Where Automation Connects.

PLX3x Series

Ethernet and Serial Gateways

March 13, 2017

USER MANUAL

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

How to Contact Us

ProSoft Technology, Inc.
9201 Camino Media, Suite #200
Bakersfield, CA 93311
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

PLX3x Series Ethernet and Serial Gateways User Manual

March 13, 2017

ProSoft Technology® Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on Adobe® Acrobat Reader file format (.PDFs). These product documentation files may also be freely downloaded from our web site: http://www.prosoft-technology.com/

Literature Content Disclaimer

This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither ProSoft Technology nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. Information in this document including illustrations, specifications and dimensions may contain technical inaccuracies or typographical errors. ProSoft Technology makes no warranty or representation as to its accuracy and assumes no liability for and reserves the right to correct such inaccuracies or errors at any time without notice. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of ProSoft Technology. All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components. When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use ProSoft Technology software or approved software with our hardware products may result in injury, harm, or improper operating results. Failure to observe this information can result in injury or equipment damage.

Important Installation Instructions

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

⚠

This Equipment is Suitable For Use in Class I, Division 2, Groups A, B, C, D or Non-Hazardous Locations Only

WARNING – Explosion Hazard – Substitution of Any Components May Impair Suitability for Class I, Division 2

WARNING – Explosion Hazard – Do Not Disconnect Equipment Unless Power Has Been Switched Off Or The Area is Known To Be Non-Hazardous

Agency Approvals and Certifications

CE Mark
UL/cUL Class I Div II
ATEX Zone 2
CB Safety
RoHS

Contents

1 Start Here

In This Chapter

1.1 System Requirements

The ProSoft Configuration Builder configuration software for the gateway requires the following minimum hardware and software components:

- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- 128 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)
- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)

Supported operating systems:

- Microsoft Windows 7 (32 bit)
- Microsoft Windows XP Professional with Service Pack 1 or 2

1.2 Package Contents

The following components are included with your gateway, and are all required for installation and configuration. The quantity of cables provided depends on the specific protocol combination being used.

> **Important:** Before beginning the installation, please verify that all of the following items are present.

Gateway with Ethernet Port

Qty.	Part Name	Part Number	Part Description
1	Ethernet cable	RL-CBL025	5’ straight-through cable
1	Mini screwdriver	HRD250	Tool for wiring and securing the power connector
1	Power connector	J180	PLX3x gateway power connector

Gateway with Two Ethernet Ports

Qty.	Part Name	Part Number	Part Description
1	Ethernet cable	RL-CBL025	5’ straight-through cable
1	Mini screwdriver	HRD250	Tool for wiring and securing the power connector
1	Power connector	J180	PLX3x gateway power connector

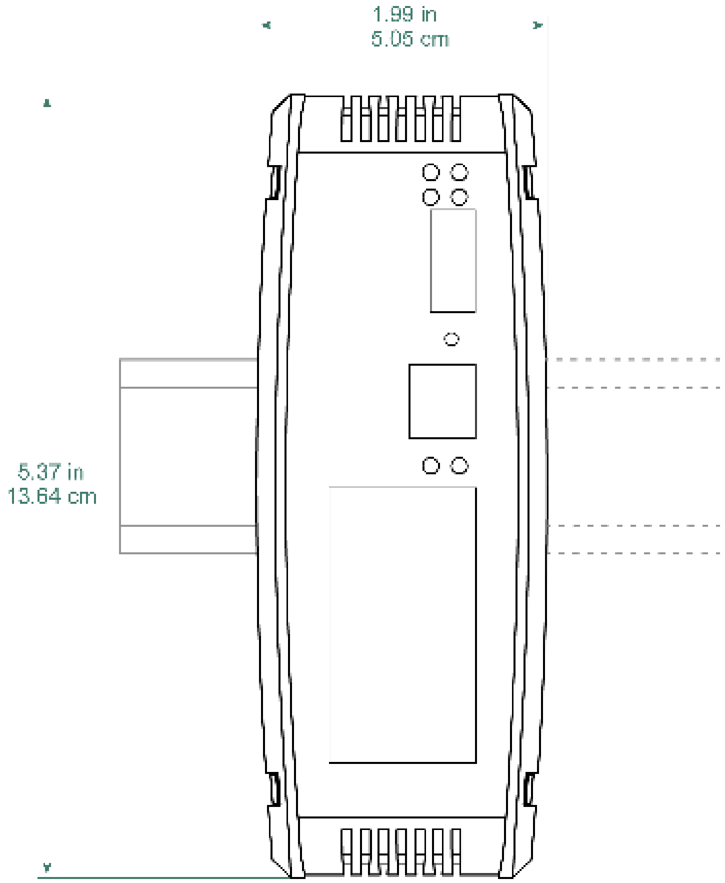
Gateway with Ethernet Port and Single Serial Port

Qty.	Part Name	Part Number	Part Description
1	Ethernet cable	RL-CBL025	5’ straight-through cable
1	DB9 to Screw Terminal Adaptor	1454-9F	DB9 to screw terminal adapter
1	RJ45-DB9M Serial Adapter Cable	CABLE14	RJ45 to DB9 male serial adapter cable
1	Power Connector	J180	PLX3x gateway power connector
1	Mini screwdriver	HRD250	Tool for wiring and securing the power connector

Gateway with Ethernet Port and Four Serial Ports

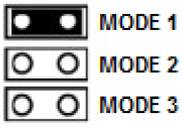
Qty.	Part Name	Part Number	Part Description
1	Ethernet cable	RL-CBL025	5’ straight-through cable
4	DB9 to Screw Terminal Adaptor	1454-9F	DB9 to screw terminal adapter
4	RJ45-DB9M Serial Adapter Cable	CABLE14	RJ45 to DB9 male serial adapter cable
1	Power Connector	J180	PLX3x gateway power connector
1	Mini screwdriver	HRD250	Tool for wiring and securing the power connector

1.3 Mounting the Gateway on a DIN-rail



1.4 Jumper Settings

There are three sets of jumper settings located on the back of the module.



- **MODE 1 - Development Mode Jumper:** This is the top jumper, used for firmware updates only. The two pins should NOT be jumpered during normal operation.

- **MODE 2 - Default IP Jumper:** This is the middle jumper. The default IP address of the ProLinx gateway is 192.168.0.250. Set this jumper to put the gateway's IP address back to the default.

- **MODE 3 - Reserved:** This is the bottom jumper, reserved for internal ProSoft Technology use only. The firmware will not run when these pins are shorted.

1.5 SD Card

PLX3x products can be ordered with an optional SD card (Part Number SDI-1G). In the event of a disaster, the SD card can be moved from one module to the next and resume operation. Below is a list of how the module will act - with and without an SD card.

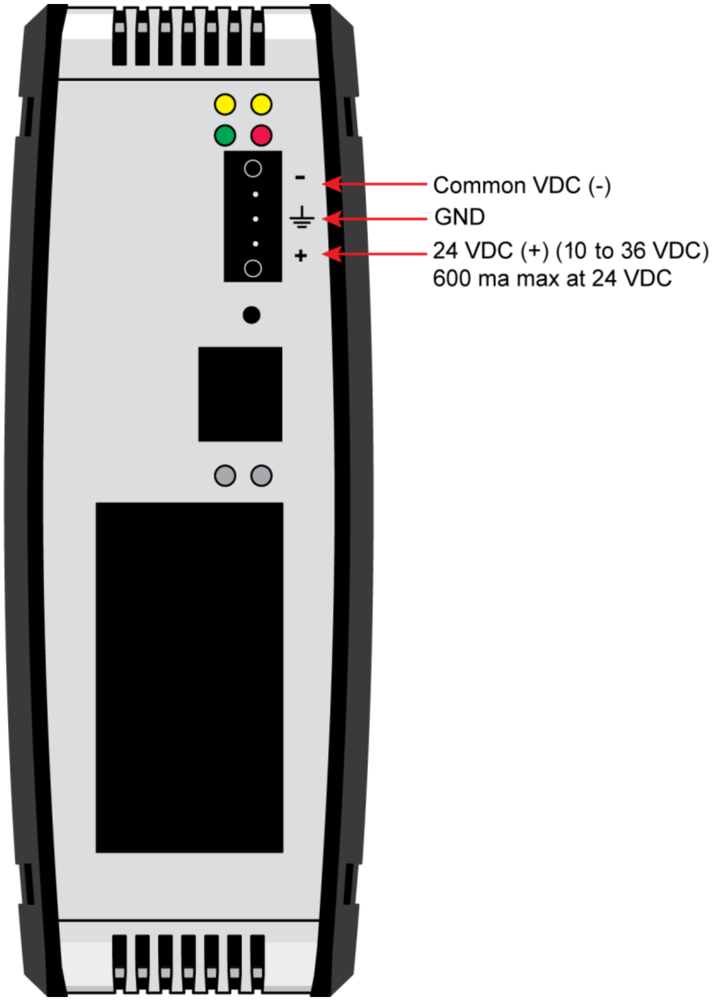
Without an SD Card

- Configuration data is downloaded to the internal memory of the module.
- If a blank SD Card is inserted in to the module after the module has been configured, the configuration data will not be transferred to the SD card. The configuration data would need to be downloaded to the module while the SD card is in place.

With an SD Card

- Configuration data is downloaded to the SD Card
- The configuration data is not transferred from the SD card to the internal memory of the module. If the SD card is removed and power is cycled to the module, the module will load the configuration data from the module's memory. If there is no configuration data in the module's memory, it will be restored to factory default.

1.6 Connecting Power to the PLX3x Gateway



WARNING: Ensure that polarity is not reversed when applying power to the gateway. This will cause damage to the gateway's power supply.

2 Configuring Your Gateway

In This Chapter

ProSoft Configuration Builder (PCB) is a convenient and powerful software tool for managing your gateway configuration. Use PCB to configure a new project, or to transfer an existing project to a new device. You can also to use PCB to retrieve a configuration from a working gateway by uploading the configuration from the gateway.

2.1 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder (PCB)* software to configure the gateway. You can always get the newest version of *ProSoft Configuration Builder* from the ProSoft Technology website.

To install ProSoft Configuration Builder from the ProSoft Technology website

1. Open your web browser and navigate to http://www.prosoft-technology.com/pcb
2. Click the link at the *Current Release Version* section to download the latest version of *ProSoft Configuration Builder*.
3. Choose **SAVE** or **SAVE FILE** when prompted.
4. Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
5. When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

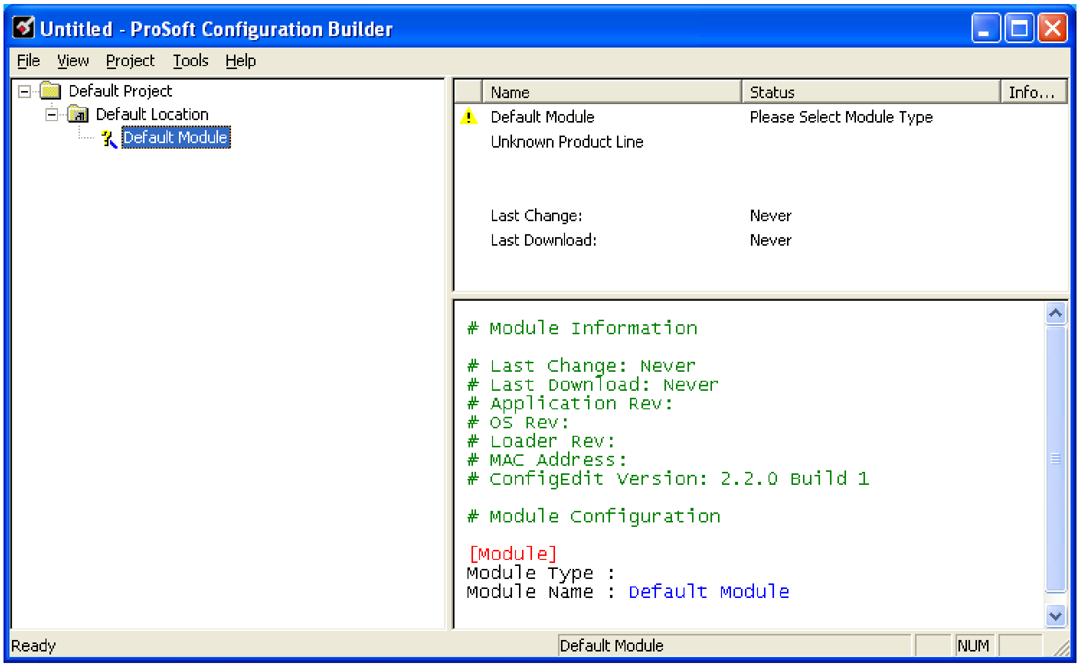
2.2 Using the Online Help

Most of the information needed to help you use *ProSoft Configuration Builder* is provided in a Help System that is always available whenever you are running *ProSoft Configuration Builder*. The Help System does not require an Internet connection.

To view the help pages, start *ProSoft Configuration Builder*, open the **HELP** menu, and then choose **CONTENTS.**

2.3 Setting Up the Project

To begin, start ProSoft Configuration Builder (PCB). If you have used other Windows configuration tools before, you will find the screen layout familiar. ProSoft Configuration Builder's (PCB's) window consists of a tree view on the left, and an information pane on the upper right side, and a configuration pane on the lower right side of the window. When you first start PCB, the tree view consists of folders for Default Project and Default Location, with a Default Gateway in the Default Location folder. The following screen shows the PCB window with a new project.

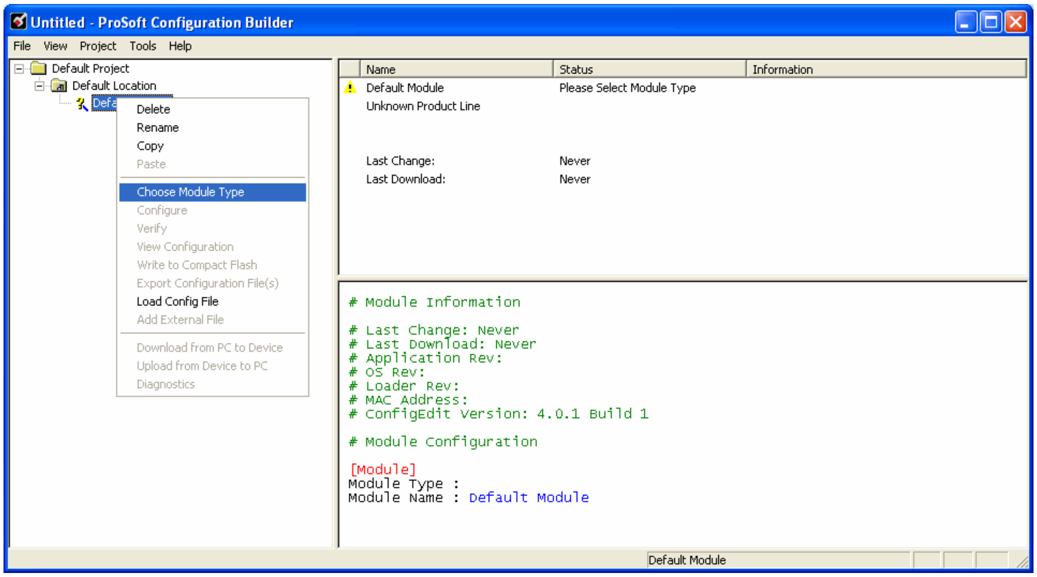

Untitled - ProSoft Configuration Builder
File View Project Tools Help
Default Project
Default Location
Default Module
Ready
Default Module
NUM

Name	Status	Info...
Default Module	Please Select Module Type	
Unknown Product Line		
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.2.0 Build 1

# Module Configuration

[Module]
Module Type :
Module Name : Default Module
```

<u>To add the gateway to the project</u>

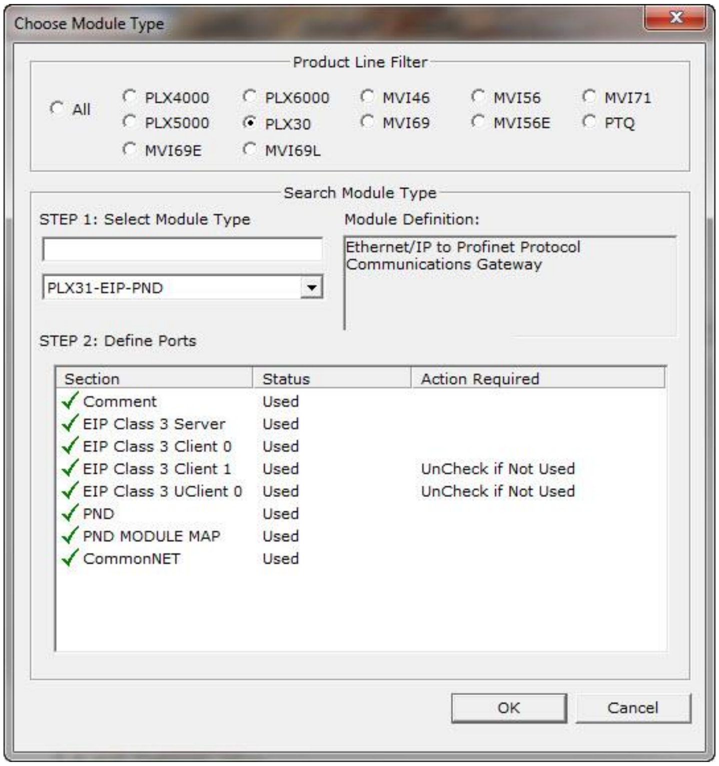
1 Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.



Name	Status	Information
Default Module	Please Select Module Type	
Unknown Product Line		
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 4.0.1 Build 1

# Module Configuration

[Module]
Module Type :
Module Name : Default Module
```

2 On the shortcut menu, select **CHOOSE MODULE TYPE**. This action opens the *Choose Module Type* dialog box.



Choose Module Type

Product Line Filter

○ All ○ PLX4000 ○ PLX6000 ○ MVI46 ○ MVI56 ○ MVI71
○ PLX5000 ● PLX30 ○ MVI69 ○ MVI56E ○ PTQ
○ MVI69E ○ MVI69L

Search Module Type

STEP 1: Select Module Type

PLX31-EIP-PND

Module Definition:

Ethernet/IP to Profinet Protocol Communications Gateway

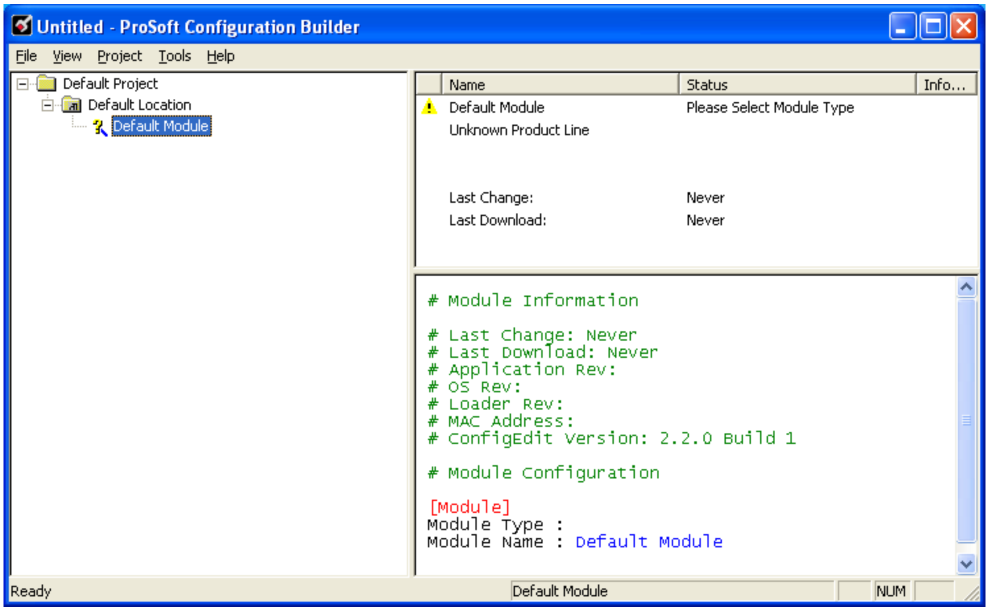
STEP 2: Define Ports

Section	Status	Action Required
✓ Comment	Used	
✓ EIP Class 3 Server	Used	
✓ EIP Class 3 Client 0	Used	
✓ EIP Class 3 Client 1	Used	UnCheck if Not Used
✓ EIP Class 3 UClient 0	Used	UnCheck if Not Used
✓ PND	Used	
✓ PND MODULE MAP	Used	
✓ CommonNET	Used	

3 In the *Product Line Filter* area of the dialog box, select the PLX30 radio button.

4 In the *STEP 1: Select Module Type* drop-down list, select the model number that matches your gateway, and then click **OK** to save your settings and return to the *PCB Main* window.

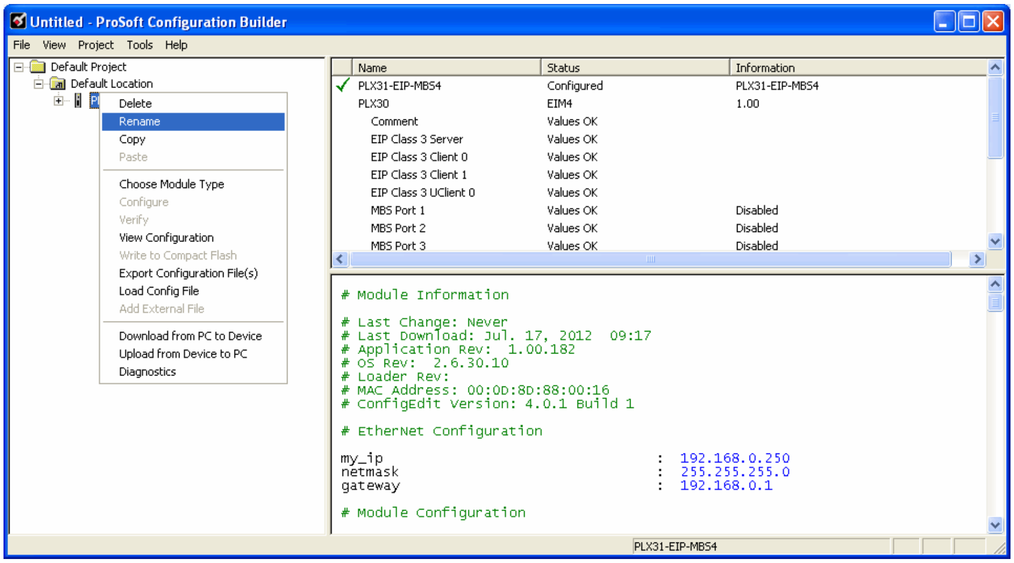
2.4 Renaming PCB Objects


Untitled - ProSoft Configuration Builder
File View Project Tools Help
Default Project
Default Location
Default Module
Ready
Default Module
NUM

Name	Status	Info...
Default Module	Please Select Module Type	
Unknown Product Line		
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.2.0 Build 1

# Module Configuration

[Module]
Module Type :
Module Name : Default Module
```

The *Default Project* and *Default Location* folders may be renamed in the tree view. Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**



Name	Status	Information
✓ PLX31-EIP-MBS4	Configured	PLX31-EIP-MBS4
PLX30	EIM4	1.00
Comment	Values OK	
EIP Class 3 Server	Values OK	
EIP Class 3 Client 0	Values OK	
EIP Class 3 Client 1	Values OK	
EIP Class 3 UClient 0	Values OK	
MBS Port 1	Values OK	Disabled
MBS Port 2	Values OK	Disabled
MBS Port 3	Values OK	Disabled

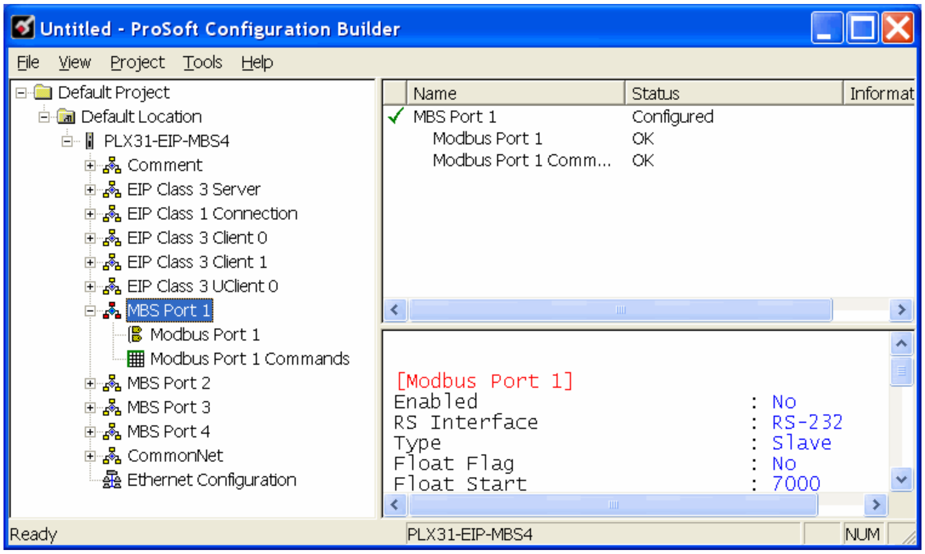
```
# Module Information

# Last Change: Never
# Last Download: Jul. 17, 2012  09:17
# Application Rev:  1.00.182
# OS Rev:  2.6.30.10
# Loader Rev:
# MAC Address: 00:0D:8D:88:00:16
# ConfigEdit Version: 4.0.1 Build 1

# EtherNet Configuration

my_ip                                      :  192.168.0.250
netmask                                    :  255.255.255.0
gateway                                    :  192.168.0.1

# Module Configuration
```

1 Type the name to assign to the object.
2 Click *away* from the object to save the new name.

2.5 Configuring the Drivers

1 Click the **[+]** sign next to the Gateway icon to expand gateway information.

2 Click the **[+]** sign next to any icon to view gateway information and configuration options.



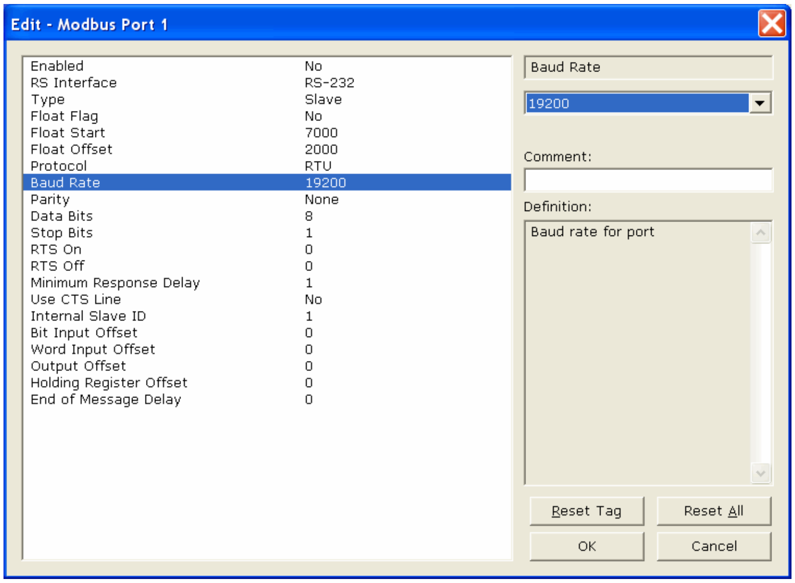
	Name	Status	Informat
✓	MBS Port 1	Configured	
	Modbus Port 1	OK	
	Modbus Port 1 Comm...	OK	

