

BridgeWay ***Ethernet to DeviceNet Gateway*** ***User Manual***

Part No. AB7603



Pyramid Solutions
1850 Research Drive, Suite 300
Troy, Michigan 48083

Phone 248-524-3890
Web www.pyramid-solutions.com



PYRAMID SOLUTIONS

Publication PUB-AB7603-008

Table of Contents

Preface.....	iv
How to Use This Manual	iv
Important User Information	iv
Related Documentation.....	v
BridgeWay Module Description	1-1
Overview.....	1-1
Theory of Operation.....	1-2
DeviceNet Features.....	1-3
Ethernet Features.....	1-3
IT-Features.....	1-4
System Requirements	1-4
Hardware Description	1-6
Installation.....	2-1
Installation and Operation Requirements	2-1
Power and Network Connections	2-2
Connecting Power	2-3
Connecting DeviceNet.....	2-4
Connecting to Ethernet.....	2-4
Configuration	3-1
Ethernet Network Configuration	3-1
DeviceNet Network Configuration	3-8
DeviceNet I/O Configuration.....	3-10
Quick Start	4-1
DeviceNet Network Configuration	4-1
DeviceNet I/O Configuration.....	4-8
Ethernet Network Configuration	4-11
Using the Ethernet File System.....	4-13
DeviceNet Interface.....	5-1
Network Communications.....	5-1
Configuration	5-1
Automatic Baud Rate Detection	5-1

Slave Device Communication.....	5-2
Scan Cycles.....	5-2
I/O Message Types	5-3
I/O Mapping	5-3
Proxy for Group 2 Only Devices	5-3
Run/Idle Mode	5-4
Automatic Device Recovery (ADR)	5-5
Interaction with I/O Tables.....	5-6
EtherNet/IP Interface	6-1
Product Features	6-1
CIP Objects	6-1
CIP Messaging	6-2
Assembly Objects and Connections.....	6-4
Notes About Using ControlLogix I/O Connections....	6-10
CIP Bridging	6-11
Modbus/TCP Interface.....	7-1
Supported Commands.....	7-1
Supported Exception Codes.....	7-2
Modbus/TCP Addressing.....	7-3
I/O Data Content.....	7-4
I/O Data Format.....	7-5
File System.....	8-1
File System Conventions	8-1
Security.....	8-2
Structure	8-4
Default Files.....	8-5
Virtual File System.....	8-6
System Files	8-6
Configuration Files.....	8-7
Password Files	8-11
Other Files	8-13
BridgeWay Web Page Files.....	8-17
IT Functionality	9-1
Default User Accounts.....	9-1

The FTP Server	9-2
The Telnet Server	9-2
HTTP Server.....	9-8
SSI Functionality.....	9-9
Email Client	9-23
Displaying I/O Data on a Web Page.....	9-24
Status and Diagnostics.....	10-1
BridgeWay LEDs	10-1
Diagnostic Web Pages	10-4
Status Assembly.....	10-7
Specifications.....	11-1
Environmental Specifications	11-1
EMC Directive Compliance.....	11-1
Electrical Specifications.....	11-1
Mechanical Specifications	11-2
I/O Data Sizes	11-2
Connectors.....	12-1
Power	12-1
DeviceNet	12-2
Ethernet RJ45.....	12-3
Auxiliary RS-232 9 Pin D-Subminiature.....	12-4
Warranty	13-1
Support	14-1
Technical Product Assistance	14-1
Contact Information	14-1

Preface

How to Use This Manual

This manual provides an overview of the BridgeWay Ethernet to DeviceNet Gateway. It describes how to configure and operate it.

Important User Information

The data and illustrations found in this document are not binding. We reserve the right to modify our products in line with our policy of product development. The information in this document is subject to change and should not be considered as a commitment by Pyramid Solutions. Pyramid Solutions assumes no responsibility for errors that may appear in this document.

There are many applications of the BridgeWay module. Those responsible for the use of this device must satisfy themselves that all necessary steps have been taken to verify an application meets all performance and safety requirements including any applicable laws, regulations, codes, and standards.

The illustrations and samples in this guide are intended solely for the purpose of example. Pyramid Solutions does not assume responsibility or liability for actual use based upon the examples shown in this publication.

DeviceNet is a trademark of Open Device Vendor Association (ODVA), Inc.

EtherNet/IP is a trademark of ControlNet International LTD.

RSLinx, RSNetWorx are trademarks of Rockwell Software.

MS-DOS and Windows are trademarks of the Microsoft Corporation.

Related Documentation

Document Name	Author	Web Page
DeviceNet Specification	ODVA	www.odva.org
EtherNet/IP Specification	ControlNet International LTD.	www.odva.org
Modbus/TCP	Schneider Automation	www.modicon.com/openbus/index.html

Table 2-1 Related Documentation

BridgeWay Module Description

Overview

The BridgeWay Ethernet to DeviceNet Gateway allows you to seamlessly connect your Information or Control level networks with your Device level network.

The Ethernet to DeviceNet Gateway provides full DeviceNet Master functionality allowing connectivity to 63 DeviceNet slaves devices along with an Ethernet TCP/IP interface that supports IT protocols such as SMTP, FTP, HTTP and control protocols such as EtherNet/IP and Modbus/TCP.

Examples of BridgeWay Ethernet to DeviceNet applications:

- BridgeWay can be used as a gateway to connect information or control level networks to device level networks for programming, configuration, control or data collection. (E.g. Modbus/TCP to DeviceNet)
- BridgeWay can provide router/bridge functionality to connect EtherNet/IP to DeviceNet.
- BridgeWay can provide an internal web server to allow remote viewing of data via a web browser and can offer email and file transfer capabilities to enhance your application.

Theory of Operation

The BridgeWay provides centralized data storage, the “PassageWay™”, for data that is shared between the DeviceNet and Ethernet networks. Data is placed into the PassageWay by one network interface, allowing the data to be read through the other network interface.

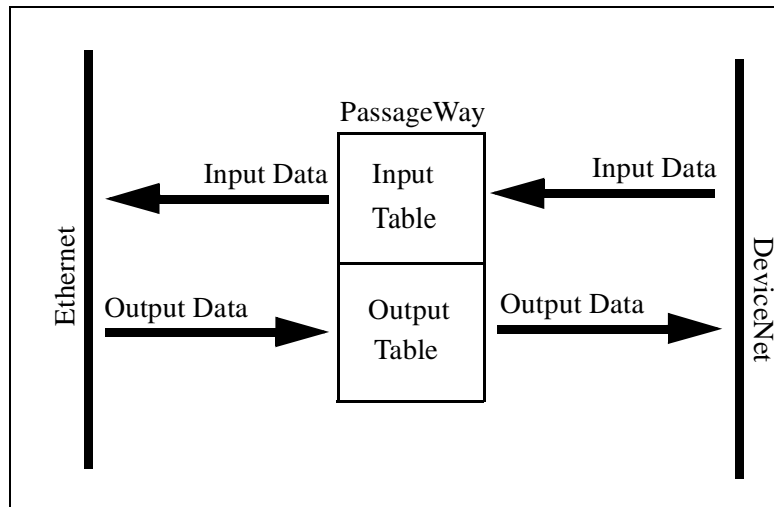


Figure 1-1 BridgeWay PassageWay Operation

The BridgeWay appears as a single device on either network using standard protocol mechanisms. No special, or extended, protocol features are required of the devices on either network to read or write the data flowing through the PassageWay; all cross-network activity is transparent to the devices on either network.

DeviceNet Features

- DeviceNet Master scanner functionality supporting up to 63 DeviceNet slave devices
- Explicit Messaging and Bit Strobe, Poll, and Change of State (COS) I/O connections.
- Baud rates of 125, 250, and 500 Kbps.
- Automatic baud rate detection option may be enabled or disabled.
- Automatic Address Recovery can be configured to replace a faulted slave device with a replacement device at the same MAC ID.
- Configuration Recovery can be configured for slave devices so that a newly replaced slave can be configured to the same settings of the device it replaces. Combined with Automatic Address Recovery this feature is known as Automatic Device Recovery (ADR).
- Transfers 500 bytes Input (IN) and 500 bytes Output (OUT) I/O.

Ethernet Features

- Supports the EtherNet/IP protocol, Adapter Class with I/O Server, Message Client, Message Server, and CIP Message Routing.
- Supports the Modbus/TCP protocol with up to 8 simultaneous connections. Conforms to the Modbus/TCP specification 1.0.
- Features UDP and TCP/IP protocol stack.
- Supports DHCP/Bootp and ARP for IP Address configuration.

IT-Features

- The BridgeWay features a flexible file system with two security levels. The size available for user files is approximately 1.4 Mbyte.
- An FTP server provides easy file management using standard FTP clients.
- A Telnet server featuring a command line interface similar to the MS-DOS™ environment.
- A flexible HTTP server (Web server) with Server Side Includes (SSI) functionality. These are commands to the web server embedded in the HTML code. This enables the user to access the IN/OUT area using a customizable web page interface.
- Firmware updates of the DeviceNet control portion of the BridgeWay using the RS232 port and BridgeWay Configuration Tool.
- Email client capability. (The BridgeWay can only connect to mail servers that do not require a username/password to access. A mail server such as that used by many internet providers could not be used).

System Requirements

The following hardware and software components are needed to use the BridgeWay Ethernet to DeviceNet device.

Required Hardware

- BridgeWay Ethernet to DeviceNet module.
- DeviceNet cabling, power, and devices forming a DeviceNet network.
- Ethernet cabling and power.
- PC or other controller with access to the Ethernet network.
- PC to execute DeviceNet Configuration Software. The DeviceNet slave devices the BridgeWay communicates with are specified using a DeviceNet Configuration Software Tool such as RSNetWorx for DeviceNet from Rockwell Software or HMS AnyBus Net Tool-DN.
- 24 VDC power to the BridgeWay module. (DeviceNet power may be used.)

Optional Hardware

- A PC with a serial RS232 COM port to be used by the BridgeWay Configuration Tool Software for downloading firmware updates.
- RS232 null-modem cable (pins 2 and 3 swapped) from the PC to the BridgeWay module.
- DIN rail to mount the BridgeWay.

Required Software

- DeviceNet configuration software such as RSNetWorx for DeviceNet to configure DeviceNet devices and BridgeWay's DeviceNet operation.
RSLink version 2.31 or later is required.

Optional Software

- BridgeWay Configuration Tool Software to update BridgeWay's firmware.

Hardware Description

All connections, whether power or fieldbus, to the BridgeWay are made on one end of the module. Phoenix connectors are provided for power and DeviceNet connections. A RJ-style connector is provided for Ethernet connection. There is a 9-pin D-Subminiature connector for the auxiliary RS-232 port that is used for field firmware upgrades. See “Installation” on page 2-1 for more details on the connectors.

There is an 8 position dip switch on the end of the module that can be used to select a portion of a default IP address that may be used to permit an intranet connection. See “Ethernet Network Configuration” on page 3-1 for more details on configuring the IP address using the switches.

On the front of the BridgeWay module are 7 LEDs that are used for status indication. These LEDs provide visual status for the overall module, the DeviceNet interface, and the Ethernet interface. See “BridgeWay LEDs” on page 10-1 for details on how the LEDs are used.

The back of the module has a DIN rail mount to allow the module to be mounted on a DIN rail.

Installation

Installation and Operation Requirements

- Power, input and output (I/O) wiring must be in accordance with Class 1, Division 2 wiring methods - article 501-4(b) of the National Electric Code, NFPA 70 and in accordance with local codes.
- **Warning - Explosion Hazard** - Substitution of components may impair suitability for Class 1, Division 2.
- **Warning - Explosion Hazard** - When in hazardous locations turn off power before replacing or wiring modules.
- **Warning - Explosion Hazard** - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.
- Terminal tightening torque must be between 5-7 lbs-in (0.5-0.8 Nm).
- For use in Class 2 circuits only.
- Suitable for surrounding temperature of 65 degrees C maximum.
- Use 60/75 C copper wire only.

Power and Network Connections

The power and network connections to the BridgeWay are made on the end of the module. Figure 2-1 indicates the location of each connector.

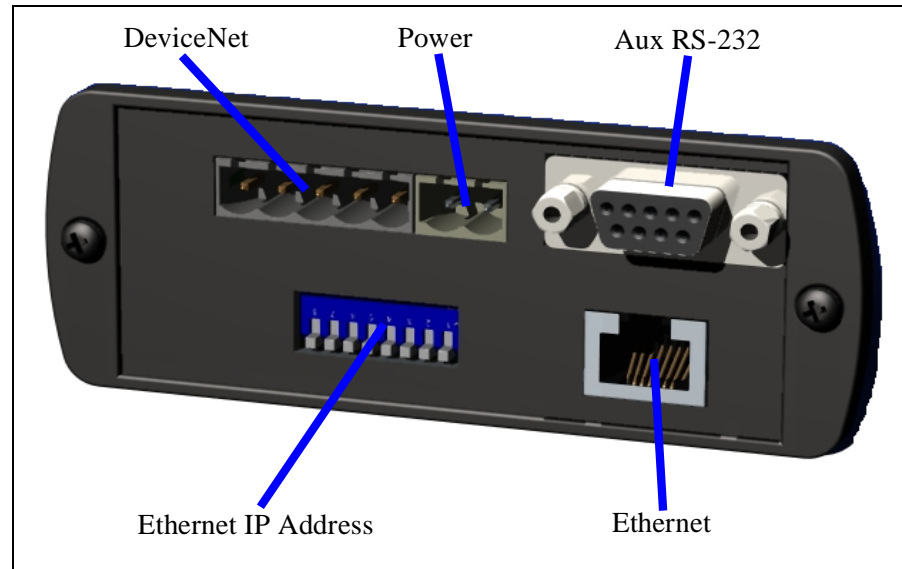


Figure 2-1 BridgeWay Power and Network Connections

Connecting Power

The power connection is a 2-pin terminal block located on the end of the module. The female terminal block connector is provided with the BridgeWay. Connections to be made are illustrated in Figure 2-2.

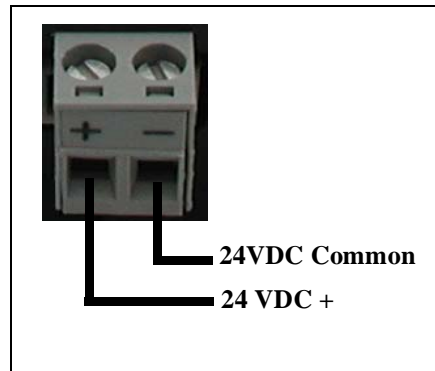


Figure 2-2 Power Connection

The BridgeWay requires 24 volts DC power. The module will start immediately when power is applied (There is no On/Off switch on the module).

Connecting DeviceNet

The DeviceNet network connection is a 5-pin terminal block located next to the power connection on the end of the module. The female terminal block connector is provided with the BridgeWay. Connections to be made are illustrated in Figure 2-3.

