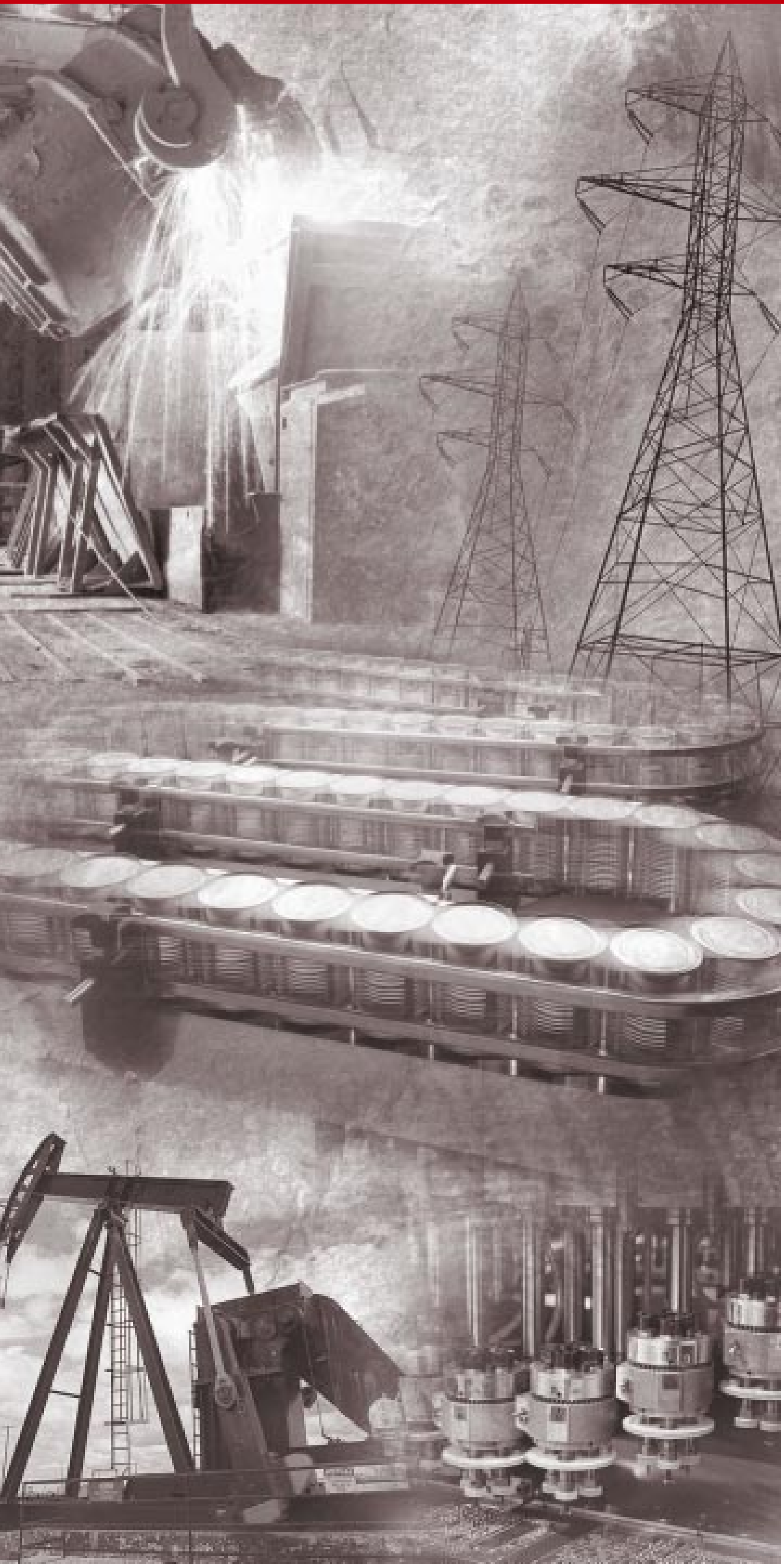


Allen-Bradley

Enhanced DeviceNet™ Communications Module

Catalog Numbers: 1203-GU6 and 1336-GM6

Firmware: 2.xxx

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Rockwell Automation does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Rockwell Automation publication SGI-1.1, *Safety Guidelines for the Application, Installation, and Maintenance of Solid-State Control* (available from your local Rockwell Automation office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

⚠ **ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attention statements help you to:

- Identify a hazard.
- Avoid the hazard.
- Recognize the consequences.

Important: Identifies information that is critical for successful application and understanding of the product.

Summary of Changes

The information below summarizes the changes made to the company wide templates since the last release.

Updated Information

Changes to the adapter firmware include the following:

Change	Description
Additions	Added an Allen-Bradley proprietary object used by the PCP group.
Corrections	Corrected known bugs in the firmware.

Changes to this manual include the following:

Location	Description
P-3, 7-1, F-1	Attention regarding Explicit messaging was added.
1-4	Information was added in the footnotes.
Chapter 4	Instructions for using DeviceNet Manager were replaced with instructions for using RSNetWorx for DeviceNet.
Chapter 5	Instructions for using DeviceNet Manager were replaced with instructions for using RSNetWorx for DeviceNet.
6-2	Data in the I/O image was corrected.
6-4	Information was added about the Logic Command Bits
6-6	A change was made to the data in rung 0010.
6-7	A change was made to the data in rung 0003.
Appendix B	Instructions for using DeviceNet Manager were removed.
B-14	Value range for parameter 27 was added. Description for parameter 31 was corrected.
C-20, C-22, C-23	Service Code for the Set_Attribute_Single was corrected.

Notes:

Table of Contents

Preface

Using this Manual

Chapter 1

Overview

Chapter 2

Installation

Chapter 3

Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection

Chapter 4

Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet

Chapter 5

Configuring a Scanner to Communicate with the Adapter

Chapter 6

Ladder Logic Programming

Chapter 7

Using DeviceNet Explicit Messaging

Chapter 8

Troubleshooting

Appendix A

Specifications

Appendix B

Enhanced DeviceNet Adapter’s Parameters

Appendix C

DeviceNet Objects

Appendix D

Supported PCCC Messages

Appendix E

N-File Addresses

Appendix F

Supported Emulated Block Transfer Commands

Appendix G

Event Queue Messages

Using this Manual

Objectives

Read this preface to become familiar with the organization of the manual. In this preface, you will read about the following:

- Who should use this manual.
- An overview of the Enhanced DeviceNet™ communications adapter.
- The purpose of this manual.
- Terms and abbreviations.
- Conventions used in this manual.
- Rockwell Automation support.

Who Should Use this Manual?

Use this manual if you are responsible for installing, wiring, programming, or troubleshooting control systems that use the Enhanced DeviceNet communications adapter.

This manual is intended for qualified service personnel responsible for setting up and servicing the Enhanced DeviceNet communications adapter. You must have previous experience with and a basic understanding of electrical terminology, programming procedures, networking, required equipment and software, and safety precautions.

Purpose of this Manual

This manual is a learning and reference guide for the Enhanced DeviceNet communications adapter. It describes the procedures needed to install, configure, and troubleshoot the adapter.

Contents of this Manual

This manual contains the following information:

Chapter	Title	Contents
Preface	*Using this Manual*	Describes the purpose, background, and scope of this manual. Also provides information on safety precautions and technical support.
1	*Overview*	Provides an overview of the communications adapter, DeviceNet, and SCANport.
2	*Installation*	Provides procedures for installing the 1203-GU6 module or 1336-GM6 board.
3	*Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*	Provides procedures for configuring the 1203-GU6 using a serial connection. Topics include how to set up a serial connection to the 1203-GU6, how to navigate the module's software, edit parameters, perform a flash upgrade to its firmware, and view its event queue.
4	*Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*	Provides procedures for configuring the 1203-GU6 module or 1336-GM6 board over DeviceNet using RSNetWorx for DeviceNet.
5	*Configuring a Scanner to Communicate with the Adapter*	Provides procedures for using RSNetWorx for DeviceNet to configure scanners to communicate with devices such as the 1203-GU6 module and 1336-GM6 board.
6	*Ladder Logic Programming*	Provides an example ladder logic program for a PLC and an example ladder logic program for an SLC.
7	*Using DeviceNet Explicit Messaging*	Provides information about explicit messaging, including messaging with PLCs, messaging with SLCs, and using messages to control the SCANport product.
8	*Troubleshooting*	Provides information on the various states of the adapter's LEDs, including Network Status, Module Status, and SCANport Status.
A	*Specifications*	Provides the specifications for the 1203-GU6 module and 1336-GM6 board.
B	*Enhanced DeviceNet Adapter's Parameters*	Provides information on the adapter's parameters and how to use them to configure the adapter.
C	*DeviceNet Objects*	Defines the DeviceNet object classes, class services, and attributes that are supported by the Enhanced DeviceNet adapter.
D	*Supported PCCC Messages*	Provides a reference list of PCCC messages supported by the Enhanced DeviceNet adapter.
E	*N-File Addresses*	Lists the N-files to which messages can be written.
F	*Supported Emulated Block Transfer Commands*	Provides a reference list of emulated block transfer commands.
G	*Event Queue Messages*	Provides a list of messages you may see in the event queue.

Related Publications

Title	Publication Number
RSNetWorx for DeviceNet Getting Results Manual	9398-DNETGR
1771-SDN Scanner Configuration Manual	1771-6.5.118
1747-SDN DeviceNet Scanner Configuration Manual	1747-6.5.2
DeviceNet Cable System Planning and Installation Manual	DN-6.7.2

Safety Precautions

Please read the following safety precautions carefully.

⚠ **ATTENTION:** Only personnel familiar with SCANport products and associated machinery should plan or implement the installation, start-up, configuration, and subsequent maintenance of the Enhanced DeviceNet communications adapter. Failure to comply may result in personal injury and/or equipment damage.

⚠ **ATTENTION:** Remove all power from the SCANport product before installing the 1336-GM6 board. Failure to disconnect power may result in death or serious injury. Verify all power is removed before installing the 1336-GM6 board.

⚠ **ATTENTION:** Hazard of equipment damage exists. If explicit messages are programmed to frequently write parameter data to certain drive products, the EEPROM (Non-Volatile Storage) will quickly exceed its life cycle and cause the product to malfunction. Do not create a program that frequently uses explicit messages to write parameter data to a product. Datalinks do not write to the EEPROM and should be used for frequently changed parameters.

Terms and Abbreviations

For a complete listing of Allen-Bradley terminology, refer to the Allen-Bradley *Industrial Automation Glossary*, Publication AG-7.1.

Terms	Definition
DeviceNet	An open network that provides probabilistic I/O control through a managed bit-wise non-destructive multiplexing scheme.
Enhanced DeviceNet adapter	Both the 1203-GU6 module and the 1336-GM6 board are enhanced DeviceNet adapters. In this manual, the term “adapter” is used when both the module and the board are referred to.
RSNetWorx for DeviceNet	A Rockwell Software application that can be used to set up DeviceNet networks and configure connected devices. RSNetWorx for DeviceNet (version 2.22.18) and RSLinx (version 2.10.118) were used for examples in the manual. Different versions may differ in appearance and procedures.
SCANport	A standard peripheral communications interface for various Allen-Bradley drives and power products.
SCANport Peripheral	A device that provides an interface between SCANport and a network. It is often referred to as an adapter. For example, the Enhanced DeviceNet adapter is a SCANport peripheral.
SCANport Product	A device that uses the SCANport communications interface to communicate with one or more peripheral devices. For example, a motor drive such as a 1336 PLUS is a SCANport product.

Conventions Used in this Manual

The following conventions are used throughout this manual:

- Bulleted lists provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for chapter names and for parameter names.
- **Bold** type is used for names of menus, menu options, screens, and dialog boxes.

Important: This type of paragraph contains tips or notes that have been added to call attention to useful information.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with more than 75 sales/support offices, more than 500 authorized distributors, and more than 250 authorized systems integrators located throughout the United States alone. In addition, Rockwell Automation representatives are in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- Sales and order support.
- Product technical training.
- Warranty support.
- Support service agreements.

Technical Product Support

If you need to contact Rockwell Automation for technical assistance, please call your local Rockwell Automation representative.

Refer to http://www.ab.com for updates and supporting documentation.

Chapter **1**

Overview

Chapter Objectives

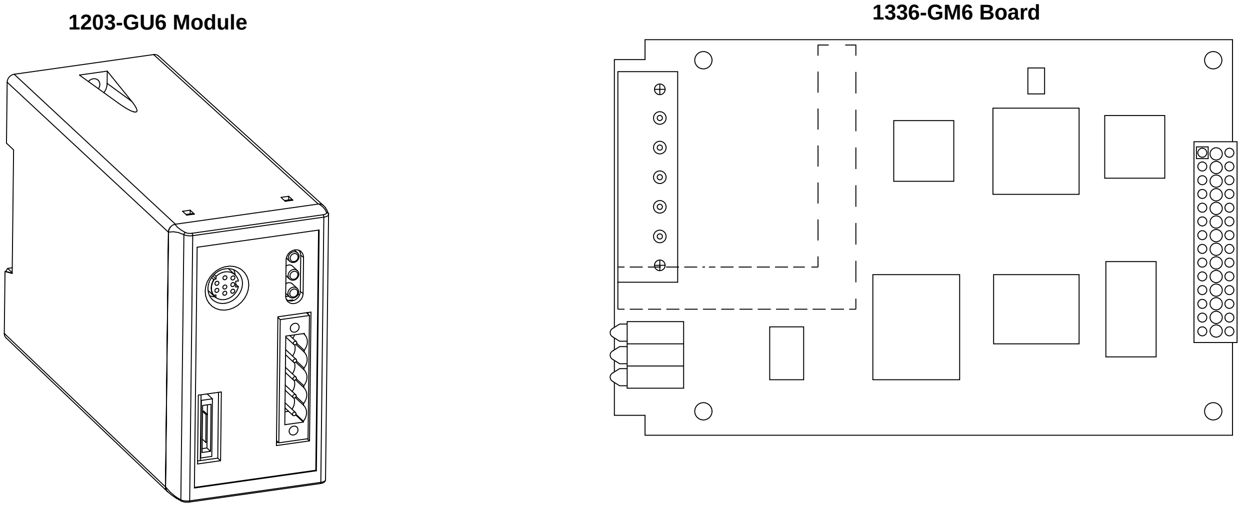
Chapter 1 provides an overview of your Enhanced DeviceNet communications adapter. In this chapter, you will read about the following:

- Function of the 1203-GU6 module or 1336-GM6 board.
- Features of the 1203-GU6 module and 1336-GM6 board.
- SCANport products.
- Parts and hardware of the 1203-GU6 module and 1336-GM6 board.
- Steps for setting up the adapter.
- Required tools and equipment.

Overview of the Communications Adapter

There are two types of Enhanced DeviceNet adapters: the 1203-GU6 module and 1336-GM6 board.

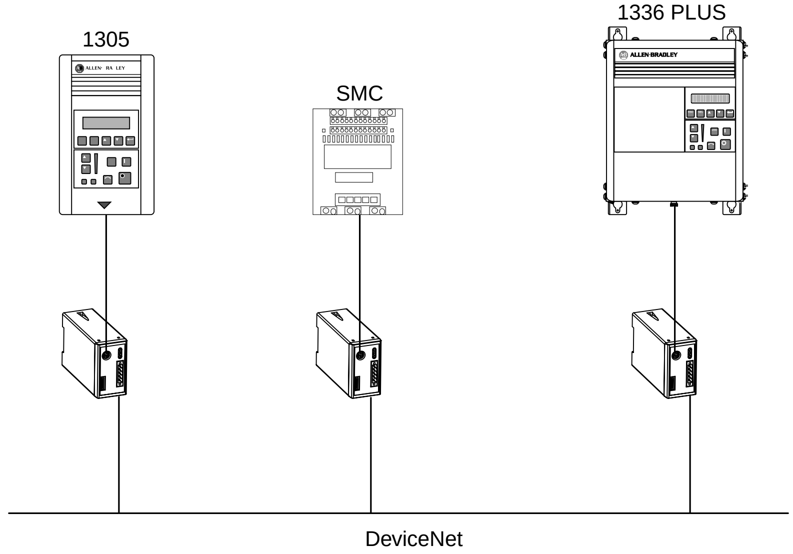
Figure 1.1 1203-GU6 Module and 1336-GM6 Board



The 1203-GU6 module mounts on a DIN rail and connects to the SCANport product via a SCANport cable. The 1336-GM6 board is mounted directly onto selected SCANport products so it connects to the SCANport product via an internal SCANport connector.

Both types of Enhanced DeviceNet communications adapter provide an electronic communications interface between a DeviceNet network and any single SCANport product.

Figure 1.2 Example DeviceNet NetWork



In Figure 1.2, a SCANport cable connects a 1203-GU6 module to a SCANport product through a port on the SCANport product. A DeviceNet cable connects the module to the DeviceNet network. The module then translates the DeviceNet messages into SCANport messages that can be understood by the connected product.

Features of the Communications Adapter

The DeviceNet network is an open, global industry-standard communication network designed to provide an interface through a single cable from a programmable controller directly to “smart” devices such as sensors, push buttons, motor starters, simple operator interfaces and drives.

The 1203-GU6 module and 1336-GM6 board let you connect your SCANport products to a DeviceNet network. These adapters feature the following:

- Flash upgradeability allows for field updates in the event of changes to the adapter’s firmware.
- COS (Change of State) capability lets you customize this device’s activity on the network by configuring the adapter to report only new data.
- Cyclic operation lets you customize the devices’s activity on the network by configuring the adapter to report its data at specific intervals.

- Peer I/O capabilities let the drive’s I/O (logic command, reference, logic status, feedback and datalinks) be broadcast to or received from other drives connected via 1203-GU6 or 1336-GM6 adapters.
- Software configuration lets you configure the adapter using RSNetWorx for DeviceNet or HyperTerminal (a standard Windows 95 program).
- Faulted Node Recovery lets you change items, such as a node address, of a device even when it is faulted on the network. (This feature requires the support of proper PC software tools.)
- User-configurable fault response provides the ability to customize the adapter’s actions to communication errors.
- A miniature RS-232 port on the front of the 1203-GU6 (not available on the 1336-GM6) provides access for your PC to change parameters in the module using HyperTerminal. It also provides a DF1 connection for DriveTools32. (You must configure DriveTools32 and RSLinx for DF1 communications and then connect to the module using a 1203-SFC cable [sold separately].)
- A new Module Status LED helps to diagnose network, module, and SCANport product health.
- Re-designed power supply receives its operating power from the connected drive product. Only minimal DeviceNet-side transceiver power is consumed by the user-supplied 24V DC power source.

SCANport Products

Some SCANport products support one peripheral; others support up to six peripherals. The table below lists SCANport products, the number of peripherals each supports, the minimum and maximum I/O words, and the type of adapter that can be used.

Product	Number of Peripherals Supported	I/O Words		Adapter Use	
		Minimum	Maximum	1203-GU6	1336-GM6
1305 AC MICRO Drive[4]	5	0	10	Yes	No
1336 IMPACT™ Drive	6[1]	0	10	Yes	Yes[2]
1336 PLUS AC Drive	6[1]	0	10	Yes	Yes[2]
1336 PLUS II Drive	6[1]	0	10	Yes	Yes
1336 FORCE™ Drive	6[1]	0	10	Yes	Yes[2][3]
1394 AC Mult-Axis Motion Control System	5	0	10	Yes	No
SMC Dialog Plus	1	0	2	Yes	No
SMP-3 Smart Motor Protector	2	0	2	Yes	No
1397 Digital DC Drive	5	0	10	Yes	No
1557 Medium Voltage Drive	5	0	10	Yes	No

[1] Lower horsepower products may not support a sixth peripheral. Refer to your user manual to verify that your product supports a sixth peripheral.

[2] B frame or larger is required.

[3] A standard adapter board is required.

[4] Early versions of the 1305 drive may not support some communications.

Important: To connect multiple peripherals to a SCANport product, a port expander may be required. Refer to your product’s documentation for more information.

Important: If you intend to use datalinks to communicate with and control your SCANport product, verify that your SCANport product supports datalinks before enabling them in the adapter.

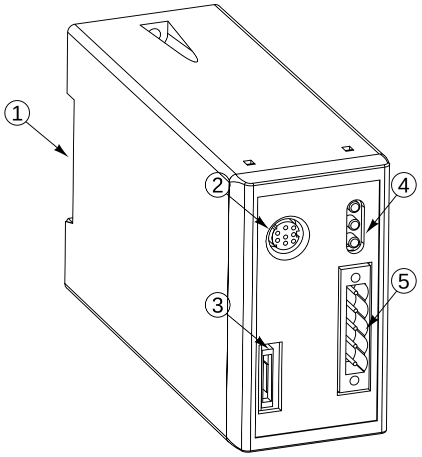
Hardware and Parts Description

The hardware included with the adapter depends on the adapter that you have.

1203-GU6 Module Hardware

Figure 1.3 illustrates and the following table lists the main parts of the 1203-GU6 Enhanced DeviceNet communications module:

Figure 1.3 Parts of the 1203-GU6 Module

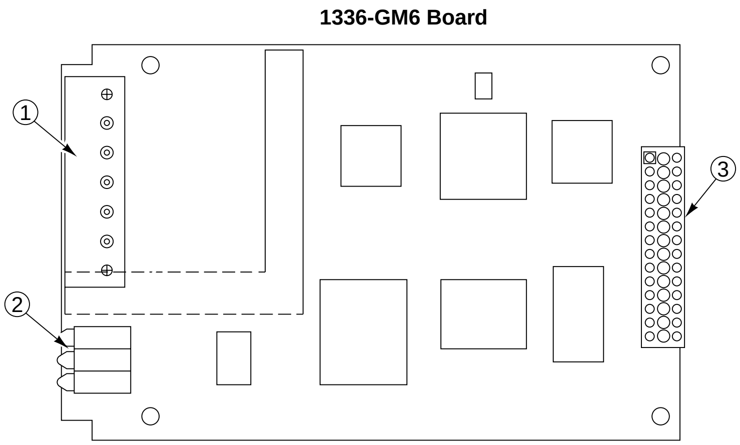


Number	Part	Description
1	DIN Rail Mount	Securely attaches and electronically grounds the module to the DIN rail.
2	SCANport Connection	Provides a standard SCANport 8-pin circular mini-DIN connector for the SCANport cable.
3	RS-232 Serial Port	Provides a connection for terminals capable of RS-232 serial communications. This port can be used to edit the module’s parameters, download a file needed to perform a flash to the module’s operating code, and support devices that monitor and test the module. A 1203-SFC serial cable and a PC running a terminal emulation program or a VT100-compatible terminal are required to use this port.
4	Bi-Color LEDs	Indicate the status of the DeviceNet media channel, of the SCANport connection, and of the module. For more information, refer to Chapter 8, Troubleshooting.
5	DeviceNet Connection	Provides a 5-pin Phoenix connector to attach the module to the network.
Not shown	5-Pin Plug-In Connector	This part is supplied with the module. The 5-pin plug-in connector (P/N 22112-215-01) is a connector to attach to the cable.
Not shown	10-Pin Plug-In Connector	This part is supplied with the module. The 10-pin plug-in connector (P/N 94220605) is a connector to attach to the cable.

1336-GM6 Board Hardware

Figure 1.4 illustrates and the following table lists the main parts of the 1336-GM6 Enhanced DeviceNet communications board:

Figure 1.4 Parts of the 1336-GM6 Board



Number	Part	Description
1	DeviceNet Connection	Provides a 5-pin Phoenix connector to attach the module to the DeviceNet network.
2	Bi-Color LEDs	Indicate the status of the DeviceNet media channel, of the SCANport connection, and of the module. For more information, refer to Chapter 8, *Troubleshooting*.
3	SCANport Connection	Provides a 14-pin connector containing power and SCANport communication circuitry.
Not Shown	Kit	Provides the necessary materials for mounting the board to the SCANport product. These materials include one grounding wrist strap (P/N 22803-016-01), four Phillips mounting screws (P/N 28159-036-26), four stand-off nylon headers (P/N 22805-030-01), one 5-pin connector (P/N 22112-215-01), and one snap-in comm housing (P/N 188578) with mounting instructions (P/N 189572).

Overview of Setting Up the Adapter

To set up the Enhanced DeviceNet adapter, you must perform the following tasks:

1. Install the module or mount the board. Refer to Chapter 2, *Installation*.
2. Set the adapter's node address and configure the adapter's parameters. Refer to the following table:

If using:	Then refer to:
1203-GU6 module's serial connection	Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*
1203-GU6 module's DeviceNet connection	Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*
1336-GM board	Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*

3. Configure a scanner (either PLC or SLC) to communicate with the Adapter. Refer to Chapter 5, *Configuring a Scanner to Communicate with the Adapter*.
4. If necessary, create a ladder logic program to control the SCANport product. Refer to Chapter 6, *Ladder Logic Programming*.

Required Tools and Equipment

The tools and equipment required, depend on if you are using a 1203-GU6 module or 1336-GM6 board.

1203-GU6 Module

To install and configure a 1203-GU6 module, you need the following:

- Enhanced DeviceNet communications module (1203-GU6).
- 5-pin or 10-pin plug-in connector (supplied with module).
- 35 x 7.5 mm DIN rail A (Allen-Bradley part number 199-DR1; 46277-3; EN 50022).
- 1/8" flathead screwdriver.
- Appropriate cables for SCANport and DeviceNet connections. Refer to the Selecting Cables section in Chapter 2, *Installation*.
- A 1203-SFC serial cable.
- A PC running either a Windows terminal emulation program (e.g., HyperTerminal) or a VT100-compatible terminal.

- A PC that is:
  - Running RSNetWorx for DeviceNet.
  - Connected to and communicating with the DeviceNet network using a 1784-PCD card or a 1770-KFD adapter.
  - Running RS Linx.
  - Running RSLogix5 (if using PLC) or RSLogix500 (if using SLC).

Important: Refer to http://www.software.rockwell.com for more information on these software products.

1336-GM6 Board

To install and configure a 1336-GM6 board, you need the following:

- Enhanced DeviceNet communications board (1336-GM6).
- A kit that includes one grounding wrist strap, four Phillips mounting screws, four stand-off nylon headers, one 5-pin connector and one snap-in comm housing with mounting instructions (supplied with board).
- #1 Phillips screwdriver.
- Appropriate cable for the DeviceNet connection. Refer to the Selecting Cables section in Chapter 2, *Installation*.
- A PC that is:
  - Running RSNetWorx for DeviceNet.
  - Connected to and communicating with the DeviceNet network using a 1784-PCD card or a 1770-KFD adapter.
  - Running RS Linx.
  - Running RSLogix5 (if using PLC) or RSLogix500 (if using SLC).

Important: Refer to http://www.software.rockwell.com for more information on these software products.

Installation

Chapter Objectives

Chapter 2 provides the information that you need to install the 1203-GU6 module or 1336-GM6 board. In this chapter, you will read about the following:

- Required tools and equipment.
- Selecting cables.
- Installing the adapter.
- Removing the adapter.

How you install the adapter depends on whether you are installing a 1203-GU6 module or 1336-GM6 board. Refer to the following table for information:

If installing:	Then refer to:
1203-GU6 module	"Installing a 1203-GU6 Module" on page 1
1336-GM6 board	"Installing a 1336-GM6 Board" on page 6

Installing a 1203-GU6 Module

Follow these procedures to install a 1203-GU6 module.

Required Tools and Equipment

To install your 1203-GU6 module, you will need the following tools and equipment:

- Enhanced DeviceNet communications module (1203-GU6).
- Either a 5-pin or 10-pin plug-in connector (supplied with module).
- 35 x 7.5 mm DIN rail A (Allen-Bradley part number 199-DR1; 46277-3; EN 50022).
- 1/8" flathead screwdriver.
- Appropriate cables for SCANport and DeviceNet connections. Refer to the "Selecting Cables" section in this chapter.

Selecting Cables

To connect the 1203-GU6 to the SCANport product and the DeviceNet network, you must select an appropriate DeviceNet cable and Allen-Bradley SCANport cable. Use the following information to select appropriate cables for each connection.

SCANport Cables

When selecting the SCANport cable to connect the 1203-GU6 module to the SCANport product, you need to:

- Use an Allen-Bradley SCANport cable. Refer to the table below.

Male to Male Connection		Male to Female Connection	
Length	**Catalog Number**	**Length**	**Catalog Number**
1/3 m	1202-C03	1/3 m	1202-H03
1 m	1202-C10	1 m	1202-H10
3 m	1202-C30	3 m	1202-H30
9 m	1202-C90	9 m	1202-H90

- Use less than 10 meters (33 feet) of cable between the SCANport product and adapter.
- Keep SCANport cables away from high power cables to guard against introducing noise into your system.

DeviceNet Cables

The 1203-GU6 module comes with either one 5-pin connector or a 5-pin and 10-pin connector. A drop line connects a node, such as an adapter, on the DeviceNet cable system to the DeviceNet trunk. When selecting a drop line, you need to:

- Select either a 5-pin connector or a 10-pin connector for the cable. (Both are included with the 1203-GU6 module).
  - Use the 5-pin connector if you are using a DeviceNet drop cable configuration.
  - Use the 10-pin connector if you are using a daisy-chaining configuration.
- Determine limitations of the trunk and drop cables. Refer to the following table.

Data Rates	125 Kbps	250 Kbps	500 Kbps
Thick Trunk Line	500 m (1,640 ft)	250 m (820 ft)	100 m (328 ft)
Thin Trunk Lengths	100 m (328 ft)	100 m (328 ft)	100 m ((328 ft)
Maximum Drop Length	6 m (20 ft)	6 m (20 ft)	6 m (20 ft)
Cumulative Drop Budget	156 m (512 ft)	78 m (256 ft)	39 m (128 ft)

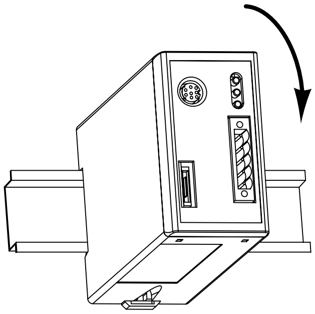
For more information on DeviceNet cables and cable systems, refer to the DeviceNet Cable System Planning and Installation Manual, Publication DN-6.7.2.

Installing the 1203-GU6 Communications Module

The following instructions explain how to physically install your Enhanced DeviceNet 1203-GU6 communications module.

1. Hook the top lip of the module's DIN rail mount onto the top of the DIN rail and then rotate the module onto the DIN rail. You will hear the module snap into a locked position.

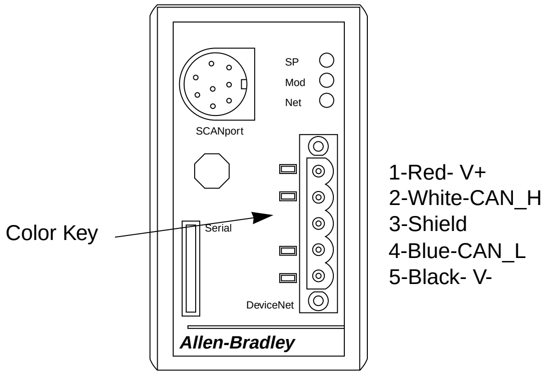
Figure 2.1 Mounting the Module onto the DIN Rail



2. Remove power from the network.
3. Insert the DeviceNet cable wires into the desired connector. Make sure you follow the color key next to the connector receptacle on the module.

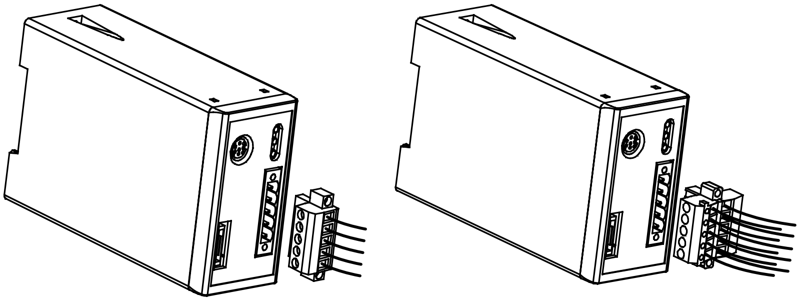
ATTENTION: If you wire the 5-pin or 10-pin header after you've connected it to the module, static control precautions are required. Device malfunction may occur if you do not follow ESD control procedures. If you are not familiar with static control procedures, refer to Allen-Bradley Publication 8000-4.5.2, *Guarding Against Electrostatic Damage*, or other applicable ESD protection handbook.

Figure 2.2 Inserting DeviceNet Cable Wires into the Connector



4. Plug the connector into the module.

Figure 2.3 Inserting a 5-pin or 10-pin Phoenix Connector

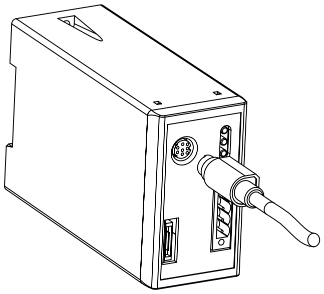


5. Use a 1/8" flathead screwdriver to screw the connector firmly into place.

6. Connect the SCANport cable to the communications adapter and then to the SCANport product.

Important: Because the adapter's power comes from the SCANport product, the adapter will not function if the SCANport product is not powered.

Figure 2.4 Inserting the SCANport Cable



7. Reapply power to the network.

8. If necessary, apply power to the connected SCANport product.

Your 1203-GU6 module is now installed. The SCANport LED is green. The network and module LEDs are blinking green. If your module's LEDs are different, refer to Chapter 8, *Troubleshooting*, for more information.

You must now edit the adapter’s node address, and you may want to edit some of its other parameters. Refer to the following table for information:

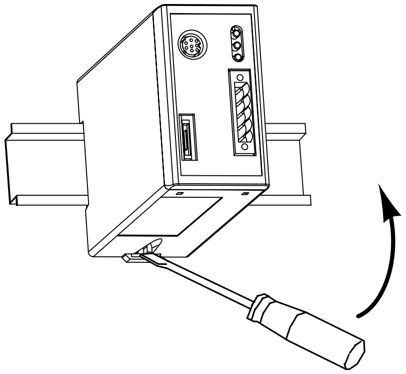
If using:	Then refer to:
1203-GU6 module’s serial connection	Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*
1203-GU6 module’s DeviceNet connection	Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*

Removing the 1203-GU6 Communications Module

To remove the 1203-GU6 communications module, you need to:

1. If necessary, remove any serial connections from the DF-1 serial port.
2. Remove the SCANport cable from the SCANport product and then from the module. This will power down the module if the SCANport product is powered.
3. Unscrew (using a 1/8" flathead screwdriver) and then unplug the 5-pin or 10-pin connector from the module.
4. Insert a flathead screwdriver into the module’s DIN rail tab release and gently push the handle toward the module to release the connection tab. Then pull the module up and off the DIN rail.

Figure 2.5 Removing the Module from the DIN Rail



Installing a 1336-GM6 Board

Follow these procedures to install a 1336-GM6 board.

Required Tools and Equipment

To install your 1336-GM6 board, you will need the following tools and equipment:

- Enhanced DeviceNet communications board (1336-GM6).
- A kit that includes one grounding wrist strap, four Phillips mounting screws, four stand-off nylon headers, one 5-pin connector and one snap-in comm housing with mounting instructions (supplied with board).
- #1 Phillips screwdriver.
- Appropriate cable for the DeviceNet connection. Refer to the Selecting Cables section below.

Selecting Cables

To connect the Enhanced DeviceNet adapter to the SCANport product and the DeviceNet network, you must select an appropriate DeviceNet cable. Use the following information to select appropriate cables for each connection.

DeviceNet Cables

The 1336-GM6 board comes with one 5-pin connector. A drop line connects a node, such as an adapter, on the DeviceNet cable system to the DeviceNet trunk. When selecting a drop line, you need to:

- Use the 5-pin connector for the cable if you are using a DeviceNet drop cable configuration.
- Determine limitations of the trunk and drop cables. Refer to the following table.