```
[Modbus Port 1]
Enabled                          : No
RS Interface                     : RS-232
Type                             : Slave
Float Flag                       : No
Float Start                      : 7000
```

3 Double-click any icon to open an *Edit* dialog box.

4 To edit a parameter, select the parameter name in the left hand pane, then edit its corresponding field in the right hand pane.

Note: Depending on the parameter, the editable field will accept typed input in the form of text or a valid numerical value, or it will have a dropdown list with options to choose from.



Edit - Modbus Port 1

Enabled	No
RS Interface	RS-232
Type	Slave
Float Flag	No
Float Start	7000
Float Offset	2000
Protocol	RTU
Baud Rate	19200
Parity	None
Data Bits	8
Stop Bits	1
RTS On	0
RTS Off	0
Minimum Response Delay	1
Use CTS Line	No
Internal Slave ID	1
Bit Input Offset	0
Word Input Offset	0
Output Offset	0
Holding Register Offset	0
End of Message Delay	0

Baud Rate

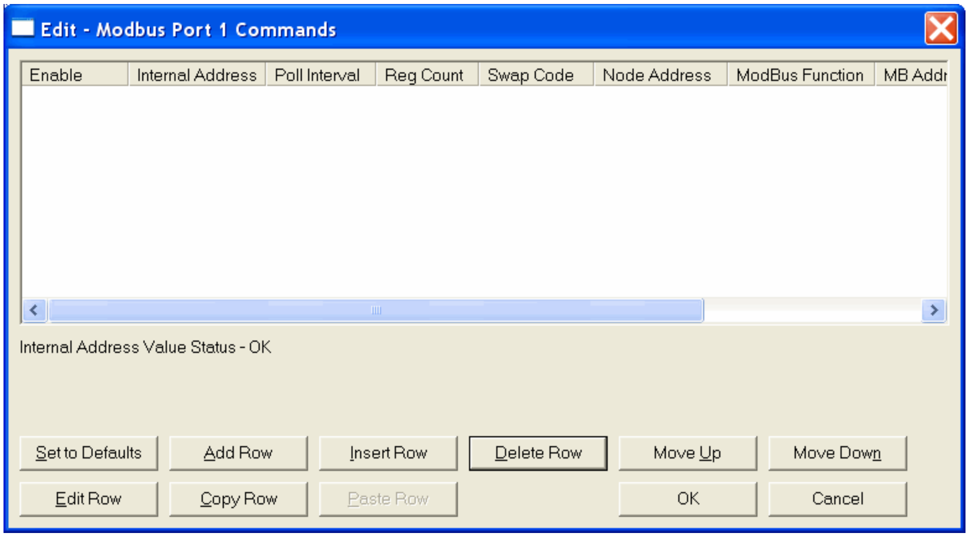
19200

Comment:

Definition:

Baud rate for port

5 Double-clicking any ▦ icon will open an *Edit* dialog box with a table.

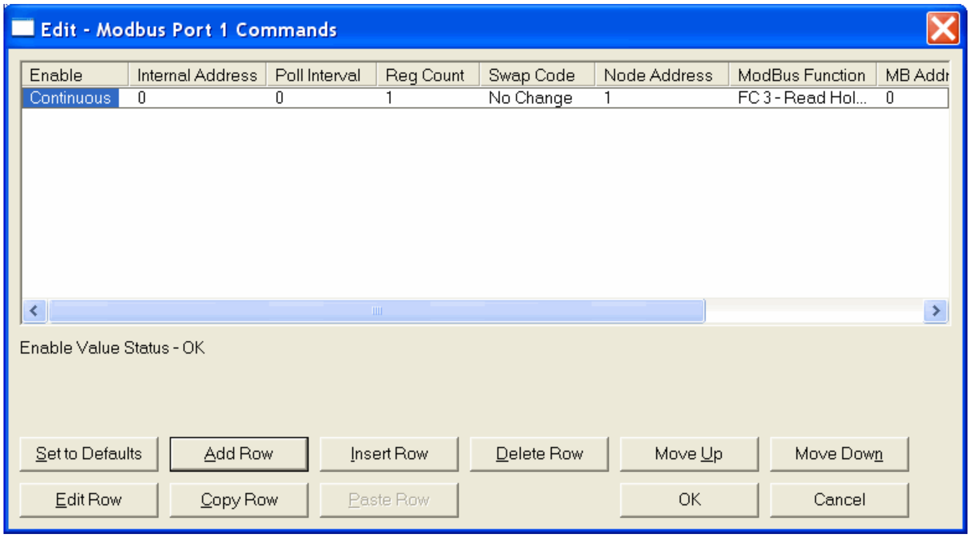


Edit - Modbus Port 1 Commands

Enable	Internal Address	Poll Interval	Reg Count	Swap Code	Node Address	ModBus Function	MB Addr

Internal Address Value Status - OK

6 To add a row to the table, click the **Add Row** button.

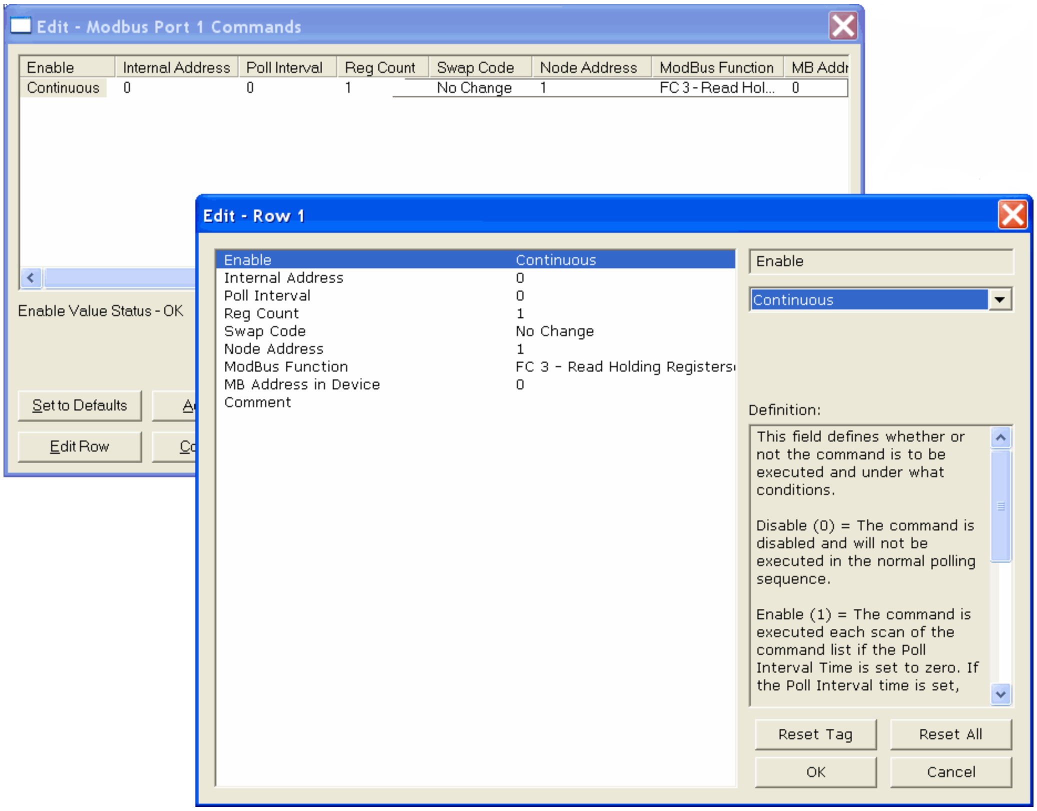


Edit - Modbus Port 1 Commands

Enable	Internal Address	Poll Interval	Reg Count	Swap Code	Node Address	ModBus Function	MB Addr
Continuous	0	0	1	No Change	1	FC 3 - Read Hol...	0

Enable Value Status - OK

7 To edit the row, click the **Edit Row** button. This will open an *Edit* dialog box where you can edit the row parameters.



Edit - Modbus Port 1 Commands

Enable	Internal Address	Poll Interval	Reg Count	Swap Code	Node Address	ModBus Function	MB Addr
Continuous	0	0	1	No Change	1	FC 3 - Read Hol...	0

Enable Value Status - OK

Edit - Row 1

Enable	Continuous
Internal Address	0
Poll Interval	0
Reg Count	1
Swap Code	No Change
Node Address	1
ModBus Function	FC 3 - Read Holding Registers
MB Address in Device	0
Comment	

Enable

Continuous

Definition:

This field defines whether or not the command is to be executed and under what conditions.

Disable (0) = The command is disabled and will not be executed in the normal polling sequence.

Enable (1) = The command is executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval time is set,

8 When configuration is complete, download the configuration to the gateway.

9 For protocol-specific configuration information, see the Configuration section in the appropriate protocol chapter of this manual:

EIP configuration (page 54)

MBTCP configuration (page 98)

MBS configuration(page127)

ASCII configuration (page 163)

SIE configuration (page 169)

PND configuration (page 196)

2.6 Using the CommonNet Data Map

Note: This is an advanced configuration feature and is **not** required for the basic operation of the gateway.

The *Data Map* section allows data to be copied between areas in the gateway's internal database.

The Data Map is especially useful for copying protocol-specific error and status data from the gateway's upper memory registers (address 4000 and up) to the user-accessible memory registers (addresses 0 to 3999). The error and status data copied into the user memory area can then be accessed by a remote device, such as an HMI or processor.

Information about upper memory addresses where the gateway places protocol-specific error and status data can be found in the Diagnostics section in the appropriate protocol chapter of this manual:

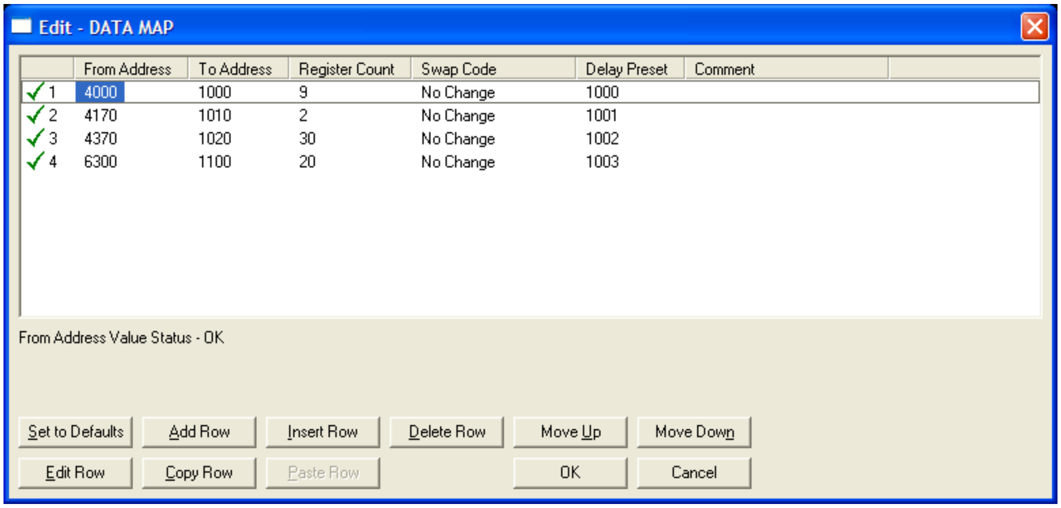
EIP diagnostics (page 69)
MBTCP diagnostics (page 106)
MBS diagnostics (page 134)
ASCII diagnostics (page 165)
SIE diagnostics (page 184)
PND diagnostics (page 216)

The Data Map can also be used to condense widely dispersed data into one contiguous data block, for simplified access.

A maximum of 100 registers per *Data Map* command can be copied, and a maximum of 200 separate copy commands can be configured.

The byte and/or word order can be rearranged during the copy process. For example, by rearranging byte or word order, floating-point values can be converted to the correct format for a different protocol.

The following illustration shows an example Data Map.



	From Address	To Address	Register Count	Swap Code	Delay Preset	Comment
✓ 1	4000	1000	9	No Change	1000	
✓ 2	4170	1010	2	No Change	1001	
✓ 3	4370	1020	30	No Change	1002	
✓ 4	6300	1100	20	No Change	1003	

From Address Value Status - OK

The following table describes the parameters for configuring the Data Map.

Parameter	Value	Description
From Address	0 to highest Status Data address	This parameter specifies the beginning internal database register address for the copy operation. This address can be any valid address in the *User Data Area* or the *Status Data Area* of the gateway.
To Address	0 to 3999	This parameter specifies the beginning destination register address for the copy operation. This address must always be within the *User Data registers* area. A destination address must be specified that will not overwrite data that has been stored in memory by one of the communication protocols running on the gateway.
Register Count	1 to 100	This parameter specifies the number of registers to copy.
Swap Code	No Change Word Swap Word and Byte Swap Byte Swap	The order of the bytes in the registers may need to be swapped during the copy process in order to change the alignment of bytes between dissimilar protocols. This parameter is helpful when dealing with floating-point or other multi-register values, as there is no standard method of storage of these data types in slave devices. **No change:** No change is made in the byte ordering (1234 = 1234) **Word Swap:** The words are swapped (1234=3412) **Word and Byte Swap:** The words are swapped, then the bytes in each word are swapped (1234=4321) **Byte Swap:** The bytes in each word are swapped (1234=2143)
Delay Preset		This parameter sets an interval for each *Data Map* copy operation. The value that is specified for the *Delay Preset* is not a fixed amount of time. It is the number of firmware scans that must transpire between copy operations.

2.7 Configuring an IP Address

Use this procedure to configure the Ethernet settings for your Gateway. You must assign an IP address, subnet mask and gateway address. After you complete this step, you can connect to the Gateway with an Ethernet cable.

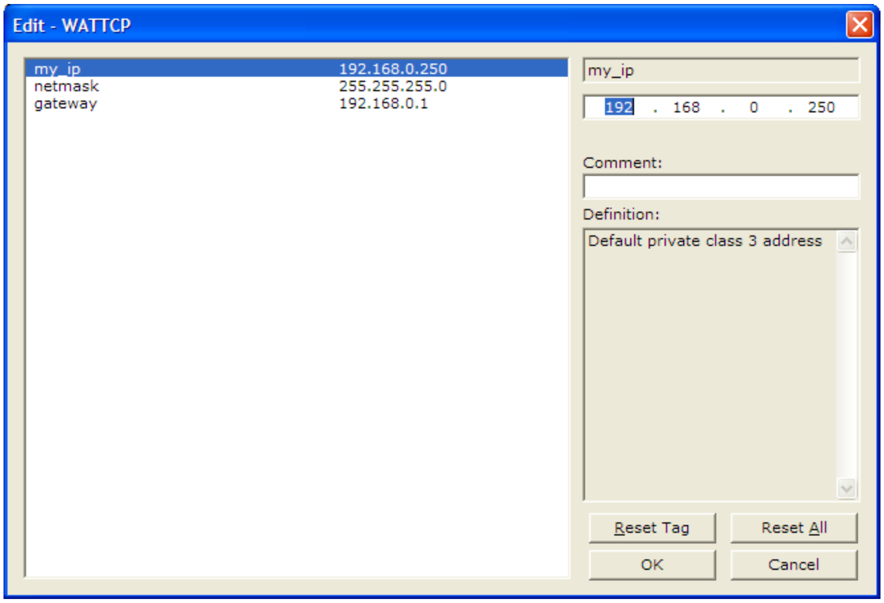
Note: The PLX32 module contains two Ethernet ports. In this case, you would specify network settings for the first Ethernet protocol on Enet P1 and another set of settings for the second Ethernet Protocol on Enet P2.

1 Determine the network settings for your Gateway, with the help of your network administrator if necessary. You will need the following information:

- IP address (fixed IP required) _____ . _____ . _____ . _____
- Subnet mask _____ . _____ . _____ . _____
- Gateway address _____ . _____ . _____ . _____

Note: The gateway address is optional, and is not required for networks that do not use a default gateway.

2 Double-click the **ETHERNET CONFIGURATION** icon. This action opens the *Edit* dialog box. The IP address shown is the gateway default IP address.



Edit - WATTCP

```
my_ip                    192.168.0.250
netmask                  255.255.255.0
gateway                  192.168.0.1
```

my_ip

192 . 168 . 0 . 250

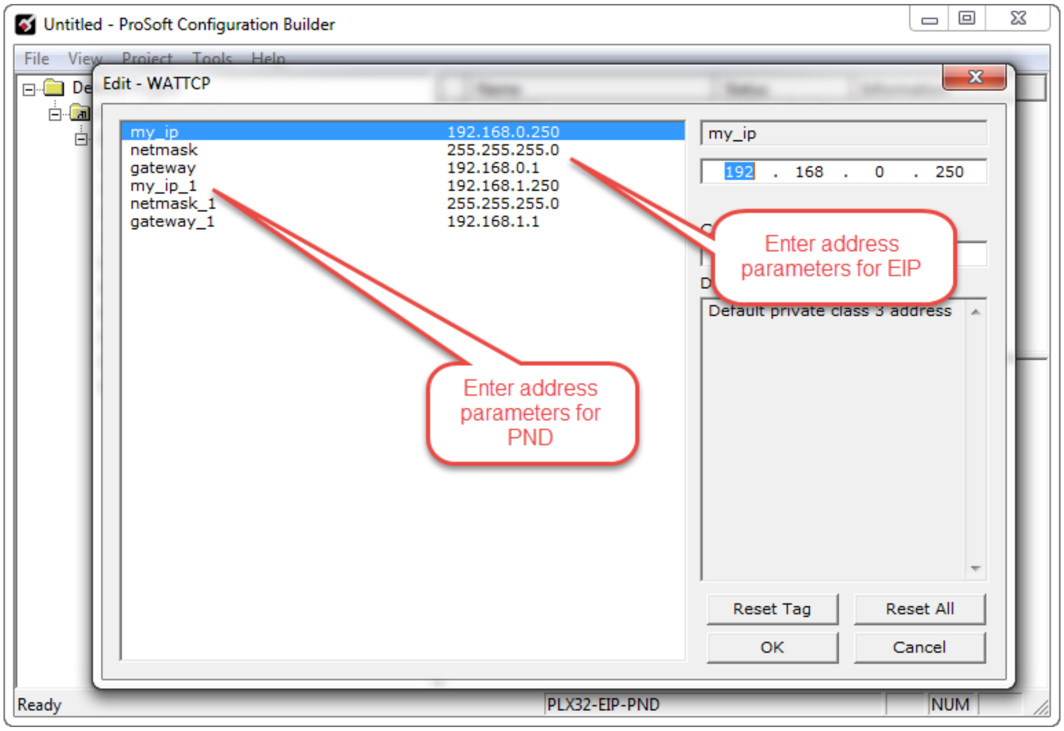
Comment:

Definition:

Default private class 3 address

3 Edit the values for *my_ip*, *netmask* (subnet mask) and *gateway* (default gateway).

> **Note:** If you are using a PLX32 module, you must specify values for both ports. My_ip is used to specify values for the first protocol. For example, if you are configuring a PLX32-EIP-MBTCP, you would specify the network values for the EIP protocol first. A second set of values are available for the second protocol; in this case, MBTCP.



my_ip	192.168.0.250
netmask	255.255.255.0
gateway	192.168.0.1
my_ip_1	192.168.1.250
netmask_1	255.255.255.0
gateway_1	192.168.1.1

my_ip

192 . 168 . 0 . 250

Default private class 3 address

PLX32-EIP-PND

Enter address parameters for EIP

Enter address parameters for PND

4 When you are finished editing, click **OK** to save your changes and return to the *ProSoft Configuration Builder* window.

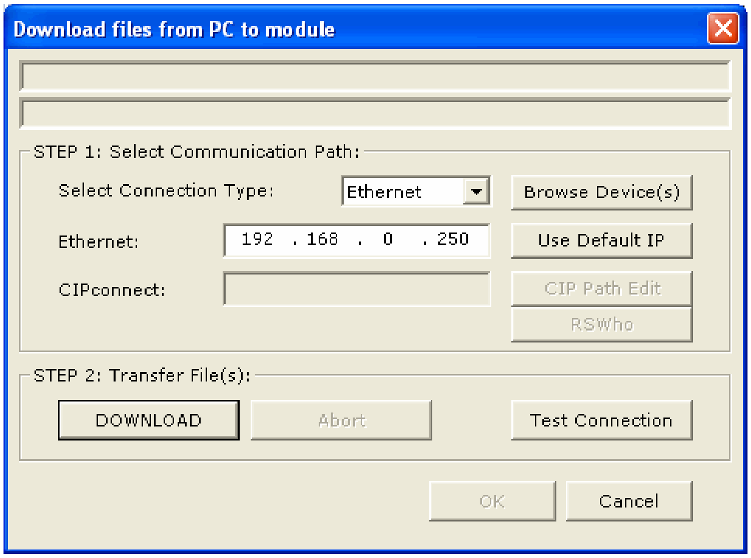
2.8 Downloading the Project to the Gateway

For the gateway to use the settings you configured, you must download (copy) the updated *Project* file from your PC to the gateway.

To download the project file

1 In the tree view in *ProSoft Configuration Builder*, click once to select the gateway.

2 Right-click the Gateway icon to open a shortcut menu. From the shortcut menu, choose **DOWNLOAD FROM PC TO DEVICE**.



Download files from PC to module

STEP 1: Select Communication Path:

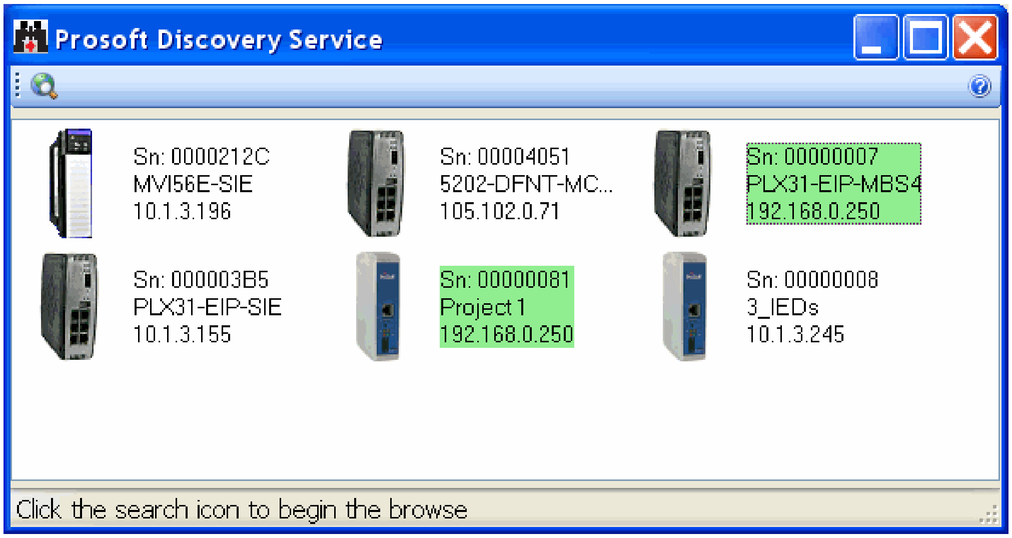
Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 250

CIPconnect:

STEP 2: Transfer File(s):

3 Click the **BROWSE DEVICE(S)** button to launch the *ProSoft Discovery Service* window, which displays the ProSoft devices on the network and their IP addresses.



Sn: 0000212C
MVI56E-SIE
10.1.3.196

Sn: 00004051
5202-DFNT-MC...
105.102.0.71

Sn: 00000007
PLX31-EIP-MBS4
192.168.0.250

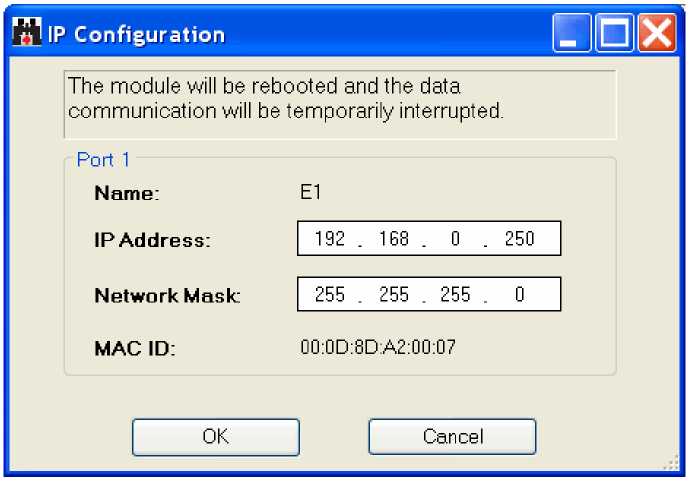
Sn: 000003B5
PLX31-EIP-SIE
10.1.3.155

Sn: 00000081
Project 1
192.168.0.250

Sn: 00000008
3_IEDs
10.1.3.245

Click the search icon to begin the browse

4 Right-click your PLX3x-series gateway and select IP Configuration from the shortcut menu.



The module will be rebooted and the data communication will be temporarily interrupted.

Port 1

Name: E1

IP Address: 192 . 168 . 0 . 250

Network Mask: 255 . 255 . 255 . 0

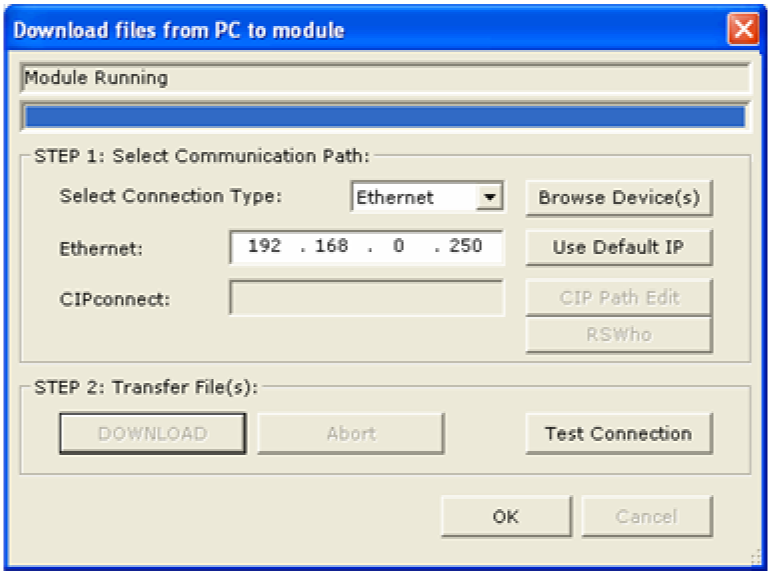
MAC ID: 00:0D:8D:A2:00:07

5 Enter the same IP address and network mask that you entered in the Ethernet configuration of the gateway. Click OK. The gateway will reboot.

6 Close the *ProSoft Discovery Service* window to return to the *Download* dialog box.

7 Click the **DOWNLOAD** button.

The gateway will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in the *Download* dialog box will display the message *Module Running*.



Download files from PC to module

Module Running

STEP 1: Select Communication Path:

Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 250

CIPconnect:

STEP 2: Transfer File(s):

2.9 Printing a Configuration File

1 Select the Gateway icon, and then click the right mouse button to open a shortcut menu.

2 On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the *View Configuration* window.

3 In the *View Configuration* window, open the **FILE** menu, and choose **PRINT**. This action opens the *Print* dialog box.

4 In the *Print* dialog box, choose the printer to use from the drop-down list, select printing options, and then click **OK**.

3 Diagnostics and Troubleshooting

In This Chapter

3.1 LED Indicators

Troubleshooting can be performed using several methods.

The first and quickest is to scan the LEDs on the gateway to determine the existence and possibly the cause of a problem. The gateway's LEDs provide valuable information such as

- The state of each port
- System configuration errors
- Application errors
- Fault indications

3.1.1 Main Gateway LEDs

LED	State	Description
PWR (Power)	Off	Power is not connected to the power terminals or source is insufficient to properly power the gateway (208 mA at 24 VDC is required).
	Solid Green	Power is connected to the power terminals.
FLT (Fault)	Off	Normal operation.
	Solid Red	A critical error has occurred. Program executable has failed or has been user-terminated and is no longer running. Press the **Reset** button or cycle power to clear the error.
CFG (Configuration)	Off	Normal operation.
	Solid Amber	The unit is in configuration mode. Either a configuration error exists, or the configuration file is currently being downloaded or read. After power-up, the configuration is read, and the unit implements the configuration values and initializes the hardware. This occurs during power cycle or after the **Reset** button is pressed.
ERR (Error)	Off	Normal operation.
	FlashingAmber	An error condition has been detected and is occurring on one of the application ports. Check configuration and troubleshoot for communication errors.
	Solid Amber	This error flag is cleared at the start of each command attempt (Master/Client) or on each receipt of data (slave/adapter/server); so, if this condition exists, it indicates a large number of errors are occurring in the application (due to bad configuration) or on one or more ports (network communication failures).
NS (Network Status) for EIP protocol only	Off	No power or noIP address
	Solid Red	Duplicate IP address
	Solid Green	Connected
	Flashing Red	Connection timeout
	Flashing Green	IP address obtained; no established connections
	Alternating Red and Green Flash	Self-test
MS (Module Status) for EIP protocol only	Off	No power
	Solid Red	Major fault
	Solid Green	Device operational
	Flashing Red	Minor fault
	Flashing Green	Standby
	Alternating Red and Green Flash	Self-test

3.1.2 Ethernet Port LEDs

LED	State	Description
LINK/ACT	Off	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	Solid Green	Physical network connection detected. This LED must be ON solid for Ethernet communication to be possible
100 Mbit	Off	No activity on the port.
	Flashing Amber	The Ethernet port is actively transmitting or receiving data.

3.1.3 Serial Port LEDs (for Gateways with Serial Ports)

LED	State	Description
RX	Off	No activity on the port.
	Flashing Green	The port is actively receiving data.
TX	Off	No activity on the port.
	Flashing Amber	The port is actively transmitting data.

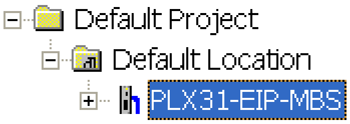
3.2 Using Diagnostics in ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) has many useful tools to help you with diagnostics and troubleshooting. You can use PCB to connect to your gateway and retrieve current status values, configuration data and other valuable information.

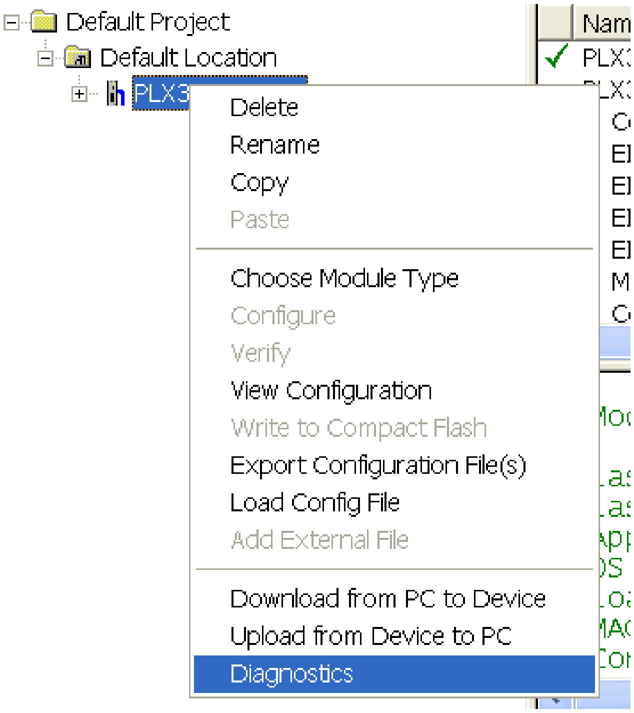
> **Tip:** You can have a ProSoft Configuration Builder Diagnostics window open for more than one gateway at a time.

To connect to the gateway's communication port

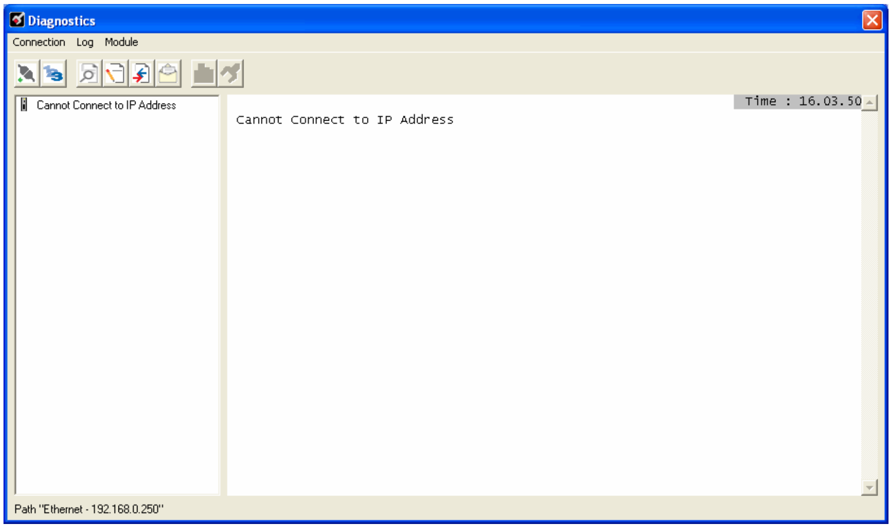
1 Start *PCB*, and then select the gateway. Click the right mouse button to open a shortcut menu.



2 On the shortcut menu, choose **DIAGNOSTICS**.



This opens the *Diagnostics* window.



Cannot Connect to IP Address

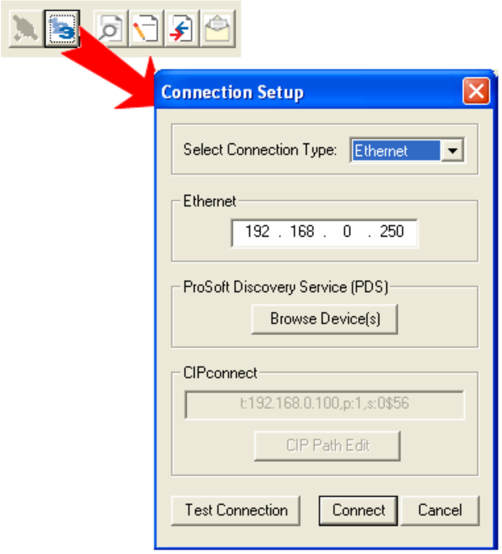
Time : 16.03.50

Cannot Connect to IP Address

Path "Ethernet - 192.168.0.250"

If there is no response from the gateway, as in the example above, follow these steps:

1 Click the *Setup Connection* button. In the *Connection Setup* dialog box, select **ETHERNET** from the *Select Connection Type* dropdown menu. Type in the gateway's IP address in the *Ethernet* field.



Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 250

CIPconnect: t:192.168.0.100,p:1,s:0$56

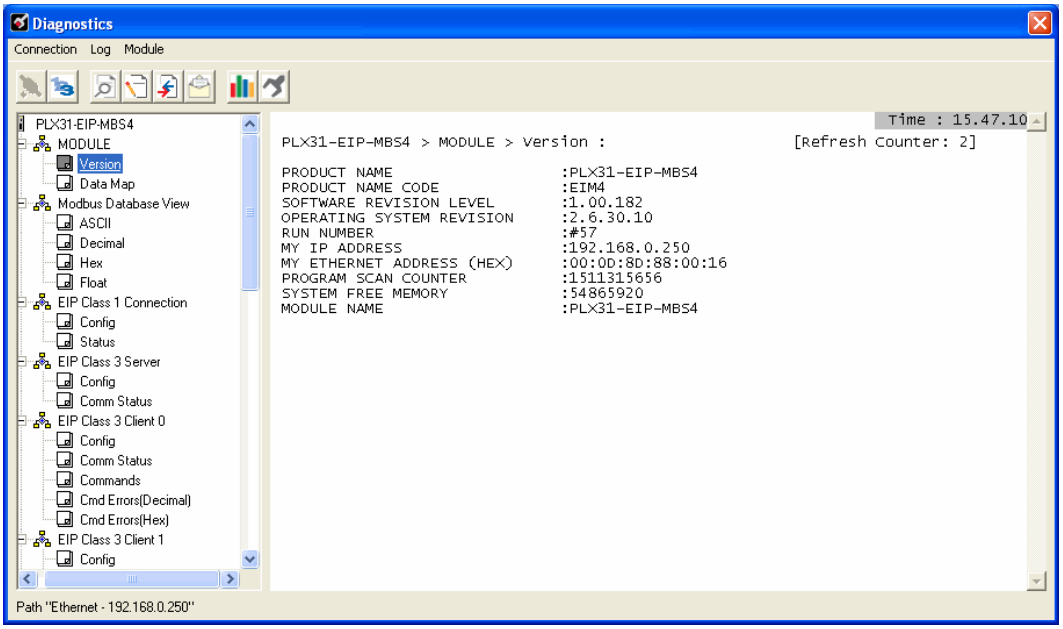
2 Click the *Connect* button. Verify that the Ethernet is connected properly between your computer's communication port and the gateway.

If you are still not able to establish a connection, contact ProSoft Technology for assistance.

3.2.1 Diagnostics Menu

The *Diagnostics* menu is arranged as a tree structure, with the *Main* menu at the top of the tree, and one or more submenus for each menu command.

The menu commands available will depend on the protocol combination of your gateway.