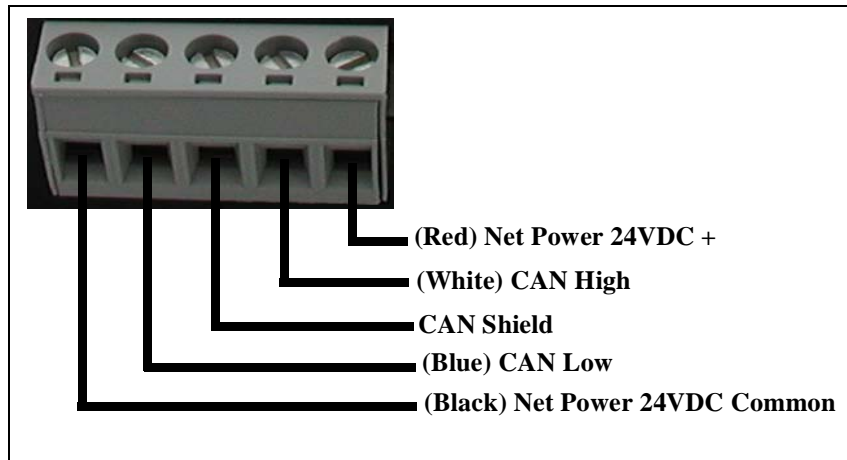


Figure 2-3 DeviceNet Connection

A 120 ohm termination resistor (not provided) may be required for proper network termination. See the DeviceNet Specification for specific rules on DeviceNet connections and termination.

For information on setting the DeviceNet network configuration (MAC ID, baud rate, etc.), see “DeviceNet Network Configuration” on page 3-8.

Connecting to Ethernet

The Ethernet connection uses a standard RJ45 connector (not provided). This is plugged into the socket on the end of the module.

For information on setting the Ethernet IP configuration (IP address, DHCP, etc.), see “Ethernet Network Configuration” on page 3-1.

Configuration

This chapter describes how the BridgeWay Ethernet to DeviceNet Gateway is configured. The next chapter walks the reader through the configuration of the BridgeWay using the commonly available configuration tools.

Ethernet Network Configuration

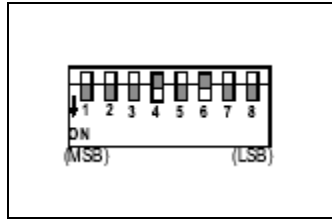
Several methods may be used to set the IP Address. These methods include; IP Address Configuration Switch, DHCP/Bootp protocol, web browser, and the ARP protocol.

Setting the IP Address with the Configuration Switch

If DHCP/BootP is not enabled or a server is not found and the Configuration Switch is non zero, on power up the value of the switch is used to form an the IP Address. The switch represents the binary value of the last byte in the 4 byte IP address. In this case it is *n*.

IP address:	192.168.1. <i>n</i>
Subnet mask:	255.255.255.0
Gateway address:	0.0.0.0 (No gateway set)

This is a private address and can only be used on a local intranet. In such a case a Web Browser such as Microsoft's Internet Explorer can be used to access the BridgeWay's web page which allows changing the IP Address, Subnet mask, and GateWay address settings.

DIP Switch Example**Figure 3-1 IP Configuration DIP Switch**

The switches are set to 00010100 (20 decimal) (The switch position is shown in White in the diagram.)

The IP address of the module will be set to 192.168.1.20.

Note: The numbers on the switches on the IP configuration DIP switch do NOT correspond to bit locations in the address value. In fact, they are reversed. i.e. bit 0 is set by switch 8.

Setting the IP Address Using DHCP/BootP

When DHCP/BootP is enabled and a DHCP or BootP server is found, the IP address, Subnet mask, and Gateway address is automatically configured by the DHCP/BootP server. It can be enabled using the BridgeWay's Settings web page.

Note: The use of DHCP is the default configuration for the BridgeWay as shipped.

Setting the IP Address Using Address Resolution Protocol (ARP)

The module's IP address can be changed using the ARP command from a PC. The new IP address will be stored in non-volatile memory. ARP requires the module's Ethernet MAC Address that is printed on a label on the back of the module.

Note: ARP cannot be used to change the subnet mask and gateway address of the BridgeWay. These can be configured using the BridgeWay's Settings web page.

Switch all 8 switches of the IP Configuration DIP switch to the ON position.

Note: The ARP/Ping capability is disabled unless all switches are ON.

On a PC connected to the BridgeWay on Ethernet bring up an MS DOS™ window and type:

```
arp -s <IP address> <MAC address>
```

The arp -s command will store the IP and MAC addresses in the PC's ARP table.

Next type:

```
ping <IP address>
```

When the Ping command is executed, the PC sends this information to the module using the MAC address. The module detects that it was addressed with the correct MAC address and adopts the IP address sent by the PC.

Next type:

```
arp -d <IP address>
```

The arp -d will remove the static route from the PC's ARP table.

Switch all 8 switches of the IP Configuration DIP switch to the OFF position to disable the feature.

This method can be used to reconfigure a module that has been previously configured, or even to reconfigure modules outside the host's subnet.

Arp/Ping Example:

The following commands will set the IP address of a BridgeWay with MAC address 00-30-11-02-00-5E to 65.106.34.252.

```
arp -s 65.106.34.252 00-30-11-02-00-5e
```

```
ping 65.106.34.252
```

```
arp -d 65.106.34.252
```

Setting the IP Address Using the Web Page

The ethernet addresses can also be configured using the Status and Settings web page resident on the BridgeWay. The Status and Settings web page appears as shown below.

BridgeWay Status & Settings	
Home Active Nodes Idle Nodes Faulted Nodes Invalid Nodes Node Status Status & Settings	
Status & Settings	
Product Name	Ethernet to DeviceNet Gateway
Vendor ID	170
Product Type	12
Product Code	1
Product Revision	1.07.01
Anybus Revision	1.18
Serial Number	0xA0004DA9
DeviceNet MAC ID	1
DeviceNet Baud Rate	500K
DeviceNet Autobaud Enabled	NO
DeviceNet Network Status	ONLINE-IDLE
DeviceNet Scanner Mode	IDLE
IP Address	<input type="text" value="65.106.34.213"/>
Subnet Mask	<input type="text" value="255.255.255.192"/>
Gateway IP Address	<input type="text" value="65.106.34.193"/>
Mail Server IP Address	<input type="text" value="0.0.0.0"/>
DHCP enabled	<input checked="" type="checkbox"/>
<input type="button" value="Submit Values"/>	
<input type="button" value="Reset Module"/>	

Figure 3-2 Status and Settings Web Page

The IP address, subnet mask, and default gateway address are displayed in the edit boxes on the web page. Changing any values and clicking the Submit Values button will set the addresses in the BridgeWay. Note that a power cycle or module reset is required for the changes to take effect.

The Reset Module button can be used to reset the BridgeWay from the web browser. The Scanner Mode will display “RESETTING...” while the module resets and comes back online. The web page will be refreshed after the module has booted.

Note: If your web browser is configured to cache web pages, it may appear that the BridgeWay has not changed address after you power cycle the module. Make sure that the browsers settings are configured to always reload pages. On Internet Explorer this is done in the Temporary Internet Files Settings dialog by selecting the “Every Visit” option for when the browser should check for page changes.

IP Address Initialization

The following flowchart describes how the IP configuration is determined when the BridgeWay is powered up.

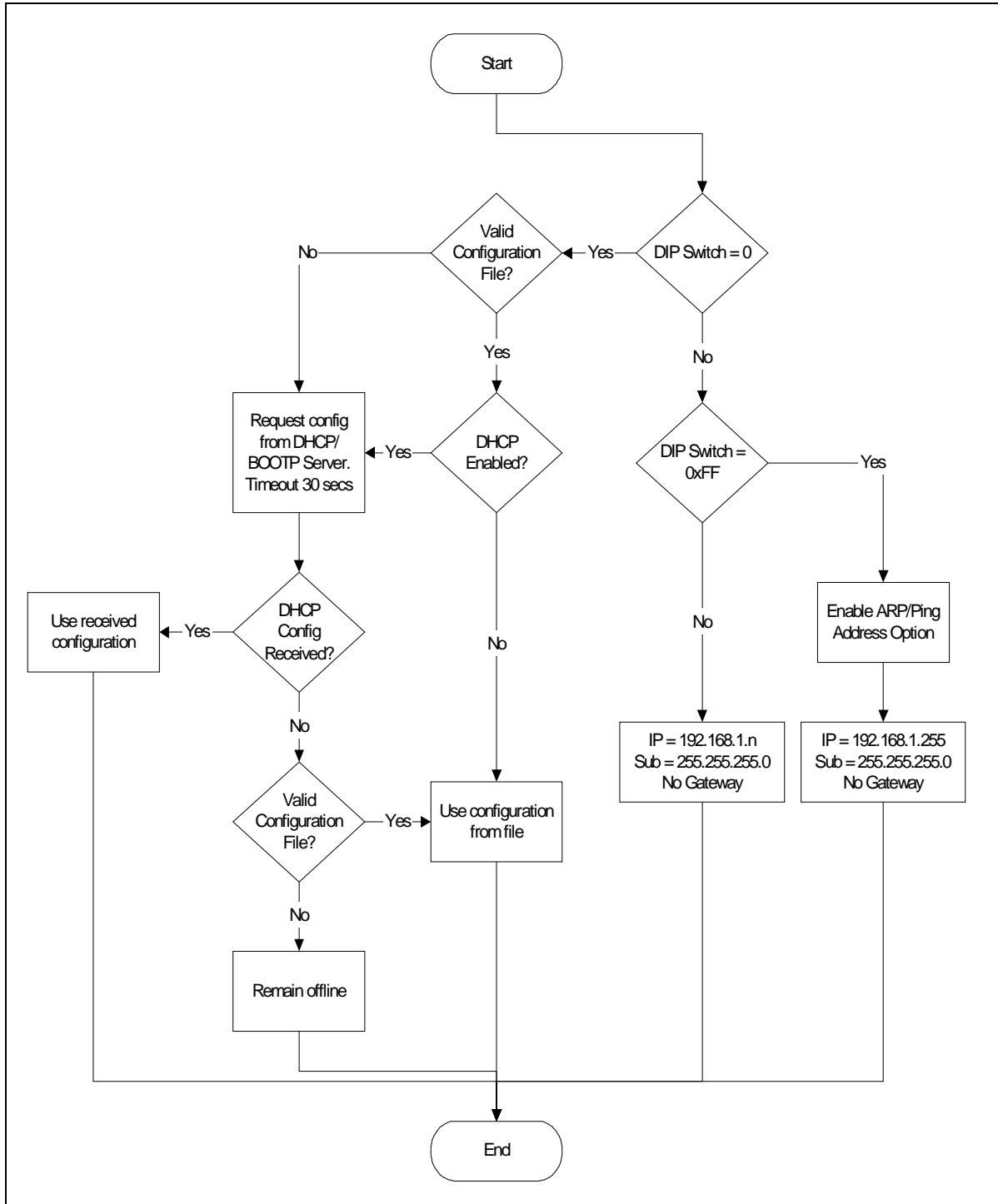


Figure 3-3 IP Configuration Initialization Sequence

DeviceNet Network Configuration

DeviceNet MAC ID

The DeviceNet MAC ID is software configurable; there are no switches used to set it.

When first powered, the factory default configuration sets the MAC ID to 63. The allowable range for MAC ID is 0 - 63.

A DeviceNet node commissioning tool (e.g. RSNetWorx or NetTool-DN) is used to set the MAC ID to the desired value for the application. The value is stored in non-volatile memory and used on subsequent power ups.

DeviceNet Baud Rate

The DeviceNet baud rate is software configurable; there are no switches used to set it. The baud rate can be set to one of the 3 supported rates, or the module can be configured to use automatic baud rate detection.

When first powered, the factory default configuration sets the baud rate to 125Kbps. The BridgeWay supports baud rates of 125, 250, and 500 Kbps.

If the baud rate is to be set, a DeviceNet node commissioning tool is used to set the baud rate to the desired value for the application. The value is stored in non-volatile memory and used on subsequent power ups.

If autobaud is desired, the option must be enabled. This is done through a class/instance editor in RSNetworx, or through the parameter editor in NetTool-DN. See “DeviceNet Network Configuration” on page 4-1 for details on setting this option using these tools.

Note: When the autobaud option is enabled, the baud rate attribute cannot be changed.

EDS File

Each device on a DeviceNet network has an associated EDS file containing all necessary information about the device. This file is used by the network configuration tools, such as RSNetWorx or NetTool-DN, during configuration of the network.

The latest version of the EDS file for the BridgeWay can be downloaded from Pyramid Solution's web site, or received by contacting Pyramid Solutions.

DeviceNet I/O Configuration

I/O Mapping

The DeviceNet I/O configuration defines the format of the Input and Output tables in the PassageWay, or the *mapping* of DeviceNet slaves' I/O data to the I/O tables. See "Theory of Operation" on page 1-2 for a discussion on the PassageWay and the use of I/O tables in the BridgeWay. As slaves are added to the BridgeWay's DeviceNet scanner configuration, the location in the I/O tables of each part of the slave's I/O data is determined and stored.

Note: The organization of the I/O tables is very important. This defines the format of the data that will be exposed to the EtherNet/IP scanner. The Input and Output table formats should be planned and documented to ensure the EtherNet/IP scanner is working with the correct data from the DeviceNet network.

I/O Size Limitations

The size of the I/O data that can be exchanged with the EtherNet/IP scanner and, hence, the size of the I/O tables is restricted as explained below.

- The Input table size cannot be larger than 500 bytes.
- The Output table size cannot be larger than 500 bytes.
- Either table may be empty (size of 0 bytes).

DeviceNet Configuration Tools

The DeviceNet I/O configuration is set using a DeviceNet configuration tool. This manual is not intended to replace the user manual for the configuration tool, so will not provide details on using the tool.

The next chapter provides an example application, and covers the use of Rockwell Software's RSNetWorx for DeviceNet and HMS' NetTool-DN.

Quick Start

This chapter provides a step by step explanation of configuration of the BridgeWay Ethernet to DeviceNet Gateway. It is intended to be used as a beginner's guide to configuring and using the BridgeWay using RSNetWorx for DeviceNet or NetTool-DN. It also demonstrates how easy it is to create your own webpage and upload it to the module.

Requirements: An Ethernet network and a PC connected to that network, running Windows™ 98/ME/2000/XP, Microsoft Internet Explorer 5.5 or higher, a DeviceNet configuration tool such as Rockwell Software's RSNetWorx for DeviceNet or HMS Net Tool-DN. RSLink version 2.31 or later is required.

DeviceNet Network Configuration

Configuration of the DeviceNet network interface involves using a DeviceNet node commissioning tool to set the BridgeWay's MAC ID and baud rate. The following sections explain how this is done using either Rockwell Software's RSNetWorx for DeviceNet or HMS' NetTool-DN.

Note: The BridgeWay defaults to 125K baud out of the box. If your DeviceNet network is not running at 125K baud, the BridgeWay must be powered up on a local network with the node commissioning tool at 125K baud. Do not attempt to commission the BridgeWay on a network configured at a different baud rate.

Node Commissioning with RSNetWorx for DeviceNet

Step 1: Connect the module to your DeviceNet network.

- Make sure a PC running RSNetWorx for DeviceNet is connected to the DeviceNet network.
- With the BridgeWay un-powered, connect the DeviceNet network cable to the DeviceNet connector of the module. (See "Connecting DeviceNet" on page 2-4)
- Power up the BridgeWay.

Step 2: Locate the module is on the network.

