

AVTRON
DeviceNet OPTION BOARD
(For Use with ACCel500 Frequency Converters)

© 2009 Avtron Industrial Automation, Inc.
Cleveland, Ohio

August 2, 2006
Rev. July 22, 2009

AVTRON
DeviceNet OPTION BOARD

TABLE OF CONTENTS

<u>SECTION</u>		<u>PAGE</u>
I	GENERAL INFORMATION.....	1-1
II	DeviceNet OPTION BOARD TECHNICAL DATA.....	2-1
	2-1 General.....	2-1
	2-2 DeviceNet Features and Functionality	2-2
III	DeviceNet	3-1
	3-1 Introduction	3-1
	3-2 DeviceNet Physical Layer and Media	3-1
IV	DeviceNet LAYOUT AND CONNECTIONS	4-1
	4-1 DeviceNet Option Board	4-1
	4-2 LED Indicators	4-2
	4-3 Connection of DeviceNet Drop-Line Cable	4-4
V	INSTALLATION OF ACCel500 DeviceNet OPTION BOARD	5-1
	5-1 Board Information Sticker.....	5-4
VI	COMMISSIONING.....	6-1
	6-1 DeviceNet Configuration Tool.....	6-1
	6-2 Setting DeviceNet Parameters with the Control Keypad	6-3
VII	DeviceNet INTERFACE.....	7-1
	7-1 I/O Messaging.....	7-1
	7-1.1 Input and Output Assemblies	7-2
	7-1.2 Control Supervisor Behavior	7-5
	7-2 Explicit Messaging	7-7
	7-2.1 List of Object Classes.....	7-7
	7-2.2 List of Services	7-8
	7-2.3 List of Data Types	7-8
	7-2.4 Reset Service	7-9
VIII	FAULT TRACKING.....	8-1
APPENDIX A	DeviceNet Interface Errors.....	A-1
APPENDIX B	Communication Interface Object Profiles.....	B-1

AVTRON
DeviceNet OPTION BOARD

SECTION I
GENERAL INFORMATION

Instead of sending and receiving information to and from frequency converters through I/O, you can connect them to a fieldbus.

Avtron ACCel500 frequency converters can be connected to the DeviceNet™ using a fieldbus board. The converter can then be controlled, monitored and programmed from the Host system. If you purchase your DeviceNet Option Board separately, please note that it shall be installed in slot E on the control board of the frequency converter.

WARNING

Internal components and circuit boards are at high potential when the frequency converter is connected at the power source. This voltage is extremely dangerous and may cause death or severe injury if you come into contact with it.

SECTION II

DeviceNet OPTION BOARD TECHNICAL DATA

2-1 GENERAL

TABLE 2-1. DeviceNet TECHNICAL DATA

DeviceNet Connections	Interface	Pluggable connector (5.08 mm)
	Data transfer method	CAN
	Transfer cable	2 wire twisted shielded cable with 2 wire bus power cable and drain
	Electrical isolation	500 VDC
Communications		ODVA 2.0 Compliant
	Baud rate	125, 250 and 500 kbaud
	Product code	0x02 (2)
	Product Type	0x02 (AC Drive)
Electrical	DeviceNet	Network supply voltage: 11 to 25V DC Network input current: 28 mA typical, 125 mA inrush (24V DC)
	Other	All other power derived from AVTRON inverter power supply
Environment	Ambient operating temperature	-10°C to 55°C
	Storing temperature	-40°C to 60°C
	Humidity	<95%, no condensation allowed
	Altitude	Max. 1000 m
	Vibration	0.5 G at 9 to 200 Hz
Safety		Fulfills EN50178 standard

2-2 DeviceNet FEATURES AND FUNCTIONALITY

TABLE 2-2. DeviceNet FEATURES AND FUNCTIONALITY

Network size	Up to 64 nodes	
Network length	Selectable end-to-end network distance varies with speed	
	Baud rate	Distance
	125 Kbps	500 m
	250 Kbps	250 m
	500 Kbps	100 m
Data packets	0-8 bytes	
Bus topology	Linear (trunkline/dropline); power and signal on the same network cable	
Bus addressing	Peer-to-peer with Multi-Cast (one-to-many); Multi-Master and Master/Slave special case; polled or change-of-state (exception-based)	
System features	Removal and replacement of devices from the network under power	

SECTION III

DeviceNet

3-1 INTRODUCTION

DeviceNet is an open network based on CAN that is designed to connect low cost industrial control devices (such as limit switches, photoelectric sensors, motor starters, process sensors, frequency converters, panel displays and operator interfaces) to a network and eliminate expensive hardwiring. The direct connectivity provides improved communication between devices as well as important device-level diagnostics not easily accessible or available through hardwired I/O interfaces. The DeviceNet Model is application-independent; it provides the communication services needed by various types of applications.

Many of today's lower level industrial control devices must retain their low cost/low resource characteristics even when directly connected to a network. DeviceNet takes this into consideration by defining a specific instance of the Model for communications typically seen in a Master/Slave application. This is referred to as the Predefined Master/Slave Connection Set.

DeviceNet allows the interchangeability of simple devices while making interconnectivity for more complex devices possible. In addition to reading the state of discrete devices, DeviceNet provides the capability to report temperatures, to read the load current in a motor starter, to change the deceleration rate of drives, or to count the number of packages that have passed on a conveyor in the previous hour.

3-2 DeviceNet PHYSICAL LAYER AND MEDIA

The basic trunkline-dropline topology provides separate twisted pair busses for both signal and power distribution. Thick or thin cable can be used for either trunklines or droplines. End-to-end network distance varies with data rate and cable size.

Devices can be powered directly from the bus and communicate with each other using the same cable. Nodes can be removed or inserted from the network without powering down the network.

Power taps can be added at any point in the network which makes redundant power supplies possible. The trunkline current rating is 8 amps. An opto-isolated design option allows externally powered devices (e.g. AC drives starters and solenoid valves) to share the same bus cable. Other CAN-based networks allow only a single power supply (if at all) for the entire network.

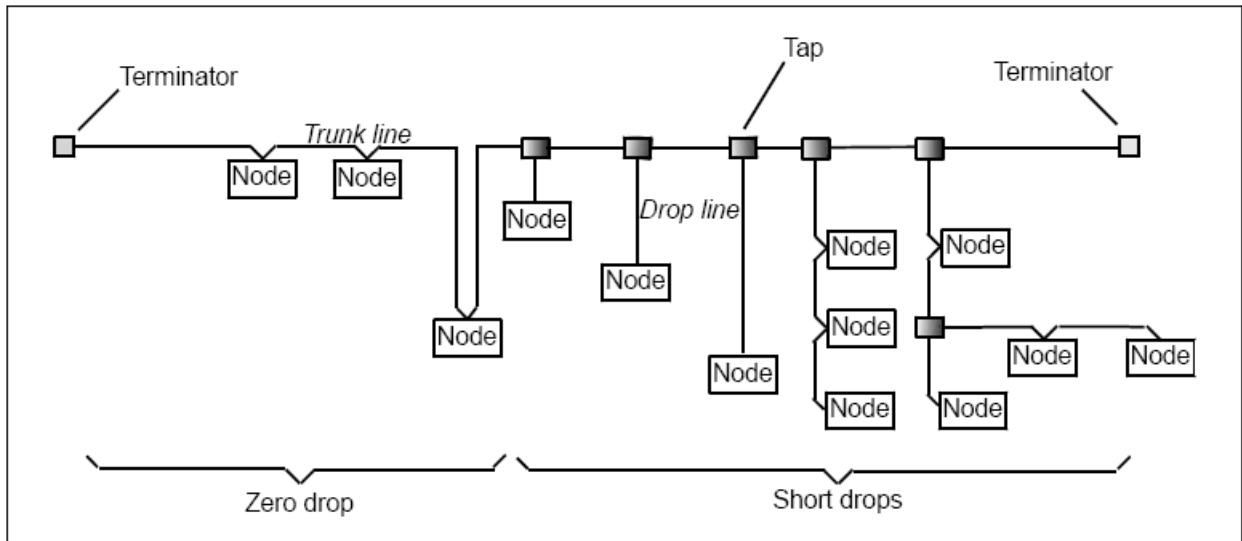


Figure 3-1. Thick or Thin Cable Can Be Used for Either Trunklines or Droplines

SECTION IV

LAYOUT AND CONNECTIONS

The DeviceNet Board is connected to the fieldbus through a 5-pin pluggable bus connector (board OPTC7). The communication with the control board of the frequency converter takes place through the standard Avtron Interface Board Connector.