Data Rates	125 Kbps	250 Kbps	500 Kbps
Thick Trunk Line	500 m (1,640 ft)	250 m (820 ft)	100 m (328 ft)
Thin Trunk Lengths	100 m (328 ft)	100 m (328 ft)	100 m ((328 ft)
Maximum Drop Length	6 m (20 ft)	6 m (20 ft)	6 m (20 ft)
Cumulative Drop Budget	156 m (512 ft)	78 m (256 ft)	39 m (128 ft)

For more information on DeviceNet cables and cable systems, refer to the DeviceNet Cable System Planning and Installation Manual, Publication DN-6.7.2.

Electrostatic Discharge Precautions

Please read the following safety precaution carefully before installing the 1336-GM6 communications board.

⚠ **ATTENTION:** The 1336-GM6 communications board contains ESD (Electrostatic Discharge) sensitive parts. Static control precautions are required when installing, testing, or servicing this board. Device malfunction may occur if you do not follow ESD control procedures. If you are not familiar with static control procedures, refer to Allen-Bradley Publication 8000-4.5.2, *Guarding Against Electrostatic Damage*, or other applicable ESD protection handbook.

Installing the 1336-GM6 Communications Board

The following instructions explain how to physically install an Enhanced DeviceNet 1336-GM6 communications board.

Important: If you are attaching the communications board to a 1336 PLUS II, refer to the one-page insert included with the kit for mounting instructions.

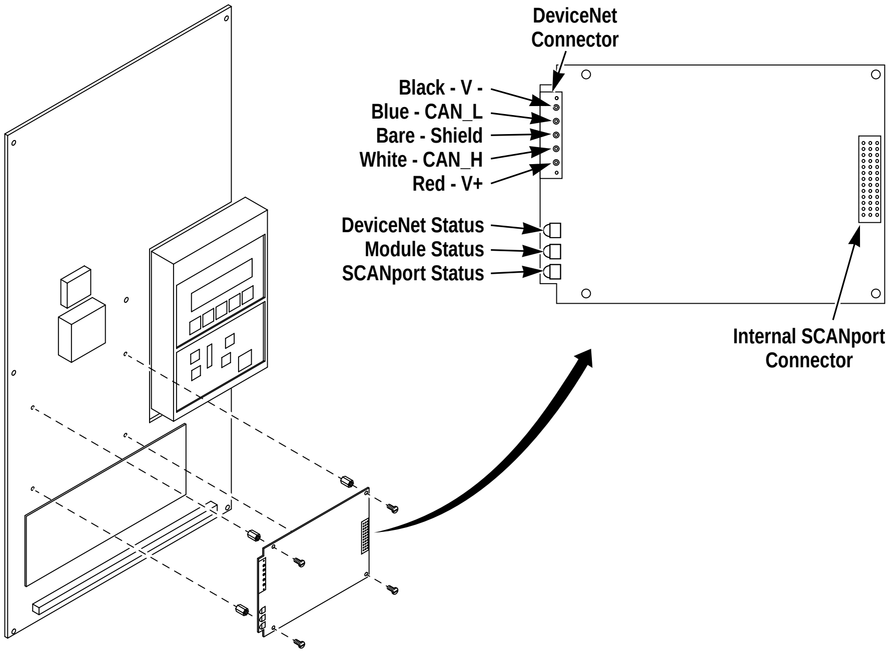
Important: To prevent damage to the board, you must wear a grounding wrist strap when handling the 1336-GM6 communications board.

⚠ **ATTENTION:** Remove all power from the SCANport product before installing the 1336-GM6 board. Failure to disconnect power may result in death or serious injury. Verify all power is removed before installing the 1336-GM6 board.

1. Make sure you've turned off the power supply to the SCANport product and DeviceNet network and verified the drive is not receiving power.

2. Screw the four stand-off nylon headers into the appropriate spaces on the drive's main control board.

Figure 2.6 **Mounting the Open Style Communications Board**



3. Insert the pins located on the 1336-GM6 into the 14-pin SCANport header on the drive. The board should sit squarely on the stand-offs.

4. Using a #1 Phillips screwdriver and the four supplied mounting screws, screw the board securely into place, being careful not to overtighten.

5. Attach the wires to the supplied DeviceNet connector.

Important: Refer to Figure 2.6 to verify you've wired the connector correctly to the board. Failure to wire the board correctly may damage the circuitry or cause the device to not function.

ATTENTION: Static control precautions are required if you wire the connector when it is already connected to the adapter. It is not recommended that you do this. Device malfunction may occur if you do not follow ESD control procedures. If you are not familiar with static control procedures, refer to Allen-Bradley Publication 8000-4.5.2, *Guarding Against Electrostatic Damage*, or other applicable ESD protection handbooks.

6. Plug the connector into the board receptacle and screw the two screws into place using a 1/8" flathead screwdriver.

7. Reapply power to the SCANport product.

8. Reapply power to the DeviceNet network.

Your 1336-GM6 board is now installed. The SCANport LED is green. The network and module LEDs are blinking green. If your module's LEDs are different, refer to Chapter 8, *Troubleshooting*, for more information.

You must now edit the adapter's node address, and you may want to edit some of its other parameters. Refer to Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*.

Removing the 1336-GM6 Communications Board

To remove the 1336-GM6 communications board, you need to:

Important: To prevent damage to the board, you must wear a grounding wrist strap when handling the 1336-GM6 communications board.

Important: If you are removing the communications board from a 1336 PLUS II, refer to the one-page insert included with the kit for special mounting instructions.

⚠ **ATTENTION:** Remove all power from the SCANport product before removing the 1336-GM6 board. Failure to disconnect power may result in death or serious injury. Verify all power is removed before installing the 1336-GM6 board.

1. Turn off the power supply to the drive and DeviceNet network.

2. Unscrew (using a 1/8" flathead screwdriver) and then unplug the DeviceNet connector from the communications board.

3. Unscrew and remove the board's four mounting screws with a #1 Phillips screwdriver.

4. Making sure not to bend the pins as they slide out of the 14-pin SCANport header, gently pull the communications board away from the main control board.

5. Unscrew and remove the four stand-offs from the main control board.

Notes:

Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection

Chapter Objectives

Chapter 3 provides information that you need to configure the 1203-GU6 module using a serial connection. In this chapter, you will read about the following:

- Factory-default settings for the module.
- Equipment necessary to make a serial connection to the 1203-GU6 module.
- Connecting either a PC running terminal emulation software or VT100-compatible terminal to the 1203-GU6 module.
- Using HyperTerminal to configure or edit the module's parameters, display its event queue, view its I/O data, view DF-1 statistics or serial number, and perform a flash upgrade.

Important: DriveExplorer™ can also be used to configure the adapter via its serial port. Refer to the DriveExplorer documentation from more information. To configure a 1203-GU6 module over the DeviceNet network or to configure a 1336-GM6 board, refer to Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet.*

Factory Default Settings for the 1203-GU6 Module

The factory-default settings of the Enhanced DeviceNet adapter include the following:

- 16-bit Logic Command/Status enabled for polling.
- 16-bit Reference/Feedback enabled for polling.
- If the scanner is put into program mode or the network fails, the SCANport product will be faulted by the module.
- A node address of 63.
- DeviceNet autobaud detection enabled.
- Serial baud rate of 9600.

You should change the node address by editing the *DN Node Address* (2) parameter. You must change the autobaud detection if no other devices on your DeviceNet network have a fixed data rate by editing *DN Data Rate* (3) parameter in the module.

Important: Refer to Appendix B, *Enhanced DeviceNet Adapter's Parameters*, for more information on changing the node address or data rate.

If you wish to change other functions (e.g., Fault Configurable inputs) or add more functions (e.g., datalinks), you must edit the adapter's parameters. To do so, refer to:

- Appendix B, *Enhanced DeviceNet Adapter's Parameters*, for detailed information about the adapter's parameters.
- Instructions in this chapter on establishing a serial connection.
- Instructions in this chapter on editing parameters.

Required Tools and Equipment

To make a serial connection to the module, you need the following:

- A 1203-SFC serial cable.
- Either PC running a terminal emulation program (e.g., HyperTerminal) or a VT100-compatible terminal.

Establishing a Serial Connection to the Module

The 1203-GU6 module's software lets you do the following:

- Edit its parameters.
- View its event queue.
- View its I/O data values.
- View DF1 statistics.
- View its serial number.
- Perform a flash upgrade.

To access this software, you must make a serial connection to the module from a PC running terminal emulation software or from a VT100-compatible terminal. Refer to the following table:

If using:	Refer to page:
PC running terminal emulation software	3-3
VT100-compatible terminal	3-8

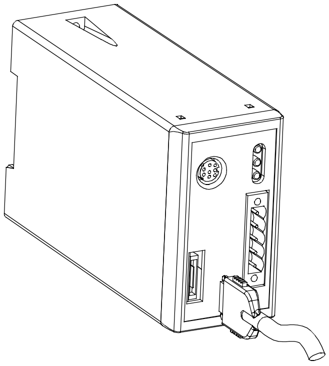
Using a PC Running Terminal Emulation Software

A variety of terminal emulation programs can be used to establish a serial connection to the module. The following instructions describe how to establish the initial serial connection to the module using a PC running HyperTerminal software. Future connections to the module can use this same configuration by clicking the icon added to the initial screen when the configuration is saved.

Important: The following procedures use HyperTerminal on a computer running Windows 95. If you are not using HyperTerminal or if you are using HyperTerminal on Windows NT, steps to establish a serial connection and screens may differ from the following steps and screens.

1. Connect a 1203-SFC serial cable to your PC's RS-232 serial port and then to the serial port on the module.

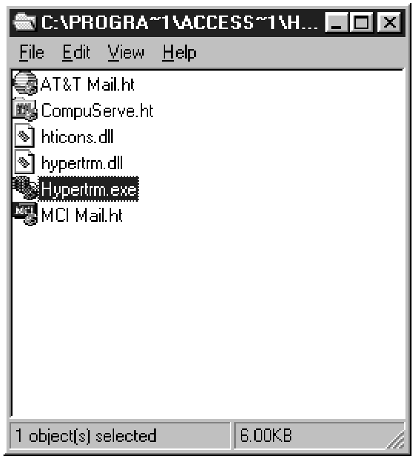
Figure 3.1 Connecting the 1203-SFC Serial Cable to the Serial Port



2. In Windows 95, click the **Start** button and then select **Programs**, **Accessories**, and **HyperTerminal**.

Important: A **HyperTerminal** dialog box appears in either List or Icon mode. Figure 3.2 shows the dialog box in List mode. To change from Icon mode to List mode, select **View**, then **List**.

Figure 3.2 Example HyperTerminal Dialog Box

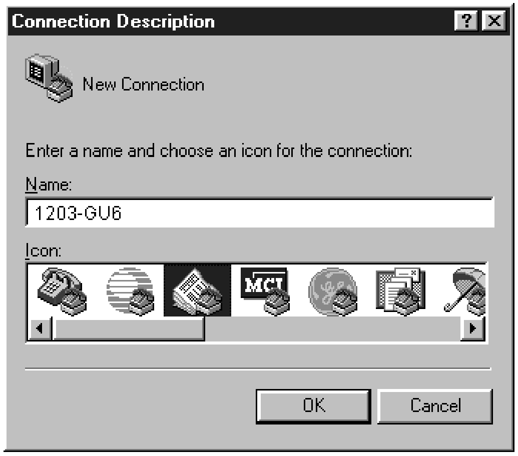


AT&T Mail.ht
CompuServe.ht
hticons.dll
hypertrm.dll
Hypertrm.exe
MCI Mail.ht

1 object(s) selected 6.00KB

3. Double-click **HyperTrm.exe**. The **Connection Description** dialog box appears.

Figure 3.3 Example Connection Dialog Box



New Connection

Enter a name and choose an icon for the connection:

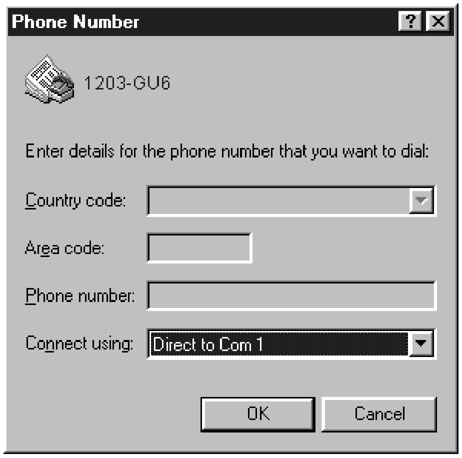
Name:
1203-GU6

Icon:

4. Enter a name in the **Name** field and select any icon from the **Icon** field.

5. Click **OK**. The **Phone Number** dialog box appears.

Figure 3.4 Example Phone Number Dialog Box



Phone Number

1203-GU6

Enter details for the phone number that you want to dial:

Country code:

Area code:

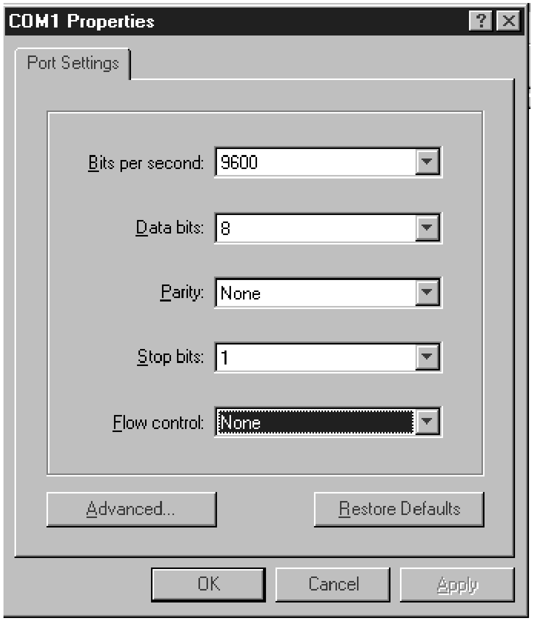
Phone number:

Connect using: Direct to Com 1

6. In the **Connect Using** field, select the appropriate communications port (usually COM1 or COM2).

7. Click **OK**. The **Comm Properties** dialog box appears.

Figure 3.5 Example Comm Properties Dialog Box



COM1 Properties

Port Settings

Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

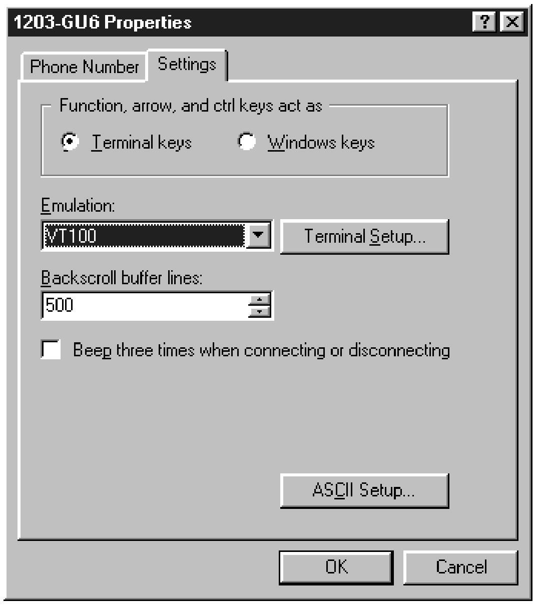
8. Select the following settings:

- **9600** in the **Bits per second** field.
  If you have previously set the module's *Serial Port Rate* parameter to enable 19200 bps, set the bps to 19200 in this field.
- **8** in the **Data bits** field.
- **None** in the **Parity** field.
- **1** in the **Stop bits** field.
- **None** in the **Flow control** field.

9. Click **OK**. The **HyperTerminal** screen appears.

10. From the **File** menu, select **Properties**. The **Properties** dialog box appears.

Figure 3.6 **Properties Dialog Box**



1203-GU6 Properties

Function, arrow, and ctrl keys act as

● Terminal keys ○ Windows keys

Emulation: VT100

Backscroll buffer lines: 500

□ Beep three times when connecting or disconnecting

11. Click the **Settings** tab.

12. In the **Function, arrow, and ctrl keys act as** field, verify **Terminal keys** is selected.

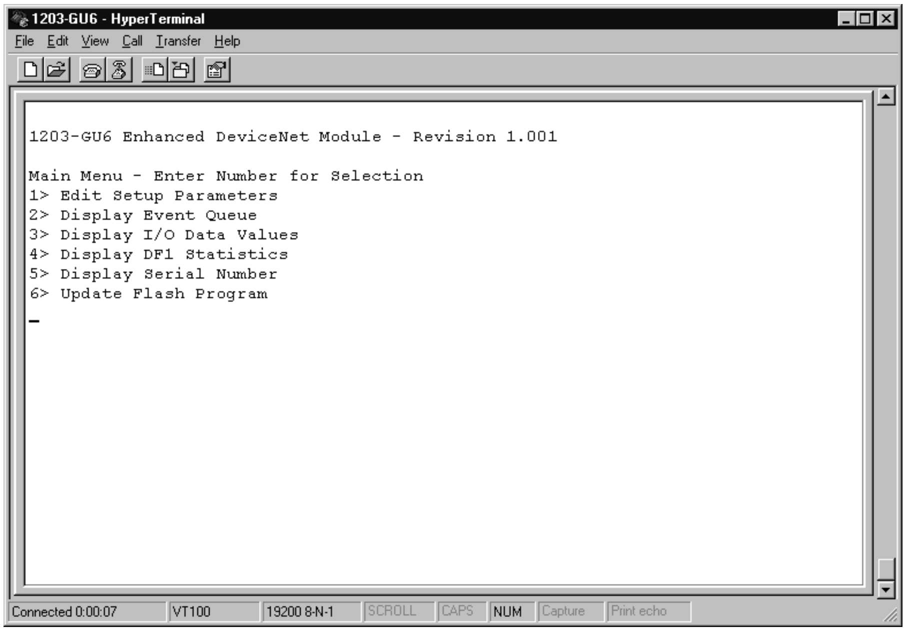
13. In the **Emulation** field, verify **VT100** is selected.

14. Click **OK**.

15. From the **File** menu, select **Save**. The configuration is saved and the icon you selected will appear in the **HyperTerminal** window the next time you make a connection to the module.

16. Press the **Enter** key. The main menu of the Enhanced DeviceNet application appears.

Figure 3.7 Main Menu



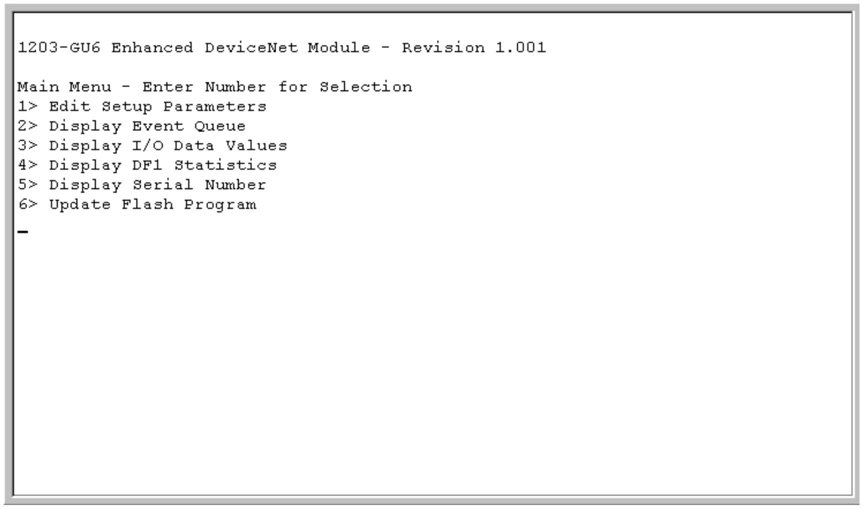
```
1203-GU6 Enhanced DeviceNet Module - Revision 1.001

Main Menu - Enter Number for Selection
1> Edit Setup Parameters
2> Display Event Queue
3> Display I/O Data Values
4> Display DF1 Statistics
5> Display Serial Number
6> Update Flash Program
_
```

Using a VT100-Compatible Terminal

The following instructions describe how to establish a serial connection to the module using a VT100-compatible terminal.

1. Connect a 1203-SFC serial cable to your terminal and then to the serial port on the module. See Figure 3.1.
2. Start your terminal.
3. Select the following settings:
   - **9600** in the **Bits per second** field.
     If you've already set the module's *Serial Port Rate* parameter to enable 19200 bps, set the bps to 19200 in this field.
   - **8** in the **Data bits** field.
   - **None** in the **Parity** field.
   - **1** in the **Stop bits** field.
   - **None** in the **Flow Control** field.
4. Press the **Enter** key. The main menu of the Enhanced DeviceNet application appears.

Figure 3.8 Main Menu



```
1203-GU6 Enhanced DeviceNet Module - Revision 1.001

Main Menu - Enter Number for Selection
1> Edit Setup Parameters
2> Display Event Queue
3> Display I/O Data Values
4> Display DF1 Statistics
5> Display Serial Number
6> Update Flash Program
_
```

Navigation Techniques

To perform any of the functions of the module's software (e.g., editing parameters), you need to know the following navigation techniques:

Press:	To
1, 2, 3, 4, 5, 6, 7, 8, 9, 0	Select an option in the Main Menu (1 – 6) or enter a value for a parameter in the parameter screen (0 – 9).
Escape	Return to Main Menu or abort changes to a parameter.
Down Arrow	View the next parameter.
Up Arrow	View the previous parameter.
Right Arrow	View the next value for a parameter.[①]
Left Arrow	View the previous value for a parameter.[①]
Enter	Save a value for a parameter.

[①] In some parameters the right and left arrow keys let you navigate through the bits.

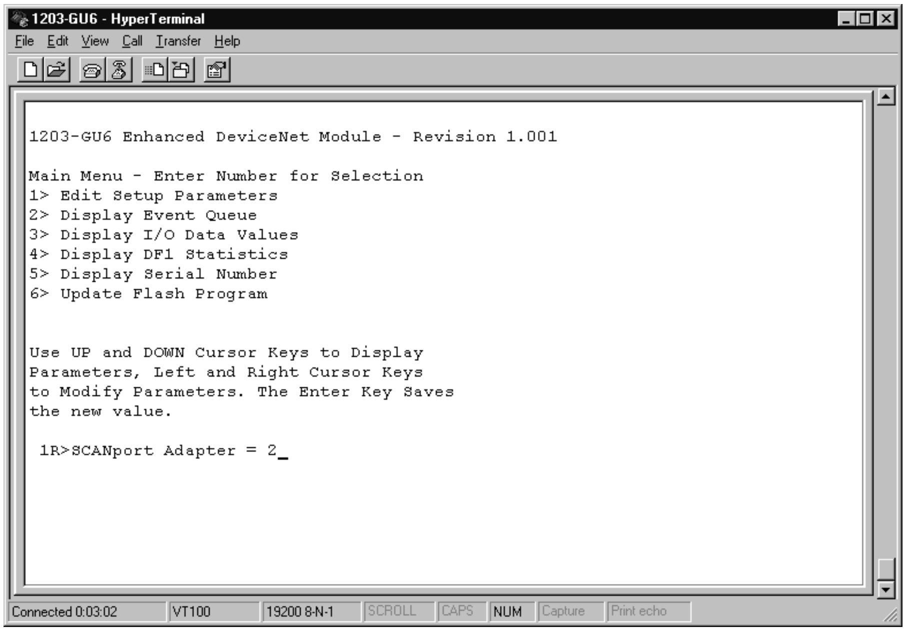
Editing the Module's Parameters

If you do not want to use the module's default settings, you must edit its parameters. Refer to Appendix B, *Enhanced DeviceNet Adapter's Parameters*, for a detailed list of parameters and how to configure them.

1. Establish a serial connection to access the module's software. Refer to the Establishing a Serial Connection to the Module section earlier in this chapter.

2. Press **1** to select **1> Edit Setup Parameters**. The first parameter appears on the bottom of the screen.

Figure 3.9 Edit Parameters



```
1203-GU6 Enhanced DeviceNet Module - Revision 1.001

Main Menu - Enter Number for Selection
1> Edit Setup Parameters
2> Display Event Queue
3> Display I/O Data Values
4> Display DF1 Statistics
5> Display Serial Number
6> Update Flash Program


Use UP and DOWN Cursor Keys to Display
Parameters, Left and Right Cursor Keys
to Modify Parameters. The Enter Key Saves
the new value.

 1R>SCANport Adapter = 2_
```

3. If necessary, scroll through the list of parameters by pressing the **Up arrow** key or **Down arrow** key.

4. Edit parameters as necessary using the **left arrow** and **right arrow** keys. Refer to the “Navigation Techniques” section earlier in this chapter for information on changing parameter values. Refer to Appendix B, *Enhanced DeviceNet Adapter’s Parameters*, for acceptable values for each parameter.

5. If necessary, reset the module by enabling the *Reset Adapter* parameter (22). Refer to Appendix B, *Enhanced DeviceNet Adapter’s Parameters*, to see if the parameter you changed requires the module to be reset in order to take effect.

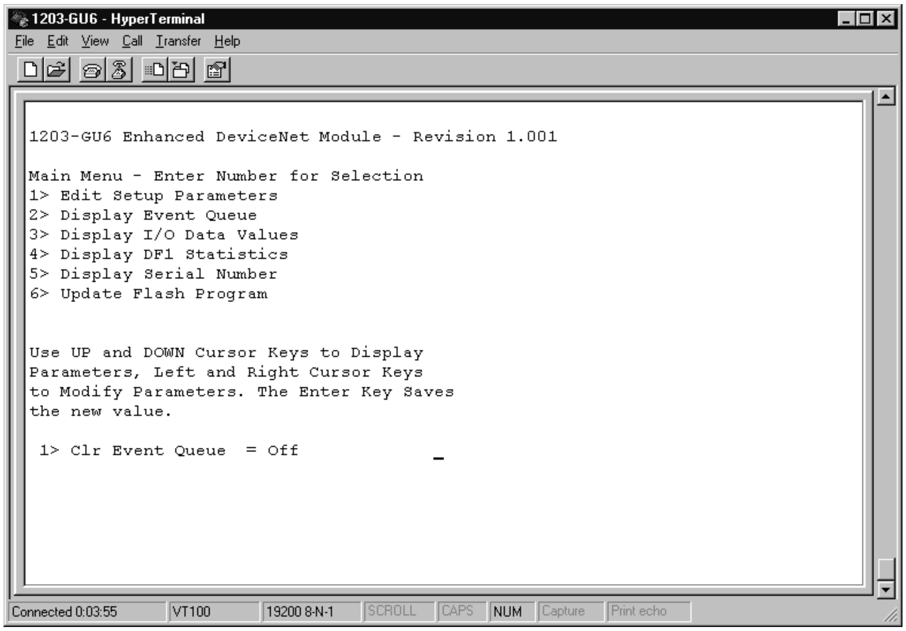
Displaying and Clearing the Module’s Event Queue

If an unexpected problem occurs with the module, you may need to check the event queue to view events that have happened in the module. Follow these instructions:

1. Establish a serial connection to access the module’s software. Refer to the “Establishing a Serial Connection to the Module” section earlier in this chapter.

2. Press **2** to select **2> Display Event Queue**. The event queue appears.

Figure 3.10 The Event Queue



```
1203-GU6 Enhanced DeviceNet Module - Revision 1.001

Main Menu - Enter Number for Selection
1> Edit Setup Parameters
2> Display Event Queue
3> Display I/O Data Values
4> Display DF1 Statistics
5> Display Serial Number
6> Update Flash Program


Use UP and DOWN Cursor Keys to Display
Parameters, Left and Right Cursor Keys
to Modify Parameters. The Enter Key Saves
the new value.

 1> Clr Event Queue  = Off
```

3. Scroll through the list of Event Queue parameters by pressing the **Up Arrow** or **Down Arrow** key. Refer to Appendix G, *Event Queue Messages*, for a list of faults.

Number	Name	Description
1	Clr Event Queue	Enable = Clears the event queue. Ready = Leaves the event queue as is.
2 – 7	Event Queue 1 — Event Queue 6	Event in the event queue. Most recent event is listed in Event Queue 1.
8	Flash Upgrades	Number of times the adapter has been flash upgraded.

4. If desired, clear the current fault in the adapter by setting *Clr Event Queue* (1) to **Enable** and pressing the **Enter** key.

Important: The Fault is cleared in the module and a “Clear Fault” event is added to the Event Queue.

5. Press the **Escape** key to return to the main menu.

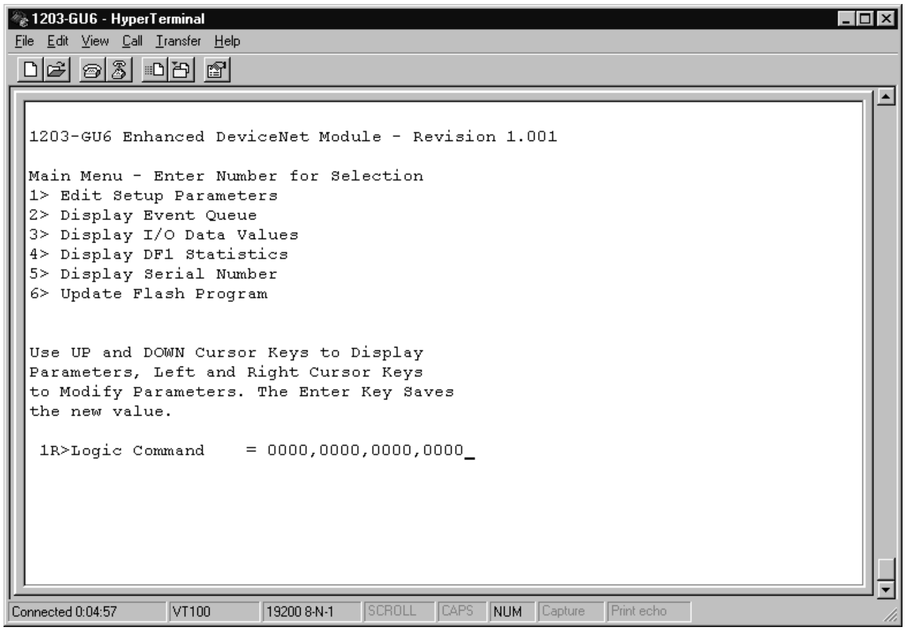
Viewing I/O Data Values

If you need to verify that a command you've sent to a drive is actually passing through the module or to diagnose similar I/O concerns, you can view the I/O data values. To view I/O data values, follow these instructions:

Important: You must understand the configuration of the SCANport product to interpret the I/O data values.

1. Establish a serial connection to access the module's software. Refer to the "Establishing a Serial Connection to the Module" section earlier in this chapter.
2. Press **3** to select **3> Display I/O Data Values**. The **I/O Data Screen** appears.

Figure 3.11 I/O Data Values



```
1203-GU6 Enhanced DeviceNet Module - Revision 1.001

Main Menu - Enter Number for Selection
1> Edit Setup Parameters
2> Display Event Queue
3> Display I/O Data Values
4> Display DF1 Statistics
5> Display Serial Number
6> Update Flash Program


Use UP and DOWN Cursor Keys to Display
Parameters, Left and Right Cursor Keys
to Modify Parameters. The Enter Key Saves
the new value.

 1R>Logic Command    = 0000,0000,0000,0000_
```

3. Scroll through the list of I/O Data parameters by pressing the **Up Arrow** or **Down Arrow** key.

Number	Name	Description
1	Logic Command	Buffer for Logic Command data
2	Logic Status	Buffer for Logic Status data
3	Reference Data	Buffer for Reference data
4	Feedback Data	Buffer for Feedback data
5 – 20	Data A1 In Val — Data D2 Out Val	Data going to (Input) or coming from (Output) the SCANport device

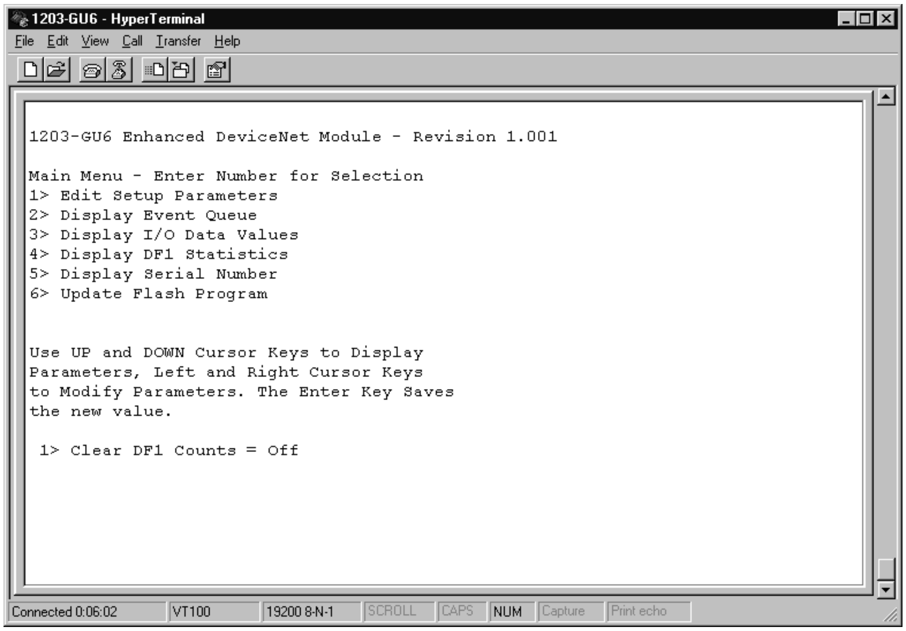
4. Press **Escape** to return to the Main Menu.

Viewing DF-1 Statistics

If you want to check for bad packets or perform other serial port debugging procedures, you can view DF-1 statistics. You may also want to use the DF-1 statistics if you are developing your own DF-1 driver. To view DF-1 statistics, follow these instructions:

1. Establish a serial connection to access the module's software. Refer to the "Establishing a Serial Connection to the Module" section earlier in this chapter.
2. Press **4** to select **4> Display DF1 Statistics**. The first parameter appears.

Figure 3.12 DF1 Statistics



```
1203-GU6 Enhanced DeviceNet Module - Revision 1.001

Main Menu - Enter Number for Selection
1> Edit Setup Parameters
2> Display Event Queue
3> Display I/O Data Values
4> Display DF1 Statistics
5> Display Serial Number
6> Update Flash Program


Use UP and DOWN Cursor Keys to Display
Parameters, Left and Right Cursor Keys
to Modify Parameters. The Enter Key Saves
the new value.

 1> Clear DF1 Counts = Off
```

3. Scroll through the list of DF1 Statistics by pressing the **Up Arrow** or **Down Arrow** key.

Number	Name	Description
1	Clear DF1 Counts	Ready = Leaves status parameters as they are Enable = Reset all status parameters
2	DF1 Packets Sent	Total number of DF1 packets sent by the module
3	DF1 Packets Rcvd	Total number of DF1 packets received by the module
4	Undelivered Msgs	Total number of messages sent that were not acknowledged
5	ENQ Sent	Total number of inquiries sent by the module
6	ENQ Rcvd	Total number of inquiries received by the module
7	NAKs Received	Total number of NAKs received by the module
8	NAK Bad Packet	Total number of NAKs sent by the module because of corrupt data
9	NAK No Memory	Total number of NAKs sent by the module because the previous command did not yet complete and there was no place to save the new command
10	Duplicate Msgs	Total number of messages received by the module with the same TNS number as the previous message

4. If desired, reset the current DF1 protocol statistics by setting *Clear DF1 Counts* to **Enable** and pressing the **Enter** key.
5. Press **Escape** to return to the Main Menu.

Viewing Your Module’s Serial Number

Each Enhanced DeviceNet module has a unique serial number. To view the serial number, follow these instructions:

1. Establish a serial connection to access the module’s software. Refer to the Establishing a Serial Connection to the Module section earlier in this chapter.
2. Press **5** to select **5> Display Serial Number**. The serial number for your communications module appears.
3. Press **Escape** to return to the Main Menu.

Performing a Flash Upgrade to the Module

You can upgrade the Enhanced DeviceNet adapter's firmware using the module's serial port. To ensure a successful flash, we recommend the following:

- If using a laptop, turn off the FIFO buffers in HyperTerminal. In Windows 95, select **File** and then **Properties**. In the **Properties** dialog box, click **Configure** and then click **Advanced**. Uncheck **Use FIFO buffers.**
- Flash the upgrade file from a local hard disk (not a network drive or floppy disk).
- Run only your terminal emulation software while performing the flash.
- Disable the screen saver.

