0000\_0000\_0000\_0000\_0000

Bit#1...bit#32

For bit description, see FBS on page 62.



APPENDIX A

# Gemini DeviceNet CE Compliance

**IN THIS CHAPTER**

- [CE Compliance](#) ..... 66
- [Installation Instructions](#) ..... 66

The Gemini family of products is designed to meet the requirements of global regulatory agencies.

Gemini products, when installed in accordance with instructions in the appropriate *Hardware Installation Guide*, are compliant with CE directives. This appendix gives instructions for additional steps necessary for installing the Gemini DeviceNet Option into a CE compliant system.

## Installation Instructions

### Gemini Drive – Installation Instructions

1. Follow the standard installation and configuration instructions in your Gemini drive's *Hardware Installation Guide*.
2. Follow additional installation instructions for CE compliance in *Appendix C – Regulatory Compliance* of your drive's *Hardware Installation Guide*.

### Gemini DeviceNet Option – Installation Instructions

1. Install DeviceNet cabling and associated hardware in accordance with the DeviceNet specifications (available from [www.odva.com](http://www.odva.com)). Technical papers discussing how to implement a DeviceNet network are also available at this web site.
2. Consult your DeviceNet Master's guide for additional requirements for the Master installation.

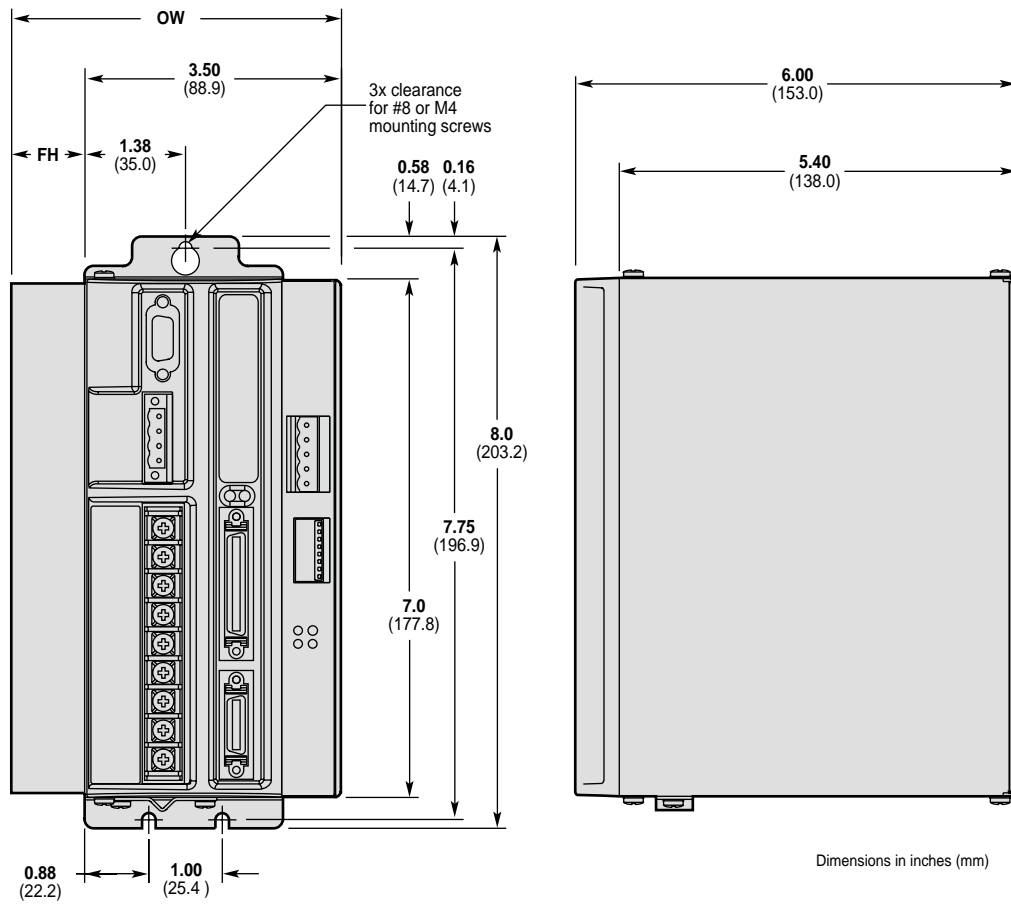
APPENDIX B

# Gemini Dimensions

**IN THIS CHAPTER**

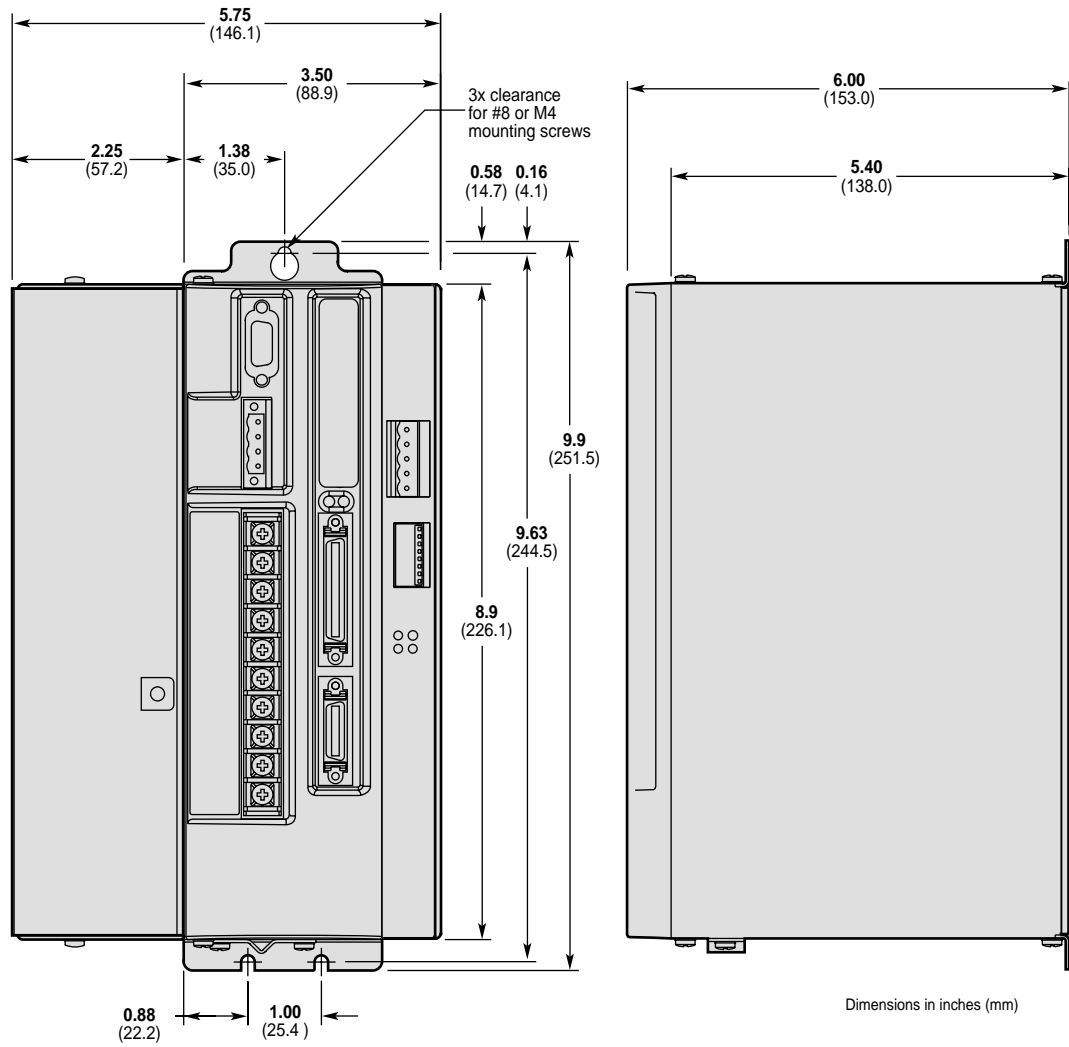
- Gemini Drive Dimensions..... 68
- Gemini Panel Layout Dimensions ..... 71

