# *Anybus X-gateway*
# *Ethernet to DeviceNet Gateway*
# *User Manual*

Part No**. AB7607**
Doc.Id. HMSI-168-24
For Firmware Revision **2.03.01** and Later
Manual Revision 1.20

*Connecting Devices™*

*HALMSTAD • CHICAGO • KARLSRUHE • TOKYO • BEIJING • MILANO • MULHOUSE • COVENTRY • PUNE • COPENHAGEN*

HMS Industrial Networks
Mailing address: Box 4126, 300 04 Halmstad, Sweden
Visiting address: Stationsgatan 37, Halmstad, Sweden

E-mail: info@hms-networks.com
Web: www.anybus.com

# Table of Contents



# Preface

## *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 HMS Industrial Networks. HMS Industrial Networks assumes no responsibility for errors that may appear in this document

There are many applications of the Anybus X-gateway 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. HMS Industrial Networks does not assume responsibility or liability for actual use based upon the examples shown in this publication.

**FAIL-SAFE OR CRITICAL OPERATIONS**

This product is not designed, intended, authorized, or warranted to be suitable for use or resale as control equipment in, or for other applications related to, hazardous or potentially-hazardous environments or applications requiring high-availability or fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, public works, weapons systems, or any other application in which the failure of a product could lead to property damage, death, personal injury, or environmental damage.

### *Related Documentation*

| Document Name | Author | Web Page |
|---|---|---|
| DeviceNet Specification | ODVA | www.odva.org |
| EtherNet/IP Specification | ODVA | www.odva.org |
| Modbus/TCP | Modbus-IDA | www.modbus.org |

**Table 2-1 Related Documentation**

DeviceNet is a trademark of Open DeviceVendor 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.

# Anybus X-gateway Module Description

## *Overview*

The Anybus Ethernet to DeviceNet X-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 Anybus X-gateway Ethernet to DeviceNet applications:

- The X-gateway 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)
- The X-gateway can provide router/bridge functionality to connect EtherNet/IP to DeviceNet.
- The X-gateway 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 X-gateway 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.

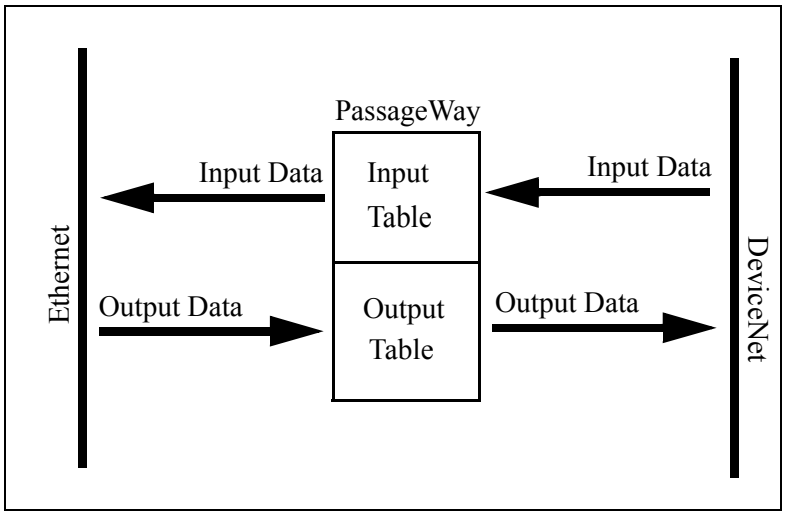



**Figure 1-1 Anybus X-gateway PassageWay Operation**

The X-gateway 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, Change of State (COS), and Cyclic 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 496 bytes DeviceNet slave input data and 492 bytes DeviceNet slave output data.
- DeviceNet Quick Connect.
- Active Node List monitoring tracks the online status of all DeviceNet nodes. Bridged CIP messages will be immediately rejected if the target node is not online.
- Configurable input safe state mode determines the state of slave input data when a slave's I/O connection times out.

### *Ethernet Features*

- Supports the EtherNet/IP protocol, Adapter Class with I/O Server, 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.
- Address may be set via DHCP/Bootp, DIP switch, or software configuration.

## IT-Features

- The X-gateway 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 customizeable web page interface.
- Firmware updates of the Anybus X-gateway using the RS232 port and Anybus X-gateway Configuration Tool (BWConfig).
- Email client capability.

## System Requirements

The following hardware and software components are needed to use the Anybus X-gateway Ethernet to DeviceNet device.

### Required Hardware

- Anybus X-gateway Ethernet to DeviceNet module.
- DeviceNet cabling, power, and devices forming a DeviceNet network.
- Ethernet cabling.
- PC or controller with access to the Ethernet network.
- PC to execute DeviceNet Configuration Software. The DeviceNet slave devices the X-gateway 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 X-gateway module. (DeviceNet power may be used.)

### Optional Hardware

- A PC with a serial RS232 COM port to be used by the Anybus X-gateway Configuration Tool Software for setting DeviceNet and Ethernet network configuration.

- RS232 null-modem cable (pins 2 and 3 swapped) from the PC to the X-gateway module. This may use either a serial port or a USB serial adapter on the PC.

- DIN rail to mount the X-gateway.

### Required Software

- DeviceNet configuration software such as RSNetWorx for DeviceNet or HMS NetTool-DN-D to configure DeviceNet devices and X-gateway's DeviceNet operation. RSLinx version 2.31 or later is required. **RSNetworx v7.0 or later is required to support the full 128K of the ADR configuration recovery data; earlier versions only support up to 64K bytes of data. NetTool-DN-D, as of v3.3.1, supports up to 64K of configuration data.**

### Optional Software

- Anybus X-gateway Configuration Tool Software (BWConfig) for DeviceNet and Ethernet network configuration.

### *Hardware Description*

All connections, whether power or fieldbus, to the X-gateway are made on one end of the module. Phoenix-style 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 network interface configuration. 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 3-6 for more details on configuring the IP address using the switches.

On the front of the X-gateway 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 “Anybus X-gateway 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 X-gateway are made on the end of the module. Figure 2-1 indicates the location of each connector.

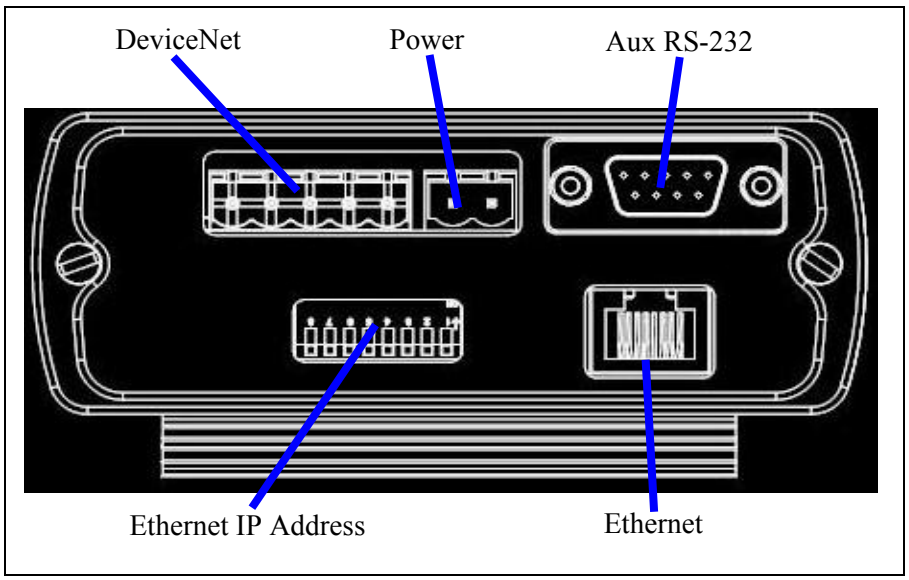



**Figure 2-1 Anybus X-gateway Power and Network Connections**

### *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 X-gateway. Connections to be made are illustrated in Figure 2-2.

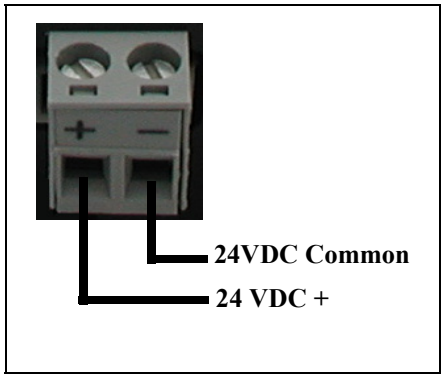



**Figure 2-2 Power Connection**

The X-gateway 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 X-gateway. Connections to be made are illustrated in Figure 2-3.

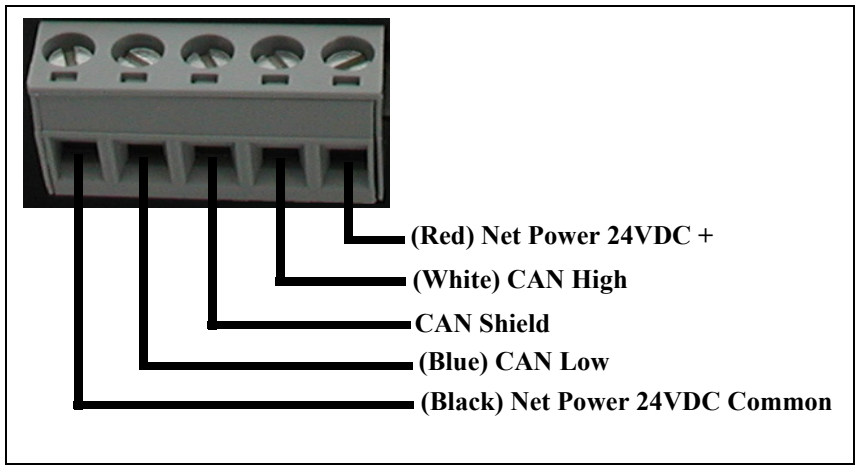



**Figure 2-3 DeviceNet Connection**

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 3-15.

### *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 3-6.

### *Configuration Port Connector*

The configuration port is the 9-pin D-Subminiature male connector on the end of the X-gateway. The connector has a standard RS-232 DTE pin configuration. The connections to be made as shown below.

| Pin | Connection |
|---|---|
| 2 | Receive Data |
| 3 | Transmit Data |
| 5 | Signal Ground |

The X-gateway is connected to a PC for configuration using a null-modem cable. A null-modem cable has pins 2 and 3 swapped so that the PC's Transmit line is connected to the X-gateway's Receive line, and the PC's Receive line is connected to the X-gateway's Transmit line.

**Note:** The Anybus X-gateway does not make use of the modem control signals specified for a DTE connector. Connecting the module through devices, such as isolation modules, which assume control of these lines may cause the BWConfig communications to be unreliable.

# Configuration

This chapter describes how the Anybus Ethernet to DeviceNet X-gateway is configured. The next chapter walks the reader through the configuration of the X-gateway using the commonly available configuration tools.

## *Anybus X-gateway Configuration Tool (BWConfig)*

The Anybus X-gateway Configuration Tool allows you to configure the parameters associated with the Ethernet and DeviceNet network interfaces.

BWConfig is a Microsoft Windows application that communicates with the Anybus X-gateway over a standard RS-232 serial link using the PC serial port or USB serial adapter. BWConfig is compatible with Microsoft Windows 95, 98, NT, 2000, and XP.

### Installing the Tool

Install BWConfig from the CD by running *Setup.exe* which is found in the CD's root directory.

If you have downloaded BWConfig from the web site, unzip the downloaded file into a temporary directory and run *Setup.exe* which is found in the temporary directory.

### Connecting to the X-gateway Module

Connect the PC running BWConfig to the X-gateway module using a standard Null-Modem (pins 2 and 3 swapped) serial cable between the PC serial port or USB serial adapter and the 9-pin D-Sub connector on the module. It does not matter which PC serial port you use, BWConfig will scan each available port and detect the connection automatically. No serial port configuration is required; BWConfig will automatically set the baud rate.

### Starting the Tool

Launch BWConfig from the *BridgeWay Configuration* folder in the Windows Start Menu.

When BWConfig is started, it will attempt to locate an X-gateway module on one of the PC serial ports. If a module is found, the status area of the tool will be updated to show the module type and status of the module that was located.