Important: To perform a flash upgrade to your module's firmware through the serial port, you must use a PC running terminal emulation software that supports an X-modem CRC binary transfer.

Important: When you request a flash upgrade, the SCANport product may fault and stop. To exit the flash upgrade option **before** the download has completed, simultaneously press the **Control** and **X** keys.

To perform a flash upgrade, you need to:

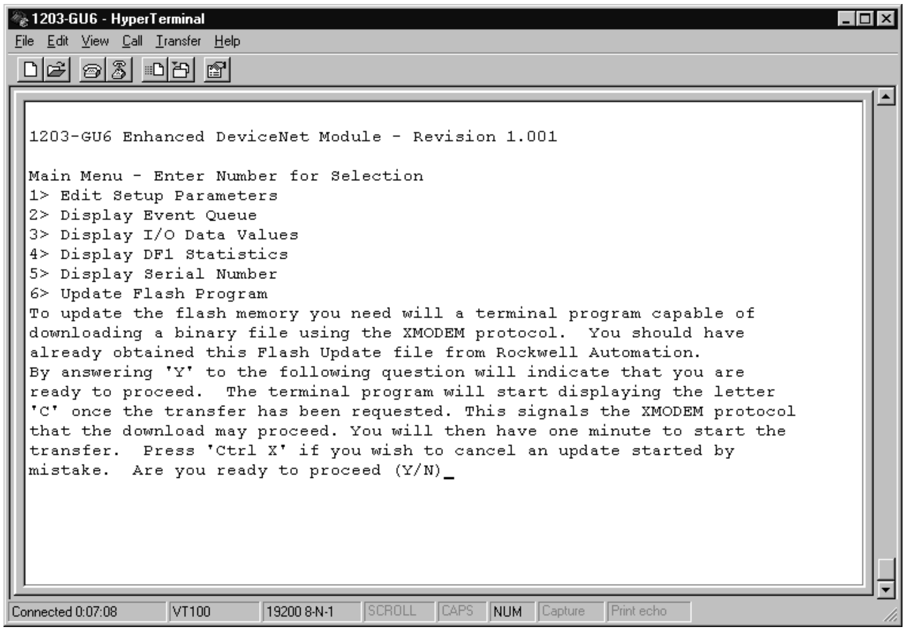
1. Establish a serial connection to access the module's software. Refer to the "Establishing a Serial Connection to the Module" section earlier in this chapter.

2. Obtain the software file that contains the upgrade and record its location. Note the firmware/revision number above the Main Menu options in order to verify the upgrade later.

3. Press **6** to select **6> Update Flash Program**. The following screen appears in terminal mode.

⚠ **ATTENTION:** Hazard of personal injury or death exists when stopping a drive to perform a flash upgrade. When you perform a flash upgrade, the drive will fault and stop the motor if the drive is receiving control data from the 1203-GU6 module. Make sure the motor will stop safely or the drive will receive control data from an alternate source before beginning a flash upgrade.

Figure 3.13 Initial Update Flash Program Screen

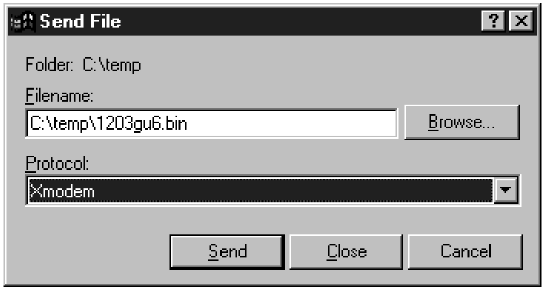


```
1203-GU6 Enhanced DeviceNet Module - Revision 1.001

Main Menu - Enter Number for Selection
1> Edit Setup Parameters
2> Display Event Queue
3> Display I/O Data Values
4> Display DF1 Statistics
5> Display Serial Number
6> Update Flash Program
To update the flash memory you need will a terminal program capable of
downloading a binary file using the XMODEM protocol.  You should have
already obtained this Flash Update file from Rockwell Automation.
By answering 'Y' to the following question will indicate that you are
ready to proceed.  The terminal program will start displaying the letter
'C' once the transfer has been requested. This signals the XMODEM protocol
that the download may proceed. You will then have one minute to start the
transfer.  Press 'Ctrl X' if you wish to cancel an update started by
mistake.  Are you ready to proceed (Y/N)_
```

4. Press **Y** to verify that you want to perform a flash upgrade when prompted.

5. From the **Transfer** menu, select **Send File**. The **Send File** dialog box appears.

Figure 3.14 Send File Dialog Box



Folder: C:\temp

Filename: C:\temp\1203gu6.bin

Protocol: Xmodem

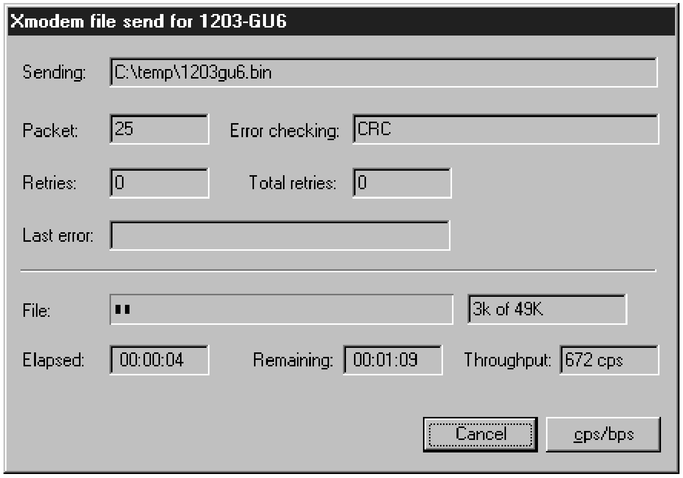
6. In the **Filename** field, select the file that contains the flash upgrade.

Important: You can click the **Browse** button to locate the file that contains the flash upgrade.

7. In the **Protocol** field, select **Xmodem**.

8. Click **Send**. A dialog box appears to report the flash is in progress.

Figure 3.15 Flash In Progress Dialog Box



Sending: C:\temp\1203gu6.bin

Packet: 25 Error checking: CRC

Retries: 0 Total retries: 0

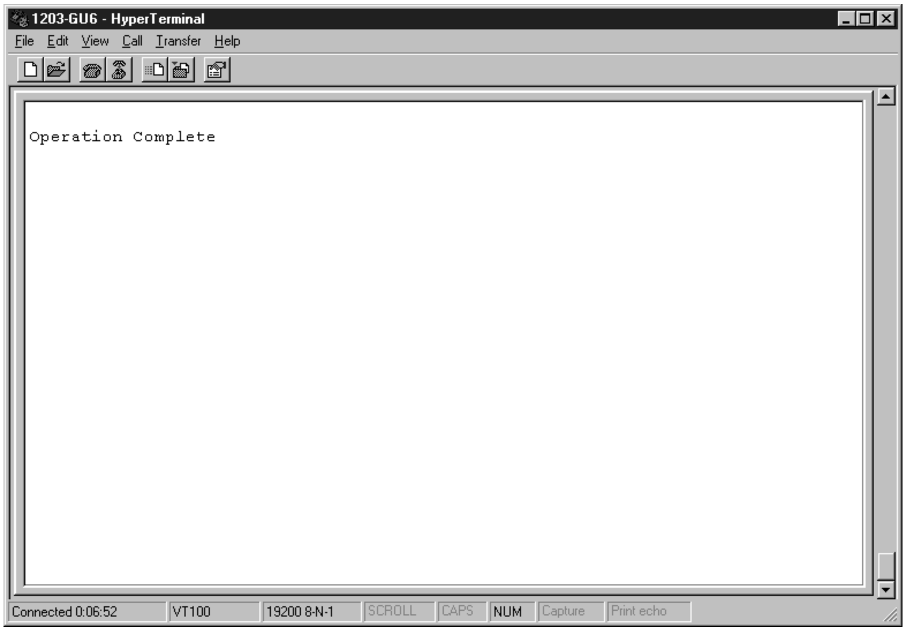
Last error:

File: 3k of 49K

Elapsed: 00:00:04 Remaining: 00:01:09 Throughput: 672 cps

When the flash is complete, a message appears to tell you the download is complete.

Figure 3.16 Message Reporting the Flash is Complete



```
Operation Complete
```

9. Press **Enter** to return to the main menu.

10. Verify that the new main menu displays the new revision data.

Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet

Chapter Objectives

Chapter 4 provides information that you need to configure the 1203-GU6 module or 1336-GM6 board over the DeviceNet network. In this chapter, you will read about the following:

- RSNetWorx for DeviceNet software.
- Equipment necessary to use RSNetWorx for DeviceNet software.
- Editing the 1203-GU6 or 1336-GM6 adapter's parameters using RSNetWorx for DeviceNet.

Important: You must use RSNetWorx for DeviceNet to configure the 1336-GM6 board. The 1203-GU6 module can be configured using RSNetWorx for DeviceNet, DriveExplorer, or terminal emulation software (Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*).

Important: Refer to Appendix B, *Enhanced DeviceNet Adapter's Parameters*, for information on changing the node address or data rate.

What is RSNetWorx for DeviceNet?

RSNetWorx for DeviceNet is a Rockwell Software application that can be used to set up DeviceNet networks and configure connected devices. RSNetWorx for DeviceNet (version 2.22.18) and RSLinx (version 2.10.118) were used for examples in the manual. Different versions may differ in appearance and procedures.

After installing or mounting the adapter, you can use RSNetWorx for DeviceNet to configure or edit the adapter's parameters.

Required Equipment and Software

Before configuring or editing your adapter's parameters, your PC must be:

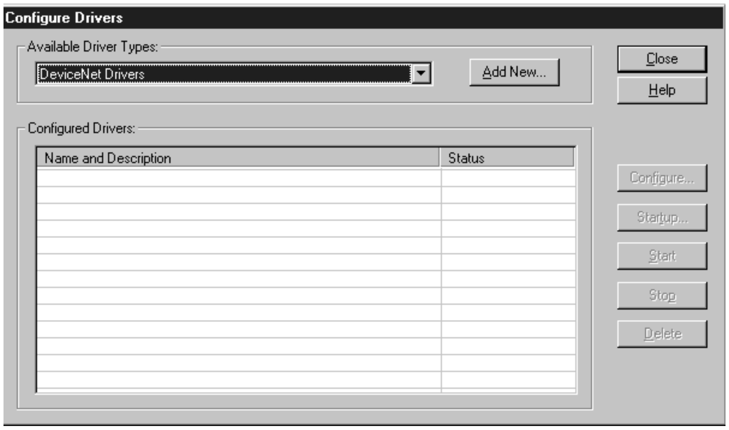
- Running RSNetWorx for DeviceNet. Refer to http://www.software.rockwell.com for more information on this product.
- Connected to and communicating with the DeviceNet network using a DeviceNet interface such as a 1784-PCD card or a 1770-KFD adapter.

Going Online

To use RSNetWorx for DeviceNet, you must first set up a driver in RSLinx. The driver provides a communications link between the computer and DeviceNet network. Then, you can view the devices on a DeviceNet network by going online. A device may appear as an unrecognized device if RSNetWorx for DeviceNet does not have an EDS file for it.

1. Start RSLinx, and select **Communications > Configure Drivers** to display the Configure Drivers dialog box.

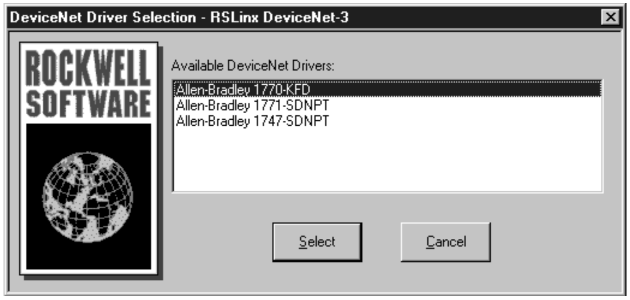
Figure 4.1 Configure Drivers Dialog Box


Configure Drivers

Available Driver Types: DeviceNet Drivers

2. In the Available Driver Types box, select **DeviceNet Drivers**, and then click **Add New**. The DeviceNet Driver Selection dialog box appears.

Figure 4.2 DeviceNet Driver Selection Dialog Box


DeviceNet Driver Selection - RSLinx DeviceNet-3

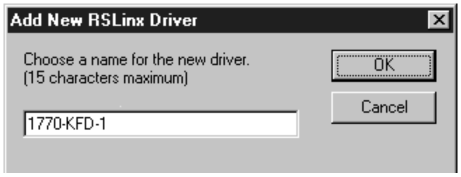
Available DeviceNet Drivers:

Allen-Bradley 1770-KFD
Allen-Bradley 1771-SDNPT
Allen-Bradley 1747-SDNPT

3. In the Available DeviceNet Drivers list, select the adapter connected to your computer, and then click **Select**. A Driver Configuration dialog box appears.

4. Configure the driver for your computer and network settings, and then click **OK**. The Configure Drivers dialog box reports the progress of the configuration. The Add New RSLinx Driver dialog box appears.

Figure 4.3 Add New RSLinx Driver Dialog Box

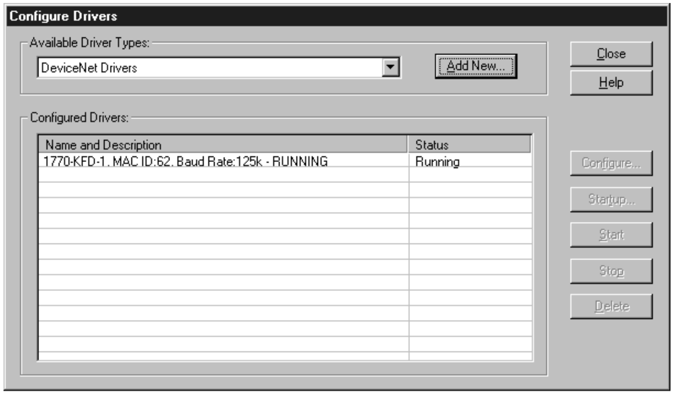

Add New RSLinx Driver

Choose a name for the new driver.
(15 characters maximum)

1770-KFD-1

5. Type a name (if desired), and then click **OK**. The Configure Drivers dialog box reappears, and the new driver is in the Configured Drivers List

Figure 4.4 Configure Drivers Dialog Box with a DeviceNet Driver



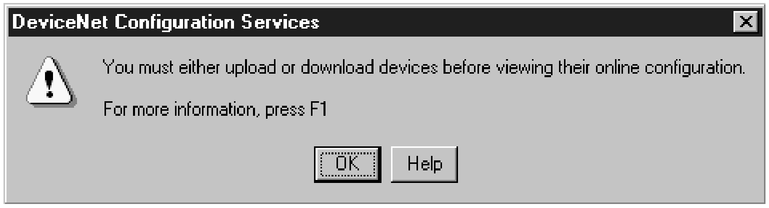
Available Driver Types: DeviceNet Drivers

Name and Description	Status
1770-KFD-1, MAC ID:62, Baud Rate:125k - RUNNING	Running

6. Click **Close** to close the dialog box. Leave RSLinx running.

7. Start RSNetWorx for DeviceNet, and then select **Network > Online**. If the Browse for Network dialog box appears, RSLinx has drivers for multiple networks. Select your DeviceNet network, and click **OK**. A message appears.

Figure 4.5 DeviceNet Configuration Services Message

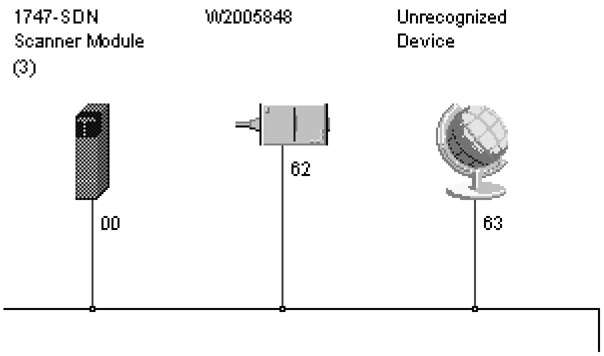


You must either upload or download devices before viewing their online configuration.

For more information, press F1

8. Click **OK** to go online. The devices in the network appear in the Configuration view.

Figure 4.6 Online in RSNetWorx for DeviceNet

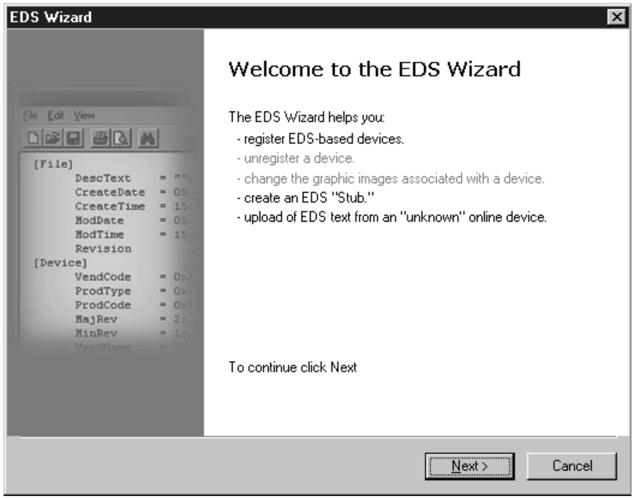


Creating an EDS File

If the adapter and SCANport product appear as an unrecognized device, create an EDS file.

1. Right-click the “Unrecognized Device” icon (for example, node 63 in Figure 4.6), and select **Register Device** in the menu. The EDS Installation wizard appears.

Figure 4.7 EDS Wizard



EDS Wizard

Welcome to the EDS Wizard

The EDS Wizard helps you:
- register EDS-based devices.
- unregister a device.
- change the graphic images associated with a device.
- create an EDS "Stub."
- upload of EDS text from an "unknown" online device.

To continue click Next

2. Click **Next** to display the next step.
3. Select **Upload EDS**, and then click **Next**.
4. Type a description (if desired), and then click **Next**.
5. Under Polled, select **Enabled**, type **4** in the Input Size and Output Size boxes, and then click **Next**. RSNetWorx will upload the EDS file from the product and adapter.
6. Click **Next** to select an icon for the node. We recommend that you use the icon for your product. You can change icons by clicking **Change icon**.
7. Click **Next** to view a summary, and then click **Next** again to accept it.
8. Click **Finish** to finish the EDS creation. A new icon represents the SCANport product and adapter in the Configuration View.

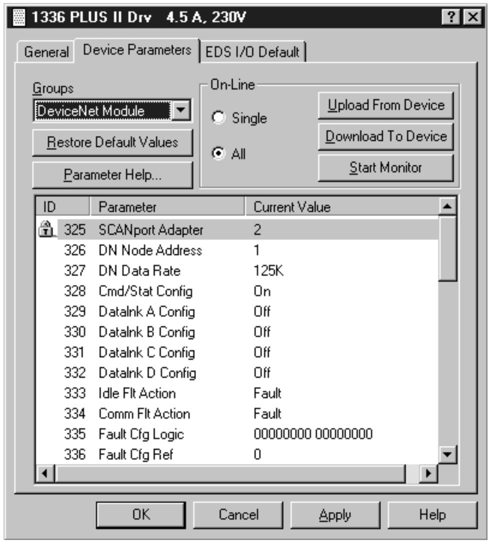
Accessing and Editing Parameters

Parameters in the SCANport product and adapter can be edited with RSNetWorx. The adapter parameters are appended to the list of product parameters. In Figure 4.8, for example, the drive has 323 parameters, so parameter 324 is the first adapter parameter.

1. After creating an EDS file, right-click on the icon for the SCANport product and adapter and select **Properties**. The Device Edit dialog box appears.

2. Click the **Device Parameters** tab. If an EDS Editor message appears, click **Upload** to load the parameter values in the product to the computer.

Figure 4.8 Device Parameters Page in the Edit Dialog Box



Groups: DeviceNet Module

On-Line: ○ Single ● All

ID	Parameter	Current Value
325	SCANport Adapter	2
326	DN Node Address	1
327	DN Data Rate	125K
328	Cmd/Stat Config	On
329	Datalnk A Config	Off
330	Datalnk B Config	Off
331	Datalnk C Config	Off
332	Datalnk D Config	Off
333	Idle Flt Action	Fault
334	Comm Flt Action	Fault
335	Fault Cfg Logic	00000000 00000000
336	Fault Cfg Ref	0

Parameters are displayed in numerical order under Parameter. Parameters for the DeviceNet adapter are appended to the end of the parameter list for the SCANport product. You can either scroll through the list or select a specific group of parameters in the Groups box (for example, DeviceNet Module). The available groups and the numbers of the adapter parameters will vary based on the type of product that is connected to the adapter.

3. In the Current Value column, double-click a value to edit it.

Notes:

Configuring a Scanner to Communicate with the Adapter

Chapter Objectives

Chapter 5 provides instructions for configuring your scanner to communicate with either the 1203-GU6 module or 1336-GM6 board. This allows the product connected to the adapter to be an active node on the DeviceNet network. In this chapter, you will read about the following:

- Equipment and software needed for the configuration.
- Configuring a PLC or SLC scanner to communicate with the adapter.

Required Equipment and Software

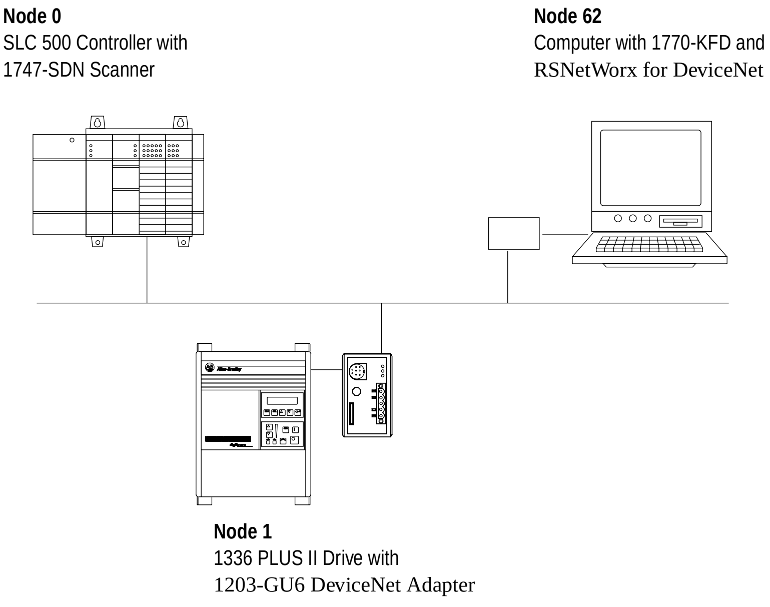
Before configuring the scanner, your PC must be:

- Running RSNetWorx for DeviceNet. Refer to http://www.software.rockwell.com for more information.
- Connected to and communicating with the DeviceNet network using a DeviceNet interface such as a 1784-PCD card or a 1770-KFD adapter.

Example DeviceNet Network

After the adapter is configured, it and the connected product will be a single node on the network. This chapter provides the steps that are needed to configure a simple network like the network in Figure 5.1.

Figure 5.1 **Example DeviceNet Network**

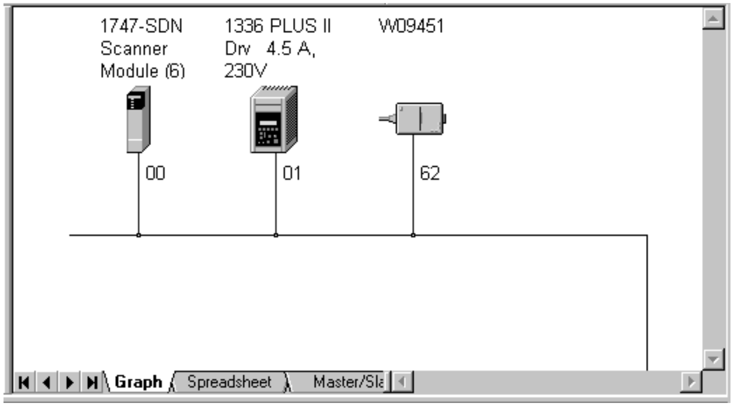


Setting Up the Scan List

For the scanner to communicate with a product, the scanner must be configured and the product's node number must be added to its scan list.

1. Go online with RSNetWorx for DeviceNet. Refer to the "Going Online" section in Chapter 4.
2. Select **Network > Single Browse Path**. The devices on the network are displayed in the configuration view.

Figure 5.2 Configuration View (Graph)

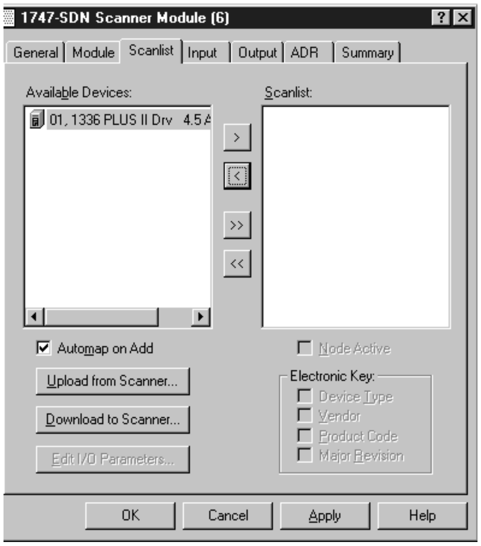


3. Right-click the DeviceNet scanner (node 00 in Figure 5.2) and select **Properties**. The Scanner Module dialog box appears.

Important: If your scanner is an unrecognized device, you must create an EDS file for it and then configure it. Create an EDS file by following the instructions in the "Creating an EDS File" section in Chapter 4. Configure the scanner using the General and Module tabs. Click **Help** or refer to your scanner documentation if you need more information.

4. Click the **Scanlist** tab. A message box prompts you to upload.
5. Click **Upload**. Data is uploaded from the scanner, and then the Scanlist page appears.

Figure 5.3 Scanlist Page in the Device Edit Dialog Box



Available Devices: 01, 1336 PLUS II Drv 4.5 A

☑ Automap on Add

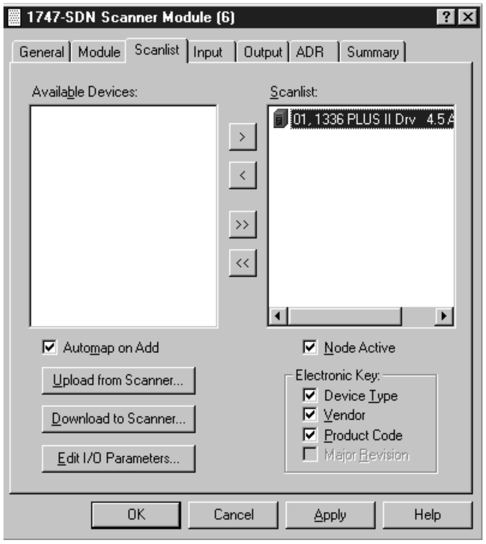
□ Node Active

□ Device Type
□ Vendor
□ Product Code
□ Major Revision

6. Select the **Automap on Add** box (a check mark will appear).

7. Under Available Devices, select the SCANport product, and then click > (Right Arrow) to add it to the scanlist.

Figure 5.4 Scanlist Page in the Scanner Module Dialog Box



Scanlist: 01, 1336 PLUS II Drv 4.5 A

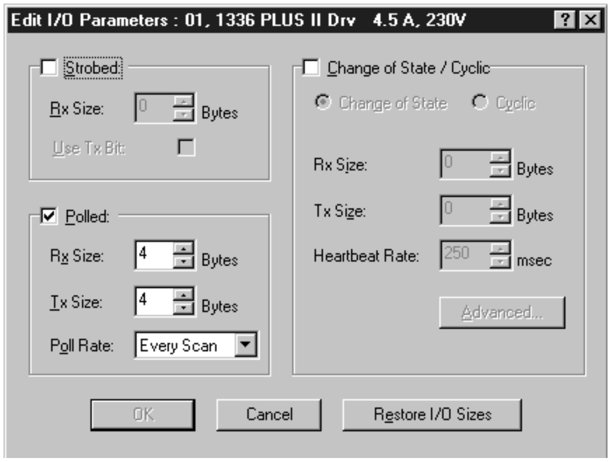
☑ Automap on Add

☑ Node Active

☑ Device Type
☑ Vendor
☑ Product Code
□ Major Revision

8. Under Scanlist, select the SCANport product, and then click **Edit I/O Parameters**. The Edit I/O Parameters dialog box appears.

Figure 5.5 Edit I/O Parameters Dialog Box



Edit I/O Parameters : 01, 1336 PLUS II Drv 4.5 A, 230V

□ Strobed:
Rx Size: 0 Bytes
Use Tx Bit: □

☑ Polled:
Rx Size: 4 Bytes
Tx Size: 4 Bytes
Poll Rate: Every Scan

□ Change of State / Cyclic
● Change of State ○ Cyclic
Rx Size: 0 Bytes
Tx Size: 0 Bytes
Heartbeat Rate: 250 msec

9. Select the type(s) of data exchange (Polled, Change of State, and/or Cyclic). In our example, we selected Polled.

10. Type the number of bytes that will be required for your I/O in the Rx Size and Tx Size. The size will depend on the I/O that you enabled in the adapter. The I/O is set using the *Cmd/Stat Config* (Parameter 4) and the *Datalink x Cfg* (Parameters 5 – 8) in the adapter.

In our example, we typed 4 in the Rx Size and Tx Size boxes because we have enabled only the *Cmd/Stat Config* (Parameter 4) for I/O in the adapter. In the Output image, our product uses one 16-bit word for the logic command and one 16-bit word for the reference. A 16-bit word is two bytes. In the Input image, our product uses one 16-bit word for logic status and one 16-bit word for feedback. Therefore, the logic command/status uses 2 bytes and the reference/feedback uses 2 bytes, totaling 4 bytes.

11. Set the scan rate:

Data Exchange	Rate to set
Polled	Polled Rate
Change of State	Heartbeat Rate
Cyclic	Send Rate

Click **Help** for more information.

12. Click **OK**. If you changed any settings, a Scanner Applet appears and asks if it is OK to unmap the I/O. Click **Yes** to continue. The Edit I/O Parameters dialog box closes and then the Scanner Module dialog box (Figure 5.4) reappears. You will map the I/O in the next section in this chapter.

Mapping the Product’s Data in the Scanner

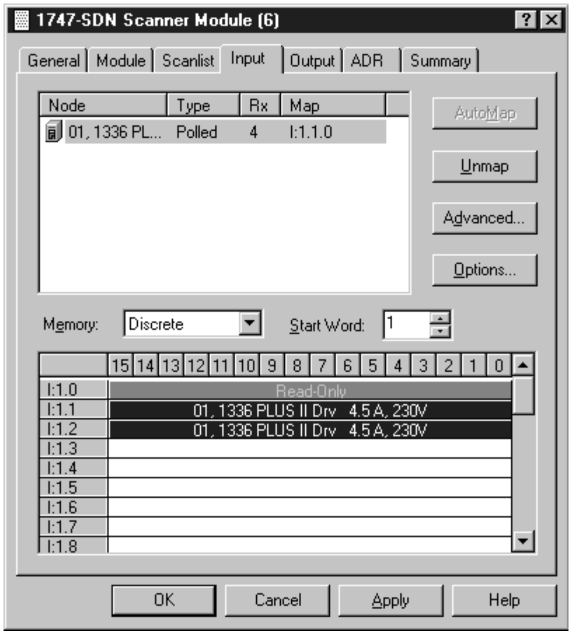
Data from I/O messages must be mapped in the scanner. This mapping determines where a ladder logic program can find data transmitted on the network. You must map the following:

For:	Refer to:
Mapping the Input I/O	page 5-5
Mapping the Output I/O	page 5-6

Mapping the Input I/O

1. In the Scanner Module dialog box, click the **Input** tab. (If you need to display this dialog box, right-click the scanner in the configuration view. See Figure 5.2.)

Figure 5.6 Input Page on the Scanner Module Dialog Box



1747-SDN Scanner Module (6)

Node	Type	Rx	Map
01, 1336 PL...	Polled	4	I:1.1.0

Memory: Discrete

Start Word: 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I:1.0	Read-Only															
I:1.1	01, 1336 PLUS II Drv 4.5 A, 230V															
I:1.2	01, 1336 PLUS II Drv 4.5 A, 230V															
I:1.3																
I:1.4																
I:1.5																
I:1.6																
I:1.7																
I:1.8																

If you selected the **Automap on Add** box (Figure 5.3) in the Scanlist page (Figure 5.4) and did not change any settings, RSNetWorx has already mapped the I/O. If I/O is not mapped, click **AutoMap** to map it. If you need to change the mapping, click **Advanced** and change the settings. Click **Help** for assistance.

2. In the Memory box, select a location in scanner memory.

Scanner	Memory Locations
1747-SDN	Discrete or M-File
1771-SDN	Block Xfer 62 – 57

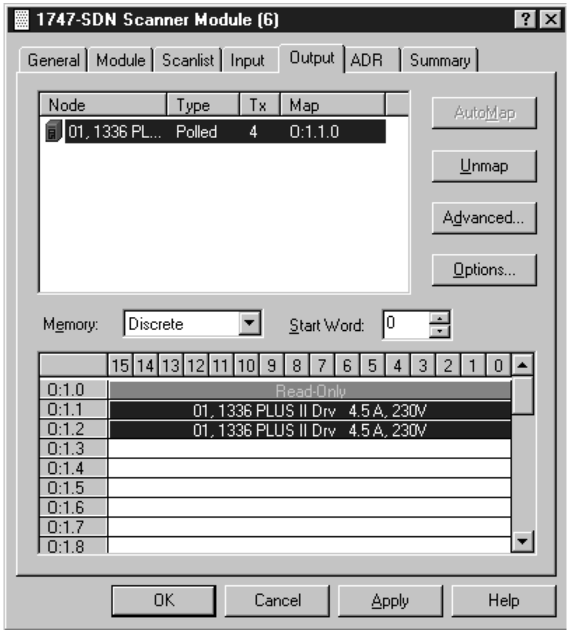
In our example, we are using a 1747-SDN and selected Discrete.

3. In the Start Word box, select the word in memory at which the data should start. In our example, it is 1.

Mapping the Output I/O

1. In the Scanner Module dialog box, click the **Output** tab. (To display this dialog box, right-click the scanner in the configuration view. See Figure 5.2.)

Figure 5.7 Output Page on the Scanner Module Dialog Box



Node	Type	Tx	Map
01, 1336 PL...	Polled	4	0:1.1.0

Memory: Discrete

Start Word: 0

	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0:1.0	Read-Only
0:1.1	01, 1336 PLUS II Drv 4.5 A, 230V
0:1.2	01, 1336 PLUS II Drv 4.5 A, 230V
0:1.3	
0:1.4	
0:1.5	
0:1.6	
0:1.7	
0:1.8	

If you selected the **Automap on Add** box (Figure 5.3) in the Scanlist page (Figure 5.4) and did not change any settings, RSNetWorx has already mapped the I/O. If I/O is not mapped, click **AutoMap** to map it. If you need to change the mapping, click **Advanced** and change the settings. Click **Help** for assistance.

2. In the Memory box, select a location in scanner memory.

Scanner	Memory Locations
1747-SDN	Discrete or M-File
1771-SDN	Block Xfer 62 – 57

In our example, we are using a 1747-SDN and selected Discrete.

3. In the Start Word box, select the word in memory at which the data should start. In our example, we selected 1.

Saving the Configuration

After configuring a scanner, you must download it to the scanner. You should also save it to a file on your computer.

1. In the Scanner Module dialog box, click **Apply** to save the configuration to the scanner. A Scanner Configuration Applet appears and asks if it is OK to download the changes.
2. Click **Yes** to download the changes. The changes are downloaded and then the Scanner Module dialog box reappears.
3. Click **OK** to close the Scanner Module dialog box.
4. Select **File > Save**. If this is the first time that you saved the project, the Save As dialog box appears. Navigate to a folder, type a file name, and click **Save** to save the configuration to a file.