# Gemini Drive Dimensions

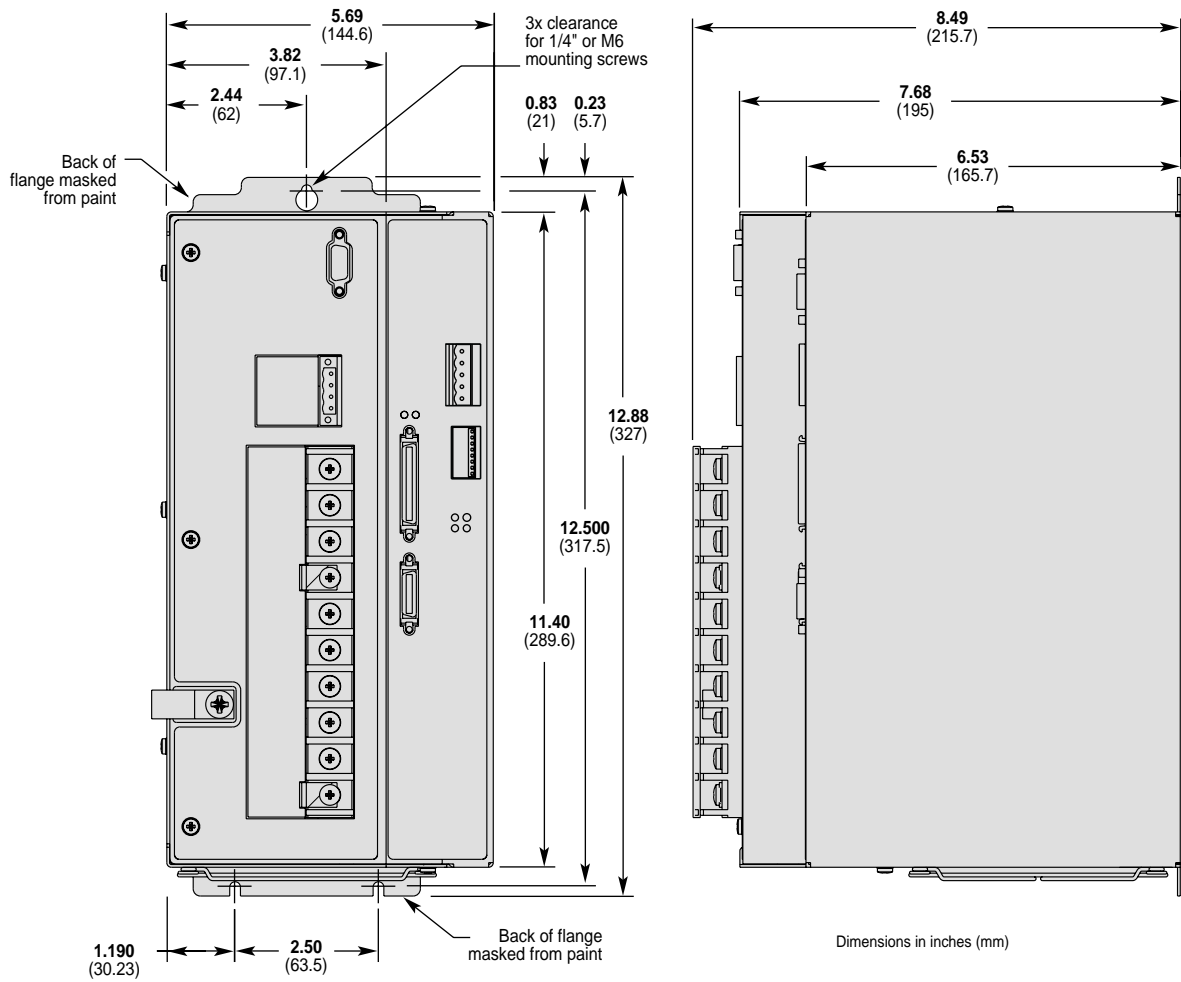


Product	OW Overall Width inches (mm)	FH Fin Height inches (mm)
GV6-L3n-DN	3.88 (98.6)	0.38 (9.5)
GV6-U3n-DN	3.88 (98.6)	0.38 (9.5)
GV6-U6n-DN	4.50 (114.3)	1.00 (25.4)
GV6-U12n-DN	4.50 (114.3)	1.00 (25.4)
GT6-L5-DN	3.88 (98.6)	0.38 (9.5)
GT6-L8-DN	4.50 (114.3)	1.00 (25.4)