- RSNNetWorx allows browsing on the network to identify devices.
- Select the *Network* menu option and pull down menu.
- Select the *Single Pass Browse* option and wait for browsing to complete. If this is the first time RSNNetWorx has been used with a BridgeWay, the BridgeWay's icon should indicate "Unrecognizable Device".

Step 3: Register the BridgeWay EDS file in RSNNetWorx.

RSNetWorx requires an electronic data sheet (EDS) to recognize a device and its capabilities. An EDS file is available on the Pyramid Solutions web site. The EDS file must be registered with RSNNetWorx before configuration can continue.

- Select the *Tools* menu option and pull down menu.
- Select the *EDS Wizard* option.
- Click on *Next*.
- Select *Register an EDS File* option and click *Next*.
- Select *Register a Single File* and enter, or browse to, the location of the EDS file for BridgeWay.
- Click *Next* or *Finish* for the remaining option screens.
- Select the *Single Pass Browse* option and wait for browsing to complete. Now an icon identifying the device as the BridgeWay module should appear.

Step 4: Put the BridgeWay in Idle Mode

The BridgeWay powers up in Idle mode, and is controlled by the EtherNet/IP scanner through the output command register. Make sure that the module is in Idle mode by verifying that the BridgeWay Status LED is flashing green (the LED is solid green when the module is in Run mode). If the module is not in Idle mode, place the connected Ethernet scanner in Idle or Program mode.

Step 5: Set the DeviceNet MAC ID and Baud Rate

- Select the *Tools* menu option and pull down menu.
- Select the *Node Commissioning* option. Another screen appears.
- Click on *Browse* and choose the DeviceNet network.
- When the browse is completed, double click on the BridgeWay icon.
- Enter the desired MAC address and/or baud rate, then click the *Apply* button.

Note: The BridgeWay will automatically reset if a new MAC ID is entered. If only the baud rate is changed the BridgeWay must be power cycled before the new baud rate will take effect.

Note: When the MAC ID is changed, the BridgeWay's I/O configuration is cleared.

Step 6: Enabling the Autobaud Option

If it is desirable to have the BridgeWay automatically determine the network baud rate, the Autobaud option must be enabled.

- Highlight the BridgeWay module by left clicking on its icon.
- Select the *Device* menu option and pull down menu.
- Select the *Class Instance Editor* option. A pop up Message box appears. Click on *Yes*. Another screen appears.
- There are several parts to this screen. Make sure the check box titled *Values in Decimal* is NOT checked. At the top right is an *Object Address* with 3 text boxes. Set the values in these boxes as follows:
 - *Class* set to 3.
 - *Instance* set to 1.
 - *Attribute* set to 64h.
- To the left of the *Object Class* section is one titled *Service Code*. There's a text box with a pull down selection titled *Description*. Pull down the selections and select "Set Single Attribute".
- The box titled *Data Sent to the Device* is now available. At the far left of this box enter a "01" to enable autobaud, or a "00" to disable it. Then click on the *Execute* button.
- A message should appear in the *Data received from device box* saying the execution was completed.

Note: Changes to the autobaud option configuration do not take effect until the module has been power cycled.

Note: If the BridgeWay is the only master on the DeviceNet network, DO NOT enable autobaud. Automatic baud detection requires there to be traffic on the network, there is typically no traffic until the master establishes connections.

Node Commissioning with NetTool-DN

Step 1: Connect the module to your network

- Make sure a PC running NetTool-DN (version 1.0.0.1 or later) is connected to the DeviceNet network via the NetTool-DN RS-232 interface adapter.
- With the BridgeWay un-powered, connect the DeviceNet network cable to the DeviceNet connector of the module. (See “Connecting DeviceNet” on page 2-4)
- Power up the BridgeWay.

Step 2: Locate the module is on the network.

- Start NetTool-DN on the PC.
- NetTool-DN starts up and displays a screen prompting for a network name. Enter a name such as “BridgeWay” to refer to the network and click *Ok*. A blank screen then appears.
- Select the *Tools* menu item and pull down its menu selections. Select *Configure Drivers For...* option.
- Highlight the name of the network and click on it. A Driver Dialog box appears.
- Click on *7262 Serial RS232 DeviceNet Tool Adapter* to highlight it and click *Ok*. A screen to configure the RS-232 communications between the Adapter and the PC appears.
- Select the PC serial port being used to connect to the NetTool-DN RS-232 adapter.
- Set the DeviceNet baud rate to 125K baud. Set the MAC ID to a value that will not conflict with devices already on the network. (including the BridgeWay)
- Click *Go Online*. A confirmation message indicating that the adapter has gone online should appear. Click *Ok*.
- NetTool-DN should now display a network screen with the icons for the devices it finds on the DeviceNet network. If this is the first time that NetTool-DN has been used with a BridgeWay, the BridgeWay’s icon will indicate “No EDS file registers for this device”.

Step 3: Register the BridgeWay EDS file with NetTool-DN

NetTool-DN requires an electronic data sheet (EDS) to recognize a device and its capabilities. An EDS file is available on the Pyramid Solutions web site. The EDS file must be registered with NetTool-DN before configuration can continue.

- From the *Tools* menu, select *Install EDS Files*.
- Enter the path, or browse to the location of the EDS file for the BridgeWay.
- Click *Open*.
- Select the *Tools* menu option, then *Update*, and click on the network name. The BridgeWay icon should be properly displayed on the network screen.

Step 4: Put the BridgeWay in Idle Mode

The BridgeWay powers up in Idle mode, and is controlled by the EtherNet/IP scanner through the output command register. Make sure that the module is in Idle mode by verifying that the BridgeWay Status LED is flashing green (the LED is solid green when the module is in Run mode). If the module is not in Idle mode, place the connected Ethernet scanner in Idle or Program mode.

Step 5: Set the DeviceNet MAC ID

- Right click on the BridgeWay icon and select *Device*.
- Pull down the next menu and select *Change Node Address*.
- Select or enter the desired MAC ID and click *Ok*.

Note: The BridgeWay will automatically reset if a new MAC ID is entered.

Note: When the MAC ID is changed, the BridgeWay's I/O configuration is cleared.

Step 6: Enabling the Autobaud Option

If it is desirable to have the BridgeWay automatically determine the network baud rate, the Autobaud option must be enabled.

- Highlight the BridgeWay module by right clicking with the cursor on its icon.
- Select the *Device* menu option and pull down menu, then select *Properties*. A parameter screen is displayed.
- Click on *Upload*. The parameter values will be read from the device.
- Put the cursor “Autobaud” parameter value. Click to highlight it and then click again to get a menu. From this menu select “Enable” or “Disable” to enable or disable the autobaud option respectively.
- Click *Download* to send the mode change to the BridgeWay.
- Click *Close*.

Note: Changes to the autobaud option configuration do not take effect until the module has been power cycled.

Note: If the BridgeWay is the only master on the DeviceNet network, DO NOT enable autobaud. Automatic baud detection requires there to be traffic on the network, there is typically no traffic until the master establishes connections.

DeviceNet I/O Configuration

DeviceNet I/O configuration involves using a DeviceNet configuration tool to set the BridgeWay's scan list and I/O table mapping. The following sections explain how this is done using either Rockwell Software's RSNetWorx for DeviceNet or HMS' NetTool-DN.

I/O Configuration Using RSNetWorx

Step 1: Set up BridgeWay module's DeviceNet scan list

In most cases it will be necessary to return the BridgeWay to Idle mode as described in "Step 4: Put the BridgeWay in Idle Mode" on page 4-2. Once in Idle mode the following steps should be taken to configure the scan list.

- Select the *Network* menu and *Browse Single Scan*. Wait for browsing to complete.
- Select the *Network* menu and *Upload*. Wait for the device information to be uploaded from the network.
- Double click on the BridgeWay icon to bring up the module description screen. Several tabs appear on the top of the screen.
- Click the *Scanlist* tab. The screen shows 2 columns. On the left is a list of "Available devices" that may be added to the scan list. On the right is a list of devices that are configured in the scan list.
- Check the *AutoMap on Add* check box.
- Select the devices whose I/O is to be exchanged with the EtherNet/IP scanner from the "Available devices" column. Click the ">" button for each one to move it to the scan list.
- Select the *Input* tab. The Input mapping screen is displayed. The top portion gives a list of the devices in the scan list that the BridgeWay receives input data from. The bottom shows the location in the Input table where the data will be placed for each device. **This shows the format of the Input table of the BridgeWay. This is the format of the input data that will be sent to the EtherNet/IP scanner. See "I/O Mapping" on page 3-4.**

- Select the *Output* tab. The Output mapping screen is displayed. The top portion gives a list of the devices in the scan list that the BridgeWay will send output data to. The bottom shows the location in the Output table where the data will be placed for each device. **This shows the format of the Output table of the BridgeWay. This is the format of the output data that will be sent to the BridgeWay from the EtherNet/IP scanner. See “I/O Mapping” on page 3-4**
- Click the *Apply* button, and *Yes* to download the scanlist to the BridgeWay.
- The BridgeWay starts scanning as soon as it finds entries in its scanlist. However, in Idle mode, output data will not be sent to the devices.

Note: Automap is used in this example to for simplicity. In some cases, the user may wish to organize the I/O data in other ways; this can be done using the *Advanced* data table editor in the Input and Output tabs. See the RSNetWorx manual for complete details.

I/O Configuration Using NetTool-DN

Step 1: Set up BridgeWay module's DeviceNet Scanlist

In most cases it will be necessary to return the BridgeWay to Idle mode as described in “Step 4: Put the BridgeWay in Idle Mode” on page 4-6. Once in Idle mode the following steps should be taken to configure the scan list.

- From the network display screen right click on the BridgeWay icon and select *Device*. Pull down its associated menu and select *Properties*. This displays the Parameters screen.
- Click on the *Scanner* tab. The scan list display screen appears with two columns. The left column displays a list of devices found on the network that can be added to the scanlist. The right column displays the devices that are configured in the scanlist.
- Click *Upload* to get the current settings.
- Select the devices whose I/O is to be exchanged with the EtherNet/IP scanner from the left column. Click the “>” button for each one to move it to the scan list. A screen displaying the I/O configuration for the device will be displayed; click *Ok*.
- Click the *Input* tab. A screen is displayed for mapping the input data.
- Select the device whose input data is to be mapped and click *AutoMap*. **This sets the format of the Input table of the BridgeWay. This is the format of the input data that will be sent to the EtherNet/IP scanner. See “I/O Mapping” on page 3-4.**
- Click the *Output* tab. A screen is displayed for mapping the output data.
- Select the device whose output data is to be mapped and click *AutoMap*. **This sets the format of the Output table of the BridgeWay. This is the format of the output data that will be sent to the BridgeWay from the EtherNet/IP scanner. See “I/O Mapping” on page 3-4**
- Select the *Scanlist* tab, and click the *Download* button to download the scanlist to the BridgeWay.
- The BridgeWay starts scanning as soon as it finds entries in its scanlist. However, in Idle mode, output data will not be sent to the devices.

Note: Automap is used in this example to for simplicity. In some cases, the user may wish to organize the I/O data in other ways. See the NetTool-DN manual for complete details on how to accomplish this.

Ethernet Network Configuration

Step 1: Connect the BridgeWay Module to Your Network

- Connect the Ethernet network cable to the RJ-45 fieldbus connector on the end of the BridgeWay.

Step 2: Configure the BridgeWay IP Address Using Arp/Ping

- Set all 8 switches on the IP Address Configuration DIP switch to the ON position.
- Turn the power ON.
- Open an MS-DOS™ window on the PC.
- Type 'arp -s <IP address> <MAC address>'

Substitute <MAC address> with the MAC address of your BridgeWay module. The MAC address is printed on a label on the back of the module. (Don't include the "<" or ">" characters shown above.) Separate every 2 digits of the MAC address with a dash (-).

Ask your network administrator for an unused IP number.

Substitute <IP address> with the IP number you wish to use for the module. (Don't include the "<" or ">" characters.)

Note: The BridgeWay's subnet mask and gateway address cannot be changed using this method.

Example:

```
arp -s 65.106.34.252 00-30-11-02-00-5e
```

- Type 'ping <IP address>'

Example:

```
ping 65.106.34.252
```

- You should see a message similar to below indicating a connection.

Example:

```
Reply from 65.106.34.252 Bytes=32 Time=271ms TTL=30
```

- Type 'arp -d <IP address>'

Example:

```
arp -d 65.106.34.252
```

The BridgeWay module will now adopt the IP address that was specified in the 'arp -s' command.

- Set all 8 switches on the IP Address Configuration DIP switch to the OFF position.

Using the Ethernet File System

Step 1: Browse the file system

- Open a web browser window on the PC.
- Type 'FTP://<IP address>' in the address field. (Substitute <IP address> with the IP address you are using for the module). (Don't include the "<" or ">" characters.)
- When prompted for a username enter "admin".
- When prompted for a password enter "admin".

You can now browse the file system. You should see subdirectories "/web", "/pswd" and "/user" and three files "/index.htm", "/ehtcfg.cfg", and "/telwel.cfg.

Step 2: Create some Files Using Telnet

- Click on the Windows 'Start' menu and select 'Run.'
- Type 'telnet <IP address>'. Substitute <IP address> with the IP address you are using for the module. (Don't include the "<" or ">" characters.)

Example:

```
telnet 65.106.35.252
```

- You will be prompted for a username, enter "admin", and a password, "admin".
- The Telnet client will be opened, and connected to the module. You can now browse the user file system using a command line interface.
- Type 'help' for a quick explanation on the available commands.
- Type 'md mydirectory'. You have now created a directory called 'mydirectory'.
- Type 'dir' to view the directory.
- To move inside the directory, type 'cd mydirectory'.
- Type 'mkfile myfile'. You have now created an empty file called 'myfile'. Let's put something in it.
- Type 'append myfile "Easy file handling!"'. You have now added the line 'Easy file handling!' to your new file.
- View the files contents by typing 'type myfile'

- Exit the telnet program.
- Open a web browser window on the PC.
- Type 'FTP://<IP address>' in the address field. Substitute <IP address> with the IP address you are using for the module. Don't include the "<" or ">" characters
- The directory and file that you created earlier using the Telnet application should appear. (If your files are not present, press 'F5' to update the window content)
- Don't close this window yet. If you are not in the root directory, make sure you are.

Step 3: Create and Upload a Web Page

- Open a text editor to create a text file.
- Type the following into the new file *including the "<" and ">" characters*:

```
<html>
<head>
<title>BridgeWay</title>
</head>
<body>
<center><h3>Hello world!</h3>Amazing.</center>
</body>
</html>
```

- Save the file using the filename 'hello.htm'.
- To upload the web page to the module, simply drag it to the web browser window that you opened earlier.

Note: This example requires Windows™ Internet Explorer 5.5 or higher, but it is possible to use any FTP client. However, the procedure may not be similar to this example.

Step 4: View a Web Page

- Open a web browser window on the PC.
- Type 'HTTP://<IP address>/hello.htm' in the URL field. (Substitute <IP address> with the IP address you are using for the module). Don't include the "<" or ">" characters.
- The web page that you downloaded in the previous step should be displayed in the browser.