4-1 DeviceNet OPTION BOARD

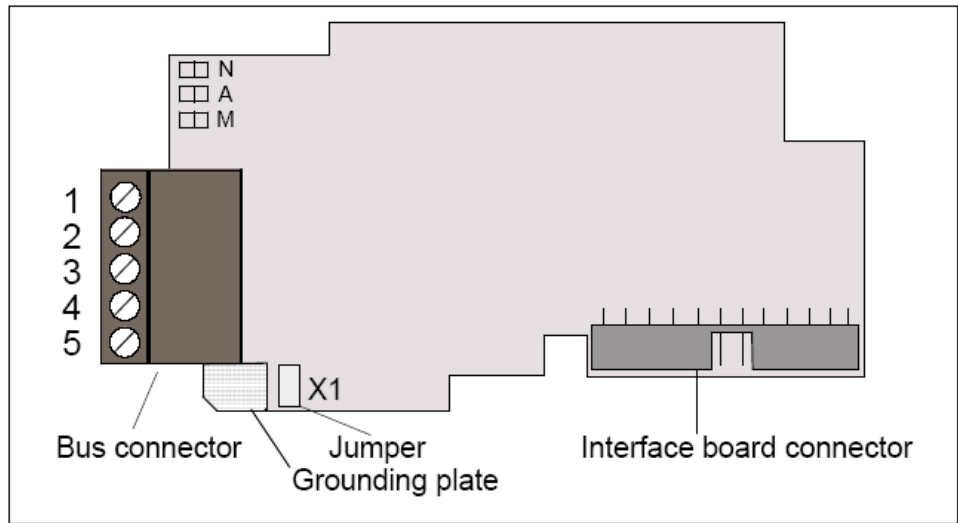


Figure 4-1. DeviceNet Option Board OPC7

TABLE 4-1. OPTC7 BUS CONNECTOR SIGNALS

Signal	Connector	Description
V-	1	Communication power supply, Ground
CAN_L	2	Communication signal, Low
Drain	3	Cable shield (bare)
CAN_H	4	Communication signal, High
V+	5	Communication power supply, +24V

4-2 LED INDICATORS

The DeviceNet Option Board includes three LED status indicators next to the connector: Network Status (N), Node Address (A) and Module Status (M). Network Status provides information on the network connection status, Node Address blinks the MAC ID of the unit while it is powered, and Module Status provides information on the DeviceNet module.

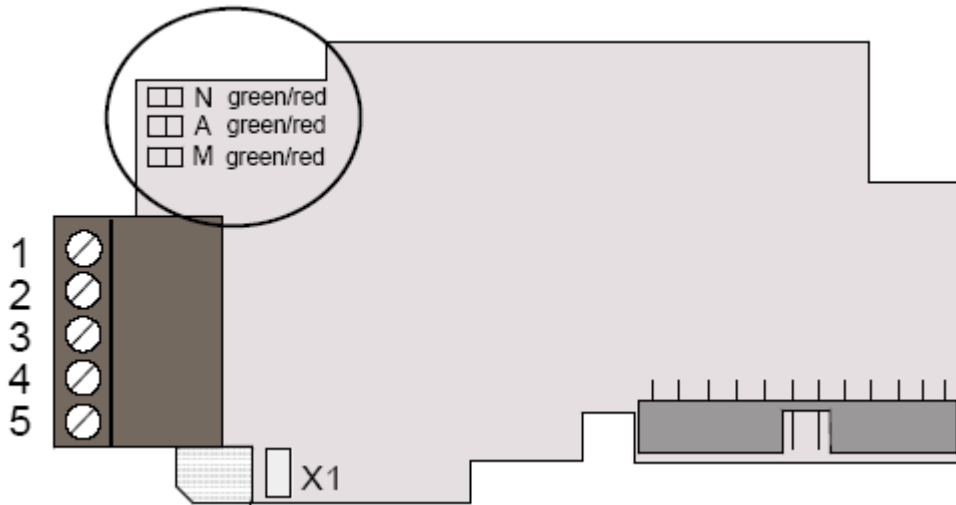


Figure 4-2. LED Indicators on the DeviceNet Board

Network status LED (N)

LED is:	Meaning:
OFF	The OPTC7 is not on-line - The device has not completed the Dup_MAC_ID test yet - If the Module status LED is off, the device is not powered
Green	The OPTC7 is on-line and allocated to a Master
Blinking Green	The OPTC7 has passed the Dup_MAC_ID test, is on-line, but is not allocated to a master
Blinking Red	One or more I/O connections are in the Timed-Out state
Red	The OPTC7 cannot communicate on the network (Duplicate MAC ID or Bus-off)

Module status LED (M)

LED is:	Meaning:
OFF	There is no power applied to the OPTC7
Green	The OPTC7 is operating normally
Blinking Green	The OPTC7 board is in Standby state or the device needs commissioning due to a missing, incomplete or incorrect configuration
Blinking Red	The option board has detected a Recoverable Fault
Red	The option board has detected an Unrecoverable Fault

Node Address LED (A)

This LED blinks the MAC ID of the unit while it is powered. The tens are displayed with red blinks and the ones with green blinks. The unit plays the tens, then the ones and finally delays about 2 seconds before repeating the sequence.


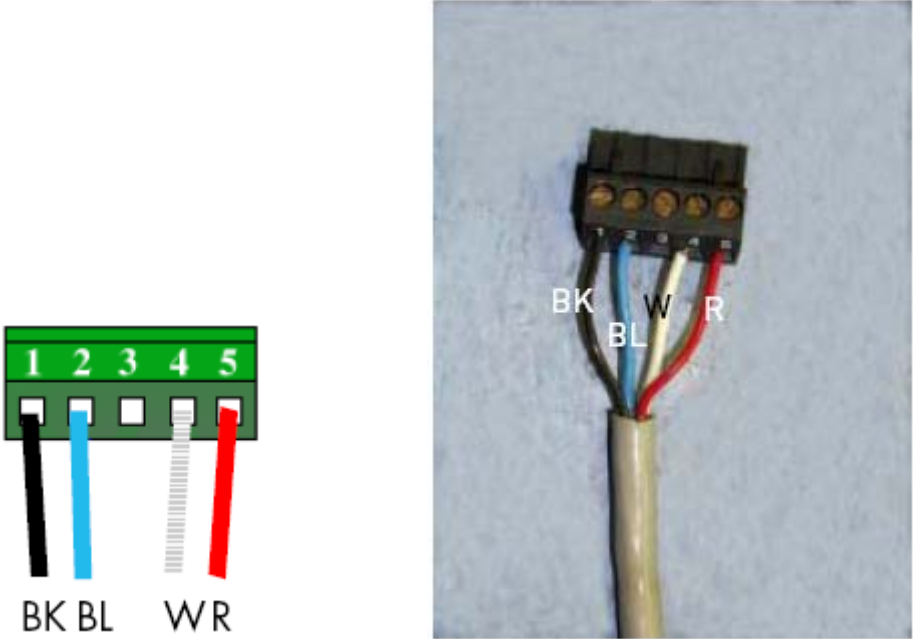
LED test

A LED test is performed at power-up. The following sequence is performed:

1. All LEDs off
2. All LEDs green (0.25 s)
3. All LEDs red (0.25 s)
4. All LEDs off
5. Start of normal operation

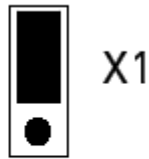
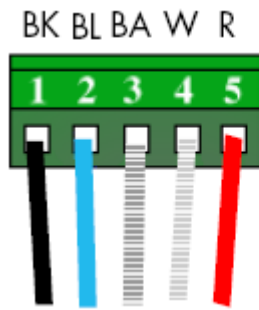
4-3 CONNECTION OF DeviceNet DROP-LINE CABLE

The following instructions lead you through the connection of the OPTC7 to the Devicenet system and show you the power-up and grounding of the board.

<p>A</p>	<p>Lead the DeviceNet drop-line cable through the right gridded hole on the bottom of the ACCel500 frequency converter. Make a sufficiently wide opening for the cable by cutting the grid as wide as necessary (see Figure E in Section V).</p> 
<p>B</p>	<p>Connect the 4 colored wires into the bus connector in the following order from the left: black, blue, NONE, white, red.</p> 

C

It is recommended that the bare wire is not connected to any of the terminals. If it however has to be connected, use the terminal 3 as shown in the picture:



Note: If the bare wire is connected, jumper X1 must be set to OFF position.

SECTION V

**INSTALLATION OF
DeviceNet OPTION BOARD**

CAUTION

Certain circuit card assemblies and their components, typically integrated circuits, may be damaged by seemingly undetectable electrostatic discharge (ESD). Care must be exercised during handling/repair of these items. Use electrostatic discharge precautionary procedures.




NOTE

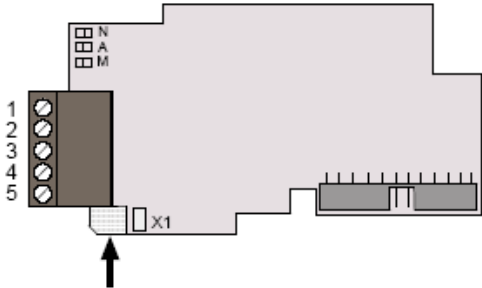

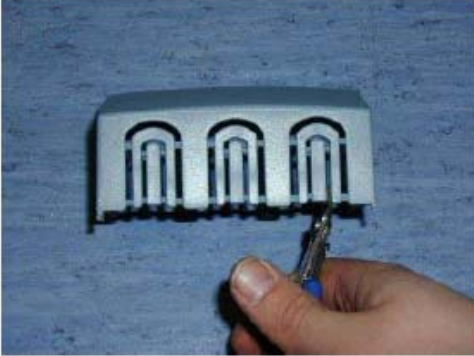

These instructions apply only to field installations. Otherwise, the board has already been installed for you at the factory.