If a module is not connected to the PC, or is powered off, when the tool is started, the status area will indicate that no module was detected. Make sure that the module is powered and the connection is made, then press the Refresh button on the BWConfig tool bar; this will cause the tool to rescan the serial ports for a module.

### BWConfig User Interface

The Anybus X-gateway Configuration Tool's user interface is shown in Figure 3-1.

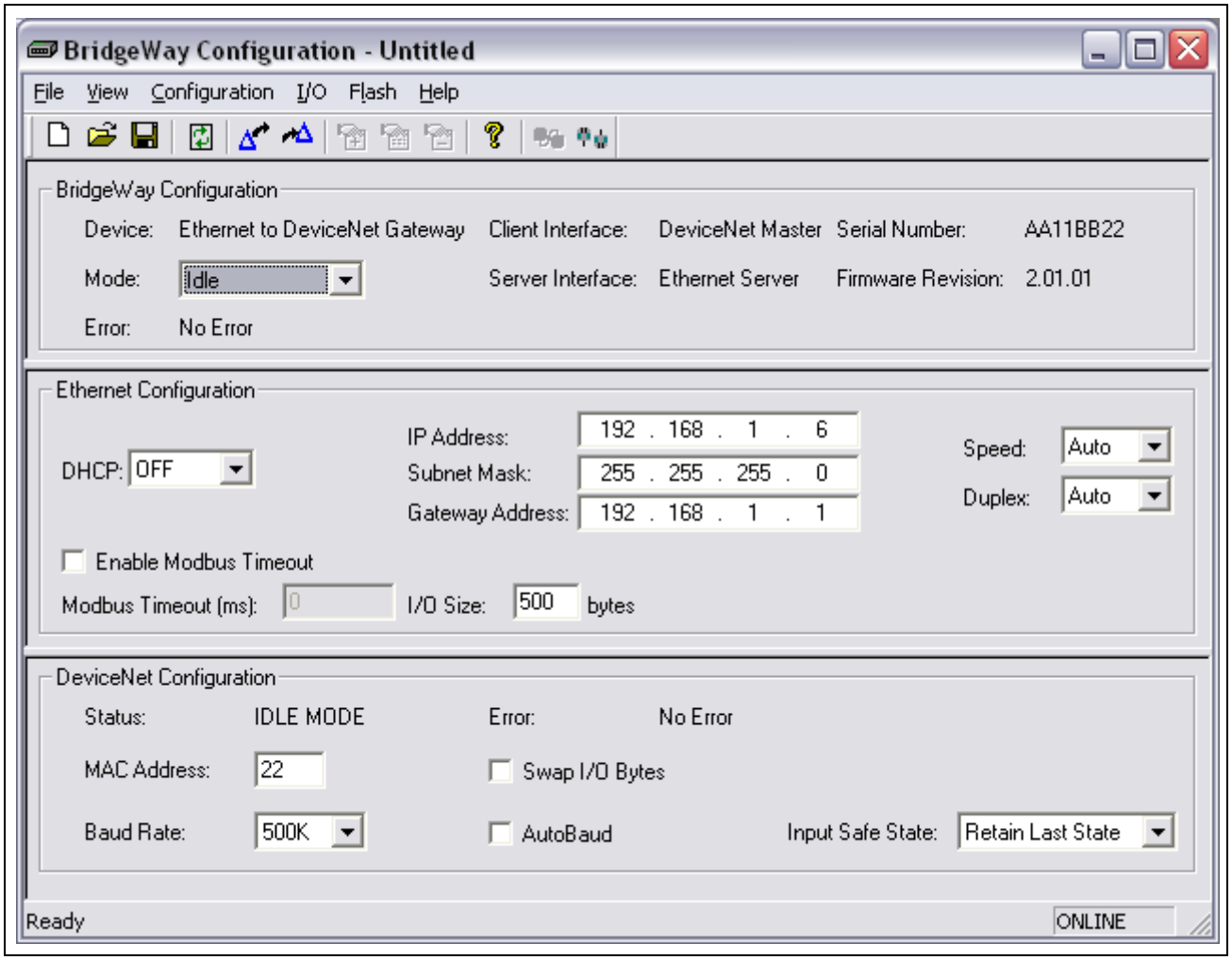



**Figure 3-1 BWConfig User Interface**

### Display Panes

The BWConfig display is divided into 3 panes.

| | |
|---|---|
| Anybus X-gateway Configuration | Module type and status information about the X-gateway module that was detected. |
| Ethernet Configuration | Configuration of Ethernet network parameters. |
| DeviceNet Configuration | Configuration of DeviceNet network parameters and status of the network interface. |

### Tool Operations

The following operations are available through the BWConfig menus and tool bar.

| | |
|---|---|
| New File | Create a new X-gateway configuration for the selected type of module. |
| Open File | Open a previously saved X-gateway configuration. |
| Save File | Save the current X-gateway configuration to a file. |
| Refresh Device Status | Refresh the module identity and status information. This will update the current status information shown by the tool. This can also be used to start the detection process if a module has not been detected by the tool, or the connection has been changed to a different module. |
| Upload Configuration | Read the configuration that is currently stored in the X-gateway module. This will overwrite any configuration that is displayed on the tool's user interface. |
| Download Configuration | Send the configuration shown on the tool's user interface to the X-gateway module. |

| | |
|---|---|
| Offline Configuration | Offline configuration will allow a configuration to be created and saved without being connected to a module. |
| Flash Update | Perform a field upgrade of the X-gateway module's firmware.<br><br>**Note:** Care should be taken when upgrading firmware, an incomplete update could cause irreparable harm to the module. |

## *Ethernet Network Configuration*

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

### Setting the IP Address with BWConfig

The Ethernet network configuration pane in BWConfig contains the parameters used to control the behavior of the Ethernet network interface. The parameters are described in Table 3-1 below. Refer to Figure 3-1 to see how each parameter is displayed on the user interface.

| Parameter | Description | Allowable Range |
|---|---|---|
| DHCP Enable | If DHCP is enabled, the module will receive its IP configuration from a DHCP server on the network.<br>If no DHCP server is available, the module will revert to the last saved IP configuration. | On or Off |
| IP Address | The IP address the module will use on the Ethernet network.<br>If DHCP is enabled, and a DHCP server is found, this address is ignored. If a DHCP server is not found, this address is used. | Valid IP address |
| Subnet Mask | The subnet mask the module will use on the Ethernet network.<br>If DHCP is enabled, and a DHCP server is found, this mask is ignored. If a DHCP server is not found, this mask is used. | Valid IP sub-net mask |
| Gateway Address | The IP address of the gateway module on the network.<br>If DHCP is enabled, and a DHCP server is found, this address is ignored. If a DHCP server is not found, this address is used. | Valid IP address |
| Network Speed | The speed that the module will communicate at on the Ethernet network.<br>If the network speed is set to Auto, the module will auto-negotiate network speed. | 10, 100, or Auto |

**Table 3-1 Ethernet Network Configuration Parameters**

| Parameter | Description | Allowable Range |
|---|---|---|
| Network Duplex | The duplex setting that the module will use to communicate on the Ethernet network.<br>If the network duplex is set to Auto, the module will auto-negotiate duplex. | Half, Full, or Auto |
| Modbus Timeout | The Modbus Timeout option provides a means to detect the loss of the Modbus Scanner from the Ethernet network. If the option is enabled, and no Modbus requests are received within the configured timeout period, the module Run/Idle status will be set to Idle.<br><br>**Important:** Do not enable the Modbus Timeout if an EtherNet/IP Scanner is used with the Anybus X-gateway. The X-gateway will be prohibited from entering Run mode if there is no Modbus messages. | 0-65000ms |
| I/O Size | The I/O Size parameter provides the means to configure the maximum size of the Input and Output Assembly objects. This is useful when accessing the X-gateway Assembly object using Class 3 or UCMM messages with modules that do not support large assembly buffer sizes.<br><br>The I/O size includes the status and command headers as well as the DeviceNet slave data. See “I/O Data Summary” on page 6-10 for details.<br><br>The actual output assembly size will be 4 bytes less than the I/O size configured. Again, refer to “I/O Data Summary” on page 6-10 for details.<br><br>I/O table sizes below 500 truncate the input and output tables. Any DeviceNet slave data that may be mapped beyond the configured I/O table size will not be transferred to Ethernet.<br><br>Suggested maximum sizes for various EtherNet/IP devices:<br>MicroLogix 252<br>SLC 5/05 248<br>ControlLogix 500 | 4-500 |

**Table 3-1 Ethernet Network Configuration Parameters**

### 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 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.*n* |
| 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 X-gateway's web page which allows changing the IP Address, Subnet mask, and GateWay address settings.

**Note:** A non-zero DIP switch setting will override any other Ethernet configuration that is done.

DIP Switch Example

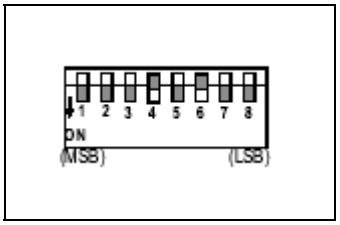



**Figure 3-2 IP Configuration DIP Switch**

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 BWConfig or the X-gateway's Settings web page.

**Note:** The use of DHCP is the default configuration for the X-gateway 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 X-gateway. These can be configured using the X-gateway'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 X-gateway 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 X-gateway 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 X-gateway. The Status and Settings web page appears as shown below.

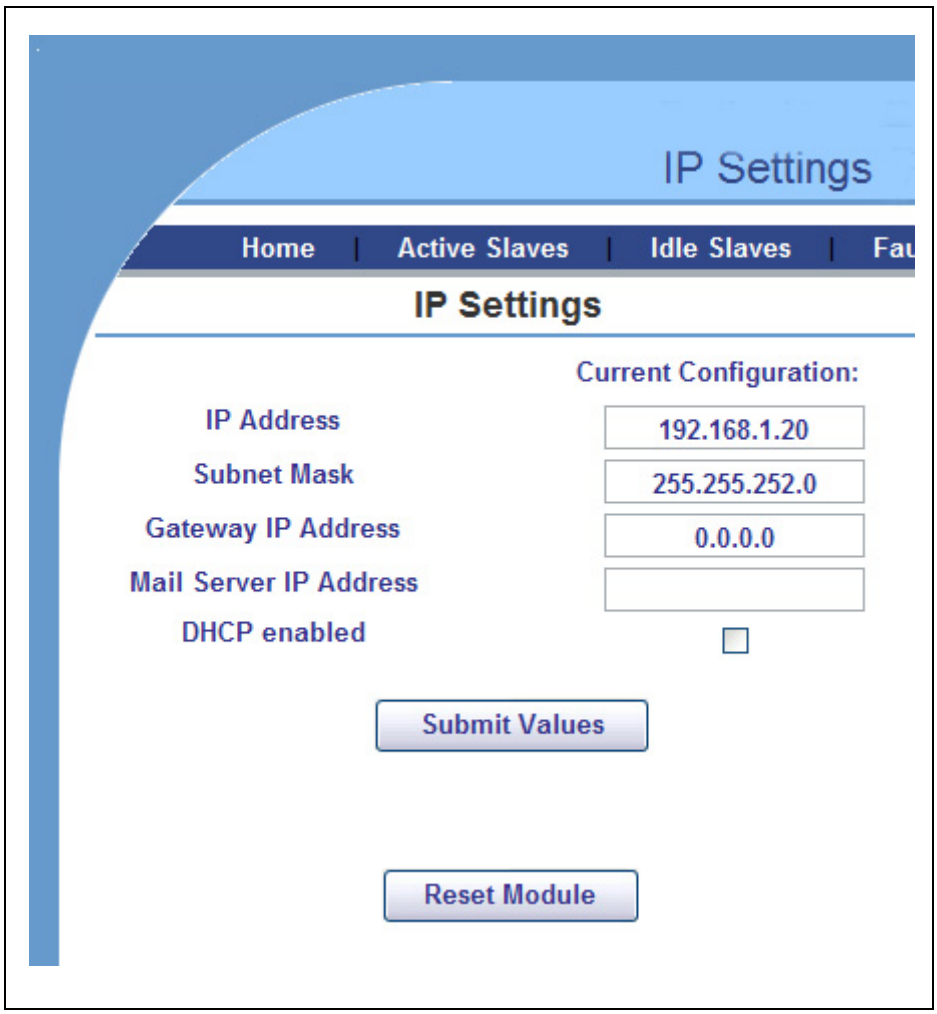



**Figure 3-3 Status and Settings Web Page**

The IP address, subnet mask, gateway and mail server addresses 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 X-gateway. 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 X-gateway from the web browser. A status of "RESETTING..." will display 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 X-gateway 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 X-gateway is powered up.