```
                                                              Time : 15.47.10
PLX31-EIP-MBS4 > MODULE > Version :                    [Refresh Counter: 2]

PRODUCT NAME                          :PLX31-EIP-MBS4
PRODUCT NAME CODE                     :EIM4
SOFTWARE REVISION LEVEL               :1.00.182
OPERATING SYSTEM REVISION             :2.6.30.10
RUN NUMBER                            :#57
MY IP ADDRESS                         :192.168.0.250
MY ETHERNET ADDRESS (HEX)             :00:0D:8D:88:00:16
PROGRAM SCAN COUNTER                  :1511315656
SYSTEM FREE MEMORY                    :54865920
MODULE NAME                           :PLX31-EIP-MBS4
```

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the gateway to stop communicating with the processor or with other devices, resulting in potential data loss or other communication failures. Use these commands only if you fully understand their potential effects, or if you are specifically directed to do so by ProSoft Technology Technical Support Engineers.

The following menu commands are common to all PLX3x-series gateways:

Menu Command	**Submenu Command**	**Description**
Module	Version	Displays the gateway's current software version and other important values. You may be asked to provide this information when calling for technical support.
	Data Map	Displays the gateway's Data Map configuration.
Database View	ASCII	Displays the contents of the gateway's database in ASCII character format.*
	Decimal	Displays the contents of the gateway's database in decimal number format.*
	Hex	Displays the contents of the gateway's database in hexadecimal number format.*
	Float	Displays the contents of the gateway's database in floating-point number format.*

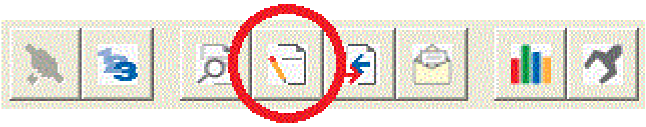
* Use the scroll bar on the right edge of the window to navigate through the database. Each page displays 100 words of data. The total number of pages available depends on your gateway's configuration.

3.2.2 Capturing a Diagnostic Session to a Log File

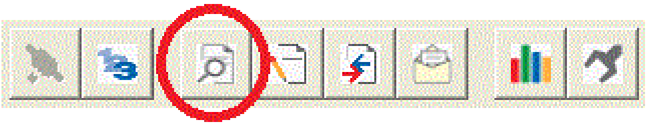
You can capture anything you do in a Diagnostics session to a log file. This feature can be useful for troubleshooting and record-keeping purposes, and for communication with ProSoft Technology's technical support team.

1 Open a *Diagnostics* window.

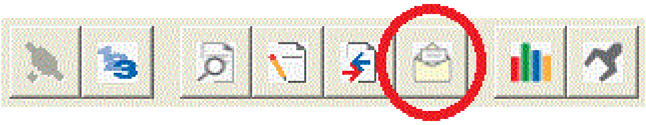
2 To log a Diagnostics session to a text file, click the **Log File** button on the toolbar at the top of the *Diagnostics* window. Click the button again to stop the capture.



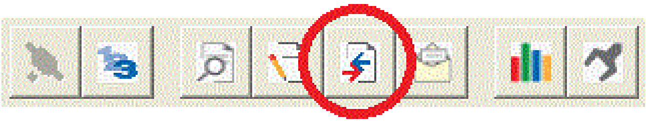
3 To view the log file created, click the **View Log File** button. The log file will open as a text file, which can be renamed and saved to a different location.



4 To email the log file to ProSoft Technology's technical support team, click the **Email Log File** button. (For this to work, Microsoft Outlook must be installed on your PC.)



5 If you do multiple sequential captures, PCB will append data from a new capture to the end of the previously captured data. If you want previous data to be cleared from the log file each time you start a new capture, click the **Clear Data** button.



3.2.3 Using the Data Analyzer (Serial Protocols Only)

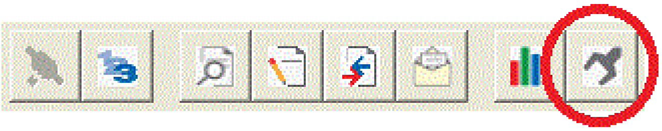
The Data Analyzer is an extremely valuable troubleshooting tool available in PCB. It allows you to “see” the data packets entering and leaving the serial ports on the gateway. You can also capture this data to a log file.

Note: The PCB Data Analyzer is for serial ports only. To analyze data traffic on an Ethernet port, we recommend using a network protocol analyzer available on the Internet, such as Wireshark.

To use the Data Analyzer

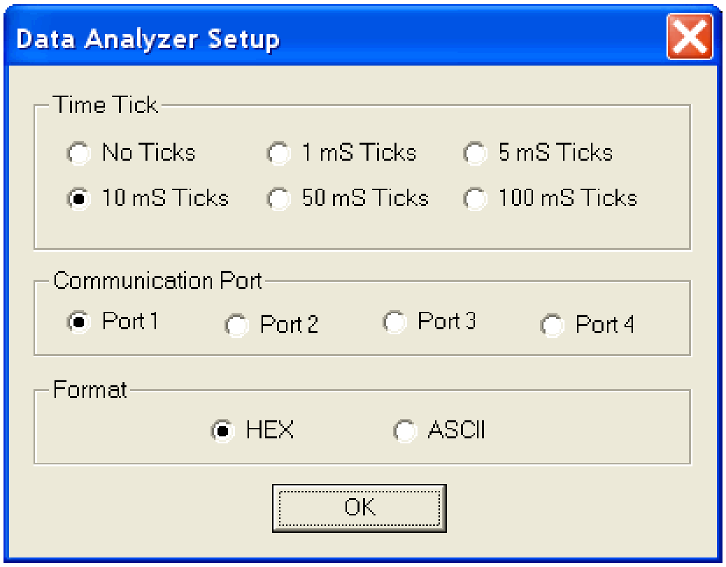
1 Open the *Diagnostics* window in PCB.

2 On the toolbar at the top of the window, click the **Setup Data Analyzer** button.



3 In the *Data Analyzer Setup* dialog box, specify the time tick interval, the serial port number, and whether the data packet contents should be displayed in hexadecimal number or ASCII character format. Click **OK**.

Note: The time tick is a symbol (_TT_) displayed on the Data Analyzer screen that allows you to estimate time intervals during a Data Analyzer session. The time tick will print at the time interval you specify in the *Data Analyzer Setup* dialog box. For example, if you select 10 mS Ticks, it will print every 10 milliseconds.



Data Analyzer Setup

Time Tick

○ No Ticks ○ 1 mS Ticks ○ 5 mS Ticks
● 10 mS Ticks ○ 50 mS Ticks ○ 100 mS Ticks

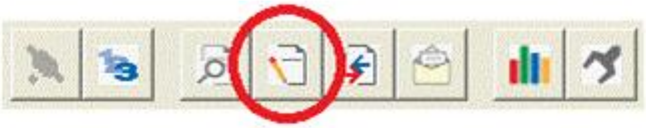
Communication Port

● Port 1 ○ Port 2 ○ Port 3 ○ Port 4

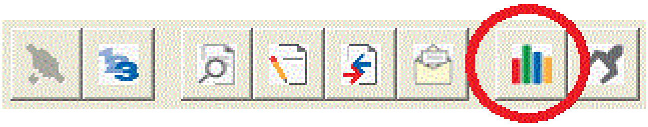
Format

● HEX ○ ASCII

4 If you wish to capture the Data Analyzer session to a log file, click the **Log File** button.



5 Click the **Start Data Analyzer** button to start the Data Analyzer. Click it again to stop it.



6 The example below is part of a capture of standard Modbus data packets. It is displayed in hexadecimal number format.
7 Data LEAVING the serial port is enclosed in angle brackets <>.
8 Data ENTERING the port is enclosed in square brackets [ ].
9 Each set of brackets holds one word (2 bytes) of data.