NOTE

Make sure that the frequency converter is switched off before an option or fieldbus board is changed or added.

Before taking any commissioning actions, carefully read the safety instructions in the ACCel500 Frequency Converters manual.

A	ACCel500 frequency converter.	
B	Remove the cable cover.	
C	Open the cover of the control unit.	

<p>D</p>	<p>Install DeviceNet option board in slot E on the control board of the frequency converter. Make sure that the grounding plate (see below) fits tightly in the clamp.</p>  
<p>E</p>	<p>Make a sufficiently wide opening for your cable by cutting the grid as wide as necessary.</p> 
<p>F</p>	<p>Close the cover of the control unit and the cable cover.</p> 

5-1 BOARD INFORMATION STICKER

The DeviceNet option board package delivered by the factory includes a sticker (shown below). Please mark the board type (1), the slot into which the board is mounted (2) and the mounting date (3) on the sticker. Finally, attach the sticker on your drive.

Drive modified:

<input type="checkbox"/> Option board:	Date:.....
in slot:	A B C D E	
<input type="checkbox"/> IP54 upgrade/Collar		Date:.....
<input type="checkbox"/> EMC level modified:	H <input type="checkbox"/> T / T <input type="checkbox"/> H	Date:.....

Callout 1 points to the first field. Callout 2 points to the slot letters. Callout 3 points to the date field.

SECTION VI

COMMISSIONING

Before beginning, read Section VIII, Commissioning, in the ACCel500 Frequency Converters manual.

NOTE

You must select Fieldbus as the active control place, if you wish to control the frequency converter through fieldbus. See the ACCel500 Frequency Converters manual.

The recommended method for setting the DeviceNet parameters is with a DeviceNet Configuration Tool (see section 6-1). However, the parameters can also be set with the control keypad (see section 6-2), but then it must be ensured that the parameter settings correspond to the settings in the Master system.

6-1 DeviceNet CONFIGURATION TOOL

Before using the OPTC7 option board, you must configure the device baud rate and node address to the desired values. This can be done by using a DeviceNet Configuration tool (e.g., Netview or RSNetworx for DeviceNet). The default baud rate is 125 Kbaud and node address 63. All devices must have the same baud rate.

Since all new devices are factory set to node address 63, it is recommended that you change the address. You must also check or set the following attributes before use:

TABLE 6-1. MONITORING DATA CLASS (0XAA) – INSTANCE ATTRIBUTES

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
120 0x78	Polled Input Assembly Type	Get_Attribute_Single, Set_Attribute_Single	71 70 73 75	C7	Input assembly used by the polled connection
121 0x79	Polled Output Assembly Type	Get_Attribute_Single, Set_Attribute_Single	21 20 23 25	C7	Output assembly used by the polled connection
110 0x6E	SafeState Type	Get_Attribute_Single, Set_Attribute_Single	0 0 2	C6	<p>Selects Safe State response to errors which specify safe state operation. Currently only a loss of connection other than by de-allocation is a safe state error.</p> <p>Warning: Review the application for safe operation before specifying a value for this attribute.</p> <p>0 = DriveFault (falut and stop) 1 = No Action (hold last speed) 2 = Preset Speed/Direction</p>
111 0x6F	PresetDir	Get_Attribute_Single, Set_Attribute_Single	0 0 1	C1	<p>Sets safe state direction of rotation if the Safe State Behaviour attribute specifies “Preset Speed/Direction”</p> <p>Warning: Review the application for safe operation before specifying a value for this attribute.</p> <p>Inverter will require external stop.</p> <p>0 = Forward 1 = Reverse</p>
112 0x70	PresetRPM	Get_Attribute_Single, Set_Attribute_Single	0 0 30000	C7	<p>Sets safe state speed reference (RPM) if the Safe State Behaviour attribute specifies “Preset Speed/Direction”</p> <p>Warning: Review the application for safe operation before specifying a value for this attribute.</p> <p>Inverter will require external stop.</p>
113 0x71	PresetTq	Set_Attribute_Single, Set_Attribute_Single	0 0 10000	C7	<p>Sets safe state torque reference (0,00%) if the Safe State Behaviour attribute specifies “Preset Speed/Direction”</p> <p>Warning: Review the application for safe operation before specifying a value for this attribute.</p> <p>Inverter will require external stop.</p>

6-2 SETTING DeviceNet PARAMETERS WITH THE CONTROL KEYPAD

If the control keypad is used in setting the parameters of the Avtron DeviceNet board, certain values to appropriate parameters must be given in menu M7 (for locating the expander board menu see ACCel500 Frequency Converters manual).

NOTE

If you use the keypad for setting the parameters, make sure that the settings correspond to the settings in the Master system.

Expander Board Menu (M7)

The Expander board menu makes it possible for the user 1) to see what expander boards are connected to the control board and 2) to reach and edit the parameters associated with the expander board.

Enter the following menu level (G#) with the Right Arrow button. At this level, you can browse through slots A to E with the Up and Down buttons to see what expander boards are connected. On the lowermost line of the display, you also see the number of parameter groups associated with the board. If you press the Right Arrow button once again, you will reach the parameter group level where there are two groups: Editable parameters and Monitored values. A further press on the Right Arrow button takes you to either of these groups.

DeviceNet Parameters

To set the DeviceNet board parameters, enter the level P7.5.1.# from the Parameters group (G7.5.1). Assign desired values to all DeviceNet parameters (see Figure 6-1 and Table 6-2).

NOTE

The DeviceNet cable must be connected before any parameters can be edited. See Figure 4-1.

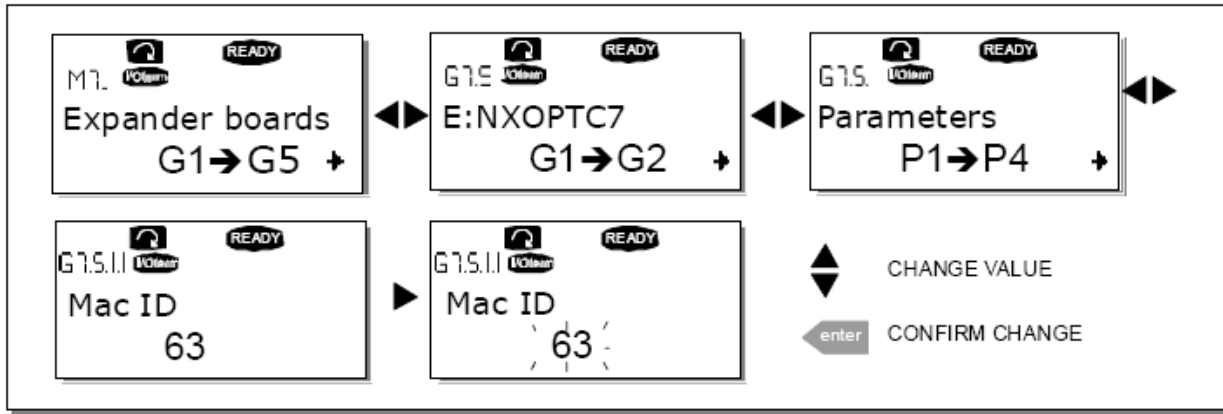


Figure 6-1. Changing the DeviceNet Board Parameter Values

TABLE 6-2. DeviceNet PARAMETERS

#	Name	Default	Range	Description
1	MAC ID	63	0...63	
2	BAUD RATE	125 kBaud	1 - 125 kBaud 2 - 250 kBaud 3 - 500 kBaud	Communication speed in baud Note! If the value of this parameter is changed, the new value will be valid after the next power-up.
3	I/O POLL TYPE	21/71	1 – 20/70 2 – 21/71 3 – 23/73 4 – 25/75	

NOTE

Every device that is connected to the bus must have an individual MAC ID. Before the values of these parameters can be changed, the drive must be in STOP mode and the communication power supply must be connected.

DeviceNet Status

To see the present status of the DeviceNet Fieldbus, enter the DeviceNet Status page from the Monitor menu (G7.5.2). See Figure 6-2 and Table 6-3 below.