DeviceNet Interface

Network Communications

The BridgeWay Ethernet to DeviceNet Gateway acts as a DeviceNet Master or a slave. The BridgeWay, as a master, can exchange I/O data with up to 63 nodes. The module can also act as a slave to another DeviceNet Master, exchanging the contents of its I/O tables with the master.

Configuration

The BridgeWay is configured using a DeviceNet configuration tool such as RSNetWorx for DeviceNet or NetTool-DN. The tool will access the module over the DeviceNet network. The BridgeWay supports a Scanner Configuration and Scan List object as the configuration interface over DeviceNet.

Automatic Baud Rate Detection

Depending on its configuration, the BridgeWay can set its DeviceNet baud rate automatically. If the autobaud option is enabled, the module will detect the current network baud rate and set its baud rate accordingly before joining the network. If the option is disabled, the module will join the network with the configured baud rate.

Slave Device Communication

The BridgeWay continuously attempts to establish connections with devices configured in the scan list (list of configured slaves). Once connections are established, the module performs all necessary steps to configure the required I/O messaging.

The BridgeWay provides explicit message proxy services for all group 2 only slaves. Once any Group 2 only devices are configured, the BridgeWay sends “keep alive” messages to the devices in addition to the I/O messages. This function prevents the explicit message connections between the BridgeWay and the slave from timing out. This eliminates the need to re-establish an explicit connection should the BridgeWay need to send configuration data or serve as a proxy.

Scan Cycles

The BridgeWay employs a scan cycle for producing poll and strobe I/O messages.

A scan cycle consists of the following:

- A bit-strobe output message (if devices are configured in the scan list are configured for bit-strobe).
- A poll command message for each device configured for polled I/O.
- A configurable delay before the next scan cycle.

The configurable delay is the Inter-Scan Delay (ISD). The ISD is a Scanner Configuration Object attribute. The delay begins when the last poll command message is transmitted and ends after the specified time has elapsed.

The BridgeWay also supports a background polling mechanism. A foreground to background polling ratio can be specified to allow polling of devices at certain scan cycle intervals.

I/O Message Types

The BridgeWay supports all I/O messaging types specified by the DeviceNet protocol. These include strobe, poll, COS, COS Unacknowledged, Cyclic, and Cyclic Unacknowledged I/O messages. I/O messaging and I/O parameters are configured using the DeviceNet configuration tool.

I/O Mapping

The contents and layout of the data in the I/O tables is defined during configuration of the scan list. The input and output data of each slave is configured, or mapped, to specific locations in the input and output tables.

Proxy for Group 2 Only Devices

The BridgeWay provides the capabilities necessary for being a Group 2 Only Client as defined for the Predefined Master/Slave Connection Set. Group 3 explicit messages destined for a group 2 only device that is configured as a slave to the BridgeWay will be intercepted and relayed to the slave.

Run/Idle Mode

The BridgeWay has two modes of operation, Run and Idle. In both modes the BridgeWay's DeviceNet master maintains communication with slave devices in its scan list.

In Run mode the BridgeWay sends output data to the slaves and receives input data. Since it is actively sending output data affecting slave device operation, the BridgeWay rejects attempts to alter its configuration and disrupt communications; it must first be put in Idle mode.

In Idle mode the BridgeWay still receives input data from the slaves but it does not send output data. In Idle mode the BridgeWay configuration can be changed.

The Run/Idle mode of the BridgeWay is controlled through the command registers at the front of the output data from the EtherNet/IP scanner (See "Output Assembly" on page 6-6). The module powers up in Idle mode.

If no EtherNet/IP I/O connection is active, the Run/Idle mode of the BridgeWay is set through an attribute of the Identity object. To change the Run/Idle mode, use a DeviceNet messaging tool and send the following message:

```
Service:      Set_Attribute_Single
Class:        1
Instance:     1
Attribute:    103 (67h)
Request Data: 00 for Idle, 01 for Run
```

Note: When the BridgeWay is reset or powered up, it begins operation in Idle mode.

Automatic Device Recovery (ADR)

This is a feature of the DeviceNet master which allows a slave node that has dropped off the network (Fault, power loss, etc.) to be replaced with another device of the same type. There are 2 parts to ADR, Address Recovery, and Configuration Recovery.

Note: ADR is only available when using Rockwell Software's RSNetWorx for DeviceNet as the configuration tool.

Address Recovery

Address Recovery is responsible for automatically setting a new device's address to that of a slave that has lost communications. The steps followed by ADR are:

1. When the master detects loss of a slave, it begins to monitor for a device at MAC ID 63.
2. An identical device is added to the network at MAC ID 63.
3. The master verifies that the new device at 63 is exactly the same kind as the slave that was lost.
4. The master changes the new device's MAC ID from 63 to that of the lost slave.

Configuration Recovery

Configuration Recovery is responsible for setting the configuration of a slave device to the configuration that is stored in the BridgeWay. The slave's configuration is stored in the BridgeWay's non-volatile memory. Whenever the BridgeWay establishes communication with the slave device, the configuration is downloaded to the slave.

Configuration recovery serves 2 purposes. 1. If a new device is added to the network to replace a faulted slave, after Address Recovery is completed, Configuration Recovery will configure the new device. 2. Configuration Recovery guarantees that the slave devices will always run the same configuration.

Interaction with I/O Tables

The DeviceNet interface in the BridgeWay accesses the I/O tables as slave I/O connections are processed by the DeviceNet master; there is no buffering or timed updates of the I/O within the module. Safeguards are in place to ensure data integrity by prohibiting simultaneous access by the Ethernet and DeviceNet interfaces. There is no synchronization between the 2 network interfaces.

When an I/O connection with a slave requires that output data be sent to the slave, it will be read from the Output table. The data read is what was placed there by the last write to the Output table by the Ethernet interface.

When input data is received on a slave's I/O connection, it is copied to the Input table. This data is available to be read by the Ethernet interface and sent to the EtherNet/IP scanner on the next data exchange.

EtherNet/IP Interface

EtherNet/IP is based on the Control and Information protocol (CIP) which is also the application layer for DeviceNet to exchange data between nodes.

Product Features

The BridgeWay contains EtherNet/IP Adapter Class functionality. Being an I/O Server it can respond to requests for I/O messages but it does not generate such requests. A Scanner device would generate these requests. The BridgeWay supports Message Server and Message Client, functionality. This means it can act as a target and originator for messaging.

CIP Objects

CIP makes use of abstract object modeling to describe the communications of a product. Objects are well defined subsets of a device's functionality. They have functions that they perform called Services and data variables called Attributes. If more than one copy of an object is needed each copy is called an Instance. We have implemented the same objects as other modules that are based on the CIP protocol.

CIP Messaging

Two types of messaging are used. The regular or repeated transport of a specific set of data items is known as Implicit Messaging. Both parties agree in advance and allocate resources for the data being transported. The connection ID within the Implicit message defines the meaning of the data and establishes the transport rate and transport class. The term Implicit Messaging can be interchangeable with the term I/O Messaging.

A one time transport of a data item where both parties do not know in advance of what's being transferred is known as Explicit Messaging. Explicit messaging is used for point to point type messaging. The protocol of the message data describes (addresses) the data to be transferred. In object modeling the address is in terms of class number, instance number, and attribute number.

Messages can be sent as Connected or Unconnected. With Connected Messaging device resources are reserved in advance of data transfer and are dedicated and always available. Unconnected messaging provides a means for a device to send a request without establishing a connection prior to data transfer. This is accomplished through the UCMM or UnConnected Message Manager of the EtherNet/IP protocol. With UCMM all objects are accessible.

All Explicit Messages have message data defined in a format called the Message Router Protocol Data Unit (MR_PDU). There are Requests and Responses.

The MR_PDU Request format includes a Service code, Path Size, Path, and data, if any for the Service. The Path is an encoded series of bytes or Segments describing the location of the data item involved in the transfer. The Logical Segment is most often used. It describes the Class, Instance, and Attribute of the data. The Path may also include a Port Segment. The Port segment describes a path or way to another network. There are two ports on the BridgeWay, one for EtherNet/IP and one for DeviceNet.

I/O Messaging

The BridgeWay allows an EtherNet/IP scanner access to the I/O data of DeviceNet slaves. The data produced by the DeviceNet slaves is collected in the Input Table (IN) of the BridgeWay and becomes the EtherNet/IP Input I/O to the EtherNet/IP scanner. EtherNet/IP Output data from the scanner is stored in the BridgeWay's Output Table (OUT) and sent to the DeviceNet slaves which consumes it.

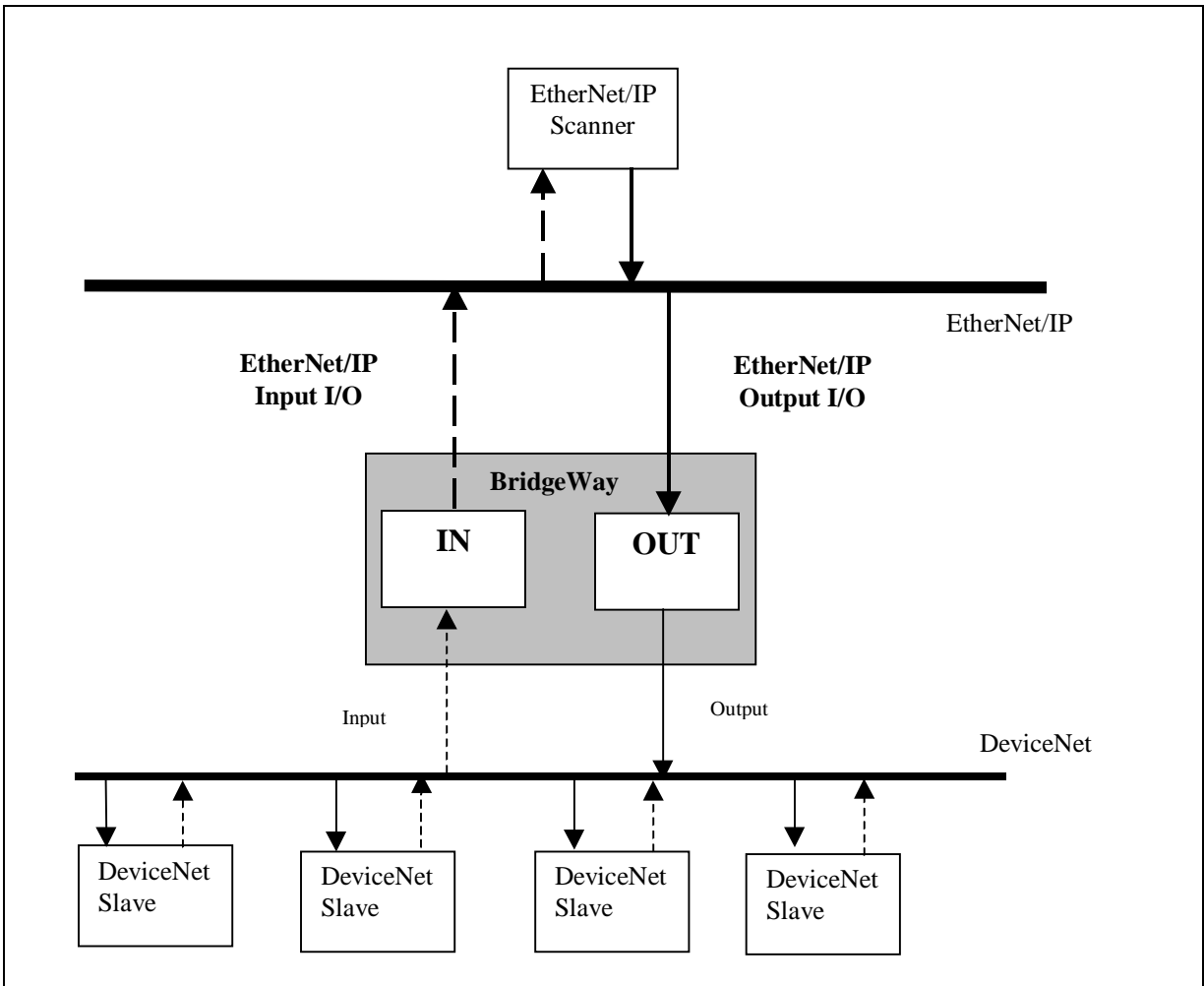


Figure 6-1 BridgeWay I/O Transfer

Assembly Objects and Connections

There are 3 Assembly Object instances accessible from EtherNet/IP: input, output and status. The input and output assemblies are linked to the input and output data tables. The status assembly provides current status information about the Bridge-Way.

The assembly instances associated with these 3 assemblies are listed below.

Assembly Instance	Description	Size in Bytes
100	Input	508 max
101	Status	128
150	Output	508 max

Table 6-1 EtherNet/IP Assembly Instances

Connection Points

Class 1 connections can be established to these assemblies using the connection points listed in Table 6-2.

Conn Point	Description	Size in Bytes	Use
3	Heartbeat	0	Output connection point for input-only connections.
100	Input	0-508	Input connection point.
101	Status	128	Input connection point.
150	Output	0-508	Output connection point.

Table 6-2 EtherNet/IP Connection Points

Connection sizes, when connecting to the input and output assemblies can be set according to the size of the I/O data tables configured in the DeviceNet scanner and the status and command words in the I/O assemblies. (see the assembly formats below) If a connection is created with a size larger than that configured in the DeviceNet scanner, the extra data will be filled with 0.

Input Assembly

The input assembly contains a 32-bit status register followed by the data in the BridgeWay's input data table.

Byte Offset	Size in Bytes	Description
0	4	Status register.
4	Up to 504	Input data.

Table 6-3 Input Assembly Format

The input data format and content is determined by the DeviceNet scanner configuration. The data appears in the table as it is mapped from the DeviceNet input connections. The input data in the assembly is 504 bytes long; however, only the size of the configured input data table will be used, the remaining space will be filled with 0.

The status register is a bit string with the following bit definitions.

Bit	Description
0	BridgeWay is in Run mode. (Cleared if in Idle mode.)
1	BridgeWay is faulted.
2	DeviceNet network interface is disabled.
3	Communication has failed with at least 1 DeviceNet slave.
4	At least 1 DeviceNet slave is an incorrect device type.
5	DeviceNet network interface is bus-off.
6	Duplicate MAC ID error.
7	No DeviceNet power.
8-31	Not used.

Table 6-4 Input Status Register Bit Definitions

Output Assembly

The output assembly contains a 32-bit run/idle register and a 32-bit command register followed by the data in the BridgeWay's output data table.

Byte Offset	Size in Bytes	Description
0	4	Run/Idle register.
4	4	Command register.
8	Up to 500	Output data.

Table 6-5 Output Assembly Format

The output data format and content is determined by the DeviceNet scanner configuration. The data appears in the table as it is mapped to the DeviceNet output connections. The output data in the assembly is 500 bytes long; however, only the size configured for the output data table will be used, the remaining space will be ignored.

The Run/Idle register is a bit string with the following bit definitions.

Bit	Description
0	System Run Mode. Used in conjunction with the Local Run Mode bit in the Command register to determine the run mode of the BridgeWay. Both bits must be set for the BridgeWay to be in Run mode; otherwise the module will be in Idle mode.
1-31	Not used.

Table 6-6 Output Run/Idle Register Bit Definitions

The Command register is a bit string with the following bit definitions.

