Allen-Bradley

1747-SDN DeviceNet Scanner Module

Catalog Number
1747-SDN, Series C

User Manual

Rockwell Automation

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://literature.rockwellautomation.com) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.
ATTENTION	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
SHOCK HAZARD	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
BURN HAZARD	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

The information below summarizes the changes to this manual since the last publication.

To help you find new and updated information in this release of the manual, we have included change bars as shown to the right of this paragraph.

This manual contains this updated information.

Topic	Page
The length of the M0 and M1 files	64
Numeric code 65 for normal operation when AutoScan is enabled was added	91
DeviceNet explicit messaging chapter was added	Chapter 8
AutoScan function chapter was added	Chapter 9
Information about programming the module by using the SLC M0 and M1 files was added	123…127
Information on the 1747-SDN module's firmware history was added	Appendix C
Information about data organization was added	Appendix D
Information about explicit message program control was added	Appendix E

Appendix C

1747-SDN Module Firmware History

Appendix D

Data Organization

Appendix E

Explicit Message Program Control

Glossary

Index

Introduction

This user manual is designed to provide you enough information to get a small example application up and running. Use this manual if you are knowledgeable about DeviceNet and SLC 500 products, but may not have used the products in conjunction. The information provided is a base; modify or expand the examples to suit your particular needs.

The manual contains instructions on configuring a DeviceNet network by using RSLinx and RSNetWorx for DeviceNet software. It also describes how to use the SLC 500 pass-through feature to communicate with the DeviceNet network for adjustment and tuning of network devices via an Ethernet and Data Highway Plus (DH+) network.

The example application demonstrates how to perform control on a DeviceNet network by using an SLC 500 processor and the 1747-SDN module. You use RSLogix 500 programming software to create a ladder logic program to control a photoeye and a RediSTATION operator interface.

IMPORTANT	This user manual should be used in conjunction with the 1747-SDN DeviceNet Scanner Module Installation Instructions, publication 1747-IN058. The installation instructions contain important information on configuring your scanner.

Audience

This manual is intended for control engineers and technicians who are installing, programming, and maintaining a control system that includes an SLC 500 processor communicating on a DeviceNet network through a 1747-SDN module.

We assume that you:

- are developing a DeviceNet network by using a SLC 500 processor in conjunction with a 1747–SDN module.
- know each of your device’s I/O parameters and requirements.
- understand SLC processor programming and operation.
- are experienced with the Microsoft Windows environment.
- are familiar with RSNetWorx for DeviceNet software.

The Example Application

This manual describes how to set up an example application. The manual provides examples of each step of the setup, with references to other manuals for more details.

System Components

We used the following devices and software for the example application. For your own application, substitute your own devices to fit your needs. The recommended configurations in this user manual will help you set up the test system and get it working. Your eventual configuration will depend on your application.

TIP If you use different software or fimware versions of these products, some of your dialogs may appear slightly different from those shown in the example.

Quantity	Product Name	Catalog Number	Series/Revision
Hardware			
1	SLC 500 modular chassis	1746-A4, 1746-A7, 1746-A10, 1746-A13	B
1	SLC 500 power supply	1746-P1, 1746-P2, 1746-P3, 1746-P4, 1746-P5, 1746-P6	-
1	SLC 5/04 processor	1747-L541, 1747-L542, 1747-L543	-
1	SLC 5/05 processor (Ethernet network)	1747-L551, 1747-L552, 1747-L553	-
1	DeviceNet scanner module	1747-SDN/B	-
1	ControlNet RS-232 interface module	1747-KFC15	B
1	DeviceNet quad-tap	1492-DN3TW	-
1	RediSTATION operator interface module	2705-T*x*DN1*x*42*x-xxxx*	-
1	Series 9000 photoeye	42GNU-9000 or equivalent	-
1	DeviceNet RS-232 interface module	1770-KFD	-
1	RS-232 cables	1787-RSCABL/A (personal computer to 1770-KFD)	-
	DeviceNet dropline or trunk cables, as needed	1787-PCABL, 1787-TCABL, 1787-MCABL	-

Quantity	Product Name	Catalog Number	Series/Revision
1	24V power supply	Any regulated 24V dc, 8 A	-
1	Personal computer	IBM-compatible Pentium+ Windows 2000 or later	-
Software			
	RSLogix 500	9324-RL0300*xxx*	Rev 4.00
	RSNetWorx for DeviceNet	9357-DNETL3	Rev 2.22
	RSLinx	9355-WAB	Rev 2.10

Common Techniques Used in This Manual

The following conventions are used throughout this manual:

- Bulleted lists provide information, not procedural steps.
- Numbered lists provide sequential steps.

TIP This symbol identifies helpful tips.

Additional Resources

These documents contain additional information concerning related Rockwell Automation products.

Resource	Description
1747-SDN DeviceNet Scanner Module Installation Instructions, publication 1747-IN058	Provides information on installing and connecting the module.
ControlFlash Firmware Upgrade Kit Quick Start, publication 1756-QS105	Provides instructions on using the ControlFlash utility to upgrade the firmware.
Getting Results with RSLogix 500, publication LG500-GR002	Provides information on RSLogix 500 software.
Getting Results with RSLinx Classic, publication LINX-GR001	Provides information on RSLinx software.
DeviceNet Media Design and Installation Guide, publication DNET-UM072	Provides information on using DeviceNet communication network.
Getting Results with RSNetWorx for DeviceNet, publication DNET-GR001	Provides information on using RSNetWorx for DeviceNet software.
DeviceNet RS-232 Interface Module, publication 1770-5.6	Provides information on connecting and installing the DeviceNet RS-232 Interface module.
SLC 500 ControlNet RS-232 Interface User Manual, publication 1747-5.34	Provides information about the 1747-KFC15 module.

Resource	Description
ControlNet Coax Media Planning and Installation Guide, publication CNET-IN002	Provides information on planning and installing ControlNet coax media systems.
RediSTATION operator interface User Manual, publication 2705-UM001	Provides information on installing and using the RediSTATION operator interface.
SLC 500 Module Hardware Style User Manual, publication 1747-UM011	Provides information on installing, wiring, startup, and maintenance of SLC modular hardware.
Quick Start for experienced Users, publication 1747-10.4	Provides information on features, setup, configuration, and communication for the SLC 500 Ethernet processors.

You can view or download publications at http://literature.rockwellautomation.com. To order paper copies of technical documentation, contact your local Rockwell Automation distributor or sales representative.

Before You Begin

What This Chapter Contains

This chapter provides an overview of communication between the SLC 500 processor and DeviceNet devices via the 1747-SDN module. The configuration data tables and the RSNetWorx for DeviceNet software dialogs and dialogs used to configure the data tables are also described.

The following table identifies what this chapter contains and where to find specific information.

What You Need to Know

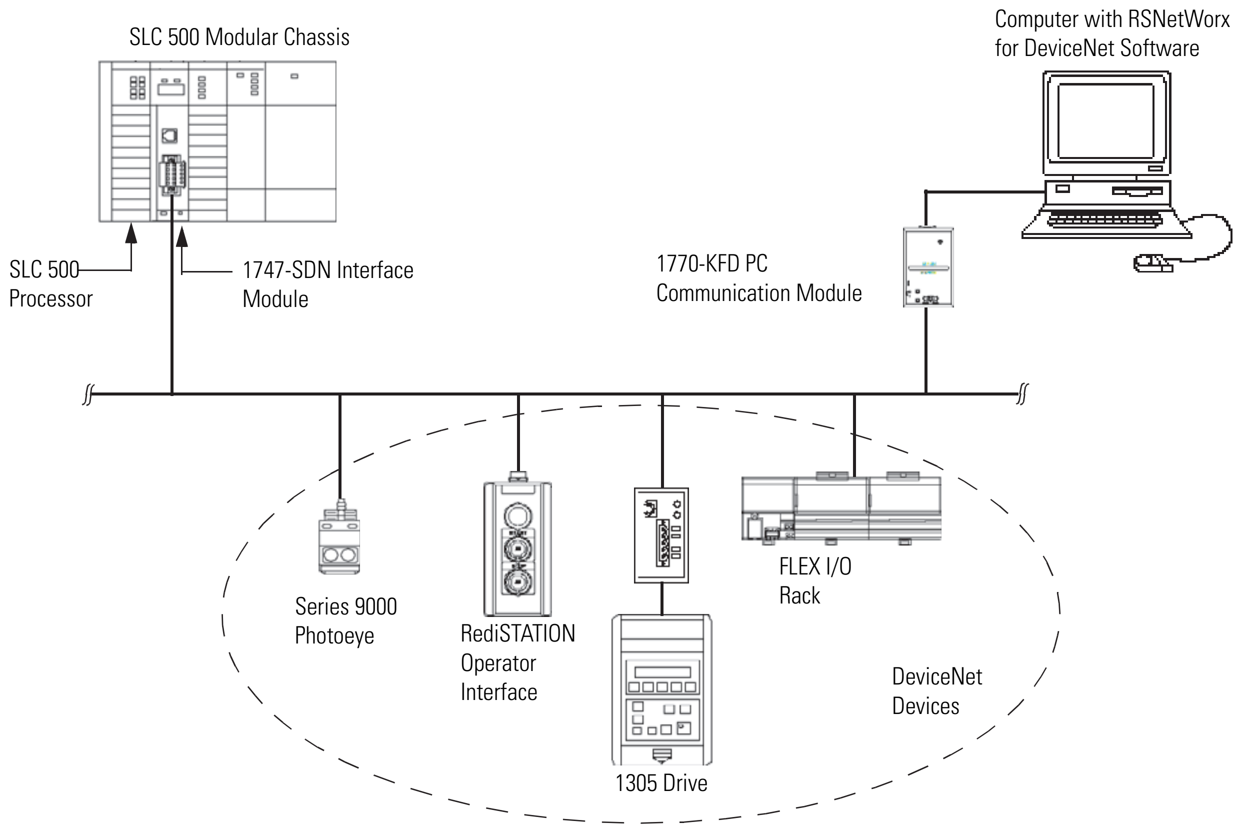
Before configuring your 1747-SDN module, you must understand:

- the data exchange between the an SLC 500 processor and DeviceNet devices through the 1747-SDN module.
- user-configurable 1747-SDN module data tables.
- the role of RSNetWorx for DeviceNet software.

What Your 1747-SDN Module Does

In a typical configuration, the 1747-SDN module acts as an interface between DeviceNet devices and the SLC 500 processor.

Typical DeviceNet Network



The 1747-SDN module communicates with DeviceNet devices over the network to:

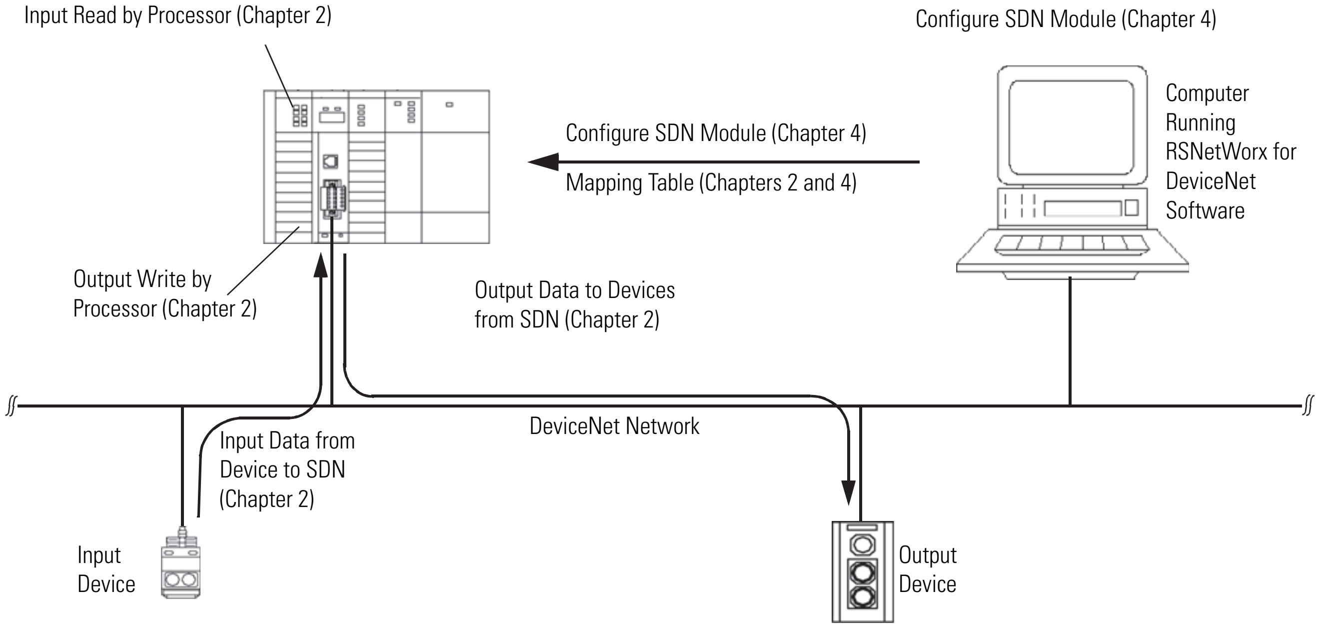
- read inputs from a device.
- write outputs to a device.
- download configuration data.
- monitor a device's operational status.

The 1747-SDN module communicates with the processor in the form of M1/M0 File Transfers and/or Discrete I/O. Information exchanged includes the following:

- Device I/O data
- Status information
- Configuration data

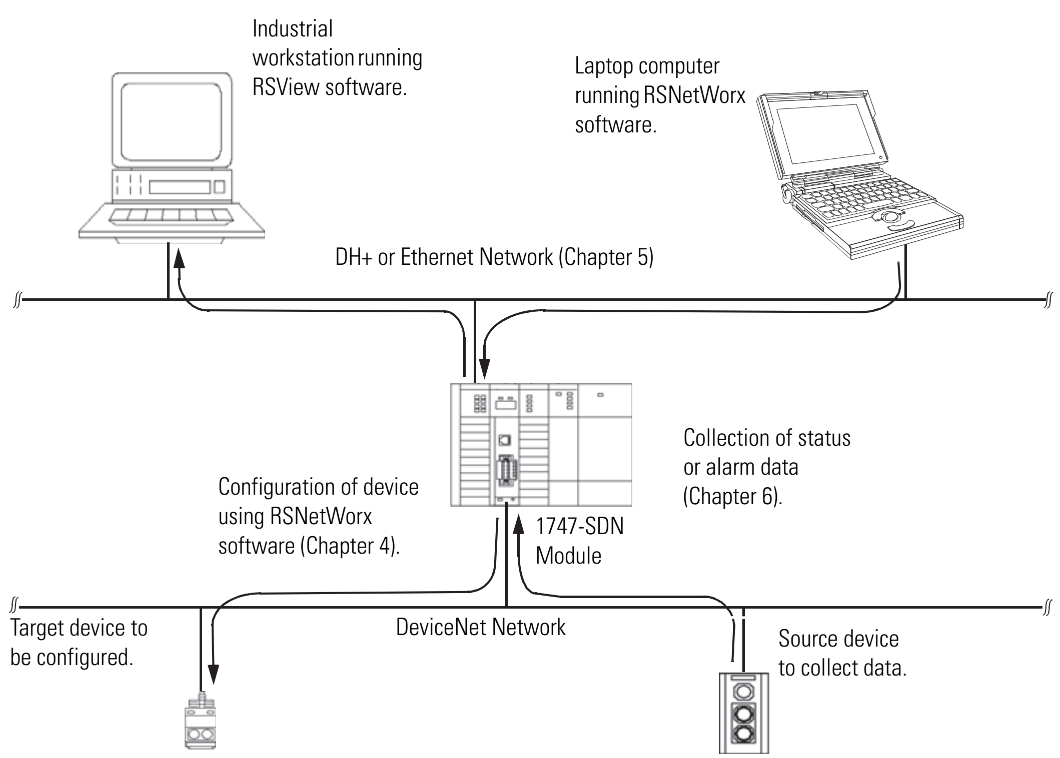
A processor to I/O DeviceNet configuration is shown in the following figure. See the referenced chapters for more information.

Processor to I/O



The 1747-SDN interface module can also be used to bridge a DeviceNet network with another network.

Configuring Devices and Data Collection on Higher-level Networks via SLC 500/SDN Module



Communicating with Your Devices

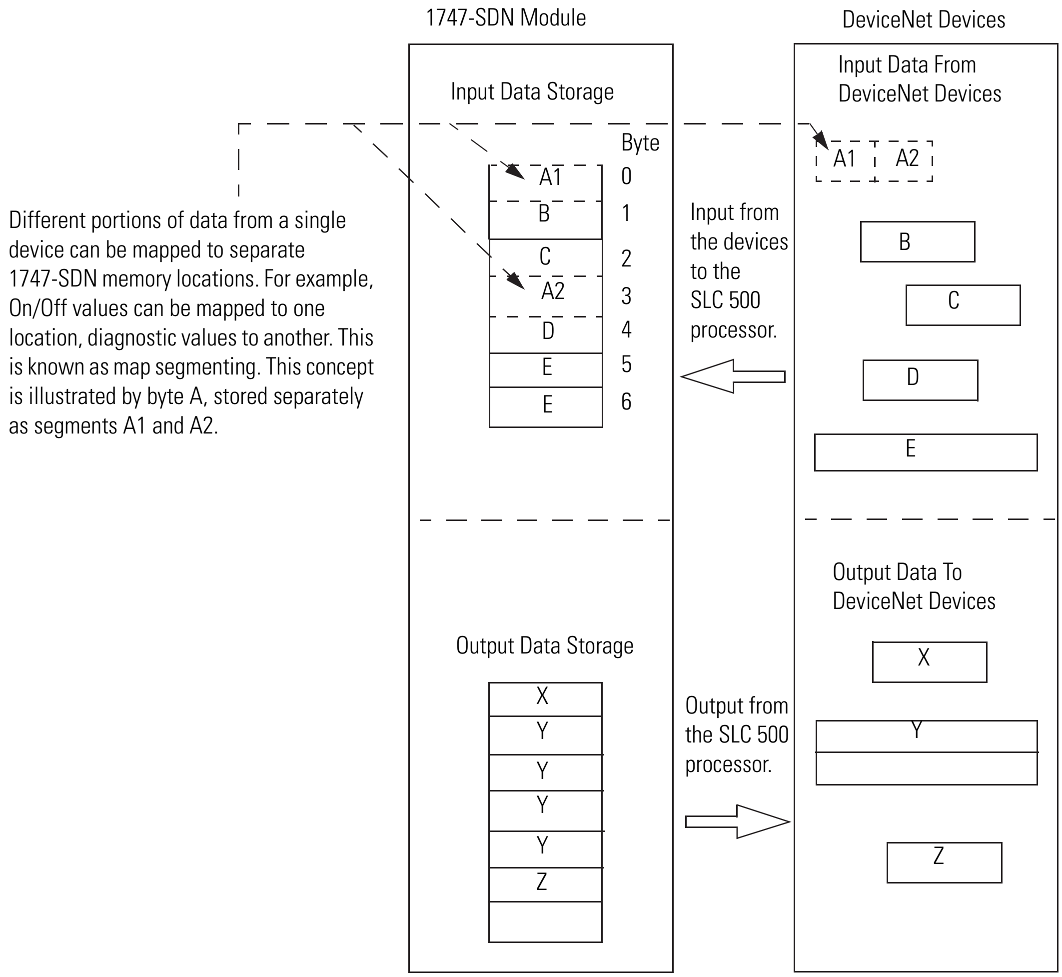
The 1747-SDN module communicates with a device via strobe, poll, change of state, and/or cyclic messages. It uses these messages to solicit data from or deliver data to each device. Data received from the devices, or input data, is organized by the 1747-SDN module and made available to the processor. Data received from your SLC 500 processor, or output data, is organized in the 1747-SDN module and sent on to your devices.

IMPORTANT

Throughout this document, input and output are defined from the SLC 500 processor's point of view. Output is data sent from the SLC 500 processor to a device. Input is data collected by the SLC 500 processor from a device.

All data sent and received on a DeviceNet network is in byte lengths. A device may, for example, produce only two bits of input information. Nevertheless, since the minimum data size on a DeviceNet network is one byte, two bits of information are included in the byte of data produced by the device. In this example (only two bits of input information), the upper six bits are insignificant.

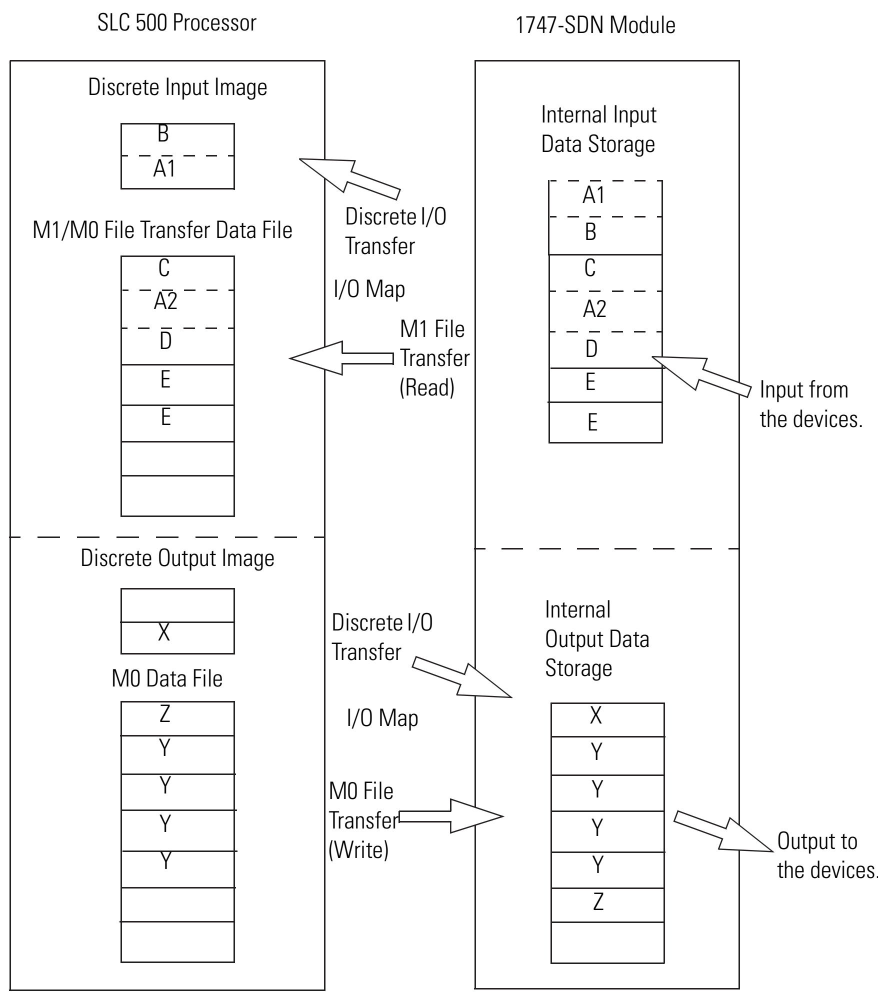
Communicating With Other Devices



Communicating with Your SLC 500 Processor

The 1747-SDN module does not send data to your processor. Data transferred between the module and the processor must be initiated by the processor. Output data is sent, or written, to the scanner by your processor by placing the data in the M0 file. This data is organized in the scanner, which in turn passes the data on to the scanned devices via strobe, poll, change of state, or cyclic messages.

Data Flow



1747-SDN Module Data Tables

To manage the flow of data between your SLC 500 processor and the network devices, the 1747-SDN module uses the following data tables:

- Scanner configuration table (SCT)
- Scanlist table (SLT)
- Device input data table
- Device output data table
- Device active table
- Device failure table
- Client/Server transaction tables

You can configure the first two of these data tables through RSNetWorx for DeviceNet software.

- Scanner configuration table (SCT)
- Scanlist table (SLT)

These two tables are stored in the 1747-SDN module's nonvolatile memory and used to construct all other data tables.

Scanner Configuration Table (SCT)

The SCT controls basic information your 1747-SDN module needs to function on your DeviceNet network. It tells your 1747-SDN module:

- if it can transmit and receive input and output data.
- how long it waits after each scan before it scans the devices again.
- when to send out its poll messages.

Scanlist Table (SLT)

The SLT supports I/O updating for each of your devices on the network. It also makes it possible for your 1747-SDN module to make device data available to your SLC processor. The SLT tells your 1747-SDN module:

- which device node addresses to scan.
- how to scan each device (strobe, poll, change of state, cyclic, or any valid combination).
- how often to scan your devices.
- exactly where in each device's total data to find the desired data.

- the size of the input data/output data.
- exactly where to map the input or output data for your processor to read or write.
- how your processor reads each device's input data (M1/M0 file or discrete I/O).

Data Table Information

User-configured Tables	Data in This Table	RSNetWorx Software Configuration Dialog
SCT	• Basic operation parameters • I/O communication data (enable/disable) • Interscan delay • Background poll ratio	1747-SDN module configuration
SLT	• Device-specific identification data	Scanlist editor (SLE)
	• Data transfer method • Transmit/receive data size	Edit device I/O parameters
	• Input and output data source and destination locations	These values can be configured automatically through the AutoMap function or manually through the Data Table Map

RSNetWorx Software as a Configuration Tool

RSNetWorx for DeviceNet software is used to configure the 1747-SDN module's data tables. This software tool connects to the 1747-SDN module over the DeviceNet network via a computer RS-232 interface (1770-KFD module) or PC Card (1784-PCD, 1784-PCID, or 1784-PCIDS).

TIP RSNetWorx for DeviceNet software can also communicate with the 1747-SDN module via an Ethernet or Data Highway Plus network.

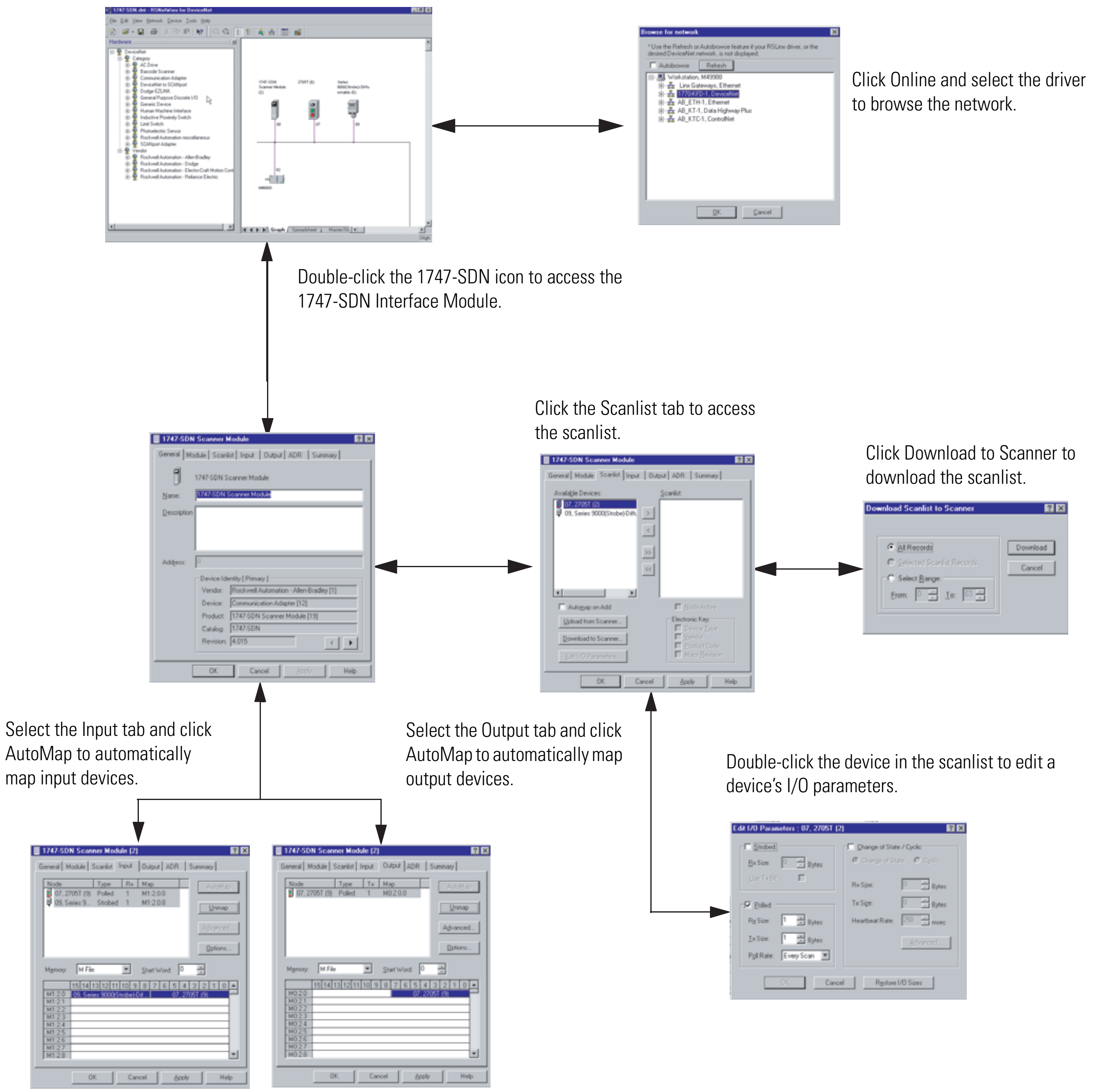
See Chapter 5.

The configuration dialog map below shows the RSNetWorx for DeviceNet dialogs used to configure the 1747-SDN module and the navigation paths between them.

The use of these dialogs is described in Chapter 4.

RSNetWorx for DeviceNet Configuration Dialog Map

The main RSNetWorx for DeviceNet dialog.



What's Next?

The remaining sections of this manual provide the following information:

- Chapter 2 covers the configuration process planning stage through a data mapping example.
- Chapter 3 describes the hardware setup for the example application.
- Chapter 4 covers configuration of the DeviceNet network by using RSNetWorx for DeviceNet software.
- Chapter 5 describes how to configure a DeviceNet network from another network.
- Chapter 6 describes how to create, download, and run the example application program.
- Chapter 7 covers the diagnostics provided for troubleshooting the 1747-SDN module.
- Chapter 8 covers DeviceNet explicit messaging.
- Chapter 9 covers the AutoScan feature.

Planning Your Configuration and Data Mapping Your Devices

What This Chapter Contains

This chapter introduces questions you should ask before configuring your 1747-SDN communication module. In addition, it presents an example DeviceNet network and I/O data mapping scheme for a photoeye and a RediSTATION operator interface. The following table identifies what this chapter covers and where to find specific information.

What You Need to Know

To map data via your 1747-SDN communication module, you must understand the following:

- Network requirements
- Input data mapping
- Output data mapping

Beginning the Process

Planning before configuring your 1747-SDN module helps you do these things:

- Use your memory and bandwidth efficiently
- Cater to device-specific needs and requirements
- Give priority to critical I/O transfers
- Leave room for expansion

You need to know what is on your network. You should be familiar with each device's:

- communication requirements.
- I/O importance and size.
- frequency of message delivery.

At this point in your planning, it is advantageous for you to have some idea of how the network could be expanded. I/O data mapping can be performed automatically by the RSNetWorx software. But when mapping your I/O, you also have the opportunity to allot room for future I/O. This can save time and effort in the future.

For example, RSNetWorx software automatically maps the devices as efficiently as possible, but the result is that multiple devices may share the same word location in memory. However, you can also have the system map the devices such that no two devices share the same memory location by selecting the Dword align option when performing automapping. You can also manually map the devices if you need to assign or reassign them to specific memory locations.

For details, refer to the Help dialogs provided by the RSNetWorx for DeviceNet software. Additional support can be found at the Rockwell Software website: http://www.software.rockwell.com.