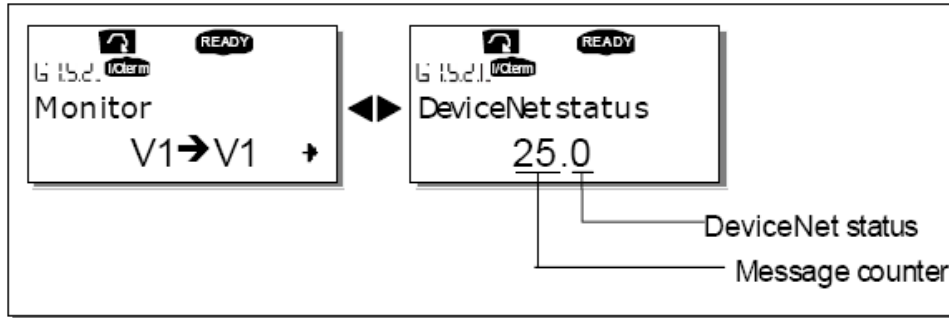


Figure 6-2. Accessing the DeviceNet Status Page

TABLE 6-3. DeviceNet STATUS INDICATORS

DeviceNet Status	
0	Non-existent or no bus power
1	Configuring state
3	Established
4	Time out

SECTION VII

DeviceNet INTERFACE

DeviceNet provides two different types of messaging. They are called I/O Messaging and Explicit Messaging.

7-1 I/O MESSAGING

I/O polling messages are for time-critical, control-oriented data. The messages are transferred between the devices all the time and they are used for continuous control of the frequency converter. They provide a dedicated, special-purpose communication path between a producing application (master) and one or more consuming applications (slaves). They are exchanged across single or multi-cast connections and typically use high priority identifiers. I/O polling messages contain no protocol in the 8-byte data field. The meaning of the message is implied by the connection ID (CAN identifier). Before messages are sent using these IDs, both the device sending and receiving them must be configured. The configuration contains the source and destination object attribute addresses for the master and the slave.

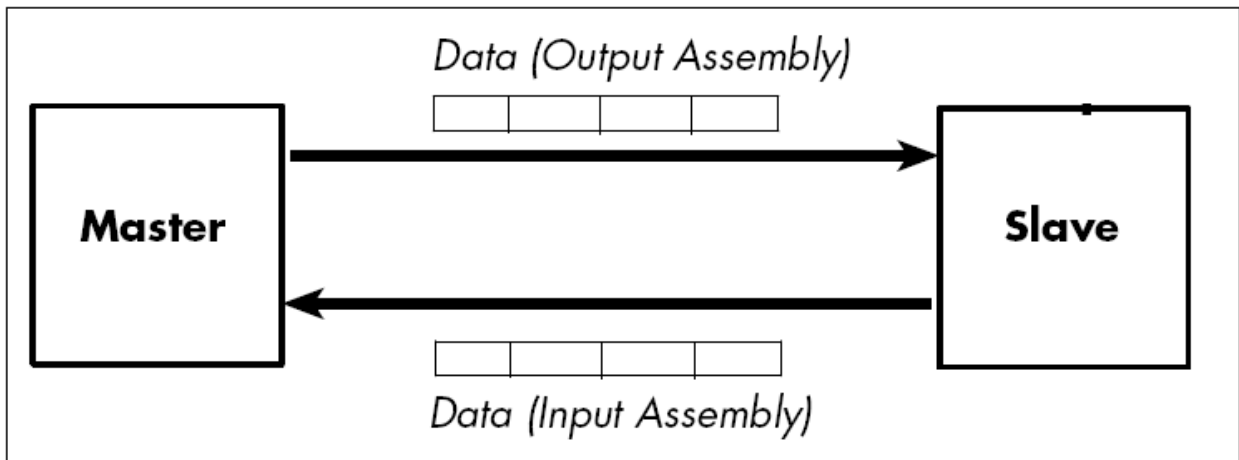


Figure 7-1. DeviceNet I/O Messaging

The contents of the data message can be chosen the following way: Monitoring Data Object (0xAA), Instance 1; Attributes 120 (Input Assembly) and 121 (Output Assembly). See the following section for the Input and Output Assemblies.

7-1.1 INPUT AND OUTPUT ASSEMBLIES

Output 20

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0						FaultReset		RunFwd
Byte 1								
Byte 2	Speed Reference (Low Byte)							
Byte 3	Speed Reference (High Byte)							

Output 21 (Default)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0		NetRef	NetCtrl			FaultReset	RunRev	RunFwd
Byte 1								
Byte 2	Speed Reference (Low Byte)							
Byte 3	Speed Reference (High Byte)							

Output 23

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0		NetRef	NetCtrl			FaultReset	RunRev	RunFwd
Byte 1								
Byte 2	Speed Reference (Low Byte)							
Byte 3	Speed Reference (High Byte)							
Byte 4	Torque Reference (Low Byte)							
Byte 5	Torque Reference (High Byte)							

Output 25

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0		NetRef	NetCtrl			FaultReset	RunRev	RunFwd
Byte 1								
Byte 2	Speed Reference (Low Byte)							
Byte 3	Speed Reference (High Byte)							
Byte 4	Process Reference (Low Byte)							
Byte 5	Process Reference (High Byte)							

Output 70

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0						Running1		Faulted
Byte 1								
Byte 2	Speed Actual (Low Byte)							
Byte 3	Speed Actual (High Byte)							

Output 71

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	AtReference	RelFromNet	CtrolFromNet	Ready	Running2	Running1	Warning	Faulted
Byte 1	DriveState							
Byte 2	Speed Actual (Low Byte)							
Byte 3	Speed Actual (High Byte)							

Output 73

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	AtReference	RelFromNet	CtrolFromNet	Ready	Running2	Running1	Warning	Faulted
Byte 1	Drive State							
Byte 2	Speed Actual (Low Byte)							
Byte 3	Speed Actual (High Byte)							
Byte 4	Torque Actual (Low Byte)							
Byte 5	Torque Actual (High Byte)							

Output 75

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	AtReference	RelFromNet	CtrolFromNet	Ready	Running2	Running1	Warning	Faulted
Byte 1	DriveState							
Byte 2	Speed Actual (Low Byte)							
Byte 3	Speed Actual (High Byte)							
	Process Actual (Low Byte)							
	Process Actual (High Byte)							

The assemblies are listed above and are grouped in pairs as follows:

- 20/70
- 21/71 (Default)
- 23/73
- 25/75 (Most versatile)

Speed reference (for 21/71 and 25/75):

The speed reference will not transfer to the drive unless the NetRef bit (Bit 6) is high. If a reference is transferred to the drive and Bit 6 goes low, the reference will stay at its last value.

This bit can be viewed at the drive from *FB Fix Cntrl Wrđ* Bit 9.

The speed reference word is sent to the drive as synchronous motor speed. The drive internally converts this into percent motor speed with two decimal points and shows up in the application as *FB Spđ Ref*.

Example: 1800 RPM motor
 Master sends 900 rpm
 FB Spđ Ref = 50.00%

The drive is not set up to use this value without reconfiguring the speed reference to this parameter's ID number.

No application scaling is required as the drive normally uses percent motor speed as its reference.

Run Commands (for 21/71 and 25/75):

The run command must first be enabled by setting the NetCtrl bit (Bit 5). The drive must also be in the ready mode before the run bits will show up.

This bit can be viewed at the drive from *FB Fix Cntrl Wrd* Bit 8.

Two bits are used to supply the drive with run forward and reverse

Setting RunFwd (Bit 0) will translate into the drive as *FB Run*.
 Setting RunRev (Bit 1) will set both the drives *FB Run* and *RB Dir*.

The drive is not set up to use these bits without reconfiguring the drive run inputs to these parameter ID numbers.

Fault Reset (for 21/71 and 25/75):

The master can set a fault reset command bit setting the FaultReset (Bit 2). This gets mapped to the drive as bit *FB flt Res*. Configure *Fault Reset* to *FB flt Res* ID number to implement it.

Process Reference (For 25/75):

The process reference does not require the NetRef bit to be high to be transmit to the drive. It also does not get scaled and comes through as the same number transmitted. This value can be monitored or used by the value *A FB AINI*.

Process Actual (For 25/75):

The process Actual comes from the drives FB AOUT1 parameter and can be configured to any drive ID number to transmit the data.

Devicenet Faults:

Upon a device net fault, the drive will turn off the *FB Run* and *FB Dir* bits and stop the drive. Also, if enabled, the drive will fault out on *FB Comm fault*. See the *FBCommFaultResp* parameter to set up to fault, warn, or no action for the drive.

W A R N I N G

If communications returns and the drive fault is cleared, the drive will re-enable the run bits and start to run unless the master turned the bits off.

Conclusion:

The Devicenet Assemblies provide a quick way to get a motor spinning but is very limited on the amount of information sent. The board takes care of the link's health and if it's active.

7.1.2 CONTROL SUPERVISOR BEHAVIOR

The State Transition Diagram provides a graphical description of the states and corresponding state transitions for the control supervisor.