```
<2D><00><2E><00><2F><00><30><00><31><16><E0><R->_TT__TT_[05][10][00][28][00][0A]
[C1][82]_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT_<R+><06><04><00><32><00><0A><D0><75><R->_TT__TT_[06][04]
[14][00][00][00][00][00][00][00][00][00][00]_TT_[00][00][00][00][00][00][00][00]
[00][00][F3][97]_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT_<R+><06><10><00><32><00><0A><14><00><32><00><33><00>
<34><00><35><00><36><00><37><00><38><00><39><00><3A><00><3B><1E><26><R->_TT__TT_
[06][10][00][32][00][0A][E0][76]_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_<R+><01><03><00><00><00><0A><C5><CD>
<R->_TT_[01][03][14][00]_TT_[00][00][00][00][00][00][00][00][00][00][00][00][00]
[00][00][00][00][00][00][A3][67]_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_<R+><01><10><00><00><00><0A><14><04>
<D2><10><E1><16><2E><22><3D><3F><FF><00><05><00><06><00><07><00><08><00><09><86>
<49><R->_TT__TT__TT_[01][10][00][00][00][0A][40][0E]_TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_<R+><02><03><00>
<0A><00><0A><E5><FC><R->_TT__TT__TT_[02][03][14][00][00][00][00][00][00][00][00]
[00][00][00][00][00][00][00][00][00][00][00][00][F7][82]_TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_
_TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT__TT_<R+><02><10>
<00><0A><00><0A><14><00><0A><00><0B><00><0C><00><0D><00><0E><00><0F><00><10><00>
<11><00><12><00><13><A5><08><R->_TT__TT_[02][10][00][0A][00][0A][60][3F]_TT__TT_
```

For Modbus protocol users: To interpret the data packets, refer to the Modbus Protocol Specification, which can be found in this manual (page 141) or at www.modbus.org.

3.3 Gateway Status Data in Upper Memory

The gateway places useful status data in dedicated upper memory locations in its internal database. The Data Map functionality of the gateway can be used to map this data into the normal user data range of the gateway's database (registers 0 through 3999). It can be accessed by remote devices, such as HMIs or processors. See *Using the CommonNet Data Map* (page 25).

3.3.1 General Gateway Status Data in Upper Memory

The following table describes the contents of the gateway's general status data area.

Register Address	Description
4000 through 4001	Program Cycle Counter
4002 through 4004	Product Code (ASCII)
4005 through 4009	Product Revision (ASCII)
4010 through 4014	Operating System Revision (ASCII)
4015 through 4019	OS Run Number (ASCII)

3.3.2 Protocol-Specific Status Data in Upper Memory

The gateway also has upper memory locations for protocol-specific status data.

Information about upper memory addresses where the gateway places status data for its protocol drivers can be found in the Diagnostics sections of the protocol chapters:

EIP diagnostics (page 69)
MBTCP diagnostics (page 106)
MBS diagnostics (page 134)
ASCII diagnostics (page 165)
SIE diagnostics (page 184)
PND diagnostics (page 216)

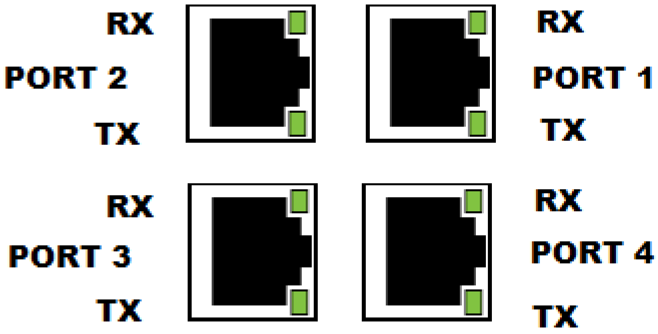
4 Hardware Information

In This Chapter

4.1 Hardware Specifications

Specification	Description
Power Supply	24 VDC nominal 10 VDC to 36 VDC allowed Positive, Negative, GND Terminals
Current Load	208mA normal @ 24 VDC normal 300 mA maximum @ 36 VDC maximum
Operating Temperature	-25°C to 70°C (-13°F to 158°F)
Storage Temperature	-40°C to 80°C (-40°F to 176°F)
Relative Humidity	5% to 95% RH with no condensation
Dimensions (Height x Width x Depth)	Standard: 5.38 in x 1.99 in x 4.38 in (13.67 cm x 5.05 cm x 11.13 cm)
LED Indicators (On all gateways)	Configuration (CFG) and Error (ERR) Communication Status Power (PWR) and Hardware Fault (FLT) Network Status (NS) EtherNet/IP™ Class I or Class III Connection Status (EtherNet/IP Only) Module Status (MS) Module Configuration Status (EtherNet/IP Only) Ethernet Communication Port Link/Activity and 100mbit Serial Communication Port Receive (RX) and Transmit (TX)
Ethernet Port (S)	10/100Mbit full-duplex RJ45 Connector Electrical Isolation 1500 Vrms at 50 Hz to 60 Hz for 60 seconds, applied as specified in section 5.3.2 of IEC 60950: 1991 Ethernet Broadcast Storm Resiliency = less than or equal to 5000 [ARP] frames-per-second and less than or equal to 5 minutes duration
Serial Port Isolation	2500 Vrms port signal isolation per UL 1577 serial port communication signal uses RF(Radio Frequency) modulation signal as isolation media, IC chip model is Silicon Labs Si844x(Si8440,Si8441,Si8442).
Shipped With Each Unit	2.5 mm screwdriver J180 Power Connector (1 to 4) RJ45-DB9M Serial Adapter Cable (serial protocol only) (1 to 4) DB9 to Screw Terminal Adapter (serial protocol only)

4.1.1 Serial Port Specifications



Type	Specifications
Serial Port Isolation	2500 Vrms port signal isolation per UL 1577 serial port communication signal uses RF (Radio Frequency) modulation signal as isolation media, IC chip model is Silicon Labs Si844x (Si8440,Si8441,Si8442).
Serial Port Protection	RS-485/422 port interface lines TVS diode protected at +/- 27V standoff voltage. RS-232 port interface lines fault protected to +/- 36V power on, +/- 40V power off.

4.2 Serial Port Cables (for Gateways with Serial Ports)

This section contains information on the cable and pinout assignments for the PLX3x gateway's serial ports (RS-232/422/485). The PLX3x gateway may come with one or four serial ports, depending on the configuration purchased.

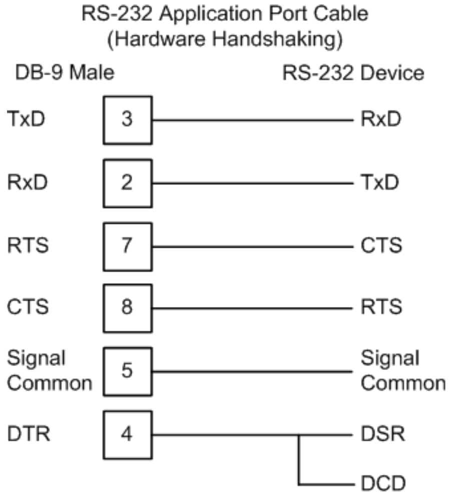
Note:
The PLX31-EIP-MBS4 gateway contains four serial communication ports.

The PLX31-EIP-MBS gateway contains one serial communication port.

Each physical serial port has a RJ45 jack connector. A six-inch RJ45 to DB9Male adapter cable is provided for each serial port. The DB9Male adapter cable provides connections for RS-232, wired as Data Terminal Equipment (DTE), RS-422 and RS-485.

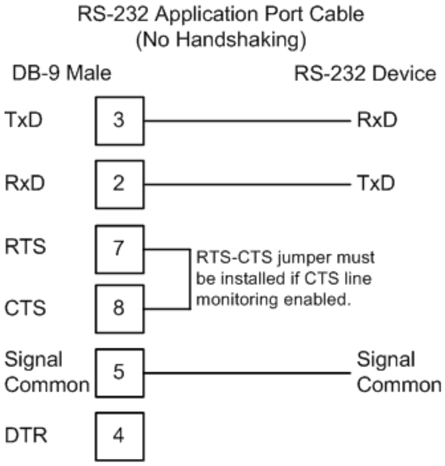
4.2.1 RS-232 - Null Modem (DTE with Hardware Handshaking)

This type of connection is used when the device connected to the gateway requires hardware handshaking (control and monitoring of modem signal lines). To enable hardware handshaking, set the port configuration to use RTS/CTS handshaking. (For MBS protocol, set the *Use CTS Line* parameter to **Yes**. For ASCII protocol, set the *Handshaking* parameter to **Yes**.)



4.2.2 RS-232 - Null Modem (DTE without Hardware Handshaking)

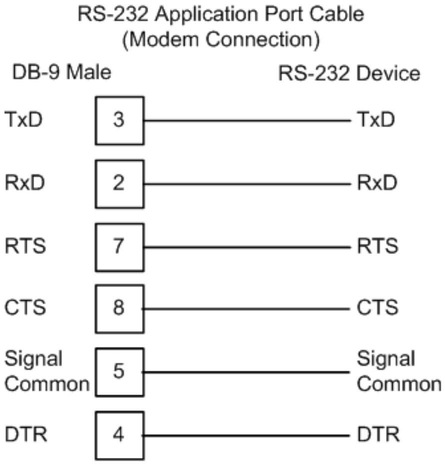
This type of connection can be used to connect the gateway to a computer or field device communication port.


RS-232 Application Port Cable
(No Handshaking)
DB-9 Male
RS-232 Device
TxD
3
RxD
RxD
2
TxD
RTS
7
CTS
8
RTS-CTS jumper must be installed if CTS line monitoring enabled.
Signal Common
5
Signal Common
DTR
4

Note: If the port is configured to use RTS/CTS handshaking, then a jumper is required between the RTS and the CTS line on the gateway connection.

4.2.3 RS-232 - DTE to DCE Modem Connection

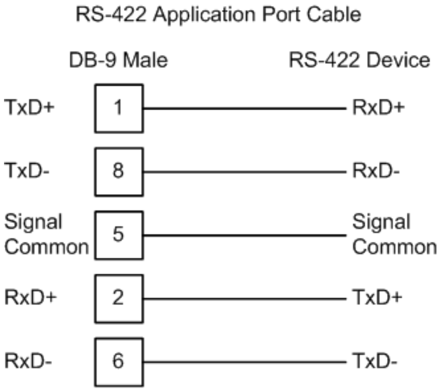
This type of connection is required between the gateway and a modem or other communication device.



For most modem applications, RTS/CTS handshaking should be enabled in the port configuration.

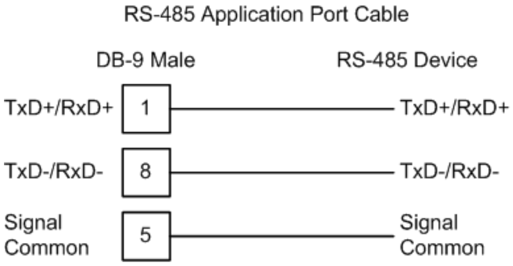
4.2.4 RS-422 Interface Connection

The following illustration applies when the RS-422 interface is selected.



4.2.5 RS-485 Interface Connection

The following illustration applies when the RS-485 interface is selected.



NOTE: This type of connection is commonly called a *RS-485 half-duplex, 2-wire* connection. If you have RS-485 4-wire, full-duplex devices, they can be connected to the gateway's serial ports by wiring together the TxD+ and RxD+ from the two pins of the full-duplex device to Pin 1 on the gateway and wiring together the TxD- and RxD- from the two pins of the full-duplex device to Pin 8 on the gateway. As an alternative, you could try setting the gateway to use the RS-422 interface and connect the full-duplex device according to the RS-422 wiring diagram. For additional assistance, please contact ProSoft Technical Support.

5 EIP Protocol

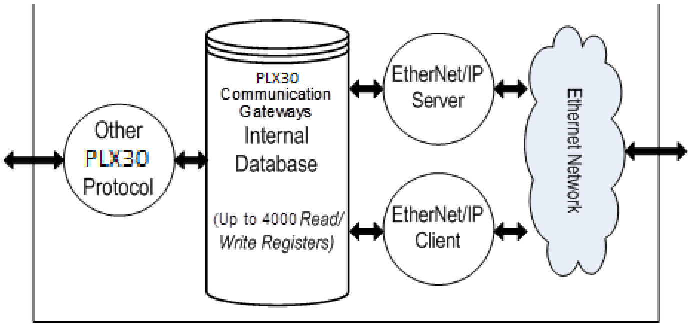
In This Chapter

This chapter contains information specific to the PLX3x-series gateway EtherNet/IP (EIP) protocol driver.

5.1 EIP Functional Overview

The PLX3x-series EIP gateway can be used to interface many different protocols into the Rockwell Automation family of processors as well as other software-based solutions.

The following illustration shows the functionality of the EtherNet/IP protocol.



The EIP driver supports the following connections:

Class	Connection Type	Number of Connections
Class 1	I/O	Depends on the gateway model: PLX31-EIP-MBTCP - 2 connections PLX31-EIP-MBS - 2 connections PLX31-EIP-MBS4 - 8 connections PLX31-EIP-ASCII - 1 connection PLX31-EIP-ASCII4 - 4 connections PLX31-EIP-SIE – 2 connections
Class 3	Connected Client	2
	Unconnected Client	1
	Server	5

5.1.1 EtherNet/IP™ Client

In Client mode, the gateway controls the read/write data transfer between the gateway and other EtherNet/IP devices.

Number of Clients Supported	Connected Clients: 2 Unconnected Clients: 1
Command List	Support for 100 commands per Client, each configurable for command type, IP address, register to/from addressing and word/bit count.
Polling of Command List	User-configurable polling of commands, including disabled, continuous and on change of data (write only).
Other Configurable Parameters	Number of Commands (up to 100 per Client) Min Command Delay Response Timeout Retry Count Command Error Pointer

5.2 EIP Configuration

5.2.1 EIP Class 3 Server Connection

The EIP Class 3 Server Connection is used when the gateway is acting as a server (slave) device responding to message instructions initiated from a Client (Master) device such as an HMI, DCS, or PLC5.

Configuring EIP Class 3 Server Connections in PCB

The PLX3x Server connection file size is user selectable for 100 or 1000 integers. If a value of 100 is selected valid registers will be from N10:0 to N10:99. If a value of 1000 is selected valid registers will be from N10:0 to N10:999.

Accessing the Gateway's Internal Memory

Data Type	Tag Name	Length of Each Element in CIP Message	Array Range for 4000 Element Database
BOOL	BOOLData[ ]	1	0 to 63999
Bit Array	BITAData[ ]	4	0 to 1999
SINT	SINTData[ ]	1	0 to 7999
INT	INT_Data[ ]	2	0 to 3999
DINT	DINTData[ ]	4	0 to 1999
REAL	REALData[ ]	4	0 to 1999

The following tables define the relationship of the gateway's internal database to the addresses required in the MSG instructions:

MSG Instruction Type - CIP

Database Address	CIP Integer	CIP Boolean	CIP Bit Array	CIP Byte	CIP DINT	CIP Real
0	Int_data[0]	BoolData[0]	BitAData[0]	SIntData[0]	DIntData[0]	RealData[0]
999	Int_data[999]	BoolData[15984]		SIntData[1998]		
1000	Int_data[1000]	BoolData[16000]	BitAData[500]	SIntData[2000]	DIntData[500]	RealData[500]
1999	Int_data[1999]	BoolData[31984]		SIntData[3998]		
2000	Int_data[2000]	BoolData[32000]	BitAData[1000]	SIntData[4000]	DIntData[1000]	RealData[1000]
2999	Int_data[2999]	BoolData[47984]		SIntData[5998]		
3000	Int_data[3000]	BoolData[48000]	BitAData[1500]	SIntData[6000]	DIntData[1500]	RealData[1500]
3999	Int_data[3999]	BoolData[63999]		SIntData[9998]		

MSG Instruction Type - PCCC

Database Address	File size 100	Database Address	File size 100
0	N10:0	0	N10:0
999	N19:99	999	N19:99
1000	N20:0	1000	N20:0
1999	N29:99	1999	N29:99
2000	N30:0	2000	N30:0

EtherNet/IP Explicit Messaging Server Command Support

The following commands are supported in the PLX3x gateway.

Basic Command Set Functions

Command	Function	Definition	Supported in Server
0x00	N/A	Protected Write	X
0x01	N/A	Unprotected Read	X
0x02	N/A	Protected Bit Write	X
0x05	N/A	Unprotected Bit Write	X
0x08	N/A	Unprotected Write	X

PLC-5 Command Set Functions

Command	Function	Definition	Supported in Server
0x0F	0x00	Word Range Write (Binary Address)	X
0x0F	0x01	Word Range Read (Binary Address)	X
0x0F		Typed Range Read (Binary Address)	X
0x0F		Typed Range Write (Binary Address)	X
0x0F	0x26	Read-Modify-Write (Binary Address)	
0x0F	0x00	Word Range Write (ASCII Address)	X
0x0F	0x01	Word Range Read (ASCII Address)	X
0x0F	0x26	Read-Modify-Write (ASCII Address)	

SLC-500 Command Set Functions

Command	Function	Definition	Supported in Server
0x0F	0xA1	Protected Typed Logical Read With Two Address Fields	X
0x0F	0xA2	Protected Typed Logical Read With Three Address Fields	X
0x0F	0xA9	Protected Typed Logical Write With Two Address Fields	X
0x0F	0xAA	Protected Typed Logical Write With Three Address Fields	X
0x0F	0xAB	Protected Typed Logical Write With Mask (Three Address Fields)	

5.2.2 EIP Class 1 Connection

The EIP Class 1 Connection is used when the gateway acts as an EIP adapter transferring data to and from a PLC (the EIP scanner), using a direct I/O connection. Direct I/O connections can be used to transfer large amounts of data quickly.

The PLX3x-series EIP gateway can handle up to eight I/O connections (depending on the model), each with 248 words of input data and 248 words of output data. Rockwell Automation customers running RSLogix 5000 v.20 and higher can take advantage of premier integration with an Add-on profile.

Adding the Gateway to RSLogix5000 v.20

1. Open up RSLinx and browse to the PLX3x gateway.
2. Open up a short cut window by right clicking on the gateway.
3. Select **Upload EDS from device.**

Note: RSLogix5000 may need to be restarted in order to complete the installation.

4. Once RSLogix5000 has been restarted, add a **New Module** under the EtherNet/IP bridge in the I/O tree.
5. In the *Module Type Vendor Filters* window set the filter options to ProSoft Technology.
6. Select the corresponding PLX3x gateway and click **Create**
7. In the next window set the IP address to the address of the PLX3x gateway. To add I/O connections click the **Change** button.
8. Here up to eight I/O connections can be added. The I/O connections have a fixed size of 496 bytes of input data and 496 bytes of output data. When finished click ok.
9. In the Module properties window each I/O connection can be configured with its own RPI time.

Adding the Gateway to RSLogix5000 v.16 through v.19

(Class 1 connections are not supported in v.15 and older)

1. Add a **New Module** under the EtherNet/IP bridge in the I/O tree.
2. Click **Find** and search for *Generic EtherNet Bridge* click **Create**.
3. Set the IP address to the gateway. This creates the communication path from the processor to the PLX3x gateway

4. Next add a **New Module** under the *Generic EtherNet Bridge* and add a CIP Connection. Here the parameters for the I/O connection are specified. The input and output sizes need to match the input and output sizes configured in PCB. The **Address** field value represents the connection number in PCB. By default all of the connections have 248 Input words, 248 Output words, and 0 Configuration words. The Comm format should be set to Data type INT, and the Assembly instances should be "1" for input, "2" for output, and "4" for configuration.
5. A CIP Connection will need to be added and configured for each I/O connection.

Configuring EIP Class 1 Connections in PCB

There are four configurable parameters for each I/O connection in PCB.

Parameter	**Value Range**	**Description**
Input Data Address	0 to 3999	This parameter specifies the starting address within the gateway’s virtual database for data transferred from the PLC to the module
Input Size	0 to 248	This parameter specifies the number of Integers being transferred to the PLC's input image (248 integers max)
Output Data Address	0 to 3999	This parameter specifies the starting address within the gateway’s virtual database for data transferred from the module to the PLC
Output Size	0 to 248	This parameter specifies the number of integers being transferred to the PLC's output image (248 integers max)

5.2.3 EIP Class 3 Client/UClient [x] Connection

The PLX3x gateway supports two connected Clients and one unconnected Client (most devices use connected Clients; be sure refer to the user manual of the target device for verification).

The EIP Class 3 Client [x] Connections are used when the gateway is acting as a Client/Master initiating message instructions to the server/slave devices. The PLX3x EIP protocol supports three Connected Client Connections. Typical applications include SCADA systems, and SLC communication.

The EIP Class 3 UClient Connection is used when the gateway is acting as a Client/Master initiating message Instructions to the server/slave devices. The PLX3x EIP protocol supports one Unconnected Client Connection. Unconnected messaging is a type of Ethernet/IP explicit messaging that uses TCP/IP implementation. Certain devices, such as the AB Power Monitor 3000 series B, support unconnected messaging. Check your device documentation for further information about its Ethernet/IP implementation.

Class 3 Client/UClient [x]

This section specifies the configuration for the EIP Client (Master) device on the network port.

Parameter	Value	Description
Minimum Command Delay	0 to 65535 milliseconds	This parameter specifies the number of milliseconds to wait between the initial issuances of a command. This parameter can be used to delay all commands sent to servers to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.
Response Timeout	0 to 65535 milliseconds	This parameter specifies the amount of time in milliseconds that a Client will wait before re-transmitting a command if no response is received from the addressed server. The value to use depends on the type of communication network used, and the expected response time of the slowest device connected to the network.
Retry Count	**0** to **10**	This parameter specifies the number of times a command will be retried if it fails.

Class 3 Client/UClient[x] Commands

There is a separate command list for each of the different message types supported by the protocol. Each list is processed from top to bottom, one after the other, until all specified commands are completed, and then the polling process begins over again.

This section defines the EtherNet/IP commands to be issued from the gateway to server devices on the network. These commands can be used for data collection and/or control of devices on the TCP/IP network.

In order to interface the virtual database with Rockwell Automation Programmable Automation Controllers (PACs), Programmable Logic Controllers (PLCs), or other EtherNet/IP server devices, you must construct a command list.

The following tables describe the command list parameters for each message type.

Class 3 Client/UClient [x] Commands SLC500 2 Address Fields

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **Enable** - The Command is executed each scan of the command list **Disable**- The command is disabled and will not be executed **Conditional Write** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 3999	Specifies the database address in the modules internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0-65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command the command executes no more frequently than every 10 seconds.
Reg Count		Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **None** - No change is made (abcd) **Word swap** - The words are swapped (cdab) **Word and Byte swap** - The words and bytes are swapped (dcba) **Byte swap** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command
Slot	-1	Use a value of -1 when interfacing to an SLC 5/05. These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix rack, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	501 509	Specifies the function code to be used in the command. 501 – Protected Typed Read 509 – Protected Typed Write
File Type	Binary Counter Timer Control Integer Float	Specifies the file type to be associated with the command.

Parameter	Value	Description
	ASCII String Status	
File Number	-1	Specifies the PLC-5 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.
Element Number		Specifies the element in the file where the command will start.
Comment		This field can be used to give a 32 character comment to the command.

Class 3 Client/UClient [x] Commands SLC500 3 Address Fields

This command is typically used when accessing data in a Timer or Counter. I.e. T.1.1.2 is the address of the accumulator in Timer 1.

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **Enable** - The Command is executed each scan of the command list **Disable**- The command is disabled and will not be executed **Conditional Write** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 3999	Specifies the database address in the modules internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command the command executes no more frequently than every 10 seconds.
Reg Count		Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **None** - No change is made (abcd) **Word swap** - The words are swapped (cdab) **Word and Byte swap** - The words and bytes are swapped (dcba) **Byte swap** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command
Slot	-1	Use a value of -1 when interfacing to an SLC 5/05. These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	502 510 511	Specifies the function code to be used in the command. 502 - Protected Typed Read 510 - Protected Typed Write 511 - Protected Typed Write w/Mask

Parameter	Value	Description
File Type	Binary Counter Timer Control Integer Float ASCII String Status	Specifies the file type to be associated with the command.
File Number	-1	Specifies the SLC 500 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.
Element Number		Specifies the element in the file where the command will start.
Sub Element		Specifies the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes.
Comment		This field can be used to give a 32 character comment to the command.

Class 3 Client/UClient [x] Commands PLC5 Binary

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **Enable** - The Command is executed each scan of the command list **Disable**- The command is disabled and will not be executed **Conditional Write** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 3999	Specifies the database address in the modules internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command the command executes no more frequently than every 10 seconds.
Reg Count		Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **None** - No change is made (abcd) **Word swap** - The words are swapped (cdab) **Word and Byte swap** - The words and bytes are swapped (dcba) **Byte swap** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command
Slot	-1	Use a value of -1 when interfacing to a PLC5 These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	100 101 102	Specifies the function code to be used in the command. 100 - Word Range Write 101 - Word Range Read 102 - Read-Modify-Write
File Number	-1	Specifies the PLC5 file number to be associated with the command. If a value of -1 is entered for the parameter, the field will not be used in the command, and the default file will be used.

Parameter	Value	Description
Element Number		Specifies the element in the file where the command will start.
Sub Element		Specifies the sub-element to be used with the command. Refer to the AB documentation for a list of valid sub-element codes.
Comment		This field can be used to give a 32 character comment to the command.

Class 3 Client/UClient [x] Commands PLC5 ASCII

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **Enable** - The Command is executed each scan of the command list **Disable**- The command is disabled and will not be executed **Conditional Write** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 3999	Specifies the database address in the modules internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command the command executes no more frequently than every 10 seconds.
Reg Count		Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **None** - No change is made (abcd) **Word swap** - The words are swapped (cdab) **Word and Byte swap** - The words and bytes are swapped (dcba) **Byte swap** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies IP address of the target device to be addressed by this command
Slot	-1	Use a value of -1 when interfacing to a PLC5 These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	150 151 152	Specifies the function code to be used in the command. 150 - Word Range Write 151 - Word Range Read 152 - Read-Modify-Write
File String		Specifies the PLC-5 Address as a string. For example N10:300
Comment		This field can be used to give a 32 character comment to the command.

Class 3 Client/UClient [x] Commands Controller Tag Access

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **Enable** - The Command is executed each scan of the command list **Disable**- The command is disabled and will not be executed **Conditional Write** - The Command executes only if the internal data associated with the command changes
Internal Address	0 to 3999	Specifies the database address in the modules internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from specified data area.
Poll Interval	0 to 65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command the command executes no more frequently than every 10 seconds.
Reg Count		Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **None** - No change is made (abcd) **Word swap** - The words are swapped (cdab) **Word and Byte swap** - The words and bytes are swapped (dcba) **Byte swap** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command
Slot	-1	Use a value of -1 when interfacing to a PLC5 These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	332 333	Specifies the function code to be used in the command. 332 - CIP Data Table Read 333 - CIP Data Table Write
Data Type	Bool SINT INT DINT REAL DWORD	Specifies the data type of the target controller tag name.
Tag Name		Specifies the controller tag in the target PLC.

Class 3 Client/UClient [x] Commands Basic

Parameter	Value	Description
Enable	Enable Disable Conditional Write	Specifies if the command should be executed and under what conditions. **Enable** - The command is executed each scan of the command list **Disable**- The command is disabled and will not be executed **Conditional Write** - The command executes only if the internal data associated with the command changes
Internal Address	0 to 3999	Specifies the database address in the module's internal database to be associated with the command. If the command is a read function, the data received in the response message is placed at the specified location. If the command is a write function data used in the command is sourced from the specified data area.
Poll Interval	0-65535	Specifies the minimum interval to execute continuous commands. The parameter is entered in 1/10 of a second. If a value of 100 is entered for a command the command executes no more frequently than every 10 seconds.
Reg Count		Specifies the number of data points to be read from or written to the target device.
Swap Code	None Word swap Word and Byte swap Byte swap	Specifies if the data from the server is to be ordered differently than it was received. This parameter is typically used when dealing with floating-point or other multi-register values. **None** - No change is made (abcd) **Word swap** - The words are swapped (cdab) **Word and Byte swap** - The words and bytes are swapped (dcba) **Byte swap** - The bytes are swapped (badc)
IP Address	xxx.xxx.xxx.xxx	Specifies the IP address of the target device to be addressed by this command
Slot	-1	Use a value of -1 when interfacing to an SLC 5/05. These devices do not have a slot parameter. When addressing a processor in a ControlLogix or CompactLogix, the slot number corresponds to the slot in the rack containing the controller being addressed.
Func Code	1 2 3 4 5	Specifies the function code to be used in the command. 1 - Protected Write 2 - Unprotected Read 3 - Protected Bit Write 4 - Unprotected Bit Write 5 - Unprotected Write
Word Address		Specifies the word address where to start the operation.
Comment		This field can be used to give a 32 character comment to the command.

5.3 EIP Diagnostics

5.3.1 PCB Diagnostics Menu

The best way to troubleshoot the EIP driver is to use ProSoft Configuration Builder to access the diagnostic capabilities of the gateway through the Ethernet debug port. For instructions on how to access *Diagnostics*, see *Using Diagnostics in ProSoft Configuration Builder* (page 36).

The following table summarizes the status information available in PCB for the EIP driver.

Connection Type	**Submenu Item**	**Description**
EIP Class 1	Config	Configuration settings for Class 1 Connections.
	Status	Status of the Class 1 Connections: Displays any configuration error present, as well as the number of Class 1 Connections.
EIP Class 3 Server	Config	Configuration settings for Class 3 Server Connections.
	Comm Status	Status information for each Class 3 Server Connection: Displays port numbers, IP addresses, socket status, and read and write counts.
EIP Class 3 Client/UClient [x]	Config	Configuration settings for Class 3 Client/UClient Connections.
	Comm Status	Status information for Class 3 Client/UClient [x] commands. Displays a summary of all the errors resulting from Class 3 Client/UClient [x] commands.
	Commands	Configuration for the Class 3 Client/UClient [x] command list.
	Cmd Errors (Decimal)	Displays current error codes for each command on the Class 3 Client/UClient [x] command list in decimal number format. A zero means there is currently no error for the command.
	Cmd Errors (Hex)	Displays current error codes for each command on the Class 3 Client/UClient [x] command list in hexadecimal number format. A zero means there is currently no error for the command.

5.3.2 EIP Status Data in Upper Memory

The EIP driver has an associated status data area located in the gateway's upper memory. The Data Map functionality of the gateway can be used to map this data into the normal user data range of the gateway's database. See *Using the CommonNet Data Map* (page 25).

Note that all the status values are initialized to zero (0) at power-up, cold boot and during warm boot.

EIP Client Status Data

The following table lists the addresses in upper memory where general error and status data for each EIP connected and unconnected Client can be found.

EIP Client	Address Range
Connected Client 0	7900 through 7909
Connected Client 1	8100 through 8109
Unconnected Client 0	12800 through 12809

The content of each Client's status data area is structured the same. The following table describes the content of each register in the status data area.

Offset	Description
0	Number of Command Requests
1	Number of Command Responses
2	Number of Command Errors
3	Number of Requests
4	Number of Responses
5	Number of Errors Sent
6	Number of Errors Received
7	Reserved
8	Current Error Code
9	Last Error Code

EIP Client Command List Error Data

A status/error code is held in upper memory for each command in each EIP Client's command list.

The following table lists the addresses in upper memory that hold command list error data for each EIP Client.

EIP Client	Address Range
Connected Client 0	7910 through 8009
Connected Client 1	8110 through 8209
Unconnected Client 0	12810 through 12909

The first word in each Client's command list error data area contains the status/error code for the first command in the Client's command list. Each successive word in the command error list is associated with the next command in the list. Therefore, the size of the command list error data area depends on the number of commands defined.

The structure of the command list error data area (which is the same for all Clients) is displayed in the following table.

Offset	**Description**
0	Command #1 Error Code
1	Command #2 Error Code
2	Command #3 Error Code
3	Command #4 Error Code
4	Command #5 Error Code
.	.
.	.
.	.
97	Command #98 Error Code
98	Command #99 Error Code
99	Command #100 Error Code

A non-zero error code indicates an error. To interpret the status/error codes, refer to EIP Error Codes (page 72).