The Example Network

The following example illustrates a data mapping plan for a DeviceNet network. Note that even if the mapping is performed automatically by the RSNetWorx software, you must know where the devices are mapped in order to use them in your network.

Example Network Devices

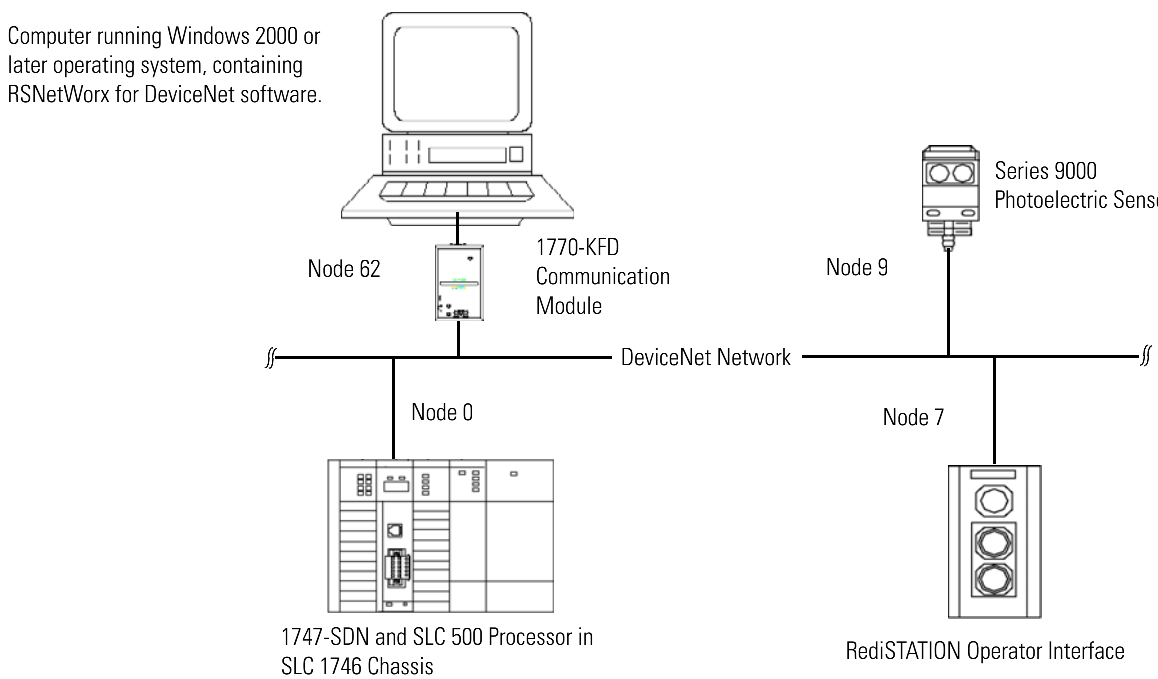
This example network has the following devices:

- A computer running RSNetWorx for DeviceNet software
- A 1747-SDN communication module interfacing an SLC 500 processor with the DeviceNet network
- A Series 9000 photoelectric sensor (strobed)
- A RediSTATION operator interface (polled)

IMPORTANT	In the following example, output is data sent to a device from a controller. Input is data collected from a device by a controller.

The system you set up is shown below.

Example Network



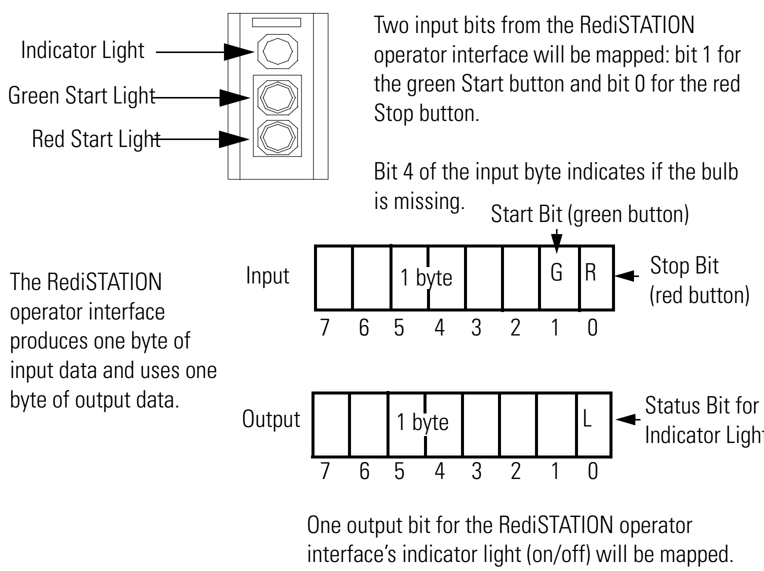
IMPORTANT	Each end of the DeviceNet trunk cable must be properly terminated with a resistor. Refer to the DeviceNet Media Design Installation Guide, publication DNET-UM072, for detailed information.

RediSTATION Operator Interface Input and Output Data Mapping

The RediSTATION operator interface has both inputs and outputs that must be mapped. The input byte is mapped to the 1747-SDN module's M1 file and then to the SLC 500 processor's input data file. The output byte is mapped to the 1747-SDN module's M0 file and then to the SLC 500 processor's output data file.

The mapping procedure, using RSNetWorx for DeviceNet software, is described on page 45.

RediSTATION Operator Interface



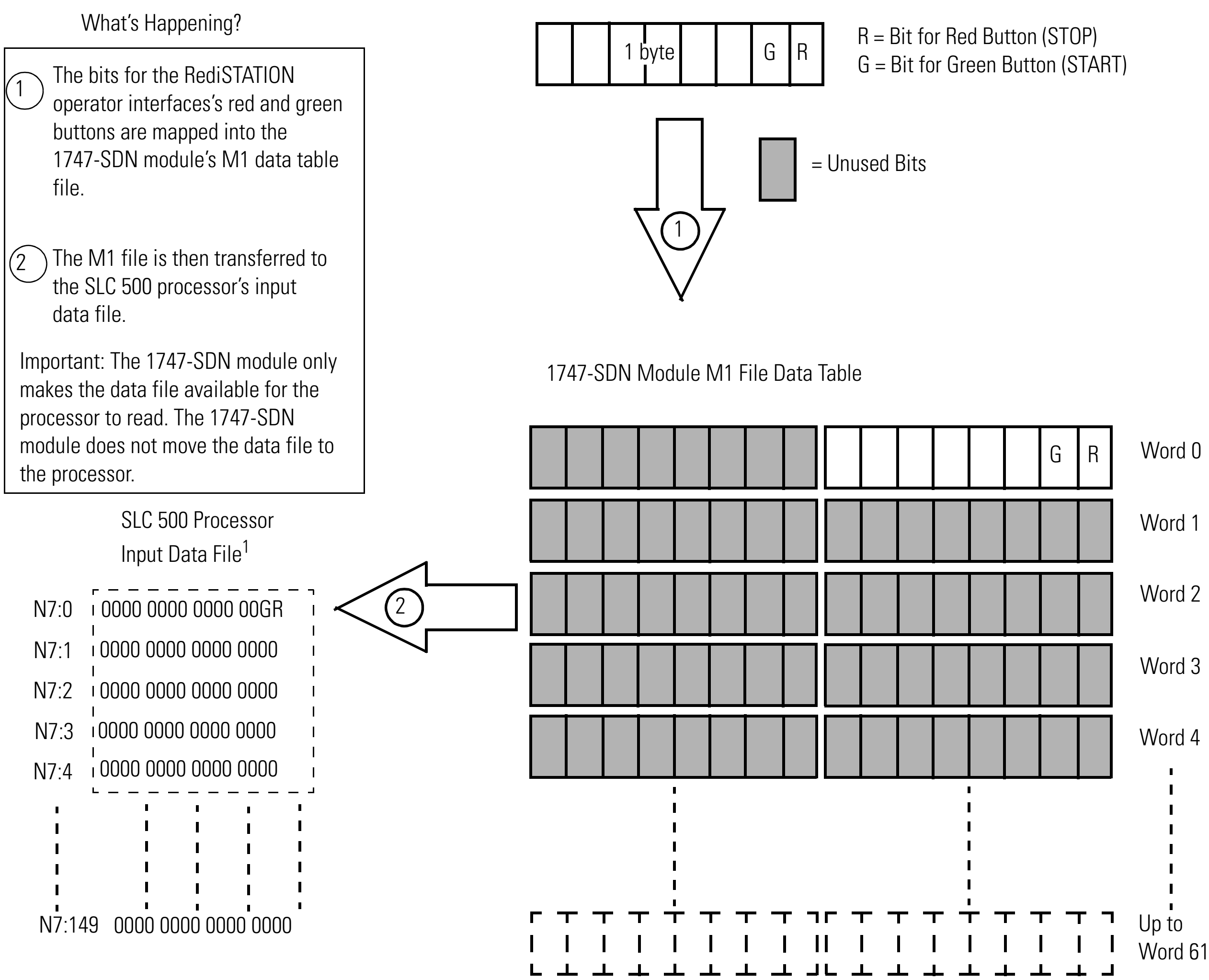
In the RediSTATION operator interface's bits for the red and green buttons and the indicator light status bit:

- 1 = ON.
- 0 = OFF.

Mapping RediSTATION Input Data for an M1 File Data Table Read

The following is an example of input data mapping for the RediSTATION operator interface.

RediSTATION Input Byte



[1]This mapping is based upon the example in chapters 4 and 6. The mapping for your system may be different.

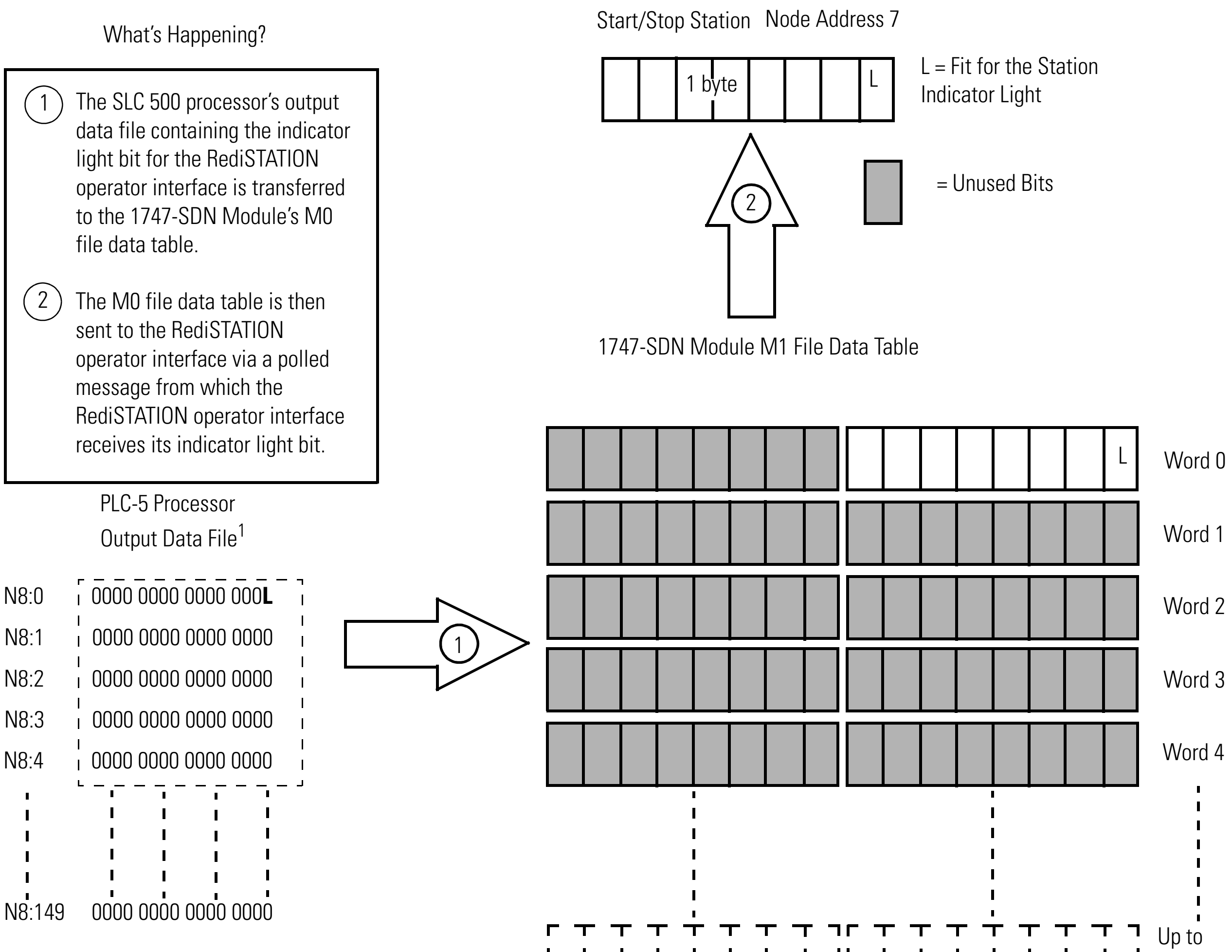
Example: The green START button from the RediSTATION operator interface appears in the SLC 500 processor's input file at address N7:0/1.

The red STOP button from the RediSTATION operator interface appears in the SLC 500 processor's input file at address N7:0/0.

Mapping RediSTATION Output Data for an M0 File Data Table Write

The RediSTATION operator interface's output is mapped to the 1747-SDN module's M0 file. Within the output byte is a bit for the indicator light. The output data file is then transferred from the SLC 500 processor application to turn the light on or off.

RediSTATION Output Byte



[1] This mapping is based upon the example in chapter 4. The actual mapping for your system may be different.

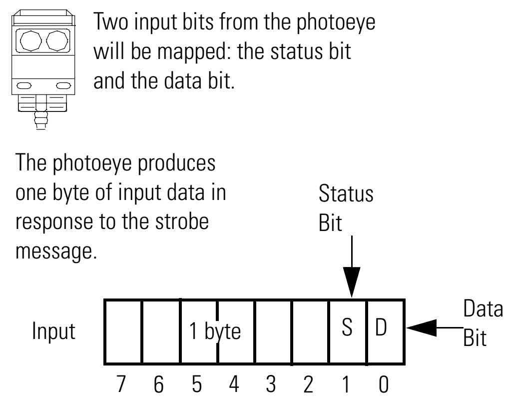
Example: The RediSTATION operator interface's indicator light (L) is taken from N8:1/0 in the SLC 500 processor's output data file.

Photoeye Input Data Mapping

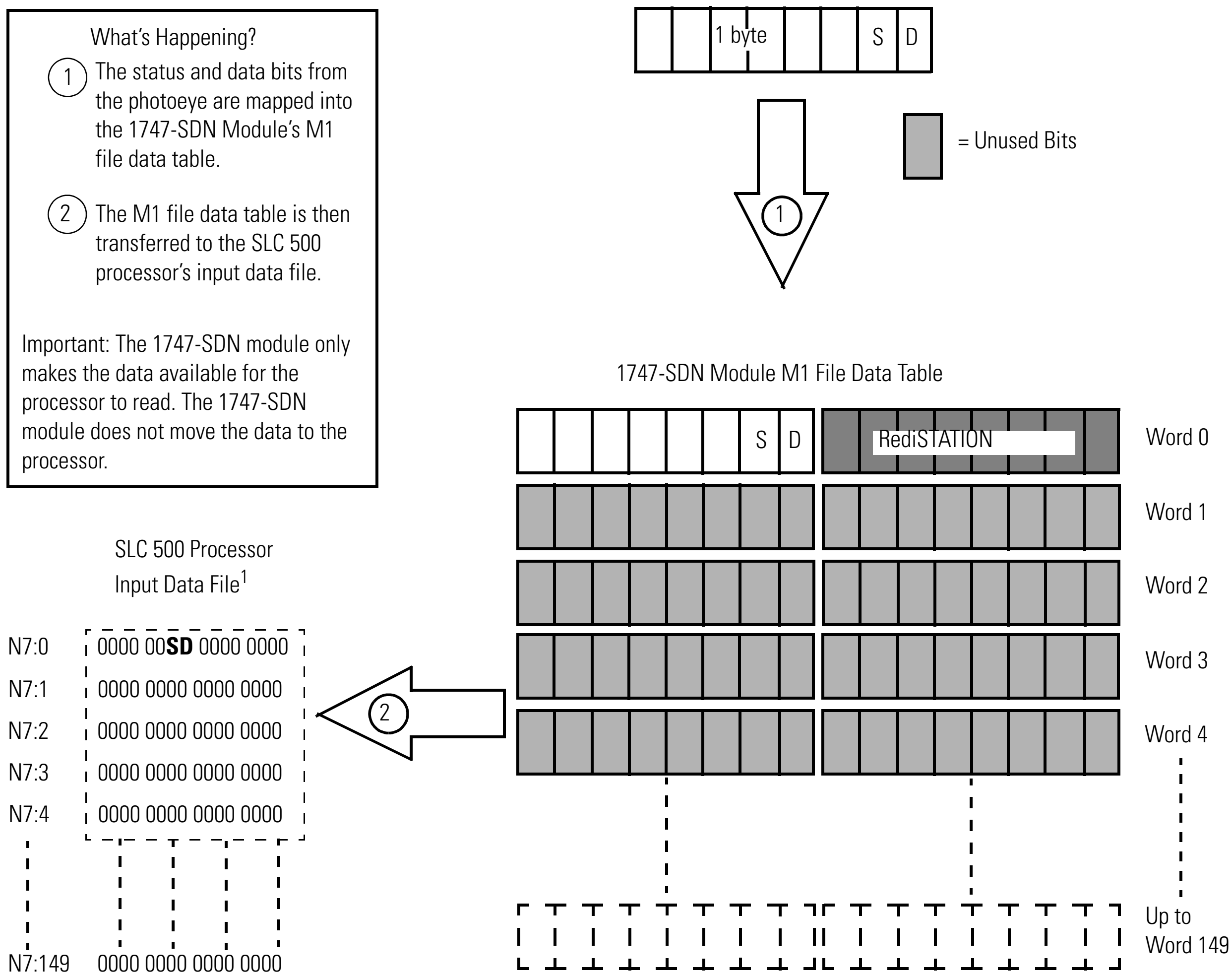
The photoelectric sensor (photoeye) inputs are mapped to the 1747-SDN module's M1 file and then to the SLC 500 processor's input data file. The procedure for doing this by using RSNetWorx for DeviceNet software is described in chapter 4.

The photoeye has no outputs to map.

Series 9000 Photoeye



Mapping Photoeye Input Data for an M1 File Data Table Read



[1] This mapping is based upon the examples in chapters 4 and 6. The actual mapping for your system may be different.

Example: The Status bit from the photoeye appears in the SLC 500 processor's input data file at address N7:0/9.

The Data bit from the photoeye appears in the SLC 500 processor's input data file at address N7:0/8.

What's Next?

Chapter 3 describes how to set up the system hardware for the example application.

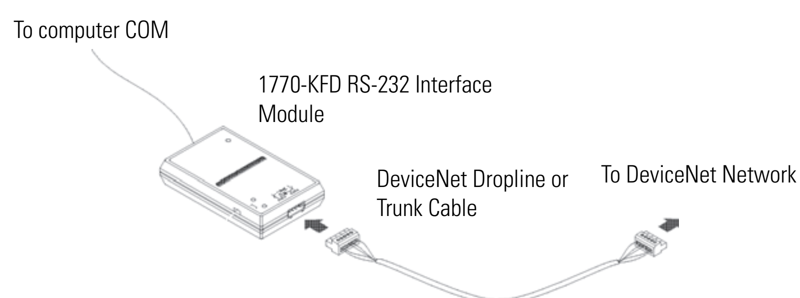
Hardware Setup

What This Chapter Contains

This chapter describes how to set up the hardware for the example application. The following table describes what this chapter contains and where to find specific information.

Installing the 1770-KFD Module

Connect the RS-232 connector on the 1770-KFD communication interface module to one of the serial ports on your computer workstation (COM1). Connect the DeviceNet connector on the 1770-KFD module to a DeviceNet drop or trunk cable. You can make this connection in several ways; for example, by using a DeviceNet Quad Tap (catalog number 1492-DN3TW) as shown on page 45.



For detailed directions on how to install the 1770-KFD module, see the DeviceNet RS-232 Interface Module Installation Instructions, publication 1770-5.6.

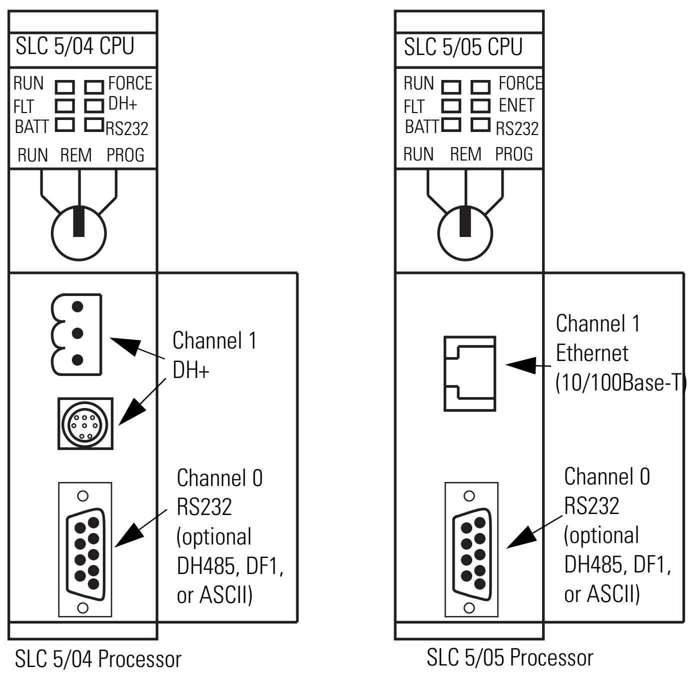
Installing the SLC 500 Processor

These sections give you information on installing your processor and getting it connected.

Identifying Processor Features

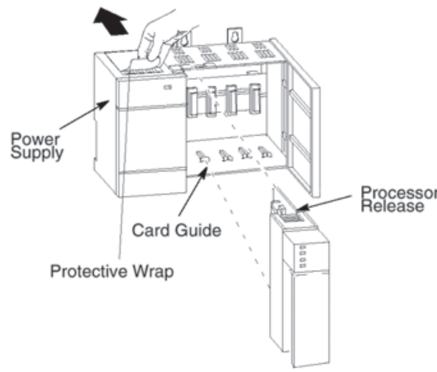
Refer to the following figures to identify the features of your SLC 5/04 or SLC 5/05 processor.

Processor Features



Make sure system power is off; then insert the processor into slot 0 of the 1746 I/O chassis.

Chassis Install



IMPORTANT	Insert the SLC 500 processor into the left slot (slot 0), as shown above. Remove the protective wrap after installing the processor.

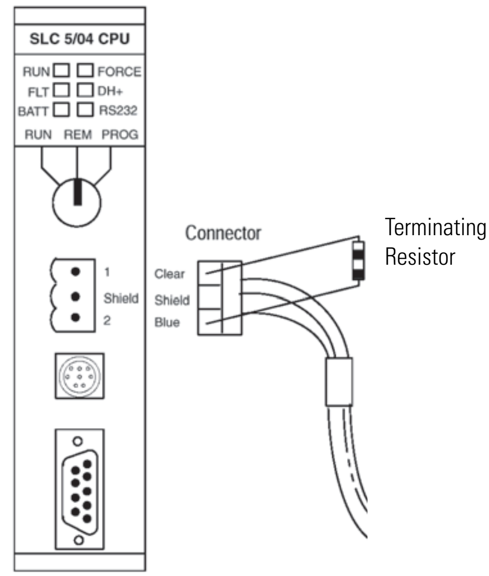
Establishing Data Highway Plus Communication

For the examples using the Data Highway Plus (DH+) network in Chapters 5 and 6 of this manual, we installed a 1784-PKTX communication card in the host platform and an SLC 5/04 processor with these default Channel 1 DH+ configurations:

- DH+ node address = 1
- Communication rate = 57.6 Kbps

Connect Channel 1 of the SLC 5/04 processor to the DH+ network by using the three-pin connector on the front of the module.

DH+ Network Connection



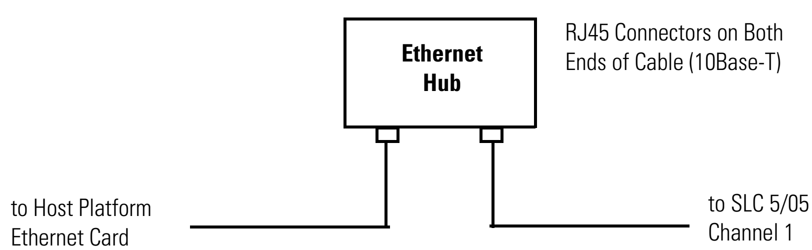
See Chapter 5 for information on configuring the SLC 5/04 processor's DH+ communication.

Installing an Ethernet SLC 500 Processor (SLC 5/05 Processor)

In order to communicate with your SLC 500 processor over an Ethernet network, you must install an Ethernet version of the processor (SLC 5/05 processor, catalog numbers 1747-L551, 1747-L552, or 1747-L553).

Connect channel 1 of the Ethernet SLC 5/05 processor to an Ethernet hub by using 10 Base-T cable as shown below.

Ethernet Hub Connection



See Chapter 5 for information on configuring the SLC 5/05 processor's Ethernet communication.

Configuring the RS-232 Port for the ControlNet Interface

If you need to communicate with your SLC 500 processor via a ControlNet network, you must install a 1747-KFC15 ControlNet interface module in the chassis with your processor. You can use either a SLC 5/04 or a SLC 5/05 processor. The 1747-KFC15 module connects to the SLC 500 processor via the processor's RS-232 port (channel 0).

TIP	You can use your RSLogix 500 programming software to set the SLC 5/04 processor's RS-232 channel configuration.

Installing the ControlNet RS-232 Interface Module

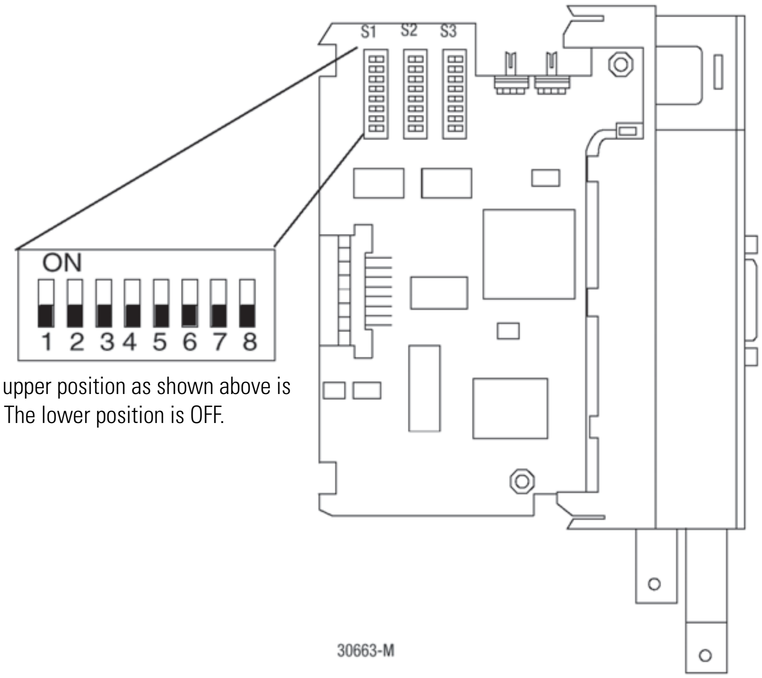
To communicate with the SLC 500 processor via a ControlNet network, you must install a 1747-KFC15 ControlNet RS-232 interface module in the 1746 I/O chassis close to the processor.

Configuring the 1747-KFC15 Module's RS-232 Port

IMPORTANT	The communication parameters of 1747-KFC15 module must match those of the SLC 500 processor.

The RS-232 serial port on the 1747-KFC15 module is configured by using three banks (S1, S2, and S3) of DIP switches mounted on the module's printed circuit board.

Dip Switch Setting



The upper position as shown above is ON. The lower position is OFF.

For the example application, we used the following configuration to match the configuration of the SLC 500 processor's RS-232 port described in the previous section.

Channel 0 Configuration

Attribute	Value
DF1 station address	0
Communication rate	19.2 Kbps
Full/Half-duplex	Full-duplex
Parity	None
Handshake	No handshaking
Diagnostic command execution	Disabled
Duplicate detect	Enabled
Error detect	CRC
Retries	3
DF1 ACK timeout	1.0 s

For this configuration, set the switches as shown in these tables.

Bank S1 DIP Switches

Switch	Setting	Position			
Switches 1…3	Upper digit of DF1 station address	**SW1**	**SW2**	**SW3**	**Digit**
		ON	ON	ON	0
Switches 4…6	Lower digit of DF1 station address	**SW4**	**SW5**	**SW6**	**Digit**
		ON	ON	ON	0
Switches 7…8		Both OFF			

Bank S2 DIP Switches

Switch	Setting	Position			
Switches 1…3	Upper digit of DF1 station address	**SW1**	**SW2**	**SW3**	**Digit**
		ON	OFF	OFF	19200
Switch 4	Full/Half-duplex	OFF = Full-duplex			
Switch 5	Parity	OFF = No parity			
Switch 6	Odd/even parity	Not applicable			
Switch 7	Handshake	OFF = Hardware handshake disabled			
Switch 8	Diagnostic command execution	OFF = Disabled			

Bank S3 DIP Switches

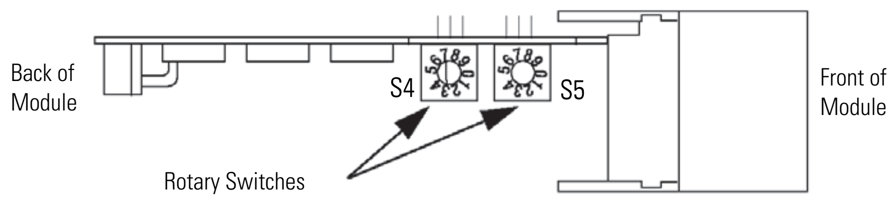
Switch	Setting	Position				
Switch 1	Duplicate detect	ON = Duplicate detect on				
Switch 2	Error detect	OFF = CRC error check				
Switches 3…4	Number of retries	**SW3**	**SW4**	**Number of retries**		
		ON	ON	0		
Switches 5…8	DF1 ACK time-out	**SW5**	**SW6**	**SW7**	**SW8**	**Time Out**
		OFF	OFF	OFF	OFF	3.2

Refer to the SLC 500 ControlNet RS-232 Interface User Manual, publication 1747-5.34, for information on setting and verifying the 1747-KFC15 and SLC 500 processor communication parameters.

Configuring the 1747-KFC15 Module's ControlNet Node Address

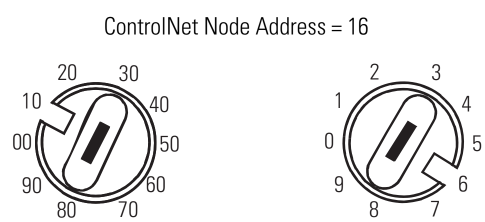
The 1747-KFC15 modules's ControlNet node address is set by rotary switches S4 and S5 on the top of the module. Switch S5 sets the upper digit of the address and S4 the lower. These switches can be turned by hand while holding the module in the orientation illustrated below.

Switch Location



We set the ControlNet node address to 16 for the example application.