Notes:

Ladder Logic Programming

Chapter Objectives

Chapter 6 provides information needed to create the PLC or SLC Ladder Logic program that the controller will use to transmit control I/O and messages to and from the SCANport product. In this chapter, you will read about the following:

- Equipment and software needed to create either a PLC or SLC ladder logic program.
- PLC and SLC ladder logic programs.
- Creating a PLC or SLC ladder logic program.
- Example PLC and SLC ladder logic programs to control the drive.

This chapter assumes you are familiar with the hardware components and programming procedures necessary to operate DeviceNet and SCANport devices, including the following:

- PLC-5 or SLC-5/03.
- 1771-SDN or 1747-SDN scanner.
- Ladder programming.
- RSLogix5 or RSLogix500.

You may need to refer to the documentation associated with these products to create a ladder logic program.

Required Equipment

Before creating a ladder logic program for the PLC or SLC, your PC must be:

- Running RSLogix5 and RSLinx if using a PLC. Refer to http://www.software.rockwell.com for more information on these products.
- Running RSLogix500 and RSLinx if using an SLC. Refer to http://www.software.rockwell.com for more information on these products.
- Connected to and communicating with the controller.

What is RSLogix?

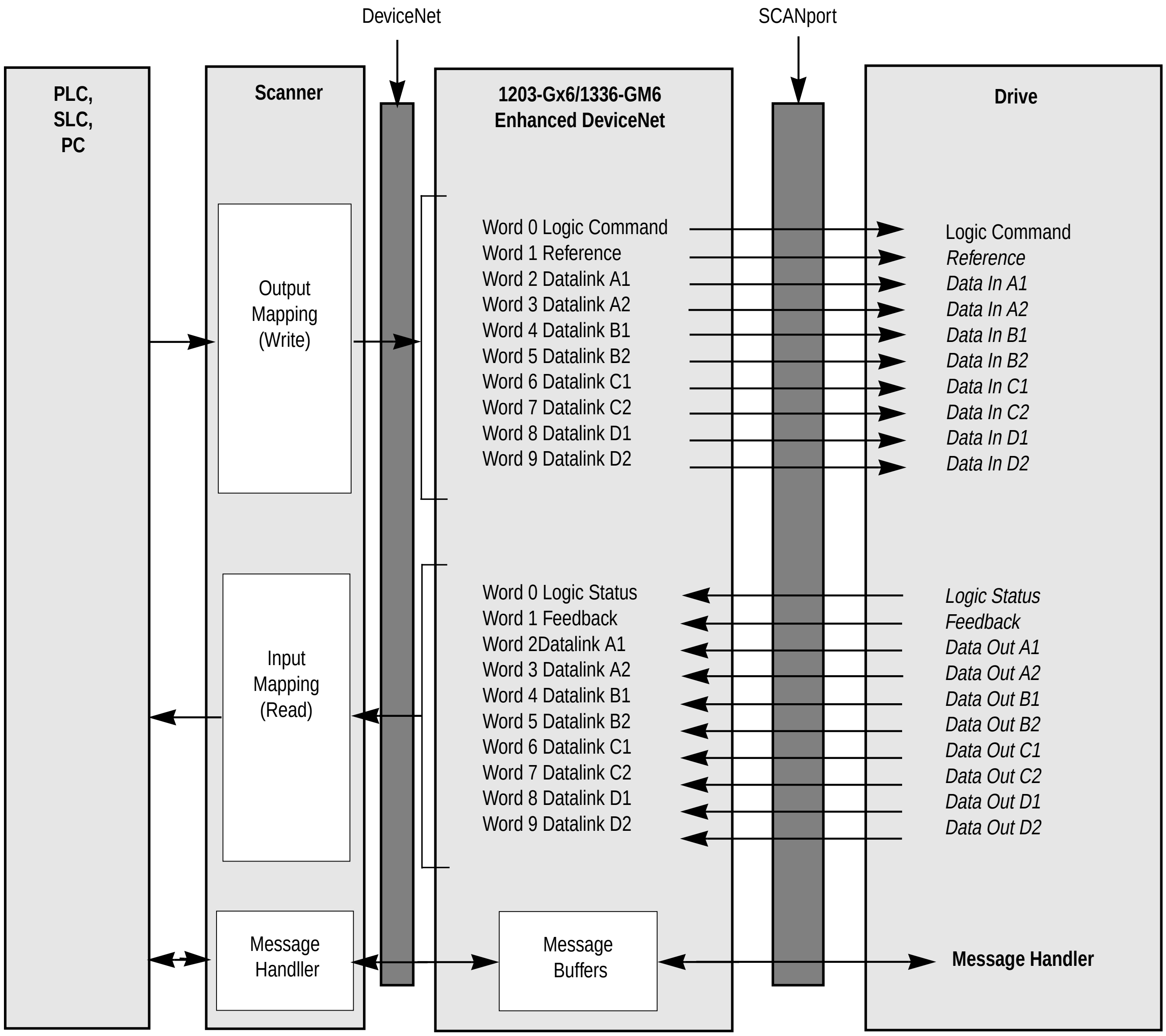
RSLogix5 (for the PLC-5) and RSLogix500 (for the SLC-5/03) software let you create the ladder logic programs you need and download them to the PLC or SLC. They also let you monitor the program as the PLC or SLC is using it.

For more information on RSLogix5 or RSLogix500, consult the respective software's documentation.

What are Ladder Logic Programs?

A PLC or SLC ladder logic program lets you control the drive and the messaging from the PLC or SLC to the drive. Figure 6.1 shows how the I/O image table for a DeviceNet scanner relates to a drive, such as a 1336 PLUS drive, when an Enhanced DeviceNet communications adapter is used. Note that the location of the first word (n) depends on the I/O mapping.

Figure 6.1 I/O Image Table



Important: Datalinks are optionally enabled in the adapter and configured in the product. Refer to Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection* or Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet* and your product's user manual for more information. Examples of reference/feedback include speed, torque, and frequency.

Example Ladder Logic Programs

The following are example ladder logic programs for a 1305, 1336 PLUS, or 1336 PLUS II drive.

ATTENTION: The example ladder logic program shown in this manual is intended solely for purpose of example. Because there are many variables and requirements associated with any particular installation, Rockwell Automation does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the example shown in this publication.

Important: Refer to the user manual for your SCANport product for specific I/O definitions. Different SCANport products have different Logic Command, Logic Status, Reference and Feedback I/O interpretations.

The 1305, 1336 PLUS, or 1336 PLUS II drive in this example accepts the following Logic Command Data from the controller.

Logic Command Bits																Function	Description
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
															X	Stop	1=Stop, 0=No Operation
														X		Start	1=Start, 0=No Operation
													X			Jog	1=Jog, 0=No Operation
												X				Clear Faults	1=Clear, 0=No Operation
										X	X					Direction	00=No Operation, 01=Forward 10=Reverse, 11=Hold Direction
									X							Local	1=Local, 0=Multiplexed
								X								MOP Increment	1=Increment MOP, 0=No Operation
						X	X									Accel Rate Select	00=No Operation, 01=Rate 1 10=Rate 2, 11=Hold Rate
				X	X											Decel Rate Select	00=No Operation, 01=Rate 1 10=Rate 2, 11=Hold Rate
	X	X	X													Reference Selection	000=No Operation 001=External Reference 1 (Par 5) 010=External Reference 2 (Par 6) 011=Preset 3 100=Preset 4 101=Preset 5 110=Preset 6 111=Preset 7
X																MOP Decrement	1=Decrement MOP, 0=No Operation

The 1305, 1336 PLUS, or 1336 PLUS II drive in this example sends the following Logic Status Data to the PLC.

Logic Status Bits																Function	Description
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
															X	Enabled	1=Enabled, 0=Not Enabled
														X		Running	1=Running, 0=Not Running
													X			Command Direction	1=Forward, 0=Reverse
												X				Rotating Direction	1=Forward, 0=Reverse
											X					Acceleration	1=Accelerating, 0=Not
										X						Deceleration	1=Decelerating, 0=Not
									X							Warning	1=Warning Present, 0=Not
								X								Fault	1=Faulted, 0=Not Faulted
							X									At Reference	1=At Speed, 0=Not At Speed
				X	X	X										Local	000=Terminal I/O has Local 001=Port 1 has Local 010=Port 2 has Local 011=Port 3 has Local 100=Port 4 has Local 101=Port 5 has Local 110=Port 6 has Local 111=Multiplexed Control
X	X	X	X													Reference Source	0000=External Reference 1 0001 – 0111=Presets 1 – 7 1000=External Reference 2 1001 – 1110=Port 1 – 6 Direction 1111=Jog

PLC Ladder Logic Example

The following example uses a PLC-5, a 1771-SDN DeviceNet scanner, and a 1203-GU6 to control a 1305, 1336 PLUS, or 1336 PLUS II drive.

The example program shows how to obtain status information from the drive and how to control it (e.g., starting the drive, stopping the drive, jogging the drive, sending reference, and clearing faults). When you understand this example, you should be able to customize the program to fit you application needs.

The example assumes that there is an operator's station wired to an I/O module in slot zero of module group zero of rack zero.

Important: You may want to verify a device has not failed using word 0 of block transfer 62 before sending control data. If a device has failed, use block transfer 52 to find out which device failed. Refer to the *1771-SDN DeviceNet Scanner Module Manual*, Publication 1771-5.14, for more information.

Figure 6.2 Example PLC Ladder Logic Program

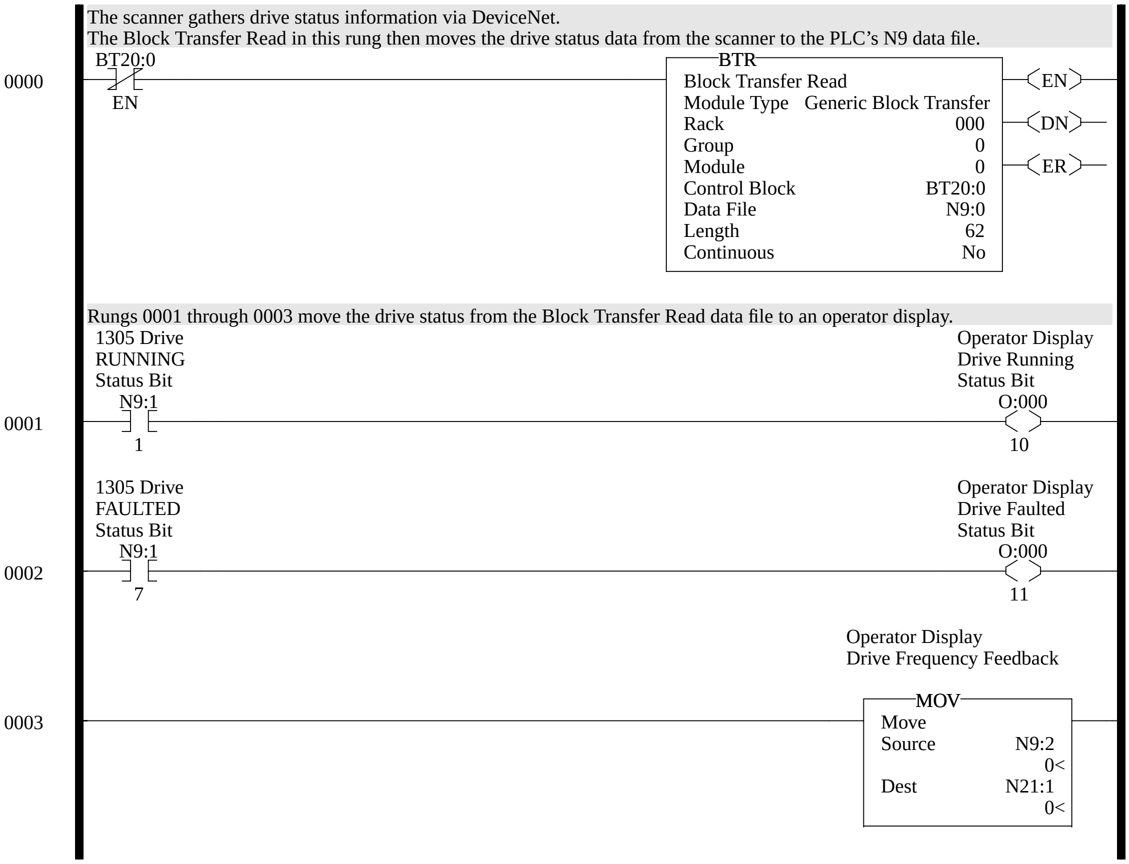
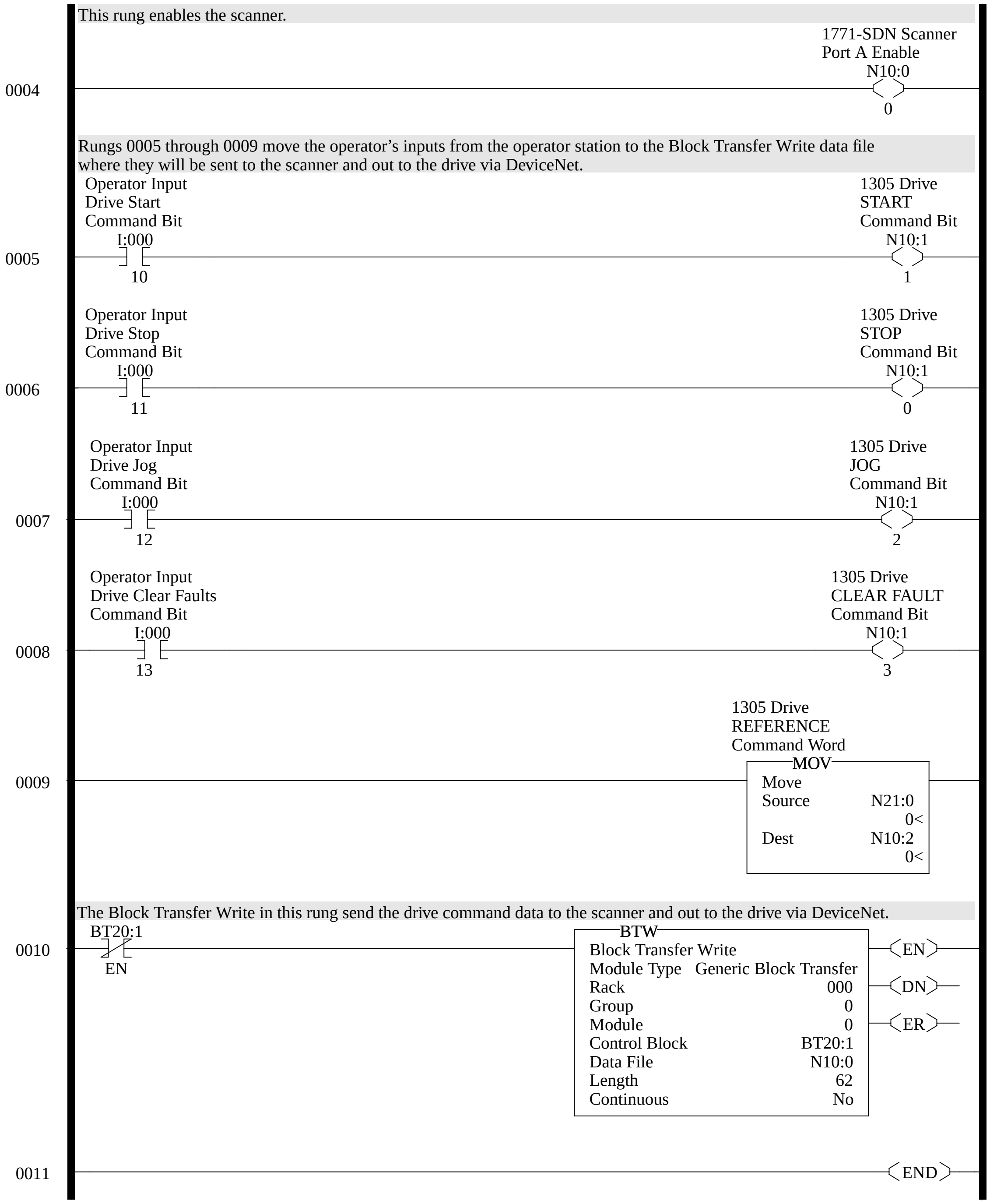


Figure 6.2 Example PLC Ladder Logic Program (Continued)



The following table represents the control file for the block transfers.

Offset	EN	ST	DN	ER	CO	EW	NR	TO	RW	RLEN	DLEN	FILE	ELEM	R	G	S
BT20:0	0	0	0	0	0	0	0	0	0	62	0	9	0	00	0	0
BT20:1	0	0	0	0	0	0	0	0	0	62	0	10	0	00	0	0

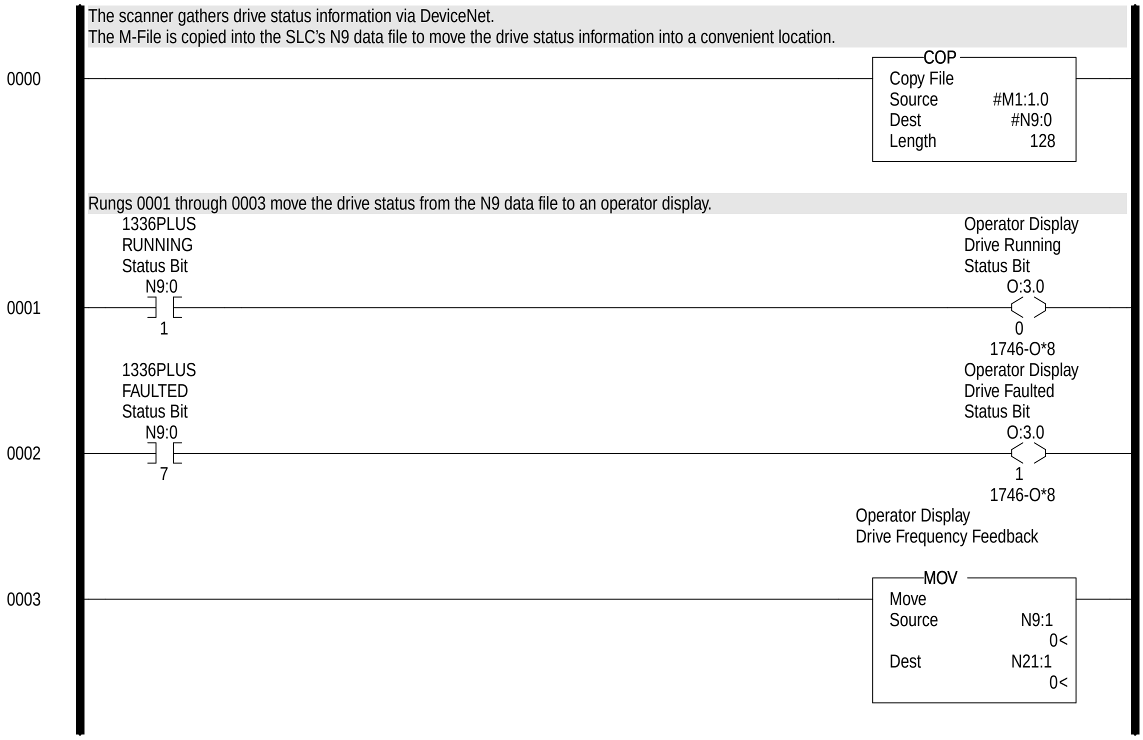
SLC Ladder Logic Program Example

The following example uses a SLC-5/03, a 1747-SDN DeviceNet scanner, and a 1203-GU6 to control a 1336 PLUS, 1336 PLUS II or 1305 drive.

The example assumes that there is an operator's station wired to an I/O module in slot one of module group zero of rack zero.

Important: You may want to verify a device has not failed using word I:S.0. If a device has failed, read the appropriate M1 File to find out which device failed. Refer to the *1747-SDN DeviceNet Scanner Module Manual*, Publication 1747-5.8, for more information.

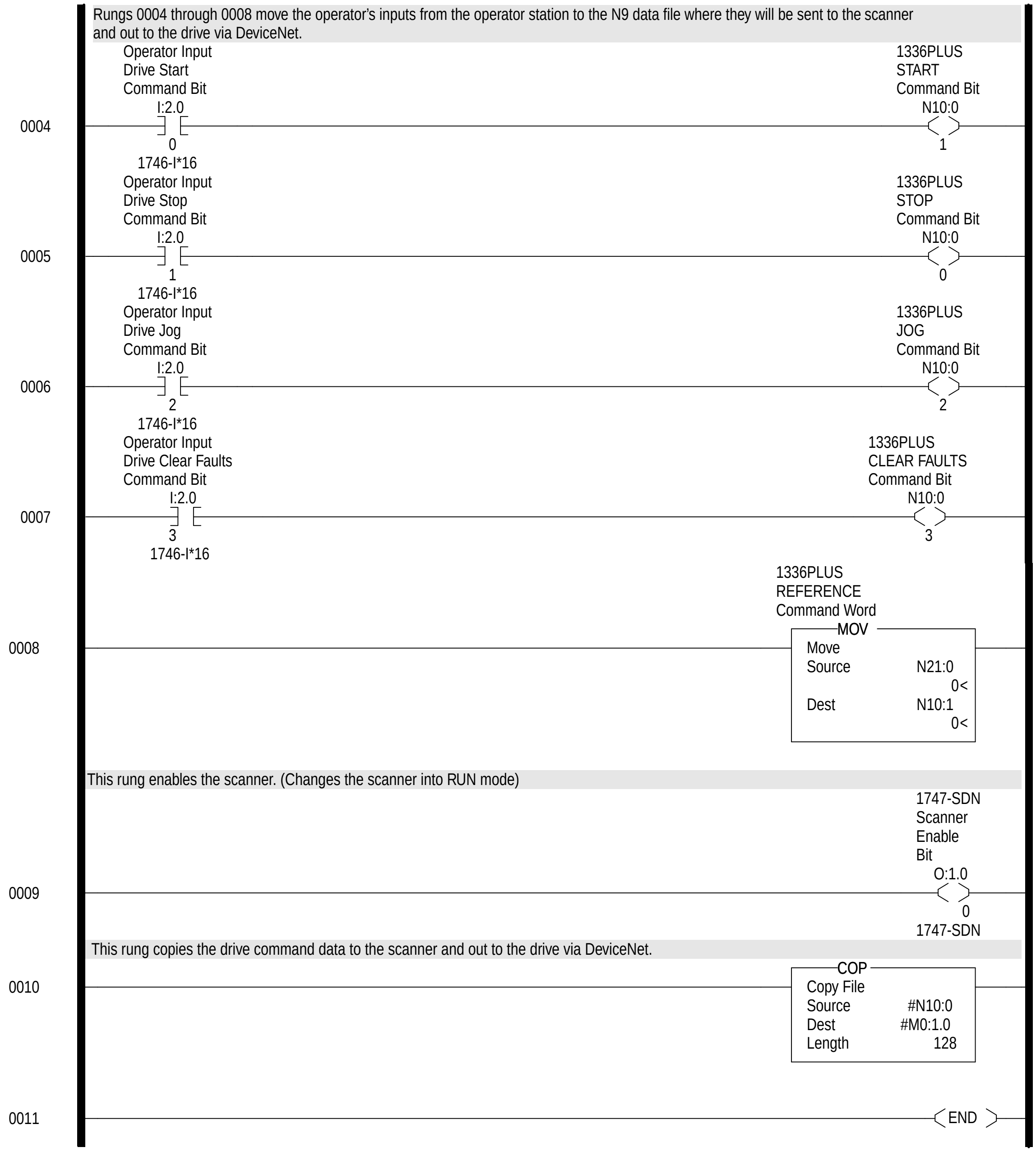
Figure 6.3 **Example SLC Ladder Logic Program**



The scanner gathers drive status information via DeviceNet.
The M-File is copied into the SLC's N9 data file to move the drive status information into a convenient location.

Rungs 0001 through 0003 move the drive status from the N9 data file to an operator display.

Figure 6.3
Example SLC Ladder Logic Program (Continued)



Rungs 0004 through 0008 move the operator's inputs from the operator station to the N9 data file where they will be sent to the scanner and out to the drive via DeviceNet.

This rung enables the scanner. (Changes the scanner into RUN mode)

This rung copies the drive command data to the scanner and out to the drive via DeviceNet.

Chapter **7**

Using DeviceNet Explicit Messaging

Chapter Objectives

Chapter 7 provides information you need to monitor and configure the SCANport device using explicit messaging on DeviceNet. In this chapter, you will read about the following:

- Required equipment.
- Message translations.
- Messaging guidelines for the 1771-SDN scanner.
- Messaging guidelines for the 1747-SDN scanner.
- Example messages.
- Using messages to control SCANport products.
- Writing to register objects.

Refer to Appendix C, *DeviceNet Objects*, for information on object data support.

Required Equipment

Before using messaging, your PC must be:

- Running Logix5 and RSLinx if you are using a PLC. Refer to http://www.software.rockwell.com for more information on these products.
- Running RSLogix500 and RSLinx if you are using an SLC. Refer to http://www.software.rockwell.com for more information on these products.
- Connected to and communicating with the controller.

Message Translations

The communications adapter provides electronic translations of DeviceNet explicit messages into SCANport messages and back. The format of all explicit messages supported by the adapter is 8:16. The class field is 8 bits long, and the instance field is 16 bits long.

ATTENTION: Hazard of equipment damage exists. If explicit messages are programmed to frequently write parameter data to certain drive products, the EEPROM (Non-Volatile Storage) will quickly exceed its life cycle and cause the product to malfunction. Do not create a program that frequently uses explicit messages to write parameter data to a product. Datalinks do not write to the EEPROM and should be used for frequently changed parameters.

Messaging for the 1771-SDN Scanner

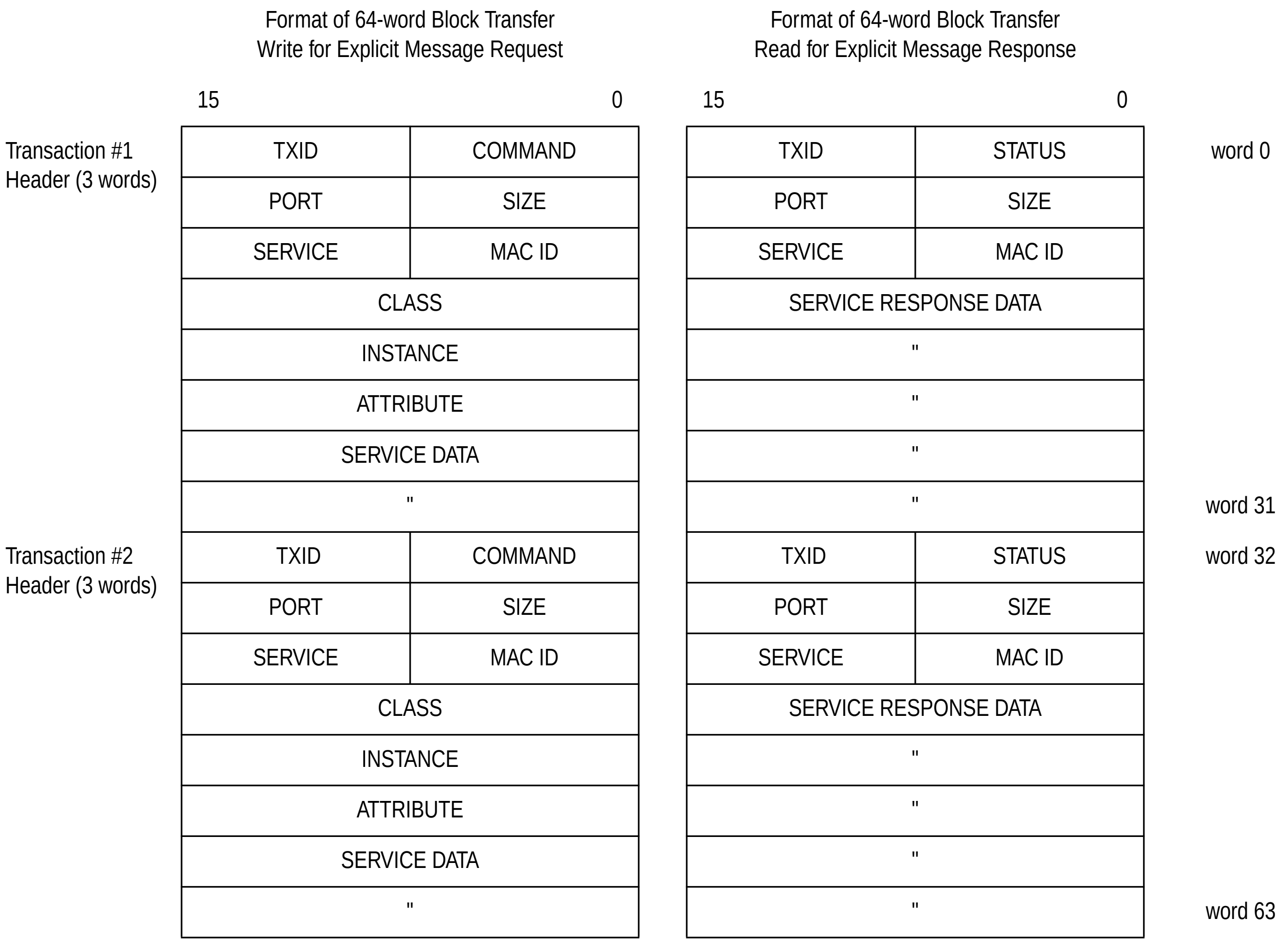
The PLC uses a 64-word Block Transfer Write (BTW) to copy an Explicit Message into the 1771-SDN scanner. Ten explicit message buffers are available within the 1771-SDN scanner. When the BTW completes, the scanner executes the message. The PLC must then poll the scanner by performing a 64-word Block Transfer Read (BTR) to complete the message. (Note that the PLC can transfer two Explicit Messages per BTW or BTR and the scanner can have up to 10 Explicit Messages active at any time.)

When the BTR completes, the data received by the PLC will contain information about the status of the current Explicit Message being processed by the scanner. If an Explicit Message has completed, the STATUS code in the Explicit Message Response is set to 1 and the response message contains the data requested.

If the message status indicates that it is not completed, the BTR should be repeated until the message is complete.

The format of Request and Response messages is in Figure 7.1Figure 7.1.

Figure 7.1 Format of DeviceNet Messages



	Format of 64-word Block Transfer Write for Explicit Message Request		Format of 64-word Block Transfer Read for Explicit Message Response		
	15	0	15	0	
Transaction #1 Header (3 words)	TXID	COMMAND	TXID	STATUS	word 0
	PORT	SIZE	PORT	SIZE	
	SERVICE	MAC ID	SERVICE	MAC ID	
	CLASS		SERVICE RESPONSE DATA		
	INSTANCE		"		
	ATTRIBUTE		"		
	SERVICE DATA		"		
	"		"		word 31
Transaction #2 Header (3 words)	TXID	COMMAND	TXID	STATUS	word 32
	PORT	SIZE	PORT	SIZE	
	SERVICE	MAC ID	SERVICE	MAC ID	
	CLASS		SERVICE RESPONSE DATA		
	INSTANCE		"		
	ATTRIBUTE		"		
	SERVICE DATA		"		
	"		"		word 63

Transaction Blocks are divided into two parts:

- **Transaction header** — contains information that identifies the transaction to the scanner and processor.
- **Transaction body** — in a request, this contains the DeviceNet Class, Instance, Attribute and Service Data portion of the transaction. In a response, this contains the Service Data only.

Each of the data attributes in the transaction header are one byte in length:

Data Field	Description
TXID	Transaction ID — when the processor creates and downloads a request to the scanner, the processor's ladder logic program assigns a TXID to the transaction. This is a one-byte integer in word 31 the range of 1 to 255. The scanner uses this value to track the transaction to completion, and returns the value with the response that matches the request downloaded by the processor.
COMMAND	In each message request, a command code instructs the scanner how to administer the request: 0 = Ignore transaction block (block empty) 1 = Execute this transaction block 2 = Get status of transaction TXID 3 = Reset all client/server transactions 4-255 = Reserved
STATUS	In each message response, the status code provides the processor with status on the device and its response: 0 = Ignore transaction block (block empty) 1 = Transaction completed successfully 2 = Transaction in progress (not ready) 3 = Error — slave not in scan list 4 = Error — slave off-line 5 = Error — DeviceNet port disabled or off-line 6 = Error — transaction TXID unknown 7 = Unused 8 = Error — Invalid command code 9 = Error — Scanner out of buffers 10 = Error — Other client/server transaction in progress 11 = Error — could not connect to slave device 12 = Error — response data too large for block 13 = Error — invalid port 14 = Error — invalid size specified 15 = Error — connection busy 16-255 = Reserved
Port	The DeviceNet port where the transaction is routed. The port can be zero (Channel A) or one (Channel B) on a 1771-SDN scanner.
Size	The size of the transaction body in bytes. The transaction body can be up to 29 words (58 bytes) in length. If the size exceeds 29 words, an error code will be returned.
SERVICE	The service attribute contains the DeviceNet service request and response codes that match the corresponding request for the TXID.
MAC ID	The DeviceNet network address of the slave device where the transaction is sent. This value can range from 0 to 63. The port and MAC ID attributes coupled together identify the target slave device. The slave device must be listed in the scanner module's scan list and be on-line for the Explicit Message transaction to be completed.

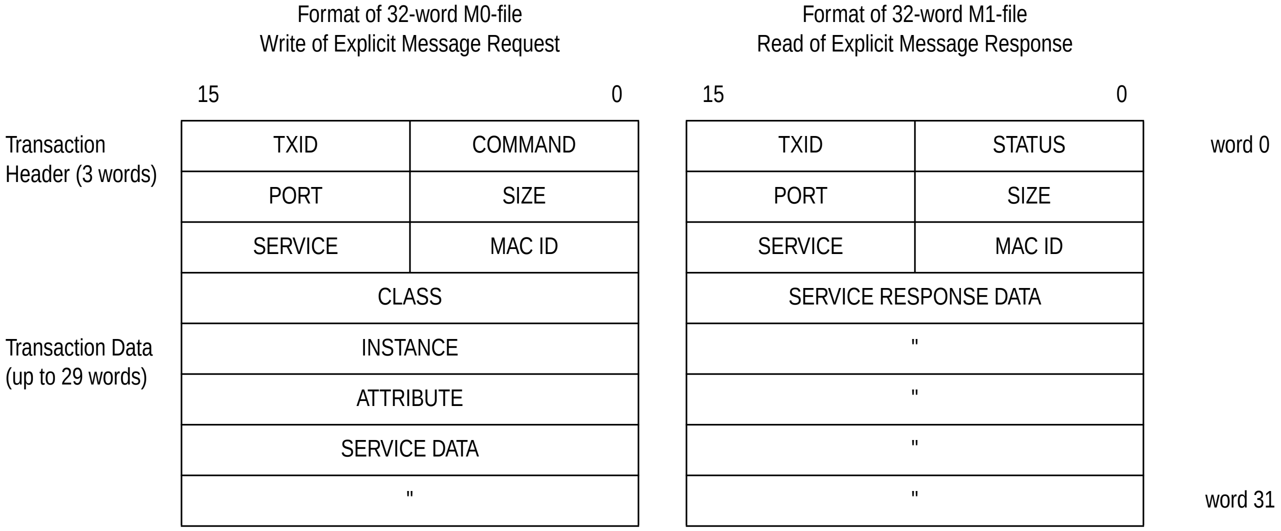
Messaging for the 1747-SDN Scanner

The SLC copies an Explicit Message into the scanner’s M0-file. When the copy is completed the scanner moves the message into a queue for processing. Up to 10 Explicit Messages can be in this queue.

When the scanner receives a response message it is placed into a queue. The first response in the queue is available from the M1-file. When the message delete command is copied into the scanner the message is complete and the next available response will appear in the M1-file.

The format of Request and Response messages is in Figure 7.2Figure 7.2.