Bit	Description
0	Local Run Mode. Used in conjunction with the System Run Mode bit in the Command register to determine the run mode of the BridgeWay. Both bits must be set for the BridgeWay to be in Run mode; otherwise the module will be in Idle mode.
1	Fault. Sets a fault condition in the BridgeWay.
2	Disable DeviceNet network.
3	Not used.
4	Reset the BridgeWay module.
5-31	Not used.

Table 6-7 Output Command Register Bit Definitions

Status Assembly

The status assembly is a collection of status and diagnostic information for the BridgeWay DeviceNet interface. The information in the assembly is updated once a second.

Note: All information in the status assembly is stored in little endian format. The least significant byte of multi-byte values is stored first.

Byte Offset	Size in Bytes	Data Type	Name	Description
0	4	UDINT	Scan Counter	The number of DeviceNet I/O scans that have taken place since the BridgeWay was powered up.
4	8	64-bit Bit-string	Faulted Node Table	Indicates which DeviceNet slaves are faulted. Each bit represents the status of the slave at the corresponding MAC ID.
12	8	64-bit Bit-string	Auto Verify Error Table	Indicates which DeviceNet slaves are the incorrect device type. Each bit represents the status of the slave at the corresponding MAC ID.
20	8	64-bit Bit-string	Idle Node Table	Indicates which DeviceNet slaves are in Idle mode. Each bit represents the status of the slave at the corresponding MAC ID.
28	8	64-bit Bit-string	Active Node Table	Indicates which DeviceNet nodes are configured in the BridgeWay's scan list. Each bit represents the status of the slave at the corresponding MAC ID.

Table 6-8 Status Assembly Format

Byte Offset	Size in Bytes	Data Type	Name	Description
36	4	ASCII[4]	Status Display	Mimics a 4-character alpha-numeric display. If there are no faults, the display indicates the BridgeWay MAC ID and its Run/Idle status. If there are faults, the display will scroll through the MAC IDs of the faulted nodes and display the error code associated with each. See Table 10-6, "Node Status Codes," on page 10-5 for a list of error codes.
40	1	USINT	BridgeWay MAC ID	The DeviceNet MAC ID of the BridgeWay.
41	1	USINT	Scanner Status	The current status of the DeviceNet scanner. See Table 10-6, "Node Status Codes," on page 10-5 for a list of status and error codes.
42	1	USINT	Scrolling MAC ID	The scrolling address and status fields scroll through the address and status of all DeviceNet slaves that are faulted. This scrolling includes the BridgeWay scanner itself. If there are no faulted nodes, both the scrolling address and status are set to 0. The scrolling fields change once a second.
43	1	USINT	Scrolling Status	
44	20	USINT[20]	Reserved	
64	64	USINT[64]	Node Status Table	The current status of each DeviceNet slave node. Each array element is the status of the node at the corresponding MAC ID. If a node is not configured in the BridgeWay scan list, the status value will be set to 0. The BridgeWay scanner status appears at the entry associated with the BridgeWay MAC ID. See Table 10-6, "Node Status Codes," on page 10-5 for a list of status codes.

Table 6-8 Status Assembly Format (Continued)

Notes About Using ControlLogix I/O Connections

When configuring I/O connections between a Rockwell Automation ControlLogix EtherNet/IP scanner and the BridgeWay, the Generic EtherNet/IP device type should be used.

The Run/Idle register is automatically inserted at the front of the output data and the application has no control over its use. The System Run Mode bit is set according to the Run/Program mode of the controller. The ControlLogix application sees byte 4 as the beginning of the output assembly.

The output size must be set to 4 bytes less than the actual assembly size to account for the Run/Idle header that is automatically added by the controller.

The maximum connection size that ControlLogix allows is 500 bytes. Hence the total 508 bytes of the input and output areas cannot be accessed when using ControlLogix.

CIP Bridging

The EtherNet/IP protocol provides bridging capabilities to allow a device on the EtherNet/IP network to access a device on the DeviceNet network through Explicit Messaging. The BridgeWay Ethernet to DeviceNet Gateway allows a device on EtherNet/IP to send an Explicit Message to a device on DeviceNet and receive its response. In this way the device on EtherNet/IP can directly access the objects of any DeviceNet device to configure or access data.

To send an Explicit Message to a DeviceNet device, the Unconnected Send or Forward Open services of the Connection Manager Object are used. The the MAC ID of the destination DeviceNet node along with a network port address must be used in the Unconnected Send and Forward Open service. Refer to Volume 1, Chapter 10 of the EtherNet/IP specification for further information on CIP Bridging.

The BridgeWay supports multi-hop bridged paths. If a routing path routes the message through the local DeviceNet network to another network, via another bridge, the BridgeWay will correctly route the message to the next bridge using an Unconnected Send service over DeviceNet.

Note: The BridgeWay does not support message routing from DeviceNet to EtherNet/IP.

Port Addresses

Port Address	Network
2	EtherNet/IP
3	DeviceNet

Table 6-9 Port Addresses

Modbus/TCP Interface

The BridgeWay supports Modbus/TCP commands. The implementation of the Modbus/TCP server is done according to the Modbus/TCP specification 1.0. All commands according to class 0 and class 1 are implemented and a subset of the class 2 commands.

The module can handle 8 simultaneous connections.

Supported Commands

The following Modbus/TCP commands are supported by the BridgeWay.

Function Code	Function Name	Class	Affects	Address Method
1	Read Coils	1	IN/OUT	Bit
2	Read Input Discrete	1	IN/OUT	Bit
3	Read Multiple Registers	0	IN/OUT	Word
4	Read Input Registers	1	IN/OUT	Word
5	Write Coil	1	OUT	Bit
6	Write Single Register	1	OUT	Word
15	Force Multiple Coils	2	OUT	Bit
16	Force Multiple Registers	0	OUT	Word
22	Mask Write Registers	2	OUT	
23	Read/Write Registers	2	IN/OUT	

Table 7-1 Modbus Commands

Supported Exception Codes

An exception code is returned in the response when the BridgeWay is unable to service the Modbus request that was received. The following exception codes will be used by the BridgeWay.

Exception Code	Name	Description
01	Illegal Function	The module does not support the function code in the query
02	Illegal Data address	The data address received in the query is outside the initialized memory area
03	Illegal Data Value	The data in the request is illegal

Table 7-2 Exception Codes

Modbus/TCP Addressing

The BridgeWay's Input (IN) and Output (OUT) areas are set to a maximum size of 508 bytes each. The Status assembly area is 128 bytes. When accessing these areas, with Modbus commands, the addressing is done according to the following tables.

Word	Bit Address									
000h	0000h	0001h	0002h	0003h	0004h	0005h	0006h	...	000Eh	000Fh
001h	0010h	0011h	0012h	0013h	0014h	0015h	0016h	...	001Eh	001Fh
....										
0FDh	0FD0h	0FD1h	0FD2h	0FD3h	0FD4h	0FD5h	0FD6h	...	0FDEh	0FDFh

Table 7-3 Input Addressing

Word	Bit Address									
400h	4000h	4001h	4002h	4003h	4004h	4005h	4006h	...	400Eh	400Fh
401h	4010h	4111h	4012h	4013h	4014h	4015h	4016h	...	401Eh	401Fh
...										
4FDh	4FD0h	4FD1h	4FD2h	4FD3h	4FD4h	4FD5h	4FD6h	...	4FDEh	4FDFh

Table 7-4 Output Addressing

Word	Bit Address									
100h	1000h	1001h	1002h	1003h	1004h	1005h	1006h	...	100Eh	100Fh
101h	1010h	1111h	1012h	1013h	1014h	1015h	1016h	...	101Eh	101Fh
...										
13Fh	13F0h	13F1h	13F2h	13F3h	13F4h	13F5h	13F6h	...	13FEh	13FFh

Table 7-5 Status Addressing

Bit Addressing Examples

- To reference the first bit of the Input Table use address 0000h.
- To reference the 15th bit of the Input Table use address 000Eh (14 decimal).
- To reference the first bit of the Output Table use address 4000h (16384 decimal).
- To reference the 100th bit of the Output Table use address 4063h (16483 decimal).

Word Addressing Examples

- To reference the first word of the Input Table use address 000h.
- To reference the 10th word of the Input Table use address 009h.
- To reference the first word of the Output Table use address 400h (1024 decimal).
- To reference the 100th word of the Output Table use address 463h (1123 decimal).

I/O Data Content

The content of the input and output data is the same as the content of the I/O assemblies used in EtherNet/IP (See section “Input Assembly” on page 6-5, section “Output Assembly” on page 6-6, and section “Status Assembly” on page 6-8.)

I/O Data Format

The BridgeWay transfers I/O data between Modbus/TCP and DeviceNet without regard to data content or format. Due to this, the user is responsible for making sure that the devices on either network understand the format of the data.

DeviceNet is a little endian protocol; values are transmitted least significant byte first. Hence, all data in the I/O tables is assumed, by the DeviceNet nodes, to be stored as little endian.

Care should be taken to make sure that the Modbus/TCP master handles input data and transmits output data in a format acceptable to the target DeviceNet devices (least significant byte first).

File System

The files system is a fixed-size storage area with a hierarchical directory structure. Any data, user or application can be stored in files within the file system. Files can be grouped in directories for readability.

The file system features two security levels. Depending on security level, different users can have access to different files and directories. The file system is accessible via FTP, Telnet, and HTTP.

File System Conventions

Case Sensitivity

The file system is case sensitive. This means that the file 'pyramid.txt' is not identical to the file 'Pyramid.TXT'.

Filename / Pathname length

Filenames can be a maximum of 48 characters long. Pathnames can be 256 characters in total, filename included.

File Size

The file size is not restricted. Naturally, a file cannot be larger than the available space, see below.

Free space

There is approximately 1.4MB available for user files.

Security

The file system features two security levels: Administration and Normal. In Administration mode, the user has full access to the file system through FTP and Telnet. This enables the user to access areas of the file system that are restricted or inaccessible in Normal mode.

Normal mode is recommended for normal operation, so that web pages and other settings are protected. Administration mode is intended for product development.

The security level can be set individually for each login.

Files within the file system can be protected from web access through username/password authorization, see “Password Files” on page 8-11 and “web_accs.cfg” on page 8-12. It is also possible to configure which IP addresses and what protocols that are allowed to connect to the module, see “ip_accs.cfg” on page 8-8.

Normal mode

The BridgeWay contains a default admin password (“ad_pswd.cfg”) file so when the module is first powered it operates in normal mode (See “Default User Accounts” on page 9-1). If a valid admin password file (see “Password Files” on page 8-11) is not found, the module will default operations to Administration mode.

In normal mode the FTP and Telnet services are only enabled if there is a subdirectory called “\user”. When a normal user connects via FTP or Telnet, this directory will be their root directory. The user will not be able to access files outside this directory and its subdirectories (administrator files).

In normal mode the BridgeWay provides user/password protection for FTP and Telnet with a file called “sys_pswd.cfg” in the directory “\user\pswd”. Files in this directory cannot be accessed by a web browser. A default “sys_pswd.cfg” file is provided. The default file provides a guest user access to FTP and Telnet. This user has username “guest” and password “guest”.

To prevent unauthorized access this should be changed as soon as possible. This can be done by changing the username or password. The Administrator can access this file to add or remove users or change passwords.

If a user logs into Telnet or FTP using a username/password combination found in the admin password file (see “Password Files” on page 8-11) he will gain access to the entire system.

Administration Mode

At power up the BridgeWay contains a default admin password file (See “Default User Accounts” on page 9-1.) A user can login as an administrator by using the username “admin” and password “admin”.

To prevent unauthorized access this should be changed as soon as possible. This can be done by changing the username or password.

If no admin password file (see “Password Files” on page 8-11) is found the module operates in Administration mode. The user has full access to the file system via FTP or Telnet. *No login is needed for Telnet, and the FTP server accepts any username/password combination.*

Structure

The figure below illustrates the structure of the file system, where the system files are located, and which areas that can be accessed by normal/admin users.

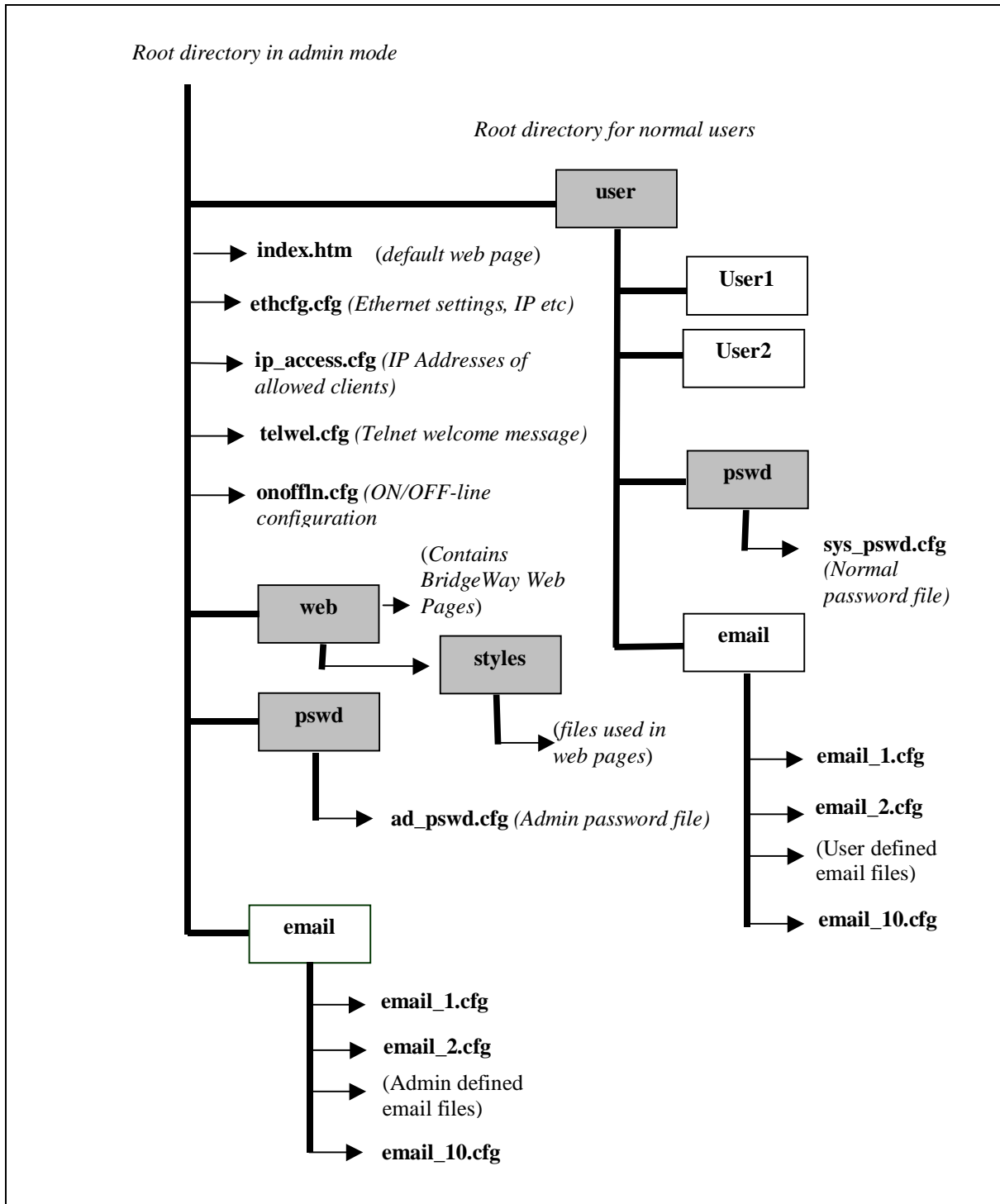


Figure 8-1 File System Directory Structure

Default Files

The following directories are already created on the BridgeWay when first powered;

- \pswd,
- \user,
- \web,
- \web\styles,
- \user\pswd.