Switch Setting



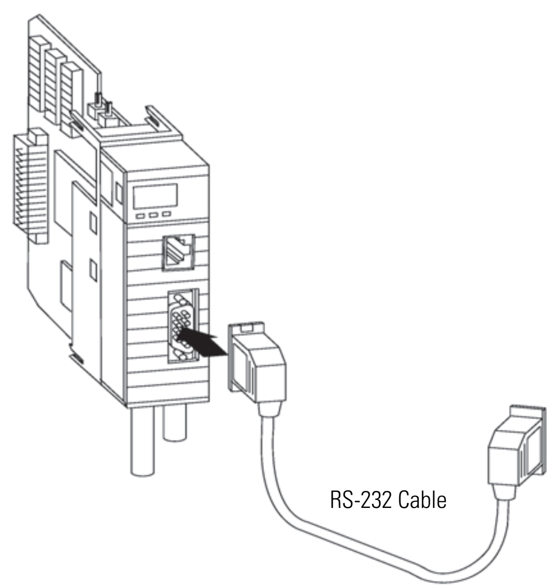
Refer to Chapter 5 for more information on configuring ControlNet communication for the example application.

Install the 1747-KFC15 Module in the Chassis

ATTENTION	Electrostatic discharge can damage semiconductor devices inside the 1747-KFC15 module. To guard against electrostatic damage, wear an approved wrist strap grounding device, or touch a grounded object to rid yourself of electrostatic charge before handling the products.

1. Remove power from your 1746 I/O chassis.
2. Install the 1747-KFC15 module into an empty I/O slot.

   The 1747-KFC15 module must be placed near enough to the SLC processor to connect the supplied RS-232 cable between them. We used slot 1 for the example application.
3. Connect the 1747-KFC15 module to your SLC 500 processor with the RS-232 cable.



Connecting the 1747-KFC15 Module to the ControlNet Network

Connect the 1747-KFC15 to the ControlNet cable system by using an approved ControlNet tap.

Refer to the ControlNet Coax Media Planning and Installation Guide, publication CNET-IN002, for complete instructions on connecting the tap to the ControlNet cable system.

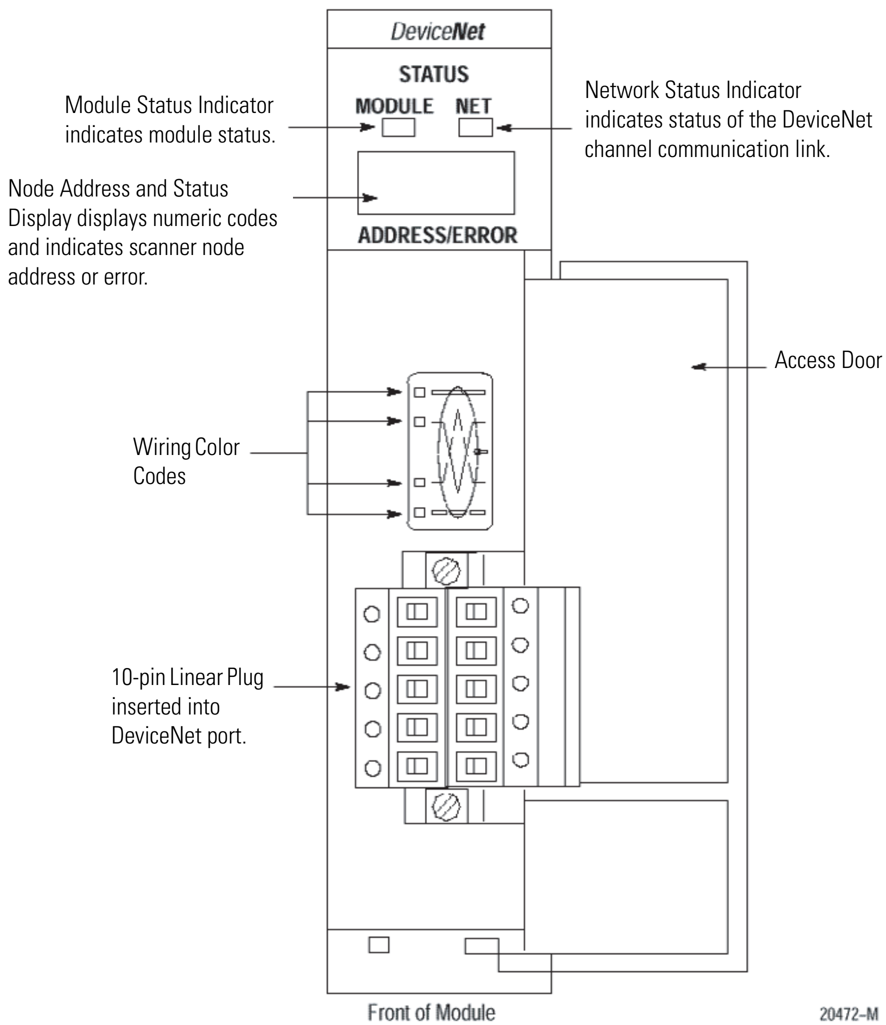
Installing the 1747-SDN Module

These sections give you information on installing your module and getting it connected.

Identifying Module Features

Use the following figure to identify the features of the 1747-SDN module.

1747-SDN Module Features



Install the 1747-SDN Module In the Chassis

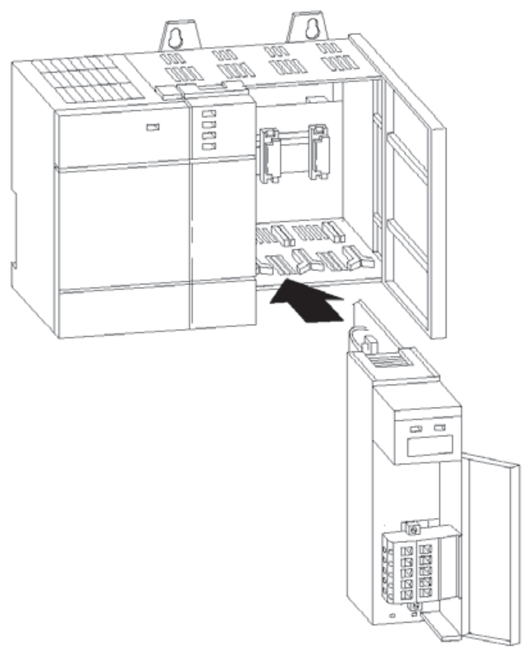
ATTENTION

Do not install the 1747-SDN module with the chassis power supply on. Installing the module with the chassis power supply on may damage the module.

Follow these steps to install your module into the chassis.

1. Turn off the chassis power supply.
2. Select a slot for the module in the chassis.

   You may use any slot except the leftmost slot, which is reserved for the SLC 500 processor.
3. Insert the module into the slot you have selected by applying firm, even pressure to seat the module in the chassis backplane connectors.



Connect the 1747-SDN Module to the DeviceNet Network

Follow these steps to connect your module to the DeviceNet network.

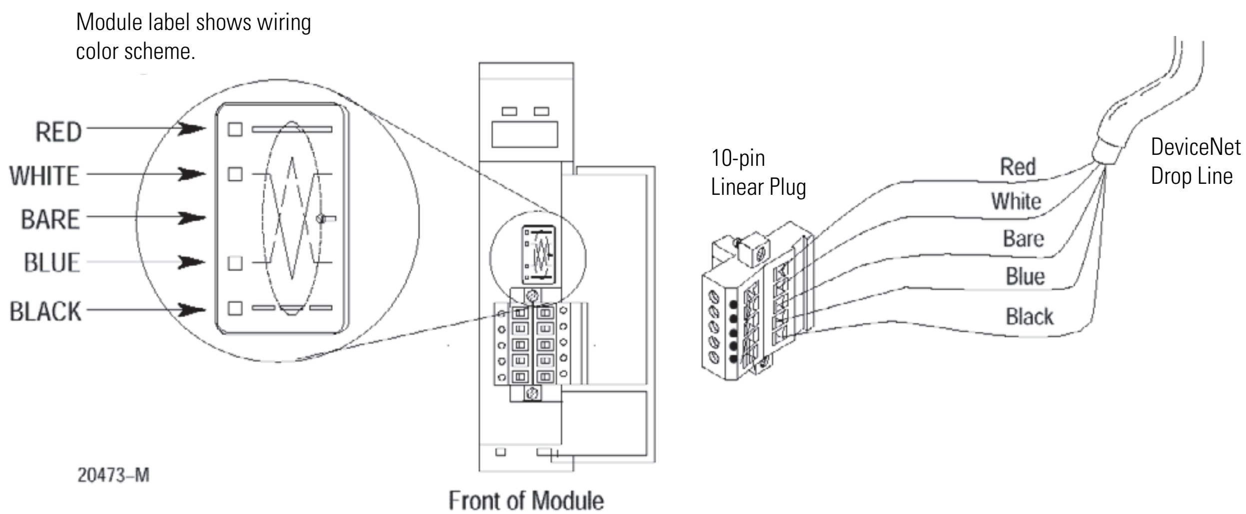
1. Turn off the network power supply.

ATTENTION

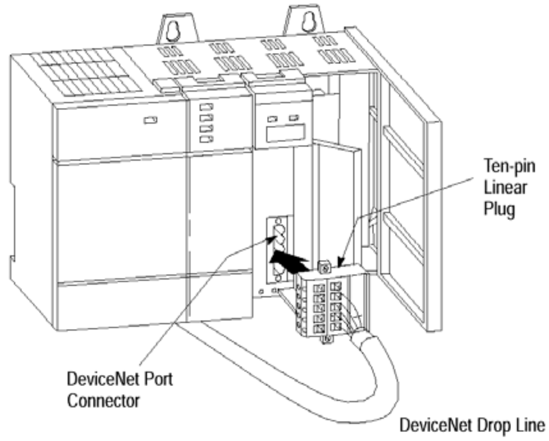
Do not wire the 1747-SDN module with the network power supply on. Wiring the module with the network power supply on may short your network or disrupt communication.

2. Connect the DeviceNet drop line to the 10-pin linear plug.

   Match the wire insulation colors to the colors shown on the label.



3. Locate the DeviceNet port connector on the front of the module, and insert the 10-pin linear plug into the connector.



TIP You use RSNetWorx for DeviceNet software to configure the module's DeviceNet address and communication rate, if needed. This is done when configuring the DeviceNet network.

See page 52.

For additional information about installing the 1747-SDN module, see the 1747-SDN DeviceNet Scanner Module Installation Instructions, publication 1747-IN058.

Installing the RediSTATION Operator Interface

Begin installing the RediSTATION operator interface by removing the six screws fastening the cover and setting the DIP switches inside.

DIP Switch Settings

Set this position	To this value	
1 2 3 4 5 6	1 - On 1 - On 1 - On 0 - Off 0 - Off 0 - Off	Node Address(1)
7 8	0 - Off 1 - On	Data Rate(2)
9 10	0 - Off 0 - Off	

(1) The DeviceNet address is 000111 (node 7).

(2) The data rate is 10 (500 KB). The output fault rate is 0 (outputs turned off). The output flash rate is 0 (outputs tuned off).

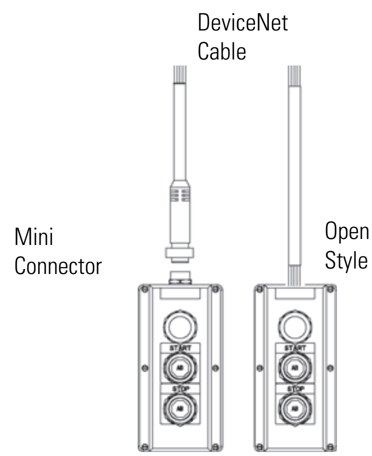
See Chapter 2 of the RediSTATION Operator Interface User Manual, publication 2705-UM001, for complete information about setting the DIP switches to configure the node address, data rate, output flash rate, and output fault state.

Refer to the following illustration as you connect the RediSTATION operator interface to the network.

TIP You do not need to disconnect incoming power from the DeviceNet network before connecting the RediSTATION operator interface.

The DeviceNet cable connects directly to the mini connector on the top of the RediSTATION enclosure or through the conduit opening (open style).

RediSTATION Connection

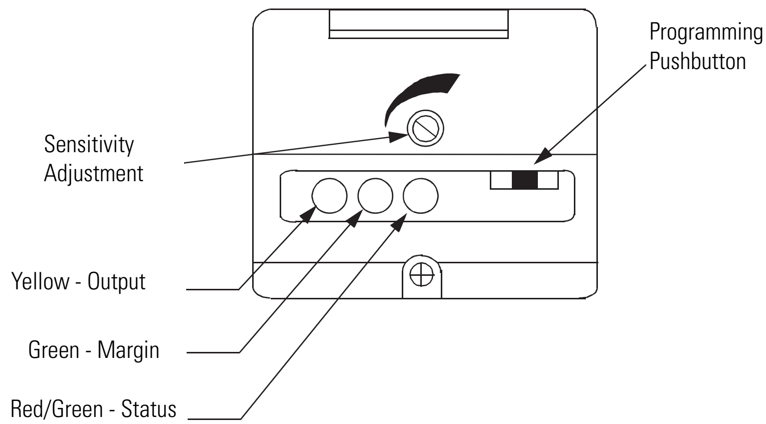


Installing the Series 9000 Photoeye

Connect the photoeye to the network and configure the photoeye as follows:

- Node address: 9
- Operating mode: Light Operate (default)
- Communication rate: 500 Kbps

Top View of Series 9000 Photoeye

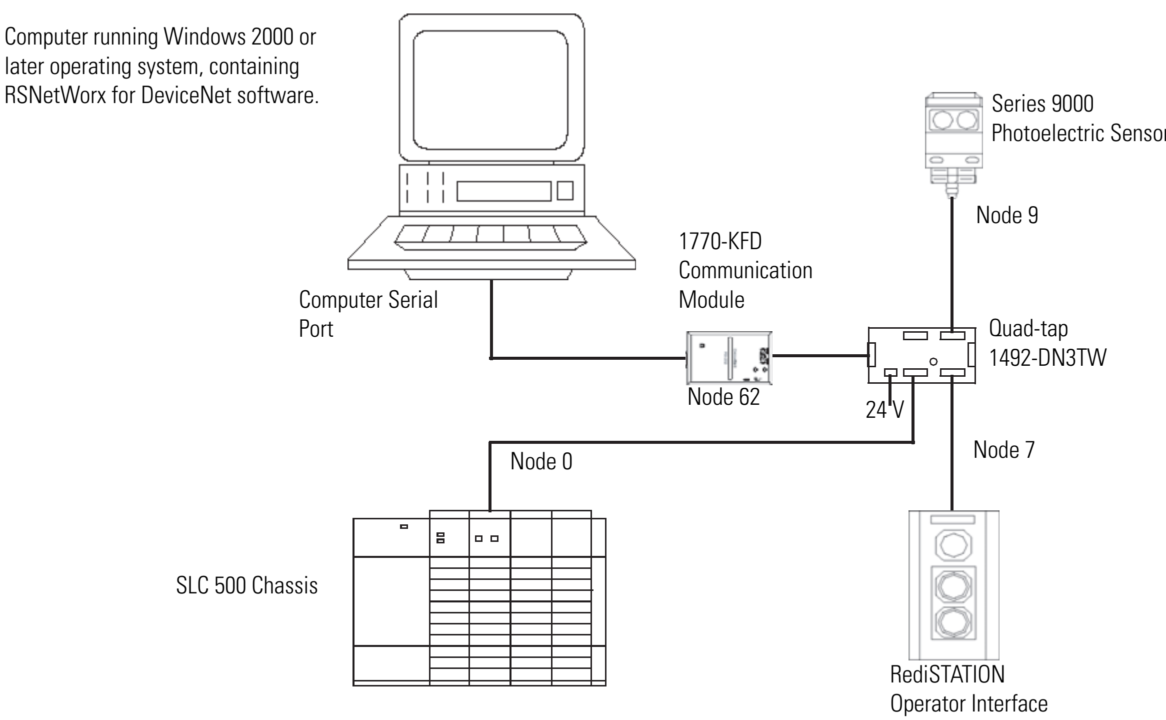


For detailed directions, see the instructions that came with your photoeye.

How Your Network Will Look

When you have finished installing all the devices, the network should look similar to this.

Typical Network



IMPORTANT	Make sure each end of the DeviceNet trunk cable is properly terminated with a resistor. Refer to the DeviceNet Media Design and Installation Guide, publication DNET-UM072, for information.

What's Next?

The next step is to configure the 1747-SDN module and perform I/O data mapping through RSNetWorx for DeviceNet software.

Configuring the DeviceNet Network

What This Chapter Contains

This chapter describes how to configure the DeviceNet network by using RSLinx and RSNetWorx for DeviceNet software.

Installing the Software

Follow these steps to install the RSLinx and RSNetWorx for DeviceNet software.

1. Insert the software CD-ROM in the drive.

   The CD-ROM supports Windows Autorun. If you have Autorun configured, the installation automatically starts when you insert the CD-ROM in your drive. If you do not have Autorun configured, perform steps 2 and 3.

2. Select Run from the Windows Start menu.

3. Browse for the Setup program on the CD ROM and open it.

4. Follow the prompts that appear as you install the software.

After software installation is complete, you use RSLinx software to configure your DeviceNet driver and RSNetWorx for DeviceNet software to configure the network.

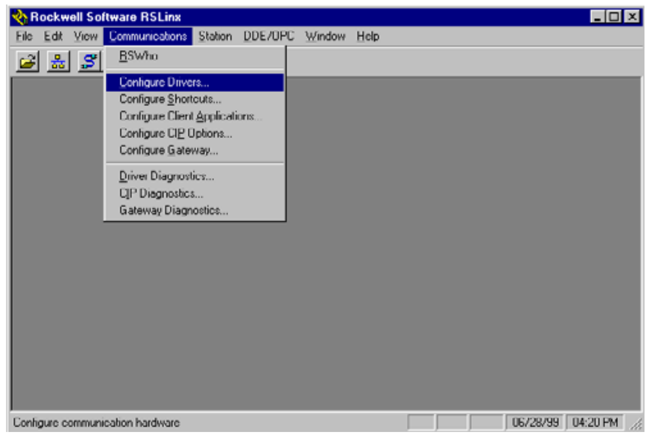
Use RSLinx Software to Configure the DeviceNet Driver

Follow these steps to configure your DeviceNet driver.

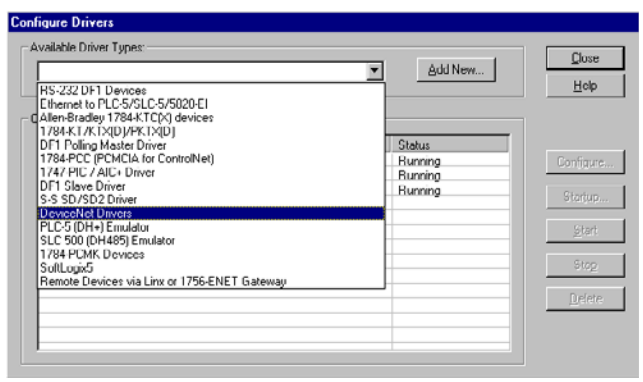
1. Start RSLinx software.

   The RSLinx main dialog opens.

2. Select Configure Drivers from the Communication menu.

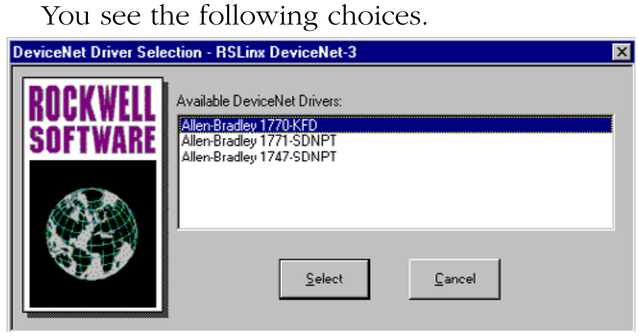


   The following dialog appears.



3. Select DeviceNet Drivers from the above pull-down menu and click Add/New.

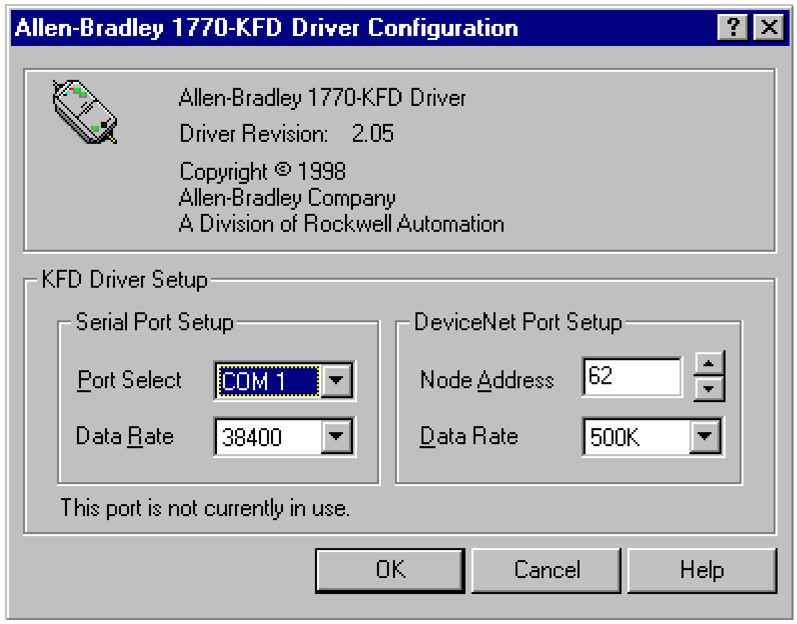
   You see the following choices.



4. Select the Allen-Bradley 1770-KFD driver.

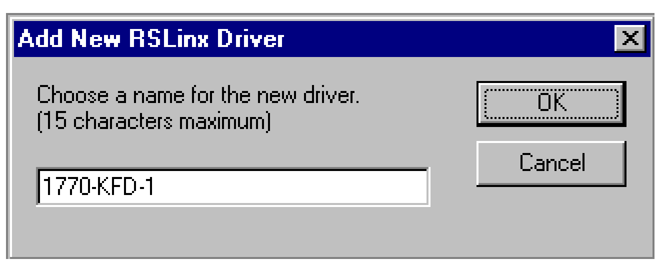
   The Driver Configuration dialog appears.

TIP Your driver setup depends on your particular system setup (COM port, communication rate, node address). Choose the appropriate settings for your system.



5. Configure the driver by using the example above as a guide and click on OK.

   The software takes a few seconds to configure the driver. When it is done the following prompt appears.



6. Select the default driver name 1770-KFD-1 and click OK.

7. Close RSLinx software.

Use the driver you just configured to browse and configure the network with RSNetWorx for DeviceNet software.

Using RSNetWorx for DeviceNet Software to Configure the 1747-SDN Module Scanlist

Use RSNetWorx for DeviceNet software to do the following tasks:

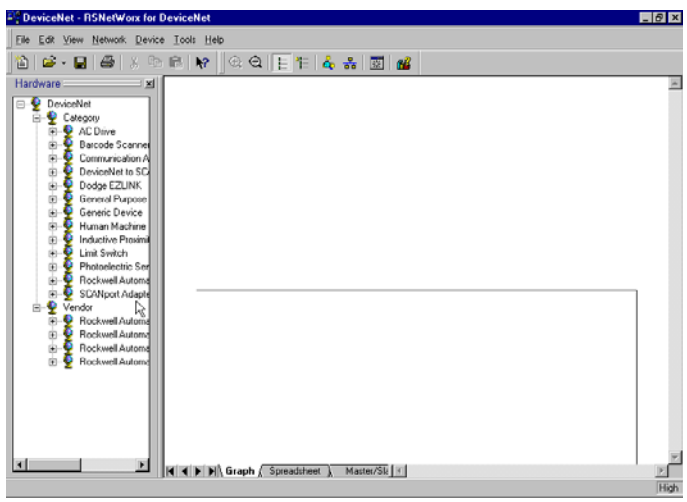
- Set up an online connection
- Set the 1747-SDN node address
- Configure the I/O devices
- Download and save your configuration

Set Up an Online Connection

Follow these steps to set up an online connection to the DeviceNet network by using the 1770-KFD driver.

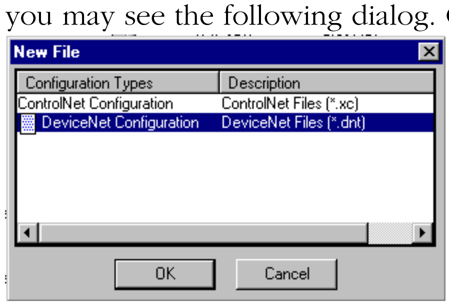
1. Start RSNetWorx for DeviceNet software.

   The following dialog appears.

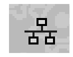


2. From the File menu, choose New.

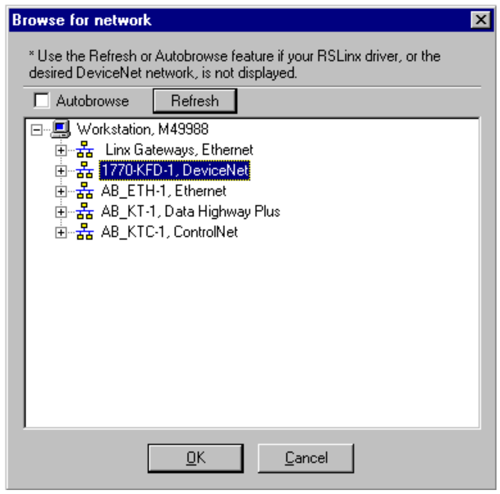
   If you have a ControlNet network configured on your system you may see the following dialog. Otherwise, proceed to step 4.



3. Highlight DeviceNet Configuration and click OK.

4. Click Online  on the toolbar.

   A list of the available drivers in RSLinx software appears. Your list may appear different from that shown below, depending upon the drivers you have configured on your system.

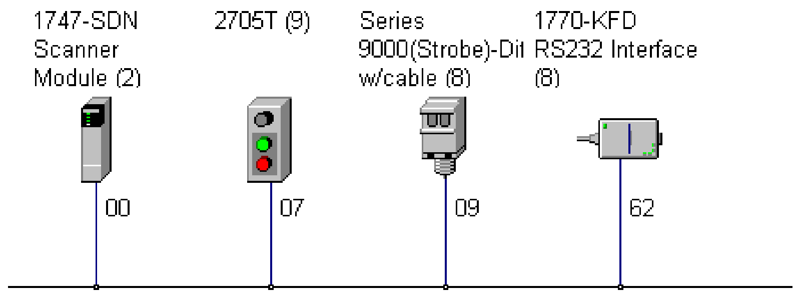


5. Select the 1770-KFD-1, DeviceNet driver and click OK.

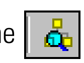
   You are prompted to upload or download devices before going online.

6. Click OK to upload the devices and go online.

RSNetWorx for DeviceNet software begins browsing for network devices. When the software is done browsing, the network displayed on your screen should appear similar to the one shown below.



TIP RSNetWorx for DeviceNet software performs a one-shot browse when you go online or choose the browse feature. The software polls for devices one time and displays the results. If a node which was online later goes offline, there is no live indication in RSNetWorx for DeviceNet software. You must manually perform a browse to detect the missing node.

To manually perform the browse, press the  button.

TIP If RSNetWorx for DeviceNet software fails to find a device, check the physical connection to the device. If the physical connection is intact, verify that the device's communication rate is the same as the 1770-KFD driver's communication rate.

Set the 1747-SDN Node Address

Once the devices are uploaded, their node addresses appear to the right of their icons. For the example application, the 1747-SDN module should have a node address of 0 (or 00). If you need to change a module's node address, use the following procedure.

TIP You can use this procedure to change the node address of other devices on the network (for example, the photoeye). You can also change the network data rate (communication rate) of some devices. Power must be cycled for baud rate changes to take effect.

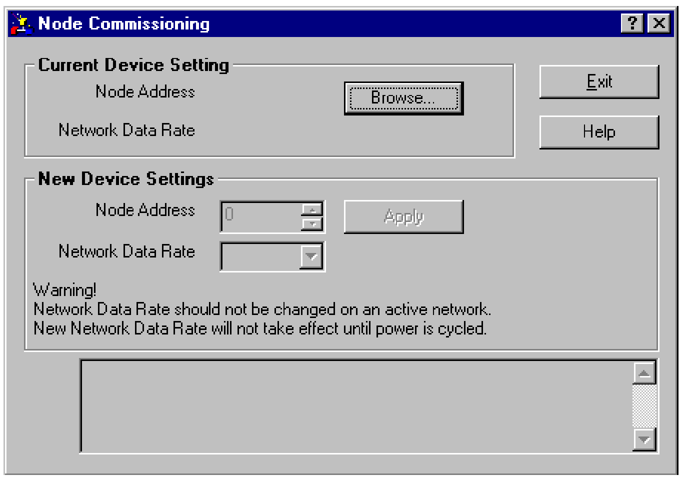
If 00 appears to the right of the 1747-SDN icon and you do not need to change the node address or baud rate of any device, skip the remainder of this section and go to Configure the I/O Devices on page 55.

IMPORTANT	The network must not be active when performing node commissioning on the 1747-SDN module. Make sure the processor is in Program mode. (Note that this applies only to the 1747-SDN module. You may commission other devices with the processor in Run mode.)

Follow these steps to set the node address.

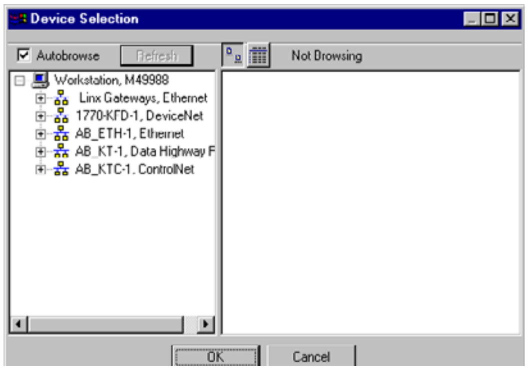
1. From the Tools menu, choose Node Commissioning.

   The Node Commissioning dialog appears.



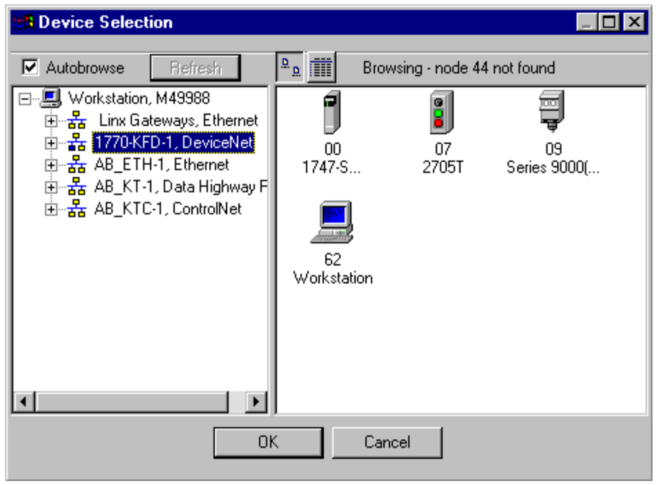
2. Click Browse.

   The Device Selection dialog appears.



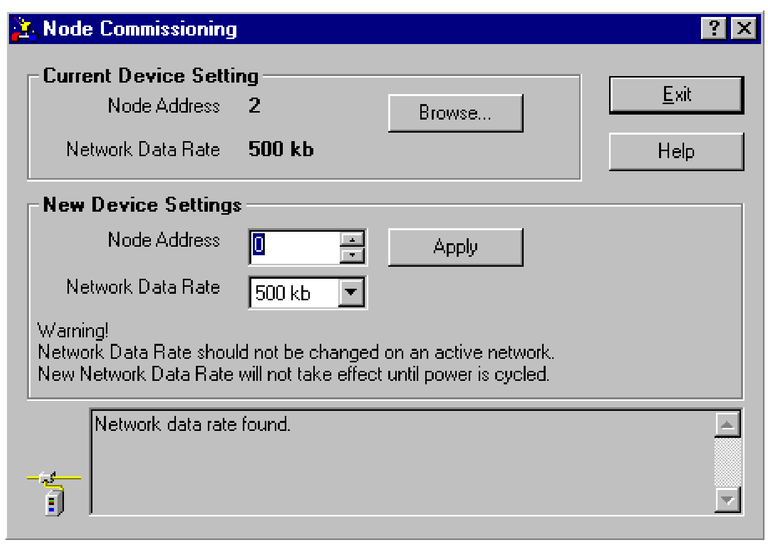
3. Select the 1770-KFD driver.