EIP Server Status Data

The following table lists the addresses in upper memory that hold status data for each EIP server.

EIP Server	**Address Range**
0	8900 through 8915
1	8916 through 8931
2	8932 through 8947
3	8948 through 8963
4	8964 through 8979

The content of each server's status data area is structured the same. The following table describes the content of each register in the status data area.

Offset	**Description**
0 through 1	Connection State
2 through 3	Open Connection Count
4 through 5	Socket Read Count
6 through 7	Socket Write Count
8 through 15	Peer IP

5.3.3 EIP Error Codes

The gateway error codes are listed in this section. Error codes returned from the command list process are stored in the command list error memory region. A word is allocated for each command in the memory area. The error codes are formatted in the word as follows: The least-significant byte of the word contains the extended status code and the most-significant byte contains the status code.

Use the error codes returned for each command in the list to determine the success or failure of the command. If the command fails, use the error code to determine the cause of failure.

Note: The gateway specific error codes (not EtherNet/IP/PCCC compliant) are returned from within the gateway and never returned from an attached EtherNet/IP/PCCC slave device. These are error codes that are part of the EtherNet/IP/PCCC protocol or are extended codes unique to this gateway. The most common errors for the EtherNet/IP/PCCC protocol are shown in the following tables:

Local STS Error Codes

Code (Int)	Code (Hex)	Description
0	0x0000	Success, no error
256	0x0100	DST node is out of buffer space
512	0x0200	Cannot guarantee delivery (Link Layer)
768	0x0300	Duplicate token holder detected
1024	0x0400	Local port is disconnected
1280	0x0500	Application layer timed out waiting for response
1536	0x0600	Duplicate node detected
1792	0x0700	Station is offline
2048	0x0800	Hardware fault

Remote STS Error Codes

Code (Int)	Code (Hex)	Description
0	0x0000	Success, no error
4096	0x1000	Illegal command or format
8192	0x2000	Host has a problem and will not communicate
12288	0x3000	Remote node host is missing, disconnected or shut down
16384	0x4000	Host could not complete function due to hardware fault
20480	0x5000	Addressing problem or memory protect rungs
24576	0x6000	Function not allowed due to command protection selection
26872	0x7000	Processor is in Program mode
-32768	0x8000	Compatibility mode file missing or communication zone problem
-28672	0x9000	Remote node cannot buffer command
-24576	0xA000	Wait ACK (1775-KA buffer full)
-20480	0xB000	Remote node problem due to download
-16384	0xC000	Wait ACK (1775-KA buffer full)
-12288	0xD000	Not used
-8192	0xE000	Not used
	0xF0nn	Error code in the EXT STS byte (nn contains EXT error code)

Errors When EXT STS Is Present

Code (Int)	Code (Hex)	Description
-4096	0xF000	Not used
-4095	0xF001	A field has an illegal value
-4094	0xF002	Less levels specified in address than minimum for any address
-4093	0xF003	More levels specified in address than system supports
-4092	0xF004	Symbol not found
-4091	0xF005	Symbol is of improper format
-4090	0xF006	Address does not point to something usable
-4089	0xF007	File is wrong size
-4088	0xF008	Cannot complete request
-4087	0xF009	Data or file is too large
-4086	0xF00A	Transaction size plus word address is too large
-4085	0xF00B	Access denied, improper privilege
-4084	0xF00C	Condition cannot be generated - resource is not available
-4083	0xF00D	Condition already exists - resource is already available
-4082	0xF00E	Command cannot be executed
-4081	0xF00F	Histogram overflow
-4080	0xF010	No access
-4079	0xF011	Illegal data type
-4078	0xF012	Invalid parameter or invalid data

Code (Int)	Code (Hex)	Description
-4077	0xF013	Address reference exists to deleted area
-4076	0xF014	Command execution failure for unknown reason
-4075	0xF015	Data conversion error
-4074	0xF016	Scanner not able to communicate with 1771 rack adapter
-4073	0xF017	Type mismatch
-4072	0xF018	1171 Gateway response was not valid
-4071	0xF019	Duplicate label
-4070	0xF01A	File is open; another node owns it
-4069	0xF01B	Another node is the program owner
-4068	0xF01C	Reserved
-4067	0xF01D	Reserved
-4066	0xF01E	Data table element protection violation
-4065	0xF01F	Temporary internal problem

Gateway Specific Error (not EIPCompliant)

Code (Int)	Code (Hex)	Description
-1	0xFFFF	CTS modem control line not set before transmit
-2	0xFFFE	Timeout while transmitting message
-10	0xFFF6	Timeout waiting for DLE-ACK after request
-11	0xFFF5	Timeout waiting for response after request
-12	0xFFF4	Reply data does not match requested byte count
-20	0xFFEC	DLE-NAK received after request
-21	0xFFEB	DLE-NAK sent after response
-200	0xFF38	DLE-NAK received after request

TCP/IP Interface Errors

Error (Int)	Error (Hex)	Description
-33	0xFFDF	Failed to connect to target
-34	0xFFDE	Failed to register session with target (timeout)
-35	0xFFDD	Failed forward open response timeout
-36	0xFFDC	PCCC/Tag command response timeout
-37	0xFFDB	No TCP/IP connection error

Common Response Errors

Error (Int)	Error (Hex)	Description
-40	0xFFD8	Invalid response length
-41	0xFFD7	CPF item count not correct
-42	0xFFD6	CPF address field error
-43	0xFFD5	CPF packet tag invalid
-44	0xFFD4	CPF bad command code
-45	0xFFD3	CPF status error reported
-46	0xFFD2	CPF incorrect connection ID value returned
-47	0xFFD1	Context field not matched
-48	0xFFD0	Incorrect session handle returned
-49	0xFFCF	CPF not correct message number

Register Session Response Errors

Error (Int)	Error (Hex)	Description
-50	0xFFCE	Message length received not valid
-51	0xFFCD	Status error reported
-52	0xFFCC	Invalid version

Forward Open Response Errors

Error (Int)	Error (Hex)	Description
-55	0xFFC9	Message length received not valid
-56	0xFFC8	Status error reported

PCCC Response Errors

Error (Int)	Error (Hex)	Description
-61	0xFFC3	Message length received not valid
-62	0xFFC2	Status error reported
-63	0xFFC1	CPF bad command code
-64	0xFFC0	TNS in PCCC message not matched
-65	0xFFBF	Vendor ID in PCCC message not matched
-66	0xFFBE	Serial number in PCCC message not matched

5.4 EIP Reference

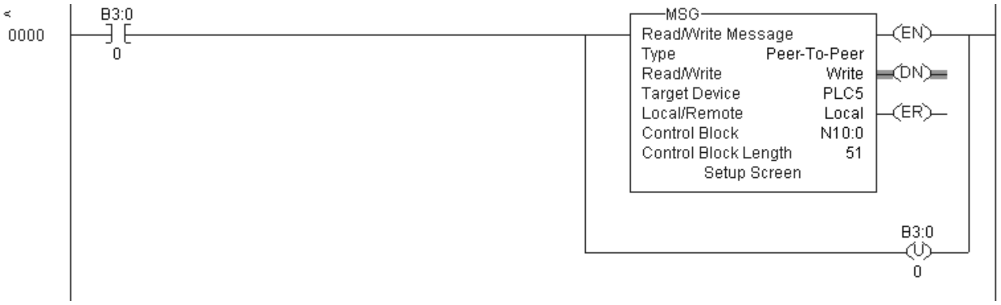
5.4.1 SLC and MicroLogix Specifics

Messaging from a SLC 5/05

The gateway can be used to receive messages from a SLC 5/05 containing an Ethernet interface. The gateway supports both read and write commands. A discussion of each operation is provided in the following topics.

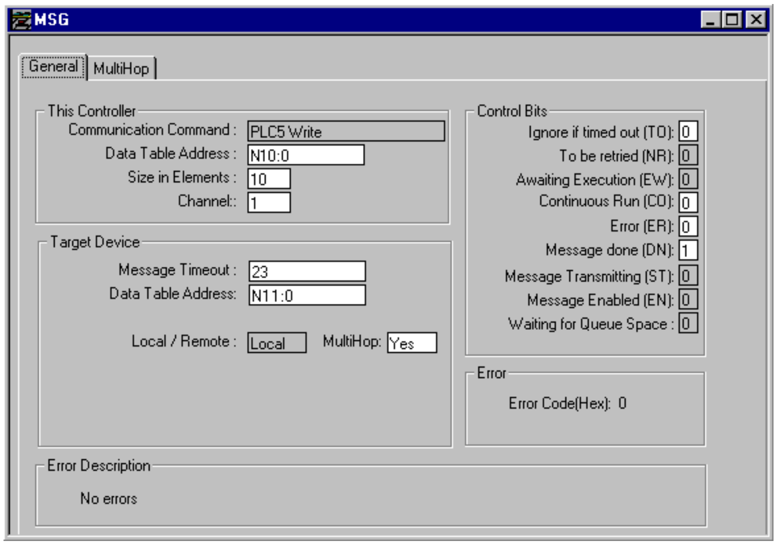
SLC5/05 Write Commands

Write commands transfer data from the SLC processor to the gateway. An example rung used to execute a write command is shown in the following diagram:



Read/Write Message
Type Peer-To-Peer
Read/Write Write
Target Device PLC5
Local/Remote Local
Control Block N10:0
Control Block Length 51
Setup Screen

Set the **READ/WRITE** parameter to **WRITE.** The gateway supports a **TARGET DEVICE** parameter value of **500CPU** or **PLC5.** In order to complete the configuration of the MSG instruction, select the **SETUP SCREEN** area of the MSG object. This displays the following dialog box.



This Controller
Communication Command : PLC5 Write
Data Table Address : N10:0
Size in Elements : 10
Channel:: 1

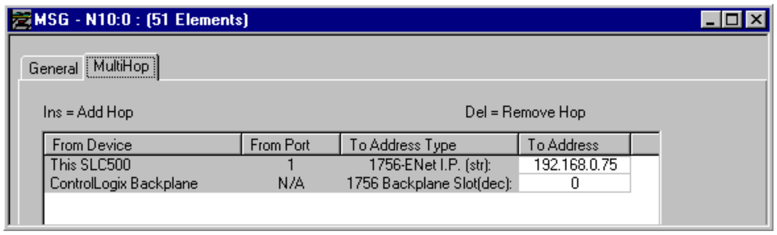
Target Device
Message Timeout : 23
Data Table Address: N11:0
Local / Remote : Local
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 1
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 0

Error Description
No errors

The **TARGET DEVICE DATA TABLE ADDRESS** must be set to a valid file element (such as, N11:0) for SLC and PLC5 messages. The **MULTIHOP** option must be set to **YES.** The **MULTIHOP** tab portion of the dialog box must be completed as displayed in the following window:



MSG - N10:0 : (51 Elements)

Ins = Add Hop

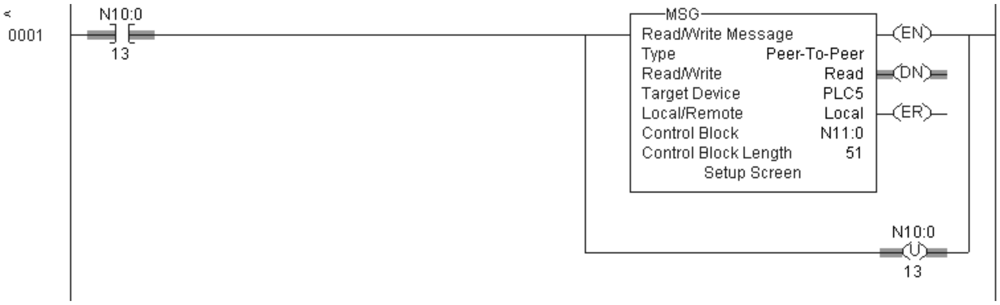
Del = Remove Hop

From Device	From Port	To Address Type	To Address
This SLC500	1	1756-ENet I.P. (str):	192.168.0.75
ControlLogix Backplane	N/A	1756 Backplane Slot(dec):	0

Set the IP address value to the gateway’s Ethernet IP address. The "Insert" key must be pressed to add the second line for ControlLogix Backplane and set the slot number to zero.

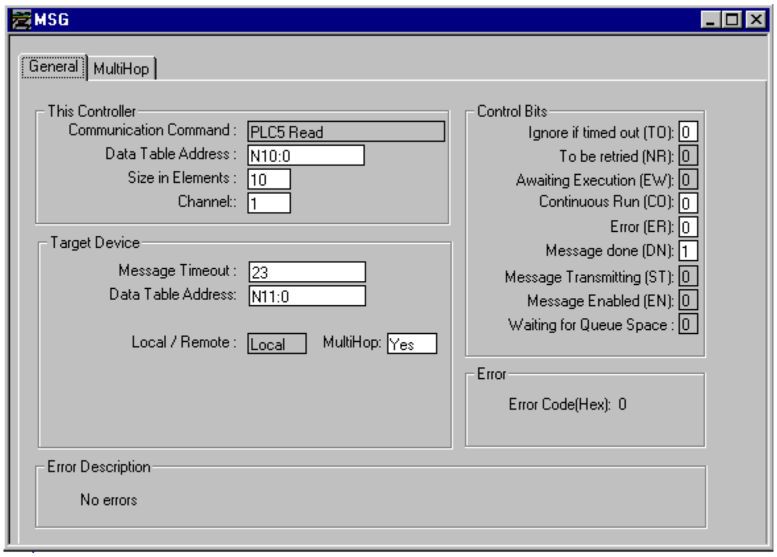
SLC5/05 Read Commands

Read commands transfer data to the SLC processor from the gateway. An example rung used to execute a read command is shown in the following diagram:



Read/Write Message	
Type	Peer-To-Peer
Read/Write	Read
Target Device	PLC5
Local/Remote	Local
Control Block	N11:0
Control Block Length	51
Setup Screen	

Set the **READ/WRITE** parameter to **READ.** The gateway supports a **TARGET DEVICE** parameter value of **500CPU** or **PLC5.** In order to complete the configuration of the MSG instruction, select the **SETUP SCREEN** area of the MSG object. This displays the following dialog box.



This Controller
Communication Command : PLC5 Read
Data Table Address : N10:0
Size in Elements : 10
Channel: 1

Target Device
Message Timeout : 23
Data Table Address: N11:0
Local / Remote : Local
MultiHop: Yes

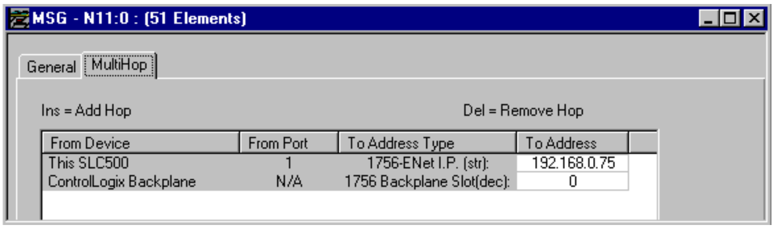
Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 1
Message Transmitting (ST): 0
Message Enabled (EN): 0
Waiting for Queue Space : 0

Error
Error Code(Hex): 0

Error Description
No errors

The **TARGET DEVICE DATA TABLE ADDRESS** must be set to a valid file element (such as, N11:0) for SLC and PLC5 messages. The **MULTIHOP** option must be set to **YES.**

Fill in the **MULTIHOP** tab portion of the dialog box as shown in the following illustration.



Ins = Add Hop
Del = Remove Hop

From Device	From Port	To Address Type	To Address
This SLC500	1	1756-ENet I.P. (str):	192.168.0.75
ControlLogix Backplane	N/A	1756 Backplane Slot(dec):	0

Set the IP address value to the gateway's Ethernet IP address. The "Insert" key must be pressed to add the second line for ControlLogix Backplane and set the slot number to zero.

SLC File Types

This section contains information specific to the SLC and MicroLogix processor based family when used with the PCCC command set. The SLC and MicroLogix processor commands support a file type field entered as a single character to denote the data table to interface with in the command. The following table defines the relationship of the file types accepted by the Gateway and the SLC file types.

File Type	Description
S	Status
B	Bit
T	Timer
C	Counter
R	Control
N	Integer
F	Floating-point
Z	String
A	ASCII

The File Type Command Code is the ASCII character code value of the File Type letter. This is the value to enter into the "File Type" parameter of the PCCC Command configurations in the data tables in the ladder logic.

Additionally, the SLC specific functions (502, 510 and 511) support a sub-element field. This field selects a sub-element field in a complex data table. For example, to obtain the current accumulated value for a counter or timer, the sub-element field should be set to 2.

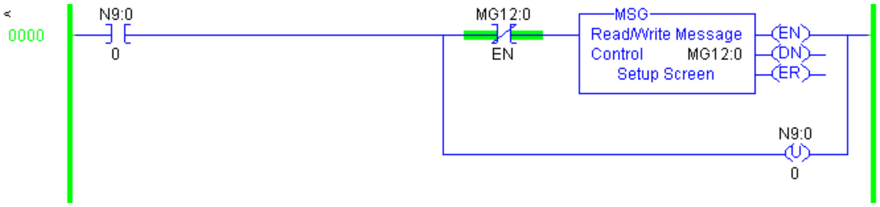
5.4.2 PLC5 Processor Specifics

Messaging from a PLC5

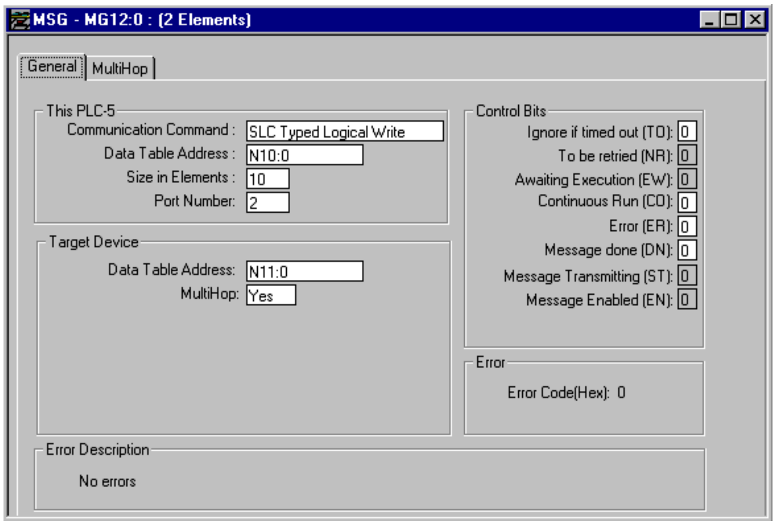
The gateway can be used to receive messages from a PLC5 containing an Ethernet interface. The gateway supports both read and write commands. A discussion of each operation is provided in the following topics:

PLC5 Write Commands

Write commands transfer data from the PLC5 processor to the gateway. An example rung used to execute a write command is shown in the following diagram:



In order to complete the configuration of the MSG instruction, select the **SETUP SCREEN** area of the MSG object. This displays the following dialog box.



This PLC-5
Communication Command : SLC Typed Logical Write
Data Table Address : N10:0
Size in Elements : 10
Port Number: 2

Target Device
Data Table Address: N11:0
MultiHop: Yes

Control Bits
Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 0
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error
Error Code(Hex): 0

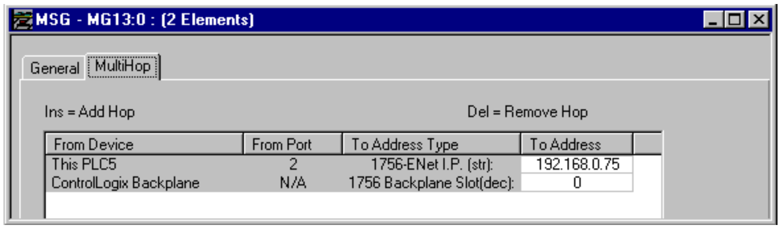
Error Description
No errors

Select the **COMMUNICATION COMMAND** to execute from the following list of supported commands.

- PLC5 Type Write
- PLC2 Unprotected Write
- PLC5 Typed Write to PLC
- PLC Typed Logical Write

The **TARGET DEVICE DATA TABLE ADDRESS** must be set to a valid file element (such as, N11:0) for SLC and PLC5 messages. For the PLC2 Unprotected Write message, set the address to the database index (such as, 1000) to consider with the command.

The **MULTIHOP** option must be set to **YES.** The **MULTIHOP** tab portion of the dialog box must be completed as shown in the following window:



MSG - MG13:0 : (2 Elements)

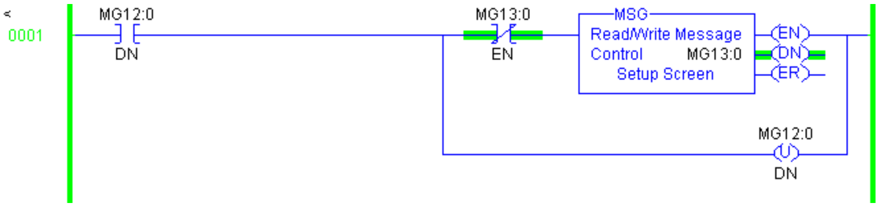
Ins = Add Hop   Del = Remove Hop

From Device	From Port	To Address Type	To Address
This PLC5	2	1756-ENet I.P. (str):	192.168.0.75
ControlLogix Backplane	N/A	1756 Backplane Slot(dec):	0

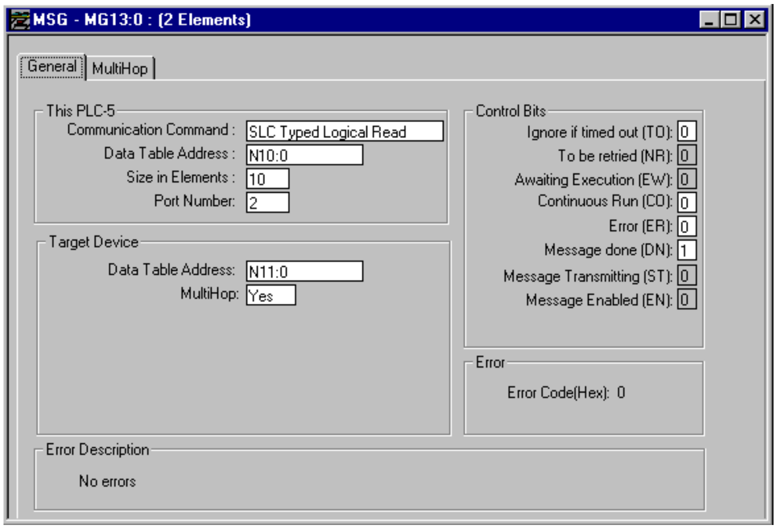
Set the IP address value to the gateway's Ethernet IP address. The "Insert" key must be pressed to add the second line for ControlLogix Backplane and set the slot number to zero.

PLC5 Read Commands

Read commands transfer data to the PLC5 processor from the gateway. An example rung used to execute a read command is shown in the following diagram:



In order to complete the configuration of the MSG instruction, select the **SETUP SCREEN** area of the MSG object. This displays the following dialog box.



MSG - MG13:0 : (2 Elements)

This PLC-5

Communication Command : SLC Typed Logical Read
Data Table Address : N10:0
Size in Elements : 10
Port Number: 2

Target Device

Data Table Address: N11:0
MultiHop: Yes

Control Bits

Ignore if timed out (TO): 0
To be retried (NR): 0
Awaiting Execution (EW): 0
Continuous Run (CO): 0
Error (ER): 0
Message done (DN): 1
Message Transmitting (ST): 0
Message Enabled (EN): 0

Error

Error Code(Hex): 0

Error Description

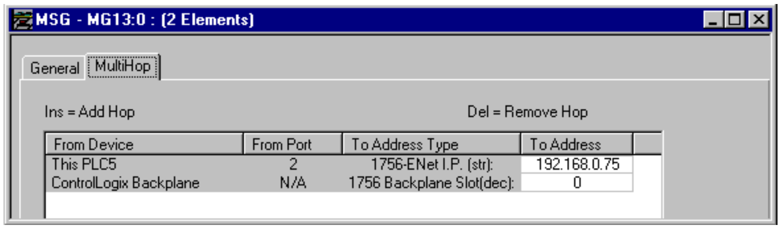
No errors

Select the **COMMUNICATION COMMAND** to execute from the following list of supported commands.

- PLC5 Type Read
- PLC2 Unprotected Read
- PLC5 Typed Read to PLC
- PLC Typed Logical Read

The **TARGET DEVICE DATA TABLE ADDRESS** must be set to a valid file element (such as, N11:0) for SLC and PLC5 messages. For the PLC2 Unprotected Read message, set the address to the database index (such as, 1000) to consider with the command.

The **MULTIHOP** option must be set to **YES.** The **MULTIHOP** tab portion of the dialog box must be completed as shown in the following window:



Ins = Add Hop

Del = Remove Hop

From Device	From Port	To Address Type	To Address
This PLC5	2	1756-ENet I.P. (str):	192.168.0.75
ControlLogix Backplane	N/A	1756 Backplane Slot(dec):	0

Set the IP address value to the gateway’s Ethernet IP address. The "Insert" key must be pressed to add the second line for ControlLogix Backplane and set the slot number to zero.

PLC-5 Sub-Element Fields

This section contains information specific to the PLC-5 processor with relation to the PCCC command set. The commands specific to the PLC-5 processor contain a sub-element code field. This field selects a sub-element field in a complex data table. For example, to obtain the current accumulated value for a counter or timer, the sub-element field should be set to 2. The tables below show the sub-element codes for PLC-5 complex data tables.

Timer / Counter

Code	Description
0	Control
1	Preset
2	Accumulated

Control

Code	Description
0	Control
1	Length
2	Position

PD*

Code	Description
0	Control
2	SP
4	Kp
6	Ki
8	Kd
26	PV

*All PD values are floating point values, so they are two words long.

BT

Code	Description
0	Control
1	RLEN
2	DLEN
3	Data file #
4	Element #
5	Rack/Grp/Slot

MG

Code	Description
0	Control
1	Error
2	RLEN
3	DLEN

5.4.3 ControlLogix and CompactLogix Processor Specifics

In order to exchange data between a Control/CompactLogix processor and the gateway, the MSG instruction is used. There are two basic methods of data transfer supported by the gateway when using the MSG instruction: Encapsulated PCCC messages and CIP Data Table messages. Either method can be used.

Encapsulated PCCC Messages

This section contains information specific to the Control/CompactLogix processor when used with the PCCC command set. The current implementation of the PCCC command set does not use functions that can directly interface with the Controller Tag Database. In order to interface with this database, the table-mapping feature provided by RSLogix 5000 must be used. The software permits the assignment of Controller Tag Arrays to virtual PLC 5 data tables. The ProSoft gateway using the PLC 5 command set defined in this document can then reach this controller data.

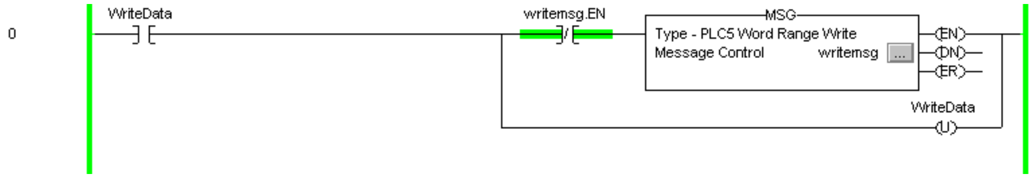
PLC5 and SLC5/05 processors containing an Ethernet interface use the encapsulated PCCC message method. The gateway simulates these devices and accepts both read and write commands. The following topics describe the support for the read and write operations.

Encapsulated PCCC Write Message

Write commands transfer data from the processor to the gateway. The following encapsulated PCCC commands are supported:

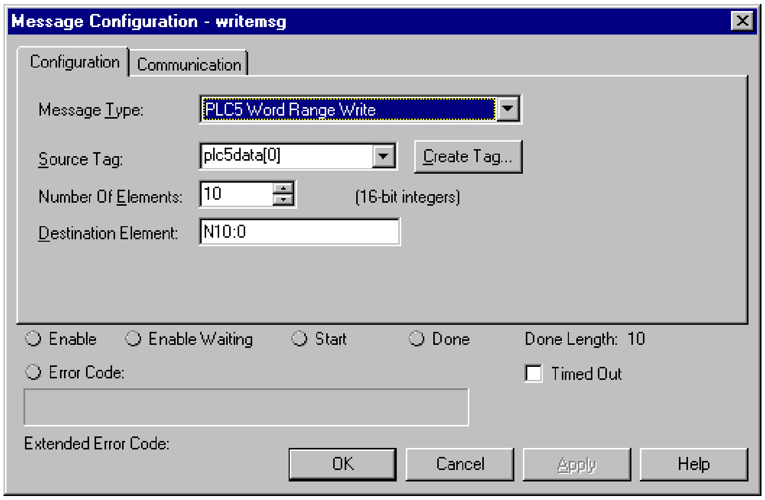
- ♦ PLC2 Unprotected Write
- ♦ PLC5 Typed Write
- ♦ PLC5 Word Range Write
- ♦ PLC Typed Write

An example rung used to execute a write command is shown in the following diagram:



Type - PLC5 Word Range Write
Message Control writemsg

The **MESSAGE CONFIGURATION** dialog box must be completed to define the data set to be transferred from the processor to the gateway. An example of the dialog box follows:



Message Configuration - writemsg

Configuration | Communication

Message Type: PLC5 Word Range Write

Source Tag: plc5data[0]

Number Of Elements: 10 (16-bit integers)

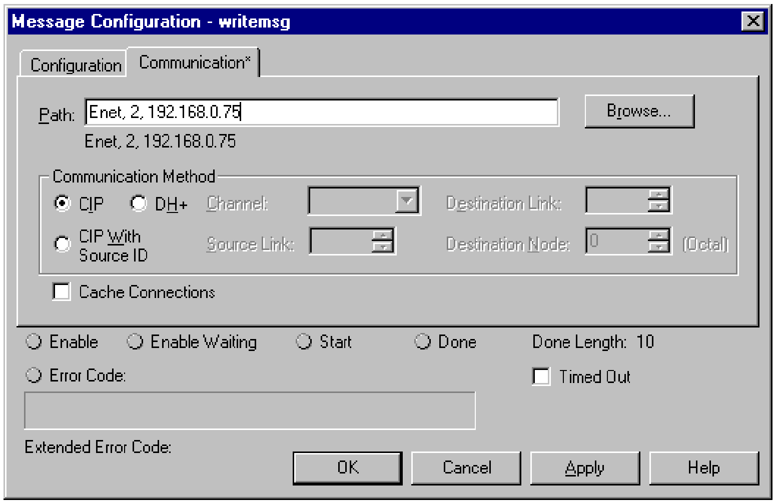
Destination Element: N10:0

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 10

○ Error Code: □ Timed Out

Extended Error Code:

Complete the dialog box for the data area to be transferred. For PLC5 and SLC messages, the **DESTINATION ELEMENT** should be an element in a data file (such as, N10:0). For the PLC2 Unprotected Write message, the **DESTINATION ELEMENT** is the address in the gateway's internal database and cannot be set to a value less than ten. This is not a limitation of the gateway but of the RSLogix software. For a PLC2 unprotected write or read function, the database address should be entered in octal format. The **COMMUNICATION** information must also be configured. The following is an example of the dialog box.



Message Configuration - writemsg

Configuration | Communication*

Path: Enet, 2, 192.168.0.75

Enet, 2, 192.168.0.75

Communication Method

● CIP ○ DH+ Channel: Destination Link:

○ CIP With Source ID Source Link: Destination Node: 0 (Octal)

□ Cache Connections

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 10

○ Error Code: □ Timed Out

Extended Error Code:

Verify that the **CIP** radio-button is selected as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown, the first element is "Enet", which is the user-defined name given to the 1756-ENET gateway in the chassis (the slot number of the ENET gateway can be substituted for the name), the second element, "2", represents the Ethernet port on the 1756-ENET gateway, and the last element of the path, "192.168.0.75", is the IP address of the gateway, the target for the message.

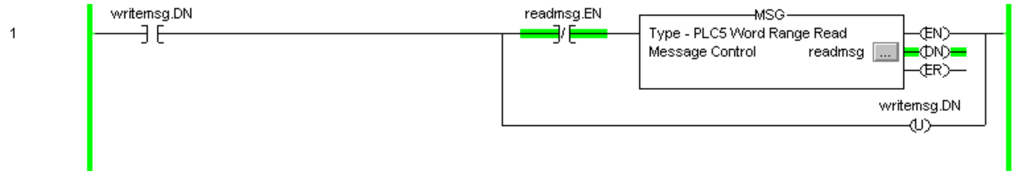
More complex paths are possible if routing to other networks using multiple 1756-ENET gateways and racks. Refer to the Support Knowledgebase for more information on Ethernet routing and path definitions.

Encapsulated PCCC Read Message

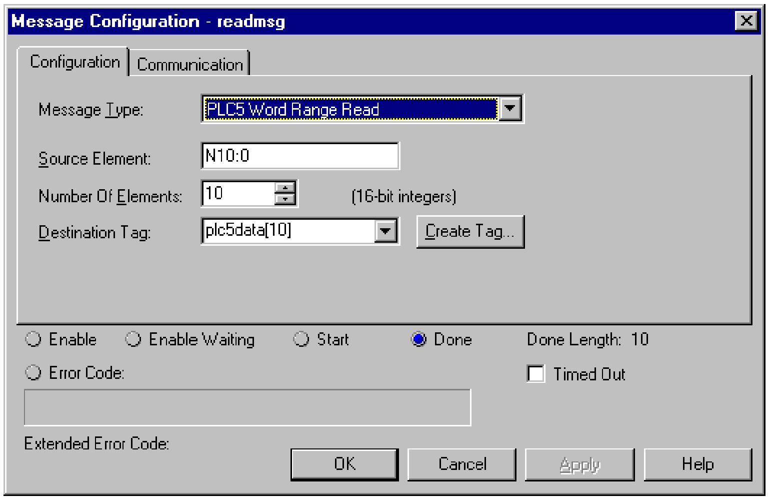
Read commands transfer data from the gateway to a processor. The following encapsulated PCCC commands are supported:

- PLC2 Unprotected Read
- PLC5 Typed Read
- PLC5 Word Range Read
- PLC Typed Read

An example rung used to execute a read command is shown in the following diagram:



The **MESSAGE CONFIGURATION** dialog box must be completed to define the data set to transfer to the processor from the gateway. An example of the dialog box follows:



Message Configuration - readmsg

Message Type: PLC5 Word Range Read

Source Element: N10:0

Number Of Elements: 10 (16-bit integers)

Destination Tag: plc5data[10]

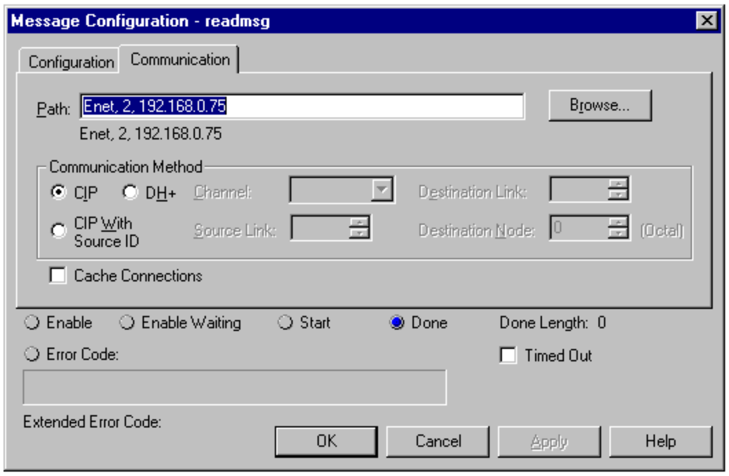
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 10

○ Error Code:

□ Timed Out

Extended Error Code:

Complete the dialog box for the data area to be transferred. For PLC5 and SLC messages, the **SOURCE ELEMENT** should be an element in a data file (such as, N10:0). For the PLC2 Unprotected Read message, the **SOURCE ELEMENT** is the address in the gateway's internal database and cannot be set to value less than ten. This is not a limitation of the gateway but of the RSLogix software. The **COMMUNICATION** information must also be configured. An example of the dialog box follows:



Message Configuration - readmsg

Path: Enet, 2, 192.168.0.75
Enet, 2, 192.168.0.75

Communication Method
● CIP ○ DH+ Channel: Destination Link:
○ CIP With Source ID Source Link: Destination Node: 0 (Octal)
□ Cache Connections

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0
○ Error Code: □ Timed Out
Extended Error Code:

Verify that the **CIP** radio-button is selected as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown, the first element is "Enet", which is the user-defined name given to the 1756-ENET gateway in the chassis (the slot number of the ENET gateway can be substituted for the name), the second element, "2", represents the Ethernet port on the 1756-ENET gateway, and the last element of the path, "192.168.0.75", is the IP address of the gateway, the target for the message.

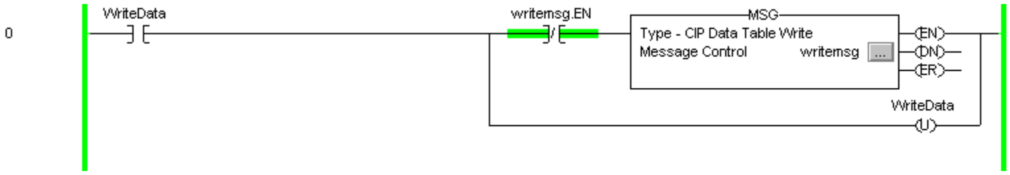
More complex paths are possible if routing to other networks using multiple 1756-ENET gateways and racks. Refer to the Support Knowledgebase for more information on Ethernet routing and path definitions.

CIP Data Table Operations

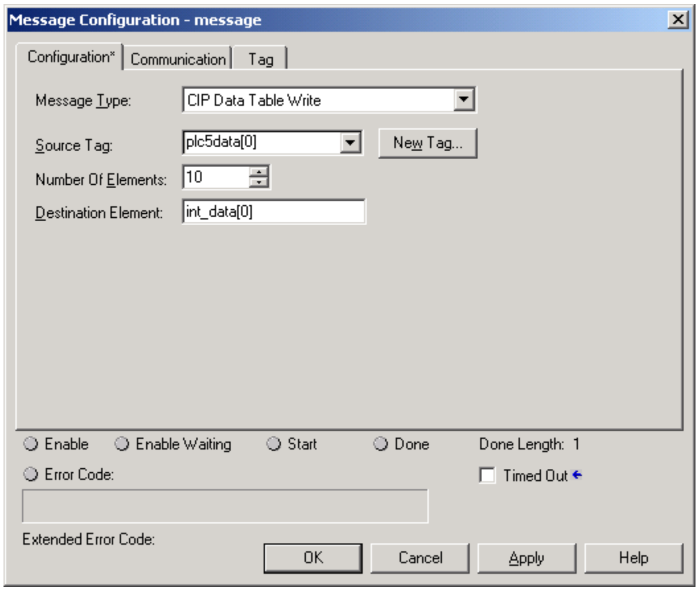
This method of data transfer uses CIP messages to transfer data between the ControlLogix or CompactLogix processor and the gateway. Tag names define the elements to be transferred. The following topics describe the support for the read and write operations.

CIP Data Table Write

CIP data table write messages transfer data from the processor to the gateway. An example rung used to execute a write command is shown in the following diagram:



The **MESSAGE CONFIGURATION** dialog box must be completed to define the data set to be transferred from the processor to the gateway. An example of the dialog box follows:



Message Type: CIP Data Table Write

Source Tag: plc5data[0]

Number Of Elements: 10

Destination Element: int_data[0]

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 1

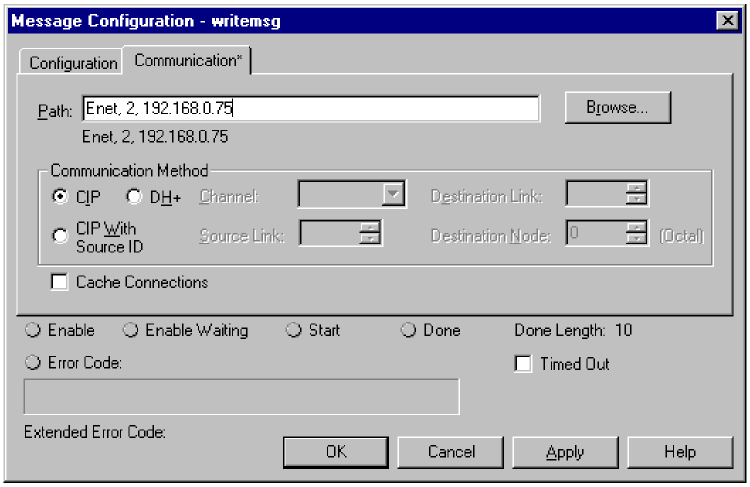
○ Error Code: □ Timed Out

Extended Error Code:

Complete the dialog box for the data area to be transferred. CIP Data Table messages require a tag database element for both the source and destination. The **SOURCE TAG** is a tag defined in the Controller Tag database. The **DESTINATION ELEMENT** is the tag element in the gateway.

The gateway simulates a tag database as an array of elements defined by the maximum register size for the gateway (user configuration parameter "Maximum Register" in the [Gateway] section) with the tag name **INT_DATA.**

In the previous example, the first element in the database is the starting location for the write operation of ten elements. The **COMMUNICATION** information must also be configured. An example of the dialog box follows:



Message Configuration - writemsg

Configuration | Communication*

Path: Enet, 2, 192.168.0.75
Enet, 2, 192.168.0.75

Communication Method
● CIP ○ DH+ Channel: Destination Link:
○ CIP With Source ID Source Link: Destination Node: 0 (Octal)

☐ Cache Connections

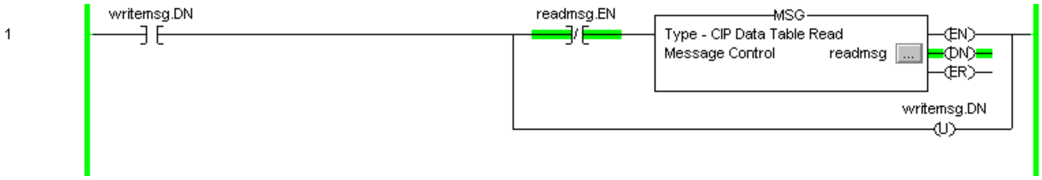
○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 10
○ Error Code: ☐ Timed Out
Extended Error Code:

Verify that the **CIP** radio-button is selected as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown, the first element is "Enet", which is the user-defined name given to the 1756-ENET gateway in the chassis (the slot number of the ENET gateway can be substituted for the name), the second element, "2", represents the Ethernet port on the 1756-ENET gateway, and the last element of the path, "192.168.0.75", is the IP address of the gateway, the target for the message.

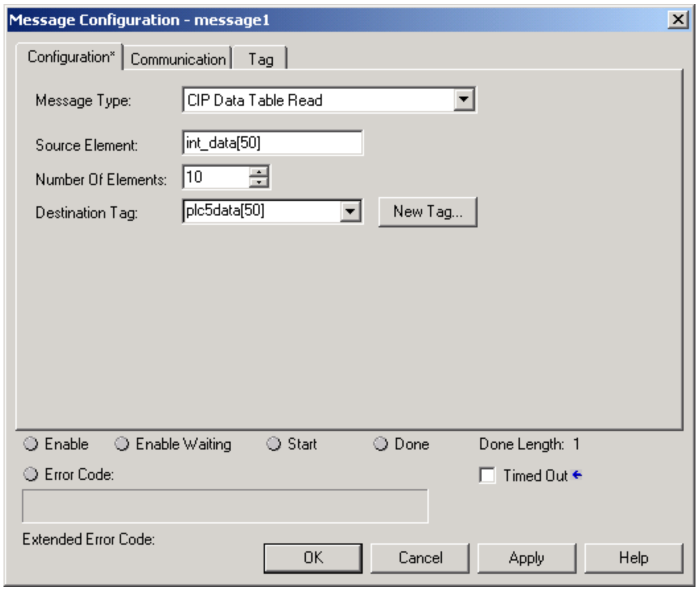
More complex paths are possible if routing to other networks using multiple 1756-ENET gateways and racks. Refer to the Support Knowledgebase for more information on Ethernet routing and path definitions.

CIP Data Table Read

CIP data table read messages transfer data to the processor from the gateway. An example rung used to execute a read command is shown:



The **MESSAGE CONFIGURATION** dialog box must be completed to define the data set to transfer to the processor from the gateway. An example of the dialog box follows:


Message Configuration - message1

Message Type: CIP Data Table Read

Source Element: int_data[50]

Number Of Elements: 10

Destination Tag: plc5data[50]

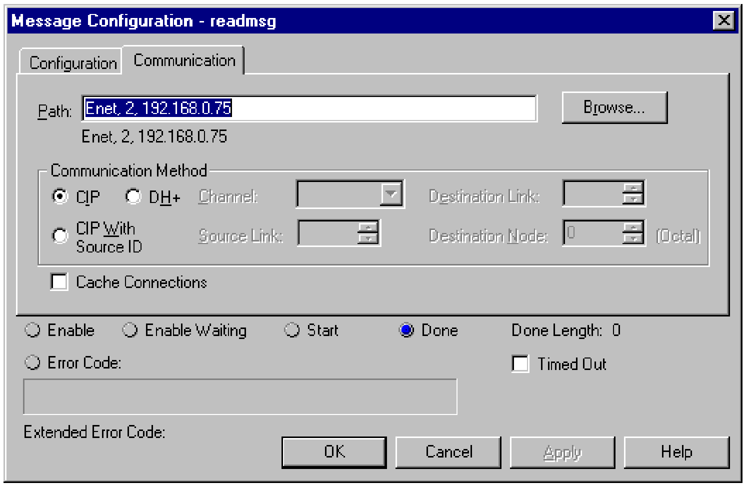
○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 1

○ Error Code:

□ Timed Out

Extended Error Code:

Complete the dialog box for the data area to be transferred. CIP Data Table messages require a tag database element for both the source and destination. The **DESTINATION TAG** is a tag defined in the Controller Tag database. The **SOURCE ELEMENT** is the tag element in the EIP gateway. The gateway simulates a tag database as an array of elements defined by the maximum register size for the gateway (user configuration parameter "Maximum Register" in the [Gateway] section) with the tag name **INT_DATA.** In the example above, the first element in the database is the starting location for the read operation of ten elements. Additionally, the **COMMUNICATION** information must also be configured. An example of the dialog box follows:


Message Configuration - readmsg

Path: Enet, 2, 192.168.0.75

Enet, 2, 192.168.0.75

Communication Method

● CIP ○ DH+ Channel: Destination Link:

○ CIP With Source ID Source Link: Destination Node: 0 (Octal)

□ Cache Connections

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

○ Error Code:

□ Timed Out

Extended Error Code:

Verify that the **CIP** radio-button is selected as the **COMMUNICATION METHOD**. The **PATH** specifies the message route from the processor to the EIP gateway. Path elements are separated by commas. In the example path shown, the first element is "Enet", which is the user-defined name given to the 1756-ENET gateway in the chassis (the slot number of the ENET gateway can be substituted for the name), the second element, "2", represents the Ethernet port on the 1756-ENET gateway, and the last element of the path, "192.168.0.75", is the IP address of the gateway, the target for the message.

More complex paths are possible if routing to other networks using multiple 1756-ENET gateways and racks. Refer to the Support Knowledgebase for more information on Ethernet routing and path definitions.

5.4.4 EIP Command Entry Form

The following form can be used to design the application's command list:

Gateway Information Data						Device Information Data						
Column #	1	2	3	4	5	6	7	8	9	10	11	12
Function Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	IP Address	Slot Number	Function Code	Function Parameters			

IP Address = IP address of processor to reach
Slot Number = -1 for PLC5 & SLC, processor slot number of ControlLogix

6 MBTCP Protocol

In This Chapter

This chapter contains information specific to the PLX3x-series gateway Modbus TCP/IP (MBTCP) protocol driver.

6.1 MBTCP Functional Overview

The PLX3x Modbus TCP/IP (MBTCP) protocol can be used to interface many different protocols into the Schneider Electric Quantum family of processors as well other devices supporting the protocol. The MBTCP protocol supports both Client and server connections.

The gateway supports a Client connection on the TCP/IP network to interface with processors (and other server based devices) using a user constructed command list of up to 100 entries. The gateway's internal database is used as the source for write commands to the remote processors. Data collected from the processors using read commands is placed in the gateway's database.

Data in the gateway's internal database is accessible for read and write operations by any node on the network supporting the MBAP (Service Port 502) or MBTCP (Service Ports 2000/2001) TCP/IP protocols. The MBAP protocol (Port 502) is a standard implementation defined by Schneider Electric and used on their Quantum processor. This open protocol is a modified version of the Modbus serial protocol. The MBTCP protocol is an embedded Modbus protocol message in a TCP/IP packet. The gateway supports up to five active server connections on Service Ports 502, five additional active server connections on Service Port 2000, and one active Client connection.

6.1.1 General Specifications - Modbus TCP/IP

The Modbus TCP/IP protocol allows multiple independent, concurrent Ethernet connections. The connections may be all Clients, all servers, or a combination of both Client and server connections.

- 10/100 MB Ethernet Communication port
- Supports Enron version of Modbus protocol for floating-point data transactions
- Configurable parameters for the client including a minimum response delay of 0 to 65535 ms and floating-point support
- Supports five independent server connections for Service Port 502
- Supports five independent server connections for Service Port 2000
- All data mapping begins at Modbus register 400001, protocol base 0.
- Error codes, network error counters, and port status data available in user data memory

Modbus TCP/IP Client

- Actively reads data from and writes data to Modbus TCP/IP devices using MBAP
- Up to 10 Client connections with multiple commands to talk to multiple servers

Modbus TCP/IP Server

- The server driver accepts incoming connections on Service Port 502 for Clients using Modbus TCP/IP MBAP messages and connections on Service Port 2000 (or other Service Ports) for Clients using Encapsulated Modbus messages.
- Supports multiple independent server connections for any combination of Service Port 502 (MBAP) and Service Port 2000 (Encapsulated)
- Up to 20 servers are supported

Modbus Commands Supported (Client and Server)	1: Read Coil Status 2: Read Input Status 3: Read Holding Registers 4: Read Input Registers 5: Force (Write) Single Coil 6: Preset (Write) Single Holding Register	15: Force (Write) Multiple Coils 16: Preset (Write) Multiple Holding Registers 22: Mask Write Holding Register (Slave Only) 23: Read/Write Holding Registers (Slave Only)
Configurable Parameters: (Client and Server)	Gateway IP Address PLC Read Start Register (%MW) PLC Write Start Register (%MW) Number of MBAP and MBTCP servers Gateway Modbus Read Start Address Gateway Modbus Write Start Address	
Configurable Parameters: (Client Only)	Minimum Command Delay Response Timeout Retry Count Command Error Pointer	
Command List	Up to 160 Modbus commands (one tag per command)	
Status Data	Error codes reported individually for each command. High-level status data available from Modbus TCP/IP Client (for example PLC)	
Command List Polling	Each command can be individually enabled or disabled; write-only-on-data-change is available	

6.1.2 Internal Database

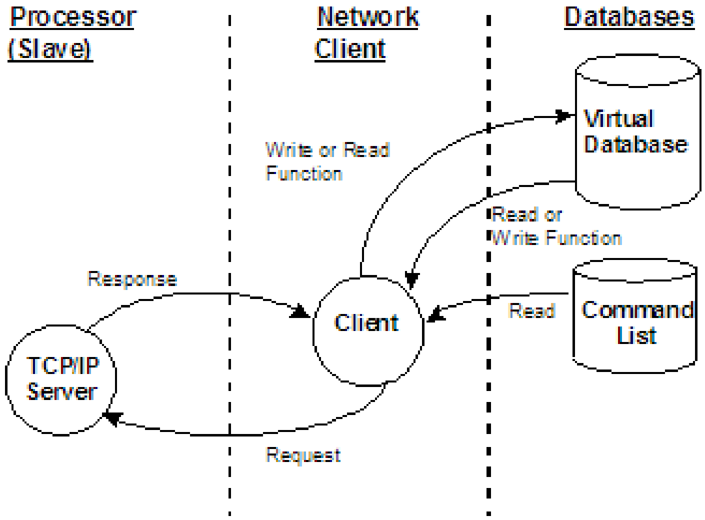