The following files are also on the BridgeWay;

- \pswd\ad_pswd.cfg,
- \telwel.cfg,
- \ethcfg.cfg,
- \index.htm
- \user\pswd\sys_pswd.cfg.

These files can be edited as needed. Each file is discussed below. The BridgeWay power must be recycled for any changes to take effect.

Virtual File System

The module contains a virtual file system, a set of files used to build the default configuration web page. These are hidden files. The files can be replaced or disabled, but not erased. A file with the same name in the file system replaces the file in the virtual file system until it is removed.

The virtual file system contains the following files:

- index.htm
- config.htm
- configform.htm
- store.htm
- logo.gif
- configuration.gif
- boarder.bg.gif
- boarder_m_bg.gif

The BridgeWay contains an “index.htm” file replacing its virtual file counterpart to provide a link to the BridgeWay’s Home page.

System Files

The module uses these files for configuration purposes. The system files are ASCII files and can be edited with any text editor. Depending on security settings, the files may be inaccessible for normal users.

Note: These files shall not be used to store any user or application data.

Configuration Files

'ethcfg.cfg'

This file contains the network configuration and is read by the module at start up.

The settings in this file are affected by SSI commands.

The components and format of the file is shown below:

[IP address]	IP address
10.10.12.212	
[Subnet mask]	Subnet mask
255.255.255.0	
[Gateway address]	Gateway address
0.0.0.0	
[SMTP address]	SMTP address – This must be configured in order to send emails
0.0.0.0	
[DHCP/BOOTP]	DHCP/BootP 'ON'-Enabled, 'OFF'-Disabled
OFF	
[Speed]	Speed 'Auto' Autonegotiation will be used
Auto	'100' Forces the module at 100mbits
	'10' Forces the module at 10mbits
[Duplex]	Duplex 'Auto' Autonegotiation will be used
Auto	'Full' Forces the module to operate only at full duplex.
	'Half' Forces the module to operate only at half duplex.

The contents of this file can be redirected by placing the line '[File path]' on the first row, and a file path on the second.

Example:

```
[File path]
\user\eth_settings.cfg
```

In this example, the settings described above will be loaded from the file 'user\eth_settings.cfg'.

This permits normal users to access the network configuration settings.

Note: The module needs to be restarted for changes in this file to have affect.

'ip_accs.cfg'

It is possible to configure which IP addresses and what protocols that are allowed to connect to the module. This information is stored in the file '\ip_accs.cfg'. The file contains one or several of the headers below.

```
[Web]
[FTP]
[Telnet]
[Modbus/TCP]
[Ethernet/IP]
[All]
```

Under each header the allowed IP addresses are written. The wildcard '*' can be used to allow series of IP addresses. If a protocol header is not given, the system will use the configuration set under the header 'All'. If the 'All' header is not given, the protocol will not accept any connections.

Example:

```
[Web]
10.10.12.*
10.10.13.*
[FTP]
10.10.12.*
[Telnet]
10.10.12.*
[All]
*.*.*.*
```

The above example will allow all IP addresses beginning with 10.10.12 to access all protocols in the module. IP numbers beginning with 10.10.13 will not be able to access the FTP and Telnet servers. The Modbus/TCP and EtherNet/IP servers will accept connections from any IP address. The contents of this file can be redirected by placing the line '[File path]' on the first row, and a file path on the second.

Example:

```
[File path]
\my_settings\ip_access_rights.cfg
```

In this example, the settings described above will be loaded from the file '\my_settings\ip_access_rights.cfg'.

Note: The module has to be restarted in order for any changes in this file to have affect.

'onoffin.cfg'

The ON/OFF line functionality is by default configured to be triggered by the Link Status. It can however be configured to be triggered by for example a Modbus command. This is done by creating the file '\onoffln.cfg'.

The components and format of the file is shown below:

```
[ON/OFF-line trigger]  ON/OFF-line trigger ['Link'|'Modbus']
Modbus                Values "Link" and "Modbus"

[Timeout]             Timeout value. A value 10 = 1000ms
10

[Commands]            Commands (this line is optional)
3, 16, 23             Selects what Modbus commands must be
                       received during the timeout period.
                       If the keyword 'ALL' is given, the ON/OFF line
                       functionality will trigger on all Modbus commands
```

The [Timeout] and [Commands] heading shall only be given if the ON/OFF-line trigger value is set to Modbus.

The contents of this file can be redirected by placing the line '[File Path]' on the first row, and a file path on the second.

Example:

```
[File path]
\my_settings\on-off-line_configuration.cfg
```

In this example, the settings described above will be loaded from the file '\my_settings\on-off-line_configuration.cfg'.

Note: The module has to be restarted in order for any changes in this file to have affect.

Password Files

'sys_pswd.cfg & ad_pswd.cfg'

These files contain user / password information for users in normal mode ('sys_pswd.cfg') and administration mode ('ad_pswd.cfg'). The files shall be located in '\user\pswd' and '\pswd' respectively.

These directories are protected from web browser access.

The file format is the following:

```
User1:password1
User2:password2
...
UserN:passwordN
```

Example:

```
JohnQ:Password
```

In this example, the username is 'JohnQ', and the password is 'Password'.

If no ':' is present, the password will be equal to the username.

Example:

```
BillH
```

In this example, both username and password will be 'BillH'.

'web_accs.cfg'

Files within the file system can be protected from web access through username/password protection. To put username/password protection to files, a file called 'web_accs.cfg' must be located in the same directory as the files to protect. If this file is available, all files within that directory and its subdirectories will be protected. Multiples of these password files may be present in the system, giving different users access to different files and directories.

The file format is the same as for the 'ad_pswd.cfg' and 'sys_pswd.cfg' files, except that the optional parameter 'Auth Name' can be added. The value of this parameter will be presented in the login window as the "Realm". If it is not given, the requested file/pathname will be presented instead.

Example:

```
User:Password
[Auth Name]
(Message goes here)
```

The contents of this file can be redirected by placing the line '[File path]' on the first row, followed by a list of password files.

Example:

```
[File path]
\user\pswd\my_passwords\web_pswd.cfg
\user\pswd\my_passwords\more_pswd.cfg
```

In this example, the accepted user/passwords will be loaded from the files '\user\pswd\my_passwords\web_pswd.cfg' and '\user\pswd\my_passwords\more_pswd.cfg'

If any errors in the format of these files is detected the user/password protection will be ignored.

Other Files

'telwel.cfg'

The default Telnet welcome message can be changed by creating this file. It shall contain the new welcome message in ASCII form.

The contents of this file can be redirected by placing the line '[File path]' on the first row, and a file path on the second.

Example:

```
[File path]
```

```
\my_settings\telnet_welcome_message.txt
```

In this example, the welcome message will be loaded from the file '\my_settings\telnet_welcome_message.txt'.

Email files (email_1.cfg,email_2.cfg to email_10.cfg)

These files contain predefined email messages and information on how and when to send them. It is possible to have a maximum of 10 admin defined email files and 10 user defined email files. The files must be named 'email_1.cfg'... 'email_10.cfg', and placed in the folders '\email' and '\user\email' respectively. If the SMTP server is not configured the email will not be sent (See "'ethcfg.cfg'" on page 8-7). If the SMTP server requires username/password authorization the email will not be sent. (There will not be any indication to the user that the system failed to send the email.)

The file must have the following format.

[Register]

Area, Offset, Type

[Register Match]

Match Value, Mask, Match Operand

[To]

Recipient(s)

[From]

Sender

[Subject]

Subject Line

[Headers]

Extra Headers

[Message]

Message Body

Parameter	Values	Description
Area	IN OUT	Source area in Input/Output
Offset	a hexadecimal (0xN) or decimal value	Source Offset in Input/Output
Type	byte word long	Source data type
Match Value	a hexadecimal (0xN) or decimal value	Value to compare with source value.
Mask	a hexadecimal (0xN) or decimal value	A logical “AND” is performed on the source data using this Mask before comparing with the Match Value
Match Operand	< = >	How the data is compared with the Match Value
Recipient(s)	text (colon separated)	Destination email address(es)
Sender	text	Sender email address
Subject	text (only 1 line)	email subject
Extra Headers	text	Optional. It may be useful to send HTML email
Message Body	text	Message

Table 8-1 Email Parameters

Example

[Register]

IN, 0x0003, byte

A byte is read from the Input area at location 3.

[Register match]

0x20, 0x7F, >

Mask Input byte with 0x7F,
if result greater than 0x20 send email.

[To]

support@your_company.com

[From]

YourDevice@your_network.com

[Subject]

Status

[Message]

Data out of range

BridgeWay Web Page Files

The BridgeWay contains several web pages in HTML files to allow changing the default configuration settings and displaying DeviceNet status. Information displayed on these pages are updated every 2.5 seconds.

NOTE: These web pages require that your browser support Java. Recent versions of Microsoft Internet Explorer do not support Java by default. The Microsoft Virtual Machine for Internet Explorer may be downloaded from Microsoft's web site at <http://v4.windowsupdate.microsoft.com/en/default.asp>.

The files associated with the web pages are in the \web directory and corresponding support files are located in \web\styles.

'\index.htm' (Re-Direct Page)

The '\index.htm' file replaces the virtual file index.htm and provides an entry screen with a link to the BridgeWay's Home page in the \web subdirectory.

'\web\index.htm' (Home Page)

The '\web\index.htm' file provides an information screen with links to other web pages in the \web subdirectory. This is considered the BridgeWay's Home page.

'\web\BW_Settings.htm' (Settings)

Click on the "Settings" link to display a web page allowing ethernet address settings to be re-configured including the subnet mask, gateway address, IP address, and DHCP enable.

'\web\BW_NodeActive.htm' (Active Nodes)

Click on the "Active Nodes" link to display a web page providing a status table of the possible 63 DeviceNet nodes with an indication of whether the node is configured in the BridgeWay's scanlist (Active). Nodes in the scanlist will have the word "Active" next to it.

'\web\BW_NodeIdle.htm' (Idle Nodes)

Click on the “Idle Nodes” link to display a web page providing a status table of the possible 63 DeviceNet nodes with an indication of whether each node is idle or in a configuration state. This is valid for nodes configured in the BridgeWay’s scanlist. Each MAC ID will have the word “Idle” or a dash (-) next to it.

'\web\BW_NodeFaulted.htm' (Faulted Nodes)

Click on the “Faulted Nodes” link to display a web page providing a status table of the possible 63 DeviceNet nodes with an indication for each node in the BridgeWay’s scanlist of whether the BridgeWay and node are communicating.

'\web\AutoVerifyTbl.htm' (Invalid Nodes)

Click on the “Invalid Nodes” link to display a web page providing a status table of the possible 63 DeviceNet nodes with an indication for each node in the BridgeWay’s scanlist of whether the node has failed auto-verification (the device’s type is incorrect).

'\web\BW_NodeStatus.htm' (Node Status)

Click on the “Faulted Nodes” link to display a web page providing a status table of the possible 63 DeviceNet nodes with additional status information for each node in the BridgeWay’s scanlist.

IT Functionality

The module features common IT functionality such as an HTTP server, FTP server, an Email client, and a Telnet server. This provides easy file management and the possibility to customize the module to provide user-friendly access to parameters.

Also, the module can be configured to report selected information via Email using the Email client.

Default User Accounts

The BridgeWay contains two user accounts on initial power up. One account is for Administration mode (username=admin, password=admin). One account is for a normal user (username=guest, password=guest).

To prevent unauthorized access this should be changed as soon as possible. This can be done by changing the username or password. The Administrator can access the Password files to add or remove users or change passwords. (See files “sys_pswd.cfg & ad_pswd.cfg” on page 8-11).

The FTP Server

It is possible to upload/download files to/from the file system using a standard FTP client. Depending on security settings, different parts of the file system can be accessed by the user:

Normal Mode / Normal User

The user must login using a valid username/password combination. The root directory will be the ‘user’ directory unless the user has admin permission, see below.

Administration Mode / Admin User

The admin user has unrestricted access to the file system.

The Telnet Server

Through a Telnet client, the user can access the file system using a command line interface similar to MS-DOS™.

Normal Mode / Normal User

The user must login using a valid username/password combination. The root directory will be the ‘user’ directory unless the user has admin permission, see below.

Administration Mode / Admin User

The user must supply a valid admin user/password combination either during login or by using the command ‘admin’ in order to get admin permission.

The admin user has full access to the file system. The root directory will be “\” and no files or folders will be hidden.

General Commands

admin

Usage:

```
admin
```

Provided that the user can supply a valid admin username/password combination, this command enables admin access in normal mode. This command has no affect in administration mode.

help

Usage:

```
help [general|diagnostic|filesystem]
```

General commands:

```
help - Help with menus
```

```
version - Display version information
```

```
exit - Exit station program
```

Also try 'help general|diagnostic|filesystem'

version

Usage:

```
version
```

This command will display version information, serial number and MAC Address of the module.

exit

Usage:

```
exit
```

This command closes the Telnet session.

Diagnostic Commands

The following commands can be viewed by the command 'help diagnostic'

arps

Usage:

arps

Display ARP stats and table

iface

Usage:

iface

Display net interface stats

sockets

Usage:

sockets

Display socket list

routes

Usage:

routes

Display IP route table

File System Operations

For commands where filenames, directory names or paths shall be given as an argument the names can be written directly or within quotes. For names including spaces the filenames must be surrounded by quotes. It is also possible to use relative pathnames using '.', '\', and '..'.

dir

Usage:

```
dir [path]
```

Lists the contents of a directory. If no path is given, the content of the current directory is listed.

md

Usage:

```
md [[path]][directory name]]
```

Creates a directory. If no path is given, the directory is created in the current directory.

rd

Usage:

```
rd [[path]][directory name]]
```

Removes a directory. The directory can only be removed if it is empty.

cd

Usage:

```
cd [path]
```

Changes current directory.

format

Usage:

format

Formats the file system. This is a privileged command and can only be called in administration mode.

del

Usage:

del [[path][filename]]

Deletes a file.

ren

Usage:

ren [[path][old name]] [[path][new name]]

Renames a file or directory.

move

Usage:

move [[source path][source file]] [[destination path]]

This command moves a file or directory from the source location to a specified destination.

copy

Usage:

copy [[source path][source file]] [[destination path]]

This command creates a copy of the source file at a specified location.

type

Usage:

```
type [[path]][filename]]
```

Types the contents of a file.

mkfile

Usage:

```
mkfile [[path]][filename]]
```

Creates an empty file.

append

Usage:

```
append [[path]][filename]] ["The line to append"]
```

Appends a line to a file.