Figure 7.2 Format of DeviceNet Messages



	Format of 32-word M0-file Write of Explicit Message Request		Format of 32-word M1-file Read of Explicit Message Response		
	15	0	15	0	
Transaction Header (3 words)	TXID	COMMAND	TXID	STATUS	word 0
	PORT	SIZE	PORT	SIZE	
	SERVICE	MAC ID	SERVICE	MAC ID	
	CLASS		SERVICE RESPONSE DATA		
Transaction Data (up to 29 words)	INSTANCE		"		
	ATTRIBUTE		"		
	SERVICE DATA		"		
	"		"		word 31

For information on M-File locations, refer to the *1747-SDN DeviceNet Scanner Module Manual*, Publication 1747-5.8.

The message buffer is composed of two sections:

- **Transaction header** — three words that contain information identifying the message transaction.
- **Transaction body** — in a request, this contains the DeviceNet Class, Instance, Attribute and Service Data portions of the transaction. In a response, this contains the Service Data only.

Each of the data fields in the transaction header are one byte in length:

Data Field	Description
TXID	Transaction ID — when the processor creates and downloads a request to the scanner, the processor's ladder logic program assigns a TXID to the transaction. This is a one-byte integer in word 31 the range of 1 to 255. The scanner uses this value to track the transaction to completion, and returns the value with the response that matches the request downloaded by the processor.
COMMAND	For each download, a command code instructs the scanner how to administer the request: 0 = Ignore transaction block (block empty) 1 = Execute this transaction block 2 = Get status of transaction TXID 3 = Reset all client/server transactions 4 = Delete this transaction block 5-255 = Reserved
STATUS	For each upload, the status code provides the processor with status on the device and its response: 0 = Ignore transaction block (block empty) 1 = Transaction completed successfully 2 = Transaction in progress (not ready) 3 = Error — Slave not in scan list 4 = Error — Slave off-line 5 = Error — DeviceNet port disabled or off-line 6 = Error — Transaction TXID unknown 7 = Unused 8 = Error — Invalid command code 9 = Error — Scanner out of buffers 10 = Error — Other client/server transaction in progress 11 = Error — Could not connect to slave device 12 = Error — Response data too large for block 13 = Error — Invalid port 14 = Error — Invalid size specified 15 = Error — Connection busy 16-255 = Reserved
PORT	The DeviceNet port used by this message. The port must be zero (Channel A) on a 1747-SDN scanner.
SIZE	The size of the transaction body in bytes. The transaction body can be up to 29 words (58 bytes) in length. If the size exceeds 29 words, an error code will be returned.
SERVICE	The service attribute contains the DeviceNet service request and response codes that match the corresponding request for the TXID.
MAC ID	The DeviceNet network address of the slave device where the transaction is sent. This value can range from 0 to 63. The port and MAC ID uniquely identify the target slave device. The slave device must be listed in the scanner module's scan list and be on-line for the Explicit Message transaction to be completed.

Examples

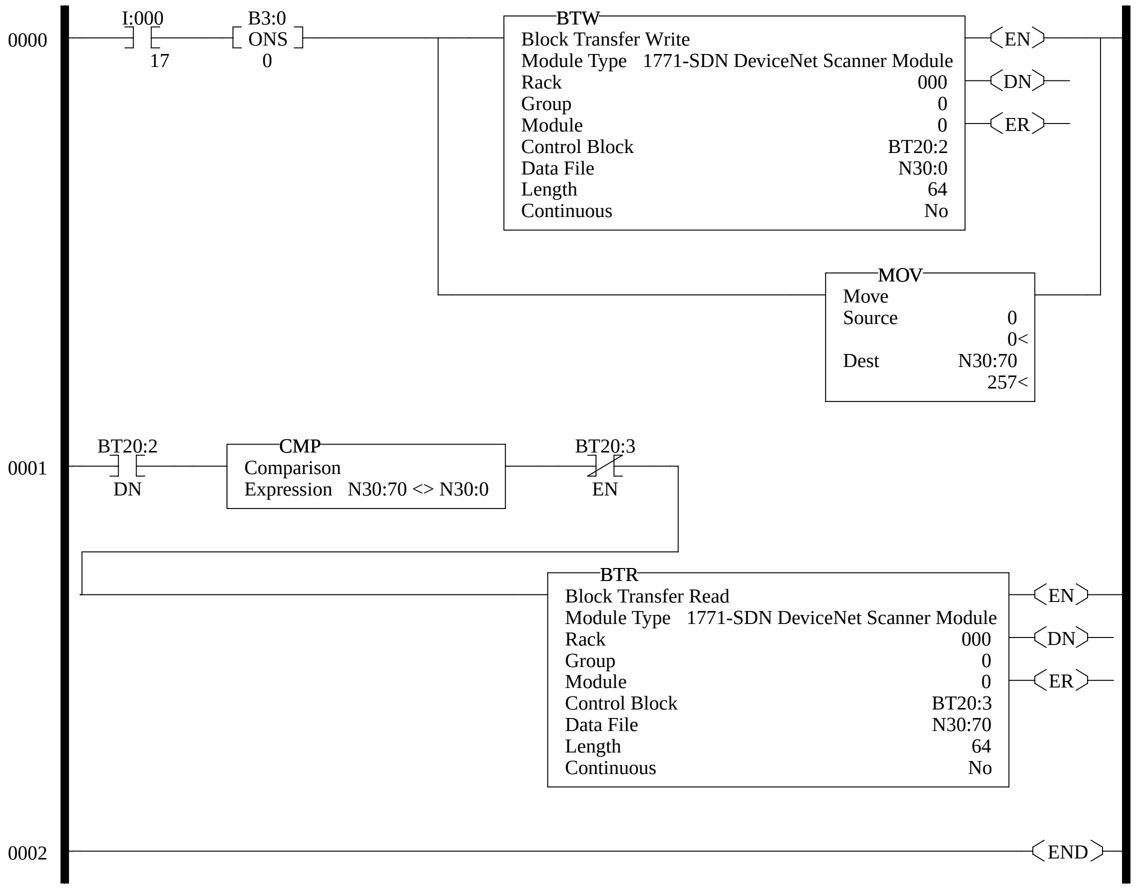
The following examples show messages used with the ladder logic programs begun in Chapter 6, *Ladder Logic Programming*.

ATTENTION: The example ladder logic program shown in this manual is intended solely for purpose of example. Because there are many variables and requirements associated with any particular installation, Rockwell Automation does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the example shown in this publication.

PLC Messaging

Figure 7.3, started in Chapter 6, *Ladder Logic Programming*, shows an example PLC ladder logic program that sends an explicit message. The message sent is contained in N30, beginning at word 0. The response data will appear in N30 starting at word 70.

Figure 7.3 PLC Messaging Example



I:000/17: When you set this instruction to the true state, the next instruction, a one-shot block transfer write, sends data to the scanner. The Move instruction then initializes the first word of the data file that is used by the block transfer read instruction in the next rung.

Instruction BT20:2.DN: This instruction will be true when the block transfer write has completed. The compare instruction that follows compares the first word of data sent from the scanner to the first word of data you send to the scanner. When the messaging function has completed, these two words will be equal.

Instruction BT20:3.EN: Any time the block transfer read is not enabled, this instruction causes the block transfer read to be enabled if the two earlier conditions are true.

BTR: Reads 64 words of data from the scanner.

The following table displays data sent to and received from the scanner. Values are in hexadecimal.

Offset	0	1	2	3	4	5	6	7	8	9
N30:0	201	6	E03	97	0	1	0	0	0	0
N30:10	0	0	0	0	0	0	0	0	0	0
N30:20	0	0	0	0	0	0	0	0	0	0
N30:30	0	0	0	0	0	0	0	0	0	0
N30:40	0	0	0	0	0	0	0	0	0	0
N30:50	0	0	0	0	0	0	0	0	0	0
N30:60	0	0	0	0	0	0	0	0	0	0
N30:70	201	6	8E03	4	0	0	0	0	0	0
N30:80	0	0	0	0	0	0	0	0	0	0
N30:90	0	0	0	0	0	0	0	0	0	0
N30:100	0	0	0	0	0	0	0	0	0	0
N30:110	0	0	0	0	0	0	0	0	0	0
N30:120	0	0	0	0	0	0	0	0	0	0
N30:130	0	0	0	0						

In this example, there were four entries in the fault queue. Notice the following about the data:

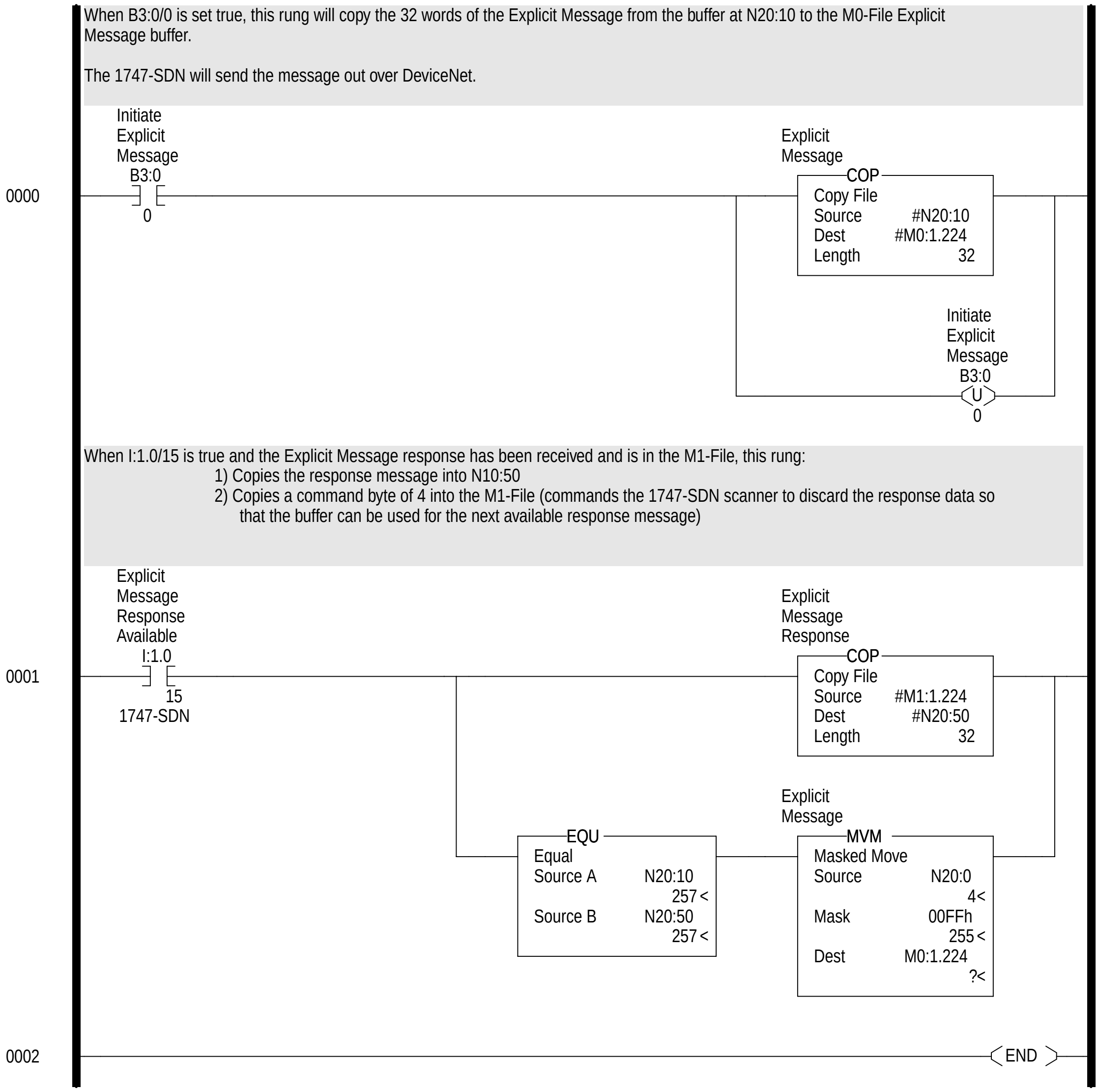
Location	Value	Meaning
N30:0	0x0201	TXID of 2. Command 1 (Execute)
N30:1	0x0006	Port 0. Size = 6 bytes (N30:3 – 5)
N30:2	0x0E03	Service E (Get Attribute Single) Node 3
N30:3	0x0097	SCANport Pass-Through Fault Object
N30:4	0x0000	Instance 0 (Class Access)
N30:5	0x0001	Attribute 1 (Number of Fault Queues)
N30:70	0x0201	TXID of 2. Status 1 (Success)
N30:71	0x0006	Port 0. Size = 6 bytes (N30:3 – 5)
N30:72	0x8E03	Service 8E (Get Attribute Single Response)
N30:73	0x0004	4 fault queues

Refer to Appendix C, *DeviceNet Objects*, for more information on object descriptions.

SLC Messaging

Figure 7.4 shows an example message in the SLC ladder logic program started in Chapter 6, *Ladder Logic Programming*.

Figure 7.4 SLC Messaging Example


When B3:0/0 is set true, this rung will copy the 32 words of the Explicit Message from the buffer at N20:10 to the M0-File Explicit Message buffer.
The 1747-SDN will send the message out over DeviceNet.
0000
Initiate Explicit Message
B3:0
0
Explicit Message
COP
Copy File
Source #N20:10
Dest #M0:1.224
Length 32
Initiate Explicit Message
B3:0
U
0
When I:1.0/15 is true and the Explicit Message response has been received and is in the M1-File, this rung:
1) Copies the response message into N10:50
2) Copies a command byte of 4 into the M1-File (commands the 1747-SDN scanner to discard the response data so that the buffer can be used for the next available response message)
0001
Explicit Message Response Available
I:1.0
15
1747-SDN
Explicit Message Response
COP
Copy File
Source #M1:1.224
Dest #N20:50
Length 32
EQU
Equal
Source A N20:10
257<
Source B N20:50
257<
Explicit Message
MVM
Masked Move
Source N20:0
4<
Mask 00FFh
255<
Dest M0:1.224
?<
0002
END

Important: To originate a scanner transaction, you must use a copy operation to M0:[slot number]:224. Then, use a copy operation to read M1:1.224 for the results. If you have more than one message enabled, you will have to use the TXID to determine which message you are reading.

The following table display data sent to and received from the scanner. Values are in hexadecimal.

Offset	0	1	2	3	4	5	6	7	8	9
N20:0	4	0	0	0	0	0	0	0	0	0
N20:10	201	6	E02	97	0	1	0	0	0	0
N20:20	0	0	0	0	0	0	0	0	0	0
N20:30	0	0	0	0	0	0	0	0	0	0
N20:40	0	0	0	0	0	0	0	0	0	0
N20:50	201	6	8E02	4	0	0	0	0	0	0
N20:60	0	0	0	0	0	0	0	0	0	0
N20:70	0	0	0	0	0	0	0	0	0	0
N20:80	0	0								

In this example, there were four entries in the fault queue. Notice the following about the data:

Location	Value	Meaning
N20:10	0x0201	TXID of 2. Command 1 (Execute)
N20:11	0x0006	Port 0. Size = 6 bytes (N30:3 – 5)
N20:12	0x0E03	Service E (Get Attribute Single) Node 3
N20:13	0x0097	SCANport Pass-Through Fault Object
N20:14	0x0000	Instance 0 (Class Access)
N20:15	0x0001	Attribute 1 (Number of Fault Queues)
N20:50	0x0201	TXID of 2. Status 1 (Success)
N20:51	0x0006	Port 0. Size = 6 bytes (N30:3 – 5)
N20:52	0x8E03	Service 8E (Get Attribute Single Response)
N20:53	0x0004	4 fault queues

Refer to Appendix C, *DeviceNet Objects*, for more information on object descriptions.

Using Messages to Control SCANport Products

Explicit messages provide multi-purpose, point-to-point communication paths between two devices. It is possible to control SCANport devices through explicit messaging on DeviceNet by following particular guidelines and by writing to various register objects that are buffering the I/O data. The guidelines are as follows:

- The adapter cannot be allocated by a master/scanner in order to allow explicit writes to the register object.

- Write access to any register object within the adapter will not be allowed if the message is passed through a connection whose expected packet rate (EPR) is zero.
- The adapter is required to mark any explicit connection after allowing a write to a register object through it.
- If a marked explicit connection times out based on the EPR, then the I/O fault action will be that configured for Communication Loss over the I/O connection.
- If a marked explicit connection is deleted, then the I/O fault action will be that configured for Idle over the I/O connection.
- Multiple explicit connections can write/overwrite the control I/O if they meet the guidelines specified. Each connection will need to be marked individually within the adapter.
- If the adapter gets allocated/re-allocated by a controller such that valid I/O data is being sent to the adapter, or if an Idle condition from the allocating controller is transitioned back to valid data, then all marked explicit connections will be reset to unmarked and future writes blocked.
- If a marked connection has its EPR value reset to zero (0) after being marked, then the connection will become unmarked.

Writing to Register Objects

Within the Enhanced DeviceNet adapter, various register objects buffer I/O in the following fashion (RO=Read Only, R/PW=Read/Write Protected):

Instance	Access	Size	Function
1	RO	See M-S Output	Poll Response I/O data to controller
2	R/PW	See M-S Output	Buffered Poll I/O data from controller
3	RO	32 bits	Logic Status & Feedback
4	R/PW	32 bits	Datalink A from SCANport Device (if enabled)
5	RO	32 bits	Datalink A to SCANport Device (if enabled)
6	R/PW	32 bits	Datalink A from SCANport Device (if enabled)
7	RO	32 bits	Datalink B to SCANport Device (if enabled)
8	R/PW	32 bits	Datalink B from SCANport Device (if enabled)
9	RO	32 bits	Datalink C to SCANport Device (if enabled)
10	R/PW	32 bits	Datalink C from SCANport Device (if enabled)
11	RO	32 bits	Datalink D to SCANport Device (if enabled)
12	R/PW	32 bits	Datalink D from SCANport Device (if enabled)
13	RO	32 bits	Logic Status and Feedback
14	R/PW	32 bits	Logic Command (Last Logic Command is ANDed with the first word in this command and ORed with the second word in the command. The reference is not modified. This command allows bit changes to the logic command without affecting the speed reference or other control bits.)

Refer to the “Class Code 0x07 — Register Object” section in Appendix C, *DeviceNet Objects* for more information on the Register Object.

Chapter **8**

Troubleshooting

Chapter Objectives

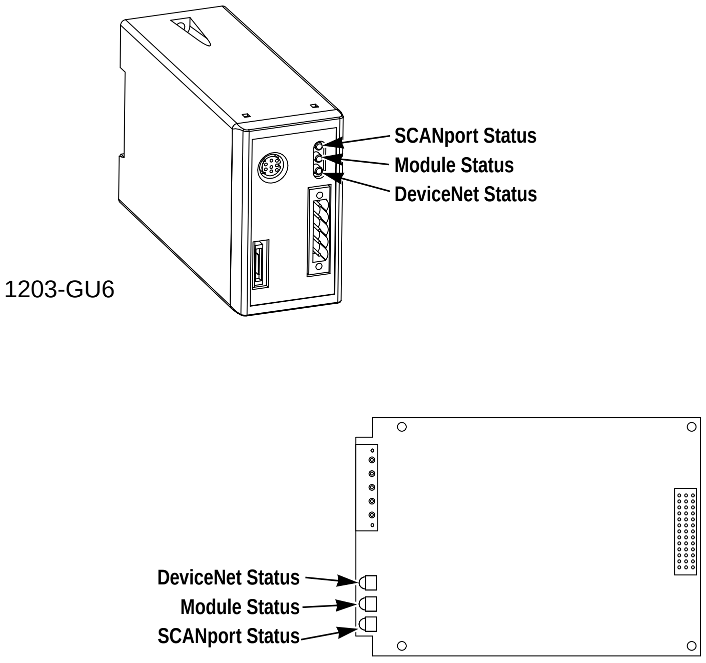
Chapter 8 provides information about the adapter’s LEDs and basic troubleshooting procedures. In this chapter, you will read about the following:

- Locating the LEDs.
- Using the LEDs to troubleshoot the adapter.

LEDs on the Enhanced DeviceNet Adapter

Your communications adapter has three LED status indicators. The LEDs provide status information about the DeviceNet network, SCANport connection, and the adapter itself. Refer to Figure 8.1.

Figure 8.1
LED Status Indicators



DeviceNet Network Status LED States

The LED closest to the DeviceNet connector is the DeviceNet Status LED, labeled "NET." It functions as follows:

LED Viewed:	If:	State:	Indicates	Action:
Network Status LED	LED is off	Not powered/Not online	No power/Duplicate ID not completed	1. Verify that the network supply is connected and that power is reaching the adapter through the connector. 2. Make sure one or more nodes are communicating on the network. 3. Make sure at least one other node on the network is operational at the same time and data rate as the adapter.
	LED is flashing green	Online/Not connected	Passed duplicate ID/No connection established	No action needed. The LED is flashing to signify that there are no open communication connections between the adapter and any other device. Any connection (I/O or explicit message) made to the adapter over DeviceNet will cause the LED to stop flashing and remain steady on for the duration of any open connection.
	LED is steady green	Online/Connected	One or more connections established	No action needed.
	LED is flashing red	Online/Time-out	I/O connection timed out	1. Bring controller back onto the network. 2. Reduce traffic or errors on the network so that messages can get through within the necessary time frame.
	LED is steady red	Network failure	Failed Duplicate ID or Bus-off	1. Ensure that all nodes have unique addresses. 2. If all node addresses are unique, examine network for correct media installation.

Module Status LED States

The middle LED, labeled "MOD," is the Module Status LED. It indicates the operation of the Enhanced DeviceNet adapter and functions as follows:

LED Viewed:	If:	State:	Indicates:	Action:
Module Status LED	LED is off	Not powered	No power	Ensure that the connected SCANport product is powered and connected to the adapter.
	LED is flashing green	Waiting for I/O data	Normal operation — No I/O, or PLC in program	No action needed. Adapter has passed all operational tests and is waiting to pass I/O data between the DeviceNet and SCANport interfaces.
	LED is solid green	Operational	Normal operation — I/O operational	No action needed.
	LED is flashing red	Configuration problem	Bad CRC of Adapter parameters or flash program	1. Power cycle the adapter to reset it. 2. Enable an adapter reset via the adapter's configuration parameter. 3. Re-flash the adapter.
	LED is steady red	Hardware failure	Failed internal or external RAM test	Replace unit.

SCANport Status LED States

The LED furthest from the DeviceNet connector is the SCANport Status LED, and is labeled “SP.” It indicates the status of the SCANport connection, and functions as follows:

LED Viewed:	If:	State:	Indicates	Action:
SCANport Status LED	LED is off	Not powered	No power	Ensure that the connected SCANport device is powered and that the product is connected to the adapter.
	LED is flashing green	Online/ I/O connecting	Requesting I/O connections or no I/O is selected	No action needed. Adapter is establishing one or more of the I/O connections with the SCANport device.
	LED is steady green	I/O operational	One or more connections established	No action needed.
	LED is flashing red	Configuration fault	SCANport problem: No communications seen from the SCANport device to request connections	1. Reseat cable properly. 2. Replace cable.
	LED is steady red	Link failure	SCANport failure: Poor cable connection does not allow proper port identification or the SCANport device isn’t allowing the configured I/O connections.	1. Check all SCANport cables and connections to the SCANport device. 2. Ensure datalinks are not enabled on a product that doesn’t support datalinks. Or, ensure that a particular datalink is not already being used by another adapter on the same SCANport device. In either case, the adapter must be properly configured and power cycled after the problem is fixed.
	LED is steady orange	Failed SCANport compatibility test		Call Rockwell Automation support.

Notes:

Specifications

Appendix Objectives

Appendix A provides the specifications that you may need to install or use either the 1203-GU6 module or the 1336-GM6 board. These adapters are non-repairable units. If they are broken, you must replace them.

1203-GU6 Specifications

The following table gives the specifications for the 1203-GU6 Enhanced DeviceNet communications module.

Category	Specifications
Dimensions	1.8" W x 3.0" H x 5.0" D (4.57 cm x 7.62 cm x 12.7 cm)
Weight	6.3 oz (179 g)
Operating Temperature	0 to +55°C (32 to 131°F)
Storage Temperature	–40 to +85°C (–40 to 185°F)
Relative Humidity (Operating)	5 to 80% non-condensing
Relative Humidity (Non-Operating)	5 to 95% non-condensing
Shock (Operating)	30g peak acceleration, 11(+/–1)ms pulse width
Shock (Non-Operating)	50g peak acceleration, 11(+/–1)ms pulse width
Vibration (Operating)	2.5g at 5Hz – 2KHz
Vibration (Non-Operating)	5g at 5Hz – 2KHz
Power Consumption	130mA at 12V supplied through SCANport, and 60mA at 24V supplied through DeviceNet
ESD Susceptibility (IEC 1000-4-2)	4KV contact, 8KV open air
Regulatory Agencies	UL 508 and CUL European Union EMC and Low Voltage Directives
DIN Rail Mounting Standard	1.38 x 0.30 in. (35 x 7.5 mm)

1336-GM6 Specifications

The following table gives the specifications for the 1336-GM6 Enhanced DeviceNet communications board.

Category	Specifications
Dimensions	4.5" W x 2.8" H x 0.5" D (11.43 cm x 7.112 cm x 1.27 cm)
Weight	1.8 oz (51 g)
Operating Temperature	0 to +55°C (32 to 131°F)
Storage Temperature	–40 to +85°C (–40 to 185°F)
Relative Humidity (Operating)	5 to 80% non-condensing
Relative Humidity (Non-Operating)	5 to 95% non-condensing
Shock (Operating)	30g peak acceleration, 11(+/–1)ms pulse width
Shock (Non-Operating)	50g peak acceleration, 11(+/–1)ms pulse width
Vibration (Operating)	2.5g at 5Hz – 2KHz
Vibration (Non-Operating)	5g at 5Hz – 2KHz
Power Consumption	150mA at 5V (supplied from Drive Control Board) 60mA at 24V (supplied through DeviceNet)
ESD Susceptibility (IEC 1000-4-2)	4KV contact, 8KV open air
Regulatory Agencies	UL 508 and CUL European Union EMC and Low Voltage Directives

⚠ **ATTENTION:** The 1336-GM6 communications board contains ESD (Electrostatic Discharge) sensitive parts. Static control precautions are required when installing, testing, or servicing this assembly. Device malfunction may occur if you do not follow ESD control procedures. If you are not familiar with static control procedures, refer to Allen-Bradley Publication 8000-4.5.2, *Guarding Against Electrostatic Damage*, or other applicable ESD protection handbook.

Enhanced DeviceNet Adapter's Parameters

Appendix Objectives

Appendix B provides information on the Enhanced DeviceNet adapter's parameters and how to configure them. In this appendix, you will read about the following:

- Setting the node address.
- Setting the data rate.
- Using datalinks and command I/O.
- Using Master-Slave communications.
- Using Peer-to-Peer communications.
- Using Fault Configurable inputs.
- Parameters in the Enhanced DeviceNet adapter.

Important: The the number for parameters appears in parenthesis after the name. This is the number in the adapter. If you are using RSNetWorx for DeviceNet, the number for each adapter parameter varies depending on the type of drive you are using.

Setting the Node Address

The Enhanced DeviceNet adapter has a default node address of 63. This address should be changed to a unique address (between 0 and 62) on your DeviceNet network.

Important: If you are installing multiple Enhanced DeviceNet adapters on the network at the same time or there is already a node 63 on the network, you need to do one of the following:

- Connect to each adapter using a serial point-to-point connection and give each a unique address.
- Power up only one adapter at a time on the network and give each a unique node address.
- If using software that supports the Fault Node Recovery feature of DeviceNet, power up all the adapters at the same time and give each a unique node address.

To set the node address, you need to:

1. Access the adapter's parameters using either a serial connection (refer to Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*) or over the DeviceNet network (refer to Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*).

2. Set the *DN Node Address* (2) parameter to the desired address.
3. Reset the adapter by setting the *Reset Adapter* (22) parameter to **Enable**.

Setting the Data Rate

The Enhanced DeviceNet adapter supports the following data rates:

- 125 Kbps
- 250 Kbps
- 500 Kbps
- Autobaud

The adapter defaults to using autobaud data rate detection.

Important: At least one continually transmitting device on the network (usually the scanner) must be set to a fixed data rate (not autobaud). This device sets the data rate for the network that the other nodes using autobaud detect.

If you want to change your adapter’s data rate, you need to:

1. Access the adapter’s parameters using either a serial connection (refer to Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*) or the DeviceNet network (refer to Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*).
2. Set the *DN Data Rate* (3) parameter to the desired value. Refer to the following table for information on acceptable values.

Data Rates	125 Kbps	250 Kbps	500 Kbps
Thick Trunk Length	500 m (1 640 ft)	250 m (820 ft)	100 m (328 ft)
Thin Trunk Length	100 m (328 ft)	100 m (328 ft)	100 m (328 ft)
Maximum Drop Length	6 m (20 ft)	6 m (20 ft)	6 m (20 ft)
Cumulative Drop Length	156 m (512 ft)	78 m (256 ft)	39 m (128 ft)

3. Reset the adapter by setting the *Reset Adapter* (22) parameter to **Enable**.

Using Datalinks and Command I/O

Command I/O provides two 16-bit words of input and two 16-bit words of output when enabled. Datalinks let you increase the size of I/O to and from a SCANport device (provided the SCANport device supports datalinks). By enabling datalinks, you can continuously change or monitor the value of a parameter without using the DeviceNet to SCANport messaging function.

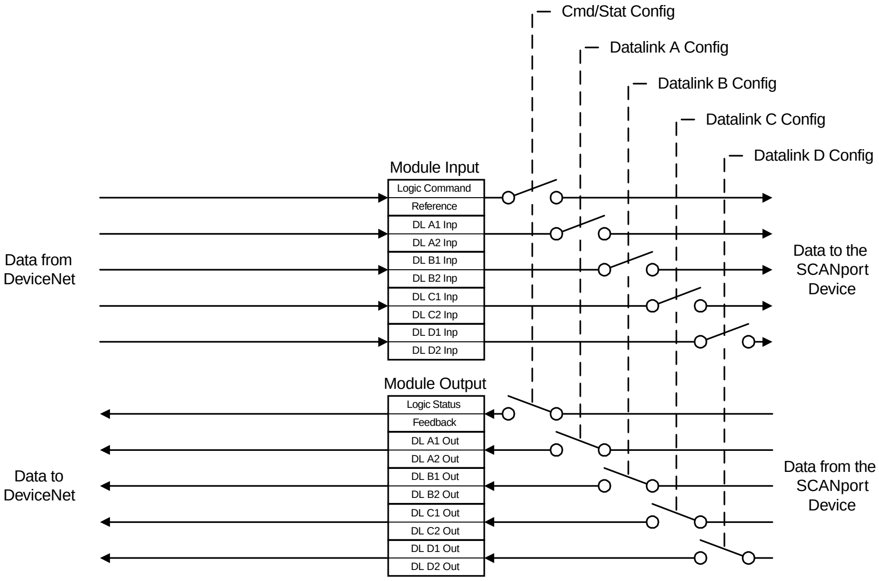
Datalinks consist of two 16-bit words of input and two 16-bit words of output when enabled. They provide up to eight words (in and out) of data if they are supported in the connected SCANport product.

SCANport devices that support this function have a group of parameters for datalink configuration. These parameters are *Data In A1 – D2* and *Data Out A1 – D2*.

If you intend to use command I/O and/or datalinks, you must do the following:

1. Access the adapter's parameters using either a serial connection (refer to Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*) or the DeviceNet network (refer to Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*).
2. Enable the *Cmd/Stat Config* (4) parameter and/or desired *DataLink* (5 – 8) parameters within the Enhanced DeviceNet adapter.

Figure B.1 Module I/O Configuration

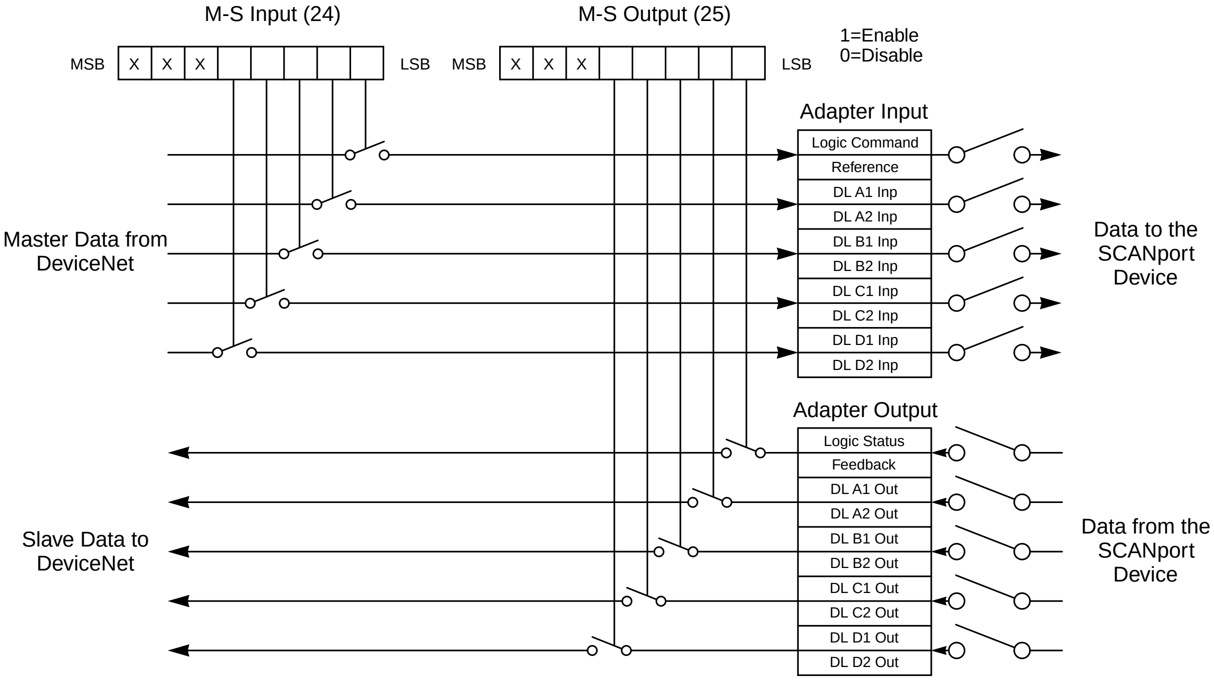


3. Configure or link the *Data In A1– D2* and *Data Out A1 – D2* parameters in the SCANport product. Refer to the documentation for your SCANport product.
4. Configure the *M-S Input* parameter and *M-S Output* parameter as desired. Refer to the "M-S Input Parameter Configurations" section or the "M-S Output Parameter Configurations" sections in this chapter.
5. Reset the adapter by setting the *Reset Adapter* (22) parameter to **Enable**.

Using Master-Slave Communications

To have your Enhanced DeviceNet adapter receive its I/O from a scanner (PLC or SLC) on the DeviceNet network, you must configure it for Master-Slave communications.

Figure B.2 Master-Slave Communications



Master-Slave communications let you transmit 2 – 10 words of I/O data between the adapter and the scanner.

How you configure your adapter and scanner depends on the type of allocation you are using. The adapter supports the following:

- Polled.
- COS (Change of State).
- Cyclic.
- Polled and COS.
- Polled and Cyclic.

Polled Allocation

In polling, the scanner sends control data to the adapter, and then the adapter responds with its status data. To use polling, you must enable polling in the adapter and in the scanner.

1. Enable the desired I/O and datalinks (parameters 4 – 8).
2. Set the *M-S Input* (24) parameter. Refer to the “M-S Input Parameter Configurations” section in this chapter.
3. Set the *M-S Output* (25) parameter. Refer to the “M-S Output

Parameter Configurations” section in this chapter.

4. Reset the adapter by setting the *Reset Adapter* (22) parameter to **Enable**.

5. When you configure the scanner, set up polled I/O. Refer to “Setting Up the Scan List” in Chapter 5.

COS (Change of State) Allocation

When you set up COS allocation, the scanner sends data to the adapter at a constant rate (called a heartbeat). If data in the adapter changes between messages from the scanner, your adapter sends its new status to the scanner. To use COS, you must enable COS in the adapter and in the scanner.