Central to the functionality of the gateway is the internal database. This database is shared between all the ports on the gateway and is used as a conduit to pass information from one device on one network to one or more devices on another network. This permits data from devices on one communication port to be viewed and controlled by devices on another communication port.

In addition to data from the Client and server, status and error information generated by the gateway can also be mapped into the internal database.

Modbus TCP/IP Client Access to Database

The Client functionality exchanges data between the PLX3x gateway's internal database and data tables established in one or more Quantum processors or other server based devices. The command list, defined in the user configuration, specifies what data is to be transferred between the gateway and each of the servers on the network. No ladder logic is required in the processor (server) for Client functionality, except to assure that sufficient data memory exists.

The following illustration describes the flow of data between the Ethernet Clients and the internal database.



Multiple Server Access to Database

The MBTCP gateway provides server functionality using reserved Service Port 502 for Modbus TCP/IP MBAP messages, as well as Service Ports 2000 and 2001 to support the TCP/IP Encapsulated Modbus version of the protocol used by several HMI manufacturers. Server support in the gateway permits Client applications (for example: HMI software, Quantum processors, etc) to read from and write to the gateway’s database. This section discusses the requirements for attaching to the gateway using Client applications.

The server driver is able to support multiple concurrent connections from several Clients. Up to five (5) Clients can simultaneously connect on Service Port 502 and five (5) more can also simultaneously connect on Service Port 2000. Service Port 2001 is used by the MBTCP protocol to pass Encapsulated Modbus commands through from the Ethernet port to the gateway’s serial port.

When configured as a server, the internal database of the MBTCP gateway is used as the source for read requests and the destination for write requests from remote Clients. Access to the database is controlled by the command type received in the incoming message from the Client. The following table specifies the relationship of the gateway’s internal database to the addresses required in the incoming Modbus TCP/IP requests.

Database Address	Modbus Address
0	40001
1000	41001
2000	42001
3000	43001
3999	44000

The following virtual addresses are not part of the normal gateway user database and are not valid addresses for standard data. However, these addresses may be used for incoming commands that are requesting floating-point data.

To use addresses in this upper range requires the following

- Set the *Float Flag* in the MBTCP server configuration to **Yes**
- Set the *Float Start* to a database address in the range below
- Set the *Float Offset* to a database address in the gateway user memory area shown above.

Remember that, once this is done, all data above the *Float Start* address must be floating-point data.

Database Address	Modbus Address
4000	44001
5000	45001
6000	46001
7000	47001
8000	48001
9000	49001
9999	50000

The MBTCP gateway must be correctly configured and connected to the network before any attempt is made to use it. Use a network verification program, such as *ProSoft Discovery Service* or the command prompt PING instruction, to verify that the gateway can be seen on the network. Use *ProSoft Configuration Builder to* confirm proper configuration of the gateway and to transfer the configuration files to and from the gateway.

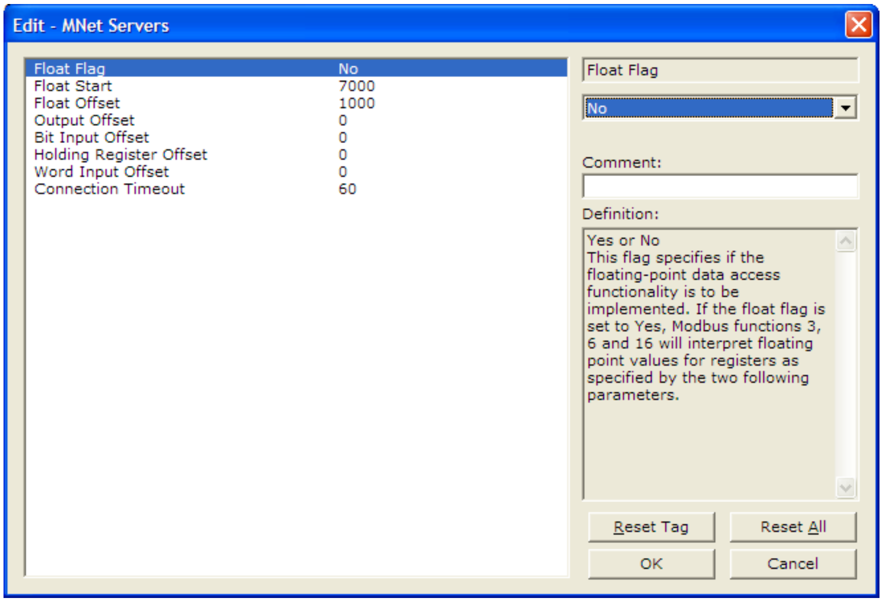
Modbus Message Routing: Port 2001

When Modbus messages are sent to the Gateway over the TCP/IP connection to port 2001, the messages are sent (routed in the Gateway) directly out the serial communication port (Port 0, if it is configured as a Modbus Master. The commands (whether a read or a write command) are immediately routed to the slave devices on the serial port. Response messages from the slave devices are routed to the TCP/IP network to be received by the originating host.

6.2 MBTCP Configuration

6.2.1 MBTCP Servers

This section contains database offset information used by the server when accessed by external Clients. These offsets can be utilized to segment the database by data type.



Float Flag	No
Float Start	7000
Float Offset	1000
Output Offset	0
Bit Input Offset	0
Holding Register Offset	0
Word Input Offset	0
Connection Timeout	60

Float Flag

No

Comment:

Definition:

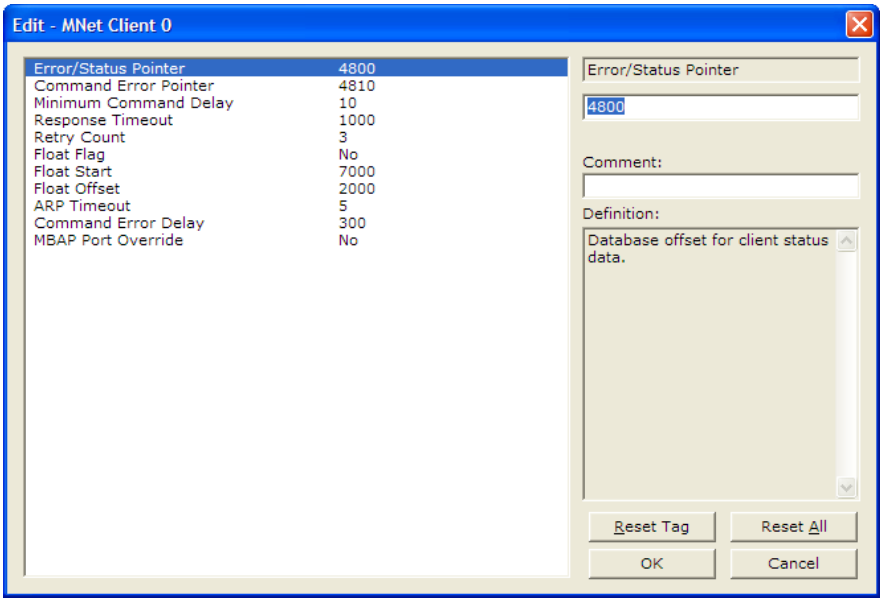
Yes or No
This flag specifies if the floating-point data access functionality is to be implemented. If the float flag is set to Yes, Modbus functions 3, 6 and 16 will interpret floating point values for registers as specified by the two following parameters.

Parameter	Value	Description
Float Flag	Yes or No	This flag specifies if the floating-point data access functionality is to be implemented. If the float flag is set to **Yes**, Modbus functions 3, 6, and 16 will interpret floating-point values for registers as specified by the two following parameters.
Float Start	0 to 65535	This parameter specifies the first register of floating-point data. All requests with register values greater than or equal to this value will be considered floating-point data requests. This parameter is only used if the Float Flag is enabled. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered floating-point data.
Float Offset	0 to 3999	This parameter specifies the start register for floating-point data in the internal database. This parameter is used only if the Float Flag is enabled. For example, if the Float Offset value is set to 3000 and the float start parameter is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.
Output Offset	0 to 3999	When the port is configured as a slave, this parameter specifies the internal database address to use as the zero address or starting point for binary output Coil data. Coil data is read by Modbus Function Code 1 commands (Read Coils) and written by Function Codes 5 (Force Single Coil) or Function Code 15 (Force Multiple

Parameter	Value	Description
		Coils). For example, if this parameter is set to 50 and a Function Code 1 command is received requesting Coil address 0 (virtual Modbus Coil address 00001 or 000001), the data returned in the response will be the value at register 50, bit 0 in the gateway's database.
Bit Input Offset	0 to 3999	This parameter specifies the offset address in the internal Modbus database for network requests for Modbus function 2 commands. For example, if the value is set to 150, an address request of 0 will return the value at register 150 in the database.
Holding Register Offset	0 to 3999	This parameter specifies the offset address in the internal Modbus database to with network requests for Modbus functions 3, 6, or 16 commands. For example, if the value is set to 50, an address request of 0 will return the value at register 50 in the database.
Word Input Offset	0 to 3999	This parameter specifies the offset address in the internal Modbus database for network requests for Modbus function 4 commands. For example, if the value is set to 150, an address request of 0 will return the value at register 150 in the database.
Connection Timeout	0 to 1200	This parameter specifies the number of seconds the server will wait to receive new data. If the server does not receive any new data during this time, it will close the connection.

6.2.2 MBTCP Client[x]

The *MBTCPClient [x]* section of the configuration specifies the parameters for the client to be emulated on the gateway. The command list for the client is entered in a separate section.



Error/Status Pointer	4800
Command Error Pointer	4810
Minimum Command Delay	10
Response Timeout	1000
Retry Count	3
Float Flag	No
Float Start	7000
Float Offset	2000
ARP Timeout	5
Command Error Delay	300
MBAP Port Override	No

Error/Status Pointer

4800

Comment:

Definition:

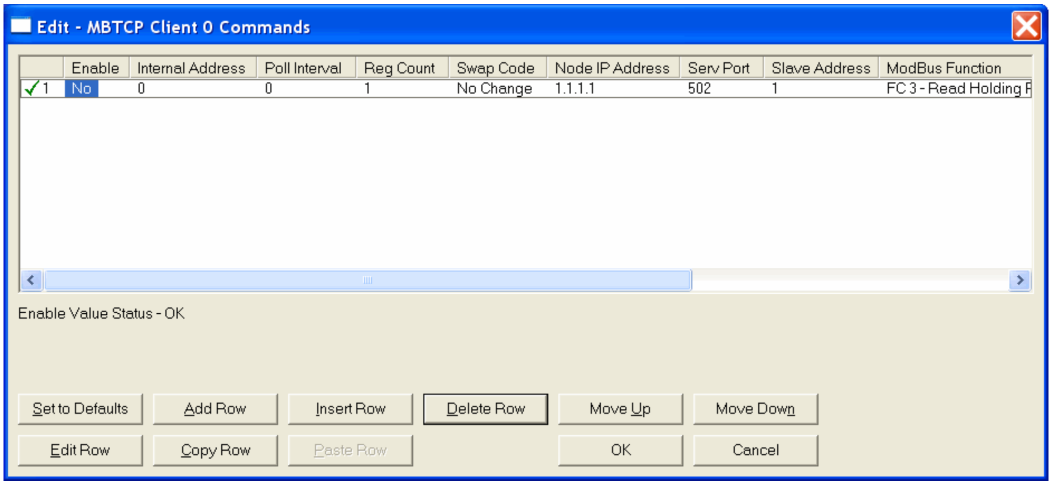
Database offset for client status data.

Parameter	Value	Description
Minimum Command Delay	0 to 32767	This parameter specifies the number of milliseconds to wait between the initial issuance of a command. It can be used to delay all commands sent to slaves to avoid "flooding" commands on the network. It does not affect retries of a command as they will be issued when failure is recognized.
Response Timeout	0 to 65535	This parameter specifies the time in milliseconds that a Client will wait before re-transmitting a command if no response is received from the addressed server. The value to use depends on the type of communication network used, and the expected response time of the slowest device on the network.
Retry Count	0 to 10	This parameter specifies the number of times a command will be retried if it fails.
Float Flag	Yes or No	This flag specifies if the floating-point data access functionality is to be implemented. If the float flag is set to **Yes**, Modbus functions 3, 6, and 16 will interpret floating-point values for registers as specified by the two following parameters.
Float Start	0 to 32767	This parameter specifies the first register of floating-point data. All requests with register values greater-than or equal to this value will be considered floating-point data requests. This parameter is only used if the Float Flag is enabled. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered as floating-point data.
Float Offset	0 to 3998	This parameter specifies the starting register for floating-point data in the internal gateway database. This parameter is used

Parameter	Value	Description
		only if the *Float Flag* is set to **Yes**. For example, if the *Float Offset* value is set to 3000 and the *Float Start* parameter is set to 7000, the data returned as floating-point data for register 47001 (or 407001) will actually come from internal gateway registers 3000 and 3001. If the requested address was 47002 (407002), the data will be returned from internal registers 3002 and 3003. If the requested address was 47101 (407101), the data will be returned from internal registers 3200 and 3201; and so on.
ARP Timeout	1 to 60	This parameter specifies the number of seconds to wait for an ARP reply after a request is issued.
Command Error Delay	0 to 300	This parameter specifies the number of 100 millisecond intervals to turn off a command in the error list after an error is recognized for the command. If this parameter is set to **0**, there will be no delay.

6.2.3 MBTCP Client[x] Commands

The *MBTCP Client[x] Commands* section defines the Modbus TCP/IP commands to be issued from the gateway to server devices on the network. These commands can be used for data collection and/or control of devices on the TCP/IP network.



	Enable	Internal Address	Poll Interval	Reg Count	Swap Code	Node IP Address	Serv Port	Slave Address	ModBus Function
✓ 1	No	0	0	1	No Change	1.1.1.1	502	1	FC 3 - Read Holding F

Enable Value Status - OK

In order to interface the PLX3x gateway with Modbus TCP/IP server devices, a command list must be constructed. The commands in the list specify the server device to be addressed, the function to be performed (read or write), the data area in the device to interface with and the registers in the internal database to be associated with the device data. The Client command list supports up to 16 commands per Client. The command list is processed from top (command #0) to bottom.

The following table describes the command list configuration parameters.

Parameter	Value	Description
Enable	YES NO CONDITONAL	Specifies if the command is to be executed and under what conditions. **No (0)** - the command is disabled and will not be executed in the normal polling sequence. **Yes (1)**- the command will be executed upon each scan of the Command List if the Poll Interval is set to zero (0). If the *Poll Interval* is set to a non-zero value, the command will be executed when the interval timer for that command expires. **Conditional (2)**- the command will execute only if the internal bit data associated with the command changes. It will also clear the bit or bits in the internal database after the write command is built. This parameter is valid only for bit-level write commands (FC 5 and 15).
Internal Address	0 to 3999 (for register-level	Specifies the database address in the gateway's internal database to use as the destination for data brought in by a read

Parameter	Value	Description
	addressing) or 0 to 63999(for bit-level addressing)	command, or as the source for data to be sent out by a write command. The database address is interpreted as a bit address or a 16-bit register (word) address, depending on the Modbus Function Code used in the command. For Modbus functions 1, 2, 5, and 15, this parameter is interpreted as a bit-level address. For Modbus functions 3, 4, 6, and 16, this parameter is interpreted as a register-level address.
Poll Interval	0 to 65535	Specifies the minimum interval between executions of continuous commands. The value is in tenths of a second. Therefore, if a value of 100 is entered, the command will execute no more frequently than once every 10 seconds.
Reg Count	1 to 125 (for registers) or 1 to 800 (for coils)	This parameter specifies the number of 16-bit registers or binary bits to be transferred by the command. Modbus functions 5 and 6 ignore this field as they apply only to a single data point. For Modbus functions 1, 2, and 15, this parameter sets the number of bits (inputs or coils) to be transferred by the command. For Modbus functions 3, 4, and 16, this parameter sets the number of registers to be transferred by the command.
Swap Code	No Change Word Swap Word and Byte Swap Byte Swap	Specifies if and how the order of bytes in data received or sent is to be rearranged. This option exists to allow for the fact that different manufacturers store and transmit multi-byte data in different combinations. This parameter is helpful when dealing with floating-point or other multi-byte values, as there is no one standard method of storing these data types. The parameter can be set to rearrange the byte order of data received or sent into an order more useful or convenient for other applications. **No change(0)** - No change is made in the byte ordering (1234 = 1234) **Word Swap (1)** -The words are swapped (1234=3412) **Word and Byte Swap (2)** - The words are swapped, then the bytes in each word are swapped (1234=4321) **Byte Swap (3)** - The bytes in each word are swapped (1234=2143) These swap operations affect 4-byte (or 2-word) groups of data. Therefore, data swapping using these *Swap Codes* should be done only when using an even number of words, such as when 32-bit integer or floating-point data is involved.
Node IP Address	xxx.xxx.xxx.xxx	IP address of the device being addressed by the command.
Serv Port	502 or other supported port on server	Service port on which communication will occur. Use a value of 502 when addressing Modbus TCP/IP servers which are compatible with the Schneider Electric MBAP specifications (this will be most devices). If the server device supports another service port, enter the value here.
Slave Address	1 to 255 (0 is a broadcast)	Specifies the node address of a remote Modbus Serial device through a Modbus Ethernet to Serial converter. Values of 1 to 255 are permitted. **Note:** Most Modbus devices only accept addresses in the range of 1 to 247, so check with the slave device manufacturer to see if

Parameter	Value	Description
		a particular slave can use addresses 248 to 255. If the value is set to zero, the command will be a broadcast message on the network. The Modbus protocol permits broadcast commands for *write* operations. **Do not** use node address 0 for *read* operations.
Modbus Function	1, 2, 3, 4, 5, 6, 15, or 16	Specifies the Modbus Function Code to be executed by the command. These function codes are defined in the Modbus protocol. (More information on the protocol is available from www.modbus.org.) The following function codes are supported by the gateway. **1** - Read Coil Status **2** - Read Input Status **3** - Read Holding Registers **4** - Read Input Registers **5** - Force (Write) Single Coil **6** - Preset (Write) Single Register **15** - Force Multiple Coils **16** - Preset Multiple Registers
MB Address in Device	Varies	Specifies the starting Modbus register or bit address in the server to be used by the command. Refer to the documentation of each Modbus server device for the register and bit address assignments valid for that device. The Modbus Function Code determines whether the address will be a register-level or bit-level OFFSET address into a given data type range. The offset will be the target data address in the server minus the base address for that data type. Base addresses for the different data types are: ♦ 00001 or 000001 (0x0001) for bit-level Coil data (Function Codes 1, 5, and 15). ♦ 10001 or 100001 (1x0001) for bit-level Input Status data (Function Code 2) ♦ 30001 or 300001 (3x0001) for Input Register data (Function Code 4) ♦ 40001 or 400001 (4x0001) for Holding Register data (Function Codes 3, 6, and 16). ♦ Address calculation examples: o For bit-level Coil commands (FC 1, 5, or 15) to read or write a Coil 0X address 00001, specify a value of 0 (00001 - 00001 = 0). o For Coil address 00115, specify 114 o (00115 - 00001 = 114) ♦ For register read or write commands (FC 3, 6, or 16) 4X range, for 40001, specify a value of 0 o (40001 - 40001 = 0). ♦ For 01101, 11101, 31101 or 41101, specify a value of 1100. o (01101 - 00001 = 1100) o (11101 -10001 = 1100) o (31101 - 30001 = 1100) o (41101 - 40001 = 1100)

Parameter	Value	Description
		Note: If the documentation for a particular Modbus server device lists data addresses in hexadecimal (base16) notation, you will need to convert the hexadecimal value to a decimal value to enter in this parameter. In such cases, it is not usually necessary to subtract 1 from the converted decimal number, as this addressing scheme typically uses the exact offset address expressed as a hexadecimal number.

6.3 MBTCP Diagnostics

6.3.1 PCB Diagnostics

The best way to troubleshoot the MBTCP driver is to use ProSoft Configuration Builder to access the diagnostic capabilities of the gateway through the Ethernet debug port. For instructions on how to access *Diagnostics*, see *Using Diagnostics in ProSoft Configuration Builder* (page 36).

6.3.2 MBTCP Status Data in Upper Memory

The MBTCP driver has an associated status data area located in the gateway's upper memory. The Data Map functionality of the gateway can be used to map this data into the normal user data range of the gateway's database. See *Using the CommonNet Data Map* (page 25).

Note that all the status values are initialized to zero (0) at power-up, cold boot and during warm boot.

MBTCP Server Status Data

The following table lists the addresses in upper memory where status data for MBTCP servers can be found.

Server Port	**Address Range**
2000	6200 through 6209
502	6210 through 6219
2001	6220 through 6229

The content of each server port's status data area is structured the same. The following table describes the content of each register in the status data area.

Offset	**Description**
0	Number of Command Requests
1	Number of Command Responses
2	Number of Command Errors
3	Number of Requests
4	Number of Responses
5	Number of Errors Sent
6	Number of Errors Received
7	Configuration Error Word
8	Current Error Code
9	Last Error Code

MBTCP Client Status Data

The following table lists the addresses in upper memory that hold status data for each MBTCP Client.

Client	Address Range
0	15500 through 15509
1	15526 through 15535
2	15552 through 15561
.	.
.	.
8	15708 through 15717
9	15734 through 15743

The content of each Client's status data area is structured the same. The following table describes the content of each register in the status data area.

Offset	Description
0	Command Request Count (total Client commands sent)
1	Command Response Count (total command responses received)
2	Command Error Count
3	Number of Request Packets
4	Number of Response Packets
5	Errors Sent
6	Errors Received
7	Reserved
8	Current Error
9	Last Error

Offsets 8 and 9 contain information about the most recent communication errors.

The Current Error (offset 8) will have a non-zero value if the currently executing Client command experiences an error.

The Last Error (offset 9) will store the most recent non-zero value error code that was reported by the Client the last time it experienced an error. Note that this value is retentive. This register will hold the last error value until the memory is cleared by a restart, reset, cold-boot, or warm-boot operation. Therefore, any value you see here may indicate an error that could have occurred at any time since the gateway was last restarted and may not indicate a current or recent error. For details on error codes, see MBTCP Client Command List Error Data (page 109).

MBTCP Client Command List Error Data

A status/error code is held in upper memory for each command in each MBTCP Client's command list.

The following table lists the addresses in upper memory that hold command list error data for each MBTCP Client.

Client	Address Range
0	15510 through 15525
1	15536 through 15551
2	15562 through 15577
.	.
.	.
8	15718 through 15733
9	15744 through 15759

The first word in each Client's command list error data area contains the status/error code for the first command in the Client's Command List. Each successive word in the Command Error List is associated with the next command in the Client Command List. Therefore, the number of valid error values is dependent upon the number of commands defined.

The structure of the command list error data area (which is the same for all Clients) is displayed in the following table.

Offset	Description
0	Command #1 Error Code
1	Command #2 Error Code
2	Command #3 Error Code
3	Command #4 Error Code
4	Command #5 Error Code
.	.
.	.
13	Command #14 Error Code
14	Command #15 Error Code
15	Command #16 Error Code

A non-zero error code for a command indicates an error. To interpret the status/error codes, refer to MBTCP Error Codes (page 109).

6.3.3 MBTCP Error Codes

Standard Modbus Exception Code Errors

Code	Description
1	Illegal function
2	Illegal data address
3	Illegal data value
4	Failure in associated device
5	Acknowledge
6	Busy; message was rejected

MBTCP Client Specific Errors

Code	Description
-33	Failed to connect to server specified in command
-35	Wrong message length in the response
-36	MBTCP command response timeout (same as -11)
-37	TCP/IP connection ended before session finished

Command List Entry Errors

Code	Description
-40	Too few parameters
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (<0 or >255)
-44	Count parameter set to 0
-45	Invalid function code
-46	Invalid swap code

Gateway Communication Error Codes

Code	Description
-2	Timeout while transmitting message
-11	Timeout waiting for response after request (same as -36)
253	Incorrect slave/server address in response
254	Incorrect function code in response
255	Invalid CRC/LRC value in response

6.4 MBTCP Reference

6.4.1 Modbus Protocol Specification

Read Coil Status (Function Code 01)

Query

This function allows the user to obtain the ON/OFF status of logic coils used to control discrete outputs from the addressed server only. Broadcast mode is not supported with this function code. In addition to the server address and function fields, the message requires that the information field contain the initial coil address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 coils to be obtained at each request; however, the specific server device may have restrictions that lower the maximum quantity. The coils are numbered from zero; (coil number 1 = zero, coil number 2 = one, coil number 3 = two, and so on).

The following table is a sample read output status request to read coils 0020 to 0056 from server device number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data # Of Pts Ho	Data # Of Pts Lo	Error Check Field
11	01	00	13	00	25	CRC

Response

An example response to Read Coil Status is as shown in Figure C2. The data is packed one bit for each coil. The response includes the server address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each coil (1 = ON, 0 = OFF). The low order bit of the first character contains the addressed coil, and the remainder follows. For coil quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the server interface device is serviced at the end of a controller's scan, data will reflect coil status at the end of the scan. Some servers will limit the quantity of coils provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status from sequential scans.

Adr	Func	Byte Count	Data Coil Status 20 to 27	Data Coil Status 28 to 35	Data Coil Status 36 to 43	Data Coil Status 44 to 51	Data Coil Status 52 to 56	Error Check Field
11	01	05	CD	6B	B2	OE	1B	CRC

The status of coils 20 to 27 is shown as CD(HEX) = 1100 1101 (Binary). Reading left to right, this shows that coils 27, 26, 23, 22, and 20 are all on. The other coil data bytes are decoded similarly. Due to the quantity of coil statuses requested, the last data field, which is shown 1B (HEX) = 0001 1011 (Binary), contains the status of only 5 coils (52 to 56) instead of 8 coils. The 3 left most bits are provided as zeros to fill the 8-bit format.

Read Input Status (Function Code 02)

Query

This function allows the user to obtain the ON/OFF status of discrete inputs in the addressed server PC Broadcast mode is not supported with this function code. In addition to the server address and function fields, the message requires that the information field contain the initial input address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 inputs to be obtained at each request; however, the specific server device may have restrictions that lower the maximum quantity. The inputs are numbered form zero; (input 10001 = zero, input 10002 = one, input 10003 = two, and so on, for a 584).

The following table is a sample read input status request to read inputs 10197 to 10218 from server number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data #of Pts Hi	Data #of Pts Lo	Error Check Field
11	02	00	C4	00	16	CRC

Response

An example response to Read Input Status is as shown in Figure C4. The data is packed one bit for each input. The response includes the server address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each input (1=ON, 0=OFF). The lower order bit of the first character contains the addressed input, and the remainder follows. For input quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as a quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the server interface device is serviced at the end of a controller's scan, data will reflect input status at the end of the scan. Some servers will limit the quantity of inputs provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status for sequential scans.

Adr	Func	Byte Count	Data Discrete Input 10197 to 10204	Data Discrete Input 10205 to 10212	Data Discrete Input 10213 to 10218	Error Check Field
11	02	03	AC	DB	35	CRC

The status of inputs 10197 to 10204 is shown as AC (HEX) = 10101 1100 (binary). Reading left to right, this show that inputs 10204, 10202, and 10199 are all on. The other input data bytes are decoded similar.

Due to the quantity of input statuses requested, the last data field which is shown as 35 HEX = 0011 0101 (binary) contains the status of only 6 inputs (10213 to 102180) instead of 8 inputs. The two left-most bits are provided as zeros to fill the 8-bit format.

Read Holding Registers (Function Code 03)

Query

Read Holding Registers (03) allows the user to obtain the binary contents of holding registers 4xxxx in the addressed server. The registers can store the numerical values of associated timers and counters which can be driven to external devices. The addressing allows up to 125 registers to be obtained at each request; however, the specific server device may have restrictions that lower this maximum quantity. The registers are numbered form zero (40001 = zero, 40002 = one, and so on). The broadcast mode is not allowed.

The example below reads registers 40108 through 40110 from server 584 number 11.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	03	00	6B	00	03	CRC

Response

The addressed server responds with its address and the function code, followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are two bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the server interface device is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Some servers will limit the quantity of register content provided each scan; thus for large register quantities, multiple transmissions will be made using register content from sequential scans.

In the example below, the registers 40108 to 40110 have the decimal contents 555, 0, and 100 respectively.

Adr	Func	ByteCnt	Hi Data	Lo Data	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field
11	03	06	02	2B	00	00	00	64	CRC

Read Input Registers (Function Code 04)

Query

Function code 04 obtains the contents of the controller's input registers at addresses 3xxxx. These locations receive their values from devices connected to the I/O structure and can only be referenced, not altered from within the controller, The addressing allows up to 125 registers to be obtained at each request; however, the specific server device may have restrictions that lower this maximum quantity. The registers are numbered for zero (30001 = zero, 30002 = one, and so on). Broadcast mode is not allowed.

The example below requests the contents of register 3009 in server number 11.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	04	00	08	00	01	CRC

Response

The addressed server responds with its address and the function code followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are 2 bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the server interface is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Each PC will limit the quantity of register contents provided each scan; thus for large register quantities, multiple PC scans will be required, and the data provided will be form sequential scans.

In the example below the register 3009 contains the decimal value 0.

Adr	Func	Byte Count	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	04	02	00	00	E9

Force Single Coil (Function Code 05)

Query

This message forces a single coil either ON or OFF. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coil is disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 0001 = zero, coil 0002 = one, and so on). The data value 65,280 (FF00 HEX) will set the coil ON and the value zero will turn it OFF; all other values are illegal and will not affect that coil.

The use of server address 00 (Broadcast Mode) will force all attached servers to modify the desired coil.

Note: Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

The example below is a request to server number 11 to turn ON coil 0173.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/off Ind	Data	Error Check Field
11	05	00	AC	FF	00	CRC

Response

The normal response to the Command Request is to re-transmit the message as received after the coil state has been altered.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/ Off	Data	Error Check Field
11	05	00	AC	FF	00	CRC

The forcing of a coil via MODBUS function 5 will be accomplished regardless of whether the addressed coil is disabled or not (*In ProSoft products,* the coil *is only affected if the necessary ladder logic is implemented).*

Note: The Modbus protocol does not include standard functions for testing or changing the DISABLE state of discrete inputs or outputs. Where applicable, this may be accomplished via device specific Program commands (*In ProSoft products, this is only accomplished through ladder logic programming).*