   The devices on the network appear in the right panel of the dialog.



4. Select the device you are commissioning in the right panel and click OK.

   The Node Commissioning dialog appears with the current settings for your 1747-SDN module. Your dialog should look similar to the one shown below.



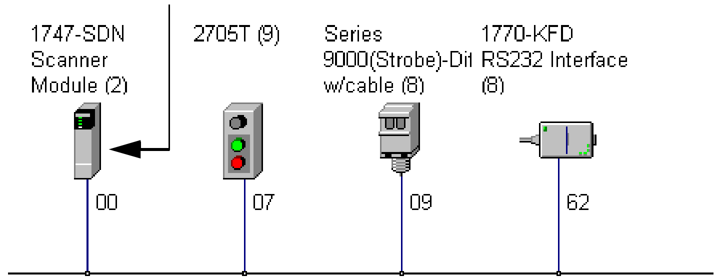
5. Enter a **0** in the New Device Settings: Node Address box.

6. Click Apply and exit the dialog.

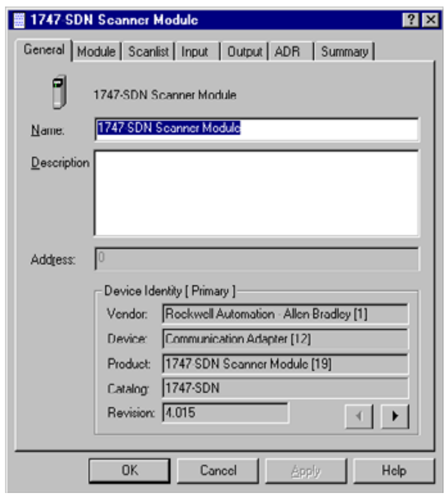
Configure the I/O Devices

Follow these steps to configure the I/O devices.

1. Double-click the 1747-SDN module icon.

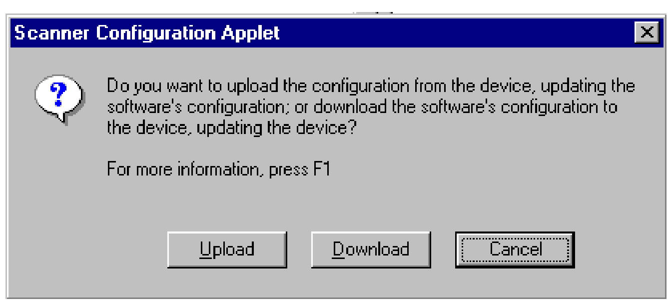


The following dialog appears.



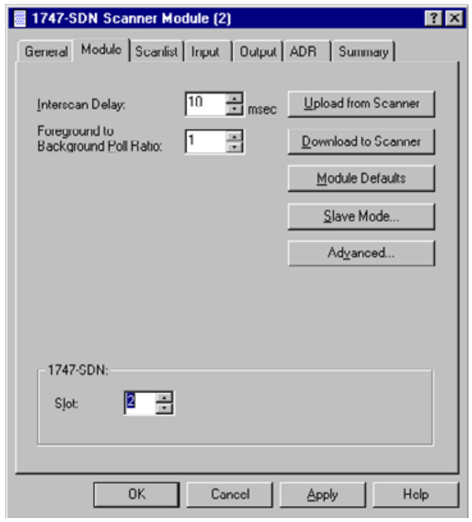
2. Click the Module tab.

You receive the following prompt.



3. Click Upload.

After uploading, the following dialog appears.

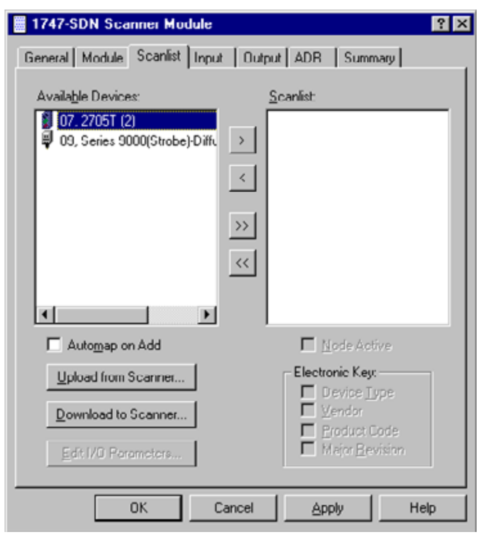


4. Verify the 1747-SDN module slot number is correct for your system.

We used slot 2.

TIP We used the Module Defaults for the other settings. For an explanation of these settings click Help.

5. Select the Scanlist tab.



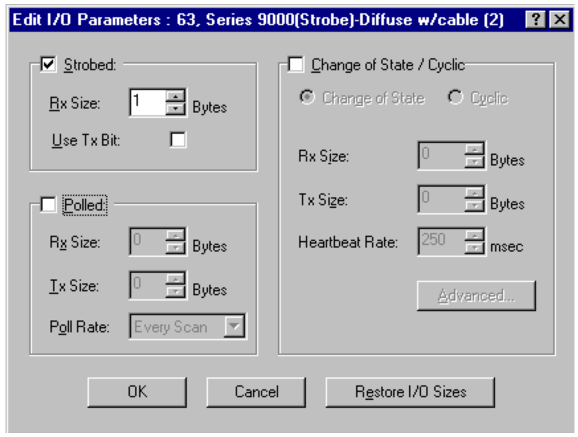
6. Verify that the Automap on Add box is not checked.

7. Click the double arrow >> to add the photoeye and RediSTATION operator interface to the Scanlist.

Verify the Photoeye Configuration

1. Double-click the photoeye in the Scanlist.

   The Edit I/O Parameters dialog appears for the photoeye.



   The I/O parameters define the configuration for the device in terms of how much and what data the device exchanges with the 1747-SDN module. By default, the photoeye will send 1 byte when it receives the strobe request.

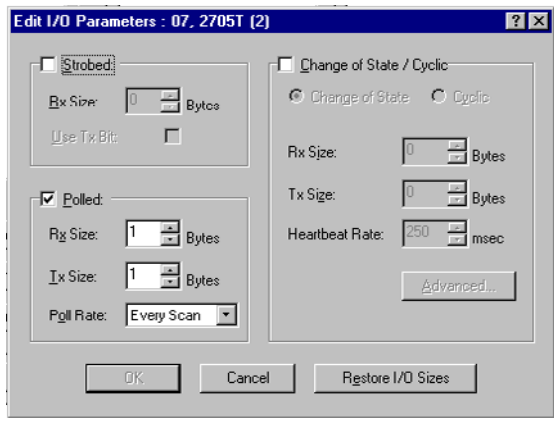
2. Verify that the photoeye parameters are set as shown above.

3. Click OK to close the photoeye Edit I/O Parameters dialog.

Verify the RediSTATION Configuration

1. Double-click the RediSTATION operator interface in the Scanlist dialog.

The Edit I/O Parameters dialog appears for the RediSTATION operator interface.



2. Verify that the Polled box is checked and that the Rx Size and Tx Size are each 1 byte.

3. Click OK to close the Edit I/O Parameters dialog for the RediSTATION operator interface.

4. Click OK again.

   You are prompted to download the changes to the 1747-SDN module.

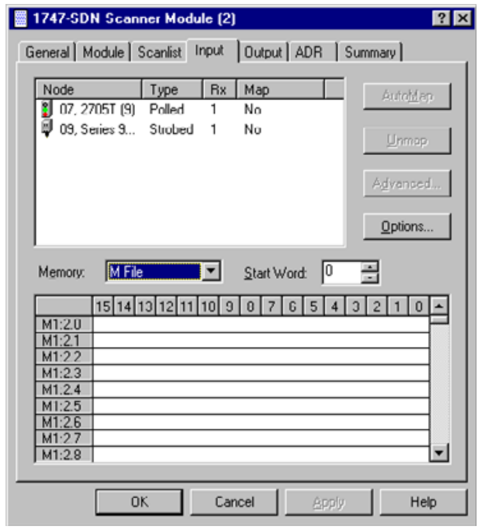
5. Click Yes to download the new configuration.

AutoMap the Devices into the Scanlist

Follow these steps to automatically map the photoeye and RediSTATION operator interface to the SLC 500 processor.

TIP	If you want to know how to map the devices manually, click Help at the bottom of the dialog and select **Map device input data manually**.

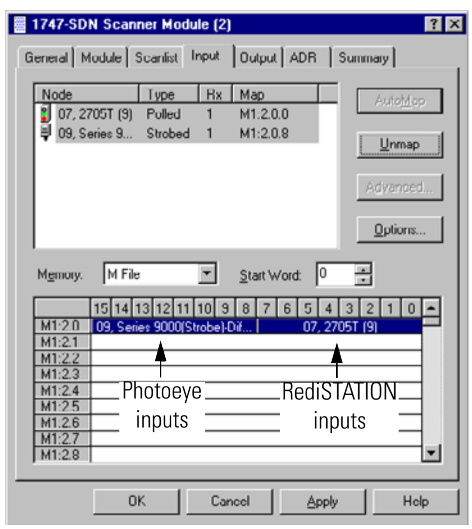
1. Double-click the 1747-SDN module icon and select the Input tab.



2. Select M File in the Memory field.

3. Highlight the RediSTATION operator interface and the photoeye and click AutoMap.

   The resulting device mapping appears in the lower panel.



In this example, the inputs from the RediSTATION operator interface appear in the M1 file for the device in slot 2 as word 0, bits 0…7.

Recall from Chapter 2 that the START button is bit 1 and the STOP button is bit 0. Therefore, the addresses for the RediSTATION inputs are:

START - M1:2.0.1

STOP - M1:2.0.0

The input from the photoeye appears in the M1 file for the device in slot 2 as word 0, bits 8…15.

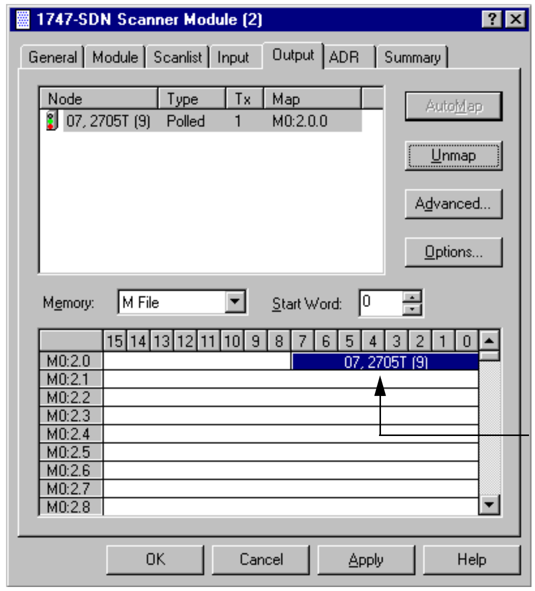
Recall from chapter 3 that the input bit is bit 0. Therefore, the address of the photoeye input is:

M1:2.0.8

4. Note the addresses assigned to the START and STOP buttons and the photoeye in your system.

   You enter these addresses in the example ladder program.

5. Select the Output tab.



After mapping, the RediSTATION output address appears here.

6. Select M File in the Memory field.

7. Highlight the RediSTATION operator interface and click AutoMap.

   In this example, the output to the RediSTATION operator interface appears in the M0 file for the device in slot 2 as word 0, bits 0…7. Recall from Chapter 3 that the indicator light is output bit 0. Therefore, the address for the RediSTATION indicator light is:

   M0:2.0.0

8. Note the address assigned to this output in your system.

   You enter this address in the example ladder logic program in the following chapter.

Download and Save Your Configuration

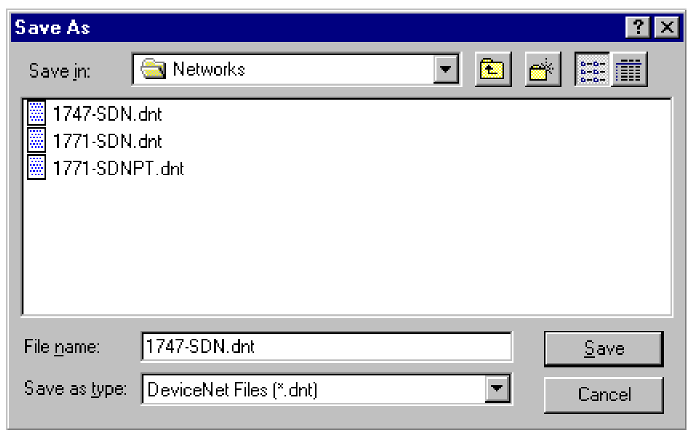
1. Click the **Scanlist** tab and then Download to Scanner.

2. Select All Records.

3. Click Download to download the configuration to the 1747-SDN module.

4. Click OK to complete the DeviceNet scanner configuration.

5. From the File menu, choose Save As.



6. Save the configuration to a DeviceNet file.

7. Close the RSNetWorx for DeviceNet software.

What's Next?

The next chapter describes how to configure the DeviceNet network remotely from an Ethernet, ControlNet, or Data Highway Plus network.

Communicating with the DeviceNet Network from Another Network

What This Chapter Contains

This chapter describes how to communicate with the DeviceNet network from another network, by using the SLC 500 pass-through feature. This feature can be used to adjust and fine tune the nodes on your network. Examples are provided for communicating from an Ethernet network and a Data Highway Plus network.

ATTENTION

The pass-through feature is not intended to replace a 1770-KFD, 1770-PCD, 1770-PCID, or 1770-PCIDS connection to the network.

Pass-through is intended only for fine tuning and adjustment of your network devices. Do not attempt to configure your entire network by using a pass-through driver or a time-out may occur.

The pass-through method is not suitable for real-time monitoring of your network devices.

The following table describes what this chapter contains and where to find specific information.

Additional Resources

These documents contain additional information on configuring other networks.

Resource	Description
SLC Modular Style Hardware Installation and Operation Manual, publication 1747-UM011	Provides information about the Data Highway Plus network.
Ethernet SLC 500 Processors Quick Start for Experienced Users, publication 1747-10.4	Provides information about the Ethernet interface.
Comer, Douglas E., Internetworking with TCP-IP, Volume 1: Protocols and Architecture, 2nd ed. Englewood Cliffs, N.J.:Prentice-Hall, 1995. ISBN 0-13-216987-8	Provides information about TCP/IP protocol and networking in general.
Tannebaum, Andrew S. Computer Networks, 2nd ed. Englewood Cliffs, N.J.: Prentice-Hall, 1989. ISBN 0-13-162959-X	

You can view or download publications at http://literature.rockwellautomation.com. To order paper copies of technical documentation, contact your local Rockwell Automation distributor or sales representative.

System Requirements

To use the pass-through feature, you must have RSLinx software, version 2.10 or later, and a 1747-SDN module at firmware revision 4.015 or later.

IMPORTANT To enable pass-through access with a SLC 500 processor, you must use RSLogix 500 software to configure the M0 and M1 files associated with the 1747-SDN module with a length of 395 words. You can access the M file configuration by launching I/O Configuration within RSLogix 500 software, and then selecting the 1747-SDN module.

The SLC 500 processor must be placed in Run mode at least once since its last power cycle for 1747-SDN pass-through transactions to succeed.

IMPORTANT You must have previously set up the network you will use to communicate with the DeviceNet network and have installed and configured the appropriate drivers and interface hardware.

The SLC 500 chassis used for these examples was set up with the following hardware mapping. The SLC 5/04 processor was used for the DH+ example. The SLC 5/05 processor was used for the Ethernet example.

Hardware Mapping

Module	Slot	DH+ Address	Ethernet Address	DeviceNet Address
SLC 500 5/04 Processor	0	1	-	-
SLC 500 5/05 Processor	0	-	130.130.130.2	-
1747-SDN	2	-	-	0

Communicating with the DeviceNet Network via an Ethernet Network

Before performing this example, the Ethernet network must be configured and running. You must use an Ethernet SLC 5/05 processor (catalog number 1747-L551, 1747-L552, or 1747-L553).

Establishing Ethernet pass-through communication involves four main steps.

1. Configure the Ethernet to SLC-5 driver to communicate with the SLC 5/05 processor over the Ethernet Network by using RSLinx software.

   This procedure is described starting on page 66.

2. Configure the DeviceNet pass-through driver to communicate with the 1747-SDN module via the SLC 5/05 processor and the Ethernet network by using RSLinx software.

   This procedure is described starting on page 68.

3. Configure the SLC 5/05 processor's Ethernet communication channel by using RSLogix 500 software when you create the example ladder program.

   The Ethernet channel configuration is described in Appendix B.

4. Use the pass-through driver with RSNetWorx for DeviceNet software to adjust and tune your DeviceNet network.

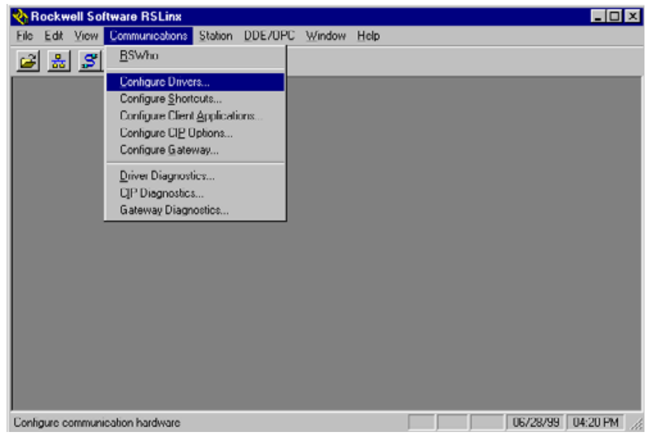
   The procedure for doing this is described starting on page 73.

Configure the Ethernet Devices Driver

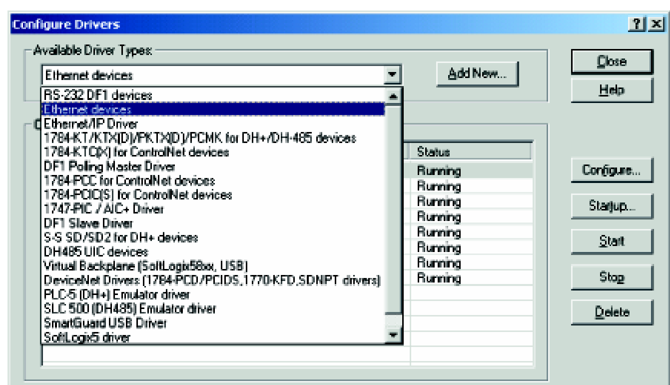
To communicate with your SLC 500 processor over the Ethernet network you must configure the Ethernet to SLC 500 driver. Follow these steps to configure the driver by using RSLinx software.

1. Start RSLinx software.

2. From the Communication menu, choose Configure Drivers.

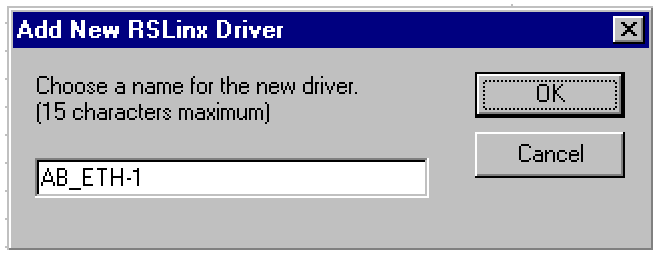


The Configure Drivers dialog appears.



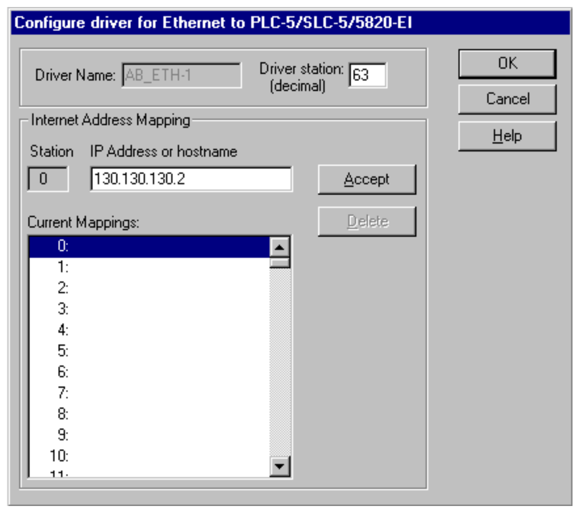
3. From the Available Driver Types pull-down menu, select Ethernet to PLC-5/SLC-5/5820-E and click Add New.

You are prompted to choose a name for the new driver.



4. Enter an appropriate driver name (for example, AB_ETH-1) and click OK.

   The Configure driver for Ethernet to PLC-5/SLC-5/5820-EI dialog opens.

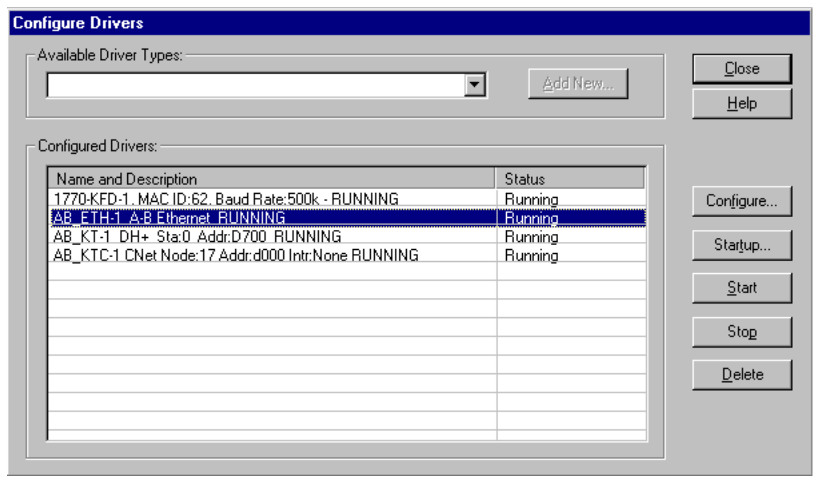


5. Enter the IP address of the SLC 5/05 processor in the IP address or hostname field (130.130.130.2 in this example).

IMPORTANT	You must configure channel 0 of the SLC 5/05 processor with the same IP address you enter here.

6. Click Accept and then OK.

   The new driver is added to the list of Configured Drivers in RSLinx software. (Your list contains the drivers you have configured.)

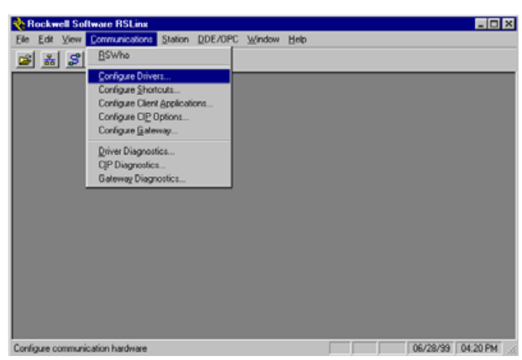


Configure the DeviceNet Network Pass-through Driver

Before you can communicate with the 1747-SDN module via the Ethernet network, you must configure the DeviceNet pass-through driver (catalog number 1747-SDNPT) by using RSLinx software, version 2.10 or later. You must also have configured the Ethernet to PLC-5/SLC-5/5820-EI driver as described in the previous section and configured the SLC 5/05 processor's Ethernet communication channel as described in Appendix B.

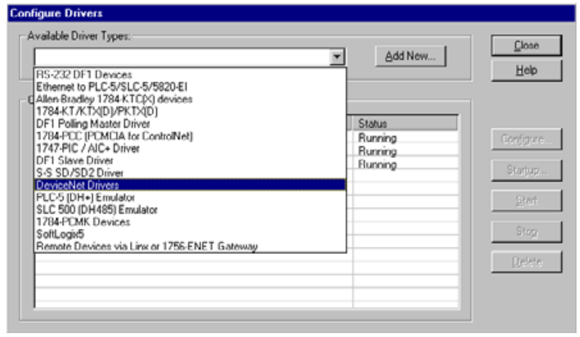
Follow these steps to configure the DeviceNet network Pass-through driver.

1. Start RSLinx software.



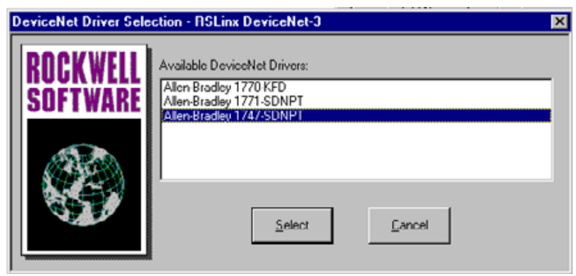
2. From the Communication menu, choose Configure Driver.

   The Configure Drivers dialog appears.



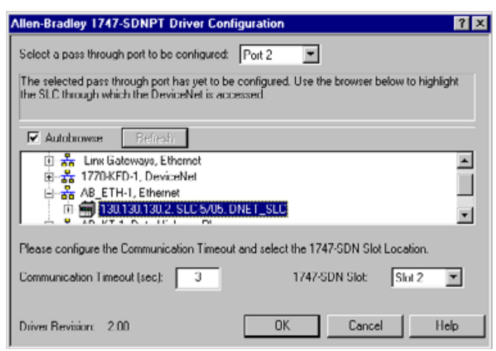
3. From the Available Driver Types pull-down menu, select DeviceNet Drivers and click Add New.

   The following list of drivers appears.



4. Select the Allen-Bradley 1747-SDNPT driver.

   The Allen-Bradley 1747-SDNPT Driver Configuration dialog opens.



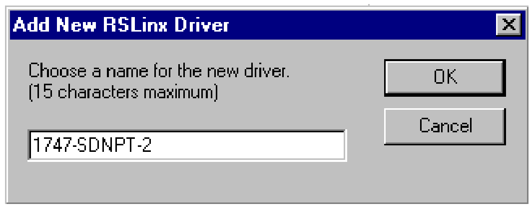
5. From the 1747-SDN Slot pull-down menu, select a pass-through port to be configured, for example Slot 2.

6. Expand your Ethernet driver (AB_ETH-1) and highlight the SLC 5/05 processor.

7. Verify that the 1747-SDN Slot is correct.

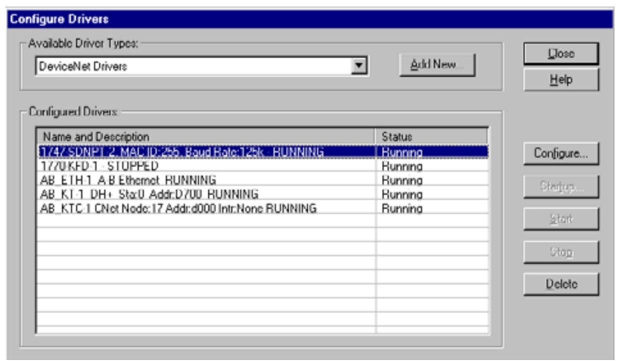
8. Click OK.

   You are prompted to enter a name for the driver.



9. Enter an appropriate driver name (for example, 1747-SDNPT-2) and click OK.

   The new driver is added to the list of Configured Drivers in RSLinx software. (Your list contains the drivers you have configured for your system.)



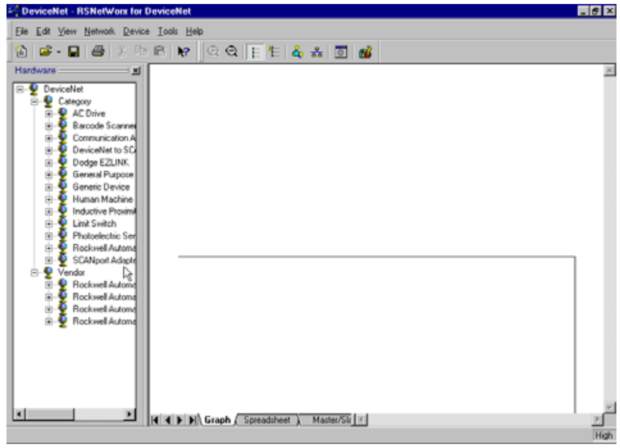
10. Close or minimize RSLinx software.

Communicate with the DeviceNet Network

You can use RSNetWorx for DeviceNet software to communicate with the DeviceNet network via the Ethernet network once the Ethernet pass-through driver is configured.

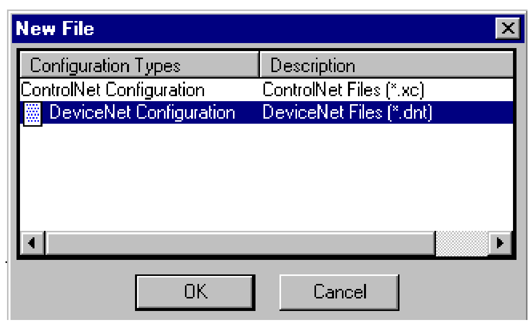
Follow these steps to communicate with the DeviceNet network.

1. Start RSNetWorx for DeviceNet software.

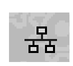


2. From the File menu, choose New.

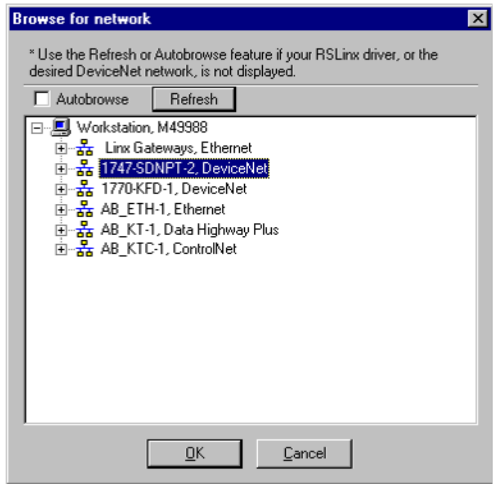
   You may see the following dialog if you have the ControlNet network configured on your system. Otherwise, proceed to step 4.



3. Select DeviceNet Configuration and click OK.

4. Click Online  on the toolbar.

The Browse for network dialog appears. You see the drivers you have configured on your system.



5. Highlight 1747-SDNPT-2, DeviceNet driver and click OK.

You receive the following prompt.

6. Click OK to upload the devices.

RSNetWorx for DeviceNet software begins browsing for network devices.

ATTENTION

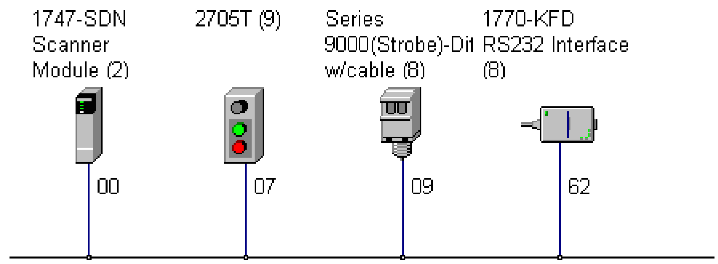
Performing a pass-through browse via the Ethernet network takes longer than browsing by using the 1770-KFD DeviceNet driver as described in chapter 4.

Note that due to the time required, the pass-through method is not suitable for configuring a network nor for real time monitoring of your network devices.

The network displayed on your screen should look similar to the one shown below when RSNetWorx for DeviceNet software is finished browsing.

TIP

If RSNetWorx for DeviceNet software fails to find a device, check the physical connection to that device.



You are now online to the DeviceNet network via the Ethernet network.

See pages 55…62 of this manual for examples of how to use RSNetWorx for DeviceNet software to adjust network parameters.