1. Enable the desired I/O and datalinks (parameters 4 – 8).

2. Set the *M-S Input* (24) parameter. Refer to the “M-S Input Parameter Configurations” section in this chapter.

3. Set the *M-S Output* (25) parameter. Refer to the “M-S Output Parameter Configurations” section in this chapter.

4. Ensure the *Cmd/Stat Config* (4) parameter is **On**.

5. Ensure the lowest bit in the *M-S Output* (25) parameter is set to **1**. For example, xxx0, 0001. This enables status/feedback to be sent over the slave connection.

6. Set the *COS Status Mask* (26) parameter to specify which bits in the logic status word will trigger a message to the scanner when changed. (0 = do not check the corresponding bit.)

7. Set the *COS Fdbk Change* (27) parameter to specify the amount of change required in the reference word needed to trigger a message to the scanner. (0 = do not check the corresponding bit)

8. Reset the adapter by setting the *Reset Adapter* (22) parameter to **Enable**.

9. When you configure the scanner, set up Change of State. Refer to “Setting Up the Scan List” in Chapter 5.

Important: After you have configured the scanner and adapter for COS, you can verify the desired heartbeat rate is used by viewing the *COS/CYC Interval* (28) parameter in the adapter.

Cyclic Allocation

When you set up cyclic allocation, your adapter sends or receives data based on a periodic time interval. To use Cyclic, you must enable Cyclic in the adapter and in the scanner.

1. Enable the desired command I/O and datalinks. Refer to the “Using Datalinks and Command I/O” section in this chapter.

2. Set the *M-S Input* (24) parameter. Refer to the "M-S Input Parameter Configurations" section in this chapter.
3. Set the *M-S Output* (25) parameter. Refer to the "M-S Output Parameter Configurations" section in this chapter.
4. Ensure the *Cmd/Stat Config* (4) parameter is set to **On**.
5. Ensure the lowest bit in *M-S Output* (25) parameter is set to **1**.
6. Reset the adapter by setting the *Reset Adapter* (22) parameter to **Enable**.
7. When you configure the scanner, set up polled I/O. Refer to "Setting Up the Scan List" in Chapter 5.

Important: After you have configured the scanner and adapter for Cyclic, you can verify the desired send rate is used by viewing the **COS/CYC Interval** (28) parameter in the adapter.

Polled and COS Allocation

You can enable both polling and COS allocations. This lets the scanner poll the adapter at a fixed interval and the adapter report its status changes to the scanner after they occur.

Refer to both the "Polled Allocation" and the "COS (Change of State) Allocation" sections in this chapter for information.

Polled and Cyclic Allocation

You can enable both polling and cyclic allocations. This allows the scanner to poll the adapter at fixed intervals and the adapter to send its status to the scanner at fixed intervals.

Refer to both the "Polled Allocation" section and the "Cyclic Allocation" section in this chapter for information.

Using Peer-to-Peer Communications

To have your adapter receive data from or transmit data to another 1203-GU6 or 1336-GM6 on the DeviceNet network, you must configure it for peer-to-peer communications. Peer-to-peer communications are best used in the following instances:

- A PLC sends data to a drive. That drive re-transmits the data to other drives on the network.
- A drive is configured on a network. It sends data to other drives on the network.

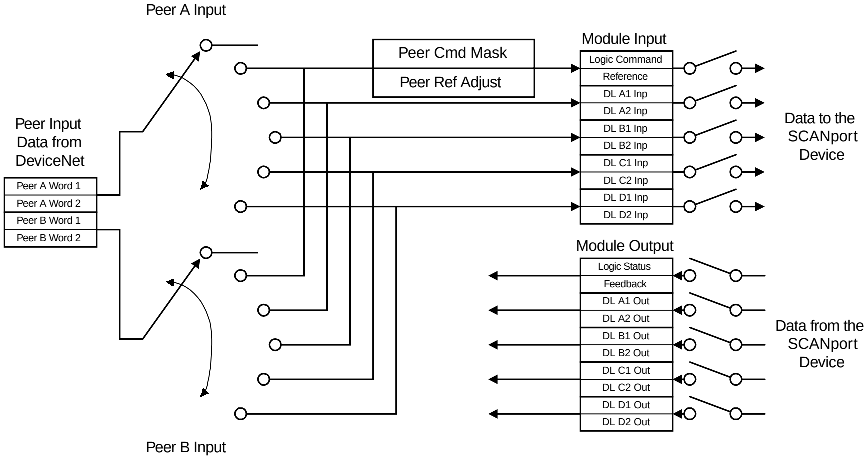
Important: After setting up peer-to-peer communications, you must make sure the configuration that you set up works as you intend it to work.

To enable peer-to-peer communications, you must enable one adapter to transmit peer I/O and one or more adapters to receive peer I/O.

Enabling the Adapter to Receive Peer I/O

To have your Enhanced DeviceNet adapter receive input data from another Enhanced DeviceNet adapter on the network, you must configure it for peer-to-peer communications.

Figure B.3 Receiving I/O from a Peer Device



In Peer-to-Peer communications, you can receive 2 or 4 I/O words from another adapter. Follow these directions:

1. Enable the desired I/O and datalinks within the adapter and SCANport product. Refer to the Using Datalinks and Command I/O section in this chapter.
2. Ensure the *Peer Inp Enable* (36) parameter is **Off**.
3. Set the *Peer Node to Inp* (34) parameter to the number of the node from which you want to receive data.
4. Set the *Peer A Input* (29) parameter to a destination for the first 2 words of data.
5. If using 4 words of input, set the *Peer B Input* (30) parameter to a destination for the second two words of data.
6. If receiving Cmd/Ref input data, set the bits in the *Peer Cmd Mask* (31) parameter according to the following table.

Important: If both Master-Slave data and Peer data are being used to control the adapter, make sure you know which one is transmitting which control bits. The adapter will receive each control bit from only one source. This includes the stop bit.

If receiving I/O from:	Then set bit to:
Master device (PLC or SLC)	0
Peer device (another Enhanced DeviceNet adapter)	1

7. If sending Cmd/Ref I/O data, set the percentage in the *Peer Ref Adjust* (23) parameter. The adapter multiplies this value with the speed reference value to determine the drive's speed.
8. Set the *Peer Inp Time-out* (35) parameter to the maximum amount of time the adapter will wait for a message before timing out.

Important: This value must be greater than the product of *Peer Out Time* (41) parameter multiplied by the *Peer Out Skip* (42) parameter in the adapter from which you are receiving I/O.

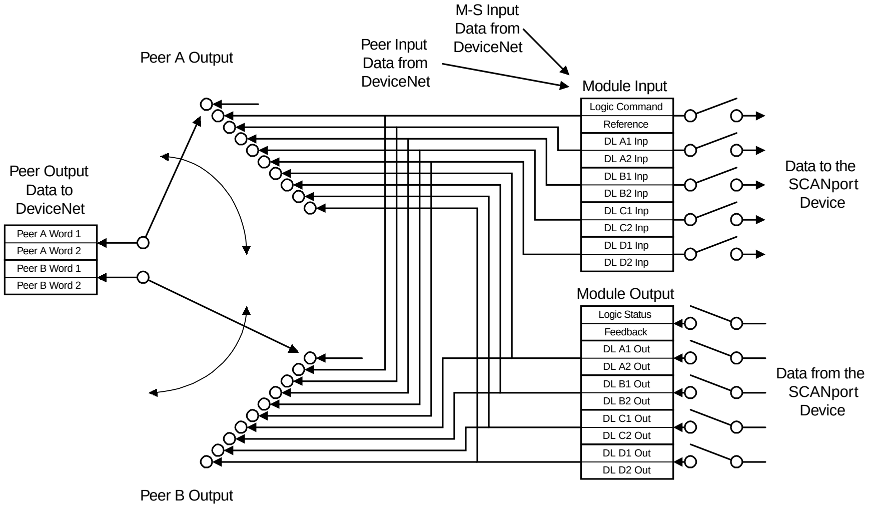
9. Set the *Peer Flt Action* (33) parameter to determine what the adapter should do if it times out.
10. Set the *Peer Inp Enable* (36) parameter to **On**.
11. Check the *Peer Inp Status* (37) parameter to verify operation. It should either be **Waiting** (meaning it is waiting for the first Tx) or **Running** (meaning it is receiving input data).

Your adapter is now configured to accept I/O data from another Enhanced DeviceNet adapter. Make sure another Enhanced DeviceNet adapter on the DeviceNet network is configured to transmit peer data. Refer to the "Enabling the Adapter to Transmit Peer I/O" section in this chapter.

Enabling the Adapter to Transmit Peer I/O

You can have your Enhanced DeviceNet adapter send I/O data to another Enhanced DeviceNet adapter on the network.

Figure B.4 Transmitting I/O to Another Adapter



To have your adapter send output data to another adapter, you need to:

1. Ensure the *Peer Output Enable* (40) parameter is **Off**.
2. Set the *Peer A Output* (38) parameter to the source of the output data.

3. If transmitting 4 words, set the *Peer B Output* (39) parameter to a different source of output data.
4. Set the *Peer Output Time* (41) parameter to the minimum time interval between peer messages.
5. Set the *Peer Output Skip* (42) parameter to a value between 1 and 16. The product of this value and the value of the *Peer Output Time* (41) parameter determine the maximum time interval between peer messages if there is not a change in status.
6. Set the *Peer Output Enable* (40) parameter to **On**.

Your adapter is now configured to transmit I/O data to another Enhanced DeviceNet adapter. Make sure another Enhanced DeviceNet adapter on the DeviceNet network is configured to receive peer data. Refer to the "Enabling the Adapter to Receive Peer I/O" section in this chapter.

Using Fault Configurable Inputs

You can select constant values that your adapter will maintain in the event of a controller mode change or error. These constant values are referred to as Fault Configurable inputs. When the controller is placed in program mode or a DeviceNet network fault occurs, the control outputs from the adapter to the SCANport product can be set to automatically switch to the constant values set in the *Fault Cfg In* parameters. This lets you define a safe operating state for controlled devices that depend on pre-programmed output from the adapter.

ATTENTION: Risk of severe bodily injury or equipment damage exists. The *Idle Flt Action* (9) and *Comm Flt Action* (10) parameters allow the user to change the default configuration that would allow the module and associated drive to continue to operate if communication is lost. Precautions should be taken to assure that your settings for these parameters and your application do not create a hazard of bodily injury or equipment damage.

If you intend to use Fault Configurable inputs, you must do the following:

1. Set desired values for the *Fault Cfg Logic* (11), *Fault Cfg Ref* (12), and *Fault Cft In* (13 – 20) parameters.
2. Set the *Idle Fault Config* (9) parameter and/or the *Comm Flt Action* (10) parameter to **Fault Cfg**.

Refer to Chapter 3, *Configuring the 1203-GU6 Enhanced DeviceNet Module Using a Serial Connection*, or Chapter 4, *Configuring the Enhanced DeviceNet Adapter Using RSNetWorx for DeviceNet*, for instructions on editing parameters.

Enhanced DeviceNet Adapter Parameters

The following table provides information on the Enhanced DeviceNet communications adapter's parameter set.

Important: When accessing this parameter set through the DeviceNet Parameter Class, add the adapter's parameter number to the number of the last parameter of the SCANport device. When accessing this parameter set through the vendor-specific SCANport Variables-Linear Class, add the adapter's parameter number to 4000H.

#	Name	Valid Values/ Settings	Default	Writable	Reset or Power Cycle	Description
1	*SCANport Adapter*	0-7	NA	No	No	Identifies the port number to-which the adapter is connected on the SCANport product.
2	*DN Node Address*	0-63	63	Yes	Yes	Identifies the DeviceNet Node Address for the adapter.
3	*DN Data Rate*	125K 250K 500K Auto	Auto	Yes	Yes	Identifies the data rate used on the DeviceNet network. **Important:** At least one node on your DeviceNet network must be configured to a data rate (125, 250, or 500 K), not autobaud.
4	*Cmd/Stat Config*	Off, On	On	Yes	Yes	Determines whether to pass logic command and analog reference control data from a DeviceNet connection to a SCANport product. If you are using a Master (scanner), parameters 24 and 25 must also be set.
5	*Datalink A Cfg*	Off, On	Off	Yes	Yes	Determines whether to pass control data contained in datalink A from a DeviceNet connection to the SCANport product. If you are using a Master (scanner), parameters 24 and 25 must also be set.
6	*Datalink B Cfg*	Off, On	Off	Yes	Yes	Determines whether to pass control data contained in datalink B from a DeviceNet connection to the SCANport product. If you are using a Master (scanner), parameters 24 and 25 must also be set.
7	*Datalink C Cfg*	Off, On	Off	Yes	Yes	Determines whether to pass control data contained in datalink C from a DeviceNet connection to the SCANport product. If you are using a Master (scanner), parameters 24 and 25 must also be set.
8	*Datalink D Cfg*	Off, On	Off	Yes	Yes	Determines whether to pass control data contained in datalink D from a DeviceNet connection to the SCANport product. If you are using a Master (scanner), parameters 24 and 25 must also be set.

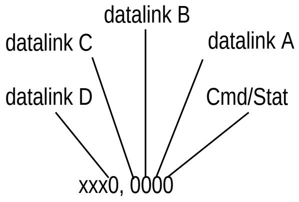
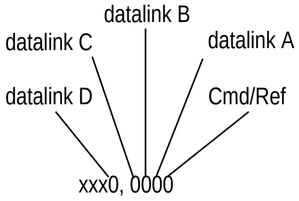
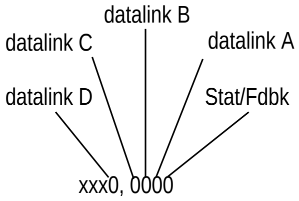
Important: For information on accessing and editing parameters, refer to Chapter 3, *Configuring the 1203-GU6 Using a Serial Connection*, or Chapter 4, *Configuring the 1203-GU6 or 1336-GM6 Adapter Using RSNetWorx for DeviceNet*.

#	Name	Valid Values/ Settings	Default	Writable	Reset or Power Cycle	Description
9	*Idle Flt Action*	Fault Zero Data Hold Last Fault Cfg	Fault	Yes	No	Determines the action the adapter should instruct the SCANport product to take if the adapter detects that the PLC is set to program mode. **Important:** If you change this parameter’s value, the user application may not be able to control the product after a fault. ⚠ **ATTENTION:** Risk of severe bodily injury or equipment damage exists. The *Idle Flt Action* (9) parameter allows the user to change the default configuration that would allow the module and associated drive to continue to operate if communication is lost. Precautions should be taken to assure that your settings for these parameters and your application do not create a hazard of bodily injury or equipment damage.
10	*Comm Flt Action*	Fault Zero Data Hold Last Fault Cfg	Fault	Yes	No	Determines the action the adapter should instruct the SCANport product to take if the adapter detects a network failure. **Important:** If you change this parameter’s value, the user application may not be able to control the product after a fault. ⚠ **ATTENTION:** Risk of severe bodily injury or equipment damage exists. The *Comm Flt Action* (10) parameter allows the user to change the default configuration that would allow the module and associated drive to continue to operate if communication is lost. Precautions should be taken to assure that your settings for these parameters and your application do not create a hazard of bodily injury or equipment damage.
11	*Fault Cfg Logic*	0 – 65535	0	Yes	No	Provides the logic command data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
12	*Fault Cfg Ref*	0 – 65535	0	Yes	No	Provides the analog reference data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.

Important: For information on accessing and editing parameters, refer to Chapter 3, *Configuring the 1203-GU6 Using a Serial Connection*, or Chapter 4, *Configuring the 1203-GU6 or 1336-GM6 Adapter Using RSNetWorx for DeviceNet*.

#	Name	Valid Values/ Settings	Default	Writable	Reset or Power Cycle	Description
13	*Fault Cfg A1 In*	0 – 65535	0	Yes	No	Provides the first word of datalink A data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
14	*Fault Cfg A2 In*	0 – 65535	0	Yes	No	Provides the second word of datalink A data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
15	*Fault Cfg B1 In*	0 – 65535	0	Yes	No	Provides the first word of datalink B data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
16	*Fault Cfg B2 In*	0 – 65535	0	Yes	No	Provides the second word of datalink B data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
17	*Fault Cfg C1 In*	0 – 65535	0	Yes	No	Provides the first word of datalink C data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
18	*Fault Cfg C2 In*	0 – 65535	0	No	No	Provides the second word of datalink C data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
19	*Fault Cfg D1 In*	0 – 65535	0	Yes	No	Provides the first word of datalink D data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
20	*Fault Cfg D2 In*	0 – 65535	0	Yes	No	Provides the second word of datalink D data to the SCANport product when the adapter is instructed to use the values for the Fault Cfg parameters.
21	*Serial Port Rate*	9600 19200	9600	Yes	Yes	Sets the baud rate for the adapter's serial/DF1 port. **Important:** If you change the baud rate in the adapter, you must also change it in your terminal emulation software or terminal.
22	*Reset Adapter*	Ready Enable Set Defaults	Ready	Yes	No	Ready = No change or reset. Enable = Resets the module. Set Defaults = Sets all parameters to their factory-default values.

Important: For information on accessing and editing parameters, refer to Chapter 3, *Configuring the 1203-GU6 Using a Serial Connection*, or Chapter 4, *Configuring the 1203-GU6 or 1336-GM6 Adapter Using RSNetWorx for DeviceNet*.

#	Name	Valid Values/ Settings	Default	Writable	Reset or Power Cycle	Description
23	*Active I/O Cfg*	0 or 1 for each bit	N/A	No	No	Displays what I/O is activated in the adapter.  datalink B datalink C datalink A datalink D Cmd/Stat xxx0, 0000 0 = Off 1 = On
24	*M-S Input*	0 or 1 for each bit	xxx0, 0001	Yes	Yes	Determines the source of the bits for input.  datalink B datalink C datalink A datalink D Cmd/Ref xxx0, 0000 0 = Peer or other input 1 = Master-Slave input For more information, refer to the "M-S Input Parameter Configurations" section in this chapter.
25	*M-S Output*	0 or 1 for each bit	xxx0, 0001	Yes	Yes	Determines the source of the bits for output.  datalink B datalink C datalink A datalink D Stat/Fdbk xxx0, 0000 0 = Peer or other output 1 = Master-Slave output For more information, refer to the "M-S Output Parameter Configurations" section in this chapter.
26	*COS Status Mask*	0 or 1 for each bit	0	Yes	No	Provides a mask of the Logic Status word to define which bits are checked for changes during COS allocation. 0 = Off (not checked) 1 = On (checked) **Important:** Refer to your SCANport product's documentation for information on its Logic Status word.

Important: For information on accessing and editing parameters, refer to Chapter 3, *Configuring the 1203-GU6 Using a Serial Connection*, or Chapter 4, *Configuring the 1203-GU6 or 1336-GM6 Adapter Using RSNetWorx for DeviceNet*.

#	Name	Valid Values/ Settings	Default	Writable	Reset or Power Cycle	Description
27	*COS Feedback Change*	0 – 65535	0 = disable	Yes	No	Determines how much (+/-) the feedback word can change before a message is sent during COS operations.
28	*COS/Cyc Interval*	N/A	N/A	No	No	Displays the interval used by the controller to check for data in the adapter during COS or Cyclic allocation.
29	*Peer A Input*	Off, Cmd/Ref DL A Input DL B Input DL C Input DL D Input	Off	Yes	No	Determines where the peer A input is sent in the SCANport product. **Important:** This parameter cannot be changed when the *Peer Inp Enable* (36) parameter is On.
30	*Peer B Input*	Off Cmd/Ref DL A Input DL B Input DL C Input DL D Input	Off	Yes	No	Determines where the peer B input is sent in the SCANport product. **Important:** This parameter cannot be changed when the *Peer Inp Enable* (36) parameter is On.
31	*Peer Cmd Mask*	0 or 1 for each bit	0	Yes	Yes	Provides a mask for the Logic Command word when it is received through peer input. 0 = Off (input received from Master) 1 = On (input received from Peer)
32	*Peer Ref Adjust*	0-200.00%	0	Yes	No	Provides the percentage of the Reference value received through peer input that will be applied to the SCANport reference value.
33	*Peer Flt Action*	Fault Zero Data Hold Last Fault Cfg	Fault	Yes	No	Determines the action the adapter should instruct the SCANport product to take if the adapter does not receive peer input in the allowed time. ⚠ **ATTENTION:** Risk of severe bodily injury or equipment damage exists. The *Peer Flt Action* (33) parameter allows the user to change the default configuration that would allow the module and associated drive to continue to operate if communication is lost. Precautions should be taken to assure that your settings for these parameters and your application do not create a hazard of bodily injury or equipment damage.
34	*Peer Node to Inp*	0-63	0	Yes	No	Determines the node address of the node producing I/O for the adapter to receive. **Important:** This parameter cannot be changed when the *Peer Inp Enable* (36) parameter is On.

Important: For information on accessing and editing parameters, refer to Chapter 3, *Configuring the 1203-GU6 Using a Serial Connection*, or Chapter 4, *Configuring the 1203-GU6 or 1336-GM6 Adapter Using RSNetWorx for DeviceNet*.

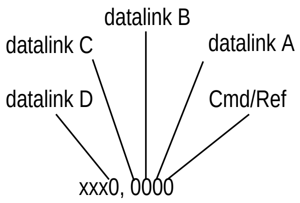
#	Name	Valid Values/ Settings	Default	Writable	Reset or Power Cycle	Description
35	*Peer Inp Time-out*	0.01-180.00 sec	10.00	Yes	No	Determines the time out time. If the adapter does not receive input from the peer node in this amount of time, it will do what is selected in *Peer Flt Action* (33) parameter.
36	*Peer Inp Enable*	Off, On	Off	Yes	No	Off = Disables peer input communications. On = Enables peer input communications.
37	*Peer Inp Status*	Off Waiting Running Faulted	NA	No	No	Displays the status of the consumed peer input connection.
38	*Peer A Output*	Off Cmd/Ref DL A Input DL B Input DL C Input DL D Input DL A Output DL B Output DL C Output DL D Output	Off	Yes	No	Determines the source of peer A output data in the SCANport product. **Important:** This parameter cannot be changed when the *Peer Out Enable* (40) parameter is On.
39	*Peer B Output*	Off Cmd/Ref DL A Input DL B Input DL C Input DL D Input DL A Output DL B Output DL C Output DL D Output	Off	Yes	No	Determines the source of peer B output data in the SCANport product. **Important:** This parameter cannot be changed when the *Peer Out Enable* (40) parameter is On.
40	*Peer Out Enable*	On, Off	Off	Yes	No	Off = Disables peer output communications. On = Enables peer output communications.
41	*Peer Out Time*	0.01 – 10.00	1.00	Yes	No	Determines the minimum interval of time between peer transmissions.
42	*Peer Out Skip*	1 – 16	1	Yes	No	Determines the maximum interval of time between peer transmissions by multiplying this value by the value in the *Peer Out Time* (41) parameter.

Important: For information on accessing and editing parameters, refer to Chapter 3, *Configuring the 1203-GU6 Using a Serial Connection*, or Chapter 4, *Configuring the 1203-GU6 or 1336-GM6 Adapter Using RSNetWorx for DeviceNet*.

M-S Input Parameter Configurations

The *M-S Input* (24) parameter has the following five configurable bits.

Figure B.5 Bits and Corresponding I/O



0 = Peer or other input
1 = Master-Slave input

When you enable *Cmd/Stat* (4) or *datalink* (5-8) parameter(s) in the adapter, you must set the corresponding bit in the *M-S Input* (24) parameter if you want the input data to come from the scanner or master device.

The following table lists possible configurations for the *M-S Input* (24) parameter and the types of allocation associated with each.

ADAPTER CONFIGURATION		ALLOCATION (Number Of Words)				
		Data Size Sent From The Controller To The Adapter				
M-S Input	**M-S Output**	**Poll Only**	**COS Only**	**Cyclic Only**	**Poll & COS**	**Poll & Cyclic**
00000	xxxx0	0	NA	NA	NA	NA
00001	xxxx0	2	NA	NA	NA	NA
00010	xxxx0	2	NA	NA	NA	NA
00011	xxxx0	4	NA	NA	NA	NA
00100	xxxx0	2	NA	NA	NA	NA
00101	xxxx0	4	NA	NA	NA	NA
00110	xxxx0	4	NA	NA	NA	NA
00111	xxxx0	6	NA	NA	NA	NA
01000	xxxx0	2	NA	NA	NA	NA
01001	xxxx0	4	NA	NA	NA	NA
01010	xxxx0	4	NA	NA	NA	NA
01011	xxxx0	6	NA	NA	NA	NA
01100	xxxx0	4	NA	NA	NA	NA
01101	xxxx0	6	NA	NA	NA	NA
01110	xxxx0	6	NA	NA	NA	NA
01111	xxxx0	8	NA	NA	NA	NA
10000	xxxx0	2	NA	NA	NA	NA
10001	xxxx0	4	NA	NA	NA	NA
10010	xxxx0	4	NA	NA	NA	NA
10011	xxxx0	6	NA	NA	NA	NA

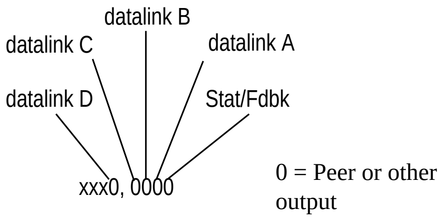
ADAPTER CONFIGURATION		ALLOCATION (Number Of Words)				
		Data Size Sent From The Controller To The Adapter				
M-S Input	M-S Output	Poll Only	COS Only	Cyclic Only	Poll & COS	Poll & Cyclic
10100	xxxx0	4	NA	NA	NA	NA
10101	xxxx0	6	NA	NA	NA	NA
10110	xxxx0	6	NA	NA	NA	NA
10111	xxxx0	8	NA	NA	NA	NA
11000	xxxx0	4	NA	NA	NA	NA
11001	xxxx0	6	NA	NA	NA	NA
11011	xxxx0	8	NA	NA	NA	NA
11100	xxxx0	6	NA	NA	NA	NA
11101	xxxx0	8	NA	NA	NA	NA
11110	xxxx0	8	NA	NA	NA	NA
11111	xxxx0	10	NA	NA	NA	NA
00000	xxxx1	0	0	0	0/0	0/0
00001	xxxx1	2	2	2	2/0	2/0
00010	xxxx1	2	2	2	2/0	2/0
00011	xxxx1	4	4	4	4/0	4/0
00100	xxxx1	2	2	2	2/0	2/0
00101	xxxx1	4	4	4	4/0	4/0
00110	xxxx1	4	4	4	4/0	4/0
00111	xxxx1	6	6	6	6/0	6/0
01000	xxxx1	2	2	2	2/0	2/0
01001	xxxx1	4	4	4	4/0	4/0
01010	xxxx1	4	4	4	4/0	4/0
01011	xxxx1	6	6	6	6/0	6/0
01100	xxxx1	4	4	4	4/0	4/0
01101	xxxx1	6	6	6	6/0	6/0
01110	xxxx1	6	6	6	6/0	6/0
01111	xxxx1	8	8	8	8/0	8/0
10000	xxxx1	2	2	2	2/0	2/0
10001	xxxx1	4	4	4	4/0	4/0
10010	xxxx1	4	4	4	4/0	4/0
10011	xxxx1	6	6	6	6/0	6/0
10100	xxxx1	4	4	4	4/0	4/0
10101	xxxx1	6	6	6	6/0	6/0
10110	xxxx1	6	6	6	6/0	6/0
10111	xxxx1	8	8	8	8/0	8/0
11000	xxxx1	4	4	4	4/0	4/0
11001	xxxx1	6	6	6	6/0	6/0

ADAPTER CONFIGURATION		ALLOCATION (Number Of Words)				
		Data Size Sent From The Controller To The Adapter				
M-S Input	M-S Output	Poll Only	COS Only	Cyclic Only	Poll & COS	Poll & Cyclic
11011	xxxx1	8	8	8	8/0	8/0
11100	xxxx1	6	6	6	6/0	6/0
11101	xxxx1	8	8	8	8/0	8/0
11110	xxxx1	8	8	8	8/0	8/0
11111	xxxx1	10	10	10	10/0	10/0

M-S Output Parameter Configurations

The *M-S Output* parameter has the following five configurable bits.

Figure B.6 Bits and Corresponding I/O



When you enable *Cmd/Stat* (4) or *datalink* (5-8) parameter(s) in the adapter, you must set the corresponding bit in the *M-S Output* (25) parameter if you want the output data to be sent to the scanner or master device.

The following table lists possible configurations for the *M-S Output* (25) parameter and the types of allocation associated with each.

M-S Output	ALLOCATION (Number Of Words)				
	Data Size Sent From The Adapter To The Controller				
	Poll Only	COS Only	Cyclic Only	Poll & COS	Poll & Cyclic
00000	0	NA	NA	NA	NA
00010	2	NA	NA	NA	NA
00100	2	NA	NA	NA	NA
00110	4	NA	NA	NA	NA
01000	2	NA	NA	NA	NA
01010	4	NA	NA	NA	NA
01100	4	NA	NA	NA	NA
01110	6	NA	NA	NA	NA
10000	2	NA	NA	NA	NA

M-S Output	ALLOCATION (Number Of Words) Data Size Sent From The Adapter To The Controller				
	Poll Only	COS Only	Cyclic Only	Poll & COS	Poll & Cyclic
10010	4	NA	NA	NA	NA
10100	4	NA	NA	NA	NA
10110	6	NA	NA	NA	NA
11000	4	NA	NA	NA	NA
11010	6	NA	NA	NA	NA
11100	6	NA	NA	NA	NA
11110	8	NA	NA	NA	NA
00001	2	2	2	2/2	2/2
00011	4	2	2	4/2	4/2
00101	4	2	2	4/2	4/2
00111	6	2	2	6/2	6/2
01011	6	2	2	6/2	6/2
01101	6	2	2	6/2	6/2
01111	8	2	2	8/2	8/2
10001	4	2	2	4/2	4/2
10011	6	2	2	6/2	6/2
10101	6	2	2	6/2	6/2
10111	8	2	2	8/2	8/2
11001	6	2	2	6/2	6/2
11011	8	2	2	8/2	8/2
11101	8	2	2	8/2	8/2
11111	10	2	2	10/2	10/2

Notes:

DeviceNet Objects

Appendix Objectives

Appendix C defines the DeviceNet object classes, class services, and attributes that are supported by the Enhanced DeviceNet adapter. These objects can be used to develop programs for the module.

This appendix assumes that you have experience in object programming.

Object Classes

The Enhanced DeviceNet adapter supports the following object classes:

Class	Object	Page
0x01	Identity	C-2
0x02	Message Router	C-4
0x03	DeviceNet	C-5
0x05	Connection	C-6
0x07	Register	C-8
0x0F	Parameter	C-10
0x10	Parameter Group	C-16
0x93	SCANport Pass-Through Parameter	C-18
0x97	SCANport Pass-Through Fault Queue	C-19
0x98	SCANport Pass-Through Warning Queue	C-21
0x99	SCANport Pass-Through Link	C-23
0x67	PCCC Object	C-25

Class Code 0x01 — Identity Object

The identity object provides identification and general information about the device.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.

Instances

The total number of instances depends on the number of microprocessors in the SCANport product connected to the module.

- Instance 1 includes information on both the adapter and the product.
- The instances for the SCANport product's microprocessors start at instance 2.
- The instance for the adapter is present after all the instances for the SCANport product's microprocessors.

Instance	Description
1	Total Product
2 through n - 1[①]	Product components
n [①]	Enhanced DeviceNet Adapter

[①] The value of n is the maximum instance in the object. This value is obtainable via class attribute 2.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Vendor ID	UINT	Identification of each vendor by number. 1 = Allen-Bradley
2	Get	Device Type	UINT	Indication of general type of product. 0x69 = Sub-Component 0x6F = SCANport Device
3	Get	Product Code	UINT	Identification of a particular product of an individual vendor. 0xXX02 = 1336 PLUS 0.5 – 10 HP -S/B 0xXX02 0xXX03 = 1336 PLUS 7.5 – 800 HP -S/B 0xXX03 0xXX07 = 1336 PLUS II -S/B 0xXX07 0xXX10 = 1336 FORCE w/ PLC Adapter -S/B 0xXX10 0xXX11 = 2364F RGU -S/B 0xXX11 0xXX12 = 1394 Motion Drive -S/B 0xXX12 0xXX13 = 1557 Medium Voltage AC Drive -S/B 0xXX13 0xXX14 =193 SMP-3 -S/B 0xXX14 0xXX15 = 150 SMC Dialog Plus -S/B 0xXX15 0xXX17 = 1305 AC Drive -S/B 0xXX17 0xXX18 = 1397 DC Drive -S/B 0xXX18 0xXX19 = 1336 VSC -S/B 0xXX19 0xXX20 = 1336T Force w/ Std Adapter -S/B 0xXX20 0xXX22 = 1336 IMPACT -S/B 0xXX22 Note: The high byte of each code indicates a particular size or configuration within a product family.
4	Get	Revision	STRUCT of	Revision of the item that this instance of the Identity Object represents. Value varies based on product.
		Major Revision	USINT	
		Minor Revision	USINT	
5	Get	Status	WORD	Summary status of product. Value varies based on product.
6	Get	Serial Number	UDINT	Serial number of product. Value varies based on product.
7	Get	Product Name	SHORT_ STRING	Human readable identification. Value varies based on product.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x05	Yes	No	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	No	Set_Attribute_Single
0x11	Yes	N/A	Find_Next_Obj_Instance

Get_Attribute_All Response

None supported.

Class Code 0x02 — Message Router Object

The Message Router Object provides a messaging connection point through which a client may address to any object class or instance residing in the physical devices.

Class Attributes

Not supported.

Instances

Instance	Description
1	Message Router Object

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
2	Get	Number available	UINT	Maximum number of connections supported by the message router.
3	Get	Number active	UINT	Number of connections currently used by system components.
4	Get	Active connections	ARRAY of UINT	A list of the connection IDs of the currently active connections. This attribute not used. Reserved for compatibility purposes.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single

Class Code 0x03 — DeviceNet Object

The DeviceNet Object is used to provide the configuration and status of a physical attachment to DeviceNet. A product must support one (and only one) DeviceNet Object per physical network attachment.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	DeviceNet Specification	Word	Returns 2

Instances

Not supported.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Mac ID	USINT	Node Address
2	Get	Data Rate	USINT	Data Rate
3	Get	BOI	BOOL	Default = 0
4	Set	Bus Off Counter	USINT	Increments if BOI is non-zero and Bus Off occurs. Can only be set to zero.
5	Get	Allocation Information	STRUCT of BYTE USINT	The allocation information a slave supports when the master allocates.
6	Get	Node Adx Switch Err	BOOL	If non-zero, the Node Address NVS value does not match the online value.
7	Get	Data Rate Switch Err	BOOL	If non-zero, the Data Rate NVS value does not match the online value.
8	Get	Node Adx Switch	USINT	The actual value in the EEPROM
9	Get	Data Rate Switch Val	USINT	The actual value in the EEPROM or the operating value after an autobaud was completed.

Common Services

Not supported.

Class Code 0x05 — Connection

The Connection Class allocates and manages the internal resources associated with both I/O and Explicit Messaging Connections. The specific instance generated by the Connection Class is referred to as a *Connection Instance* or a *Connection Object*.

Important: An externally visible interface to the Connection Class across Explicit Messaging Connections **DOES** exist. Unless otherwise noted, all services/attributes noted in the following sections are accessible using Explicit Messaging.

A Connection Object within a particular module actually represents one of the end-points of a Connection. It is possible for one of the Connection end-points to be configured and "active" (e.g., transmitting) without the other end-point(s) being present. Connection Objects are used to model the communication specific characteristics of a particular Application-to-Applications(s) relationship. A specific Connection Object Instance manages the communication-specific aspects related to an end-point.