HTTP Server

The module features a complete HTTP (web) server with Server Side Includes (SSI) functionality. Server Side Includes are commands to the web server embedded in the HTML code. When the web server encounters the commands the results of the command are inserted in the web page. SSI commands allow easy access to the IN and OUT areas of the BridgeWay module. It is possible to upload web pages to the module, giving access to data in the memory of the module using a customizable interface.

Virtual Files

The module contains a set of virtual files that can be used when building a web page for configuration of network parameters. These virtual files can be overwritten (not erased) by placing files with the same name in the root of the file system.

By using this feature it is for example possible replace a logo by uploading a new logo named '\logo.gif'. It is also possible to make links from a web page to the virtual configuration page. In that case the link shall point to '\config.htm'.

The available virtual files are:

index.htm	- Shows the contents of config.htm
config.htm	- Configuration frame page
configform.htm	- Configuration form page
store.htm	- Configuration store page
logo.gif	- Logo
configuration.gif	- Configuration picture
boarder_bg.gif	- Picture
boarder_m_bg.gif	- Picture

Security

All files except files in the directories “\user\pswd\”, “\pswd\” and files named ‘web_accs.cfg’ can be viewed by default. Other directories can be protected by placing a file called ‘web_accs.cfg’(see “Password Files” on page 8-11) in the directory to protect. The file contains a list of users that are allowed to browse that directory.

Also, it is possible to configure which IP addresses are allowed to connect to the web sever, “ip_accs.cfg” on page 8-8.

SSI Functionality

SSI functionality makes it possible to make web pages interact such as changing the data in the OUT area of the module. It is also possible to include SSI functions in emails (see “SSI Includes in emails” on page 9-23). The following are available SSI functions.

Ethernet Address Display Functions**DisplayIP**

This function returns the currently used IP address.

Syntax:

```
<?--#exec cmd_argument='DisplayIP'-->
```

DisplaySubnet

This function returns the currently used Subnet mask.

Syntax:

```
<?--#exec cmd:argument='DisplaySubnet'-->
```

DisplayGateway

This function returns the currently used Gateway address.

Syntax:

```
<!--#exec cmd_argument='DisplayGateway'-->
```

DisplayDhcpState

This function returns whether DHCP/BootP is enabled or disabled.

Syntax:

```
<!--#exec cmd_argument='DisplayDhcpState(  
    "Output when ON", "Output when OFF")'-->
```

DisplayEmailServer

This function returns the current SMTP server address.

Syntax:

```
<!--#exec cmd_argument='DisplayEmailServer'-->
```

Store Function

StoreEtnConfig

This SSI function stores a passed IP configuration to FLASH.

Syntax:

```
<?--#exec cmd_argument='StoreEtnConfig'-->
```

Include this line in a HTML page and pass a form with new IP settings to it.

Accepted fields in form:

SetIp

SetSubnet

SetGateway

SetEmailServer

SetDhcpState - value "on" or "off"

Default output:

Invalid IP address!

Invalid Subnet mask!

Invalid Gateway address!

Invalid IP address or Subnet mask!

Invalid Email Server IP address!

Configuration stored correctly.

Invalid DHCP state!

Failed to store the configuration!

For information about how to change the SSI output, please see "Changing SSI Output" on page 9-21.

Formatted Display

printf

This SSI function includes a formatted string, which may contain data from the Input (IN) Output (OUT) area, on a web page. The formatting of the string is equal to the standard C function printf().

Syntax:

```
<?--#exec cmd_argument='printf(  
                                "String to write", Arg1, Arg2,..., ArgN) '-->
```

Like the standard C function printf() the “String to write” for this SSI function contains two types of objects: Ordinary characters, which are copied to the output stream, and conversion specifications, each of which causes conversion and printing of the next successive argument to printf. Each conversion specification begins with the character “%” and ends with a conversion character.

Between the “%” and the conversion character there may be the following modifiers:

Modifier	Description
-	Specifies left adjustment of the converted argument in its field.
+	Specifies that the number will always be printed with a sign.
space	If the first character is not a sign, a space will be prefixed.
0	Specifies padding to the field with leading zeroes.
#	Specifies an alternate output form. For o, the first digit will be zero. For x or X, 0x or 0X will be prefixed to a non-zero result. For e, E, f, g and G, the output will always have a decimal point; for g and G, trailing zeros will not be removed.
width	A number specifying a minimum field width. The converted argument will be printed in a field at least this wide, and wider if necessary. If the converted argument has fewer characters than the field width it will be padded on the left (or right, if left adjustment has been requested) to make up the field width. The padding character is normally space, but can be 0 if the zero padding flag is present.
precision	A number, the precision, that specifies the maximum number of characters to be printed from a string, or the number of digits to be printed after the decimal point for e, E, or F conversions, or the number of significant digits for g or G conversion, or the minimum number of digits to be printed for an integer (leading 0s will be added to make up the necessary width)
.	A period, which separates the field width from the precision.
h	A length modifier. “h” Indicates that the corresponding argument is to be printed as a short or unsigned short.
l or L	A length modifier. “L” or “l” indicates that the argument is along or unsigned long.

Table 9-1 printf Modifiers

The conversion characters and their meaning are shown below. If the character after the “%” is not conversion character, the behavior is undefined.

Character	Argument Type	Converted To
d, i	Byte,Short	Signed Decimal Notation
o	Byte,Short	Unsigned Octal Notation (without a leading zero)
x,X	Byte,Short	Unsigned hexadecimal notation (without a leading 0x or 0X)
u	Byte,Short	Unsigned decimal notation
c	Byte,Short	Single character, after conversion to unsigned char
s	char *	Characters from the string are printed until a “\0” is reached or until the number of characters indicated by the precision have been printed
f	Long	Decimal notation of the form [-] m.dxxxxde+ -xx or [-]m.dxxxxxE+ -xx where the number of d’s is specified by the precision. The default precision is 6; a precision of 0 suppresses the decimal point.
e,E	Long	Decimal notation of the form [-] m.dxxxxde+ -xx or [-]m.dxxxxxE+ -xx where the number of d’s is specified by the precision. The default precision is 6; a precision of 0 suppresses the decimal point.
g,G	Long	“%e” or “%E” is used if the exponent is less than -4 or greater than or equal to the precision; otherwise “%f” is used. Trailing zeroes and trailing decimal point are not printed.
%		Print a “%”

Table 9-2 printf Conversion Characters

The arguments that can be passed to the SSI function *printf* are:

Argument	Description
InReadSByte(offset)	Reads a signed byte from position offset in the Input (IN) area
InReadUByte(offset)	Reads a unsigned byte from position offset in the IN area
InReadSWord(offset)	Reads a signed word (short) from position offset in the IN area
InReadUWord(offset)	Reads a unsigned word (short) from position offset in the IN area
InReadSLong(offset)	Reads a signed longword (long) from position offset in the IN area
InReadULong(offset)	Reads an unsigned longword (long) from position offset in the IN area
InreadString(offset)	Reads a string (char*) byte from position offset in the IN area
InReadFloat(offset)	Reads a floating point (float) value from position offset in the IN area
OutReadSByte(offset)	Reads a signed byte from position offset in the OUT area
OutReadUByte(offset)	Reads a unsigned byte from position offset in the OUT area
OutReadSWord(offset)	Reads a signed word (short) from position offset in the OUT area
OutReadUWord(offset)	Reads a unsigned word (short) from position offset in the OUT area
OutReadSLong(offset)	Reads a signed longword (long) from position offset in the OUT area
OutReadULong(offset)	Reads an unsigned longword (long) from position offset in the OUT area
OutReadString(offset)	Reads a string (char*) byte from position offset in the OUT area
OutReadFloat(offset)	Reads a floating point (float) value from position offset in the OUT area

Table 9-3 SSI Functions to Read Data

Note: The I/O data is stored in the same format as it is received and sent on the DeviceNet network. DeviceNet is a little endian protocol; values are transmitted least significant byte first. Hence, all data in the I/O tables is assumed, by the DeviceNet nodes, to be stored as little endian. The web server operates in a big endian environment. Some data may have to be manipulated to account for byte ordering when displaying it on a web page.

Formatted Input

scanf

This SSI function reads a string passed from an object in a HTML form, interprets the string according to the specification in format, and stores the result in the Output (OUT) area according to the passed arguments. The formatting of the string is equal to the standard C function call scanf().

Syntax:

```
<?--#exec cmd_argument='scanf(ObjName, format,
    Arg1,..., ArgN), ErrVal1,..., ErrvalN'-->
```

ObjName	The name of the object with the passed data string
format	Specifies how the passed string shall be formatted
Argn	Specifies where to write the data
ErrValn	Optional; specifies the value/string to write in case of an error.

Character	Input Data and Argument Type
d	Decimal number; byte, short
i	Number, byte, short. The number may be in octal (leading 0(zero)) or hexadecimal (leading 0x or 0X)
o	Octal number (with or without leading zero); byte, short
u	Unsigned decimal number; unsigned byte, unsigned short
x	Hexadecimal number (with or without leading 0x or 0X); byte, short
c	Characters; char*. The next input characters (default 1) are placed at the indicated spot. The normal skip over white space is suppressed; to read the next non-white space character, use "%1s"
s	Character string (not quoted); char*, pointing to an array of characters large enough for the string and a terminating "\0" that will be added.
e, f, g	Floating-point number with optional sign, optional decimal point and optional exponent; float
%	Literal "%"; no assignment is made.

Table 9-4 scanf Formats

The conversion characters d, i, o, u and x may be preceded by the letter “l” to indicate that a pointer to ‘long’ appears in the argument list rather than a ‘byte’ or a ‘short’

The arguments that can be passed to the SSI function scanf are:

Argument	Description
OutWriteByte(offset)	Writes a byte to position <i>offset</i> in the OUT area
OutWriteWord(offset)	Writes a word (short) to position <i>offset</i> in the OUT area
OutWriteLong(offset)	Writes a long to position <i>offset</i> in the OUT area
OutWriteString(offset)	Writes a string to position <i>offset</i> in the OUT area
OutWriteFloat(offset)	Writes a floating point (float) value to position <i>offset</i> in the OUT area

Table 9-5 SSI Functions to Write Data

Default output:

```
Write succeeded
Write failed
```

For information about how to change the SSI Output, see “Changing SSI Output” on page 9-21“.

Note: The I/O data is stored in the same format as it is received and sent on the DeviceNet network. DeviceNet is a little endian protocol; values are transmitted least significant byte first. Hence, all data in the I/O tables is assumed, by the DeviceNet nodes, to be stored as little endian. The web server operates in a big endian environment. Some data may have to be manipulated to account for byte ordering when writing it from a web page.

Text Function

GetText

This SSI function gets the text from an object and stores it in the OUT area.

Syntax:

```
<?--#exec cmd arbgument='GetText(  
                                "ObjName", OutWriteString (offset), n)-->
```

offset specifies the offset from the beginning of the OUT area.

n (optional) specifies maximum number of characters to read

Default output:

Success - Write succeeded

Failure - Write failed

File Functions

IncludeFile

This SSI function includes the contents of a file on a web page.

Syntax:

```
<?--#exec cmd_argument='IncludeFile(Filename) '-->
```

Default output:

Success	- <File contents>
Failure	- Failed to open <filename>

SaveToFile

This SSI function saves the contents of a passed form to a file. The passed name/value pair will be written to the file “File name” separated by the “Separator” string. The contents can either be Appended to the file or overwrite the current content of the file.

Syntax:

```
<?--#exec cmd_argument='SaveToFile(  
    “File name”, “Separator”, [Append|Overwrite]) '-->
```

Default output:

Success	- Form saved to file
Failure	- Failed to save form

String Functions

Changing SSI Output

There are two methods of changing the output strings from SSI functions:

- Changing SSI output defaults by creating a file called “\ssi_str.cfg” containing the output strings for all SSI functions in the system.
- Temporary changing the SSI output by calling the SSI function “SsiOutput()”.

SSI Output string file

If the file “\ssi_str.cfg” is found in the file system and the file is correct according to the specification below, the SSI functions will use the output strings specified in this file instead of the default strings.

The file has the following format:

[StoreEtnConfig]

Success: “String to use on success”

Invalid IP: “String to use when the IP address is invalid”

Invalid Subnet: “String to use when the Subnet mask is invalid”

Invalid Gateway: “String to use when the Gateway address is invalid”

Invalid Email server: “String to use when the SMTP address is invalid”

Invalid IP or Subnet: “String to use when the IP address and Subnet mask does not match”

Save Error: “String to use when storage fails”

Invalid DHCP state: “String to use when the DHCP state is invalid”

[scanf]

Success: “String to use on success”

Failure: “String to use on failure”

[IncludeFile]

Failure: “String to use when failure” To include filename “%s” can be included to the string once

[SaveToFile]

Success: “String to use on success”

Failure: “String to use on failure” To include filename “%s” can be included to the string once.

[GetText]

Success: “String to use on success”

Failure: “String to use on failure”

The contents of this file can be redirected by placing the line [File path] on the first row, and the actual file path on the second row.

Temporary SSI Output change

The SSI output for the next called SSI function can be changed with the SSI function “SsiOutput()” The next called SSI function will use the output according to this call. Thereafter the SSI functions use the default outputs or the outputs according to the file ‘\ssi_str.cfg’. The maximum size of a string is 128 bytes.

Syntax:

```
<?--#exec cmd_argument='SsiOutput(
    "Success string", "Failure string")'-->
```

Example:

This example shows how to change the output strings for a scanf SSI call.

```
<?--#exec cmd_argument='SsiOutput ("Parameter1 updated", "Error")'-->
<?--#exec cmd_argument='scanf("Parameter1", "%d", OutWriteByte(0))'-->
```

Email Client

It is possible to send predefined email messages to predefined receivers triggered by an event in the IN/OUT area. This area is scanned every 5 seconds. The IP address to a SMTP (mail) server must be configured (See “ethcfg.cfg” on page 8-7). The SMTP server must not require a username/password for access. It is possible to have 10 user defined, and 10 admin defined emails triggered on different events. These shall be placed in directories “\user\email” for user configurable emails and “\email” for non-user configurable emails. See “Email files (email_1.cfg,email_2.cfg to email_10.cfg)” on page 8-14.

SSI Includes in emails

For predefined emails it possible to include data. This is performed in the same way data is added to web pages using SSI functions. The supported SSI functions for emails include:

- DisplayIP
- DisplaySubnet
- DisplayGateway
- DisplayEmailServer
- printf
- IncludeFile
- SsiOutput

Displaying I/O Data on a Web Page

The following is an example of an HTML file that when uploaded to the module displays in hex the second byte of data from the IN table and the third byte of data of the OUT table using the SSI “printf” command.

```
<html>
<head>
<title>Ethernet to DeviceNet BridgeWay</title>
</head>
<body>
<center>
</h1>
<!--#exec cmd_argument='printf("IN 2 = 0x%2X",InReadUByte(2))'-->
<!--#exec cmd_argument='printf("OUT 3 = 0x%2X",OutReadUByte(3))'-->
</body>
</html>
```

Status and Diagnostics

BridgeWay LEDs

There is a group of LED indicators on the front of the BridgeWay that is used to announce the current status of the module and the network interfaces. The layout of the LEDs is shown in Figure 10-1.