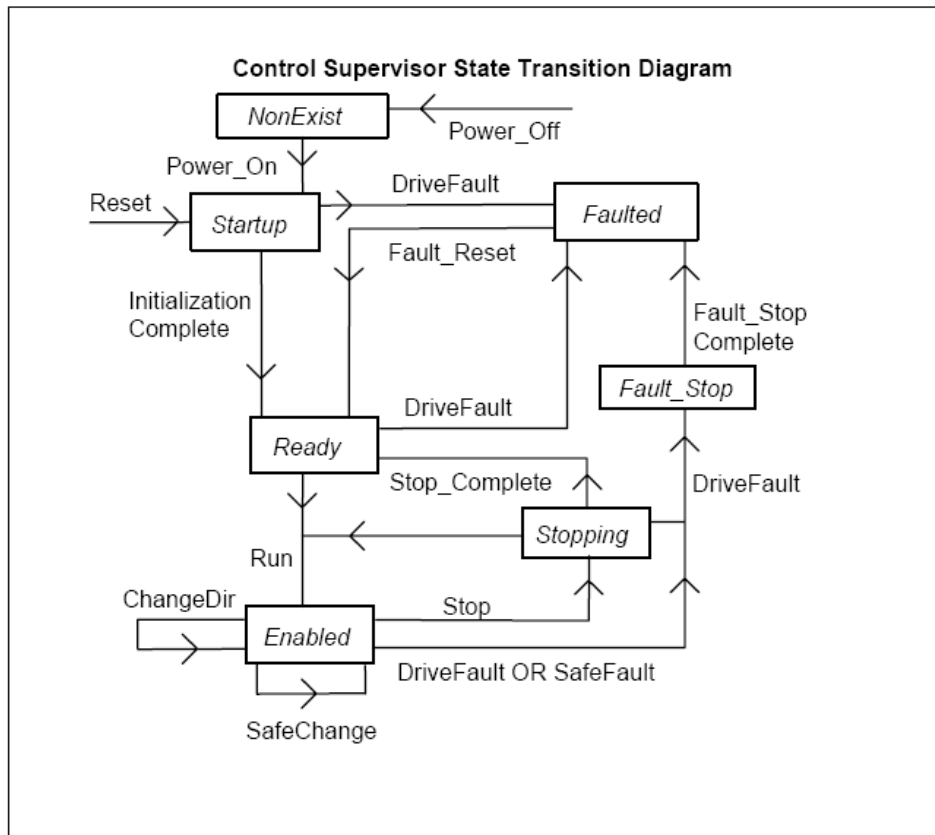


Figure 7-2. Control Supervisor State Transition Diagram

TABLE 7-1. EXPLANATION OF CONTROL SUPERVISOR STATE TRANSITION DIAGRAM

Old State	Input Conditions						Event	New State	Results Action
	CtrlFromNet	Run1Var	Run2Var	IdleMode	FwdMode	RevMode			
x	x	x	x	x	x	x	Power_Off	NonExist	
x (except NonExist)	x	x	x	x	x	x	Reset	Startup	Faulted := 0; Ready := 0 FwdMode := 0; RevMode := 0 Run1Var := 0; Run2Var := 0
NonExist	x	x	x	x	x	x	Power_On	Startup	Faulted := 0; Ready := 0 FwdMode := 0; RevMode := 0 Run1Var := 0; Run2Var := 0
Startup	x	x	x	x	x	x	DriveFault	Faulted	Faulted := 1; FaultCode := x
Startup	x	x	x	x	x	x	Initialization Complete	Ready	Ready := 1
Ready	x	x	x	x	x	x	DriveFault	Faulted	Faulted := 1; FaultCode := x; Ready := 0
Ready	1	1	0	0	x	x	Run (Fwd)	Enabled	FwdMode := 1 (Start Forward)
Ready	1	0	1	0	x	x	Run (Rev)	Enabled	RevMode := 1 (Start Reverse)
Enabled	x	x	x	x	x	x	DriveFault	Fault_Stop	Faulted := 1; FaultCode := x (Initiate Faulted Stop) FwdMode := 0; RevMode := 0; Ready := 0
Enabled	1	0	0	x	x	x	Stop	Stopping	(Initiate Stop)
Enabled	1	1	0	0	0	1	ChangeDir (Fwd)	Enabled	FwdMode := 1; RevMode := 0 (Change to Forward)
Enabled	1	0	1	0	1	0	ChangeDir (Rev)	Enabled	FwdMode := 0; RevMode := 1 (Change to Reverse)
Enabled	1	x	x	x	x	x	SafeFault	Fault_Stop	Faulted := 1; FaultCode := x (Initiate Faulted Stop) FwdMode := 0; RevMode := 0; Ready := 0 Run1Var := 0; Run2Var := 0
Enabled	1	x	x	x	x	x	SafeChange	Enabled	FwdMode := Run1Var := NOT PresetDir RevMode := Run2Var := PresetDir SpeedRef := PresetSpeed TorqueRef := PresetTorque
Stopping	x	x	x	x	x	x	DriveFault	Fault_Stop	Faulted := 1; FaultCode := x (Initiate Faulted Stop) Ready := 0
Stopping	1	1	0	0	x	x	Run (Fwd)	Enabled	FwdMode := 1 (Start Forward)
Stopping	1	0	1	0	x	x	Run (Rev)	Enabled	RevMode := 1 (Start Reverse)
Stopping	x	0	0	x	x	x	Stop_Complete	Ready	
Fault_Stop	x	x	x	x	x	x	Fault_Stop Complete	Faulted	
Faulted	x	x	x	x	x	x	Fault_Reset	Ready	Faulted := 0; Ready := 1

Start Forward, Start Reverse, Change to Forward, Change to Reverse, and Stop (Not Faulted Stop) are static outputs of the Control Supervisor state machine. They are commands to the drive when CtrlFromNet=1. When CtrlFromNet=0, control commands are from another source.

Other logic equations:

```

RefFromNet = (NetRef) AND (EnableFieldbus)
IF (RefFromNet)
{
    (Write reference frequency or torque to the drive whenever SpeedRef or TorqueRef are
    written)
}

```

When performing changes to achieve programmed Safe State:

1. Run/Stop/Direction can be changed because CtrlFromNet must equal 1 when in the Enabled State.
2. Reference in the drive can be changed to PresetSpeed or PresetTorque only if (RefFromNet = 1).

7-2 EXPLICIT MESSAGING

Explicit Messaging is used in commissioning and assigning parameter values of the DeviceNet board. Explicit messages provide multipurpose, point-to-point communication paths between two devices. They provide the typical request/response-oriented network communication used to perform node configuration and problem diagnosis. Explicit messages typically use low priority identifiers and contain the specific meaning of the message right in the data field. This includes the service to be performed and the specific object attribute address.

7-2.1 LIST OF OBJECT CLASSES

The Communication Interface supports the following object classes:

Class	Object
0x01	Identity
0x02	Message Router
0x03	DeviceNet
0x04	Assembly
0x05	DeviceNet Connection
0x28	Motor Data
0x29	Control Supervisor
0x2A	AC/DC Drive
0xA0	Parameter
0xAA	Monitoring Data

7-2.2 LIST OF SERVICES

The Services supported by these object classes are shown below.

Service Code (in hex)	Service Name	Identity		Message Router		DeviceNet		Assembly		Connection		Motor Data		Control Supervisor		AC/DC Drive		Other Objects	
		Class	Inst	Class	Inst	Class	Inst	Class	Inst	Class	Inst	Class	Inst	Class	Inst	Class	Inst	Class	Inst
05	Reset (Type 0, 1)		Y								Y								
0E	Get_Attribute_Single	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
10	Set_Attribute_Single		Y		Y		Y		Y		Y		Y		Y		Y		Y
14	Error Response	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
18	Get Member	Y		Y		Y		Y		Y		Y		Y		Y		Y	
4B	Allocate_Master/Slave_Connection_Set						Y												
4C	Release_Master/Slave_Connection_Set						Y												
Vendor Specific Services																			
47	CH Get Member	Y		Y		Y		Y		Y		Y		Y		Y		Y	

See Appendix B for the Interface Object profiles.

7-2.3 LIST OF DATA TYPES

The attribute list that follows includes information on the Data Type of each attribute. The following tables explain the Data, Structure, and Array Type codes used in the Data Type column.

Elementary Data Types

Data Type Name	Data Type Code (in hex)	Data Type Description
BOOL	C1	Logical Boolean with values TRUE and FALSE
SINT	C2	Signed 8-bit integer value
INT	C3	Signed 16-bit integer value
USINT	C6	Unsigned 8-bit integer value
UINT	C7	Unsigned 16-bit integer value
UDINT	C8	Unsigned 32-bit integer value
BYTE	D1	bit string - 8-bits
WORD	D2	bit string - 16-bits
SHORT_STRING	DA	character sting (1 byte per character, 1 byte length indicator)

Constructed Data Types

Type code	Description
A1	Abbreviated array type encoding
A2	Formal structure type encoding

7-2.4 RESET SERVICE

The following table lists the different types of resets supported by the Identity Object. Resetting the OPTC7 interface to its out-of-box configuration will set ALL attributes to their default values and change the response of the drive to a loss of communications with the OPTC7. The device will have to be re-configured for your application before resuming normal operation.

Resetting the ACCel500 inverter to its out-of-box configuration will set ALL inverter parameters to their default values. Before restarting the inverter, you must verify that it is properly configured for your application.