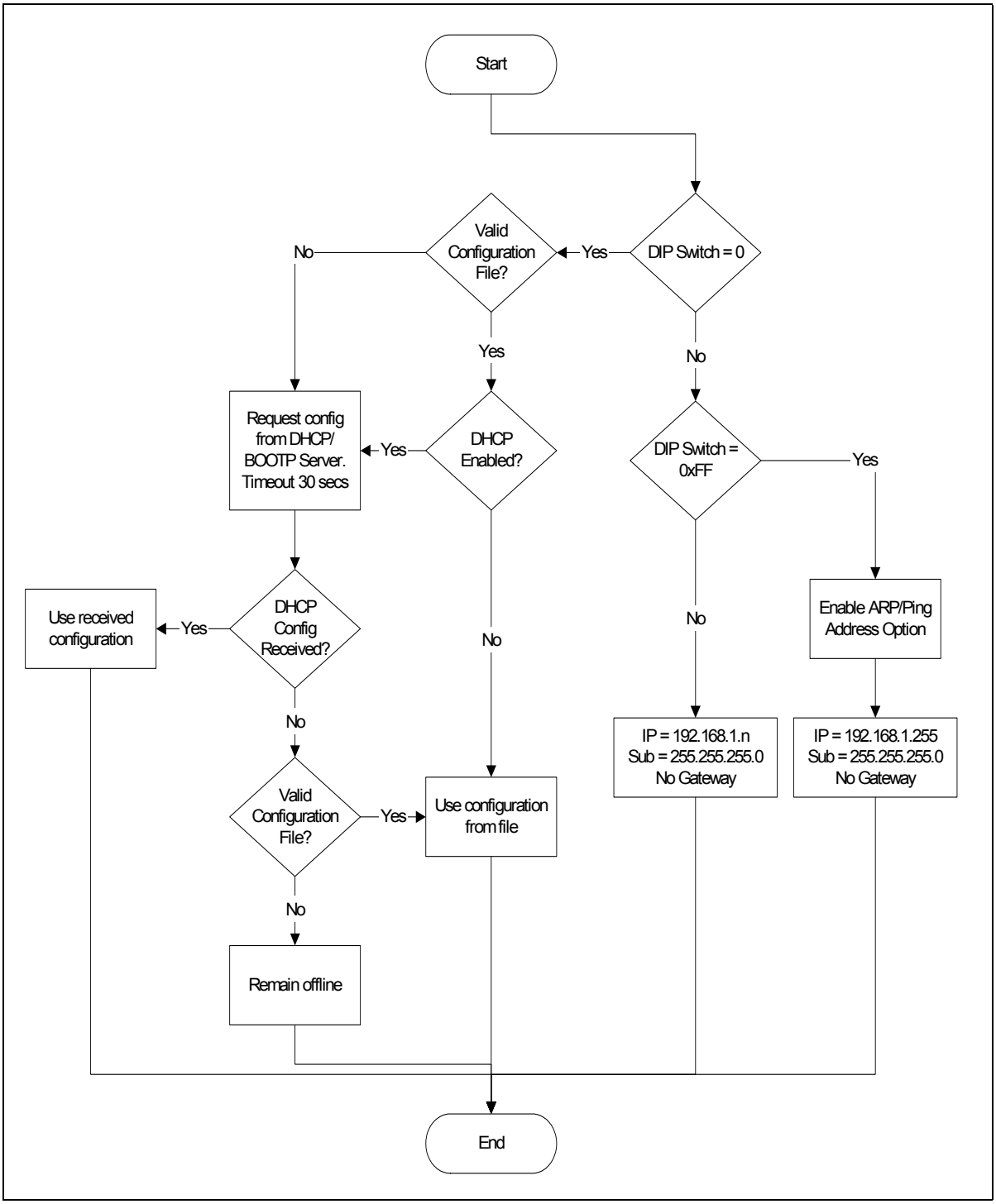



**Figure 3-4 IP Configuration Initialization Sequence**

## *DeviceNet Network Configuration*

### Setting the DeviceNet Configuration with BWConfig

The DeviceNet network configuration pane in BWConfig contains the parameters used to control the behavior of the DeviceNet network interface. The parameters are described in Table 3-2 below. Refer to Figure 3-1 to see how each parameter is displayed on the user interface.

| Parameter | Description | Allowable Range |
|---|---|---|
| MAC Address | The network address the X-gateway will use on the DeviceNet network.<br><br>The MAC address factory default is 63. | 0-63 |
| Baud Rate | The baud rate of the DeviceNet network.<br><br>The baud rate factory default is 125K | 125K<br>250K<br>500K |
| Auto Baud | Enable or disable automatic baud rate detection on the X-gateway.<br><br>The factory default setting for automatic baud detection is Disabled.<br><br>**Note**: If the X-gateway is the primary master on the DeviceNet network, do not enable automatic baud detection. | Enabled or Disabled |
| Swap I/O Bytes | Enable or disable I/O data byte swapping.<br><br>This option will swap bytes in the I/O tables on 16-bit boundaries. This is helpful when using Modbus/TCP, which expects data to be stored in reverse byte orientation from DeviceNet.<br><br>**Important:** Do not set the Swap I/O option if an EtherNet/IP Scanner is used with the X-gateway. The X-gateway will be prohibited from entering Run mode. | Enabled or Disabled |

**Table 3-2 DeviceNet Network Configuration Parameters**

| Parameter | Description | Allowable Range |
|---|---|---|
| Input Data Safe State | Define the safe state for the DeviceNet input data.<br><br>The safe state determines what will happen to the data in the input table associated with a DeviceNet slave when the connection to that slaves times out. Setting the safe state to "Retain Last State" will cause the slave's data to freeze at the last value received from the slave. Setting the value to "Zero Data" will cause all input data associated with the slave to be set to zero. | Retain Last State<br>or<br>Zero Data |

**Table 3-2 DeviceNet Network Configuration Parameters**

**Note:** The X-gateway will automatically reset after the DeviceNet configuration is downloaded from BWConfig.

### Setting the DeviceNet Configuration with Node Commissioning Tools

The DeviceNet network configuration may be set using DeviceNet node commissioning tool like RSNetworx or NetTool-DN-D. The parameters discussed above must be set through the parameter editing function of these tools. See "DeviceNet Network Configuration" on page 4-1 for details on setting the DeviceNet network configuration using these tools.

### 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-D, during configuration of the network.

The latest version of the EDS file for the X-gateway can be downloaded from HMS Industrial Networks' web site, or received by contacting HMS Industrial Networks.

## *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 X-gateway. As slaves are added to the X-gateway'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 or Modbus/TCP scanner. The Input and Output table formats should be planned and documented to ensure the Ethernet 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 or Modbus/TCP scanner and, hence, the size of the I/O tables is restricted as explained below.

- The Input table (data coming from DeviceNet devices) size cannot be larger than 496 bytes.
- The Output table (data being sent to DeviceNet devices) size cannot be larger than 492 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; hence it will not provide details on using the tool.

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

# Quick Start

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

## *DeviceNet Network Configuration*

The configuration of the DeviceNet network interface involves using a DeviceNet node commissioning tool to set the X-gateway'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-D.

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

### Node Commissioning with BWConfig

See "DeviceNet I/O Configuration" on page 3-17 for an explanation of the DeviceNet configuration parameters and how they are set using BWConfig.

If BWConfig is used for DeviceNet node commissioning, skip to DeviceNet I/O Configuration below.

### 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 X-gateway un-powered, connect the DeviceNet network cable to the DeviceNet connector of the module. (See “Connecting DeviceNet” on page 2-4)
- Power up the X-gateway.

Step 2: Locate the module on the network.

- RSNetWorx 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 RSNetWorx has been used with an Anybus X-gateway, the X-gateway’s icon should indicate “Unrecognizable Device”.

Step 3: Register the Anybus X-gateway EDS file in RSNetWorx.

RSNetWorx requires an electronic data sheet (EDS) to recognize a device and its capabilities. An EDS file is available on the HMS Industrial Networks web site. The EDS file must be registered with RSNetWorx 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 X-gateway.
- 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 Anybus X-gateway module should appear.

### Step 4: Put the X-gateway in Idle Mode

The X-gateway powers up in Idle mode, and is controlled by the Ethernet scanner through the output command register. Make sure that the module is in Idle mode by verifying that the X-gateway 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 X-gateway icon.
- Enter the desired MAC address and/or baud rate, then click the *Apply* button.

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

**Note:** When the MAC ID is changed, the X-gateway's I/O configuration is cleared.

### Step 6: Enabling the Autobaud Option

If it is desirable to have the X-gateway automatically determine the network baud rate, the Autobaud option must be enabled. (see the explanation of this option in "DeviceNet I/O Configuration" on page 3-17.)

- Highlight the X-gateway 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 X-gateway 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.

### Step 7: Setting the I/O Byte Swapping Option

If it is desirable to have the X-gateway byte swap each 16-bit word in the I/O table, the Byte Swap option must be enabled. (see the explanation of this option in "DeviceNet I/O Configuration" on page 3-17.)

- Highlight the X-gateway 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 65h.

- 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 byte swapping, 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 byte swapping option do not take effect until the module has been power cycled.

### Step 8: Setting the Input Data Safe State Option

If the input data safe state is to be “Zero Data”, this option must be configured. (The default value is “Retain Last Value”) (see the explanation of this option in “DeviceNet Network Configuration” on page 3-15.)

- Highlight the X-gateway 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 68h.

- 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 set the safe state to “Zero Data”, or a “00” to set it to “Retain Last State”. 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 this option do not take effect until the module has been power cycled.

### Node Commissioning with NetTool-DN-D

Step 1: Connect the module to your network

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

Step 2: Locate the module on the network.

- Start NetTool-DN-D on the PC.
- NetTool-DN-D starts up and displays a screen prompting for a network name. Enter a name such as “X-gateway” 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-D RS-232 adapter.
- Set the DeviceNet baud rate to125K baud. Set the MAC ID to a value that will not conflict with devices already on the network. (including the X-gateway)
- Click *Go Online*. A confirmation message indicating that the adapter has gone online should appear. Click *Ok*.
- NetTool-DN-D 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-D has been used with a X-gateway, the X-gateway’s icon will indicate “No EDS file registered for this device”.

Step 3: Register the X-gateway EDS file with NetTool-DN-D

NetTool-DN-D requires an electronic data sheet (EDS) to recognize a device and its capabilities. An EDS file is available on the HMS Industrial Networks web

site. The EDS file must be registered with NetTool-DN-D 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 X-gateway.
- Click *Open*.
- Select the *Tools* menu option, then *Update*, and click on the network name. The X-gateway icon should be properly displayed on the network screen.

### Step 4: Put the X-gateway in Idle Mode

The X-gateway powers up in Idle mode, and is controlled by the Ethernet scanner through the output command register. Make sure that the module is in Idle mode by verifying that the X-gateway 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 X-gateway 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 Anybus X-gateway will automatically reset if a new MAC ID is entered.

**Note:** When the MAC ID is changed, the Anybus X-gateway's I/O configuration is cleared.

### Step 6: Configuring the Other Options

The remaining DeviceNet configuration parameters are set through the parameter editor in NetTool-DN-D. (See the explanation of these parameters in “DeviceNet I/O Configuration” on page 3-17.)

- Highlight the X-gateway 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.
- The Baud Rate, Autobaud, and I/O Byte Swapping and Input Safe State parameters can be set by clicking on each parameter and selecting the desired value from the drop down box.
- Click *Download* to send the changes to the X-gateway.
- Click *Close*.

**Note:** Changes to the DeviceNet configuration parameters do not take effect until the module has been power cycled.

**Note:** If the Anybus X-gateway 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 X-gateway'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-D.