Communicate with the DeviceNet Network via a DH+ Network

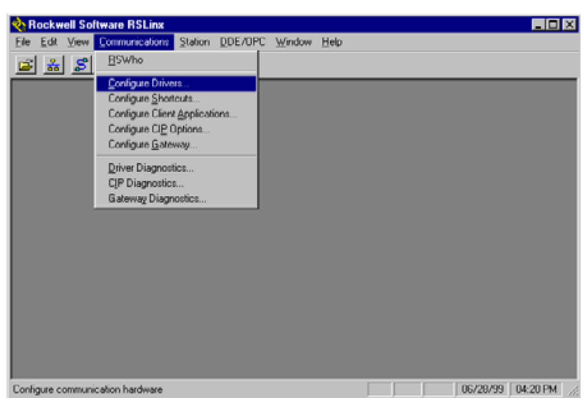
Before performing this example, the DH+ network must be configured and running. This example uses a SLC 5/04 processor. The SLC 5/04 processor lets you operate DH+ communication protocol by means of DH+ communication channel 1.

Configure the DeviceNet Pass-through Driver

You must first configure the DeviceNet pass-through driver (catalog number 1747-SDNPT) with a DH+ port by using RSLinx software, version 2.10 or later, before you can communicate with the 1747-SDN module via a DH+ network.

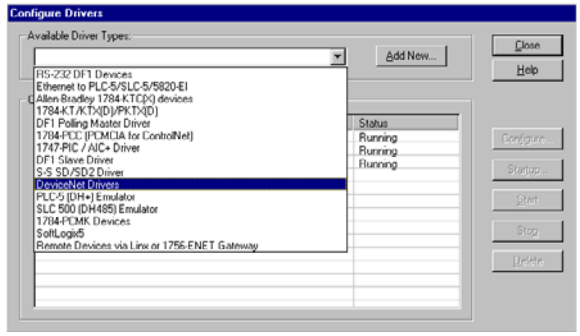
Follow these steps to configure the DeviceNet pass-through driver.

1. Start RSLinx software.



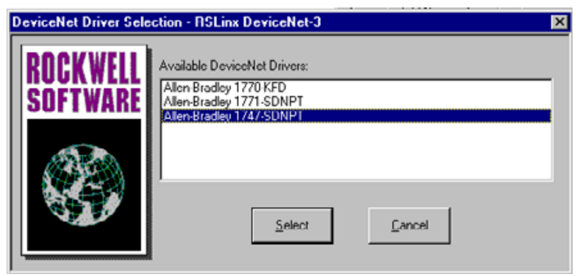
2. From the Communication menu, choose Configure Drivers.

   The Configure Drivers dialog appears.



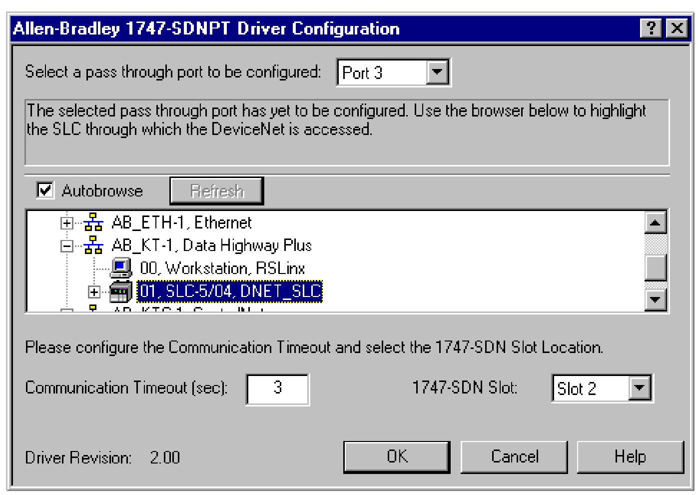
3. From the Available Driver Types pull-down menu, select DeviceNet Drivers and click Add New.

   You see the following list of drivers.



4. Select the Allen-Bradley 1747-SDNPT driver.

   The Driver Configuration dialog appears.



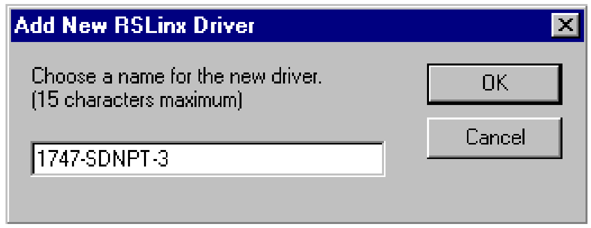
5. From the 1747-SDN Slot pull-down menu, select a pass-through port to be configured, for example Slot 3.

6. Expand your DH+ driver (AB_KT-1 above) and select the SLC 500 processor.

7. Verify that the 1747-SDN Slot is correct.

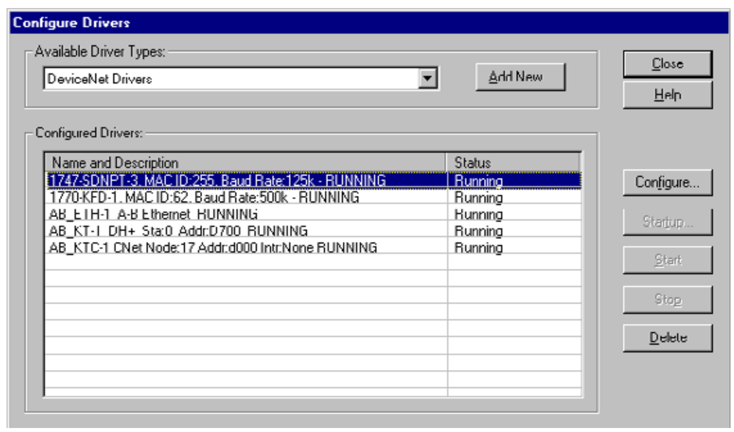
8. Click OK.

   You are prompted to enter a name for the driver.



9. Enter an appropriate driver name (for example, 1747-SDNPT-3) and click OK.

   The new driver is added to the Configured Drivers in RSLinx software. (Your list contains the drivers you have configured for your system.)



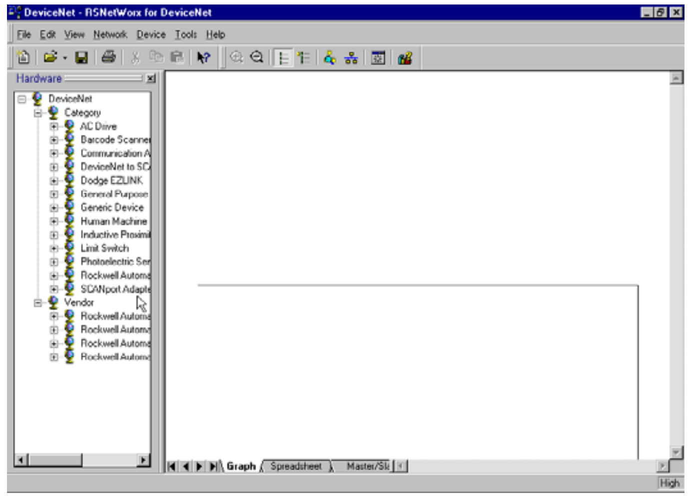
10. Close or minimize RSLinx software.

Communicate with the DeviceNet Network

You can use RSNetWorx for DeviceNet software to communicate with the DeviceNet network via the DH+ network once you have the DH+ pass-through driver configured.

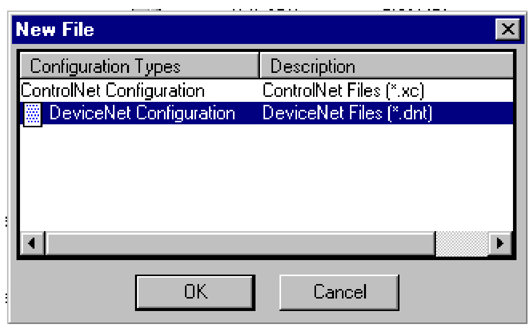
Follow these steps to communicate by using the DH+ network.

1. Start RSNetWorx for DeviceNet software.

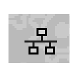


2. From the File menu, choose New.

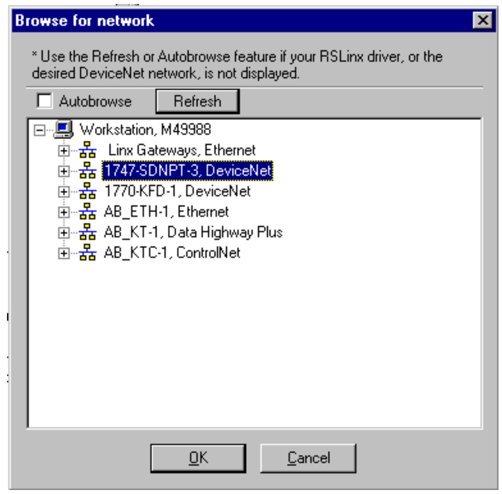
   If you have the ControlNet network configured on your system, you may see the following dialog. Otherwise, proceed to step 4.



3. Select DeviceNet Configuration and click OK.

4. Click Online  on the toolbar.

The Browse for network dialog appears. You see the drivers you have configured on your system.



5. Select the 1747-SDNPT-3 driver and click OK.

   You receive the following prompt.

6. Click OK to upload the devices.

   RSNetWorx for DeviceNet software begins browsing for network devices.

ATTENTION

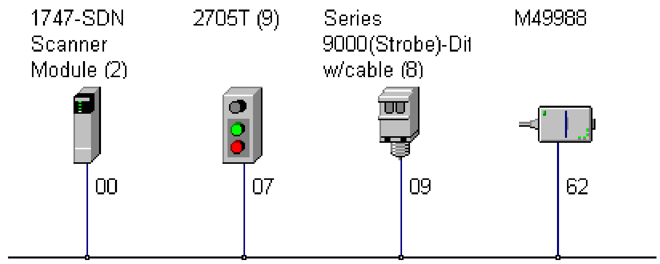
Performing a pass-through browse via the DH+ network takes longer than browsing by using the 1770-KFD DeviceNet driver as described in chapter 4.

Note that due to the time required, the pass-through method is not suitable for configuring a network nor for real time monitoring of your network devices.

The network should look similar to the one shown below when the software is finished browsing.

TIP

If RSNetWorx for DeviceNet software fails to find a device, check the physical connection to that device.



You are now online to the DeviceNet network via the Data Highway Plus network.

See pages 55…62 of this manual for examples of how to use RSNetWorx for DeviceNet software to adjust network parameters.

What's Next?

The next chapter describes how to create and run the example application program to test the DeviceNet network.

Creating and Running the Example Application Program

What This Chapter Contains

This chapter describes the procedure to create, download, and run an example ladder logic program to test the DeviceNet network. When the processor is put into Run mode, pressing the START button on the network's RediSTATION operator interface will cause the red indicator light to come on and stay on until the STOP button is pressed. Passing an object in front of the photoeye will increment a counter.

Recall that you cannot directly communicate with the SLC 500 processor over the DeviceNet network. This chapter shows how to download and run the program over a ControlNet, Ethernet, or Data Highway Plus network.

The SLC 500 I/O chassis used for these examples was set up with the following hardware.

Chassis Setup

Module	Slot	DH+ Address	Ethernet Address	ControlNet Address	DeviceNet Address
SLC 500 5/04 Processor[1]	0	1	-	-	-
SLC 500 5/05 Processor[2]	0	-	130.130.130.2	-	-
1747-KFC15	1	-	-	16	-
1747-SDN	2	-	-	-	0

(1) Used for Data Highway Plus and ControlNet examples.

(2) Used for Ethernet example.

The following table describes what this chapter contains and where to find specific information.

Install the Software

Follow these steps to install the software.

1. Insert the software CD-ROM installation disk in the drive.

   The CD-ROM supports Windows Autorun. If you have Autorun configured, the installation automatically starts when you insert the CD-ROM in your drive. If you do not have Autorun configured, perform steps 2 and 3.

2. From the Windows Start menu, choose Run.

3. Browse for the Setup program on the CD ROM and open it.

4. Follow the prompts that appear as you install the software.

Create the Example Application Program

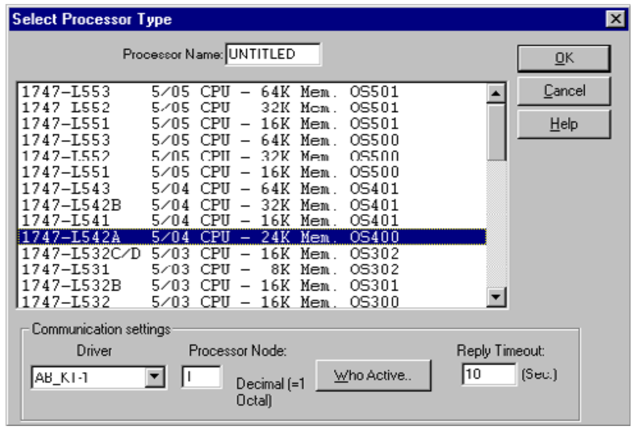
Follow these steps to create the example application program.

1. Start the RSLogix 500 programming software.

   You see the following dialog.

2. From the File menu, choose New.

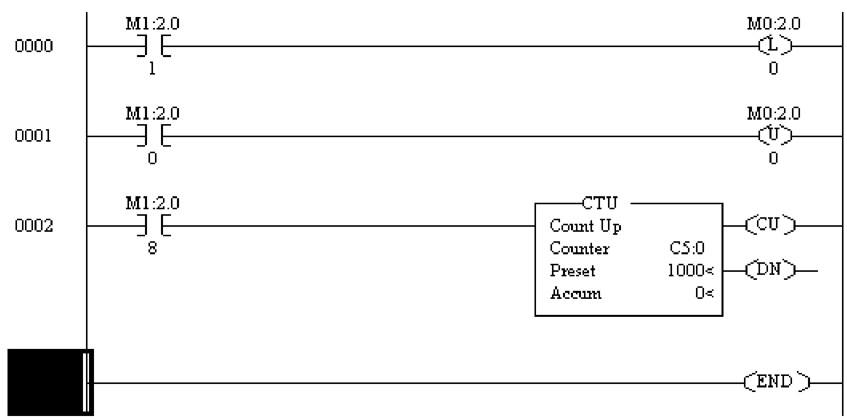
   The Select Processor Type dialog opens.



3. Select your processor type from the list (for example, 1747-L542A) and click OK.

4. Enter the following ladder program.

   Note that the program uses the addresses mapped to the devices by RSNetWorx for DeviceNet software in chapter 4.



5. Save the program by using an appropriate name, for example, 1747-SDN.

Download and Run the Program

You cannot go online to the processor directly over the DeviceNet network. The following sections provide examples of downloading and running the program by using these networks:

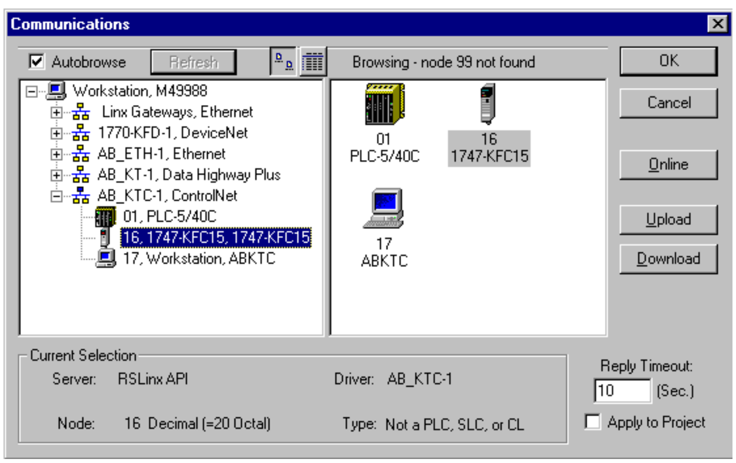
- ControlNet
- Ethernet
- Data Highway Plus

Download and Run the Program via a ControlNet Network

Follow these steps to download and run the example program via a ControlNet network.

1. From the RSLogix 500 Communication menu, choose System Comms.

   The Communication dialog appears. Your dialog may look different depending upon the drivers and other devices you have installed.



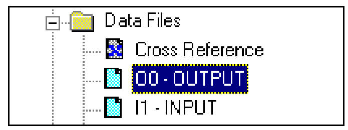
2. Click the + next to your ControlNet driver (AB_KTC-1) to expand the tree.

3. Select the 1747-KFC15 module as shown above and click Download.

   Your program is downloaded to the controller.

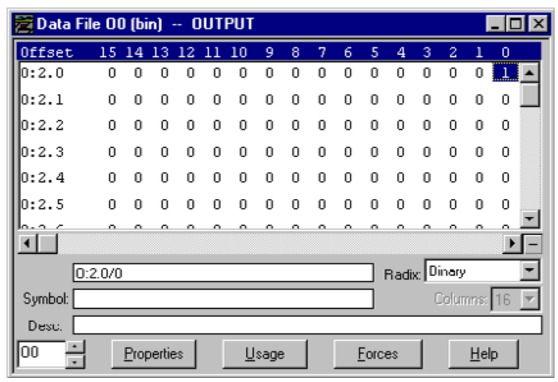
4. Change the SLC 500 processor mode to Run after the download is complete.

IMPORTANT	You must also place the 1747-SDN module in Run mode by setting the Run Bit (bit 0) in the scanner's command register. The command register is located at word 0 in the Output image table.



5. In the Data Files folder, double-click O0-Output.

   You see the 1747-SDN module's command register (file O:2.0 below).



6. Set bit 0 in the command register to 1, as shown above.

7. Press and release the START button on the RediSTATION operator interface.

   The red light should turn on. On your screen, you see rung 1 in your ladder program being energized as you press the button.

8. Pass your hand back and forth over the photoeye several times.

   On your screen you should see the counter increment.

9. Press and release the STOP button on the RediSTATION operator interface.

   The red light should turn off. On your screen, you see rung 2 in your ladder program being energized as you press the button.

This completes the ControlNet example.

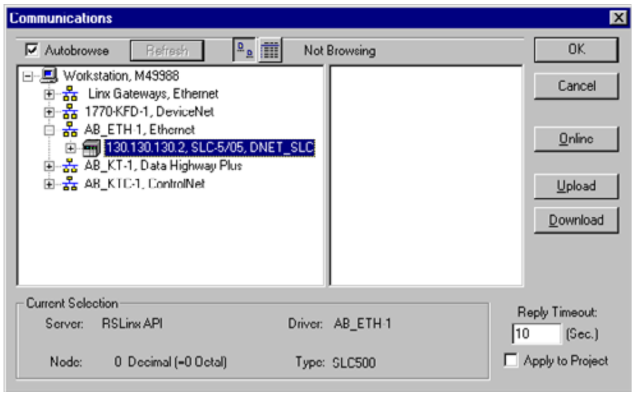
Download and Run the Program via an Ethernet Network

IMPORTANT	To communicate with your SLC 5/05 processor over an Ethernet network you must first configure the processor's Ethernet communication channel and assign it a unique IP address. See Appendix B for information on configuring the Ethernet communication channel.

Follow the procedure below to download and run the example program via an Ethernet network.

1. From the RSLogix 500 Communication menu, choose System Comms.

   The Communication dialog opens. Your dialog may look different depending upon the drivers you have installed.

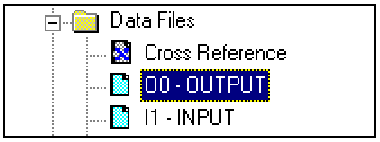


2. Click the + next to your Ethernet driver (AB-ETH-1 above) to expand the tree.

3. Select the SLC 500 processor as shown above and click Download.

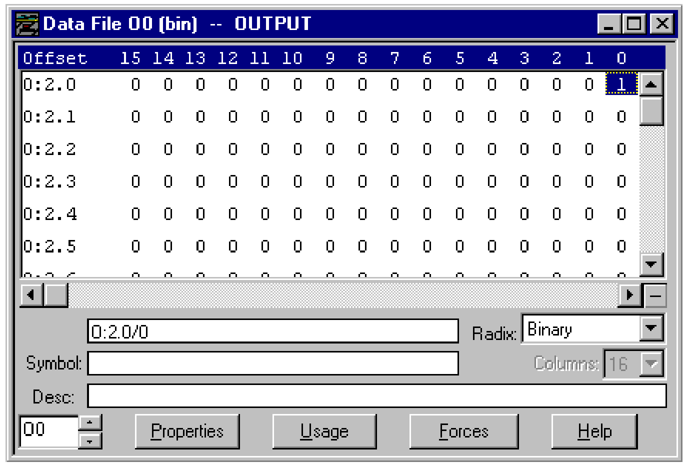
4. Put the processor in Run mode after the download is complete.

IMPORTANT	You must also place the 1747-SDN module in Run mode by setting the Run Bit (bit 0) in the scanner's command register. The command register is located at word 0 in the Output image table.



5. Double-click the O0 - OUTPUT file under the Data Files folder.

   You see the 1747-SDN module's command register (file O:2.0 below).



6. Set bit 0 in the command register to 1, as shown above.

7. Press and release the START button on the RediSTATION operator interface.

   The red light should turn on. On your screen, you see rung 1 in your ladder program being energized as you press the button.

8. Pass your hand back and forth over the photoeye several times.

   On your screen you should see the counter increment.

9. Press and release the STOP button on the RediSTATION operator interface.

   The red light should turn off. On your screen, you see rung 2 in your ladder program being energized as you press the button.

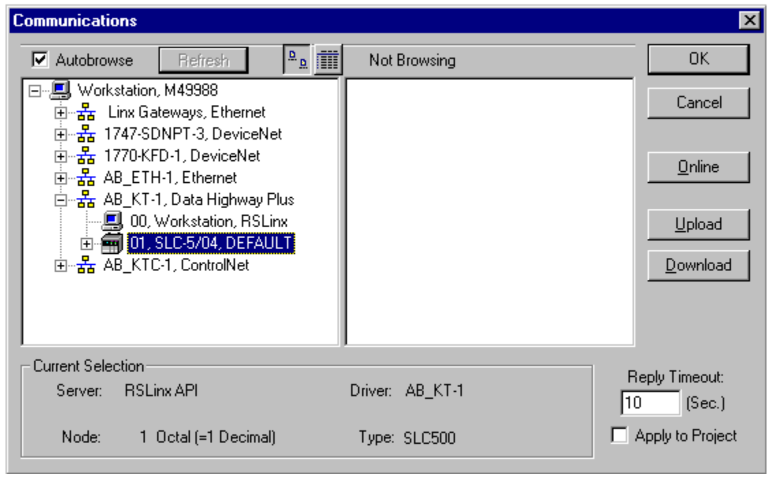
This completes the Ethernet example.

Download and Run the Program via a DH+ Network

Follow the procedure below to download and run the example program via a DH+ network.

1. From the RSLogix 500 Communication menu, choose System Communication.

   RSLinx software starts, and the Communication dialog opens.



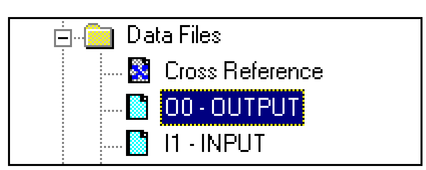
2. Select your Data Highway Plus driver (AB_KT-1, Data Highway Plus in the above example).

3. Select the + next to your Data Highway Plus driver (AB_KT-1 above) to expand the tree.

4. Select the SLC 500 processor and click Download.

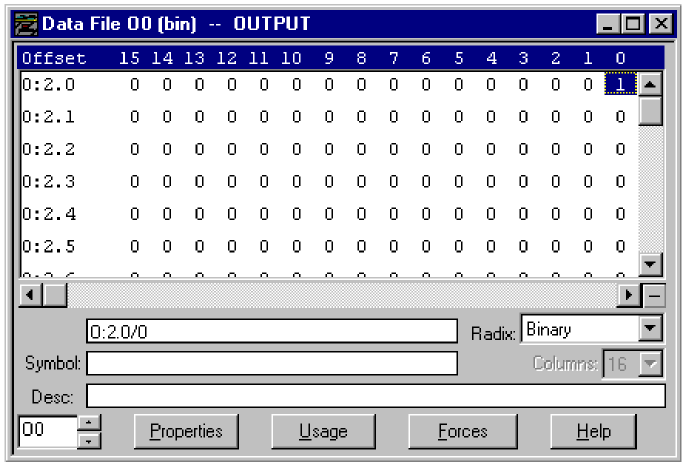
5. Go Online and put the SLC 500 processor in Run mode after the download is complete.

IMPORTANT You must also place the 1747-SDN module in Run mode by setting the Run bit (bit 0) in the scanner's command register. The command register is located at word 0 in the Output image table.



6. Double-click the O0 - OUTPUT file under the Data Files folder.

   You see the 1747-SDN module’s command register (file O:2.0 below).



7. Set bit 0 in the command register to 1.

8. Press and release the START button on the RediSTATION operator interface.

   The red indicator light should turn on. On your screen, you should see rung 0 in your ladder program being energized when you press the button.

9. Pass your hand back and forth over the photoeye several times.

   On your screen you should see the counter incrementing.

10. Press and release the STOP button on the RediSTATION operator interface.

    The red indicator light should turn off. On your screen, you should see rung 1 in your ladder program being energized when you press the button.

This completes the Data Highway Plus example.

What’s Next?

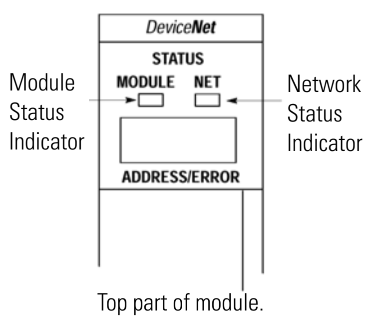
This concludes the application examples. The following chapter describes how the diagnostic indicators on the 1747-SDN module can be used for troubleshooting.

Troubleshooting

The 1747-SDN interface module is provided with status diagnostic indicators on its front panel. The diagnostics provided by these indicators are described in this chapter.

Module Status Indicator

The bicolor (green/red) Module Status indicator (MODULE) indicates whether the 1747-SDN module has power and is functioning properly.



If the MODULE indicator is	Then	Take this action
Off	There is no power applied to the module.	Verify power connections and apply power.
Green	The module is operating normally.	No action required.
Flashing Green	The module is not configured.	Configure the module.
Flashing Red	There is an invalid configuration.	Check configuration setup.
Red	The module has an unrecoverable fault.	Replace the module.

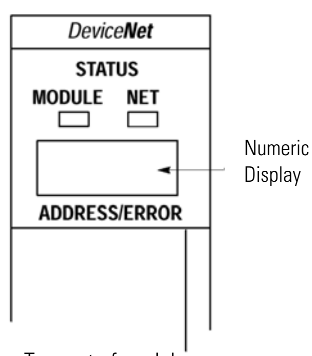
Network Status Indicator

The bicolor (green/red) network status indicator (NET) provides troubleshooting information about the DeviceNet channel communication link.

If the NET indicator is	Then	Which indicates	Take this action
Off	The device has no power or the channel is disabled for communication due to bus off condition, loss of network power, or has been intentionally disabled.	The channel is disabled for DeviceNet communication.	Power-up the module, provide network power to the channel, and be sure the channel is enabled in both the module configuration table and the module command word.
Green	Normal operation.	All slave devices in the scanlist table are communicating normally with the module.	None.
Flashing Green	The two-digit numeric display for the channel indicates an error code that provides more information about the condition of the channel.	The channel is enabled but no communication is occurring.	Configure the scanlist table for the channel to add devices.
Flashing Red	The two-digit numeric display for the channel displays an error code that provides more information about the condition of the channel.	At least one of the slave devices in the module's scanlist table has failed to communicate with the module.	Examine the failed device and the scanlist table for accuracy.
Red	The communication channel has failed. The two digit numeric display for the channel displays an error code that provides more information about the condition of the channel.	The module may be defective.	Reset module. If failures continue, replace module.

Numeric Display Code Summary

The 1747-SDN module uses numeric codes to display diagnostic information about its status. The display flashes at one second intervals. The following table summarizes the meanings of the numeric codes.



Top part of module.

Numeric Display

Numeric Code	Description	Corrective Action
0…63	Normal operation. The numeric display indicates the 1747-SDN module's node address on the DeviceNet network.	None.
65	Normal operation when AutoScan is enabled.	None.
70	Module failed Duplicate Node Address check.	Change the module channel address to another available one. The node address you selected is already in use on that channel.
71	Illegal data in scanlist table (node number alternately flashes).	Reconfigure the scanlist table and remove any illegal data.
72	Slave device stopped communicating (node number alternately flashes).	Inspect the field devices and verify connections.
73	Device's identity information does not match electronic key in scanlist table entry.	Verify that the correct device is at this node number. Make sure that the device at the scrolling node address matches the desired electronic key (vendor, product code, product type).
74	Data overrun on port detected.	Modify your configuration and check for invalid data. Check network communication traffic.
75	No traffic from other modules detected on the network.	Check the network configuration. (Scanlist may be empty.)
76	No direct network traffic for module detected.	None. The module hears other network communication.
77	Data size expected by the device does not match scanlist entry.	Reconfigure your module for the correct transmit and receive data sizes.
78	Slave device in scanlist table does not exist.	Add the device to the network, or delete the scanlist entry for that device.
79	Module has failed to transmit a message.	Make sure that your module is connected to a valid network. Check for disconnected cables.
80	Module is in Idle mode.	Put controller in Run mode. Enable Run bit in module command register.
81	Module is in Fault mode.	Check Module Command Register for fault bit set.
82	Error detected in sequence of fragmented I/O messages from device.	Check scanlist table entry for slave device to make sure that input and output data lengths are correct. Check slave device configuration.
83	Slave device is returning error responses when module attempts to communicate with it.	Check accuracy of scanlist table entry. Check slave device configuration. Slave device may be in another master's scanlist. Reboot slave device.
84	Module is initializing the DeviceNet network.	None. This code clears itself once module attempts to initialize all slave devices on the network.
85	Data size was incorrect for this device at runtime.	Slave device is transmitting incorrect length data. Verify device is not configured for variable poll connection size. Try replacing the device.

Numeric Display

Numeric Code	Description	Corrective Action
86	Device is producing zero length data (idle state) while module is in Run mode.	Check device configuration and slave node status.
87	The primary owner has not allocated the slave.	Put the primary owner online.
88	The connection choices (polled, strobed) between the primary connection and the shared input only connection do not match.	Reconfigure the shared input-only connection's choices to be the same as, or a subset of, the primary connection's choices.
89	Slave device initialization using Auto Device Replacement parameters failed.	Put the slave device into configurable mode. Check the slave's EDS file, if the slave is configured offline. Check to see if the slave device has been replaced with an incompatible device.
90	User has disabled communication port.	Check Module Command Register for DISABLE bit set.
91	Bus-off condition detected on comm port. Module is detecting communication errors.	Check DeviceNet connections and physical media integrity. Check system for failed slave devices or other possible sources of network interference.
92	No network power detected on communication port.	Provide network power. Make sure that module drop cable is providing network power to module comm port.
95	Application FLASH update in progress.	None. Do not disconnect the module while application FLASH is in progress. You will lose any existing data in the module’s memory.
97	Module operation halted by user command.	Check Module Command Register for HALT bit set.
98	Unrecoverable firmware failure.	Service or replace your module.
99	Unrecoverable hardware failure.	Service or replace your module.
E2	RAM test failure.	Service or replace your module.
E4	Lost power during FLASH upgrade.	Service or replace your module.
E5	No boot or main code.	Service or replace your module.
E9	Module memory has been flushed for factory default settings.	Cycle module power to recover.

DeviceNet Explicit Messaging

DeviceNet Explicit Message Instruction Overview

The DeviceNet Explicit Message (DEM) instruction allows generic Common Industrial Protocol (CIP) commands to be initiated to devices, such as drives, communicating on DeviceNet networks. This instruction requires RSLogix 500 software, version 7.10 or later, for programming.