Value:	Type of Reset:
0	Emulate as closely as possible the cycling of power to the OPTC7 DeviceNet Interface. This value is the default if this parameter is omitted. The ACCel500 drive shall be stopped if it is running.
1	Return the OPTC7 DeviceNet Interface AND the ACCel500 drive as closely as possible to the out-of-box (factory default) configuration, then emulate cycling of power as closely as possible. The ACCel500 drive shall be stopped if it is running.

SECTION VIII

FAULT TRACKING

The table below presents the faults related to the DeviceNet option board. For more information, see the ACCel500 Frequency Converters manual.

The DeviceNet option board status LEDs have been described in more detail in section 4-2.

TABLE 8-1. DeviceNet OPTION BOARD FAULTS

Fault code	Fault	Possible cause	Correcting measures
37	Device change	Option board changed.	Reset
38	Device added	Option board added.	Reset
39	Device removed	Option board removed.	Reset
40	Device unknown	Unknown option board.	
53	Fieldbus fault	The data connection between the DeviceNet Master and the DeviceNet option board is broken.	Check the installation. If installation is correct contact the nearest Avtron distributor.
54	Slot fault	Defective option board or slot.	Check the board and slot. Contact the nearest Avtron distributor.

You can define with parameters how the frequency converter shall react to certain faults:

TABLE 8-2. FREQUENCY CONVERTER RESPONSES TO FAULTS

Code	Parameter	Min	Max	Unit	Step	Default	ID	Note
P2.7.22	Response to fieldbus fault	0	3		1	0	733	0 = No response 1 = Warning 2 = Fault,stop acc. to 2.4.7 3 = Fault,stop by coasting
P2.7.23	Response to slot fault	0	3		1	0	734	0 = No response 1=Warning 2 = Fault,stop acc. to 2.4.7 3 = Fault,stop by coasting

APPENDIX A

DeviceNet INTERFACE ERRORS

The OPTC7 DeviceNet interface records the following events in the Event List FIFO:

AVTRON Event Name	Event Code	Event Description
No event	0x00	Default value in EventList entries.
Drive Communication Error	0x01	Inverter interface communication error with the drive.
I/O Connection Timeout – Fault_Stop	0x02	Control Supervisor transitions to Fault_Stop.
I/O Connection Timeout – No Action	0x03	Control Supervisor remains in Enabled State. Hold last speed
I/O Connection Timeout – Preset Direction and Speed	0x04	Control Supervisor remains in Enabled State. Run at preset direction and speed.
Explicit Connection Timeout – Fault_Stop	0x05	Control Supervisor transitions to Fault_Stop.
Explicit Connection Timeout – No Action	0x06	Control Supervisor remains in Enabled State. Hold last speed
Explicit Connection Timeout – Preset Direction and Speed	0x07	Control Supervisor remains in Enabled State. Run at preset direction and speed.
Low DeviceNet Voltage	0x08	Connection timeout may occur next.
Bus Off	0x09	Connection timeout may occur next.
CAN Overrun	0x0C	Connection timeout may occur next.
Configuration Consistency Value (CRC) mismatch	0x0E	The device's configuration is incorrect or incomplete. Major Recoverable Fault. An Identity Reset type 1 is needed for recovery.
Microprocessor watchdog timeout	0x0F	The device detected a serious problem with itself. Major Unrecoverable Fault.
Received explicit message is too big	0x10	Message is ignored.
Received IO message is too big	0x11	Message is ignored.
Parameter Range Error	0x12	An out-of-range parameter value exists in the drive.
I/O Connection Released – Fault_Stop	0x14	Control Supervisor transitions to Fault_Stop.
I/O Connection Released – No Action	0x15	Control Supervisor remains in Enabled State. Hold last speed
I/O Connection Released – Preset Direction and Speed	0x16	Control Supervisor remains in Enabled State. Run at preset direction and speed.
Receive_Idle – Fault_Stop	0x17	Control Supervisor transitions to Fault_Stop.
Receive_Idle – No Action	0x18	Control Supervisor remains in Enabled State. Hold last speed

AVTRON Event Name	Event Code	Event Description
Receive_Idle – Preset Direction and Speed	0x19	Control Supervisor remains in Enabled State. Run at preset direction and speed.
Explicit Connection Released – Fault_Stop	0x1A	Control Supervisor transitions to Fault_Stop.
Explicit Connection Released – No Action	0x1B	Control Supervisor remains in Enabled State. Hold last speed
Explicit Connection Released – Preset Direction and Speed	0x1C	Control Supervisor remains in Enabled State. Run at preset direction and speed.
Connection unable to read message	0x21	Error detected by connection object code
Connection unable to send message	0x22	Error detected by connection object code
Consumer unable to read message	0x23	Error detected by connection object code
Producer unable to send message	0x24	Error detected by connection object code
Producer unable to send buffer	0x25	Error detected by connection object code
Producer unable to send acknowledgment	0x26	Error detected by connection object code
Unexpected notification that message was sent	0x27	Error detected by connection object code
Explicit reply is too big	0x31	Error detected by connection object code
First fragment of an IO message is too big	0x34	Error detected by connection object code
Reassembled IO message is too big	0x35	Error detected by connection object code
IO message is too big for producer	0x36	Error detected by connection object code
IO/Explicit message is too big for producer	0x37	Error detected by connection object code

APPENDIX B

COMMUNICATION INTERFACE OBJECT PROFILES

In the following list, attributes shown in bold are stored in the non-volatile memory of the OPTC7 or ACCel500 drive and maintain their values after a power loss. All other settable attributes will power up at their default values.

Identity Object

Class Code 0x01

Identity Class (1) – Class attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 1	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"Identity" "Identity" "Identity"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.

Identity Class (1) – Class attributes (0) – continued

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance IDs supported by this class

Identity Class (1) - Instance Attributes (1)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Vendor Id	Get_Attribute_Single	443 443 443	C7	Identification of each vendor by number
2 0x02	Device Type	Get_Attribute_Single	2 2 2	C7	Indication of the general type of product
3 0x03	Product Code	Get_Attribute_Single	1 1 1	C7	This is a code assigned by the vendor to describe the device. Product code determined by interrogating the connected drive.
4 0x04	Revision	Get_Attribute_Single	{1,11} {1,11} {1,11}	A2 02 C6 C6	Revision of the item the Identity Object represents
5 0x05	Status	Get_Attribute_Single	N/A 0 255	D2	Summary Status of the Device. Defined in ODVA DeviceNet spec. Bit 5 = User fault Bit 6 = Node fault Bit 7 = System fault
6 0x06	Serial Number	Get_Attribute_Single	N/A 0x30940000 0x37FFFFFF	C8	Serial Number of the device
7 0x07	Product Name	Get_Attribute_Single	"OPTC7" "OPTC7" "OPTC7"	DA	Human readable identification
8 0x08	State	Get_Attribute_Single	N/A 0 5	C6	Present state of the device as represented by the state transition diagram. 0 = Nonexistent 1 = Device Self Testing 2 = Standby 3 = Operational 4 = Major Recoverable Fault 5 = Major Unrecoverable Fault
9 0x09	Configuration Consistency Value	Get_Attribute_Single	N/A 0 65535	C7	Contents identify configuration of device
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Message Router Object

Class Code 0x02

Message Router Class (2) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 1	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"Message Router" "Message Router" "Message Router"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service. F97
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance IDs supported by this class

Message Router Class (2) - Instance Attributes (1)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Object List	Get_Attribute_Single	N/A N/A N/A	A2 04 C7 A1 01 C7	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters.

DeviceNet Object
Class Code 0x03

DeviceNet Class (3) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	2 2 2	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"DeviceNet" "DeviceNet" "DeviceNet"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance IDs supported by this class

DeviceNet Class (3) - Instance Attributes (1)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	MAC ID	Get_Attribute_Single, Set_Attribute_Single	63 0 63	C6	Node Address.
2 0x02	Baud Rate	Get_Attribute_Single, Set_Attribute_Single	0 0 2	C6	The baud rate of the device. 00 = 125K 01 = 250K 02 = 500K
3 0x03	BOI [Bus Off Interrupt]	Get_Attribute_Single, Set_Attribute_Single	1 0 1	C1	
4 0x04	Bus-off Counter	Get_Attribute_Single, Set_Attribute_Single	0 0 255	C6	
5 0x05	Allocation Information	Get_Attribute_Single	N/A N/A N/A	A2 02 D1 C6	Allocation Choice Master's Mac ID Struct of: BYTE: Allocation Choice byte Bit 0 = explicit messaging Bit 1 = Polled I/O USINT:Master's Mac ID 0-63 valid 255 = unallocated
100 0x64	Bus-off Separation	Get_Attribute_Single, Set_Attribute_Single	50 0 255	C6	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Assembly Object