### I/O Configuration Using RSNetWorx

Step 1: Set up X-gateway module's DeviceNet scan list

In most cases it will be necessary to return the X-gateway to Idle mode as described in "Step 4: Put the X-gateway in Idle Mode" on page 4-3. 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 X-gateway 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 X-gateway 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 X-gateway. This is the format of the input data that will be sent to the EtherNet/IP scanner. See "I/O Mapping" on page 3-17.**

- 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 X-gateway 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 X-gateway. This is the format of the output data that will be sent to the X-gateway from the EtherNet/IP scanner. See “I/O Mapping” on page 3-17**

- Click the *Apply* button, and *Yes* to download the scanlist to the X-gateway.

- The X-gateway 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 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-D

Step 1: Set up X-gateway module's DeviceNet Scanlist

In most cases it will be necessary to return the X-gateway to Idle mode as described in "Step 4: Put the X-gateway in Idle Mode" on page 4-8. Once in Idle mode the following steps should be taken to configure the scan list.

- From the network display screen right click on the X-gateway 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 X-gateway. This is the format of the input data that will be sent to the EtherNet/IP scanner. See "I/O Mapping" on page 3-17.**
- 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 X-gateway. This is the format of the output data that will be sent to the X-gateway from the EtherNet/IP scanner. See "I/O Mapping" on page 3-17**
- Select the *Scanlist* tab, and click the *Download* button to download the scanlist to the X-gateway.
- The X-gateway 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 for simplicity. In some cases, the user may wish to organize the I/O data in other ways. See the NetTool-DN-D manual for complete details on how to accomplish this.

## *Ethernet Network Configuration*

### Ethernet Network Configuration using BWConfig

See section “Ethernet Network Configuration” on page 3-6 for an explanation of Ethernet network configuration using BWConfig.

### Ethernet Network Configuration using Arp/Ping

Step 1: Connect the X-gateway Module to Your Network

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

Step 2: Configure the X-gateway 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 Anybus X-gateway 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.)

  *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 X-gateway 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>Anybus X-gateway</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 Anybus Ethernet to DeviceNet X-gateway acts as a DeviceNet Master or a slave. The X-gateway, 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 second master.

## Configuration

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

## Automatic Baud Rate Detection

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

### *Scan Cycles*

The X-gateway 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 any devices 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 X-gateway 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 X-gateway 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.

### *I/O Table Byte Swapping*

The X-gateway provides an I/O byte swapping option. If the option is enabled, the data in the I/O tables is byte swapped on 16-bit boundaries. This is very useful if the Ethernet protocol being used is Modbus/TCP since Modbus assumes the byte ordering is opposite of that of DeviceNet.

Do not enable byte swapping if the Ethernet protocol being used is EtherNet/IP.

### *Input Data Safe State*

The X-gateway provides the option of configuring how the DeviceNet input data will be set when a DeviceNet slave connection faults. The safe state behavior may be configured as either “Maintain Last State” or “Zero Data”. If the option is set to Maintain Last State, the input data associated with a DeviceNet slave will be frozen to the last value received from the slave prior to the connection fault. If the option is set to Zero Data, the input data associated with the slave will be set to 0 when the connection is faulted. Only the input data associated with the particular slave’s I/O mapping configuration will be affected, all other non-faulted slaves’ data will continue to update normally. Note that this is a global setting and all slave connections will be treated in the same manner.

### *Proxy for Group 2 Only Devices*

The X-gateway 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 X-gateway will be intercepted and relayed to the slave.

### *Quick Connect Feature*

The X-gateway supports DeviceNet Quick Connect. Quick Connect is a special, shortened establishment procedure for connections to slaves. Quick Connect can be used in applications where the normal delay between when a slave comes online and the scanner establishes a connection cannot be tolerated. Quick Connect is enabled on a per-slave basis using the RSNetworx Tools->Quick Connect menu.

### *Active Node List*

The X-gateway monitors the DeviceNet network and tracks the online/offline state of all nodes on the network. The current state of each node is kept in the Active Node List which can be accessed the DeviceNet object instance attribute 13. The CIP bridging utilizes the Active Node List to determine whether a target node is online. If the target node is not online, the CIP bridging functions will immediately return an error response to the requesting node. The Active Node List monitoring and bridging functionality can be disabled by setting DeviceNet object instance attribute 102 to 1.

### *Run/Idle Mode*

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

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

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

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

The module automatically reverts to Idle mode when the Ethernet I/O messaging stops. If the Ethernet protocol is EtherNet/IP, this is handled when the I/O connection closes. If the protocol is Modbus/TCP, this is handled when no requests are received within the configured timeout period.

If no Ethernet I/O messaging is active, the Run/Idle mode of the X-gateway 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 X-gateway 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.

### 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 X-gateway. The slave's configuration is stored in the X-gateway's non-volatile memory. Whenever the X-gateway 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.

The X-gateway module will hold up to 130,560 bytes (approximately 128K) of configuration recovery data.

**Note:** RSNetworx v7.0 or later is required to support the full 128K bytes of ADR configuration recovery data; earlier versions support up to 64K bytes of data. NetTool-DN-D, as of v3.3.1, supports up to 64K bytes of configuration recovery data.

### *Interaction with I/O Tables*

The DeviceNet interface in the X-gateway 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. Transmission of data on Change of State (COS) connections is triggered when new output data is provided by the Ethernet interface in the region mapped by the connection.

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 Anybus X-gateway 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. The X-gateway 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. The X-gateway contains 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 interchanged with the term I/O Messaging.

A one time transport of a data item where both parties do not know in advance 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.

The X-gateway will handle up to 64 concurrent unconnected transactions. Up to 16 class 3 (messaging) connections are supported.

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 X-gateway, one for EtherNet/IP and one for DeviceNet.

### *I/O Messaging*

The X-gateway 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 X-gateway and becomes the EtherNet/IP Input I/O to the EtherNet/IP scanner. EtherNet/IP Output data from the scanner is stored in the X-gateway's Output Table (OUT) and sent to the DeviceNet slaves which consume it.

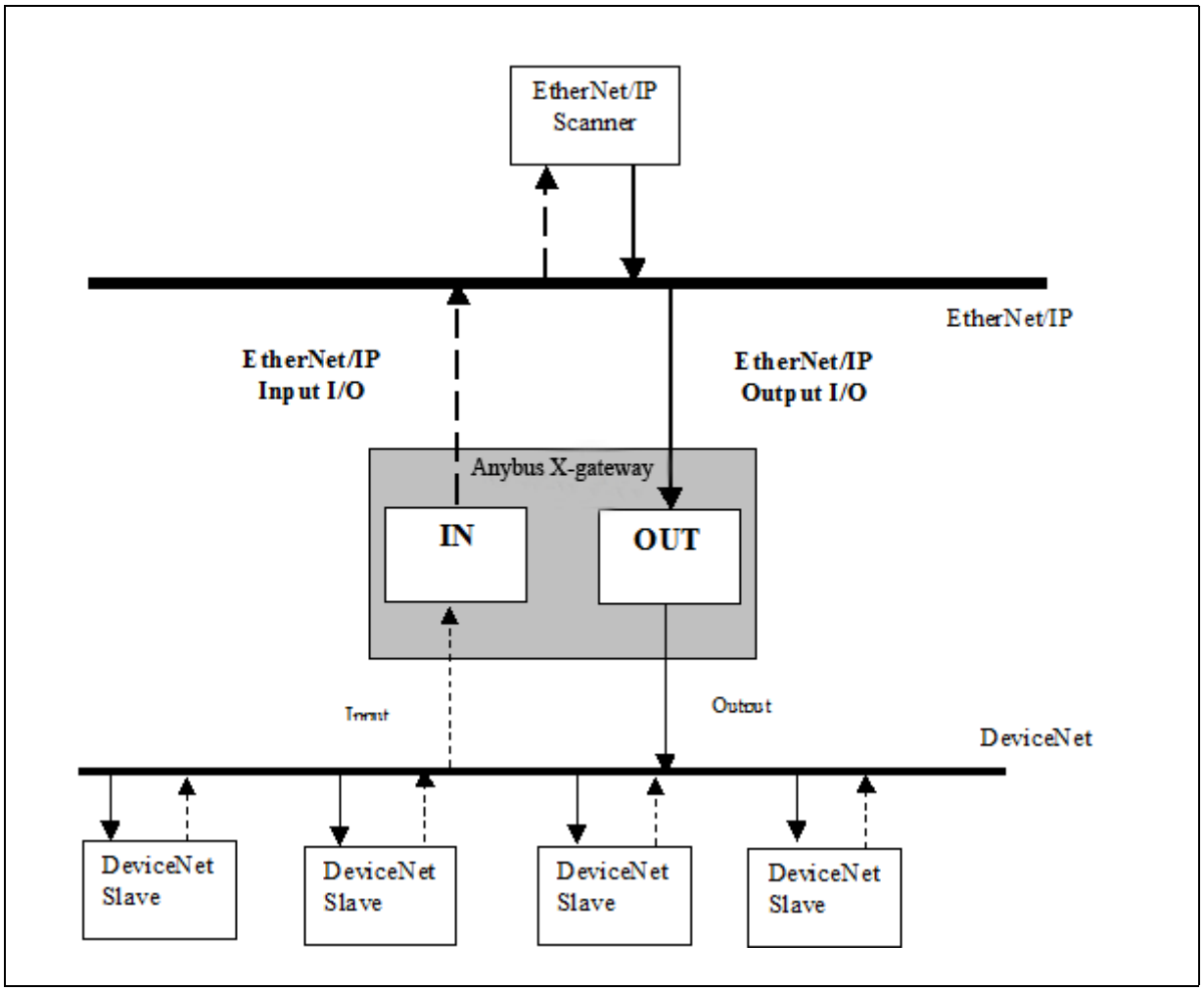



**Figure 6-1 Anybus X-gateway I/O Transfer**

## *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 X-gateway.

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

| Assembly Instance | Description | Size in Bytes |
|---|---|---|
| 100 | Input | 500 max |
| 101 | Status | 128 |
| 150 | Output | 496 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 |
|---|---|---|---|
| 198 | Input-Only Heartbeat | 0 | Output connection point for input-only connections. |
| 199 | Listen-Only Heartbeat | 0 | Output connection point for listen-only connections. |
| 100 | Input | 4-500 | Input connection point. |
| 101 | Status | 128 | Input connection point. |
| 150 | Output | 8-500 | 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 DeviceNet slave input data.

| Byte Offset | Size in Bytes | Description |
|---|---|---|
| 0 | 4 | Status register. |
| 4 | Up to 496 | DeviceNet slave input data. |

**Table 6-3 Input Assembly Format**

The DeviceNet slave input data format and content is determined by the DeviceNet scanner configuration. The DeviceNet slave data appears in the table as it is mapped from the DeviceNet input connections. The DeviceNet slave input data in the assembly is 496 bytes long; however, only the size of the configured slave input data 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 | X-gateway is in Run mode. (Cleared if in Idle mode.) |
| 1 | X-gateway is faulted. |
| 2 | DeviceNet network interface is disabled. |
| 3 | Communication has failed with at least 1 DeviceNet slave. |
| 4 | At least 1 DeviceNet slave has failed verification. |
| 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 command register followed by the DeviceNet slave output data.

| Byte Offset | Size in Bytes | Description |
|---|---|---|
| 0 | 4 | Command register. |
| 4 | Up to 492 | DeviceNet slave output data. |

**Table 6-5 Output Assembly Format**

The DeviceNet slave output data format and content is determined by the DeviceNet scanner configuration. The DeviceNet slave data appears in the table as it is mapped to the DeviceNet output connections. The DeviceNet slave output data in the assembly is 492 bytes long; however, only the size configured for the output data will be used, the remaining space will be ignored.

**Note:** EtherNet/IP I/O connections append a 32-bit Run/Idle register at the front of the output data. The actual output data transferred in the I/O connection includes this extra 4 bytes at the front of the output assembly described above.