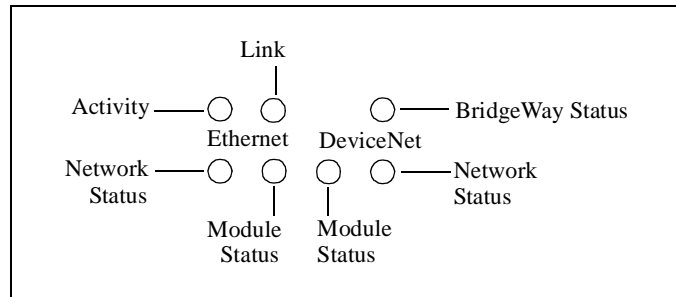


Figure 10-1 BridgeWay LEDs

BridgeWay Status LED

State	Summary	Description
Flashing Green	Idle	Module is in Idle mode.
Solid Green	Run	Module is in Run mode.
Solid Orange	Hardware Initialization	The LED will be in this state immediately after power is applied.
Flashing Red/Green	Error	A major, unrecoverable fault has been detected.
Red, Green, Orange Alternate Flashing	Self Test	A self test of the module is in progress.

Table 10-1 BridgeWay Status LED States

Major unrecoverable faults are indicated by a series of green and red flashes. If the BridgeWay Status LED is flashing red and green for an extended period of time, count the number of red and green flashes and call technical support.

DeviceNet Network Status LED

State	Summary	Description
Solid Green	Online and communicating	The BridgeWay is on the DeviceNet network and communicating with at least 1 device.
Flashing Green	Online, no communication	The BridgeWay is on the DeviceNet network and is not currently communicating with any devices.
Solid Red	DeviceNet interface fault	A major fault in the DeviceNet interface has been detected. Possible causes include Bus-off or duplicate MAC ID.
Flashing Red	Connection time-out	A connection with at least 1 slave device has timed out.
Red,Green Alternate Flashing	Self Test	A self test of the module is in progress.

Table 10-2 DeviceNet Network Status LED States**DeviceNet Module Status LED**

State	Summary	Description
Flashing Green	Initializing, standby, or not configured	The module is initializing. The DeviceNet network configuration has not been configured and is currently using default values.
Solid Green	Normal	Normal operation.
Solid Red	Unrecoverable fault	A fault that requires user intervention has been detected. Correct the problem and reset the BridgeWay.
Flashing Red	Recoverable fault.	A fault that can be corrected and does not require a BridgeWay reset has been detected. This will typically be a configuration error.
Red,Green Alternate Flashing	Self Test	A self test of the module is in progress.

Table 10-3 DeviceNet Module Status LED States

Ethernet Activity LED

The Ethernet Activity LED flashes green as Ethernet packets are received or transmitted.

Ethernet Link LED

The Ethernet Link LED indicates that the module is connected to an Ethernet network. The LED will display solid green if there is a valid physical link.

Ethernet Module Status LED

State	Summary	Description
Off	No Power	Not powered
Solid Green	Normal	The module is operating correctly.
Flashing Green	Standby	The module has not been initialized.
Solid Red	Unrecoverable fault	A fault that requires user intervention has been detected. Correct the problem and reset the BridgeWay.
Flashing Red	Recoverable fault.	A fault that can be corrected and does not require a BridgeWay reset has been detected.
Red,Green Alternate Flashing	Self Test	A self test of the module is in progress.

Table 10-4 Ethernet Module Status LED States

Ethernet Network Status LED

State	Summary	Description
Off	No Power	The module has no power or no IP address assigned.
Solid Green	Network OK and communicating	There is at least one EtherNet/IP connection. (<i>Not affected by Modbus/TCP connections.</i>)
Flashing Green	Network OK	There are no active connections. (<i>Not affected by Modbus/TCP connections.</i>)
Solid Red	Address conflict	The module's IP address is already in use by another module.
Flashing Red	Connection Time-out	One or more of the connections in which this module is the target has timed out. This state is only left if all timed out connections are re-established or if the module is reset.
Red,Green Alternate Flashing	Self Test	A self test of the module is in progress.

Table 10-5 Ethernet Network Status LED States

Diagnostic Web Pages

Status and Settings Web Page

The Status and Settings page displays the BridgeWay identification information, current status, and IP configuration. The IP configuration can be changed from this page. The module status is updated approximately every 2.5 seconds.

Active Nodes Web Page

The Active Nodes page indicates which DeviceNet nodes are currently configured as slaves to the BridgeWay's DeviceNet scanner. Each node that is configured as a slave will be displayed with "Active" next to the node's MAC ID.

Idle Nodes Web Page

The Idle Nodes page indicates which DeviceNet nodes are currently in the Idle state. If a node is Idle, the page will display "Idle" next to the node's MAC ID. Note that only nodes which are configured as slaves to the BridgeWay and the BridgeWay itself are updated on this page.

Faulted Nodes Web Page

The Faulted Nodes page indicates which DeviceNet nodes are currently in a faulted state. If a node is faulted, the page will display “Faulted” next to the node’s MAC ID. Note that only nodes which are configured as slaves to the BridgeWay and the BridgeWay itself are updated on this page.

A node is considered faulted if the BridgeWay has lost communications or is unable to establish communications with the node. The actual problem can be determined by viewing the Node Status web page.

Invalid Nodes Web Page

The Invalid Nodes page indicates which DeviceNet nodes are not the correct device type. If a node’s device type is different than that configured in the scan list, the page will display “Invalid” next to the node’s MAC ID. Note that only nodes which are configured as slaves to the BridgeWay are updated on this page.

Node Status Web Page

The Node Status page displays the current status of all DeviceNet nodes that are configured as slaves to the BridgeWay and the BridgeWay itself. The status of each node is displayed next to the node’s MAC ID. Note that only nodes which are configured as slaves to the BridgeWay and the BridgeWay itself are updated on this page.

The page will display the status textually for many of the common status values. However, to save web page size, a lot of the status values are only displayed numerically. The following table describes the meaning of each status value.

Status Code	Description
1	Ok.
60	Duplicate MAC ID test in progress.
70	Duplicate MAC ID failure.
71	Scanner configuration error.
72	Communications failed with the device.
73	Incorrect device type.

Table 10-6 Node Status Codes

Status Code	Description
74	CAN receive overrun error.
75	CAN network failure.
76	Scanner is lonely; it is not receiving any network traffic.
77	Incorrect connection size.
78	No response from the device.
79	CAN transmit failure.
80	In Idle mode.
81	Faulted.
82	DeviceNet fragmentation error.
83	Error during slave connection initialization.
84	Slave connection initialization in progress.
85	Incorrect data size received on connection.
86	Device went into Idle mode.
87	Shared master has not made connection to the device.
88	Shared master has not made the right type of connection to the device.
89	ADR keeper error.
90	CAN network disabled.
91	CAN bus-off.
92	No DeviceNet power.
95	Flash update in progress.
96	CAN port is in Test mode.
97	Device halted.
98	Firmware failure.
99	System failure.

Table 10-6 Node Status Codes (Continued)

Status Assembly

The status assembly is an assembly object instance that is accessible from Ether-Net/IP both explicitly and using an I/O connection. The status assembly contains current status and diagnostic information pertaining to the BridgeWay's DeviceNet interface. See "Status Assembly" on page 6-8 for complete details of the format and content of the assembly.

Specifications

Environmental Specifications

Temperature

Operating: 0 to 70 degrees Celsius

Non-Operating: -25 to 85 degrees Celsius

EMC Directive Compliance

This product is tested to meet the Council Directive 89/336/EC Electromagnetic Compatibility (EMC) by applying the following standards, in whole or in part, documented in a technical construction file:

- EN50081-2-EMC Generic Emission Standard, Part 2 - Industrial Environment
- EN50082-2-EMC Generic Immunity Standard, Part 2 - Industrial Environment

This product is intended for use in an industrial environment.

Electrical Specifications

DC Power

Operating voltage: 7-32v DC.

Current Requirements: 130-140 mA at 24 VDC.

Mechanical Specifications

Mechanical Rating

IP20/NEMA 1

Dimensions

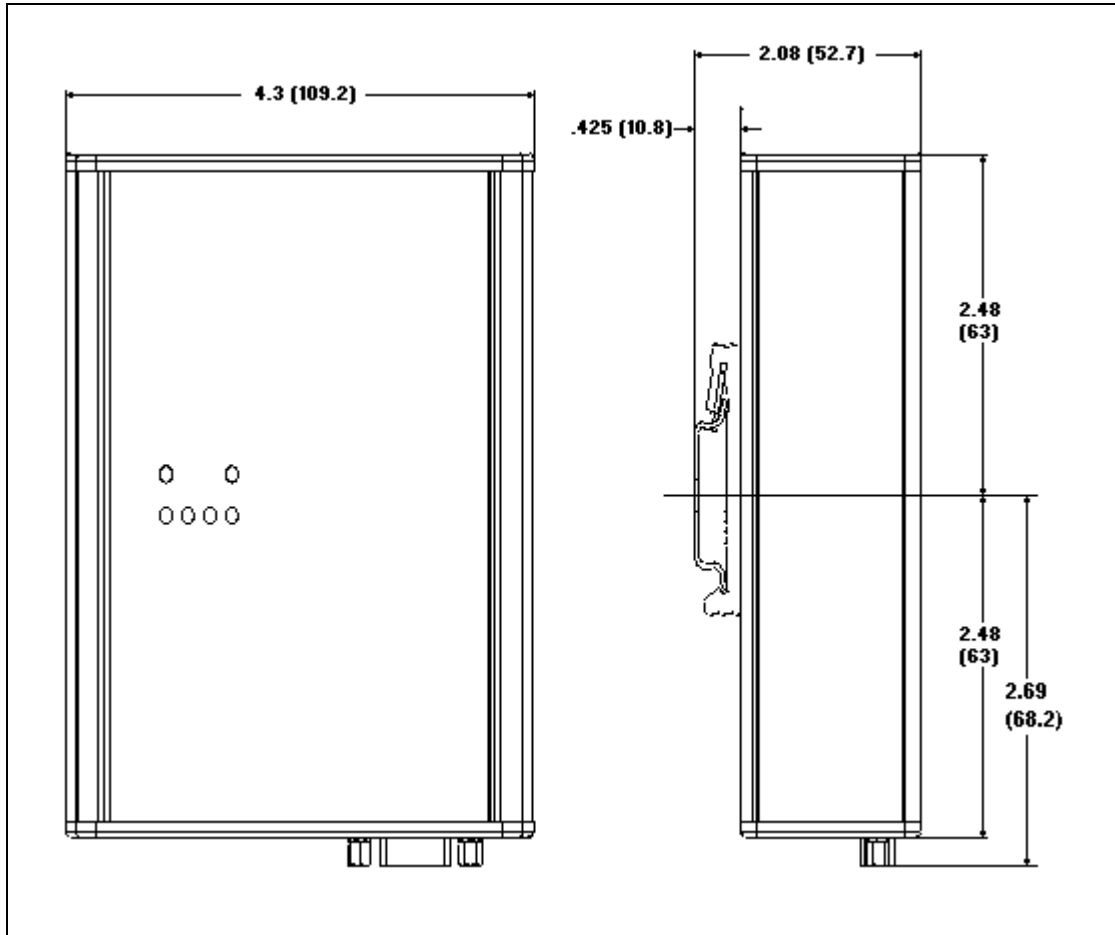


Figure 11-1 BridgeWay Mechanical Dimensions

I/O Data Sizes

Input

Maximum 508 bytes Input

Output

Maximum 508 bytes Output

Status

128 bytes of Status data

Connectors

Power

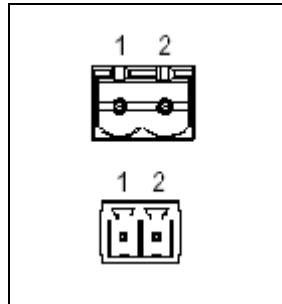
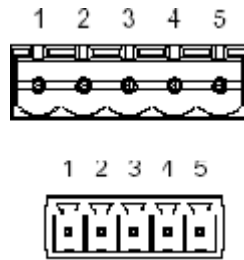


Figure 12-1 Power Connector

Pin	Connection
1	24 VDC +
2	24 VDC Common

Table 12-1 Power Connector Pin Definitions

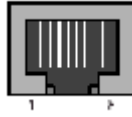
Use Phoenix connector part number MSTB 2,5/2-ST-5,08 ABGY

DeviceNet

Pin	Signal
1	V-
2	CAN_L
3	Shield
4	CAN_H
5	V+

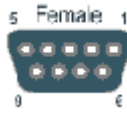
Table 12-2 DeviceNet Pinout

Use Phoenix connector part number MSTB 2,5/5-ST-5,08-ABGYAU

Ethernet RJ45

Pin	Signal
1	TD+
2	TD-
3	RD+
4	Termination
5	Termination
6	RD-
7	Termination
8	Termination

Table 12-3 RJ45 Pinout

Auxiliary RS-232 9 Pin D-Subminiature

Pin	Signal
1	NC
2	Receive
3	Transmit
4	NC
5	NC
6	NC
7	NC
8	NC
9	NC

Table 12-4 RS232 9 Pin

Warranty

Pyramid Solutions warrants all new products to be free of defects in material and workmanship when applied in the manner for which they were intended and according to Pyramid Solutions' published information on proper installation. The Warranty period is one year from the date of shipment.

Pyramid Solutions will repair or replace, at its option, all products returned to it freight prepaid, which prove upon examination to be within the Warranty definitions and time period.

Limitation of Liability.

EXCEPT AS EXPRESSLY PROVIDED HEREIN, PYRAMID SOLUTIONS MAKES NO WARRANTY OF ANY KIND, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE. PYRAMID SOLUTIONS SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF DESIGN, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NONINFRINGEMENT. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED IN THIS AGREEMENT. NO AGENT OF PYRAMID SOLUTIONS IS AUTHORIZED TO INCUR WARRANTY OBLIGATIONS ON BEHALF OF PYRAMID SOLUTIONS OR TO EXPAND OR MODIFY THE LIMITATIONS SET FORTH HEREIN.

IN NO EVENT WILL PYRAMID SOLUTIONS OR ITS SUPPLIERS HAVE ANY LIABILITY TO BUYER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES HOWEVER CAUSED AND UNDER ANY THEORY OF LIABILITY (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), INDEMNITY OR OTHERWISE) WHETHER OR NOT THE PYRAMID SOLUTIONS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

IN NO EVENT WILL PYRAMID SOLUTIONS OR ITS SUPPLIERS HAVE ANY LIABILITY TO THE BUYER FOR ANY LOST PROFITS, LOSS OF DATA, LOSS OF USE, OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, EXCEPT IN CASES OF WILLFUL MISCONDUCT BY THE PYRAMID SOLUTIONS.

Support

Technical Product Assistance

If you need to contact Pyramid Solutions for technical assistance,
ask for BridgeWay technical support at:

1-248-524-3890.

You can obtain technical assistance by email at:

ProductSupport@pyramid-solutions.com.

You can also obtain technical assistance and download documentation and firm-
ware updates online at:

http://www.pyramid-solutions.com/Expertise_DataCom_Support.htm.

Contact Information

Pyramid Solutions, Inc.
1850 Research Drive, Suite 300
Troy, Michigan 48083-2167

Phone: (248) 524-3890

1-888-PYRASOL

FAX: (248) 524-3899

Web: www.Pyramid-Solutions.com