Class Code 0x04

Assembly Class (4) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	2 2 2	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	75 75 75	C7	
3 0x03	Number of Instances	Get_Attribute_Single	8 8 8	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB	Object Name	Get_Attribute_Single	"Assembly" "Assembly" "Assembly"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A101 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance IDs supported by this class

Assembly Class (4) – Basic Control (20)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Data	Get_Attribute_Single	N/A N/A N/A	A2 05 A1 01 C1 D1 C3	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Assembly Class (4) – Torque Control (23)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Data	Get_Attribute_Single	N/A N/A N/A	A2 05 A1 01 C1 D1 C3	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Assembly Class (4) – Extended Process Control (25)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Data	Get_Attribute_Single	N/A N/A N/A	A2 05 A1 01 C1 D1 C3	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Assembly Class (4) - Basic Status (70)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Data	Get_Attribute_Single	N/A N/A N/A	A2 05 A1 01 C1 D1 C3	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Assembly Class (4) - Speed Status (71)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Data	Get_Attribute_Single	N/A N/A N/A	A2 05 A1 01 C1 D1 C3	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Assembly Class (4) - Torque Status (73)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Data	Get_Attribute_Single	N/A N/A N/A	A2 05 A1 01 C1 D1 C3	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Assembly Class (4) - Extended Process Control (75)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Data	Get_Attribute_Single	N/A N/A N/A	A2 05 A1 01 C1 D1 C3	
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

DeviceNet Connection Object

Class Code 0x05

DeviceNet Connection Class (5) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 1	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	2 2 2	C7	
3 0x03	Number of Instances	Get_Attribute_Single	2 2 2	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"DeviceNet Connection" "DeviceNet Connection" "DeviceNet Connection"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance IDs supported by this class

DeviceNet Connection Class (5) - Explicit Connection Instance (1)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	State	Get_Attribute_Single	0 0 5	C6	State of the object
2 0x02	Instance Type	Get_Attribute_Single	0 0 0	C6	Indicates either I/O or messaging connection 0=Explicit
3 0x03	Transport Class Trigger	Get_Attribute_Single	0x83 0x83 0x83	D1	Defines Behavior of the connection
4 0x04	Produced Connection Id	Get_Attribute_Single	N/A 0x403 0x5FB	C7	Placed in CAN Identifier Field when the Connection Transmits
5 0x05	Consumed Connection Id	Get_Attribute_Single	N/A 0x404 0x5FC	C7	CAN Identifier Field value that denotes message to be received
6 0x06	Initial Comm Characteristics	Get_Attribute_Single	0x21 0x21 0x21	D1	Defines the Message Group(s) across which productions and consumptions associated with this
7 0x07	Produced Connection Size	Get_Attribute_Single	41 41 41	C7	Maximum number of bytes transmitted across this Connection
8 0x08	Consumed Connection Size	Get_Attribute_Single	41 41 41	C7	Maximum number of bytes received across this Connection
9 0x09	Expected Packet Rate	Get_Attribute_Single, Set_Attribute_Single	2500 0 65535	C7	Defines timing associated with this Connection
12 0x0C	Watchdog Timeout Action	Get_Attribute_Single, Set_Attribute_Single	1 1 3	C6	Defines how to handle Inactivity/Watchdog timeouts. 1-Auto Delete 3-Deferred Delete An attempt to set this attribute to the value 2 will result in an Invalid Attribute Value error.
13 0x0D	Produced Connection Path Length	Get_Attribute_Single	0 0 0	C7	Number of bytes in the produced_connection_path length attribute
14 0x0E	Produced Connection Path	Get_Attribute_Single	N/A N/A N/A	A1 01 C6	Application Obj. producing data on this connection
15 0x0F	Consumed Connection Path Length	Get_Attribute_Single	0 0 0	C7	Number of bytes in the consumed_connection_path length attribute
16 0x10	Consumed Connection Path	Get_Attribute_Single	N/A N/A N/A	A1 01 C7	Specifies the Application Object(s) that are to receive the data consumed by this Connection Object
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	State	Get_Attribute_Single	0 0 4	C6	State of the object
2 0x02	Instance Type	Get_Attribute_Single	1 1 1	C6	Indicates either I/O or messaging connection 1 = I/O
3 0x03	Transport Class Trigger	Get_Attribute_Single	0x83 0x83 0x83	D1	Defines Behavior of the connection
4 0x04	Produced Connection Id	Get_Attribute_Single	N/A 0x3C0 0x3FF	C7	Placed in CAN Identifier Field when the Connection Transmits
5 0x05	Consumed Connection Id	Get_Attribute_Single	N/A 0x405 0x5FD	C7	CAN Identifier Field value that denotes message to be received
6 0x06	Initial Comm Characteristics	Get_Attribute_Single	1 1 1	D1	Defines the Message Group(s) across which productions and consumptions associated with this
7 0x07	Produced Connection Size	Get_Attribute_Single	4 0 6	C7	Maximum number of bytes transmitted across this Connection
8 0x08	Consumed Connection Size	Get_Attribute_Single	4 0 6	C7	Maximum number of bytes received across this Connection
9 0x09	Expected Packet Rate	Get_Attribute_Single, Set_Attribute_Single	0 0 65535	C7	Defines timing associated with this Connection
12 0x0C	Watchdog Timeout Action	Get_Attribute_Single, Set_Attribute_Single	0 0 2	C6	Defines how to handle Inactivity/Watchdog timeouts. 1-Auto Delete 3-Deferred Delete An attempt to set this attribute to the value 2 will result in an Invalid Attribute Value error.
13 0x0D	Produced Connection Path Length	Get_Attribute_Single	6 6 6	C7	Number of bytes in the produced_connection_path length attribute
14 0x0E	Produced Connection Path	Get_Attribute_Single	N/A N/A N/A	A1 01 C6	Application Obj. producing data on this connection
15 0x0F	Consumed Connection Path Length	Get_Attribute_Single	6 6 6	C7	Number of bytes in the consumed_connection_path length attribute
16 0x10	Consumed Connection Path	Get_Attribute_Single	N/A N/A N/A	A1 01 C7	Specifies the Application Object(s) that are to receive the data consumed by this Connection Object
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Motor Data Object

Class Code 0x28

Motor Data Object Class (40) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 1	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"Motor Data" "Motor Data" "Motor Data"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance IDs supported by this class

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Motor Type	Get_Attribute_Single	7 7 7	C6	Type of Motor Represented by This Instance 0 – Non-standard motor 1 - PM DC Motor 2 - FC DC Motor 3 - PM Synchronous Motor 4 - FC Synchronous Motor 5 - Switched Reluctance Motor 6 - Wound Rotor Induction Motor 7 - Squirrel Cage Induction Motor 8 - Stepper Motor 9 - Sinusoidal PM BL Motor 10 - Trapezoidal PM BL Motor
6 0x06	RatedCurrent[100mA]	Get_Attribute_Single	70 0 16000	C7	Rated Stator Current Units: [100mA]
7 0x07	RatedVoltage[V]	Get_Attribute_Single	230 180 690	C7	Rated Base Voltage Units: [V]
9 0x09	RatedFrequency[Hz]	Get_Attribute_Single	50 30 500	C7	Rated Electrical Frequency Units: [Hz]
12 0x0C	PoleCount[pair*2]	Get_Attribute_Single	4 2 16	C7	Number of poles in the motor
15 0x0F	BaseSpeed[RPM]	Get_Attribute_Single	1440 1 30000	C7	Nominal speed at rated frequency from nameplate Units: [RPM]
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Control Supervisor Object

Class Code 0x29

Control Supervisor Object Class (41) – Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 1	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	“Control Supervisor” “Control Supervisor” “Control Supervisor”	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array’s elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array’s elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance Ids supported by this class

Control Supervisor Object Class (41) - Instance Attributes (1)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	Run1	Get_Attribute_Single, Set_Attribute_Single	0 0 1	C1	Run Forward Request
4 0x04	Run2	Get_Attribute_Single, Set_Attribute_Single	0 0 1	C1	Run Reverse Request
5 0x05	NetCtrl	Get_Attribute_Single, Set_Attribute_Single	0 0 1	C1	Requests Run/Stop control to be local or from network
6 0x06	State	Get_Attribute_Single	N/A 0 7	C6	State of Control Supervisor Instance 1 = Startup 2 = Not_Ready 3 = Ready 4 = Enabled 5 = Stopping 6 = Fault_Stop 7 = Faulted
7 0x07	Running1	Get_Attribute_Single	N/A, 0 1	C1	Running Forward Status 0 = Other state 1 = Running Forward
8 0x08	Running2	Get_Attribute_Single	N/A, 0 1	C1	Running Reverse Status 0 = Other state 1 = Running reverse
9 0x09	Ready	Get_Attribute_Single	N/A, 0 1	C1	Ready to Accept a Run Event 0 = Other state 1 = Ready to accept a Run Event
10 0x0A	Faulted	Get_Attribute_Single	N/A, 0 1	C1	Fault Occurred 0 = No faults present 1 = Fault occurred (latched)
11 0x0B	Warning	Get_Attribute_Single	N/A, 0 1	C1	Warning Present 0 = No warnings present 1 = Warning present (not latched)
12 0x0C	FaultRst	Get_Attribute_Single, Set_Attribute_Single	0 0 1	C1	Fault Reset Request 0 = No action 0 -> 1 = Fault reset request 1 = No action
13 0x0D	FaultCode	Get_Attribute_Single	N/A, 0, 65535	C7	If in Faulted state, FaultCode indicates the fault that caused the transition to Faulted state. If not in Faulted state, FaultCode indicates the fault that caused the last
14 0x0E	WarnCode	Get_Attribute_Single	N/A, 0, 65535	C7	If in Enabled state, WarnCode indicates the lowest valued warning that caused the Warning bit to be TRUE.
15 0x0F	CtrlFromNet	Get_Attribute_Single	N/A, 0, 1	C1	Status of Run/Stop control source 0 = Control is local 1 = Control is from network