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 Run/Idle register to determine the run mode of the X-gateway. Both bits must be set for the X-gateway to be in Run mode; otherwise the module will be in Idle mode. |
| 1 | Fault. Sets a fault condition in the X-gateway. |
| 2 | Disable DeviceNet network. |
| 3 | Not used. |
| 4 | Reset the X-gateway module. |
| 5-31 | Not used. |

**Table 6-6 Output Command Register Bit Definitions**

### Status Assembly

The status assembly is a collection of status and diagnostic information for the X-gateway 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 X-gateway was powered up. |
| 4 | 8 | 64-bit Bit-string | Faulted Node Table | Indicates which DeviceNet slaves are faulted.<br>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.<br>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.<br>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 X-gateway's scan list. Each bit represents a device at the corresponding MAC ID. If the bit is set, that device is being actively scanned by the X-gateway's DeviceNet master. |

**Table 6-7 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.<br>If there are no faults, the display indicates the X-gateway MAC ID and its Run/Idle status.<br>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 10-6 for a list of error codes. |
| 40 | 1 | USINT | X-gateway MAC ID | The DeviceNet MAC ID of the X-gateway. |
| 41 | 1 | USINT | Scanner Status | The current status of the DeviceNet scanner. See Table 10-6, “Node Status Codes,” on page 10-6 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 X-gateway scanner itself.<br>If there are no faulted nodes, both the scrolling address and status are set to 0.<br>The scrolling fields change once a second. |
| 43 | 1 | USINT | Scrolling Status | |
| 44 | 20 | USINT[20] | Reserved | |

**Table 6-7 Status Assembly Format (Continued)**

| Byte Offset | Size in Bytes | Data Type | Name | Description |
|---|---|---|---|---|
| 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.<br>The X-gateway scanner status appears at the entry associated with the X-gateway MAC ID.<br><br>A non-zero status indicates that there is an issue with the associated node. A status of 0 indicates “OK” and is used for nodes both in and out of the scan list.<br><br>See Table 10-6, “Node Status Codes,” on page 10-6 for a list of status codes. |

**Table 6-7 Status Assembly Format (Continued)**

### *I/O Data Summary*

The following diagram illustrates how the various components of the input data are used to create the input assembly and connection data accessible from EtherNet/IP.

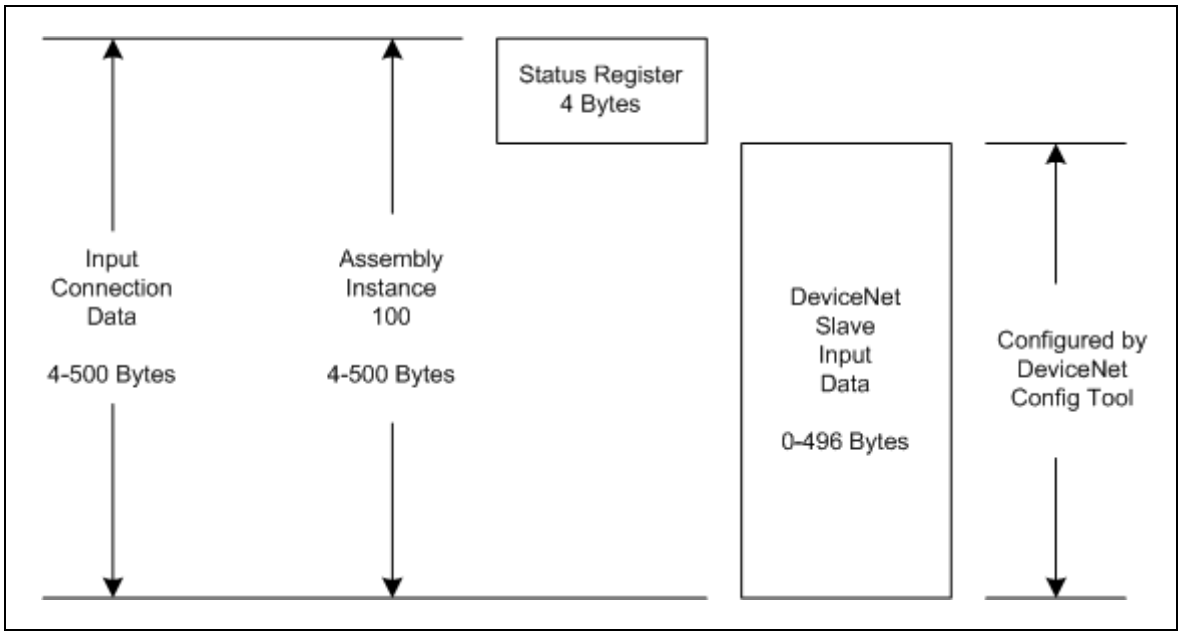



**Figure 6-2 Input Data Association**

The following diagram illustrates how the various components of the output data are used to create the output assembly and connection data accessible from EtherNet/IP.

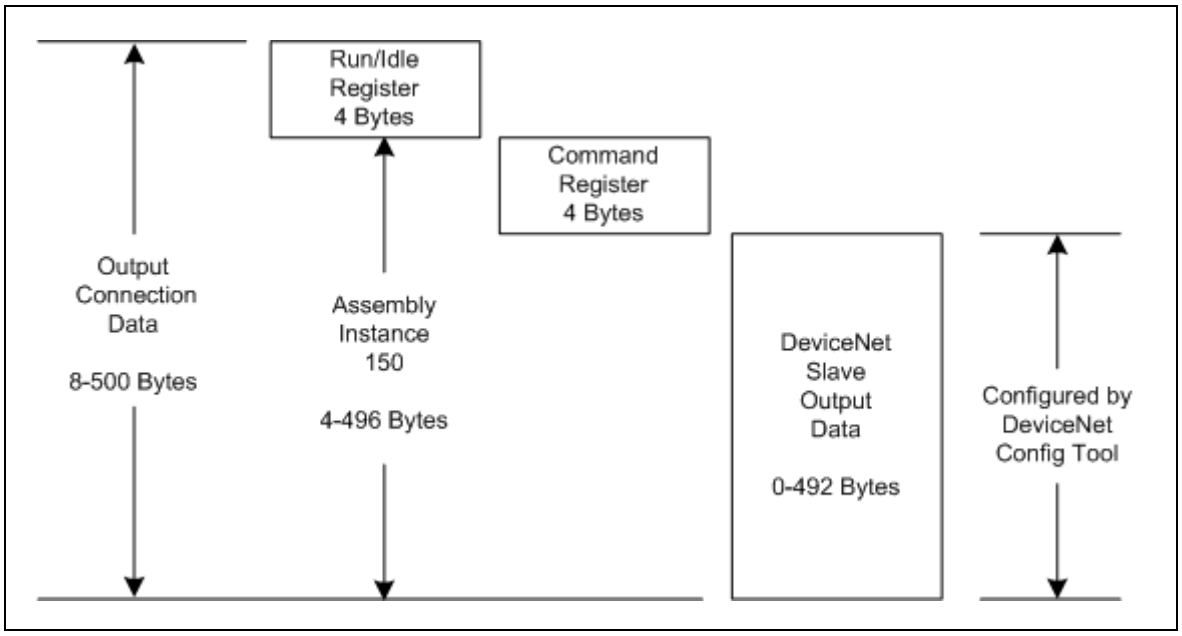



**Figure 6-3 Output Data Association**

### *Notes About Using ControlLogix I/O Connections*

When configuring I/O connections between a Rockwell Automation ControlLogix EtherNet/IP scanner and the X-gateway, 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 connection output size should be set to the configured DeviceNet slave output data size plus 4 bytes for the command register (up to a maximum of 496 bytes inclusive). The Run/Idle header is automatically added by the controller and does not come into play in the size. The connection input size should be set to the configured DeviceNet slave input data size plus 4 bytes for the input status register (up to a maximum of 500 bytes inclusive).

The status assembly may also be monitored by configuring the generic device using a “with status” comm format.

The X-gateway does not support a configuration assembly. The configuration instance in the device configuration may be set to any number since it will be ignored. Set the configuration assembly size to 0.

The figure below shows a typical ControlLogix device configuration.

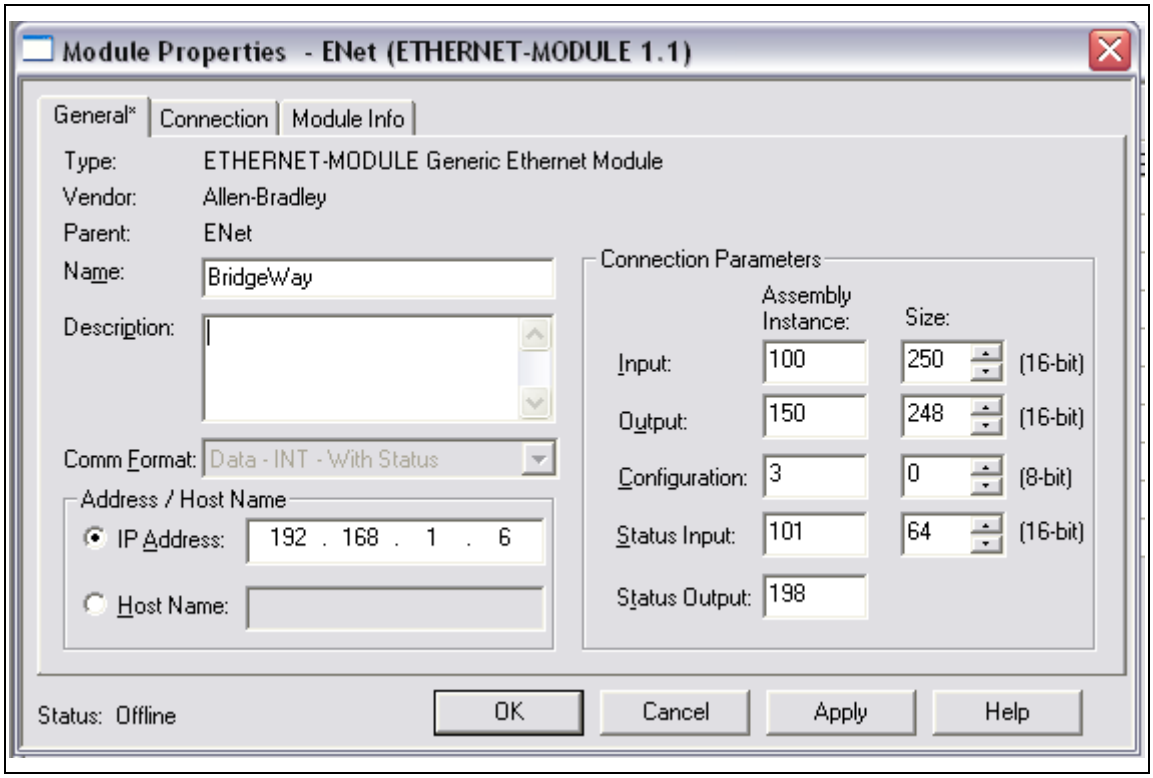



**Figure 6-4 ControlLogix Configuration**

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

**Note:** The Anybus X-gateway does not support message routing from DeviceNet to EtherNet/IP.

### Port Addresses

| Port Address | Network |
|---|---|
| 2 | EtherNet/IP |
| 3 | DeviceNet |

**Table 6-8 Port Addresses**

# Modbus/TCP Interface

The Anybus X-gateway 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 X-gateway.

| 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 X-gateway is unable to service the Modbus request that was received. The following exception codes will be used by the X-gateway.

| 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 X-gateway's Input (IN) and Output (OUT) areas are set to a maximum size of 500 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.

**Note:** Input Status and Coil bits are mapped MSB first. i.e. Coil 1 corresponds bit 15 of the associated register.

| Input Register | Input Status Bit Address | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **...** | **1** | **0** |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 15 | 16 |
| 2 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | ... | 31 | 32 |
| ..... | | | | | | | | | | |
| 250 | 3985 | 3986 | 3987 | 3988 | 3989 | 3990 | 3991 | ... | 3999 | 4000 |

**Table 7-3 Input Addressing**

| Holding Register | Coil Bit Address | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **...** | **1** | **0** |
| 1025 | 16385 | 16386 | 16387 | 16388 | 16389 | 16390 | 16391 | ... | 16399 | 16400 |
| 1026 | 16401 | 16402 | 16403 | 16404 | 16405 | 16406 | 16407 | ... | 16415 | 16416 |
| ... | | | | | | | | | | |
| 1274 | 20369 | 20370 | 20371 | 20372 | 20373 | 20374 | 20375 | ... | 20383 | 20384 |

**Table 7-4 Output Addressing**

| Input Register | Input Status Bit Address | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **...** | **1** | **0** |
| 257 | 4097 | 4098 | 4099 | 4100 | 4101 | 4102 | 4103 | ... | 4111 | 4112 |
| 258 | 4113 | 4114 | 4115 | 4116 | 4117 | 4118 | 4119 | ... | 4127 | 4128 |
| ... | | | | | | | | | | |
| 320 | 5105 | 5106 | 5107 | 5108 | 5109 | 5110 | 5111 | ... | 5119 | 5120 |

**Table 7-5 Status Addressing**

### Bit Addressing Examples

- To reference the first bit of the Input Table use Input Status bit address 16.
- To reference the 15th bit of the Input Table use Input Status bit address 2
- To reference the first bit of the Output Table use Coil bit address 16400.
- To reference the 15th bit of the Output Table use Coil bit address 16386.