The DEM instruction utilizes the explicit message capability built into the 1747-SDN DeviceNet scanner module.

While not adding any additional capability over what already exists in the scanner module, the DEM instruction greatly simplifies the programming, configuration, monitoring, and troubleshooting of explicit messages on the DeviceNet network.

Unlike I/O configured in the scanner module's scanlist, which is updated on a continual basis, explicit messages allow data to be sent and received on an as-needed basis, minimizing network traffic. For instance, you may only want to write configuration parameters to a drive once at machine start-up time.

The DEM instruction can be used with any SLC 5/03, SLC 5/04, or SLC 5/05 processor that is at OS firmware level Series C, FRN 10 or later.

The DEM instruction uses an integer control block for storing the instruction parameters and a configuration setup screen, similar to the MSG instruction. The CIP commands consist of a Service Code; the object Class, Instance, and Attribute; and Send and Receive Data (if required for the selected Service Code). The setup dialog provides a list of standard CIP Services to select from, including:

- Read Assembly.
- Write Assembly.
- Read Parameter.
- Write Parameter.
- Generic Get Attribute Single.
- Generic Set Attribute Single.

In addition, a Custom setting lets you enter any Service Code. Send data and receive data are stored in separate data table files.

DeviceNet Explicit Message (DEM)

This is an output instruction that lets you initiate unconnected CIP Generic messages via a 1747-SDN module installed in the local chassis. These messages can be initiated to any node on the same DeviceNet network as the 1747-SDN module, as long as the node is in the scanner's scanlist. The scanner can be in either Idle mode or Run mode. Each module can process only one DEM instruction at a time. The instruction is similar in operation to a standard MSG instruction.

DEM Instruction Parameters

Enter the following parameters when programming this instruction:

- Control Block is an integer file address that you select. It is a block of words, containing the status bits and other data associated with the DEM instruction. It also contains the Send and Receive data.
- Control Block Length is a display-only field that indicates how many integer file words are being used by the control block. For the DEM instruction, the length is always 70 words.

DEM Instruction Setup Screen Parameters

The following sections provide parameters for the DEM instruction setup screens.

Parameters for This Controller on the General Tab

- 1747-SDN Slot

  This pull-down menu lists all of the local slots that contain DeviceNet scanner (1747-SDN) modules within the I/O Configuration. Select the slot number of the particular module that this explicit message will be initiated through.

- Size in Send Data (Bytes)

  This field defines how many bytes of data are sent along with this explicit message command. If unsure of how much data will be sent, you may select up to the maximum size of 52 bytes when defining the instruction, and then reduce the size later based on experience.

Parameters for Target Device on the General Tab

- Message Timeout(x1 sec)

  The amount of time in seconds that the processor will wait for a reply from the scanner to the explicit message command. Range is 0, 2…255. Like the Message Timeout in a standard MSG instruction, a value of 0 disables the Message Timeout and a value of 1 second gets bumped to 2 seconds upon instruction execution. If the built-in Message Timeout is disabled, user-programmed timeout logic must be included to avoid a message lockup in the case where a reply is lost.

- DeviceNet Addr (dec)

  The target DeviceNet node address. Valid range is 0…63. If you enter in the local scanner's DeviceNet node address, the command is executed by the local scanner.

- Service

  This pull-down menu lets you select services based on name rather than Service Code. The Custom service lets you enter in any Service Code in the hexadecimal range of 1…7F. The services listed in the pull-down menu are:

  – Read Assembly.
  – Write Assembly.
  – Write Output Point.
  – Read Output Point.
  – Read Input Point.
  – Read Parameter.
  – Write Parameter.
  – Read Analog Input.
  – Write Analog Output.
  – Generic Get Attribute Single.
  – Generic Set Attribute Single.
  – Reset Identity Object.
  – Custom.

- Service Code (hex)

  This field is read-only unless the Custom Service is selected. Possible Service Codes are 1…7F (hex).

  See Volume 1 of the CIP Common Specification, Appendix A, for the list of valid explicit messaging Service Codes.

- Class (hex)/(dec)

  Possible Classes are 0…FF (hex).

  See Volume 1 of the CIP Common Specification for the list of defined Classes.

  You may either enter in a hexadecimal Class value in the (hex) field or a decimal Class value in the (dec) field.

- Instance (hex)/(dec)

  Possible Instances are 0…FFFF (hex).

  See Volume 1 of the CIP Common Specification for the list of valid Instances for each Class.

  You may either enter in a hexadecimal Instance value in the (hex) field or a decimal Instance value in the (dec) field.

- Attribute (hex)/(dec)

  Possible Attributes are 0…FF (hex).

  See Volume 1 of the CIP Common Specification for the list of valid Attributes for each Class.

  You may either enter in a hexadecimal Attribute value in the (hex) field or a decimal Attribute value in the (dec) field. Note that if the value of the Attribute is set to 0, the DeviceNet scanner does not transmit an Attribute byte. Some Service Codes require that a zero-value Attribute byte be transmitted. For these Service Codes, the first byte of the Send Data must be set to zero and the Size of Send Data (Bytes) must include this additional byte.

Definitions for Message Status Bits on the General Tab

The table below lists the various status bits associated with the DEM instruction as displayed in the DEM instruction setup screen.

DEM Instruction Setup Screen Status Bits

Bit Definition	Bit Mnemonic	Bit Address
Abort	AB	08
Error	ER	12

DEM Instruction Setup Screen Status Bits

Bit Definition	Bit Mnemonic	Bit Address
Done	DN	13
Enabled	EN	15
Waiting for slot	WS	10

- Abort bit AB (word 0, bit 8) lets you abort an executing DEM instruction by setting this bit. The ER bit will be set as soon as the AB bit is set. This bit is reset the next time the message rung goes from false to true.
- Error bit ER (word 0, bit 12) is set when the message has failed to complete successfully. This bit is reset the next time the message rung goes from false to true. Do not set or reset this bit. It is informational only.
- Done bit DN (word 0, bit 13) is set when the message has completed successfully. This bit is reset the next time the message rung goes from false to true. Do not set or reset this bit. It is informational only.
- Enabled bit EN (word 0, bit 15) is set after the message rung goes from false to true AND the module accepts this message because it is not currently processing any other explicit messages. (The module can process only one DEM instruction at a time.) If the message rung goes false before the module accepts this message, then the enable bit will remain off and the message will not be executed. This bit is reset when the message has completed with either the Done bit set or the Error bit set and the message rung goes false. If the message rung conditions remain true, you may retrigger the message instruction by resetting this bit after either the ER or DN bit has been set, indicating that the previous execution has completed.
- Waiting for Slot bit WS (word 0, bit 10) is set when the message rung goes from false to true, but the module is still processing another DEM instruction. To be sure that this message gets processed, you must leave the message rung conditions true until the WS bit is reset and the EN bit is set, indicating that the module has accepted this message for processing. Do not set or reset this bit. It is informational only.

Scanner Status, Error, and Error Description on the General Tab

The error code displays the explicit message status returned by the module. An error code of 01h means Transaction completed successfully.

DEM Instruction Scanner Codes

Scanner Code	Description of Scanner Status
2H	Transaction in progress.
3H	Slave not in scanlist.
4H	Slave offline.
5H	DeviceNet port disabled/offline.
6H	Transaction TXID unknown.
7H	Slave not responding to explicit request.
8H	Invalid command code.
9H	Scanner out of buffers.
10H	Another transaction in progress.
11H	Could not connect to slave device.
12H	Response data too large for block.
13H	Invalid port.
14H	Invalid size specified.
15H	Connection busy.

All error codes listed above result in an error code of 2.

Complete List of Valid DEM Error Codes

Error Code	Description of Error Condition
0	No error.
1	Timeout error. DeviceNet explicit message timed out by processor.
2	Scanner error. See Scanner Status.
3	User error. DeviceNet explicit message aborted by user.

For error code 4, the error description displays the CIP response error code and description as documented in the CIP Response Error Codes table.

CIP Response Error Codes

Numeric Code	Name	Description
02H	Resource unavailable	A needed resource was not available
08H	Service unsupported	Service is not defined or implemented for this class/instance
09H	Invalid attribute value	Data is invalid for the specified attribute
0BH	Already in requested state	Object is in the requested state - redundant request
0CH	Object state conflict	Not allowed with object in present state
0EH	Attribute cannot be set	Read-only attribute
0FH	Privilege violation	A permission/privilege check failed
10H	Device state conflict	Not allowed with device in present state
11H	Reply too big	Reply larger than buffer allocated when connection was established
13H	Too little data	Request included insufficient data
14H	Attribute not supported	Attribute number is incorrect
15H	Too much data	Request included extra data
16H	Object does not exist	Class/instance numbers are incorrect
18H	No stored attribute data	Attribute data was not saved prior to this request
19H	Store operation failure	Attribute data was not successfully saved
1FH	Vendor-specific error	Second byte may offer details - refer to vendor documentation
20H	Invalid parameter	Parameter associated with request is invalid
D0H FFH	Reserved and service-specific errors	Used only when none of the standard error codes supplemented by the second byte accurately describes the problem

Any time the error code is non-zero, the DEM error (ER) bit is set.

Send Data Tab

The Send Data Tab provides a convenient way of viewing and entering in data to be sent along with the explicit message command. The data is shown in byte format with a selectable radix of either decimal or hex/BCD. The display shows only the number of words that are defined in the Size of Send Data (Bytes) field, starting with the low byte of the first word. If the Size of Send Data is zero, then no data is displayed. You can also change the data being viewed, but only when offline or during an online edit. Click on the data and enter in a byte value based on the current radix (0…255 for decimal and 0 to FF for hex/BCD). The changed data gets copied to the DEM control block when the rung is accepted. To update the Send Data display with the current values stored in the DEM control block, click Refresh.

Receive Data Tab

The Receive Data Tab provides a convenient way of viewing the data that is returned by the target device in response to the explicit message command sent. The data is shown in byte format with a selectable radix of either decimal or hex/BCD. The display shows 58 bytes of receive data, starting with the low byte of the first word. To update the Receive Data display with the current values stored in the DEM control block, click Refresh.

Control Block Layout

The control block layout is shown below.

SLC 5/0*x* DeviceNet Explicit Message (DEM) Control Block Structure

	15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**0**
Word 0	EN		DN	ER		WS		AB	Error Code							
Word 1								IMR	Message Timeout Preset							
Word 2	Message Timeout Accumulator															
Word 3	Message Timer Scaled Zero															
Word 4	Unused															
Word 5	Unused															
Word 6	Transaction ID								**Scanner Command**							
Word 7	Scanner Port (always 0x00)								Size of Send Data (in bytes)							
Word 8	Service Code								**Target MAC ID**							
Word 9	Reserved (not transmitted)								**Class**							
Word 10	Instance															
Word 11	Reserved (not transmitted)								**Attribute (optional)**							
Word 12	Transaction Send Data															
…	(next 26 words)															
Word 37																
Word 38	Transaction ID								**Scanner Status**							
Word 39	Scanner Port (always 0x00)								**Size of Reply Data (in bytes)**							
Word 40	Reply Service Code								**Target MAC ID**							
Word 41	Transaction Reply/Receive Data															
…	(next 29 words)															
Word 69																

AutoScan

Overview

The DeviceNet network AutoScan feature enables a scanner to automatically map a network of slave devices into its scanlist without the use of RSNetWorx for DeviceNet software. This greatly improves the ease of setting up a DeviceNet network, especially networks comprised of simple devices.

Basically, when the feature is enabled the 1747-SDN module searches for devices on the network that are not yet mapped. Once a qualifying slave device is found, it is added to the scanner's scanlist and its I/O data is mapped into a predefined location in the scanner's I/O memory table based on the device's node address.

Implementing AutoScan

This feature requires a series C, 1747-SDN module.

The steps in this section describe how to set up the feature and how it operates. Notice that explicit messaging is used for some of the steps below. There are several ways that an explicit message can be sent on the DeviceNet network.

- A user ladder program using the DEM instruction
- External programming/configuration device (catalog number 193-DNCT)
- RSNetWorx for DeviceNet software

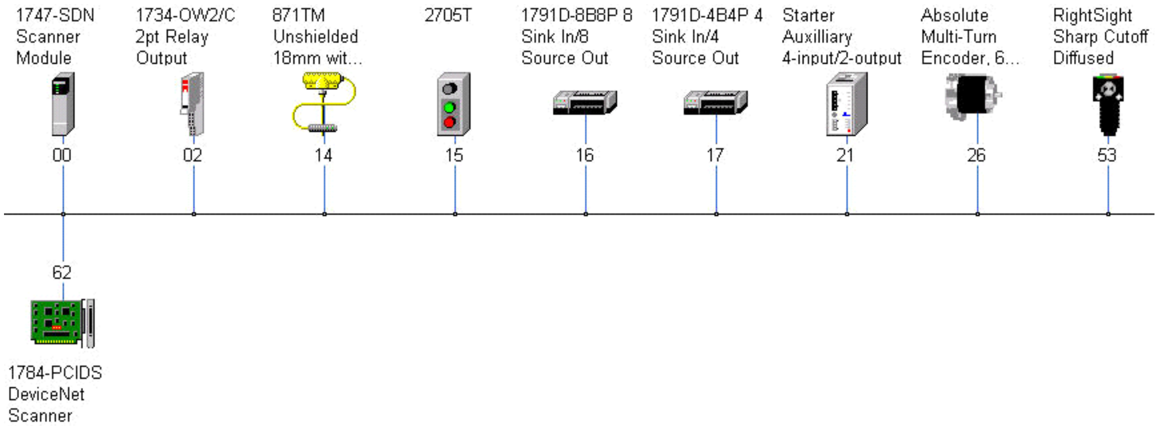
Since the purpose of the feature is to eliminate the use of RSNetWorx software, instructions on how to send an explicit message via the class instance editor in RSNetWorx for DeviceNet software is not covered in this section.

For more information on using Explicit Messaging see Chapter 8.

First you need to set up the physical network.

Make sure all devices are addressed appropriately (no address conflicts) and are communicating at the same communication rate.

The diagram below shows an example system using the 1747-SDN module.



The node addresses can be commissioned via hardware switches on the device or through other DeviceNet configurators such as the 193-DNCT DeviceNet Configuration terminal.

Refer to the DeviceNet Programming Terminal user manual, publication 193-UM009, for more information on how to set up node addresses.

Enabling AutoScan via Ladder Logic

To implement AutoScan via ladder logic, use the DeviceNet Explicit Message (DEM) instruction. The DEM instruction requires 1747 operating system series C, FRN 10 or later, and RSLogix 500 software version 7.10 or later. Use the DEM instruction to initiate multiple CIP generic messages to the scanner module.

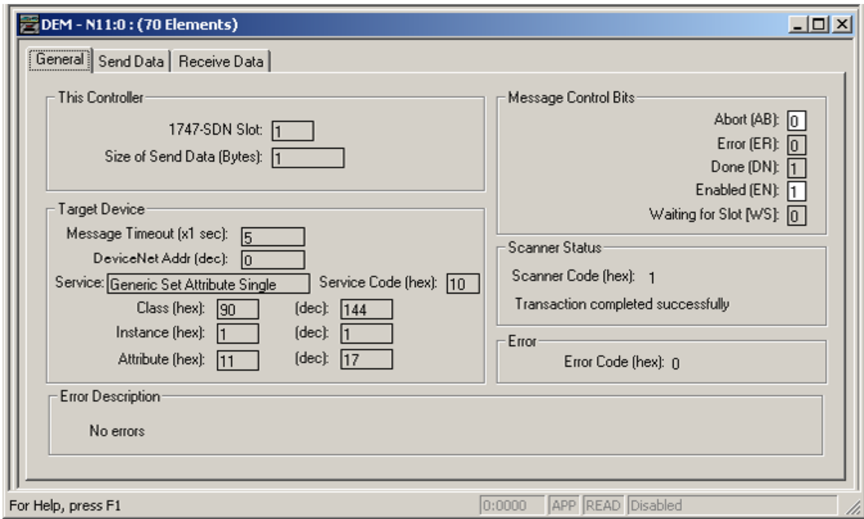
See Chapter 8 for more information on the DEM instruction.

1. Verify that the scanner is in Idle mode (bit 0 in the scanner control output word is 0) and that all slave nodes are connected and communicating on the DeviceNet network.

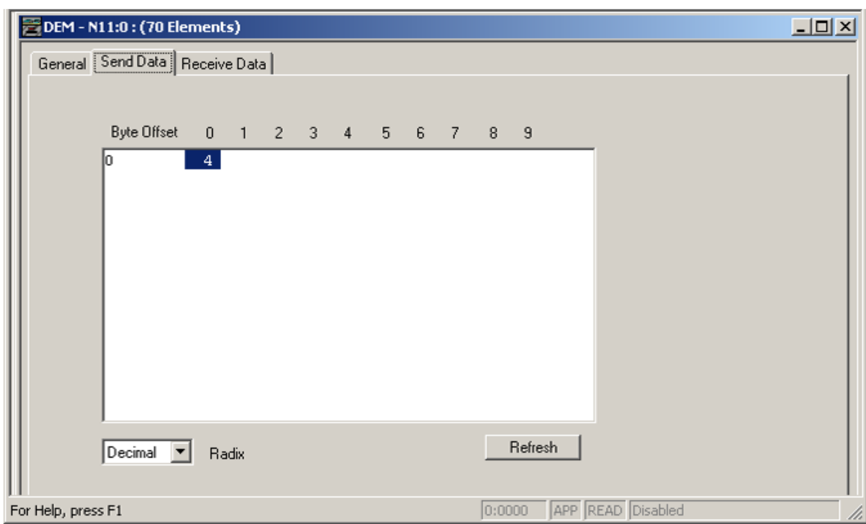
   Two DEM instructions are required and one is optional. The default AutoScan setting allocates a 4-byte entry in both the input and output memory maps in the scanner for each slave device detected on the network. If this is adequate for the applications, skip the first DEM instruction.

However, for applications where you would like to customize the I/O allocation size, configure a DEM instruction to adjust the 4-byte allocation.

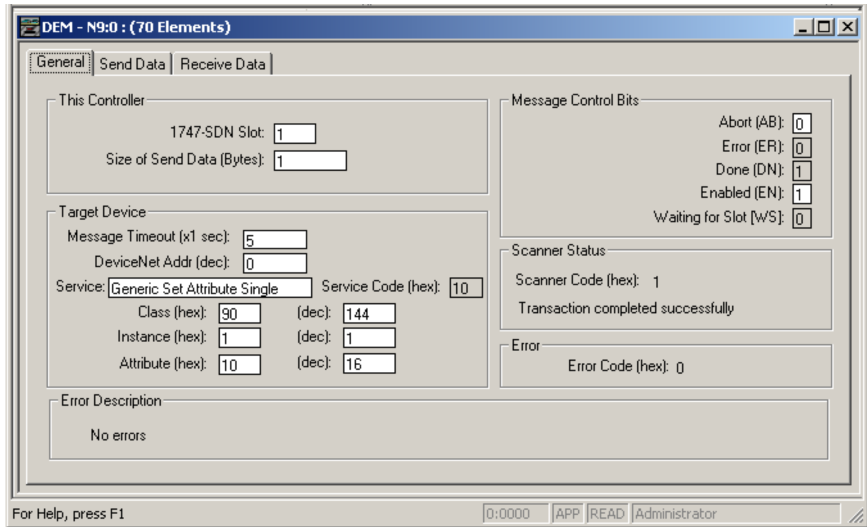
2. Fill in the fields as shown, make sure to match the 1747-SDN module's Slot and DeviceNet Address to your particular scanner module.



3. On the Send Data tab, enter the I/O allocation size, in bytes (1…32).

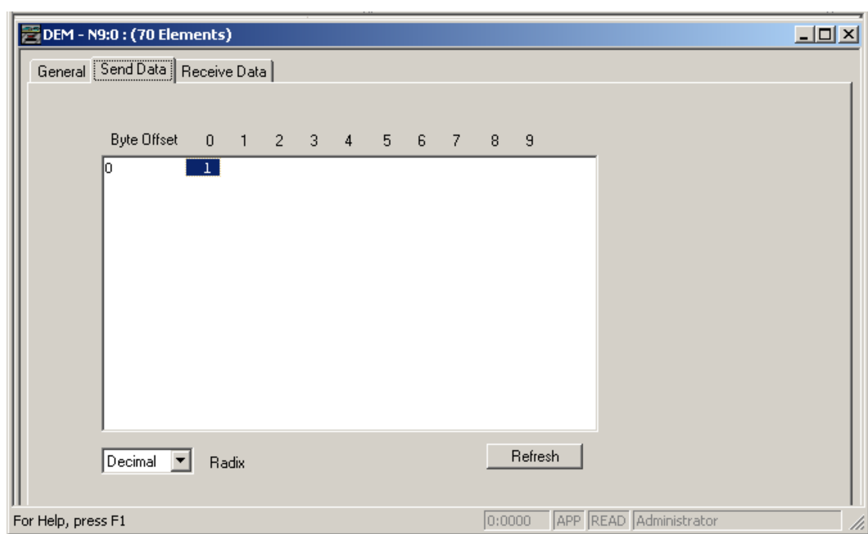


4. On the DEM instruction setup screen to initiate AutoScan, fill in the fields as follows.

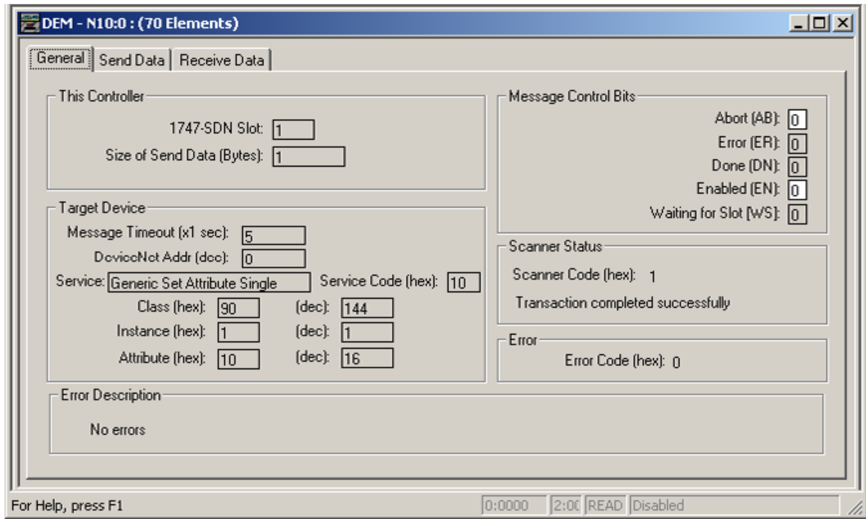


TIP	Make sure to match the 1747-SDN Slot and DeviceNet Address to your particular scanner module. All other fields should appear exactly as shown above.

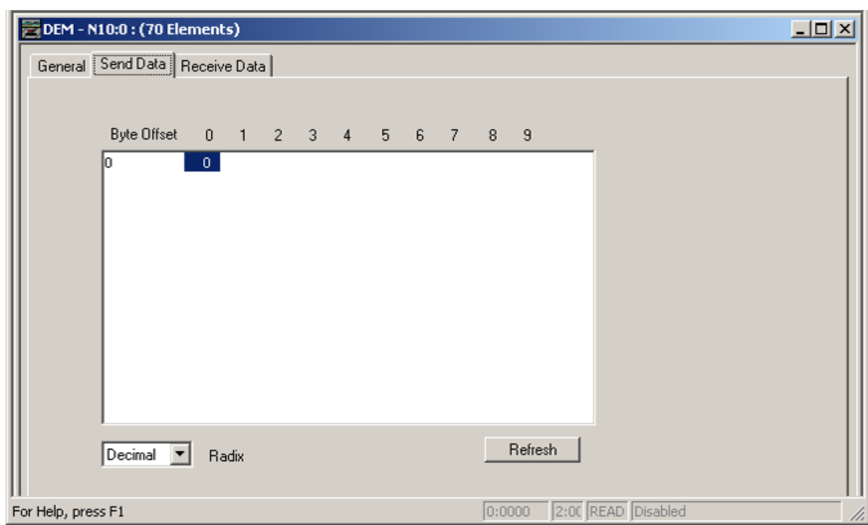
5. On the Send Data tab, enter a 1 to enable AutoScan.



6. Fill in the DEM instruction Setup dialog to disable AutoScan.



TIP The only difference between the Enable AutoScan and Disable AutoScan DEM instructions is that a 0 is sent to Disable AutoScan.



7. Download the program and then trigger each DEM instruction individually,

8. Verify that the scanner status display has stabilized prior to executing the next DEM instruction.

9. Return the scanner to Run mode (bit 0 in the scanner control output word is set to 1) and the scanner status display should be displaying the scanner node address.

   If the status display is flashing other codes as well, refer Chapter 7 for troubleshooting.

Enabling AutoScan via the DeviceNet Configuration Terminal

Follow these steps to enable the AutoScan feature using the DeviceNet configuration terminal (DNCT).

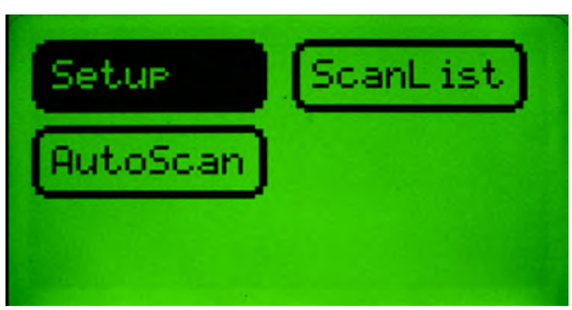
1. Verify that the scanner is in Idle mode (bit 0 in the scanner control output word is 0) and that all slave nodes are connected and communicating on the DeviceNet network.

2. Plug the DNCT terminal into the DeviceNet network.

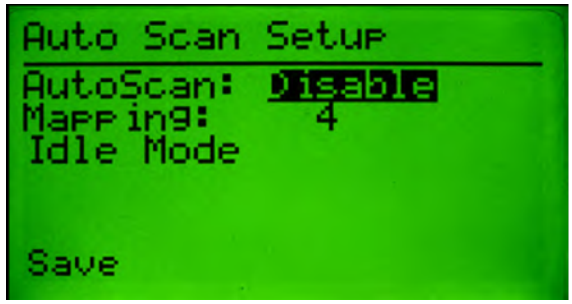
3. Select 1747-SDN module from the Network Who screen by pressing the E/Enter key.

4. Use the D/Down-arrow key to highlight Scanner, then select it by pressing the E/Enter key.

5. Use the D/Down-arrow key to highlight AutoScan, then select it by pressing the E/Enter key.



6. On the Auto Scan Setup screen, change AutoScan from Disable to Enable by pressing either the C/Up-arrow key or the D/Down-arrow key.



7. Press the B/SEL key to scroll down to Mapping.

8. Type in the number of bytes that you want to automap to each DeviceNet node (1…32).

   The default is 4 bytes.

9. Press the B/SEL key to scroll down to Save.

10. Press the E/Enter key to save your selections.

11. Notice that the 1747-SDN module status indicator flashes red then back to green, while the status display momentarily shows 72.

    When the AutoScanning is complete, the status display blinks between 65 and the node address of the scanner.

12. On the Auto Scan Setup screen, change AutoScan from Enable to Disable by pressing either the C/Up-arrow key or the D/Down-arrow key.

13. Press the B/SEL key to scroll down to Save.

14. Press the E/Enter key to save your selection.

    If all the slaves are properly mapped, the scanner status display blinks between 80 and the node address of the scanner.

15. Put the SLC controller into Run mode and verify that the scanner Run bit is set (bit 0 in the scanner control output word is 1).

    The scanner status should display only the node address of the scanner. If it is flashing other codes as well, refer to the scanner module user manual for troubleshooting.

How AutoScan Operates

AutoScan is active when the feature is enabled and the scanner is in Idle mode. When active, the scanner attempts to connect to each device not enabled in the scanlist. The scanner only checks for devices with node addresses between 0…61, inclusive. The connections to these devices are made on a round robin basis.

When a device is found, the scanner gets the Produced and Consumed data sizes from the slave devices's Connection Object instances. If either the Produced and Consumed data sizes are greater than the configured fixed data mapping size (per configured in step 2), then the node is rejected and not entered into the scanlist. For qualifying nodes, the scanner would enter the device into the scanlist and attempt to allocate an I/O connection using one of the following communication format choices (in this particular order).

Format	EPR
COS	250 ms
Poll	75 ms
Strobe	75 ms
Cyclic	500 ms

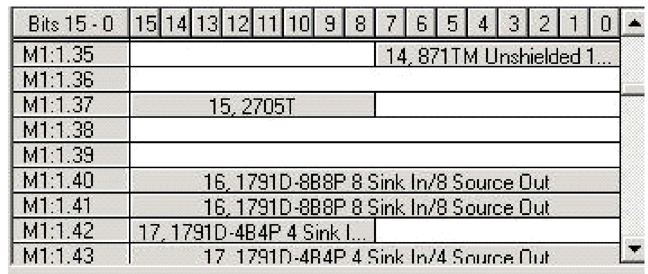
EXAMPLE If a photoeye was connected on a network that only supported strobed connections, the scanner does a couple of things. First, it recognizes that a device exists for which memory was available for the node number with the configured allocation size on a network that was not currently mapped. Then, the scanner would attempt to initiate both COS and polled connections first but the strobed connection would be selected as that is the only connection that the photoeye supported.

The input and output data is mapped into the scanner's I/O data table based on the device's node address and the configured fixed mapping size. The formula for calculating the byte offset is:

M file Input (Output) [byte] Offset = ((Node Address) x (I/O Allocation Size))

EXAMPLE

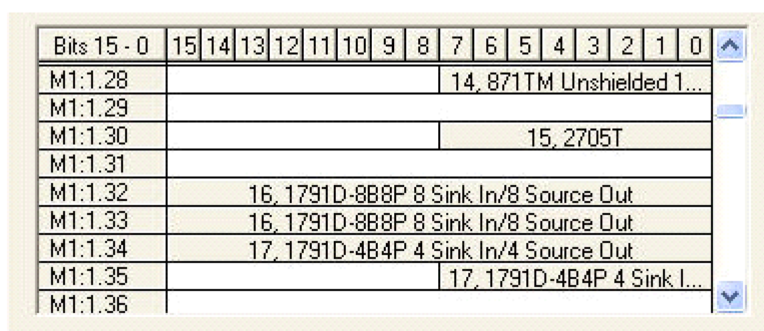
When using five bytes for each device, the input data for devices are allocated in the 1747-SDN module's input table. Note that M files are used because discrete mapping is not used with the AutoScan feature. In this case node 15 has an offset of 15*5=75 byte, therefore node 15 data is located in the data map at word 75/2 = 37.5 of the M file with the non-integer result the offset starts at the upper byte of the word.



For even I/O allocation sizes, the formula for calculating the Input or Output data offset since there are two bytes in each word is:

M file Input (Output) [Word] Offset = ((Node Address) x (I/O Allocation Size / 2))

When using the default fixed mapping size of 4 bytes or 2 words, the input data for devices are allocated in the 1747-SDN module's input table. Note that M files are used because discrete mapping is not used with AutoScan. In this case node 14 has an offset of 14*2=28 word therefore node 14 data is located in the data map at word 28 of the M file.



Other Important Information about AutoScan

The factory default setting for AutoScan is disabled for all products.

Make sure that input or output data memory size in the scanner is large enough to accommodate the size required based on the number of nodes on the network and the AutoScan I/O allocation size per node.

TIP

Nodes 0…61 inclusive are scanned and added to the scanlist if they are not already mapped and size exists. For maximum capacity for slave devices the scanner node address could be configured as node 62; node 63 could also be used but to avoid duplicate node address issues with new devices this is not suggested.