Control Supervisor Object Class (41) - Instance Attributes (1) – continued

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
16 0x10	DNFaultMode	Get_Attribute_Single, Set_Attribute_Single	0 0 2	C6	Action on loss of DeviceNet 0 = Fault and Stop 1 = Ignore 2 = Use preset speed and direction ATTENTION: Ignoring communication faults may result in equipment damage, personal injury, or death. Ensure that you understand how ignoring a communication fault may affect the operation of your system.
17 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

AC/DC Drive Object

Class Code 0x2A

AC/DC Drive Object Class (42) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 1	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"AC\DC Drive" "AC\DC Drive" "AC\DC Drive"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A N/A N/A	A1 01 C7	Array of instance IDs supported by this class

AC/DC Drive Object Class (42) - Instance Attributes (1)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
3 0x03	AtReference	Get_Attribute_Single	N/A 0 1	C1	1 = Drive actual at reference (speed or torque reference) based on mode
4 0x04	NetRef	Get_Attribute_Single, Set_Attribute_Single	0 0 1	C1	Requests torque or speed reference to be local or from the network 0 = Set Reference not DN Control 1 = Set Reference at DN Control Note that the actual status of torque or speed reference is reflected in attribute 29, RefFromNet
6 0x06	DriveMode	Get_Attribute_Single, Set_Attribute_Single	1	C6	1 = Open loop speed (Frequency) 3 = Torque control
7 0x07	SpeedActual[RPM]	Get_Attribute_Single	N/A, 0 30000	C3	Actual drive speed (best approximation) Units: RPM
8 0x08	SpeedRef[RPM]	Get_Attribute_Single, Set_Attribute_Single	1 1 30000	C3	Speed reference Units: RPM
9 0x09	CurrentActual[100mA]	Get_Attribute_Single	N/A, 0 32767	C3	Actual motor phase current Units: 100mA
10 0x0A	CurrentLimit[100mA]	Get_Attribute_Single	N/A, 0 32767	C3	Motor phase current limit Units: 100mA
11 0x0B	TorqueActual[Nm]	Get_Attribute_Single	N/A, 0 32767	C3	Actual torque Units: Nm
12 0x0C	TorqueRef[Nm]	Get_Attribute_Single, Set_Attribute_Single	0 0 10000	C3	Torque reference Units: Nm
15 0x0F	PowerActual[W]	Get_Attribute_Single	N/A, 0 32767	C3	Actual output power Units: Watts
16 0x10	InputVoltage[V]	Get_Attribute_Single	N/A 180 690	C3	Input Voltage Units: Volts
17 0x11	OutputVoltage[V]	Get_Attribute_Single	N/A 0 690	C3	Output Voltage Units: Volts
18 0x12	AccelTime[64ms]	Get_Attribute_Single	47 1 46875	C7	Acceleration time Time from 0 to HighSpdLimit Units: ms / 64 Attribute 28 (TimeScale) defines scaling factor as 64
19 0x13	DecelTime[64ms]	Get_Attribute_Single	47 1 46875	C7	Deceleration time Time from HighSpdLimit to 0 Units: ms / 64
20 0x14	LowSpdLimit[RPM]	Get_Attribute_Single	0 0 30000	C7	Minimum speed limit Units: RPM

AC/DC Drive Object Class (42) - Instance Attributes (1) -- continued

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
21 0x15	HighSpdLimit[RPM]	Get_Attribute_Single	1800 0 30000	C7	Maximum speed limit Units: RPM
29 0x1D	RefFromNet	Get_Attribute_Single	N/A 0 1	C1	Status of torque/speed reference 0 = Local torque/speed reference 1 = DeviceNet torque/speed reference
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Parameter Object

Class Code 0xA0

Parameter Class (160) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 1	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"Parameter", "Parameter", "Parameter",	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List	Get Member, CH_Get_Member	N/A, N/A, N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A, N/A, N/A,	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List	Get Member, CH_Get_Member	N/A, N/A, N/A	A1 01 C7	Array of instance IDs supported by this class

Parameter Class (160) - Class Attributes (1)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1-1999	xx	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	C7	Drive Parameters. Attributes are defined in EDS file (Depend on drive application).
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters

Monitoring Data Object

Class Code 0xAA

Monitoring Data Class (170) - Class Attributes (0)

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1 0x01	Revision	Get_Attribute_Single	1 1 65535	C7	Revision of this object
2 0x02	Max Instance	Get_Attribute_Single	1 1 1	C7	
3 0x03	Number of Instances	Get_Attribute_Single	1 1 1	C7	
4 0x04	Optional attribute list	Get_Attribute_Single	{1,176} {1,176} {1,176}	A2 04 C7 A1 01 C7	
5 0x05	Optional service list	Get_Attribute_Single	{1,16} {1,16} {1,16}	A2 04 C7 A1 01 C7	
6 0x06	Max Class Attribute ID	Get_Attribute_Single	190 190 190	C7	
7 0x07	Max Instance Attribute ID	Get_Attribute_Single	176 176 176	C7	
176 0xB0	Object Name	Get_Attribute_Single	"Monitoring Data" "Monitoring Data" "Monitoring Data"	DA	ASCII Name for the object Class
180 0xB4	Class Attribute List CH_Get_Member	Get Member,	N/A, N/A, N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes a class attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
186 0xBA	Instance Attribute List	Get Member, CH_Get_Member	N/A N/A N/A	A1 08 A2 06 C7 DA DA A1 01 C6	Each Element describes an instance attribute. The Array's elements are structs as described in the semantics section. Individual elements are accessed using the Get Member service.
190 0xBE	Instance ID List CH_Get_Member	Get Member	N/A, N/A, N/A	A1 01 C7	Array of instance IDs supported by this class.

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
1-99 0x01-0x63		Get_Attribute_Single	N/A N/A N/A	C7	Actual Values. Attributes are defined in EDS file (Depend on drive application).
101 0x65	E1	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
102 0x66	E2	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
103 0x67	E3	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
104 0x68	E4	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
105 0x69	E5	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
106 0x6A	E6	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
107 0x6B	E7	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
108 0x6C	E8	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
109 0x6D	EC	Get_Attribute_Single, Set_Attribute_Single	N/A 0 65535	C7	
110 0x6E	SafeStateType	Get_Attribute_Single, Set_Attribute_Single	0 0 2	C7	Selects Safe State response to errors which specify safe state operation. Currently only a loss of connection other than by de-allocation is a safe state error. Warning: Review the application for safe operation before specifying a value for this attribute. 0 = DriveFault (fault and stop) 1 = No Action (hold last speed) 2 = Preset Speed/Direction
111 0x6F	PresetDir	Get_Attribute_Single, Set_Attribute_Single	N/A 0 1	C7	Sets safe state direction of rotation if the Safe State Behavior attribute specifies "Preset Speed/Direction" . Warning: Review the application for safe operation before specifying a value for this attribute. Inverter will require external stop. 0 = Forward 1 = Reverse
112 0x70	PresetRPM	Get_Attribute_Single, Set_Attribute_Single	N/A 0 30000	C7	Sets safe state speed reference (RPM) if the Safe State Behavior attribute specifies "Preset Speed/Direction". Warning: Review the application for safe operation before specifying a value for this attribute. Inverter will require external stop.

#	Attribute Name	Services	Default, Minimum, Maximum	Data Type	Description
113 0x71	PresetTq	Get_Attribute_Single, Set_Attribute_Single	N/A 0 10000	C7	Sets safe state torque reference (0,00%) if the Safe State Behavior attribute specifies "Preset Speed/Direction". Warning: Review the application for safe operation before specifying a value for this attribute. Inverter will require external stop.
120 0x78	Polled Input Assembly Type	Get_Attribute_Single Set_Attribute_Single	71 70 73	C7	Input assembly used by the polled connection
121 0x79	Polled Output Assembly Type	Get_Attribute_Single Set_Attribute_Single	21 20 23	C7	Output assembly used by the polled connection
176 0xB0	User Label	Get_Attribute_Single, Set_Attribute_Single	N/A N/A N/A	DA	User Supplied name for the Instance. Maximum of 8 characters