### Word Addressing Examples

- To reference the first word of the Input Table use Input Register address 1.
- To reference the 10th word of the Input Table use Input Register address 10
- To reference the first word of the Output Table use Holding Register address 1025.
- To reference the 100th word of the Output Table use Holding Register address 1124.

## *I/O Data Content*

### Input Table

The input table contains a 32-bit status register followed by the DeviceNet slave input data.

| Modbus Input Register | Size in Words | Description |
|---|---|---|
| 1 | 2 | Status register. |
| 3 | Up to 248 | DeviceNet slave input data. |

**Table 7-6 Input Table Format**

The DeviceNet slave 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 DeviceNet slave input data in theinput table is 248 words long; however, only the size of the configured DeviceNet slave 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 | X-gateway is in Run mode. (Cleared if in Idle mode.) |
| 1 | X-gateway is faulted. |
| 2 | DeviceNet network interface is disabled. |
| 3 | Communication has failed with at least 1 DeviceNet slave. |
| 4 | At least 1 DeviceNet slave has failed verification. |
| 5 | DeviceNet network interface is bus-off. |
| 6 | Duplicate MAC ID error. |
| 7 | No DeviceNet power. |
| 8-31 | Not used. |

**Table 7-7 Input Status Register Bit Definitions**

### Output Table

The output table contains a 32-bit command register followed by the DeviceNet slave output data.

| Modbus Holding Register | Size in Words | Description |
|---|---|---|
| 1025 | 2 | System Run/Idle register |
| 1027 | 2 | Command register. |
| 1029 | Up to 246 | DeviceNet slave output data. |

**Table 7-8 Output Table Format**

The DeviceNet slave 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 DeviceNet slave output data in the table is 246 words long; however, only the size configured for the DeviceNet slave output data will be used, the remaining space will be ignored.

The System 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 X-gateway. Both bits must be set for the X-gateway to be in Run mode; otherwise the module will be in Idle mode. |
| 1-31 | Not used. |

**Table 7-9 System 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 Run/Idle register to determine the run mode of the X-gateway. Both bits must be set for the X-gateway to be in Run mode; otherwise the module will be in Idle mode. |
| 1 | Fault. Sets a fault condition in the X-gateway. |
| 2 | Disable DeviceNet network. |
| 3 | Not used. |
| 4 | Reset the X-gateway module. |
| 5-31 | Not used. |

**Table 7-10 Command Register Bit Definitions**

### Status Data Table

The status data table is a collection of status and diagnostic information for the X-gateway DeviceNet interface. The information in the table is updated approximately once a second.

| Modbus Input Register | Size in Words | Data Type | Name | Description |
|---|---|---|---|---|
| 257 | 2 | UDINT | Scan Counter | The number of DeviceNet I/O scans that have taken place since the X-gateway was powered up. |
| 259 | 4 | 64-bit Bit-string | Faulted Node Table | Indicates which DeviceNet slaves are faulted.<br>Each bit represents the status of the slave at the corresponding MAC ID. |
| 263 | 4 | 64-bit Bit-string | Auto Verify Error Table | Indicates which DeviceNet slaves are the incorrect device type.<br>Each bit represents the status of the slave at the corresponding MAC ID. |
| 267 | 4 | 64-bit Bit-string | Idle Node Table | Indicates which DeviceNet slaves are in Idle mode.<br>Each bit represents the status of the slave at the corresponding MAC ID. |
| 271 | 4 | 64-bit Bit-string | Active Node Table | Indicates which DeviceNet nodes are configured in the X-gateway's scan list. Each bit represents a device at the corresponding MAC ID. If the bit is set, that device is being actively scanned by the X-gateway's DeviceNet master. |
| 275 | 2 | ASCII[4] | Status Display | Mimics a 4-character alpha-numeric display.<br>If there are no faults, the display indicates the X-gateway MAC ID and its Run/Idle status.<br>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 6 for a list of error codes. |

**Table 7-11 Status Data Format**

| Modbus Input Register | Size in Words | Data Type | Name | Description |
|---|---|---|---|---|
| 277 | 1 | USINT | X-gateway MAC ID | The DeviceNet MAC ID of the X-gateway followed by the current status of the DeviceNet scanner. See Table 10-6, “Node Status Codes,” on page 6 for a list of status and error codes. |
| | | USINT | Scanner Status | |
| 278 | 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 X-gateway scanner itself.<br>If there are no faulted nodes, both the scrolling address and status are set to 0.<br>The scrolling fields change once a second. |
| | | USINT | Scrolling Status | |
| 279 | 10 | USINT[20] | Reserved | |
| 289 | 32 | 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.<br>The X-gateway scanner status appears at the entry associated with the X-gateway MAC ID.<br><br>A non-zero status indicates that there is an issue with the associated node. A status of 0 indicates “OK” and is used for nodes both in and out of the scan list.<br><br>See Table 10-6, “Node Status Codes,” on page 6 for a list of status codes. |

**Table 7-11 Status Data Format (Continued)**

### *I/O Data Summary*

The following diagram illustrates how the various components of the input data are used to create the input data accessible from Modbus/TCP.

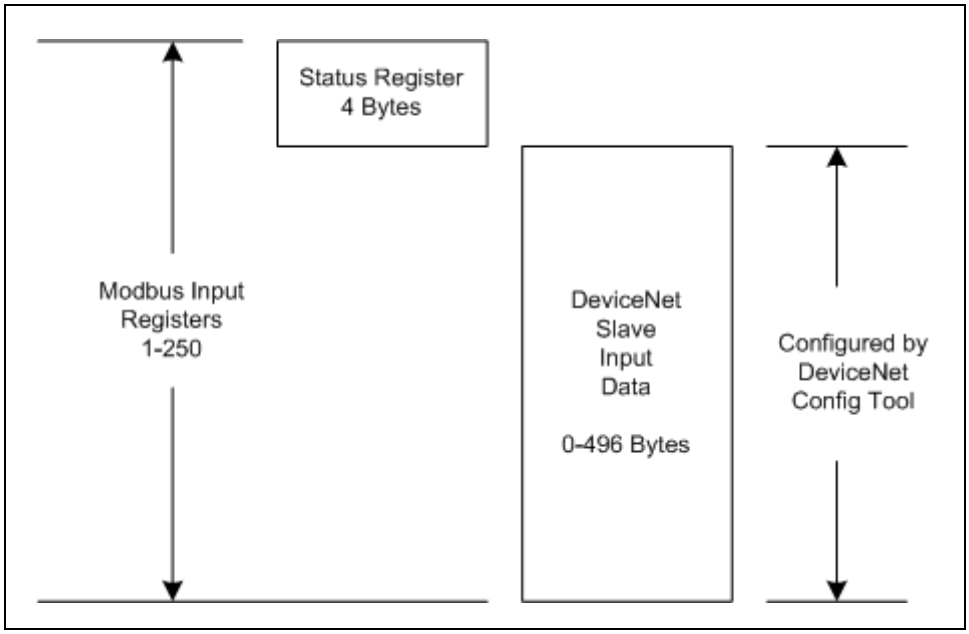



**Figure 7-1 Input Data Association**

The following diagram illustrates how the various components of the output data are used to create the output data accessible from Modbus/TCP.

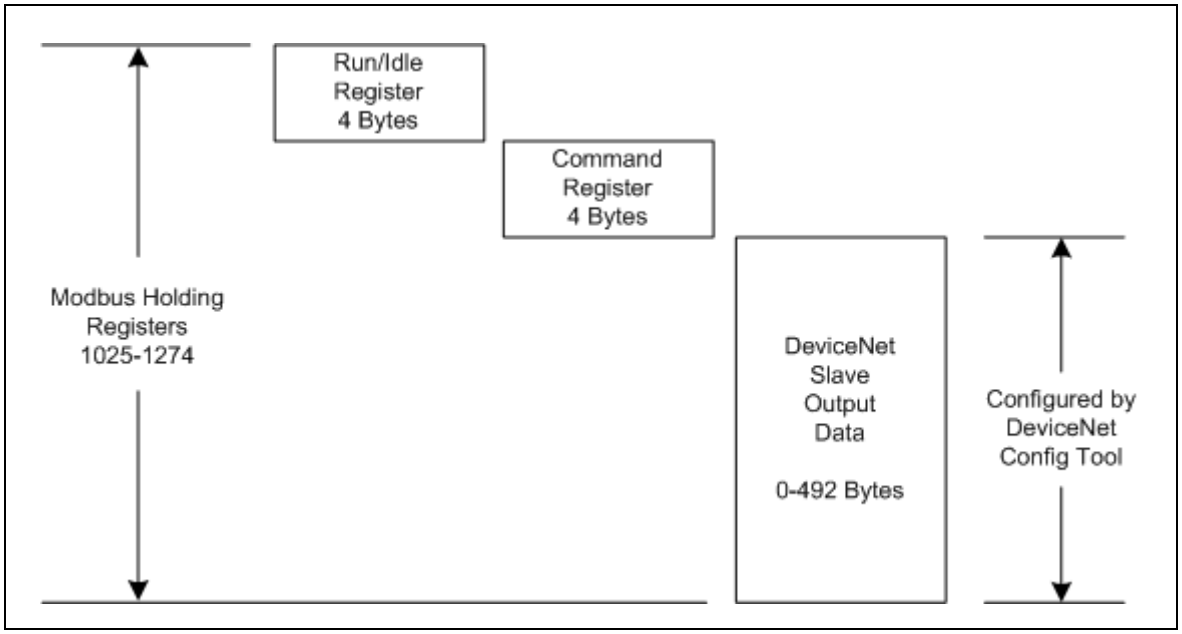



**Figure 7-2 Output Data Association**

### *I/O Data Format*

The X-gateway 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).

The I/O Byte Swap option will aid this issue by swapping the bytes on 16-bit boundaries. However, the user is still responsible for knowing where in the I/O tables DeviceNet data has been mapped.

# 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 MB 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 8-9.

### Normal mode

The Anybus X-gateway 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 X-gateway 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 X-gateway 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.