Coils that are reprogrammed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function Code 5 and (even months later), an output is connected to that coil, the output will be "hot".

Preset Single Register (Function Code 06)

Query

Function (06) allows the user to modify the contents of a holding register. Any holding register that exists within the controller can have its contents changed by this message. However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller unused high order bits must be set to zero. When used with server address zero (Broadcast mode) all server controllers will load the specified register with the contents specified.

Note Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	06	00	01	00	03	CRC

Response

The response to a preset single register request is to re-transmit the query message after the register has been altered.

Adr	Func	Data Reg Hi	Data Reg Lo	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	06	00	01	00	03	CRC

Diagnostics (Function Code 08)

MODBUS function code 08 provides a series of tests for checking the communication system between a Client device and a server, or for checking various internal error conditions within a server.

The function uses a two-byte sub-function code field in the query to define the type of test to be performed. The server echoes both the function code and sub-function code in a normal response. Some of the diagnostics cause data to be returned from the remote device in the data field of a normal response.

In general, issuing a diagnostic function to a remote device does not affect the running of the user program in the remote device. Device memory bit and register data addresses are not accessed by the diagnostics. However, certain functions can optionally reset error counters in some remote devices.

A server device can, however, be forced into 'Listen Only Mode' in which it will monitor the messages on the communications system but not respond to them. This can affect the outcome of your application program if it depends upon any further exchange of data with the remote device. Generally, the mode is forced to remove a malfunctioning remote device from the communications system.

Sub-function Codes Supported

Only Sub-function 00 is supported by the gateway.

00 Return Query Data

The data passed in the request data field is to be returned (looped back) in the response. The entire response message should be identical to the request.

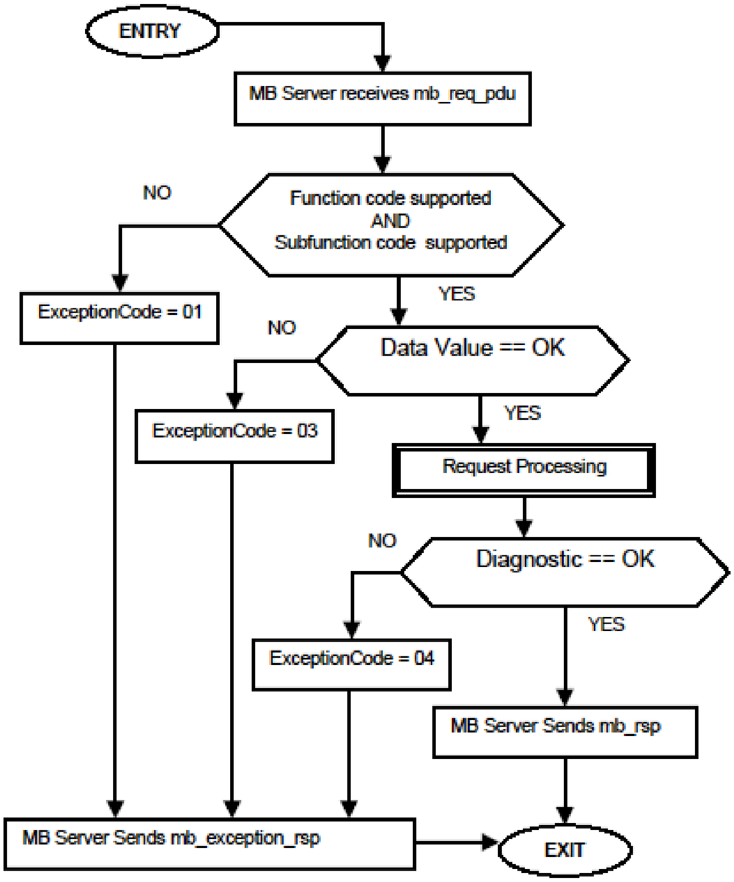
Sub-function	Data Field (Request)	Data Field (Response)
00 00	Any	Echo Request Data

Example and State Diagram

Here is an example of a request to remote device to Return Query Data. This uses a sub-function code of zero (00 00 hex in the two-byte field). The data to be returned is sent in the two-byte data field (A5 37 hex).

Request		Response	
Field Name	**(Hex)**	**Field Name**	**(Hex)**
Function	08	Function	08
Sub-function Hi	00	Sub-function Hi	00
Sub-function Lo	00	Sub-function Lo	00
Data Hi	A5	Data Hi	A5
Data Lo	37	Data Lo	27

The data fields in responses to other kinds of queries could contain error counts or other data requested by the sub-function code.



Modbus Exception Responses

When a Modbus Master sends a request to a server device, it expects a normal response. One of four possible events can occur from the Master's query:

- If the server device receives the request without a communication error, and can handle the query normally, it returns a normal response.
- If the server does not receive the request due to a communication error, no response is returned. The Master program will eventually process a timeout condition for the request.
- If the server receives the request, but detects a communication error (parity, LRC, CRC, etc...), no response is returned. The Master program will eventually process a timeout condition for the request.
- If the server receives the request without a communication error, but cannot handle it (for example, if the request is to read a non-existent output or register), the server will return an exception response informing the Master of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

Function Code Field: In a normal response, the server echoes the function code of the original request in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response.

With the function code's MSB set, the Master's application program can recognize the exception response and can examine the data field for the exception code.

Data Field: In a normal response, the server may return data or statistics in the data field (any information that was requested in the request). In an exception response, the server returns an exception code in the data field. This defines the server condition that caused the exception.

The following table shows an example of a Master request and server exception response.

Request		**Response**	
Field Name	**(Hex)**	**Field Name**	**(Hex)**
Function	01	Function	81
Starting Address Hi	04	Exception Code	02
Starting Address Lo	A1		
Quantity of Outputs Hi	00		
Quantity of Outputs Lo	01		

In this example, the Master addresses a request to server device. The function code (01) is for a Read Output Status operation. It requests the status of the output at address 1245 (04A1 hex). Note that only that one output is to be read, as specified by the number of outputs field (0001).

If the output address does not exist in the server device, the server will return the exception response with the exception code shown (02). This specifies an illegal data address for the server.

Modbus Exception Codes

Code	Name	Meaning
01	Illegal Function	The function code received in the query is not an allowable action for the server. This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	Illegal Data Address	The data address received in the query is not an allowable address for the server. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed; a request with offset 96 and length 5 will generate exception 02.
03	Illegal Data Value	A value contained in the query data field is not an allowable value for server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, because the Modbus protocol is unaware of the significance of any particular value of any particular register.
04	Slave Device Failure	An unrecoverable error occurred while the server was attempting to perform the requested action.
05	Acknowledge	Specialized use in conjunction with programming commands. The server has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the Master. The Master can next issue a poll program complete message to determine if processing is completed.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The server is engaged in processing a long-duration program command. The Master should retransmit the message later when the server is free.
08	Memory Parity Error	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server attempted to read record file, but detected a parity error in the memory. The Master can retry the request, but service may be required on the server device.
0a	Gateway Path Unavailable	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0b	Gateway Target Device Failed To Respond	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Force Multiple Coils (Function Code 15)

Query

This message forces each coil in a consecutive block of coils to a desired ON or OFF state. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coils are disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 00001 = zero, coil 00002 = one, and so on). The desired status of each coil is packed in the data field, one bit for each coil (1= ON, 0= OFF). The use of server address 0 (Broadcast Mode) will force all attached servers to modify the desired coils.

Note: Functions 5, 6, 15, and 16 are the only messages (other than Loopback Diagnostic Test) that will be recognized as valid for broadcast.

The following example forces 10 coils starting at address 20 (13 HEX). The two data fields, CD =1100 and 00 = 0000 000, indicate that coils 27, 26, 23, 22, and 20 are to be forced on.

Adr	Func	Hi Add	Lo Add	Quantity	Byte Cnt	Data Coil Status 20 to 27	Data Coil Status 28 to 29	Error Check Field	
11	0F	00	13	00	0A	02	CD	00	CRC

Response

The normal response will be an echo of the server address, function code, starting address, and quantity of coils forced.

Adr	Func	Hi Addr	Lo Addr	Quantity	Error Check Field	
11	0F	00	13	00	0A	CRC

The writing of coils via Modbus function 15 will be accomplished regardless of whether the addressed coils are disabled or not.

Coils that are not programmed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function code 15 and (even months later) an output is connected to that coil, the output will be hot.

Preset Multiple Registers (Function Code 16)

Query

Holding registers existing within the controller can have their contents changed by this message (a maximum of 60 registers). However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller (16-bit for the 184/384 and 584); unused high order bits must be set to zero.

⚠

Note: Function codes 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Hi Add	Lo Add	Quantity		Byte Cnt	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field
11	10	00	87	00	02	04	00	0A	01	02	CRC

Response

The normal response to a function 16 query is to echo the address, function code, starting address and number of registers to be loaded.

Adr	Func	Hi Addr	Lo Addr	Quantity		Error Check Field
11	10	00	87	00	02	56

7 MBS Protocol

In This Chapter

This chapter contains information specific to the PLX3x-series gateway Modbus Serial (MBS) protocol driver.

7.1 MBS Functional Overview

The Modbus Serial protocol supports both Master and slave implementations of the protocol. Each of the gateway serial ports is individually configurable to communicate to separate networks.

7.1.1 Modbus Serial Specifications

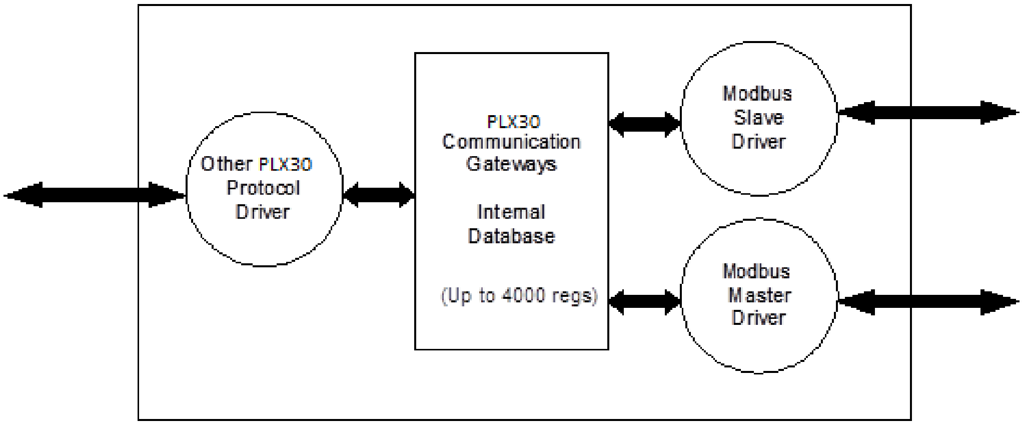
Command List	Up to 100 commands per Master port, each fully configurable for Function Code, slave address, register to/from addressing and word/bit count.
Supported Modbus Function Codes	1: Read Coil Status 2: Read Input Status 3: Read Holding Registers 4: Read Input Registers 5: Force (Write) Single Coil 6: Preset (Write) Single Holding Register 15: Force (Write) Multiple Coils 16: Preset (Write) Multiple Holding Registers
Polling of Command List	Configurable polling of command list, including continuous and on change of data, and dynamically user or automatic enabled.
Status Data	Error codes available on an individual command basis. In addition, a slave status list is maintained per active Modbus Master port.
Node Address	1 to 247 (software selectable)
RS Interface	RS232, RS422, and RS485

7.1.2 Modbus Master/Slave Port Specifications

Type	**Specifications**
General Parameters	
Internal Database	Up to 4000 registers (words) available.
Communication parameters	Port 0: Baud Rate: 110 to 115K baud Port 1, 2, 3: Baud Rate: 110 to 115K baud Stop Bits: 1 or 2 Data Size: 5 to 8 bits Parity: None, Even, Odd RTS Timing delays: 0 to 65535 milliseconds
Modbus Modes	RTU mode (binary) with CRC-16 ASCII mode with LRC error checking
Floating Point Data	Floating point data movement supported, including configurable support for Enron implementation
Modbus Function Codes	1: Read Coil Status 2: Read Input Status 3: Read Holding Registers 4: Read Input Registers 5: Force (Write) Single Coil 6: Preset (Write) Single Register 15: Force(Write) Multiple Coils 16: Force (Write) Multiple Register 22: Mask Write Holding Register (Slave Only) 23: Read/Write Holding Registers (Slave Only)
Modbus Master	
Command List	Up to 100 command per Master port, each fully configurable for function, slave address, register to/from addressing and word/bit count
Status Data	Error codes available on an individual command basis. In addition, a slave status list is maintained per active Modbus Master port.
Polling of command list	Configurable polling of command list, including continuous and on change of data
Modbus Slave	
Node address	1 to 247 (software selectable)
Status Data	Error codes, counters and port status available per configured slave port starting at memory register 4400.

7.1.3 Gateway Internal Database

The internal gateway database is central to the functionality of the gateway. This database is shared between all the ports on the gateway, permitting data from devices on one communication network port to be viewed and controlled by devices on another port or network.



Modbus Port Access to Database

The Multiple slave support in the gateway permits remote Master applications (HMI software, Quantum processors, etc) to read from and write to the gateway's database.

When configured as a slave, the internal database of the gateway is used as the source for read requests and the destination for write requests from remote Masters. Access to the database is controlled by the command type received in the incoming message from the remote Master. The following table defines the relationship of the gateway's internal database to the addresses required in the incoming Modbus TCP/IP requests.

Database Address	Modbus Address
0	40001 (five-digit addressing) or 400001 (six-digit addressing)
1000	41001 or 410001
2000	42001 or 420001
3000	43001 or 430001
3999	44000 or 440000

7.2 MBS Configuration

7.2.1 MBS Port [x]

The Modbus Port [x] sections of the **PCB Tree Window**, where x stands for 1, 2, 3 or 4, set the Modbus Master and slave port communication parameters and specify the protocol-specific settings.

Configuration Parameters Common to Master and Slave

Parameter	Value	Description
Enabled	YES or NO	Specifies if the port will be used. If the parameter is set to **No**, the port will not be used. If the parameter is set to **Yes**, the port will be used.
RS Interface	RS-232 RS-485 RS-422	Specifies the electrical interface for the ports.
Type	MASTER or SLAVE	Specifies if the port will emulate a Master or slave device. **Master** - The gateway initiates Modbus commands to one or more Modbus devices **Slave**- The gateway responds to Modbus commands initiated by a Modbus master
Float Flag	YES or NO	Specifies if the floating-point data access functionality is to be implemented. If the float flag is set to **YES**, Modbus functions 3, 6, and 16 will interpret floating-point values for registers as specified by the two following parameters.
Float Start	0 to 32767	Specifies the first register of floating-point data. All requests with register values greater than or equal to this value will be considered floating-point data requests. This parameter is only used if the Float Flag is enabled. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered as floating-point data.
Float Offset	0 to 3998	Specifies the start register for floating-point data in the internal database. This parameter is used only if the Float Flag is enabled. For example, if the Float Offset value is set to 3000 and the float start parameter is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.
Protocol	RTU or ASCII	Specifies the Modbus protocol version to be used on the port.
Baud Rate	Various	Specifies the baud rate to be used on the port.
Parity	None Odd Even	Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use. All devices communicating through this port must use the same parity setting.
Data Bits	7 or 8	Specifies the number of data bits for each word used by the protocol. All devices communicating through this port must use the same number of data bits.
Stop Bits	1 or 2	Stop bits signal the end of a character in the data stream. For most applications, use one stop bit. For slower devices that require more

Parameter	Value	Description
		time to re-synchronize, use two stop bits. All devices communicating through this port must use the same number of stop bits.
RTS On	0 to 65535	Specifies the number of milliseconds to delay data transmission after *Ready To Send* (RTS) is asserted.
RTS Off	0 to 65535	Specifies the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal is set low.
Use CTS Line	YES or NO	This parameter specifies if the *Clear To Send* (CTS) modem control line is to be used or not. If the parameter is set to NO, the CTS line will not be monitored. If the parameter is set to YES, the CTS line will be monitored and must be high before the gateway will send data. Normally, this parameter is required when half-duplex modems are used for communication (2-wire). This procedure is commonly referred to as hardware handshaking.

Master Configuration Parameters

Parameter	Value	Description
Response Timeout	0 to 65535	Specifies the command response timeout period in 1 millisecond increments. This is the time that a port configured as a Master will wait for a response from the addressed slave before re-transmitting the command (Retries) or skipping to the next command in the Command List. The value to specify depends on the communication network used and the expected response time (plus or minus) of the slowest device on the network.
Retry Count	0 to 10	Specifies the number of times a command will be retried if it fails.
End of Message Delay	0 to 65535	Specifies a time delay in milliseconds to be added to the 3.5 character time delay used by the module to recognize the end of a message. Certain applications may require validation of Modbus messages with more than 3.5 character time between consecutive bytes (example: modem applications). A value of 0 will cause the default end of message delay to be used.
Minimum Command Delay	0 to 32767	Specifies the number of milliseconds to wait between receiving the end of a slave's response to the most recently transmitted command and the issuance of the next command. This parameter can be used to place a delay after each command to avoid sending commands on the network faster than the slaves can be ready to receive them. This parameter does not affect retries of a command, as retries will be issued when a command failure is recognized.
Error Delay Counter	0 to 60000	Specifies the number of poll attempts to be skipped before trying to re-establish communications with a slave that has failed to respond to a command within the time limit set by the *Response Timeout* parameter. After the slave fails to respond, the master will skip sending commands that should have been sent to the slave until the number of skipped commands matches the value entered in this parameter. This creates a sort of *slow poll* mode for slaves that are experiencing communication problems.
Command	-1, 0 to 3900	This parameter allows the execution of commands in the Command

Parameter	Value	Description
Control Reg		List (page 131) to be controlled by setting a value of zero (0), one (1), two (2), or three (3) into a 100-register area of the gateway database, beginning at the address entered in this parameter. If commands in the list are specified with an Enable code of zero for all commands, no commands will be executed. If the value in the first control register is changed to one (1), command zero will be executed continuously. If the value is the control register for a write command is set to two (2), the command will be enabled for conditional writing, which will cause the command to be executed whenever the values in the database registers associated with the command change. Use the value of three (3) only for bit-level write commands, FC 5 and 15. If the parameter is set to three (3), the command will be executed only if the internal bit data associated with the command changes. It will also clear the bit or bits in the internal database after the write command is built. To use this feature, configure the commands as disabled (enable code = 0). This Command Control feature can be disabled by setting this parameter to a value of 0 to -1.

Slave Configuration Parameters

Parameter	Value	Description
Minimum Response Delay	0 to 65535	Specifies the number of milliseconds to wait before responding to a command received on the port from a remote Master. This delay is sometimes required to accommodate slower Master devices.
Internal Slave ID	0 to 247	Defines the virtual Modbus slave address for the port. Any commands received on the slave port, addressed to the node address entered here, will be processed by the gateway. Each slave device on the network must be assigned a unique address.
Bit Input Offset	0 to 3998	Specifies the internal database address to use as the zero address or starting point for binary Input Status data. Input Status data is read-only data, requested by Modbus Function Code 2 commands (Read Input Status). For example, if this parameter is set to 150 and a Function Code 2 command is received requesting Input Status address 0 (virtual Modbus Address 10001 or 100001), the data returned in the response will be the bit value at register 150, bit 0 in the gateway's database.
Word Input Offset	0 to 3998	Specifies the internal database address to use as the zero address or starting point for Input Register (16-bit integer) data. Input Register data is read-only data, requested by Modbus Function Code 4 commands (Read Input Registers). For example, if this parameter is set to 500 and a Function Code 4 command is received requesting Input Register address 0 (virtual Modbus address 30001 or 300001), the data returned in the response will be the value at register 500 in the gateway's database.
Output Offset	0 to 3998	Specifies the internal database address to use as the zero address or starting point for binary output Coil data. Coil data is read by Modbus Function Code 1 commands (Read Coils) and written by Function Codes 5 (Force Single Coil) or Function Code 15 (Force Multiple Coils). For example, if this parameter is set to 50 and a Function Code 1 command is received requesting Coil address 0 (virtual Modbus Coil address 00001 or 000001), the data returned in the response will be the value at register 50, bit 0 in the gateway's database.
Holding Register Offset	0 to 3998	Specifies the internal database address to use as the zero address or starting point for Holding Register (16-bit integer) data. Holding Register data is read by Modbus Function Code 3 commands (Read Holding Registers) and written by Function Code 6 (Preset Single Register) or Function Code 16 (Preset Multiple Registers). For example, if this parameter is set to 1000 and a Function Code 3 command is received requesting Holding Register address 0 (virtual Modbus address 40001 or 400001), the data returned in the response will be the value at register 1000 in the gateway's database.
End of Message Delay	0 to 65535	Specifies a time delay in milliseconds to be added to the 3.5 character time delay used by the module to recognize the end of a message. Certain applications may require validation of Modbus messages with more than 3.5 character time between consecutive bytes (example: modem applications). A value of 0 will cause the default end of message delay to be used.

7.2.2 MBS Port [x] Commands

The Modbus Port[x] Commands (where x can be 1, 2, 3 or 4) sections of the PCB tree Window are used to define a Master serial port *Command List*. This list holds the parameters needed to poll slave devices attached to a Master port.

Parameter	Value	Description
Enable	Disabled Continuous Event Command Conditional	Specifies if the command is to be executed and under what conditions. **Disabled (0)** - the command is disabled and will not be executed in the normal polling sequence. However, the command can still be activated using Command Control. **Continuous (1)** - the command will be executed upon each scan of the Command List if the Poll Interval is set to zero (0). If the *Poll Interval* is set to a non-zero value, the command will be executed when the interval timer for that command expires. **Event Command (2)** - the command will execute only if the internal data associated with the command changes. This parameter is valid only for write commands (FC 5, 6, 15, and 16). **Conditional (3)** - the command will execute only if the internal bit data associated with the command changes. It will also clear the bit or bits in the internal database after the write command is built. This parameter is valid only for bit-level write commands (FC 5 and 15).
Internal Address	0 to 3999 (for register-level addressing) or 0 to 63999 (for bit-level addressing)	Specifies the database address in the gateway's internal database to use as the destination for data brought in by a read command, or as the source for data to be sent out by a write command. The database address is interpreted as a bit address or a 16-bit register (word) address, depending on the Modbus Function Code used in the command. For Modbus functions 1, 2, 5, and 15, this parameter is interpreted as a bit-level address. For Modbus functions 3, 4, 6, and 16, this parameter is interpreted as a register-level address.
Poll Interval	0 to 65535	Specifies the minimum interval between executions of continuous commands. The value is in seconds. Therefore, if a value of 10 is entered, the command will execute no more frequently than once every 10 seconds.
Reg Count	1 to 125 (for registers) or 1 to 800 (for coils)	Specifies the number of 16-bit registers or binary bits to be transferred by the command. Modbus functions 5 and 6 ignore this field as they apply only to a single data point. For Modbus functions 1, 2, and 15, this parameter sets the number of bits (inputs or coils) to be transferred by the command. For Modbus functions 3, 4, and 16, this parameter sets the number of registers to be transferred by the command.
Swap Code	No Change Word Swap Word and Byte Swap Byte Swap	Specifies if and how the order of bytes in data received or sent is to be rearranged. This option exists to allow for the fact that different manufacturers store and transmit multi-byte data in different combinations. This parameter is helpful when dealing with floating-point or other multi-byte values, as there is no one standard method of storing these data types. The parameter can be set to rearrange the byte order of data received or sent into an order more useful or

Parameter	Value	Description
		convenient for other applications. **No change(0)** - No change is made in the byte ordering (1234 = 1234) **Word Swap (1)** -The words are swapped (1234=3412) **Word and Byte Swap (2)** - The words are swapped, then the bytes in each word are swapped (1234=4321) **Byte Swap (3)** - The bytes in each word are swapped (1234=2143) These swap operations affect 4-byte (or 2-word) groups of data. Therefore, data swapping using these *Swap Codes* should be done only when using an even number of words, such as when 32-bit integer or floating-point data is involved.
Node Address	1 to 255 (0 is a broadcast)	Specifies the node address of the Modbus slave device. Values of 1 to 255 are permitted. **Note:** Most Modbus devices only accept addresses in the range of 1 to 247, so check with the slave device manufacturer to see if a particular slave can use addresses 248 to 255. If the value is set to zero, the command will be a broadcast message on the network. The Modbus protocol permits broadcast commands for *write* operations. **Do not** use node address 0 for *read* operations.
Modbus Function	1, 2, 3, 4, 5, 6, 15, or 16	Specifies the Modbus Function Code to be executed by the command. These function codes are defined in the Modbus protocol. (More information on the protocol is available from www.modbus.org.) The following function codes are supported by the gateway. **1** - Read Coil Status **2** - Read Input Status **3** - Read Holding Registers **4** - Read Input Registers **5** - Force (Write) Single Coil **6** - Preset (Write) Single Register **15** - Force Multiple Coils **16** - Preset Multiple Registers

Parameter	Value	Description
MB Address in Device	Varies	Specifies the starting Modbus register or bit address in the server to be used by the command. Refer to the documentation of each Modbus server device for the register and bit address assignments valid for that device. The Modbus Function Code determines whether the address will be a register-level or bit-level OFFSET address into a given data type range. The offset will be the target data address in the server minus the base address for that data type. Base addresses for the different data types are: ♦ 00001 or 000001 (0x0001) for bit-level Coil data (Function Codes 1, 5, and 15). ♦ 10001 or 100001 (1x0001) for bit-level Input Status data (Function Code 2) ♦ 30001 or 300001 (3x0001) for Input Register data (Function Code 4) ♦ 40001 or 400001 (4x0001) for Holding Register data (Function Codes 3, 6, and 16). ♦ Address calculation examples: o For bit-level Coil commands (FC 1, 5, or 15) to read or write a Coil 0X address 00001, specify a value of 0 (00001 - 00001 = 0). o For Coil address 00115, specify 114 o (00115 - 00001 = 114) ♦ For register read or write commands (FC 3, 6, or 16) 4X range, for 40001, specify a value of 0 o (40001 - 40001 = 0). ♦ For 01101, 11101, 31101 or 41101, specify a value of 1100. o (01101 - 00001 = 1100) o (11101 -10001 = 1100) o (31101 - 30001 = 1100) o (41101 - 40001 = 1100) **Note:** If the documentation for a particular Modbus server device lists data addresses in hexadecimal (base16) notation, you will need to convert the hexadecimal value to a decimal value to enter in this parameter. In such cases, it is not usually necessary to subtract 1 from the converted decimal number, as this addressing scheme typically uses the exact offset address expressed as a hexadecimal number.

7.3 MBS Diagnostics

7.3.1 PCB Diagnostics

The best way to troubleshoot the MBS driver is to use ProSoft Configuration Builder to access the diagnostic capabilities of the gateway through the Ethernet debug port. For instructions on how to access *Diagnostics*, see *Using Diagnostics in ProSoft Configuration Builder* (page 36).

7.3.2 MBS Status Data in Upper Memory

Each Modbus port has an associated status data area located in the gateway's upper memory. The Data Map functionality of the gateway can be used to map this data into the normal user data range of the gateway's database. See *Using the CommonNet Data Map* (page 25).

General Modbus Error and Status Data

The following table lists the starting addresses in upper memory where general error and status data for each Modbus Master or slave port can be found.

Modbus Port	**Starting Address**
1	4400
2*	4800
3*	5200
4*	5600

*Status data for Ports 2 through 4 is only present in 4-port MBS gateways.

Note: None of the addresses are available in the Modbus address range. In order for them to be accessed via a Modbus request, they must be moved into the 0 to 3999 address range. See *Using the CommonNet Data Map* (page 25).

The status data area is initialized with zeros whenever the gateway is restarted. This occurs during a cold-start (power-on), reset (reset push-button pressed) or a warm-boot operation (commanded from a debug screen or after downloading a new configuration).

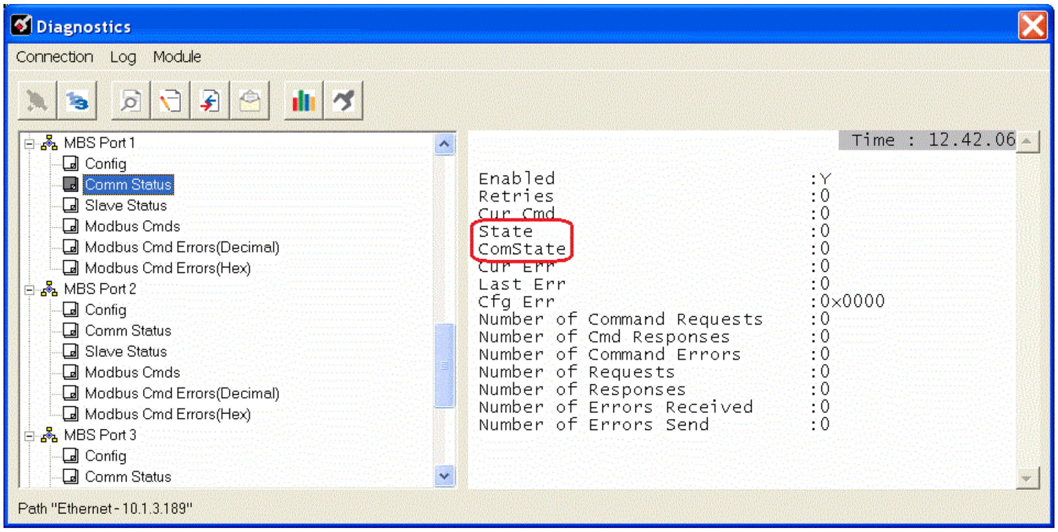
Port 1 General Modbus Error and Status Layout

The addresses listed are for Port 1only; but the format is the same for each port. The start address for each port is given in the previous section.

Example Internal Database Address	Offset	Description
4400	0	Number of Command Requests
4401	1	Number of Command Responses
4402	2	Number of Command Errors
4403	3	Number of Requests
4404	4	Number of Responses
4405	5	Number of Errors Sent
4406	6	Number of Errors Received
4407	7	Configuration Error Code
4408	8	Current Error/Index
4409	9	Last Error/Index

Slave Port: General Port STATE and COMM STATE Status

Note: There are two additional port status parameters that are not available from the General Modbus Error and Status Data area. The status values, Port STATE and Port COMM STATE (circled in red), can be found only in the PCB diagnostic menus under Comm Status for a port as shown in the following graphic.