A Connection Object on DeviceNet uses the services provided by a Link Producer and/or Link Consumer to perform low-level data transmission and reception functions.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	Revision of the Connection Object class definition upon which the implementation is based Range 1 – 65535

Instances

Instance	Description
1	Group 2 Messaging
2	Group 2 Polling
4	Group 2 COS/Cyclic
6	Group 3 Messaging
7	Group 3 Messaging
8	Group 3 Messaging
9	Group 3 Messaging
10	Group 3 Messaging

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	State	USINT	State of the connection as defined in the DeviceNet specification
2	Get	Instance type	USINT	Indicates I/O or Messaging connection
3	Get	Transport Class Trigger	USINT	The Transport Class Trigger for this instance
4	Get	Produced Cnxn ID	USINT	CAN Identifier to transmit on
5	Get	Consumed Cnxn ID	USINT	CAN Identifier to receive on
6	Get	Initial Comm Char	USINT	Defines the DeviceNet message groups that the tx/rx Cnxn's apply
7	Get	Produced Cnxn Size	UINT	Max bytes to transmit across this connection
8	Get	Consumed Cnxn Size	UINT	Max bytes to receive across this connection
9	Get/Set	EPR	UINT	Expected Packet Rate
12	Get/Set	Watchdog Action	USINT	How to handle inactivity/watchdog time-outs
13	Get	Produced Path Length	UINT	Number of bytes in the produced connection path attribute
14	Get	Produced Cnxn Path	ARRAY of USINT	Specifies the application object whose data is to be produced by this connection
15	Get	Consumed Path Length	UINT	Number of bytes in the consumed connection path attribute
16	Get	Consumed Cnxn Path	ARRAY of USINT	Specifies the application object to receive the data consumed by this application
17	Get/Set	Production Inhibit Time	UINT	Defines minimum time between new data production for COS connections

Common Services

Not supported.

Class Code 0x07 — Register Object

The Register Object is used to address individual bits or a range of bits. It may operate as either a producer (input) register or a consumer (output) register. A producer register object produces data onto the network. A consumer register object consumes data from the network.

Message writes to the Register Object can perform control functions. Therefore, message writes are only allowed when the controller is not actively controlling the module and the message write is done through a connection with a time-out value not equal to zero. Writes cannot be performed through an unconnected message. After a write, any time-out or closure of the connection may cause the SCANport product to fault.

Refer to Chapter 7, *Using DeviceNet Explicit Messaging,* for information about writing to the Register Object.

Class Attributes

Not supported.

Instances

Instance	Description
1	All polled data being read from the SCANport device (read-only)
2	All polled data written to the SCANport device (read/write)
3	Logic Status and Feedback data (read-only)
4	Logic Command and Reference data (read/write)
5	Datalink A input data (read-only)
6	Datalink A output data (read/write)
7	Datalink B input data (read-only)
8	Datalink B output data (read/write)
9	Datalink C input data (read-only)
10	Datalink C output data (read/write)
11	Datalink D input data (read-only)
12	Datalink D output data (read/write)
13	Logic Status and Feedback Data (read-only)
14	Logic Command and Reference Data[①] (read/write)

[①] The command word is set to the value of the first word of the data where there are ones in the second word of the data. Command = (word 1 and not word 2) or (word 1 and word 2). This only controls specified bits in the logic command data to the SCANport product and does not change the reference value.

Instance Attributes

Setting of an assembly attribute can only be accomplished through a connection. This feature is to prevent accidental control of the SCANport product.

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Bad Flag	BOOL	If set to 1, then attribute 4 may contain invalid, bad or otherwise corrupt data. 0 = good 1 = bad
2	Get	Direction	BOOL	Direction of data transfer 0 = Producer Register 1 = Consumer Register
3	Get	Size	UINT	Size of register data in bits
4	Conditional[①]	Data	ARRAY of BITS	Data to be transferred

[①] The access rule of Set is optional if attribute 2, Direction = 1. If Direction = 0, the access rule is Get.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Class Code 0x0F — Parameter Object

The Parameter Object provides a known, public interface for device configuration data. This object also provides all the information necessary to define and describe each individual configuration parameter of a device.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	Revision of this object. First revision, value = 1.
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.
8	Get	Parameter Class Descriptor	WORD	Bits that describe parameters.
9	Get	Configuration Assembly Instance	UINT	Instance number of the configuration assembly. This attribute is set to zero because a configuration assembly is not supported.
10	Set	Native Language	USINT	Language ID for all character array accesses. 0 = English 1 = French 2 = Spanish 3 = Italian 4 = German 5 = Japanese 6 = Portuguese

Instances

The number of instances varies based on the number of parameters in your SCANport product. The adapter parameters immediately follow the SCANport product parameters.

Instance	Description
1 through n - 42[①]	SCANport Product Parameters
n - 41 through n [①]	Module Parameters

[①] The value of n is the maximum instance in the object. This value is obtainable via class attribute 2.

Instance Attributes

Attribute ID	Access Rule	Stub/Full	Name	Data Type	Description
1	[1]	Stub	Parameter Value	Specified in Descriptor, Data Type and Data Size attributes.	Actual value of parameter. Data type specified in descriptor, data type, and data size.[1][2]
2	Get	Stub	Link Path Size	USINT	Size of Link Path attribute. If this attribute is 0, then no link is specified. Number of BYTEs in attribute 3.
3	Get	Stub	Link Path	ARRAY of path segments	Path to the object from where this parameter value is retrieved. The link path is limited to 255 BYTEs.
			Segment type/port	BYTE	
			Segment Address	Path (format depends on data contained in segment type/port)	
4	Get	Stub	Descriptor	WORD	Descriptor of parameter. Refer to the table on page C-12.
5	Get	Stub	Data Type	USINT	Data type code. Refer to the table on page C-13.
6	Get	Stub	Data Size	USINT	Number of BYTEs in attribute 1, Parameter Value.
7	Get	Full	Parameter Name String	SHORT_ STRING	A human readable string representing the parameter name. For example, “frequency #1.” The maximum number of characters is 16. (The first byte is a length code.)
8	Get	Full	Units String	SHORT_ STRING	Engineering unit string. The maximum number of characters is 4. (The first byte is a length code.)
9	Get	Full		SHORT_ STRING	The maximum number of characters is 64. (The first byte is a length code.) Always returns 0.
10	Get	Full	Minimum Value	Same as attribute 1	The minimum valid actual value to which attribute 1, Parameter Value can be set.
11	Get	Full	Maximum Value	Same as attribute 1	The maximum valid actual value to which attribute 1, Parameter Value can be set.
12	Get	Full	Default Value	Same as attribute 1	The actual value attribute 1, Parameter Value should be set to when the user wants the default for the parameter.
13	Get	Full	Scaling Multiplier	UINT	Multiplier for scaling formula.
14	Get	Full	Scaling Divisor	UINT	Divisor for scaling formula.

[1] The access rule is defined in bit 4 of instance attribute 4, the Descriptor. If bit 4 is 0 the access rule is Set and the Parameter Value can be read and written. If bit 4 is 1, the access rule is Get and the Parameter Value can only be read.

[2] Data type specified in instance attributes 4 (Descriptor), 5 (Data Type) and 6 (Data Size).

Attribute ID	Access Rule	Stub/Full	Name	Data Type	Description
15	Get	Full	Scaling Base	UINT	Base for scaling formula.
16	Get	Full	Scaling Offset	UINT	Offset for scaling formula.
17	Get	Full	Multiplier Link	UINT	Parameter object instance number of multiplier source.
18	Get	Full	Divisor Link	UINT	Parameter object instance number of base source.
19	Get	Full	Base Link	UINT	Parameter object instance number of offset source.
20	Get	Full	Offset Link	UINT	Parameter object instance number of offset source.
21	Get	Full	Decimal Precision	USINT	Specifies number of decimal places to use when displaying the scaled engineering value. Also used to determine actual increment value so that incrementing a value causes a change in scaled engineering value to this precision.

① The access rule is defined in bit 4 of instance attribute 4, the Descriptor. If bit 4 is 0 the access rule is Set and the Parameter Value can be read and written. If bit 4 is 1, the access rule is Get and the Parameter Value can only be read.

② Data type specified in instance attributes 4 (Descriptor), 5 (Data Type) and 6 (Data Size).

Bit Definitions for Instance Attribute 4

Bit	Definition	Value
0	Supports settable path	0 = Link path can not be set. 1 = Link path can be set.
1	Supports enumerated strings	0 = Enumerated strings are not supported. 1 = Enumerated strings are supported and may be read with the Get_Enum_String service.
2	Supports scaling	0 = Scaling not supported. 1 = Scaling is supported. The scaling attributes are implemented and the value presented is in engineering units.
3	Supports scaling links	0 = Scaling links not supported. 1 = The values for the scaling attributes may be retrieved from other parameter object instances.
4	Read only parameter	0 = Parameter value attribute can be written (set) and read (get). Access rule is set. 1 = Parameter value attribute can only be read. Access rule is get.
5	Monitor parameter	0 = Parameter value attribute is not updated in real time by the device. 1 = Parameter value attribute is updated in real time by the device.
6	Supports extended precision scaling	0 = Extended precision scaling is not supported. 1 = Extended precision scaling should be implemented and the value presented to the user in engineering units.

Data Types for Instance Attribute 5

Attribute ID Value	Definition	Data Type Description	Scaling Supported on this Data Type
1	WORD	16-bit word	No
2	UINT	16-bit unsigned integer	Yes
3	INT	16-bit signed integer	Yes
4	BOOL	Boolean	No
5	SINT	Short integer	Yes
6	DINT	Double integer	Yes
7	LINT	Long integer	Yes
8	USINT	Unsigned short integer	Yes
9	Not Supported	Unsigned double integer	Yes
10	Not Supported	Unsigned long integer	Yes
11	Not Supported	Single floating point format (IEEE 754)	Yes
12	Not Supported	Double floating point format (IEEE 754)	Yes
13	Not Supported	Duration (short)	Yes
14	Not Supported	Duration	Yes
15	Not Supported	Duration (high resolution)	Yes
16	Not Supported	Duration (long)	Yes
17	Not Supported	Date	No
18	Not Supported	Time of Day	No
19	Not Supported	Date and time	No
20	Not Supported	8-bit per character string	No
21	Not Supported	16-bit per character string	No
22	Not Supported	N-byte per character string	No
23	Not Supported	Short N-byte character string	No
24	Not Supported	8-bit string	No
25	Not Supported	32-bit string	No
26	Not Supported	64-bit string	No

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x01	No	Yes	Get_Attribute_All
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single
0x4B	No	Yes	Get_Enum_String
0x05	Yes	No	Reset_Request (sets all parameters to default values)
0x15	Yes	No	Restore_Request
0x16	Yes	No	Save_Request

Get_Attribute_All Response

Not supported.

At the instance level, the order of attributes returned in the Get_Attributes_All response is as follows:

Class Attribute ID	Attribute Name and Default Value
1	Parameter Value
2	Link Path Size
3	Link Path
4	Descriptor
5	Data Type
6	Data Size
7	Parameter Name String, default character count = 0
8	Units String, default character count = 0
9	Help String, default character count = 0
10	Minimum Value default = 0
11	Maximum Value default = 0
12	Default Value default = 0
13	Scaling Multiplier Default = 1
14	Scaling Divisor Default = 1
15	Scaling Base Default = 1
16	Scaling Offset Default = 0
17	Multiplier Link Default = 0
18	Divisor Link Default = 0
19	Base Link Default = 0
20	Offset Link Default = 0

Class Attribute ID	Attribute Name and Default Value
21	Decimal Precision Default = 0

Object Specific Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x4B	No	Yes	Get_Enum_String

Enumerated strings are human-readable strings that describe either a bit or a value depending on the data type of instance attribute 1, the Parameter Value. If the data type is a BYTE or WORD, the enumerated string is a bit enumerated string. If the data type is INT or UINT, the enumerated string is a value enumerated string. Any other data type does not have enumerated strings.

The table below lists the parameters for the Get_Enum_String request service.

Name	Data Type	Description of Attribute
Enumerated String Number	USINT	Number of enumerated string to retrieve (MAX value is 255).

- If the string to be returned is a bit enumerated string, then the enumerated string number represents a bit position and the Get_Enum_String service returns a string from that bit.
- If the string to be returned is a value enumerated string, then the enumerated string number represents a value and the Get_Enum_String service returns a string for that value.

The enumerated string is returned in the form of a SHORT_STRING and is 16 characters long plus the preceding length byte.

Class Code 0x10 — Parameter Group Object

The Parameter Group Object identifies and provides access to groups of parameters in a device grouping. The Parameter Group Object provides convenient access to related sets of parameters.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Parameter group version	UINT	Returns 1
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.
8	Get	Native Language	USINT	Language ID for all STRING accesses. 0 = English 1 = French 2 = Spanish (Mexican) 3 = Italian 4 = German 5 = Japanese 6 = Portuguese

Instances

The number of instances varies based on the number of groups in the SCANport product. One additional group is added for the module.

Instance	Description
1 – (n - 1)	SCANport product groups
n[①]	Module group

[①] n is the value returned by a get from class attribute 2 (max instance).

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Group Name String	SHORT_ STRING	A human-readable string representing the group name (e.g., set-up, frequency set). Maximum number of characters = 16
2	Get	Number of Members in Group	UINT	Number of parameters in group.
3	Get	1st Parameter Number in Group	UINT	Parameter instance number.
4	Get	2nd Parameter Number in Group	UINT	Parameter instance number.
n	Get	(n-2)th Parameter Number in Group	UINT	Parameter instance number.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single

Get_Attribute_All Response

Not supported.

Class Code 0x93 — SCANport Pass-Through Parameter Object

The SCANport Pass-Through Parameter Object lets you perform a scattered read or write.

Class Attributes

Not supported.

Instance Attributes

Not supported.

Common Services

Not supported.

Object-Specific Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x32	Yes	No	Scattered_Parameter_ Value_Read[1]
0x34	Yes	No	Scattered_Parameter_ Value_Write[1]

[1] Must be directed to Attribute 0, Instance 0.

The table below lists the parameters for the Scattered_Parameter_ Value_Read and Scattered_Parameter_Value_Write object-specific services:

Name	Data Type	Description
Scattered Parameters	STRUCT of	
Parameter Number	WORD	Parameter to read or write
Parameter Value	WORD	Parameter value to write (zero when reading)

Important: The STRUCT may repeat up to 32 times in a single message.

Class Code 0x97 — SCANport Pass-Through Fault Object

The SCANport Pass-Through Fault Object provides information on the product’s fault queue.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Set	Write Fault Command	BYTE	1 = Clear Faults 2 = Clear Fault Queue 3 = Reset Product
1	Get	Read Number of Fault Queue Entries	BYTE	Reads the number of fault queue entries.
2	Get	Read Fault Queue Trip Index	BYTE	Reads the index of the fault that tripped the product.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Read Fault Queue Entry Full/All Info	STRUCT of	
		Fault Text	ARRAY of BYTE	16 character string (no length information, no terminating null).
		Fault Code	WORD	Fault Code.
		Fault Time Stamp	STRUCT	
			BYTE	1/100 Second (0 – 99).
			BYTE	Second (0 – 59).
			BYTE	Minute (0 – 59).
			BYTE	Hour (0 – 23).
			BYTE	Day of Week (0 – 6).[1]
			BYTE	Date (1 – 31).
			BYTE	Month (1 – 12).
			BYTE	Year (0 – 99[2]).
128	Get	Fault Code and Time Stamp	STRUCT of	
		Fault Code	WORD	Fault Code.
		Fault Time Stamp	STRUCT of	
			BYTE	1/100 Second (0 – 99).
			BYTE	Second (0 – 59).
			BYTE	Minute (0 – 59).
			BYTE	Hour (0 – 23).
			BYTE	Day of Week (0 – 6).[1]
			BYTE	Date (1 – 31).
			BYTE	Month (1 – 12).
			BYTE	Year (0 – 99[2]).
129	Get	Read Fault Text String Only	ARRAY of BYTE	16 character string (no length information, no terminating null).

[1] Sunday is a value of zero.

[2] Year is an offset from 1990.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Class Code 0x98 — SCANport Pass-Through Warning Object

The SCANport Pass-Through Warning Object provides information on the product’s warning queue.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Set	Write Warning Command	BYTE	Write Warning Command. 1 = Clear Warnings 2 = Clear Warning Queue 3 = Reset Product
1	Set	Read Number of Warning Queue Entries	BYTE	

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Read Warning Queue Entry Full/All Info	STRUCT of	
		Warning Text	ARRAY of BYTE	16 character string (no length information, no terminating null).
		Warning Code	WORD	Fault Code.
		Warning Time Stamp (Time Stamps not available in all products)	STRUCT	
			BYTE	1/100 Second (0 – 99).
			BYTE	Second (0 – 59).
			BYTE	Minute (0 – 59).
			BYTE	Hour (0 – 23).
			BYTE	Day of Week (0 – 6).[1]
			BYTE	Date (1 – 31).
			BYTE	Month (1 – 12).
			BYTE	Year (0 – 99[2]).

[1] Sunday is a value of zero.

[2] Year is an offset from 1990.

Attribute ID	Access Rule	Name	Data Type	Description
128	Get	Warning Code and Time Stamp (Time Stamps not available in all products)	STRUCT of	
		Warning Code	WORD	Fault Code.
		Warning Time Stamp (Time Stamps not available in all products)	STRUCT of	
			BYTE	1/100 Second (0 – 99).
			BYTE	Second (0 – 59).
			BYTE	Minute (0 – 59).
			BYTE	Hour (0 – 23).
			BYTE	Day of Week (0 – 6).[1]
			BYTE	Date (1 – 31).
			BYTE	Month (1 – 12).
			BYTE	Year (0 – 99[2]).
129	Get	Read Warning Text String Only	ARRAY of BYTE	16 character string (no length information, no terminating null).

[1] Sunday is a value of zero.

[2] Year is an offset from 1990.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Class Code 0x99 — SCANport Pass-Through Link Object

The SCANport Pass-Through Link Object lets you perform a scattered read or write of a number of links or a single read or write of a link.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Set	Link Command	BYTE	1 = Clear all links.
1	Get	NVS Link Diagnostic Value	WORD	Checksum.

Instance Attributes

An instance in this class is the number of a parameter that is to get its value from another parameter.

Attribute ID	Access Rule	Name	Data Type	Description
0	Set	Parameter Link Reference[1]	WORD	

[1] The Parameter Link Reference value is the number of the parameter whose value is to be transferred.

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Object-Specific Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x32	Yes	No	Scattered_Link_Reference_Value_Read[1]
0x34	Yes	No	Scattered_Link_Reference_Value_Write[1]

[1] Must be directed to Attribute 0, Instance 0.

The table below lists parameters for Scattered_Link_Reference_Read and Scattered_Link_Reference_Write object-specific services.

Name	Data Type	Description
Scattered Link Read/Write	STRUCT of	
Parameter Number	WORD	Parameter Link Reference to read or write.
Parameter Link Reference	WORD	Link Reference value to write (zero when reading).

Important: The STRUCT may repeat up to 32 times in a single message.

Class Code 0x67 — PCCC Object

The PCCC Object is used to process encapsulated PCCC messages from DeviceNet. The PCCC Object does not implement any specific class or instance attributes, so the instance field for any received messages is ignored.

Class Attributes

Not supported.

Instance Attributes

Not supported.

Common Services

Not supported.

Object Specific Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x4B	No	Yes	Execute_PCCC
0x4D	No	Yes	Execute_Local_PCCC

Message Structure for Execute_PCCC

Request			Response		
Name	Data Type	Description	Name	Data Type	Description
Length	USINT	Length of requestor ID	Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor	Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor	Serial Number	UDINT	ASA serial number of requestor
Other	Product Specific	Identifier of user, task, etc. on the requestor	Other	Product Specific	Identifier of user, task, etc. on the requestor
CMD	USINT	Command byte	CMD	USINT	Command byte
STS	USINT	0	STS	USINT	Status byte
TNSW	UINT	Transport word	TNSW	UINT	Transport word. Same value as the request.
FNC	USINT	Function code. Not used for all CMD's.	EXT_STS	USINT	Extended status. Not used for all CMD's.
PCCC_params	ARRAY of USINT	CMD/FNC specific parameters	PCCC_results	ARRAY of USINT	CMD/FNC specific result data

Message Structure for Execute_Local_PCCC

Request			Response		
Name	Data Type	Description	Name	Data Type	Description
CMD	USINT	Command byte	CMD	USINT	Command byte
STS	USINT	0	STS	USINT	Status byte
TNSW	UINT	Transport word	TNSW	UINT	Transport word. Same value as the request.
FNC	USINT	Function code. Not used for all CMD's.	EXT_STS	USINT	Extended status. Not used for all CMD's.
PCCC_params	ARRAY of USINT	CMD/FNC specific parameters	PCCC_results	ARRAY of USINT	CMD/FNC specific result data

More Information

For more information on PCCC messages, refer to Appendix D, *Supported PCCC Messages.*

Supported PCCC Messages

Appendix Objectives

Appendix D describes the PCCC object, which allows existing PLC-type controllers to connect to the DeviceNet network. In this appendix, you will learn about the following:

- DF-1/PCCC support (1203-GU6 only).
- A list of supported PCCC messages.

DF-1/PCCC Support (1203-GU6 Only)

The 1203-GU6 translates PCCC messages into SCANport messages between the SCANport device and the module. To communicate to the SCANport device, you need:

- A PC running software that can communicate and translate the DF-1/PCCC messages supported by the module.
- A modem that supports full duplex communication (if you’re connecting the modem to a serial port that utilizes point-to-point DF-1 full duplex protocol at selected baud rates).

Supported PCCC Messages

The 1203-GU6 and 1336-GM6 support the following PCCC messages:

CMD	FNC	Message Definition	Supported?
0x06	0x03,4,5,0,1	Identify Host and Some Status	Yes
0x0F	0x67	PLC-5 Typed Write	Yes
0x0F	0x68	PLC-5 Typed Read	Yes
		Logical ASCII Address	Yes
		Logical Binary Address	No
		PLC-2 System Address	No
		PLC-3 Symbolic Address	No
0x0F	0xA1	SLC-500 Protected Typed Logical Read w/ 2 Address Fields — File, Element	No
0x0F	0xA2	SLC-500 Protected Typed Logical Read w/ 3 Address Fields — File, Element, Sub-Element	Yes
0x0F	0xA9	SLC-500 Protected Typed Logical Write w/ 2 Address Fields — File, Element	No
0x0F	0xAA	SLC-500 Protected Typed Logical Write w/ 3 Address Fields — File, Element, Sub-Element	Yes
0x0F	0xAB	SLC-500 Protected Typed Logical Write w/ 4 Address Fields — File, Element, Sub-Element, Bit Mask	No
0x0F	0x95	Encapsulate Other Protocol message	Yes

Related Documentation

For more information on PCCC messages, refer to the *DF1 Protocol and Command Set Reference Manual*, publication 1770-6.5.16.

N-File Addresses

Appendix Objectives

Appendix E provides information on the N-File addresses used when accessing the PCCC object or the DF-1 serial port. When using messages, you can use the N-file addresses to locate information about the adapter or SCANport product.

N-File Addresses

The 1203-GU-6 and 1336-GM6 support the N-file addresses shown below:

Address	N-File Addresses
N10:0	Number of SCANport product parameters
N10:1 – 999	SCANport product parameters 1 – 999 (value only)
N11:0 – 999	SCANport product parameters 1000 – 1999 (value only)
N12:0 – 999	SCANport product parameters 2000 – 2999 (value only)
N13:0	Number of SCANport adapter parameters
N13:1 – 999	SCANport adapter parameters 1 – 999 (value only)
N30:1 – 999	SCANport product parameters 1 – 999 (all information — read only)
N31:1 – 999	SCANport product parameters 1000 – 1999 (all information — read only)
N32:1 – 999	SCANport product parameters 2000 – 2999 (all information — read only)
N33:1 – 999	SCANport adapter parameters 1 – 999 (all information — read only)
N40:0 – 63	Block Transfer Emulation file
N42:5	1203-Gx2 Firmware Emulation Version — The firmware version of the 1203-Gx2 this adapter emulates for DriveTools compatibility.
N42:6	Max Network Node — The maximum DeviceNet Node Number
N42:7	Adapter Port # — The SCANport adapter port number the adapter is connected to on the SCANport product
N42:8	Reserved for future use — Always zero

Address	N-File Addresses
N50:0	Number of SCANport product parameters
N50:1 – 249	SCANport product parameters 1 – 249 (value only)
N51:0 – 249	SCANport product parameters 250 – 499 (value only)
.	
.	
.	
N61:0 – 249	SCANport product parameters 2750 – 2999 (value only)
N90:1 – 249	SCANport product parameters 1 – 249 (value only)
N91:0 – 249	SCANport product parameters 250 – 499 (value only)
.	
.	
.	
N101:0 – 249	SCANport product parameters 2750 – 2999 (value only)

Appendix **F**

Supported Emulated Block Transfer Commands

Appendix Objectives

Appendix F provides information about the Emulated Block Transfer commands supported by the DeviceNet adapter. In this appendix, you will learn about the following:

- Emulated block transfer commands.
- Emulated block transfer error response.
- Setting up data files for listed emulated block transfer commands.
- Examples of each emulated block transfer command listed.

What is Emulated Block Transfer?

Emulated block transfer is a method used by some SCANport peripherals to read and write information using PCCC messages. Some Allen-Bradley DeviceNet products can send PCCC messages. Other products can send PCCC messages using a DF-1 connection. This appendix assumes that you have experience using emulated block transfer commands with SCANport peripherals.

ATTENTION: Hazard of equipment damage exists. If explicit messages are programmed to frequently write parameter data to certain drive products, the EEPROM (Non-Volatile Storage) will quickly exceed its life cycle and cause the product to malfunction. Do not create a program that frequently uses explicit messages to write parameter data to a product. Datalinks do not write to the EEPROM and should be used for frequently changed parameters.

Supported Emulated Block Transfer Commands

The following table lists the supported emulated block transfer commands and where you can find more information on them.

Command	Page
Parameter Value Read [1]	F-3
Parameter Value Write [1]	F-4
Parameter Read Full [1]	F-5
Product ID Number Read [1]	F-8
Scattered Parameter Read Value	F-10
Scattered Parameter Write Value	F-12
NVS Functions	F-14
Fault Command Write [1]	F-15
Fault Queue Entry Read Full [1]	F-16
Fault Queue Size [1]	F-18
Trip Fault Queue Number [1]	F-19

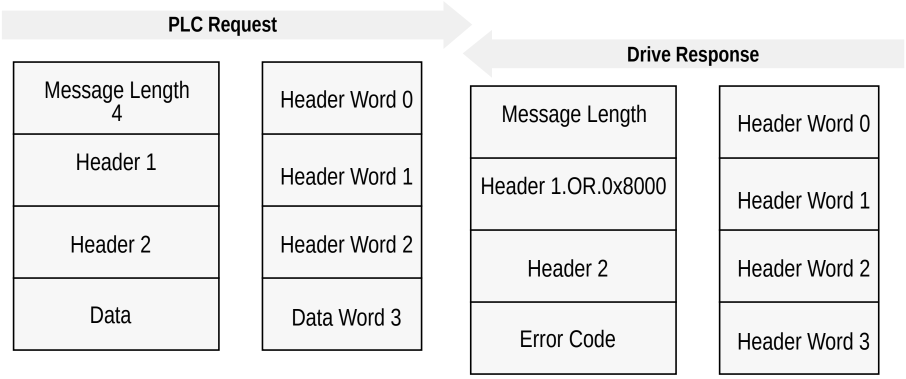
[1] This function can be accessed in the module and product. The following examples describe how to access it in the product. To do so in the module, add 16384 to the decimal value of header word 2.

Emulated Block Transfer Status Word

When an operation is unsuccessful, header word 2 of the drive response contains a negative value (bit 15 = 1).

If an error occurs, the drive also returns a status word to indicate the reason for the failure. The location of the status word is typically header word 4 in the drive response, but will depend on the message.

Figure F.1
Example Message Structure and Error Message Reply



The following table lists the error codes.

Value	Description
0	No error occurred.
1	The service failed due to an internal reason, and the drive could not perform the request.
2	The requested service is not supported.
3	An invalid value in the block transfer emulation request header word 2.
4	An invalid value in the block transfer emulation request header word 3.
5	An invalid value in the block transfer emulation request header word 2.
6	The data value is out of range.
7	There is a drive state conflict. The drive is in an incorrect state to perform the function. The drive cannot be running when you perform certain functions.

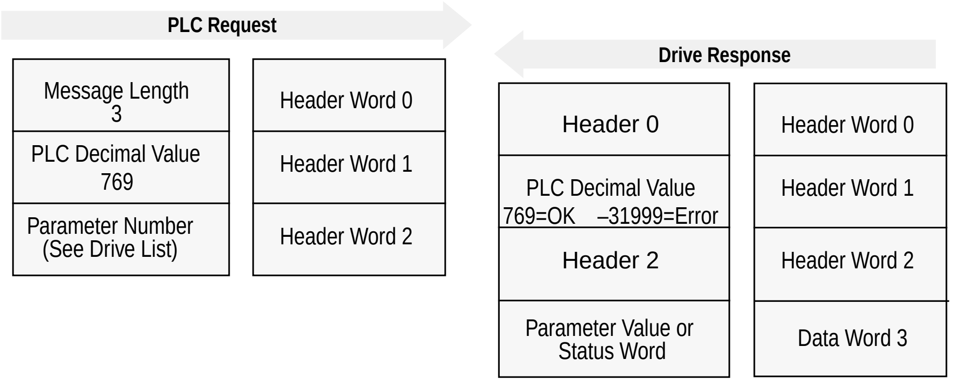
Parameter Value Read

Parameter Value Read reads the 16-bit parameter data value for the selected parameter number.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 3 words
Drive response instruction length: 1 word

Figure F.2
Message Structure



Message Operation

Parameter Value Read reads parameter values from the drive and places that value (or an error code) in word 3 of the drive response data file. The value is shown in device units. Device units are listed in the user manual for the device you are using.

If an error occurs:

- Word 3 of the response contains the status code.
- The status area of the data file is non-zero.

Example

In this example, the value of parameter 20 was requested from a 1336 PLUS drive and a value of 4096 was returned. 4096 is the internal drive unit value for *Maximum Rated Voltage*. This corresponds to a value of 100% Drive Rated Volts in Display Units.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	3	769	20*							
Drive response	6	769	20*	4069*						

* Example only — These values vary depending on parameters and products.

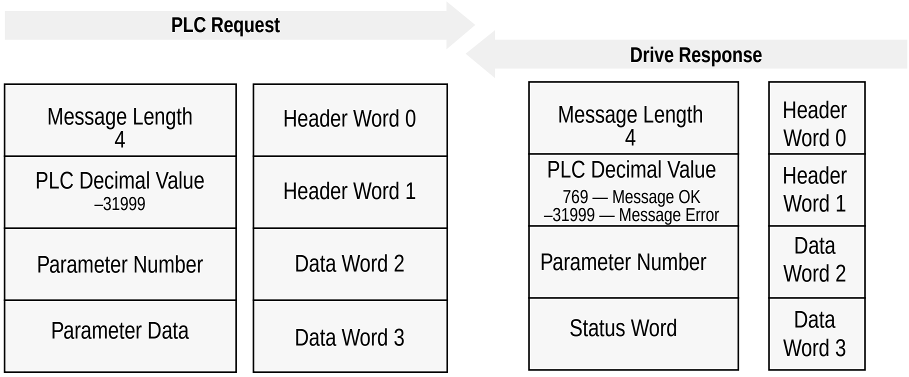
Parameter Value Write

Parameter Value Write writes a 16-bit parameter data value to the selected parameter number.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 1 word
Drive response instruction length: 4 words

Figure F.3
Message Structure



Message Operation

Parameter Value Write sends a new value to the specified parameter. The value must be in device units. Units for each parameter are listed in the device manual.

If an error has occurred, word 1 of the response returns a value of –31999, and word 3 contains a status code.

Example

In this example, a value of 4096 was sent to Parameter 20. 4096 is in drive units and indicates a value of 100% Drive Rated Volts, as defined in P147, *Drive Rated Volts*.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	4	–31999	20*	4096*						
Drive response	3	769	20*							

* Example only — These values vary depending on parameters and products.

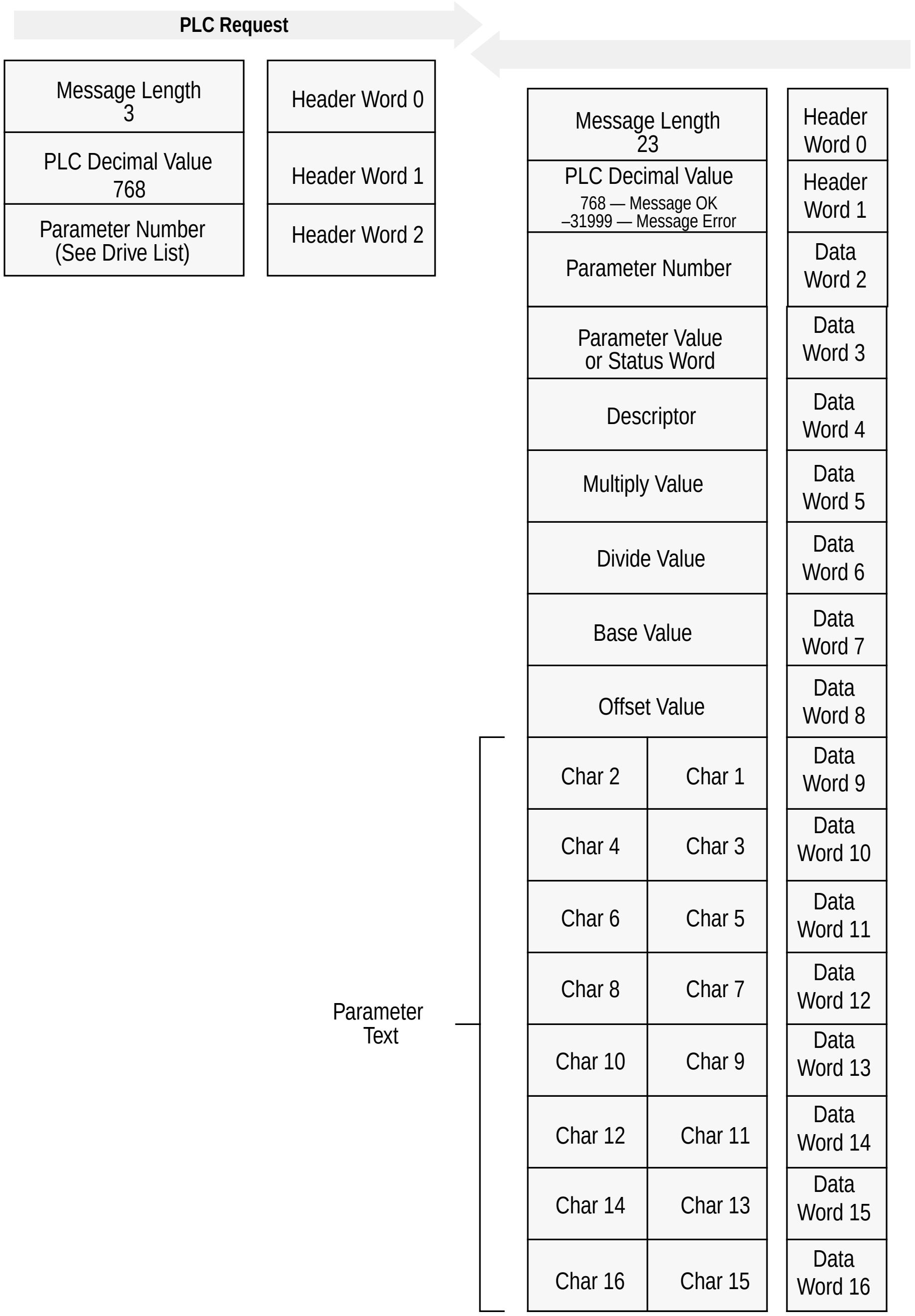
Parameter Read Full

Parameter Read Full provides all known attributes for the parameters requested. This information includes the parameter's current value, descriptor, multiply and divide value, base value, offset value, text string, group element reference, minimum value, maximum value, default value, and unit text string.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 3 words
Drive response instruction length: 20 words

Figure F.4
Message Structure

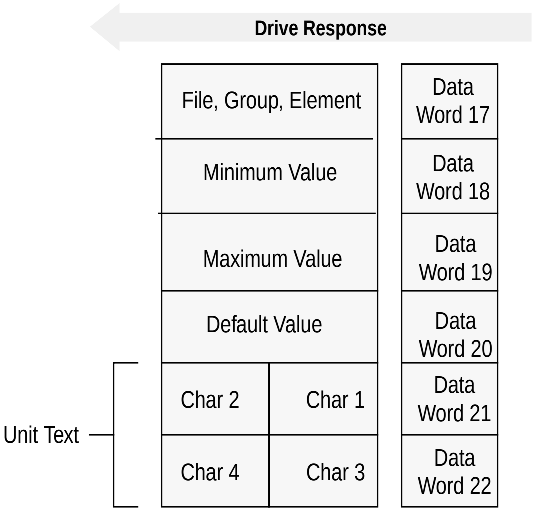


PLC Request

Message Length 3	Header Word 0
PLC Decimal Value 768	Header Word 1
Parameter Number (See Drive List)	Header Word 2

	Message Length 23		Header Word 0
	PLC Decimal Value 768 — Message OK –31999 — Message Error		Header Word 1
	Parameter Number		Data Word 2
	Parameter Value or Status Word		Data Word 3
	Descriptor		Data Word 4
	Multiply Value		Data Word 5
	Divide Value		Data Word 6
	Base Value		Data Word 7
	Offset Value		Data Word 8
Parameter Text	Char 2	Char 1	Data Word 9
	Char 4	Char 3	Data Word 10
	Char 6	Char 5	Data Word 11
	Char 8	Char 7	Data Word 12
	Char 10	Char 9	Data Word 13
	Char 12	Char 11	Data Word 14
	Char 14	Char 13	Data Word 15
	Char 16	Char 15	Data Word 16

Figure F.4
Message Structure (Continued)



Message Operation

Parameter Read Full retrieves the attributes of the specified parameter. The attributes for each parameter include the data, minimum and maximum values, and the parameter text. The response message returns this information.