## Dimensions (Shorter Enclosure)

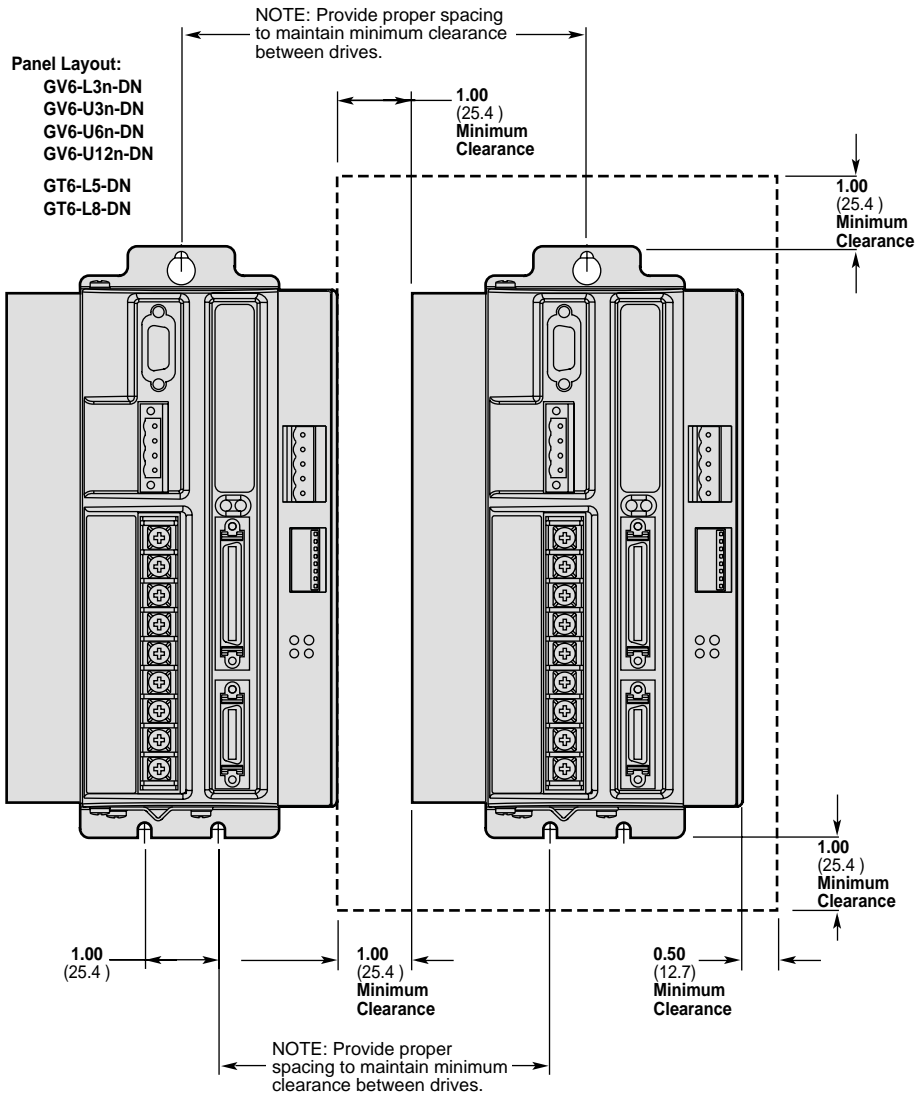


**Dimensions – GV6-H20n-DN**

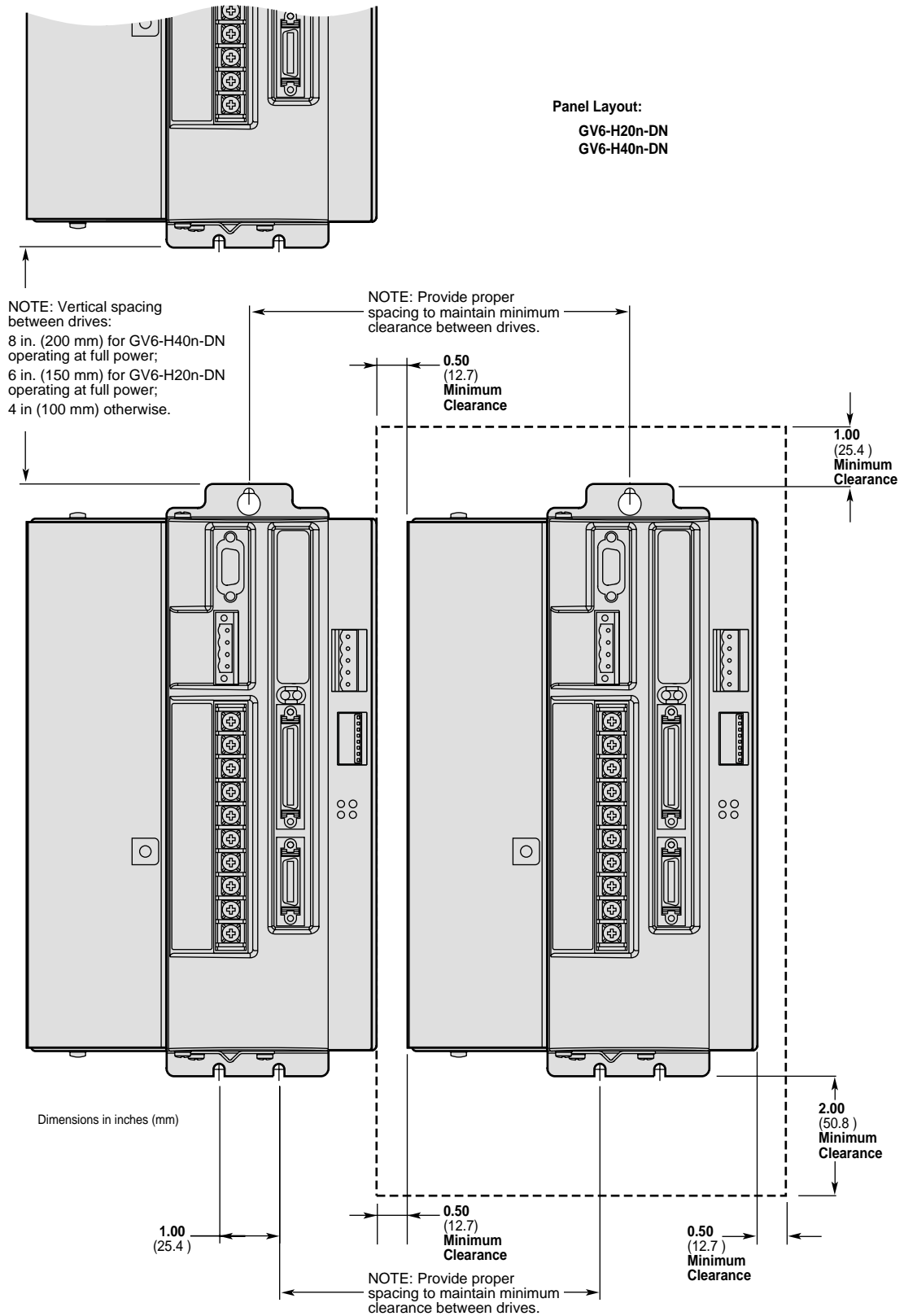


**Dimensions – GV6-H40n-DN**

# Gemini Panel Layout Dimensions



## Panel Layout Dimensions – Shorter Enclosure



### Panel Layout Dimensions – Taller Enclosure