EXAMPLE

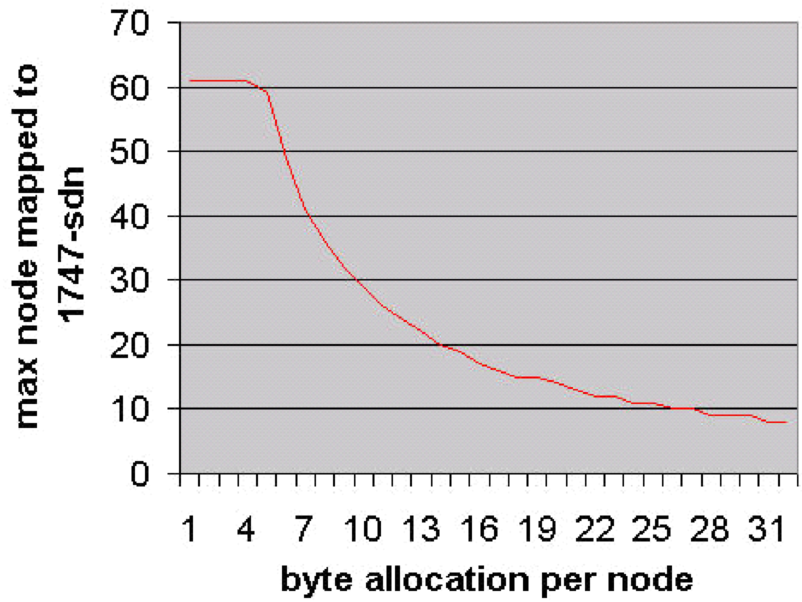
If the I/O allocation size per node is configured for 16 bytes and there are 32 slave devices on the network (node addresses 1…32), AutoScan would require 16 bytes x 32 = 512 bytes (256 words) of I/O space in both the scanner's input and output table. Assuming it is an SLC system, the maximum scanner input data table size is 150 words for input and output. The required space exceeds what the 1747-SDN module can support. The user would need to adjust the I/O allocation size or reduce the slave device count on the network to include all of the devices in the scanlist. Devices outside of the scanner's allowable I/O image space will be rejected and will not be included in the scanlist.

Scanlist

Data Allocation (bytes)	Max Node	Data Allocation (bytes)	Max Node
1	61	17	16
2	61	18	15
3	61	19	15
4	61	20	14
5	59	21	13
6	49	22	12
7	41	23	12
8	36	24	11
9	32	25	11
10	29	26	10

Scanlist

Data Allocation (bytes)	Max Node	Data Allocation (bytes)	Max Node
11	26	27	10
12	24	28	9
13	22	29	9
14	20	30	9
15	19	31	8
16	17	32	8



The AutoScan feature will automatically be disabled in the scanner as soon as a scanner property is modified by RSNetWorx for DeviceNet software. For example, any manual changes to the scanlist using RSNetWorx software will disable the AutoScan feature in the scanner.

One new status code has been added to the Node Status list. This code is presented in the Node Status table.

Node Status

Status Code (decimal)	Description of Status
65	AutoScan Active (Scanner only status)

When the scanner is in Run mode with AutoScan enabled, the scanner display alternates between 65 and the scanner node address.

When a scanner is transitioned from Run mode to Idle mode while AutoScan is enabled, it only scans the network for nodes that are not already in the scanlist. However, while in Idle mode, an AutoScan Disable mode to Enable mode transition will cause the scanner to erase the existing scanlist and scan for all nodes on the network.

The AutoScan feature enables Auto-address Recovery (AAR) for each of the configured slave devices.

The AutoScan feature checks for the Quick Connect setting in each slave device and enables Quick Connect in the scanner if it is enabled in the slave devices.

Additionally, AutoScan changes the default setting of InterScan Delay (ISD) from 10 ms to 4 ms. The reason for this is to improve network performance. The smaller value of ISD limits the access that software has for communication on the network, but when using AutoScan software communication should not be a concern.

Data Map Example

What This Appendix Contains

This appendix illustrates a basic mapping example that connects a DeviceNet network to 62 simple sensor-type devices. Each device sends one data byte that contains one data bit and one status bit. These are given in response to a strobe message.

Example Input Mapping Scheme

This example's input mapping scheme is a simplified and fixed map of discrete input data and status bits for DeviceNet devices. It is mapped to discrete inputs and the device input data table.

Example Characteristics

This example has the following characteristics:

- Strobe is used to query DeviceNet devices.
- Poll is disabled.
- The input data bit is fixed and occupies the lowest-order bit in the lowest-order byte of the strobe (bit #1).
- One bit of status data is accepted from each node responding to the strobe.
- The status data bit is fixed and occupies the next lowest-order bit in the next lowest-order byte (after the input data bit) of the strobe (bit #2).
- Input and status data bits accepted from each node are mapped to the scanner's discrete-input data table.
- Input and status data bits accepted from each node are fixed and predefined.

Example Framework

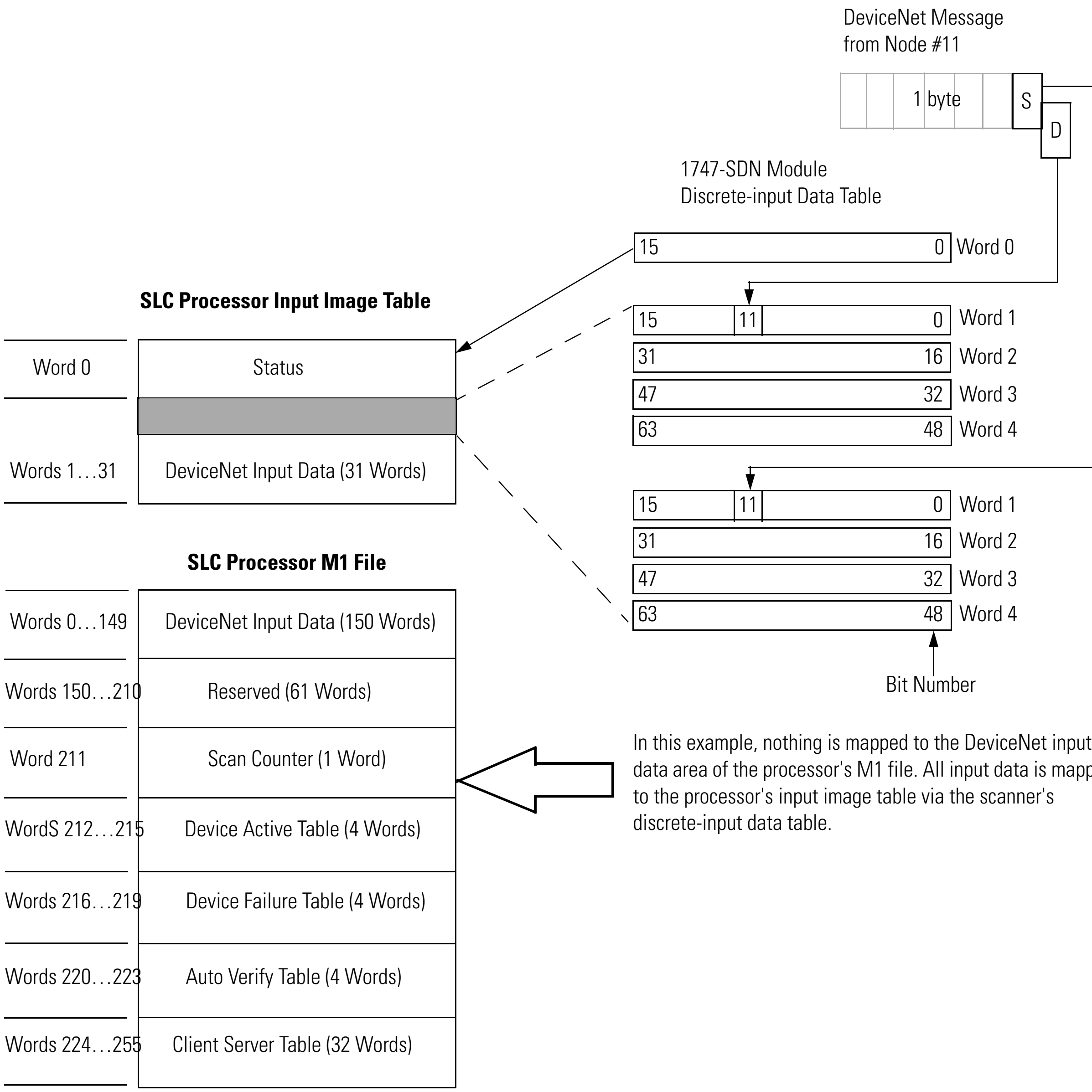
This example adheres to the following structure:

- There cannot be any 1771-SDN modules or other 1747-SDN modules on that network.
- DeviceNet devices may reside only at nodes 1…62.
- Address 0 must be used for the scanner.
- The first word in the device input data table contains the module status word (this is applicable under any mapping scheme).
- Input data and status bits received from nodes 1…62 are mapped to the scanner’s discrete-input data table.

The following illustrates an input-data mapping scheme example for the 1747-SDN module. Input bits are mapped from a device’s message, to the scanner’s input data table, and to the processor’s input image table.

The status and data bits for each scanned device are mapped to the scanner's discrete-input data table. Data bits are mapped in the first four words while status bits are mapped in the next four words of the table. The bit numbering for device data bits begins with 0. This numbering starts over in word 5 for device status bits. This feature makes it possible to directly correlate a device's MAC ID with the placement of its bits in the data table. For example, the data bit for node #11 is mapped to bit 11, word 1 in the data table. Its status bit is mapped to bit 11, word 5 of the data table.

The processor reads the scanner's data table to transfer its contents to the SLC-processor input image table.



In this example, nothing is mapped to the DeviceNet input data area of the processor's M1 file. All input data is mapped to the processor's input image table via the scanner's discrete-input data table.

Example Output Mapping Scheme

This example's output mapping scheme is a simplified and fixed map of the discrete outputs and data from the device output data table to DeviceNet devices.

Devices present in the default database are strobed only; therefore, the output data-map bits are mapped into each network's strobe message. If the discrete table is available, it serves as a source for the strobe bits; otherwise, the source is found in M1/M0 file transfer locations.

Example Characteristics

This example has the following characteristics:

- Strobe is used to send output to the DeviceNet devices.
- Poll is disabled.
- One output data bit each is sent to nodes 1…62.
- The output data bits are embedded in the 8 byte (64 bit) data portion of the DeviceNet strobe message.
- The output bit string source within the strobe message is divided across the discrete outputs in the scanner's discrete-output data table.

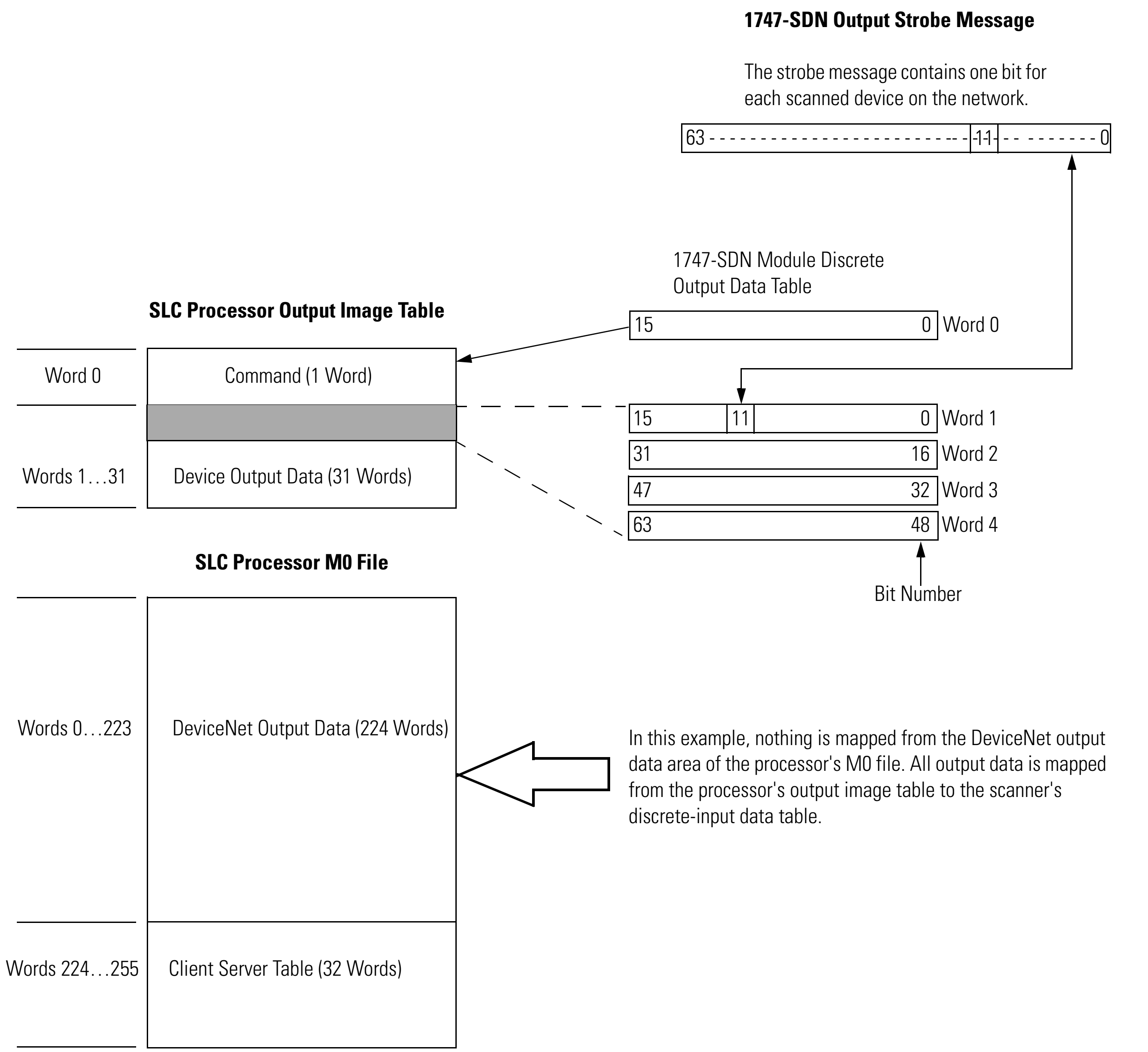
Example Framework

This example adheres to the following structure:

- When a 1747-SDN module is running this configuration, there cannot be any other 1747-SDN or 1771-SDN module on that network.
- DeviceNet devices may reside only at nodes 1…62.
- Address 0 must be used for the scanner.
- The first word in the device output-image data table always contains the module command word (this is applicable under any mapping scheme).
- Output bits received from processor for nodes 1…62 are mapped to the scanner's discrete- output data table.

The following illustrates an output-data mapping scheme example for the 1747-SDN module. Output bits are mapped from the processor's output image table, to the scanner's output data table, and to each device via strobe message.

The processor writes output data for each scanned device in the scanner's discrete-output data table. The scanner then maps each bit into a strobe message that is sent to all scanned devices. The bit number, where a particular device's data is mapped, directly corresponds to that device's MAC ID. This is true for mapping into the scanner's data table as well as the scanner's strobe message. For example, node # 11's output bit is mapped in bit number 11 of the scanner's output data table. In addition, this same output bit is mapped from bit 11 of the data table to bit number 11 in the strobe message.



Configuring the M0/M1 Files by Using RSLogix 500 Software

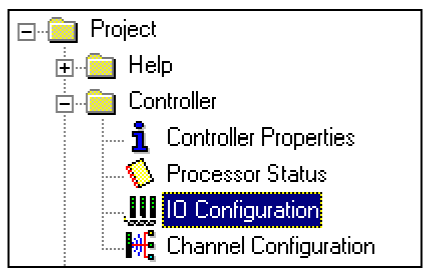
RSLogix 500 I/O Configuration

To enable pass-through access by using a SLC 500 processor, you must configure the M0 and M1 files associated with the 1747-SDN module at a minimum length of 361 words.

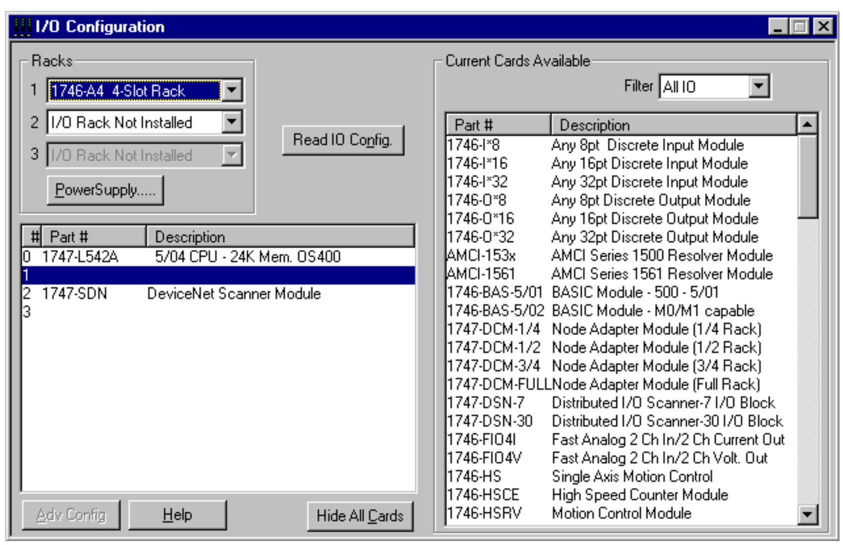
Follow these steps to configure the 1747-SDN module's M0 and M1 files by using RSLogix 500 software.

1. Run RSLogix 500 software and go offline to the SLC 500 processor.

2. Double-click IO Configuration under the Controller folder in the Project dialog.



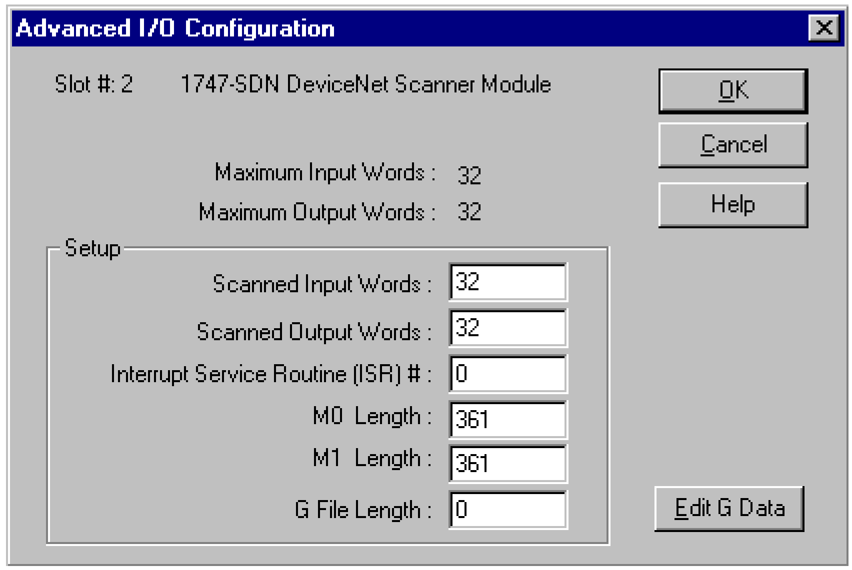
The I/O Configuration dialog opens.



3. Click Read I/O Config to upload the I/O configuration from the processor.

4. Double-click the 1747-SDN module.

   The Advanced I/O Configuration dialog opens.



5. Set the MO Length and the M1 Length to 361 (or greater) as shown above.

6. Click OK.

7. Download the changes to the processor.

IMPORTANT	The SLC 500 processor must be placed in Run mode at least one time after downloading the M0/M1configuration to enable 1747-SDN pass-through transactions.

Programming the Module by Using the SLC M0 and M1 Files

The M0 and M1 files are data files that reside in the module. There is no image for these files in the processor memory. The M0 file is a module output file and the M1 file is a module input file. Both M0 and M1 files are read/write files.

M0 and M1 files can be addressed in your ladder program and they can also be acted upon by the module, independent of the processor scan.

IMPORTANT	During the processor scan, M0 and M1 data can be changed by the processor according to ladder diagram instructions addressing the M0 and M1 files. During the same scan, the module can change M0 and M1 data, independent of the rung logic applied during the scan.

Address the M0-M1 Files

The addressing format for M0 and M1 files is as follows:

```
Mf:S.w/b
```

Where M = module
f = file (0 or 1)
S = slot (1…30)
w = word (0-maximum supplied by the module)
b = bit (0…15)

When You Cannot Use M0-M1 Data File Addresses

You can use M0 and M1 data file addresses in all instructions except the OSR instruction and the instruction parameters below.

Instruction	Parameter (characterized by file indicator #)
BSL BSR	File (bit array)
SQO SQC SQL	File (sequencer file)
LFL LFU	LIFO (stack)
FFL FFU	FIFO (stack)

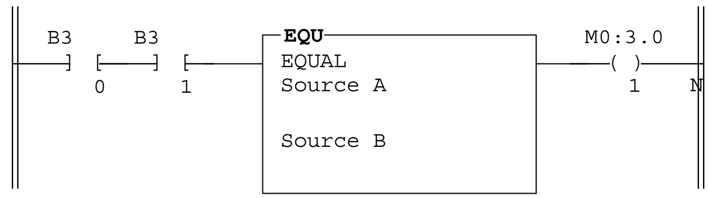
Monitor Bit Instructions with M0 or M1 Addresses

When you monitor a ladder program in the Run or Test mode, the following bit instructions, addressed to an M0 or M1 file, are indicated as false regardless of their actual true/false logical state.

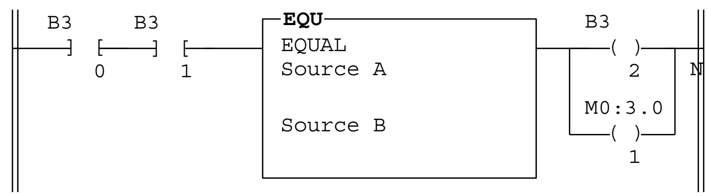
```
  Mf:S.w      Mf:S.w      Mf:S.w        Mf:S.w       Mf:S.w
----] [--   ----]/[--   ---( )-----   ---(L)-----  ---(U)-----
     b           b           b             b            b

    XIC         XIO         OTE           OTL          OTU
```

To show the state of the M0 or M1 addressed bit, transfer the state to an internal processor bit. This is illustrated below, where an internal processor bit is used to indicate the true/false state of a rung.



This rung does not show its true rung state because the EQU instruction is always shown as true and the M0 instruction is always shown as false.

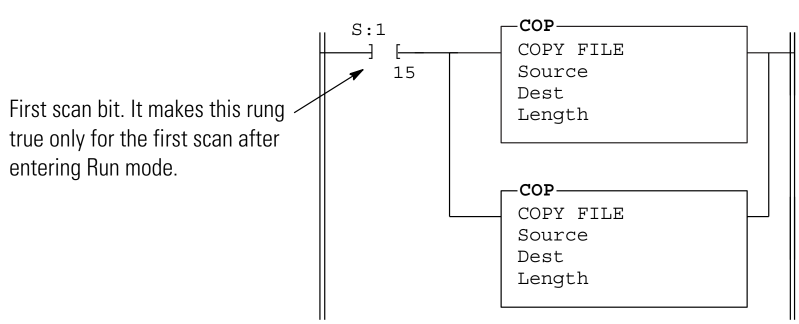


OTE instruction B3/2 has been added to the rung. This instruction shows the true or false state of the rung.

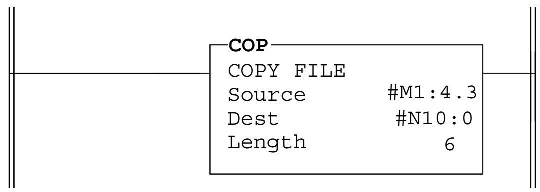
Transfer Data Between Processor Files and M0 or M1 Files

The processor does not contain an image of the M0 or M1 file so you must edit and monitor M0 and M1 file data via instructions in your ladder program. For example, you can copy a block of data from a processor data file to an M0 or M1 data file or vice versa by using the COP (copy) instruction in your ladder program.

The COP instructions below copy data from a processor bit file and integer file to an M0 file.

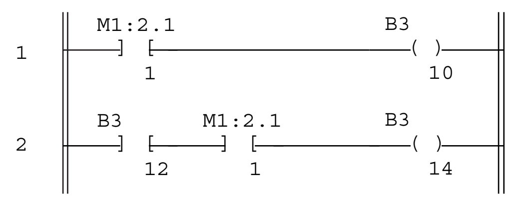


The COP instruction below copies six words of data from an M1 data file in a module positioned in slot four to an integer file (N1:0). This technique is used to monitor the contents of an M0 or M1 data file indirectly, in a processor data file. An update of these six words is made for each SLC program scan.



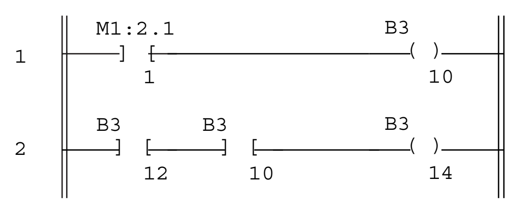
Reduce Scan Time

TIP To reduce processor scan time, use discretion when you use instructions addressing the M0 or M1 files. For example, XIC instruction M1:2.1/1 is used in rungs 1 and 2 below, adding approximately 2 ms to the scan time if you are using an 5/02, series B processor.



XIC instructions in rungs 1 and 2 are addressed to the M1 data file. Each of these instructions adds approximately 1 ms to the scan time (SLC 5/02, series B processor).

In the equivalent rungs below, XIC instruction M1:2.1/1 is used only in rung 1, reducing the scan time by approximately 1 ms.

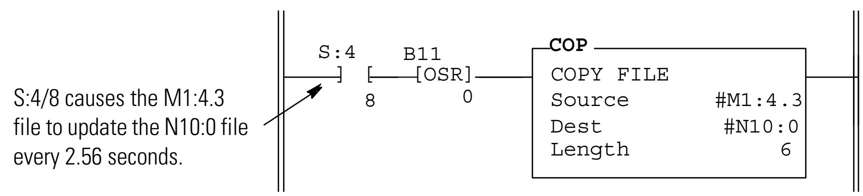


These rungs provide equivalent operation to those of the previous diagram by substituting XIC instruction B3/10 for XIC instruction M1:2.1/1 in rung 2. Scan time is reduced by approximately 1 ms (SLC 5/02 series B processor).

The first two ladder diagrams in the last section illustrate a technique you use to capture and use M0 or M1 data as it exists at a specific time. In the first diagram, bit M1:2.1/1 could change state between rungs 1 and 2. This could interfere with the logic applied in rung 2. The second diagram avoids the problem. If rung 1 is true, bit B3/10 captures this information and places it in rung 2.

The following diagram illustrates another economizing technique. The COP instruction addresses an M1 file, adding approximately 4.29 ms to the scan time if you are using an SLC 5/02, series B processor. You can save scan time by making this rung true periodically only. For example, you can use a clock bit S:4/8 (clock bits are discussed in the programming manual).

A rung such as this might be used when you want to monitor the contents of the M1 file, but monitoring need not be continuous.



In this example, a COP instruction can be used to monitor the contents of an M1 file. When the instruction goes true, the six words of data in file #M1:4.3 is captured as it exists at that time and placed in file #N10:0. All subsequent logic should address the data in #N10:0. The data will be consistent and it shortens scan time by eliminating reads to the module each time an M0 or M1 address is encountered in the program.

1747-SDN Module Firmware History

Purpose

This section describes enhancements, corrected anomalies and other important information for this revision and previous revisions of the 1747-SDN module firmware.

Revision 8.002

The 1747-SDN series C module includes these firmware enhancements.

- Embedded EDS file - The module contains its own electronic data sheet (EDS) file within its firmware. This allows RSNetWorx for DeviceNet software, 5.0 or later, to upload and register the EDS file directly from the scanner. In previous versions, you had to locate the EDS file on a website and manually install it.
- AutoScan - This feature automatically sets up communication with the devices on your DeviceNet network. You no longer have to use RSNetWorx for DeviceNet software to map the devices into a scanlist.

The 1747-SDN series C module includes all features provided in the series B module.

Revisions 7.005 and 7.006 Known Anomalies

Firmware revisions 7.005 and 7.006 contain the following known anomalies:

- Inability to flash update the module if connections to all slave devices are not made. The module displays a 98 error.
- Certain application programs (ladder programs) may cause the SLC controller to fault due to exceeded shared memory lock time. The SLC controller displays a 57 error. The module does not display an error.
- Internal performance changes may cause timing dependant applications to fail.
- Saving ADR data may cause the module to fault upon next power cycle. The module displays a E9 error. Upon next power cycle, the scanlist and configuration data defaults to out of box values.

- If a slave in the scanlist does not match the identity of the slave at that address on the network, the module does not communicate with that slave again. After the mismatch is resolved, you have to cycle power to the module to reestablish communication.

If you have not experienced any of these anomalies and want to flash upgrade this module to firmware revision 7.006, please contact Rockwell Automation Technical Support at 440-646-3223.

Revision 6.002

This section describes the enhancements and corrected anomalies for firmware revision 6.002.

Electronic Data Sheet Requirement

The 6.002 firmware revision of the module requires the latest EDS file for RSNetWorx for DeviceNet software. If the software displays the device as an unknown device, then you must download the latest EDS file.

You can get the latest EDS file online at:

http://www.ab.com/networks/eds.

Follow these steps once you are at this location.

1. Click DeviceNet.
2. Enter the catalog number: 1747-SDN.
3. Enter Major Revision: 6.
4. Enter the Minor Revision: 2.
5. Click Search.

Enhancements

This section describes the enhancements to the firmware.

Enhanced Explicit Message Pass-through Capability

With the 6.002 firmware revision, the size of the M0/M1 file could be expanded to 395 words. This can accommodate a larger pass-through message than previous firmware revisions. For example, this enhanced capability now allows DeviceNet PanelView terminal downloads.

M1 File Includes Device Active Table Section

With this revision, words 206…209 in the M1 file are now used for the Device Active Table. The scanner assigns one bit in consecutive order to consecutive device addresses starting at node 0 at M1.S.206/0. If a bit is set, it indicates that the node is in the scanner's scanlist and has successfully communicated with the module. The bit is reset if the scanner tries to communicate to the node but the node has gone offline.

Immediate Input/Output

With this revision, the scanner will act upon immediate input and output instructions (IIM, IOM) from the SLC 500 programmable controller.

Corrected Anomalies

Anomaly	Description
Reset I/O run bit in scanner copy of I/O during transition to processor Program mode.	This would prevent immediate change from Idle to Run mode before the I/O scan of the processor when the processor transitions to Run mode. This anomaly has been corrected.
Unwanted information from a previous connection remained when the former connection timed out while waiting for a fragment ack.	This prevented a new connection from being established since the connection state was incorrect. This anomaly has been corrected.
Check for error after setting EPR on slave connection.	If a device state conflict error is returned by a slave when the EPR bit is set, the scanner would not report the error. This anomaly has been corrected.

Revision 5.001

This section describes the enhancements for firmware revision 5.001.

Enhancement

The following enhancement was made to the firmware for this revision.

Scanlist Commit in Run/Idle Mode

With the 5.001 firmware revision, with the module in Idle mode, the scanlist can be committed to the module's memory while the programmable controller is in Run mode.

Revision 4.026

This section describes the enhancements for firmware revision 4.026.

Enhancement

The following enhancement was made to the firmware for this revision.

Electronic Keying

With the 4.026 firmware revision, electronic keying was enhanced to include major and minor revision checking.

Data Organization

Understand the Data Organization of the Module

The module has four data areas to transfer data, status, and command information between the module and the processor.

- SLC input image table
- SLC output image table
- SLC M1 file
- SLC M0 file

Input and Output Image Tables

The following table describes the mapping of the 1747-SDN input and output image tables and the M1 and M0 files.