If an error has occurred in reading any of the values, word 3 contains the status word.

The parameter text is returned with each data word containing two ASCII characters per word. The first and second characters are in opposite order.

Example

In this example, a **Parameter Read Full** was performed through block transfer on a 1336 PLUS drive. N10:10 shows the header message for the request. The data is returned in the response data file, starting with word 3, for parameter 20. Word 3 shows the present value in drive units. Word 4 through word 8 provide scaling information, used to convert drive units to engineering units for the Human Interface Module (HIM). Word 9 through word 16 provide the parameter name.

This example shows the response message in both binary and ASCII. Note the ASCII information beginning with word 9. The parameter name characters return in reverse order for each word. Word 9 has the ASCII value of (aM). To read this, reverse the word to read (Ma). The next word (ix), reversed, gives you (xi). These words, along with the following two words, form the word *Maximum*. You can see the parameter name *Maximum Voltage* in word 9 through word 16 of the response message. In addition, words 21 – 22 are also returned in this format. These words provide the units in which the parameter is defined. In this example it is *vlts*.

Word 17 contains the file, group, and element which are used to reference the parameter.

Words 18 – 20 contain the minimum, maximum, and default values of this parameter.

Data File Format

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PLC request	3	768	20*							
Drive response	23	768	20*	4096*	355*	1*	4096*	460*	0*	24909*
(Decimal)	27000*	30061*	8301*	28502*	29804*	26465*	8293*	1794*	1024*	4915*
	4096*	27734*	29556*							
Drive response	\00\17	\03\00	\00\14	\10\00	\01 c	\00\01	\10\00	\01\CC	\00\00	a M
(ASCII)	i x	u m	m	o V	t l	g a	e	07 02	04 00	\13 0
	\10\00	l V	s t							

* Example only — These values vary depending on parameters and products.

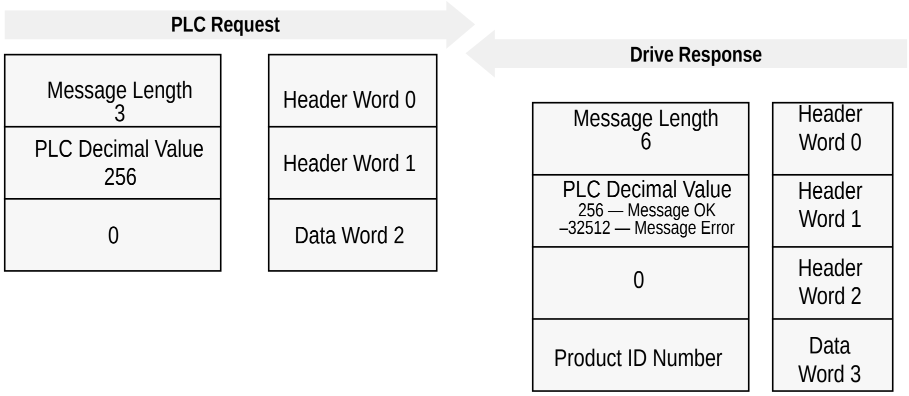
Product ID Number Read

Product ID Number Read returns the product ID of the device to which the DeviceNet adapter is connected.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 3 words
Drive response instruction length: 4 words

Figure F.5
Message Structure



Product Code (Hex)	Product Code (Decimal)	Bulletin Number	Product
0x02	2	1336S	1336 PLUS Fractional HP
0x03	3	1336S	1336 PLUS
0x07	7	1336F	1336 PLUS II
0x10	16	1336T	1336 FORCE w/PLC Adapter Board
0x11	17	2364F	2364 RGU DC Bus Regen Front End
0x12	18	1394	1394 Motion Drive
0x13	19	1557	1557 Medium Voltage AC Drive
0x14	20	193	SMP-3
0x15	21	150	SMC Dialog Plus
0x17	23	1305	1305 AC Drive
0x18	24	1397	1397 DC Drive
0x19	25	1336R	1336 Line Regeneration Package
0x20	32	1336T	1336 FORCE w/Standard Adapter Board
0x22	34	1336E	1336 IMPACT

Message Operation

Product ID Number Read, through the drive response message word 3, indicates the type of device the DeviceNet adapter is connected to. This value is defined in the message response chart shown above.

If an error has occurred, word 1 of the response returns a negative value of –32512.

Example

In this example, the **Product ID Number Read** was requested. The drive response contained a value of 3 in word 3 of its message response, indicating a connection to a 1336 PLUS drive.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	3	256	0							
Drive Response	6	256	0	3*						

* Example only — These values vary depending on parameters and products.

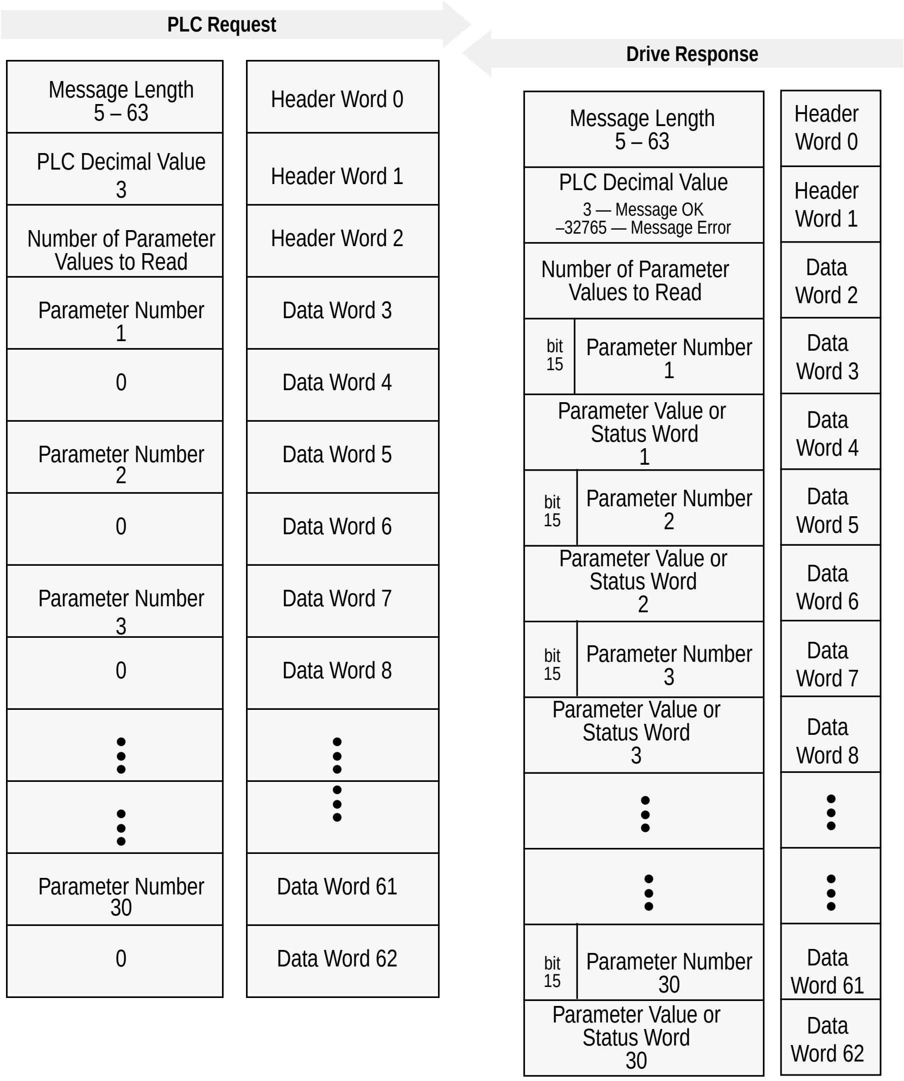
Scattered Parameter Value Read

Scattered Parameter Value Read reads a scattered list of parameters.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 5 – 63 words
Drive response instruction length: 5 – 63 words

Figure F.6
Message Structure



PLC Request

Message Length 5 – 63	Header Word 0
PLC Decimal Value 3	Header Word 1
Number of Parameter Values to Read	Header Word 2
Parameter Number 1	Data Word 3
0	Data Word 4
Parameter Number 2	Data Word 5
0	Data Word 6
Parameter Number 3	Data Word 7
0	Data Word 8
⋮	⋮
⋮	⋮
Parameter Number 30	Data Word 61
0	Data Word 62

Drive Response

Message Length 5 – 63		Header Word 0
PLC Decimal Value 3 — Message OK –32765 — Message Error		Header Word 1
Number of Parameter Values to Read		Data Word 2
bit 15	Parameter Number 1	Data Word 3
Parameter Value or Status Word 1		Data Word 4
bit 15	Parameter Number 2	Data Word 5
Parameter Value or Status Word 2		Data Word 6
bit 15	Parameter Number 3	Data Word 7
Parameter Value or Status Word 3		Data Word 8
⋮		⋮
⋮		⋮
bit 15	Parameter Number 30	Data Word 61
Parameter Value or Status Word 30		Data Word 62

Message Operation

Scattered Parameter Value Read reads a predefined group of parameter values, in any order, from the device. You define the number of parameters to read in word 2 of the request. The parameters to be read and their order is defined starting with word 3. An unused word is left between each parameter request, so the drive can respond with the parameter value, as shown.

If an error has occurred in reading any of the parameters:

- Word 1 of the drive response returns a value of –32765.
- Bit 15 of the drive response word for the number of that parameter is set.
- The drive response word for the value of that parameter returns a status word instead of returning the parameter value.

Example

In this example, eight parameters were read from a 1336 PLUS drive, as defined in word 2 of the request. Parameter numbers 5, 7, 8, 20, 18, 17, 19, and 36 were requested. The drive response returned the values of these parameters in the data file. These values are in drive units.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	19	3	8*	5*	0	7*	0	8*	0	20*
	0	18*	0*	17*	0	19*	0	36*	0	
Drive response	19	3	8*	5*	6*	7*	1000*	8*	1000*	20*
	4096*	18*	4096*	17*	51*	19*	60*	36*	6144*	

* Example only — These values vary depending on parameters and products.

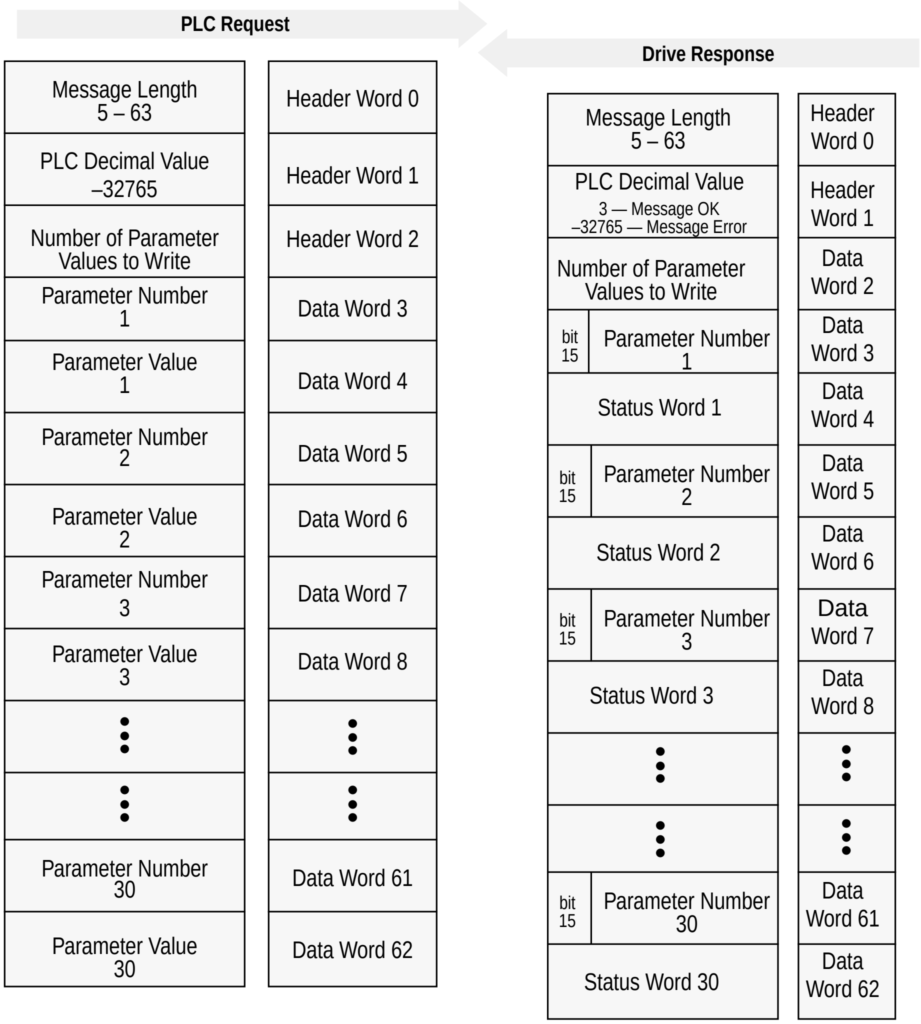
Scattered Parameter Value Write

Scattered Parameter Value Write writes to a scattered list of parameters and returns the status of each parameter. If any of the states have errors, the parameter number is negative.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 5 – 63 words
Drive response instruction length: 5 – 63 words

Figure F.7
Message Structure



PLC Request

Message Length 5 – 63	Header Word 0
PLC Decimal Value –32765	Header Word 1
Number of Parameter Values to Write	Header Word 2
Parameter Number 1	Data Word 3
Parameter Value 1	Data Word 4
Parameter Number 2	Data Word 5
Parameter Value 2	Data Word 6
Parameter Number 3	Data Word 7
Parameter Value 3	Data Word 8
⋮	⋮
⋮	⋮
Parameter Number 30	Data Word 61
Parameter Value 30	Data Word 62

Drive Response

	Message Length 5 – 63	Header Word 0
	PLC Decimal Value 3 — Message OK –32765 — Message Error	Header Word 1
	Number of Parameter Values to Write	Data Word 2
bit 15	Parameter Number 1	Data Word 3
	Status Word 1	Data Word 4
bit 15	Parameter Number 2	Data Word 5
	Status Word 2	Data Word 6
bit 15	Parameter Number 3	Data Word 7
	Status Word 3	Data Word 8
	⋮	⋮
	⋮	⋮
bit 15	Parameter Number 30	Data Word 61
	Status Word 30	Data Word 62

Message Operation

Scattered Parameter Value Write writes data values to a predefined group of device parameters in any order. You define the number of parameters to write in word 2. The parameters to be written to and their order is defined starting with word 3.

If an error occurs while writing to any of the parameters:

- Word 1 of the drive response returns a value of –32765.
- Bit 15 of the drive response word for that parameter's number is set.
- The drive response word for that parameter's status word is non-zero.

If no error has occurred:

- Word 1 of the drive response returns a value of 3.
- Each of the drive response's parameter numbers are the same as in the request.
- Each of the drive response status words returns a value of 0.

Example

In this example, six parameters were written to in a 1336 PLUS drive. Word 2 of the request defines the number of parameter values that are transferred. Beginning with word 3, the message lists each parameter number followed by the value of the parameter. The values are entered in device units.

The drive response returns the status of each parameter write. If the request was successful, a zero is returned. If an error has occurred, the response returns a status word code for the error.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	15	–32765	6*	90*	1*	150*	4*	30*	20*	31*
	10*	10*	2*	12*	5*					
Drive response	15	3	6*	90*	0*	150*	0*	30*	0*	31*
	0*	10*	0*	12*	0*					

* Example only — These values vary depending on parameters and products.

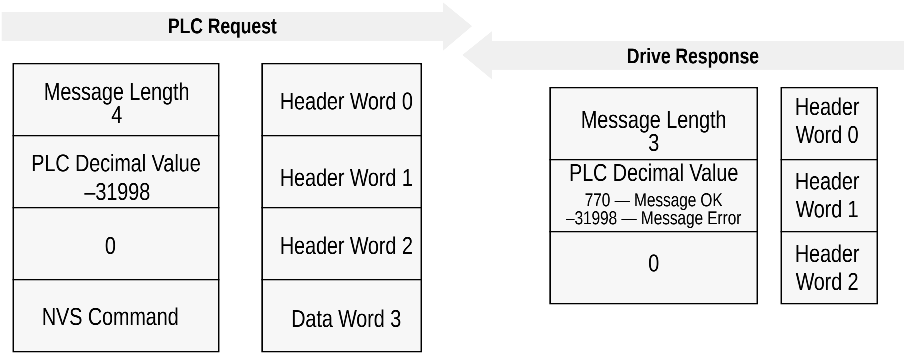
NVS Functions

NVS (Non-Volatile Storage) Functions activates the specified NVS functions.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 4 words
Drive response instruction length: 3 words

Figure F.8
Message Structure



Value	NVS Command
00	Not Used
01	NVS Save
02	NVS Recall
03	NVS Default Initialize

Message Operation

The NVS storage function allows three different message requests:

- NVS Save saves parameter information from the working memory or RAM to NVS Storage.
- NVS Recall retrieves the last saved data from NVS Storage and places it in the working memory or RAM.
- NVS Default Initialize clears the RAM and NVS Storage and sets all parameter values to default.

If an error has occurred, response word 1 returns a value of -31998.

Example

This example requests the NVS Storage Save function be performed.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	4	–31998	0*	0, 1, 2, or 3						
Drive response	3	770	0*							

* Example only — These values vary depending on parameters and products.

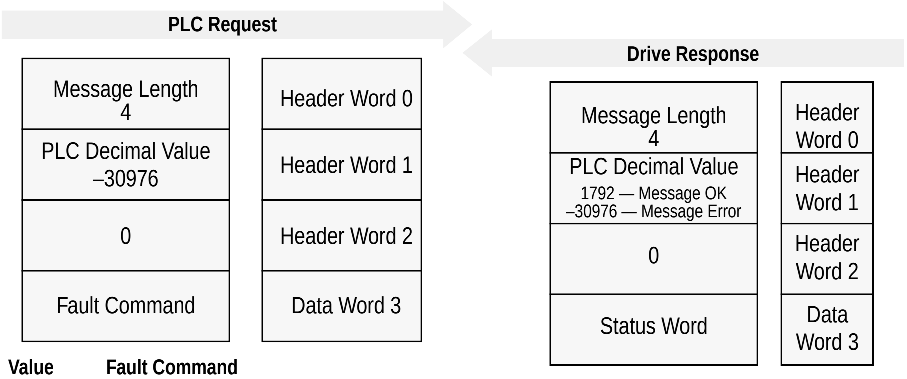
Fault Command Write

Fault Command Write activates the Clear Fault, Clear Fault Queue, and Drive Reset functions.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 4 words
Drive response instruction length: 4 words

Figure F.9
Message Structure



Value	Fault Command
00	Not Used
01	Clear Fault
02	Clear Fault Queue
03	Drive Reset (1336 FORCE Only)

Message Operation

The specified fault Clear/Reset function sends a fault handling request to the device.

- A Clear Fault request clears the last fault that occurred.
- A Clear Fault Queue clears the entire fault buffer. Certain devices may store more than one fault.
- A Drive Reset is used with the 1336 FORCE drive product only. This function resets the drive; it clears the fault queue and writes the parameter information stored in NVS Storage to RAM.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	4	–30976	0	0, 1, 2, or 3						
Drive response	4	1792	0	0*						

* Example only — These values vary depending on parameters and products.

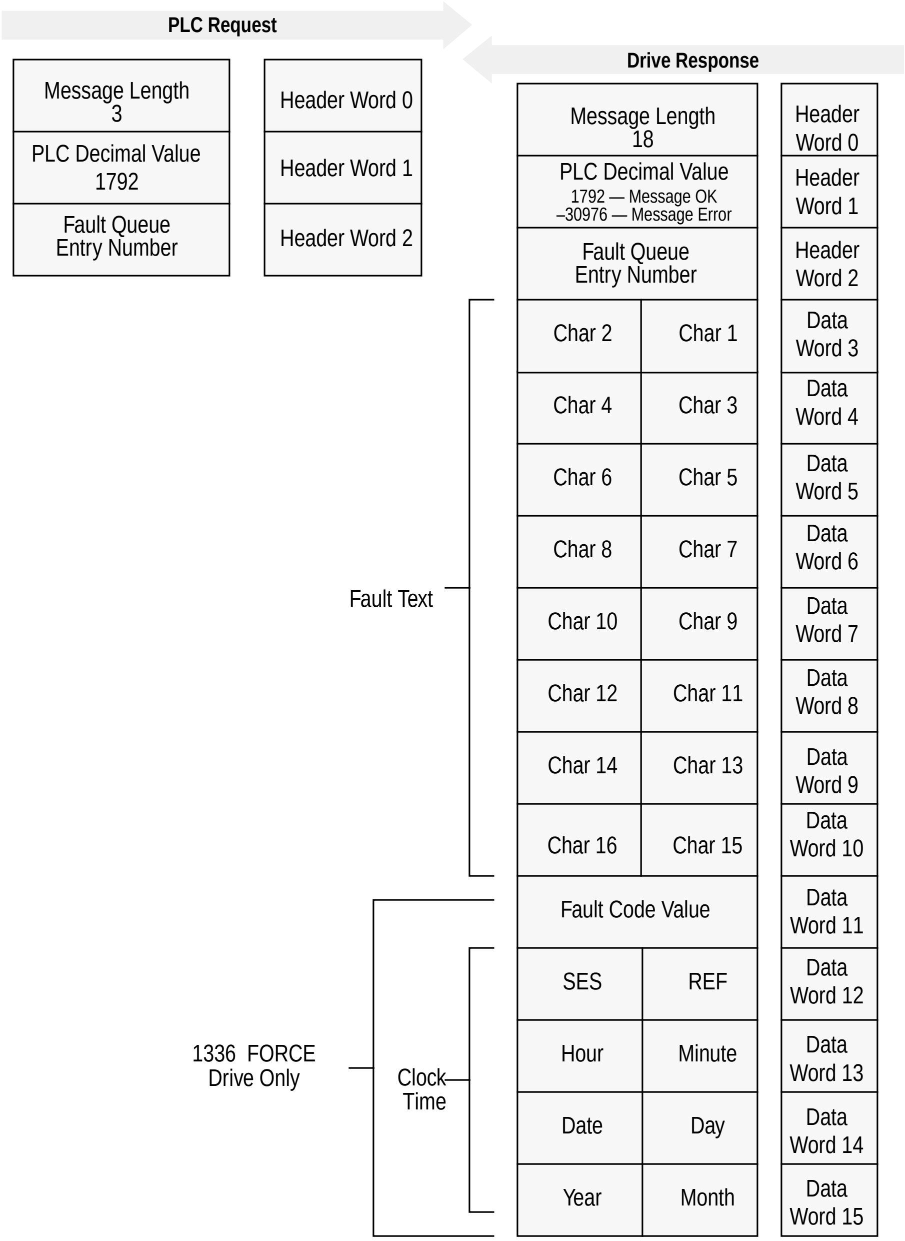
Fault Queue Entry Read Full

Fault Queue Entry Read Full reads the contents of the specified fault queue entry. A message is returned which includes the fault text and fault code associated with the specified fault queue entry. The 1336 FORCE drive also returns the time stamp associated with the fault.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 3 words
Drive response instruction length: 12 or 16 words

Figure F.10
Message Structure



PLC Request

Message Length 3	Header Word 0
PLC Decimal Value 1792	Header Word 1
Fault Queue Entry Number	Header Word 2

Drive Response

Message Length 18		Header Word 0
PLC Decimal Value 1792 — Message OK –30976 — Message Error		Header Word 1
Fault Queue Entry Number		Header Word 2
Char 2	Char 1	Data Word 3
Char 4	Char 3	Data Word 4
Char 6	Char 5	Data Word 5
Char 8	Char 7	Data Word 6
Char 10	Char 9	Data Word 7
Char 12	Char 11	Data Word 8
Char 14	Char 13	Data Word 9
Char 16	Char 15	Data Word 10
Fault Code Value		Data Word 11
SES	REF	Data Word 12
Hour	Minute	Data Word 13
Date	Day	Data Word 14
Year	Month	Data Word 15

Message Operation

Fault Queue Entry Read Full reads the contents of the fault queue specified in word 3 of the request. The response returns the fault text which can be ASCII text. Every two characters of text are in reverse order. Also, the 1336 FORCE drive returns a time stamp, indicating the day and time the fault occurred.

If an error has occurred, word 1 of the response returns a negative value.

Example

In this example, Fault Queue Entry number 3 was retrieved from a 1336 PLUS drive. The drive response returned the ASCII text *Drive Reset Flt*, with each character reversed. The fault code for this example is 22.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	3	1792	3*							
Drive response	18	1792	3*	29252*	20313*	8293*	25938*	25971*	8308*	27718*
	8303*	22*								
Drive response	\00\12	\07\00	\03\00	r D	v i	e	e R	e s	t	l F
	t	\00\16								

* Example only — These values vary depending on parameters and products.

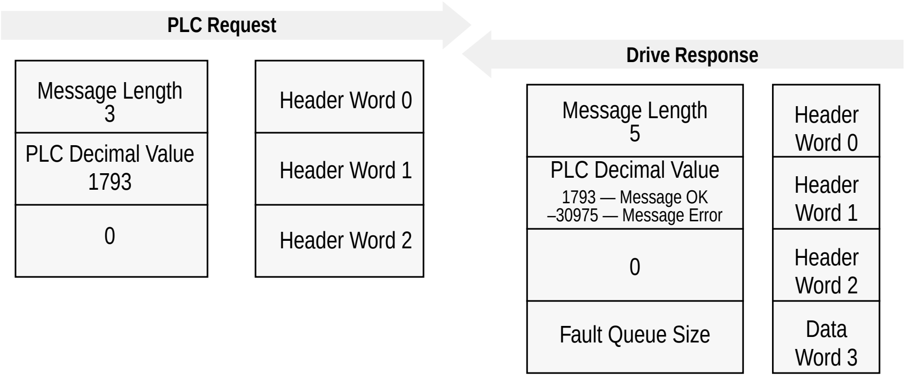
Fault Queue Size

Fault Queue Size gets the number of fault entries allowed in the fault queue.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 3 words
Drive response instruction length: 4 words

Figure F.11
Message Structure



Message Operation

Fault Queue Size reads back the size of the fault queue available in the product. Each product may have a different number of fault queue entries available for storage.

If an error has occurred, word 1 of the response returns a value of -30975.

Example

In this example, a 1336 PLUS drive was used. This product has a fault queue of four storage locations available to store faults. This value is seen in word 3 of the response header message.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	3	1793	0							
Drive response	6	1793	0	4*						

* Example only — These values vary depending on parameters and products.

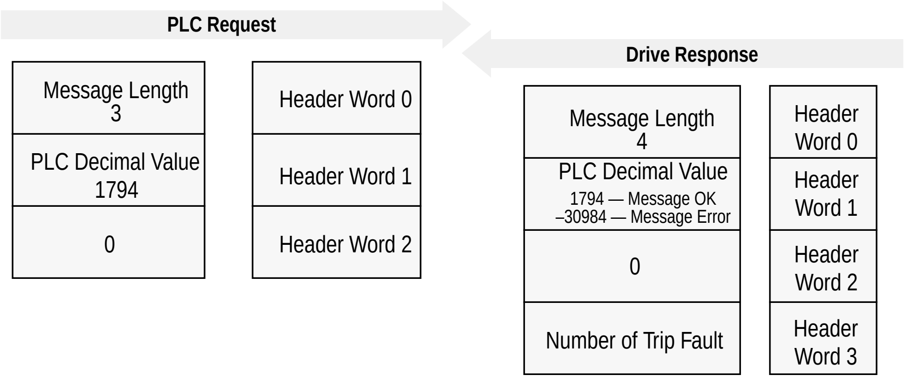
Trip Fault Queue Number

Trip Fault Queue Number provides the fault queue number of the fault that caused the device to trip.

PLC Block Transfer Emulation Instruction Data

PLC request instruction length: 3 words
Drive response instruction length: 4 words

Figure F.12
Message Structure



Message Operation

Trip Fault Queue Number provides the number of the entry in the fault queue that tripped the device in word 3 of the drive response. The fault queue number is 0 when the device is not faulted.

If an error has occurred in the block transfer, word 1 of the response is negative.

Example

In this example, the device has stored a fault in the first entry of the fault queue that caused the drive to trip. Word 3 of the response indicates the entry number.

Data File Format

	0	1	2	3	4	5	6	7	8	9
PLC request	3	1794	0							
Drive response	6	1794	0	1*						

* Example only — These values vary depending on parameters and products.

Notes:

Event Queue Messages

Appendix Objectives

Appendix G provides information on the messages in the adapter's Event Queue. This queue is only available using the DF-1 serial port on the 1203-GU6 module.

Event Queue Messages

The following table lists the messages you might receive.

Event	What It Means
Empty Queue	Nothing to report.
DeviceNet Reset	A reset to the DeviceNet Identity Object was received.
SP Msg Timeout	The SCANport product did not respond to a message request. Verify the cables are connected.
Noise Corruption	This should never occur. If you see this message, it is usually the result of external noise.
SCANport Reset	The SCANport product issued a reset command.
Logon Error	An incorrect response was made to the SCANport logon sequence.
No Pings Found	SCANport pings. (Heartbeat messages were not received for more than 2 seconds. Check cables.)
No I/O Messages	SCANport I/O messages were not received for more than 2 seconds. Possible drive problem.
Datalink Missed	Not all datalinks selected have been received in the last 2 seconds.
I/O Port Changed	The SCANport PIN ID has changed. This message is normal after a power cycle.
SCANport Bus-Off	The SCANport CAN connection turn bus-off. Normally caused by too much noise.
SP Fault Msg	The drive issued a fault broadcast message.
SP PIN ID 7	A SCANport PIN ID of seven indicates that no SCANport product is connected to the adapter.
DN Poll Timeout	A polled I/O connection has timed out.
DN I/O Timeout	An expected packet rate timer expired on a DeviceNet I/O connection.
DN I/O Too Long	A DeviceNet I/O message was longer than the configured length to be received. You may need to reconfigure the I/O length in the scanner.
Manual Reset	*Reset Adapter* was set to cause an adapter reset.
Bad I/O Fragment	A DeviceNet I/O fragment was recived out of sequence. Possible line noise problem.
Idle I/O Message	The DeviceNet scanner was placed in the Program mode.
Peer I/O Timeout	A peer I/O message was not received for the time-out period.
Adapter Reset	The adapter microprocessor was reset. Normal after a power cycle.
Bad EEPROM CRC	The adapter parameter storage EEPROM has a corrupt CRC. To correct this, save a new value to one of the adapter parameters and check that all adapter parameters are at their desired values.
SP Fault Clear	A SCANport product has issued a fault cleared message.
DN COS Timeout	A Change of State (COS) connection has timed out.

Event	What It Means
DN Poll Allocate	A polled connection has been allocated.
DN COS Allocate	A Change of State (COS) I/O connection has been allocated.
DN Poll Closed	A poll I/O connection was explicitly closed.
DN COS Closed	A change of state connection was explicitly closed.

Notes:

Notes

Notes

We Want Our Manuals to be the Best!

You can help! Our manuals must meet the needs of you, the user. This is your opportunity to make sure they do just that. By filling out this form you can help us provide the most useful, thorough, and accurate manuals available. Please take a few minutes to tell us what you think - then mail this form or FAX it.
FAX: your local Allen-Bradley Sales Office or 414/512-8579

PUBLICATION NAME ____________________

PUBLICATION NUMBER, DATE AND PART NUMBER (IF PRESENT) ____________________

✔ **CHECK THE FUNCTION THAT MOST CLEARLY DESCRIBES YOUR JOB.**

□ SUGGEST/RESPONSIBLE FOR THE PURCHASE OF EQUIPMENT
□ DESIGN/IMPLEMENT ELECTRICAL SYSTEMS
□ SUPERVISE FLOOR OPERATIONS
□ MAINTAIN/OPERATE PROGRAMMABLE MACHINERY
□ TRAIN/EDUCATE MACHINE USERS

✔ **WHAT LEVEL OF EXPERIENCE DO YOU HAVE WITH EACH OF THE FOLLOWING PRODUCTS?**

	NONE	LITTLE	MODERATE	EXTENSIVE
PROGRAMMABLE CONTROL	□	□	□	□
AC/DC DRIVES	□	□	□	□
PERSONAL COMPUTERS	□	□	□	□
NC/CNC CONTROLS	□	□	□	□
DATA COMMUNICATIONS/LAN	□	□	□	□

✔ **RATE THE OVERALL QUALITY OF THIS MANUAL BY CIRCLING YOUR RESPONSE BELOW. (1) = POOR (5) = EXCELLENT**

HELPFULNESS OF INDEX/TABLE OF CONTENTS	1	2	3	4	5
CLARITY	1	2	3	4	5
EASE OF USE	1	2	3	4	5
ACCURACY AND COMPLETENESS	1	2	3	4	5
QUALITY COMPARED TO OTHER COMPANIES' MANUALS	1	2	3	4	5
QUALITY COMPARED TO OTHER ALLEN-BRADLEY MANUALS	1	2	3	4	5

✔ **WHAT DID YOU LIKE MOST ABOUT THIS MANUAL?**

✔ **WHAT DID YOU LIKE LEAST ABOUT THIS MANUAL?**

✔ **PLEASE LIST ANY ERRORS YOU FOUND IN THIS MANUAL (REFERENCE PAGE, TABLE, OR FIGURE NUMBERS).**

✔ **DO YOU HAVE ANY ADDITIONAL COMMENTS?**

✔ **COMPLETE THE FOLLOWING.**

NAME ____________________ COMPANY____________________

TITLE ____________________ DEPARTMENT____________________

STREET ____________________ CITY ____________ STATE ______ ZIP ____________

TELEPHONE ____________________ DATE ____________________

CUT ALONG DOTTED LINE ✂

FOLD HERE

FOLD HERE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 413 MEQUON, WI

POSTAGE WILL BE PAID BY ADDRESSEE

ALLEN-BRADLEY
Attn: Marketing Communications
P.O. Box 760
Mequon, WI 53092-9907

Reach us now at www.rockwellautomation.com

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

Americas Headquarters, 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444
European Headquarters SA/NV, avenue Herrmann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40
Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1203-5.12 – May 2000
P/N191009 (02)
Supersedes Publication 1203-5.12 – August 1998