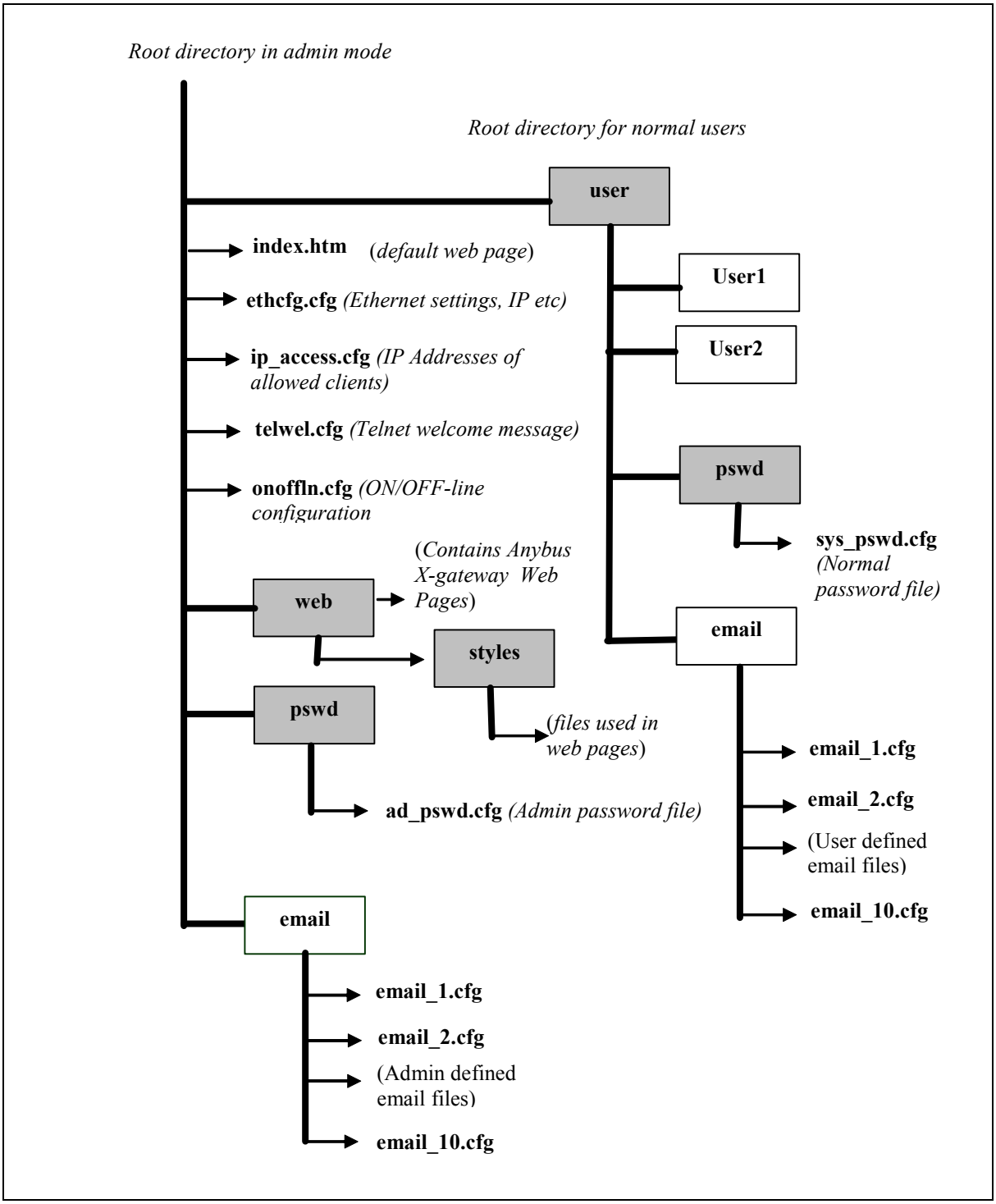



**Figure 8-1 File System Directory Structure**

### *Default Files*

The following directories are already created on the X-gateway when first powered;

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

The following files are also on the X-gateway;

\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 X-gateway 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 X-gateway contains an “index.htm” file replacing its virtual file counterpart to provide a link to the X-gateway’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]<br>10.10.12.212 | IP address |
| [Subnet mask]<br>255.255.255.0 | Subnet mask |
| [Gateway address]<br>0.0.0.0 | Gateway address |
| [SMTP address]<br>0.0.0.0 | SMTP address – This must be configured in order to send emails |
| [SMTP username]<br>username | The user name required by the SMTP server. Do not include this parameter if the server does not require a username and password. |
| [SMTP password]<br>password | The password required by the SMTP server. Do not include this parameter if the server does not require a username and password. |
| [DNS1 address]<br>0.0.0.0 | Needed to be able to resolve host names. |
| [DNS2 address]<br>0.0.0.0 | Needed to be able to resolve host names. |
| [Domain name]<br>domain | The default domain name for not fully qualified host names. |
| [DHCP/BOOTP]<br>OFF | DHCP/BootP 'ON'-Enabled, 'OFF'-Disabled |

| | |
|---|---|
| [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.

## *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).

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<br>OUT | Source area in Input/Output |
| Offset | a hexadecimal (0xN) or decimal value | Source Offset in Input/Output |
| Type | byte<br>word<br>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 | <<br>=<br>> | 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

### *Anybus X-gateway Web Page Files*

The X-gateway 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 X-gateway'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 X-gateway'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 X-gateway'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 X-gateway’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 X-gateway’s scanlist of whether the X-gateway 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 X-gateway’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 X-gateway’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 Anybus X-gateway 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.

df

Usage:

df

Displays file system information.

## *HTTP Server*

The module features a complete HTTP (web) server with Server Side Include (SSI) functionality. Server Side Includes are commands to the web server embedded in the HTML code. When the web server encounters the commands, the command is executed and the results of the command are inserted into the web page. SSI commands allow easy access to the IN and OUT data areas of the X-gateway 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 |
| configform2.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 the 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 8-9.

## *SSI Functionality*

SSI functionality makes it possible to make web pages interact with module data. e.g. 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 9-25). The following are the available SSI functions.

### Ethernet Address Display Functions

DisplayIP

This function returns the currently used IP address.

*Syntax:*

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

DisplayMacId

This function returns the MAC ID in the format xx:xx:xx:xx:xx:xx.

*Syntax:*

```
<?--#exec cmd_argument=’DisplayMacId’-->
```

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'-->
```

DisplayDNS1

This function returns the address of the primary DNS server.

*Syntax:*

```
<?--#exec cmd_argument='DisplayDNS1'-->
```

### DisplayDNS2

This function returns the address of the secondary DNS server.

*Syntax:*

```
<?--#exec cmd_argument='DisplayDNS2'-->
```

### DisplayHostName

This function returns the host name.

*Syntax:*

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

### DisplayDomainName

This function returns the default domain name.

*Syntax:*

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

### DisplaySMTPUser

This function returns the username used for SMTP authentication.

*Syntax:*

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

### DisplaySMTPPswd

This function returns the password used for SMTP authentication.

*Syntax:*

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

### 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 9-23.

### 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.

| Charac-ter | 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.dddddde+ -xx or [-]m.ddddddE+-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.dddddde+ -xx or [-]m.ddddddE+-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 OUTarea |
| 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 accessed by the web page is in the same format as the data accessed via the Ethernet network via EtherNet/IP or Modbus/TCP. 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 depending on the configuration of the Swap I/O Bytes parameter in the Ethernet configuration.

### 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 |
| Arg*n* | Specifies where to write the data |
| ErrVal*n* | 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 *offset* in the OUT area |
| OutWriteWord(offset) | Writes a word (short) to position *offset* in the OUT area |
| OutWriteLong(offset) | Writes a long to position *offset* in the OUT area |
| OutWriteString(offset) | Writes a string to position *offset* in the OUT area |
| OutWriteFloat(offset) | Writes a floating point (float) value to position *offset* 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 9-23“.

**Note:** The I/O data accessed by the web page is in the same format as the data accessed via the Ethernet network via EtherNet/IP or Modbus/TCP. 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 depending on the configuration of the Swap I/O Bytes parameter in the Ethernet configuration.

### 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

SaveDataToFile

This SSI function saves the data of a passed form to a file. The Object Name parameter is optional and, if specified, only the data from that object will be stored. If no object is specified, the data from all objects in the form will be stored. The contents can either be Appended to the file or Overwrite the current contents of the file.

*Syntax:*