Words	SLC Input Image	Words	SLC Output Image
0	Status	0	Command
1…31	DeviceNet Input Data (31 words)	1…31	DeviceNet Output Data (31 words)

Words	SLC M1 File	Words	SLC M0 File
0…149	DeviceNet Input Data (150 words)	0…149	DeviceNet Output Data (150 words)
150…205	Reserved (56 words)	150…223	Reserved (74 words)
206…209	Device Active Table (4 words)		
210	Node Address/Status Indicator (1 word)		
211	Scan Counter (1 word)		
212…215	Device Idle Table (4 words)		
216…219	Device Failure Table (4 words)		
220…223	Auto Verify Failure Table (4 words)		
224…255	Explicit Message Program Control (32 words)	224…255	Explicit Message Program Control (32 words)
256…394	Pass-through (139 words)	256…394	Pass-through (139 words)

Upload Input Data from the Module to the SLC Processor

The SLC 500 processor reads input data from the module by using two methods.

- Input image table
- M1 file transfer

Input Image Table

The input image table is a 32-word table for the module slot that is updated by the processor with each program scan. The first word (word 0) is reserved for the module status register. The remaining 31 words can be used to transfer DeviceNet input data to the SLC input image table. The addressing format is:

I:S.w/b

Where S = slot
w = word (0…31)
b = bit (0…15)

Module Status Register

The module status register is located at word 0 in the input image area for the slot. Bits 0…5 echo back to the processor, the current state of bits 0…5 of the module command register. The echoes verify that the commands were executed. The module sets the remaining bits when it detects a problem. The bits latch in the ON state until the problem clears. Bits 6 and 8 indicate that you should read the device failure table for more specific information about which devices failed.

You can use bit 6 to keep the communication port in the Idle mode until the bit clears. When the bit clears, this indicates that all devices in the scanner's scanlist are up and available. When the devices are available, you can put the port in Run mode. If a device failure is detected, you can put the communication into the Idle mode, so that all output devices go to a safe state.

The SLC program can monitor the bits in the module status register and set the appropriate bits of the module command register to automatically control the operating mode of the module should a device failure occur.

Bit Table

Status Word I:s.0		Operating Mode Description
Bit	**Operating Mode**	
0	1 = Run mode, 0 = Idle mode (echoed from the module command register)	Run The module maps output data from its scanner output table (M0) and discrete outputs to each device on the network. Inputs are received and mapped into the scanner input table (M1) and discrete inputs. Outputs on the network are under SLC program control. Idle The scanner does not map output data to the devices, but keeps network connections to devices open so device failures can be detected. Input data is returned from devices, and mapped into the scanner input table (M1) and the discrete inputs. Outputs on the network are not under program control and will be in their configured Idle state. The scanner must be put into this mode to perform offline configuration of the scanner database tables.
1	1 = fault network (echoed from the module command register)	Fault Network The scanner has stopped communicating with devices on the network. No outputs or inputs are mapped. Outputs on the network are not under program control. If scanner was in Run mode, devices will go to their fault state.
2	Reserved	
3	Reserved	
4	1 = disable network (echoed from the module command register)	Disable Network The DeviceNet channel is disabled for communication. No communication may occur over this channel. Outputs on the network are not under program control. If scanner was in Run mode, devices will go to their fault state.
5	Reserved	
6	1 = device failure (at least one device failed)	Device Failure One or more of the devices in the scanner’s scanlist has failed to communicate with the scanner.
7	Reserved	
8	1 = autoverify failure (at least one device has failed auto verify)	Autoverify Failure One or more of the devices in the scanner’s scanlist is returning an incorrect number of bytes of data in response to a connection establishment, according to the information stored in the scanner’s scanlist.
9	Reserved	
10	1 = communication failure	Communication Failure There is no communication on the port.
11	Reserved	
12	1 = duplicate node address failure	Duplicate Node Address Failure There is another node with the same address as the scanner on the network.
13	Reserved	
14	Reserved	
15	1 = Explicit Message Program Control Response available in M1 file 0 = Empty	Explicit Message Program Control A response to a previously sent Explicit Message is now available to be read/interpreted by the ladder program.

SLC M1 File

The SLC M1 file is a file with up to 256 words or more that can be used to transfer a large quantity of information to the module with a single SLC instruction.

IMPORTANT	Transferring data using this file takes more time than using the input image table.

The first 150 words are used for data transfer from the module. The remaining 106 words are reserved for:

- device active table.
- node status.
- scan counter.
- device idle table.
- device failure table.
- auto verify table.
- explicit message program control.

For a details on the mapping of input and output image tables, refer to page 133.

Device Active Table

Words 206…209 in the M1 file are used for the Device Active Table. The scanner assigns one bit in consecutive order to consecutive device addresses starting at node 0 at M1.S.206/0. If a bit is set, it indicates that the node is in the scanner's scanlist and has successfully communicated with the module. The bit is reset if the scanner tries to communicate to the node but the node has gone offline.

Node Address/Status Indicator

Word 210 is used for node address and scanner diagnostic information displayed in numeric codes. The high byte is the node address and the low byte is the status for that node. These codes and their descriptions are listed on page 91.

Scan Counter

Word 211 is used for the module scan counter. The module increments this counter whenever a scan of the DeviceNet devices is completed. The counter rolls over when it reaches a maximum value of 65,535. It is located at M1:S.211.

Device Idle Table

Words 212…215 in the M1 file are used for the device idle table. This table indicates that there are devices on the network in Idle mode. The module tracks devices in Idle mode by assigning one of the 64 bits in the table to each device on the network. The bits are assigned in consecutive order to consecutive device addresses starting at mode 0 at M1.S.212/0.

Device Failure Table

Words 216…219 in the M1 file are used for the device failure table. This table indicates communication failures of devices on the network. The module tracks device failures by assigning one of the 64 bits in the table to each device on the network. The bits are assigned in consecutive order to consecutive device addresses starting at mode 0 at M1.S.216/0.

Auto Verify Failure Table

Words 220…223 in the M1 file are used for the auto verify failure table. The auto verify failure table is used to verify that data size received from the device matches the setting in the scanner's scanlist entry for that node. This check occurs at connection establishment time. The module tracks auto verify failures by assigning one of the 64 bits in the table to each device on the network. The bits are assigned in consecutive order to consecutive device addresses starting with node 0 at M1:S.220/0. If the bit is set, the corresponding node has failed to verify.

Explicit Message Program Control

Words 224…255 are used for Explicit Message Program Control. Use this feature to configure device parameters on your DeviceNet network via the M0 and M1 files in the SLC processor that is controlling these devices.

Download Output Data to the Module

The SLC 500 processor writes output data to the module by using two methods.

- Output image table
- M0 file transfer

Output Image Table

The output image table is a 32-word table for the module slot that is updated from the processor with each program scan. The first word (word 0) of this table is reserved for the module command register. The remaining 31 words can be used to transfer data from the SLC output table to the DeviceNet nodes.

Module Command Register

The module command register is located at word 0 in the output image area for the slot. To execute a command, set the appropriate bits in the module command word by using SLC ladder instructions.

The following table describes the functionality of the command register bits.

Command Register Bits

Command Word 0:S.0		Operating Mode Description
Bit	Operating Mode	
0	1 = Run mode 0 = Idle mode	Run The module maps output data from its scanner output table (M0) and discrete outputs to each device on the network. Inputs are received and mapped into the scanner input table (M1) and discrete inputs. Outputs on the network are under SLC program control. Idle The scanner does not map output data to the devices, but keeps network connections to devices open so device failures can be detected. Input data is returned from devices, and mapped into the scanner input table (M1) and the discrete inputs. Outputs on the network are not under program control and will be in their configured idle state. The scanner is put into this mode to perform online configuration of the scanner database tables.
1	1 = fault network	Fault Network The scanner stops communicating with devices on the network. No outputs or inputs are mapped. Outputs on the network are not under program control. If scanner was in Run mode, devices will go to their fault state.
2	1 = restore to factory defaults/flush memory	Restore to Factory Defaults/Flush Memory The scanner may receive a scanlist or configuration that causes inappropriate scanner behavior. It may be necessary to restore to the module's factory defaults to recover from this state. This procedure is described below.
3	Reserved[1]	
4	1 = disable network	Disable Network The DeviceNet channel is disabled for communication. No communication may occur over this channel. Outputs on the network are not under program control. If scanner was in Run mode, devices will go to their fault state.
5	Reserved[1]	
6	1 = halt scanner	Halt Scanner All scanner operations stop when this command is issued. No communication occur over either DeviceNet port. No M-file or discrete I/O mapping occurs. Outputs on the network are not under program control. If scanner was in Run mode, devices will go to their fault state.
7	1 = reboot	Reboot This command causes the scanner to reset as though power had been cycled. When this command is issued, all scanner communication stops for the duration of the scanner's initialization sequence. Outputs on the network are no longer under program control. If scanner was in Run mode, devices will go to their fault state.
8…15	Reserved[1]	

(1) All reserve bits must be set to zero or improper operation may result.

Restore Factory Default Settings/Flush Memory

Follow these steps to restore the module's factory default settings.

1. While the SLC processor in Program mode, clear the module command word 0 and set bit 2 of word 0 for the module to ON or 1 (starting with bit 0 going from right to left, this is the third bit).

IMPORTANT	If other bits in the module command word are set, then the module will generate an error.

2. Set bit 0 of word 0 for the module to OFF or 0.
3. On the SLC processor, cycle the keyswitch from Program mode to Run mode then back to Program mode.

IMPORTANT	The ladder logic may set bit 0 of word 0 for the module. If this happens, then you must disable or delete this rung from the ladder program.

4. When the code E9 is displayed on the Address/Error status display, the default settings have been restored.
5. Set bit 2 of word 0 for the module to OFF or 0.
6. Cycle power to the 1746 chassis to restore normal operation to the module.

   The module will now have its factory default settings:

   - node 63.
   - 125 Kbps.
   - no scanlist.

SLC M0 File

The SLC M0 file is a 256 word file that can be used to transfer a large quantity of information to the module with a single SLC instruction. Transferring data by using this file can take several scans and more time than by using the output image table.

The first 150 words are used for sending data to DeviceNet nodes. The next 74 words are reserved for future use, the next 32 words are used for explicit message program control, and the final 139 words are used for pass-through.

Please see page 133 for more details on the mapping of the input and output image tables and the M1 and M0 files.

Explicit Message Program Control

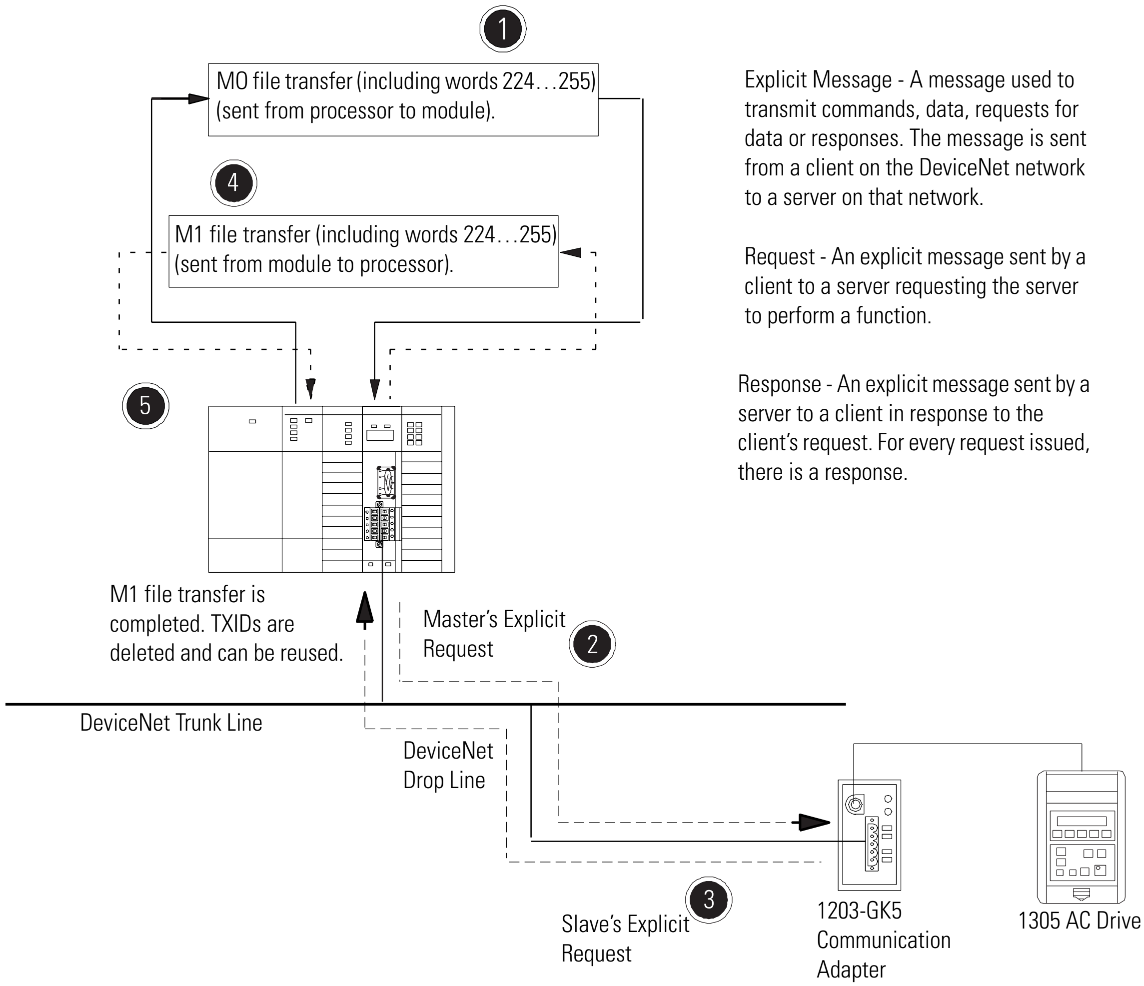
Using Explicit Message Program Control

Use the Explicit Message Program Control feature to configure device parameters on your DeviceNet network via the M0 and M1 files.

Use the Explicit Message Program Control feature to:

- transmit configuration data from your module to its slave devices on your DeviceNet network.
- receive status and diagnostics from these devices on your DeviceNet network.
- make runtime adjustments to device parameters according to changing conditions detected by your processor.

How the Explicit Message Program Control Feature Works



Explicit Message - A message used to transmit commands, data, requests for data or responses. The message is sent from a client on the DeviceNet network to a server on that network.

Request - An explicit message sent by a client to a server requesting the server to perform a function.

Response - An explicit message sent by a server to a client in response to the client's request. For every request issued, there is a response.

1. Format an M0 file transfer in the processor to send an Explicit Message Request to the module (download).
2. The module transmits the Explicit Message Request to the slave device over the DeviceNet network.
3. The slave device transmits the Explicit Message Response back to the scanner and is queued into a file transfer buffer.
4. The processor uses an M1 file transfer to retrieve the Explicit Message Response from the scanner's buffer (upload).
5. Format an M0 file transfer with a Delete Response Command and the current transaction ID read in step 4.

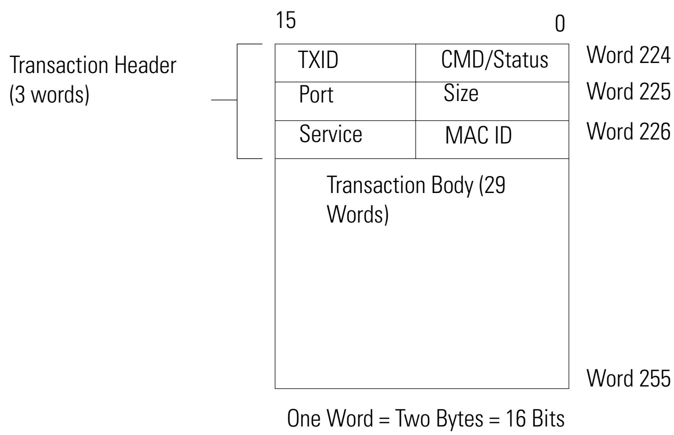
   The transaction IDs are deleted and can be reused.

The module requires a precisely-formatted M0 and M1 file transfer size of 32 words including words 224…255. The module uses the file memory content as a client/server request.

How to Format the Explicit Message Transaction Block

Up to ten 32-word transaction blocks may be queued within the module for Explicit Message Program Control. The transaction blocks accommodate both the download of Explicit Message Requests and the upload of Explicit Message Responses.

The module can accommodate one request or response for each transaction block. You must format each transaction block as shown in the following figure.



The transaction block is 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 only the response message.

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

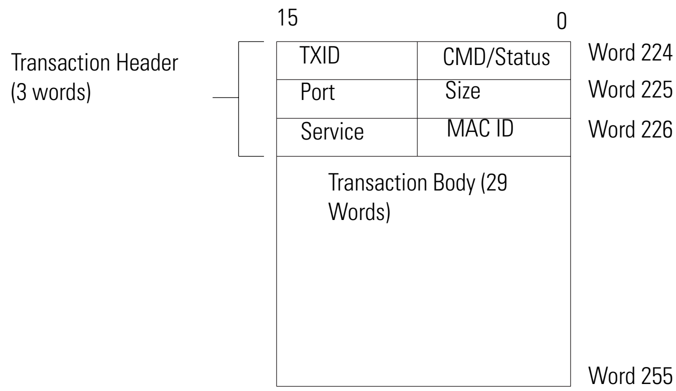
- Command/status – for each download, you assign a command code to instruct the scanner how to administer the request.

Command Code	Description
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 transaction from response queue
5…255	Reserved

For each upload, the status code provides the processor with status on the device and its response.

Status Code	Description
0	Ignore transaction block (block empty)
1	Transaction completed successfully
2	Transaction in progress (not ready)
3	Error – slave not in scanlist
4	Error – slave offline
5	Error – DeviceNet port disabled/offline
6	Error – transaction TXID unknown
7	Error – slave not responding to explicit request
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

- TXID (transaction ID) – when you create and download a request to the scanner, the processor's ladder logic program assigns a TXID to the transaction. This is a one-byte integer in the range of 1…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. The ladder logic program monitors rollover and usage of TXID values.



One Word = Two Bytes = 16 Bits

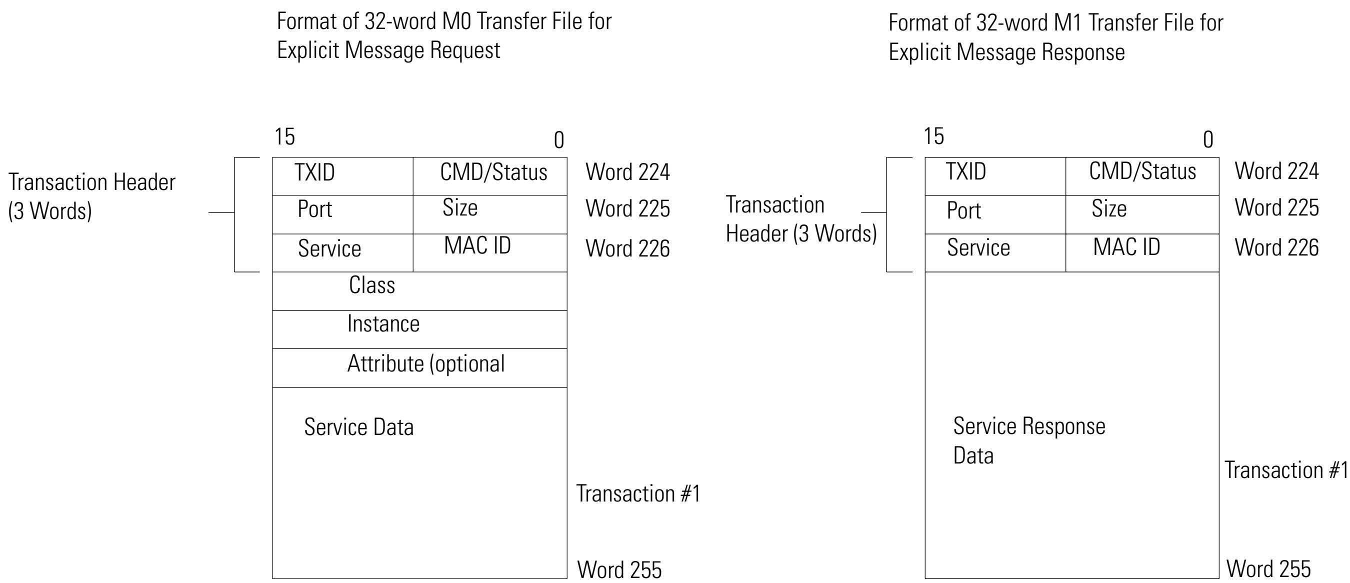
- Size – the size of the transaction body in bytes. The transaction body can be as many as 29 words (58 bytes) in length. If the size exceeds 29 words, an error code will be returned.
- Port – the DeviceNet port (zero) where the transaction is routed.
- MAC ID (node address) – the DeviceNet network address of the slave device where the transaction is sent. This value can range from 0…63.

  The port and MAC ID attributes coupled together identify the target slave device. The slave device must be listed in the module's scanlist and be online for the Explicit Message transaction to be completed successfully.

- Service – for each Explicit Message Request and Response, the service attribute contains the service request and response codes that match the corresponding request for the TXID.

The following figure describes the format and mapping of transaction blocks for request and response messages in the module.

Transaction Blocks



How the Processor and Module Manage Messages

File transfer operations between the processor and the scanner always originate in the processor. The module can wait only for the processor to download a transaction block to the module or request an upload of a transaction block from the module.

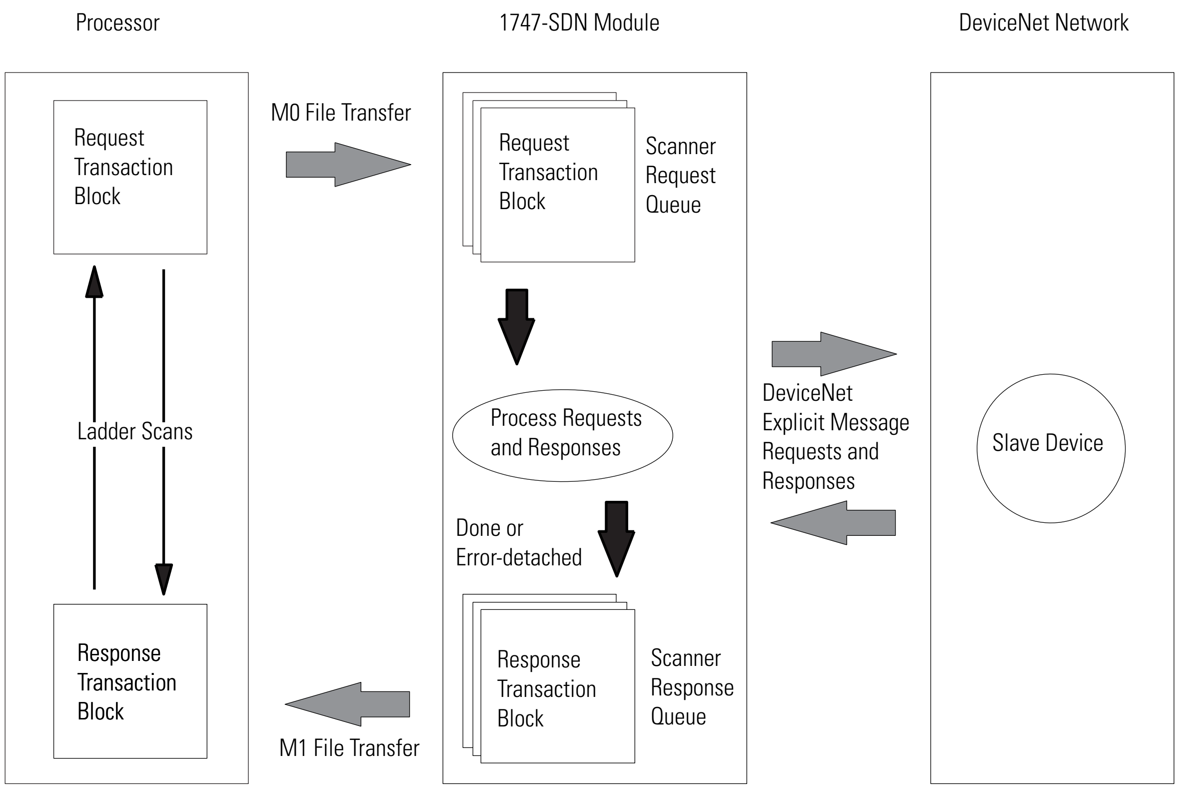
Once an Explicit Message Request transaction block is downloaded to the module, a ladder logic program in the processor polls the module for the transaction block containing the Explicit Message Response for that request. This is done by the processor with an M1 file transfer on the module.

Depending on the network load, the scanner could take a few seconds to complete the request. When a response is loaded, bit 15 of the module status register is set to 1. The program may have to poll the module a number of times before the scanner returns a Response Transaction Block.

The module recognizes I/O data and control as higher priorities over explicit messaging on the DeviceNet network.

Message lengths and slave device types impact transaction message completion times. If the processor has queued multiple Explicit Message Transactions to the module for multiple slave devices, the transactions with the slaves may not complete in the order in which the requests were received. The slave responses are queued to the 32 word M1 file transfer in the order in which they are received.

As response transaction blocks are uploaded, the processor's program matches the responses to the requests by using the TXID field.



Explicit Message Program Control Limitations

- The processor is always the DeviceNet client and the slave is always the DeviceNet server.
- A maximum of ten Explicit Message Request Transaction Blocks with the execute command can be queued to the module at any time. For example, ten M0 file transfers containing one transactions each, can be queued at any time. The module receives and deletes any additional client/server requests with the execute command over the maximum of ten.

As transactions are removed from the queue and response transaction blocks are returned to the processor, additional transaction blocks can be issued in their place, as long as the total does not exceed ten.

- The module supports one transaction block per upload and download.
- Request Transaction Blocks can be queued only for slave devices of the module and must appear in the module's scanlist.
- If a slave device is not communicating at the time the module processes its Request Transaction Block, the module will return an error status for that transaction.
- Check documentation of destination device for specifics concerning services supported and Class Instance Attribute information. At a minimum, the module supports the following DeviceNet services in Request Transaction Blocks.

Service Name	Service Code	Example
Get_Attribute_Single	0E hex	Upload a single parameter value from a device
Set_Attribute_Single	10 hex	Download a single parameter value to a device
Get_Attribute_All	01 hex	Upload all parameter values from a device
Set_Attribute_All	02 hex	Download all parameter values to a device

- All transaction blocks are processed, therefore an unused transaction block must be left blank.
- Client/Server commands and requests with transaction IDs that are in use are ignored by the module.
- If a slave device returns a DeviceNet error in response to the request downloaded from the processor, the scanner recognizes the error as a successful transaction (status code =1).

A failure to respond to the request within the number of retries or timeout period specified for the Explicit Message Connection is recognized by the module as an error. The error code is returned in the status attribute of the transaction header.

Explicit Messaging Error Codes

Error codes have two bytes of data. The first byte is a General Error Code and the second is an optional Additional Code Field that may contain additional information about the error. If this field is unused, the value 0FFH is shown. The following table describes explicit messaging error codes.

CIP Response Error Codes

Numeric Code	Name	Description
02H	Resource unavailable	A needed resource was not available
08H	Service unsupported	Service is not defined or implemented for this class/instance
09H	Invalid attribute value	Data is invalid for the specified attribute
0BH	Already in requested state	Object is in the requested state - redundant request
0CH	Object state conflict	Not allowed with object in present state
0EH	Attribute cannot be set	Read-only attribute
0FH	Privilege violation	A permission/privilege check failed
10H	Device state conflict	Not allowed with device in present state
11H	Reply too big	Reply larger than buffer allocated when connection was established
13H	Too little data	Request included insufficient data
14H	Attribute not supported	Attribute number is incorrect
15H	Too much data	Request included extra data
16H	Object does not exist	Class/instance numbers are incorrect
18H	No stored attribute data	Attribute data was not saved prior to this request
19H	Store operation failure	Attribute data was not successfully saved
1FH	Vendor-specific error	Second byte may offer details - refer to vendor documentation
20H	Invalid parameter	Parameter associated with request is invalid
D0H FFH	Reserved and service-specific errors	Used only when none of the standard error codes supplemented by the second byte accurately describes the problem

Bridge

The module's support of explicit message transfer.

Background Poll Ratio

Sets the frequency of poll messages to a device in relation to the number of I/O scans. For example, if the ratio is set at 10, that device will be polled once every 10 scans.

Change of State

A type of I/O data communication. The module can send and receive data with slave devices that have the change of state feature. Data is sent whenever a data change occurs. Data is updated at the rate of the heartbeat.

Cyclic

A type of I/O data communication. The module can send and receive data with slave devices that have the cyclic feature. Data is sent only at a user-configurable rate.

Discrete I/O

The transfer of 1…32 words between an SLC 500 processor and the 1747-SDN module. All 32 words of input data and all 32 words of output data are updated on each SLC program scan.

Dual Mode

The module is in Dual mode when it serves as a master to one or more slaves and as a slave to another master simultaneously.

EDS

Electronic data sheets are vendor-supplied templates that specifies how information is displayed as well as what is an appropriate entry (value).

Explicit Messaging

A type of messaging used for lower priority tasks, such as configuration and data monitoring.

Heartbeat Rate

Devices that are configured for change of state data can also send a heartbeat signal to indicate proper operation.

Host Platform

The computer on which the application software is run.

I/O

An abbreviation for input and output.

Implicit Messaging

The type of messaging used for high-priority I/O control data; for example, change of state, cyclic, polled, or strobed.

Input Data

Data produced by a DeviceNet device and collected by the module for the host platform to read.

Interscan Delay

The time between I/O scans (polled and strobed). It is the time the 1747-SDN module waits between the last poll message request and the start of the next scan cycle.

M1/M0 File Transfer

A method of moving large amounts of data between a SLC 500 processor and the 1747-SDN module. It transfers files containing a maximum of 256 words and may take more than one SLC program scan to complete.

MAC ID

The network address of a DeviceNet node.

Network

The DeviceNet network or the RSNetWorx for DeviceNet software representation of the network.

Node

Hardware that is assigned a single address on the network (also referred to as a device).

Offline

When the host platform is not communicating on the network.

Online

When the host platform is configured and enabled to communicate on the network.

Output Data

Data produced by the host platform that is written to the module's memory. This data is sent by the module to DeviceNet devices.

Polled

A type of input/output-data communication. A polled message solicits a response from a single, specified device on the network (a point-to-point transfer of data).

Processor

The SLC 500 programmable controller.

Record

The node address and channel-specific memory assigned in the module's nonvolatile storage for a node in the scanlist.

Rx

An abbreviation for receive.

Scanlist

The list of devices (nodes) with which the scanner is configured to exchange I/O data.

Scanner

The function of the 1747-SDN module to support the exchange of I/O with slave modules.

Slave Mode

The 1747-SDN module is in Slave mode when it is placed in another 1747-SDN module's scanlist as a slave device.

Strobed

A type of I/O data communication. A strobed message solicits a response from each strobed device (a multicast transfer). It is a 64-bit message that contains one bit for each device on the network.

Tx

An abbreviation for transmit.

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://support.rockwellautomation.com, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect Support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://support.rockwellautomation.com.

Installation Assistance

If you experience a problem with a hardware module within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your module up and running.

United States	1.440.646.3223 Monday – Friday, 8am – 5pm EST
Outside United States	Please contact your local Rockwell Automation representative for any technical support issues.

New Product Satisfaction Return

Rockwell tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning, it may need to be returned.

United States	Contact your distributor. You must provide a Customer Support case number (see phone number above to obtain one) to your distributor in order to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for return procedure.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1747-UM655B-EN-P - June 2007
Supersedes Publication 1747-6.5.5 - August 2000