```
<?--#exec cmd_argument='SaveDataToFile(
                "File name", "Object name", [Append|Overwrite])'-->
```

*Default output*:

Success - Form data saved to file

Failure - Failed to save form data

### 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 2 times a second. The IP address to a SMTP (mail) server and any required username and password must be configured (See“‘ethcfg.cfg’” on page 8-7). 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
- DisplayMACID
- DisplaySubnet
- DisplayGateway
- DisplayDNS1
- DisplayDNS2
- DisplayHostName
- DisplayDomainName
- DisplayEmailServer
- DisplayDHCPState
- DisplaySMTPUser
- DisplaySMTPPswd
- 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>Anybus Ethernet to DeviceNet X-gateway</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

## *Anybus X-gateway LEDs*

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

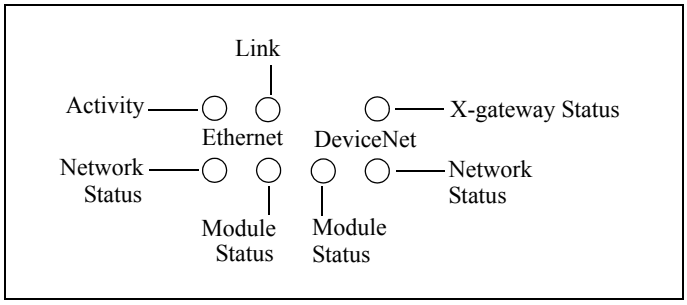



**Figure 10-1 Anybus X-gateway LEDs**

### X-gateway 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 Anybus X-gateway Status LED States**

Major unrecoverable faults are indicated by a series of green and red flashes. If the X-gateway 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 X-gateway is on the DeviceNet network and communicating with at least 1 device. |
| Flashing Green | Online, no communication | The X-gateway 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.<br><br>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 the requires user intervention has been detected. Correct the problem and reset the X-gateway. |
| Flashing Red | Recoverable fault. | A fault that can be corrected and does not require a X-gateway 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 controlled by an EtherNet/IP scanner in Run mode. |
| Flashing Green | Standby | The module is not controlled by a scanner in Run mode. Note that this is the normal state when using Modbus/TCP masters. |
| Solid Red | Unrecoverable fault | A fault the requires user intervention has been detected. Correct the problem and reset the X-gateway. |
| Flashing Red | Recoverable fault. | A fault that can be corrected and does not require a X-gateway 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. *(Not affected by Modbus/TCP connections.)* |
| Flashing Green | Network OK | There are no active connections. *(Not affected by Modbus/TCP connections.)* |
| 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 X-gateway 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.

### DeviceNet Who Web Page

The DeviceNet Who page displays all devices that have been detected on the DeviceNet network. Each node that is detected by the X-gateway will be displayed by its device name next to its MAC address. Note that this page does not update automatically and must be refreshed using the Refresh button provided on the page.

### Active Slaves Web Page

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

### Idle Slaves Web Page

The Idle Slaves page indicates which DeviceNet slaves 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 X-gateway and the X-gateway itself are updated on this page.

### Faulted Slaves Web Page

The Faulted Slaves page indicates which DeviceNet slaves 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 X-gateway and the X-gateway itself are updated on this page.

A node is considered faulted if the X-gateway 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 Slaves Web Page

The Invalid Slaves page indicates which DeviceNet slaves 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 X-gateway are updated on this page.

### Slave Status Web Page

The Slave Status page displays the current status of all DeviceNet nodes that are configured as slaves to the X-gateway and the X-gateway itself. The status of each slave is displayed next to the node's MAC ID. Note that only nodes which are configured as slaves to the X-gateway and the X-gateway 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 |
|---|---|
| 0 | Ok.<br><br>Note that this value will be used for nodes both in and out of the scan list to indicate that there is no issue with the node. |
| 60 | Duplicate MAC ID test in progress.<br><br>This status is only used for the local MAC ID. |
| 70 | Duplicate MAC ID failure.<br><br>Another node on the DeviceNet network has the same MAC ID.<br><br>This status is only used for the local MAC ID. |
| 72 | Device communications failed.<br><br>The I/O connections with a slave have timed out. |
| 73 | Incorrect device type.<br><br>Device verification has failed with a slave when attempting to start I/O connections. The level of verification is determined by the scanlist entry. The following identity information may be checked during verification depending on the configuration:<br>Vendor ID<br>Revision<br>Device Type<br>Product Code |
| 75 | CAN network quiet.<br><br>No CAN packets have been received from the network for more than 10 seconds.<br><br>This status is only used for the local MAC ID. |

**Table 10-6 Node Status Codes**

| Status Code | Description |
|---|---|
| 76 | No messages for scanner.<br><br>No CAN packets specifically for the DeviceNet scanner have been received in more than 10 seconds.<br><br>This status is only used for the local MAC ID. |
| 77 | Incorrect connection size.<br><br>The connection size configured in the scanlist entry for the slave does not match the actual required connection size specified by the slave. |
| 78 | No device response.<br><br>A connection could not be established with the slave because it did not respond. |
| 79 | CAN DUP-MAC transmit failure.<br><br>The scanner was unable to transmit the duplicate MAC detection message on the CAN network.<br><br>This status is only used for the local MAC ID. |
| 80 | In Idle mode.<br><br>The device is in Idle mode.<br><br>This status is only used for the local MAC ID. |
| 81 | In Fault mode.<br><br>The device is in Fault mode. Fault mode is set using the Fault bit in the output command register. i.e. Fault mode is controlled by the EtherNet/IP or Modbus/TCP master controller. It can be used to indicate a system fault detected at a higher level. When the device is in Fault mode, all DeviceNet network activity is disabled.<br><br>This status is only used for the local MAC ID. |
| 83 | Error during slave connection initialization.<br><br>An error occurred while creating the I/O connections to the slave (beyond the identity mismatch or I/O size errors). This error is triggered by error responses from the slave during the connection establishment sequence. |

**Table 10-6 Node Status Codes (Continued)**

| Status Code | Description |
|---|---|
| 84 | Slave connection initialization in progress.<br><br>The I/O connection establishment sequence to this slave is in progress. |
| 85 | Incorrect data size received on connection.<br><br>The amount of data received with the last connected message does not match the connection size. |
| 86 | Device went into Idle mode.<br><br>The slave is in Idle mode as indicated by the slave sending idle packets on the input connection. Idle packets are of zero length and are used to keep the connection open, yet not move any data when the device is in Idle mode. |
| 87 | Shared master error.<br><br>The slave scanlist entry is configured for input sharing and the primary master has not made connection to the device. |
| 88 | Shared master choice error.<br><br>The slave scanlist entry is configured for input sharing and the primary master has not made the right type of connections to the device. |
| 89 | ADR error.<br><br>An error occurred during auto device replacement or auto configuration. This is triggered when the slave returns an error response during an auto device replacement or auto configuration message sequence. |
| 90 | CAN network disabled.<br><br>The CAN network has been disabled. The network is disabled by setting the Disable bit in the output command register.<br><br>This status is only used for the local MAC ID. |
| 91 | CAN bus-off.<br><br>Indicates that the CAN controller is in the Bus-Off state.<br><br>This status is only used for the local MAC ID. |

**Table 10-6 Node Status Codes (Continued)**

| Status Code | Description |
|---|---|
| 92 | No DeviceNet power.<br><br>Indicates that there is no network power detected on the DeviceNet network.<br><br>This status is only used for the local MAC ID. |
| 95 | Flash update in progress.<br><br>Indicates that a firmware update is currently in progress.<br><br>This status is only used for the local MAC ID. |

**Table 10-6 Node Status Codes (Continued)**

### *Status Assembly*

The status assembly is an assembly object instance that is accessible from EtherNet/IP both explicitly and using an I/O connection. The status assembly contains current status and diagnostic information pertaining to the X-gateway's DeviceNet interface. See "Status Assembly" on page 6-7 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: 12-30v DC.

Current Requirements: 130-140 mA at 24 VDC.

## *Mechanical Specifications*

### Mechanical Rating

IP20/NEMA 1

### DIN Rail Mount

The X-gateway connects to a DIN 3 rail.

### Dimensions

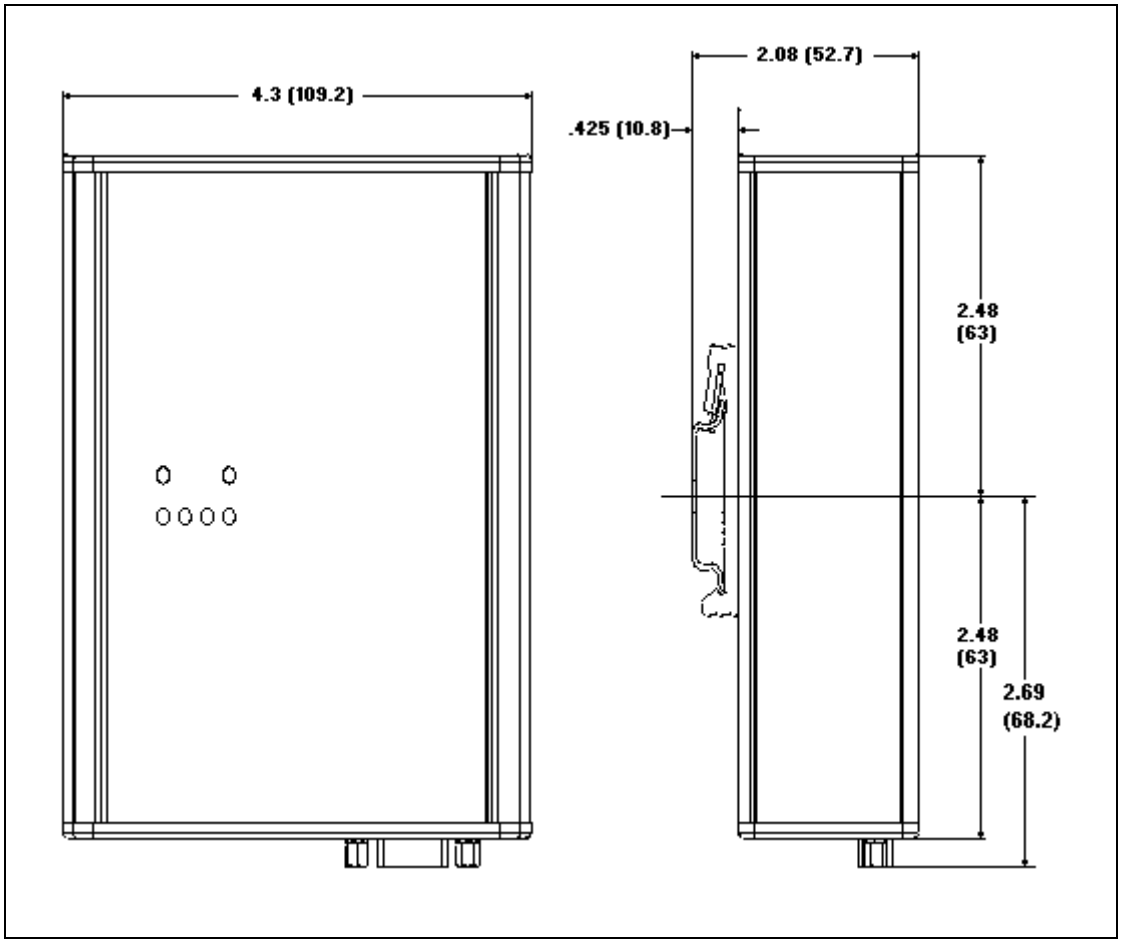



**Figure 11-1 X-gateway Mechanical Dimensions**

## *Data Sizes*

### Input

Maximum 500 bytes input including the status register.

### Output

Maximum 496 bytes output including the command register.

### Status

128 bytes of Status data.

### ADR Configuration Recovery

130,560 bytes of configuration recovery data.

# Connectors

## *Power*

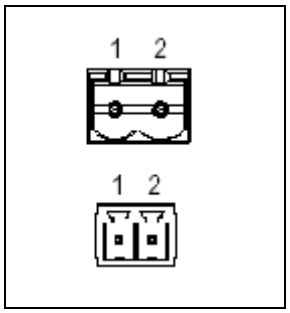


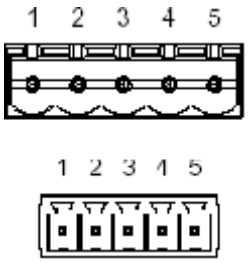

**Figure 12-1 Power Connector**

| 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*

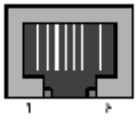



| Pin | Signal |
|---|---|
| 1 | V- |
| 2 | CAN_L |
| 3 | Shield |
| 4 | CAN_H |
| 5 | V+ |

**Table 12-2 DeviceNet Pinout**

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

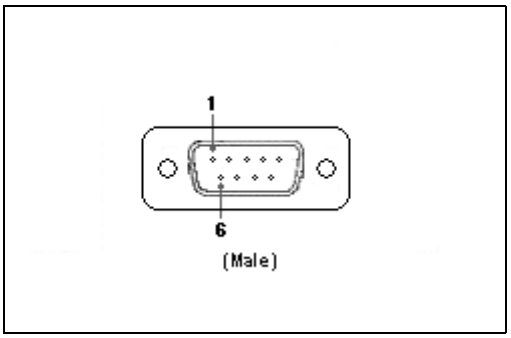
### *Ethernet RJ45*



| 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 | Signal Ground |
| 6 | NC |
| 7 | NC |
| 8 | NC |
| 9 | NC |

**Table 12-4 RS232 9 Pin**

# Support

For technical support, consult the online FAQ (www.anybus.com), or contact the nearest support center:

| Sales | | Support | |
|---|---|---|---|
| HMS Sweden (Head Office) | | | |
| E-mail: | sales@hms-networks.com | E-mail: | support@hms-networks.com |
| Phone: | +46 (0) 35 - 17 29 56 | Phone: | +46 (0) 35 - 17 29 20 |
| Fax: | +46 (0) 35 - 17 29 09 | Fax: | +46 (0) 35 - 17 29 09 |
| Online: | www.anybus.com | Online: | www.anybus.com |
| HMS North America | | | |
| E-mail: | us-sales@hms-networks.com | E-mail: | us-support@hms-networks.com |
| Phone: | +1-312 - 829 - 0601 | Phone: | +1-312-829-0601 |
| Toll Free: | +1-888-8-Anybus | Toll Free: | +1-888-8-Anybus |
| Fax: | +1-312-629-2869 | Fax: | +1-312-629-2869 |
| Online: | www.anybus.com | Online: | www.anybus.com |
| HMS Germany | | | |
| E-mail: | ge-sales@hms-networks.com | E-mail: | ge-support@hms-networks.com |
| Phone: | +49 (0) 721-989777-000 | Phone: | +49 (0) 721-989777-000 |
| Fax: | +49 (0) 721-989777-010 | Fax: | +49 (0) 721-989777-010 |
| Online: | www.anybus.de | Online: | www.anybus.de |
| HMS Japan | | | |
| E-mail: | jp-sales@hms-networks.com | E-mail: | jp-support@hms-networks.com |
| Phone: | +81 (0) 45-478-5340 | Phone: | +81 (0) 45-478-5340 |
| Fax: | +81 (0) 45-476-0315 | Fax: | +81 (0) 45-476-0315 |
| Online: | www.anybus.jp | Online: | www.anybus.jp |
| HMS China | | | |
| E-mail: | cn-sales@hms-networks.com | E-mail: | cn-support@hms-networks.com |
| Phone: | +86 (0) 10-8532-3183 | Phone: | +86 (0) 10-8532-3023 |
| Fax: | +86 (0) 10-8532-3209 | Fax: | +86 (0) 10-8532-3209 |
| Online: | www.anybus.cn | Online: | www.anybus.cn |
| HMS Italy | | | |
| E-mail: | it-sales@hms-networks.com | E-mail: | it-support@hms-networks.com |
| Phone: | +39 039 59662 27 | Phone: | +39 039 59662 27 |
| Fax: | +39 039 59662 31 | Fax: | +39 039 59662 31 |
| Online: | www.anybus.it | Online: | www.anybus.it |
| HMS France | | | |
| E-mail: | fr-sales@hms-networks.com | E-mail: | fr-support@hms-networks.com |
| Phone: | +33 (0) 3 68 368 034 | Phone: | +33 (0) 3 68 368 033 |
| Fax: | +33 (0) 3 68 368 031 | Fax: | +33 (0) 3 68 368 031 |
| Online: | www.anybus.fr | Online: | www.anybus.fr |
| HMS UK & Eire | | | |
| E-mail: | uk-sales@hms-networks.com | E-mail: | support@hms-networks.com |
| Phone: | +44 (0) 1926 405599 | Phone: | +46 (0) 35 - 17 29 20 |
| Fax: | +44 (0) 1926 405522 | Fax: | +46 (0) 35 - 17 29 09 |
| Online: | www.anybus.co.uk | Online: | www.anybus.com |
| HMS Denmark | | | |
| E-mail: | dk-sales@hms-networks.com | E-mail: | support@hms-networks.com |
| Phone: | +45 (0) 35 38 29 00 | Phone: | +46 (0) 35 - 17 29 20 |
| Fax: | +46 (0) 35 17 29 09 | Fax: | +46 (0) 35 - 17 29 09 |
| Online: | www.anybus.com | Online: | www.anybus.com |
| HMS India | | | |
| E-mail: | in-sales@hms-networks.com | E-mail: | in-support@hms-networks.com |
| Phone: | +91 (0) 20 40111201 | Phone: | +91 (0) 20 40111201 |

| Sales | | Support | |
|---|---|---|---|
| Fax: | +91 (0) 20 40111105 | Fax: | +91 (0) 20 40111105 |
| Online: | www.anybus.com | Online: | www.anybus.com |