```
                                     Time : 12.42.06

Enabled                         :Y
Retries                         :0
Cur Cmd                         :0
State                           :0
ComState                        :0
Cur Err                         :0
Last Err                        :0
Cfg Err                         :0x0000
Number of Command Requests      :0
Number of Cmd Responses         :0
Number of Command Errors        :0
Number of Requests              :0
Number of Responses             :0
Number of Errors Received       :0
Number of Errors Send           :0
```

Path "Ethernet - 10.1.3.189"

These status values are state registers. They are used as ‘scratchpad’ areas by the operating firmware to keep track of the current logical state of activities on a slave port. These state registers are constantly changing as the gateway progresses through the various stages needed to process communication on the serial ports. This processing happens faster than can be followed, unless a port error causes the value to remain constant for some noticeable length of time.

The diagnostic screen displays are 'snapshot' type screens; that is, they display the current value at the time they are called but do not continue to automatically update the data displayed. They are not live-data screens. Therefore, to see a change in STATE or COMM STATE, you must repeatedly call the screen by pressing the appropriate menu key to have the screen refreshed. Depending on the refresh timing you may or may not see a change in the displayed values every time you call for an update.
The STATE register may display any of the following values.

STATE Value	Description
-2	Preparing port, Flushing all buffers, Scan for RTS-Off, or Waiting for port enable signal
-1	Waiting for receipt of data
0	Undefined state
1	Receiving a message from the Master
2	Building a slave response message
7	Modbus master is fetching next command
2000	Sending slave response to the Master

The COMM STATE register may display any of the following values.

COMM STATE Value	Description
0	Port not sending data (Wait for send)
1	Setting up minimum Response Delay Timer
101	Waiting for Minimum Response Delay to timeout
2	Turning on RTS line and starting RTS-On Delay Timer
3	Waiting for RTS-On Delay timer to timeout
4	Sending data
5	Waiting for all data to be sent and starting RTS-Off Delay Timer
6	Waiting for RTS-Off Delay timeout and then turning OFF RTS line after timeout

Master Port: Command List Errors

The individual command errors for each Master port are returned to the address locations specified in the following table. Each port can have up to 100 commands configured. Each configured command will use one word of these data areas to store a value representing the execution status from the most recent command execution attempt.

Modbus Port	Address Range
1	4410 to 4509
2*	4810 to 4909
3*	5210 to 5309
4*	5610 to 5709

*Status data for Ports 2 through 4 is only present in 4-port MBS gateways

Port 1 Command Error List Layout

The first word in the specified register location contains the status/error code for the first command in the port's *Command List*. Successive words in the *Command Error List* are associated with corresponding commands in the list.

The addresses listed are for Port 1 only; but the format is the same for each port. The start address for each port is given in the previous section.

Internal Database Address (Example)	Offset	Description
4410	0	Command #1 Error Code
4411	1	Command #2 Error Code
4412	2	Command #3 Error Code
4413	3	Command #4 Error Code
4414	4	Command #5 Error Code
...	...	...
...	...	...
...	...	...
4507	97	Command #98 Error Code
4508	98	Command #99 Error Code
4509	99	Command #100 Error Code

Note that the values in the *Command Error List* tables are initialized to zero (0) at power-up, cold boot, and warm boot. If a command executes successfully, the value in the associated register will remain at zero (0), indicating no command error was detected. Any non-zero value in this table indicates the corresponding command experienced an error.

The data in this table is dynamic. It is updated each time a command is executed. Therefore, if the command fails once and succeeds on the next attempt, the Error Code from the previously failed attempt will be replace with zero and be lost. Error Codes are not archived in the gateway's database. To see if the port has experienced an error since the most recent restart and what the most recently occurring error was, if any, you can check the Last Error/Index.

<u>Master Port: Modbus Slave List Status</u>

Slave List Status values for each Master port are returned to the address locations specified in the following table.

Modbus Port	Address Range
1	4510 to 4764
2*	4910 to 5164
3*	5310 to 5564
4*	5710 to 5964

*Status data for Ports 2 through 4 is only present in 4-port MBS gateways.

Port 1 Slave List Status Layout

The addresses listed are for Port 1 only; but the format is the same for each port. The start address for each port is given in the previous section.

Internal Database Address (Example)	Offset	Description
4510	0	Slave #1 Status
4511	1	Slave #2 Status
4512	2	Slave #3 Status
4513	3	Slave #4 Status
4514	4	Slave #5 Status
	.	.
	.	.

The slave status list contains the current poll status of each slave device on a Master port. Slaves attached to a Master port can have one of three states.

0	The slave has not defined in the command list for the Master port and will not be polled from the Command List.
1	The slave is configured to be polled by the Master port and the most recent communication attempt was successful.
2	The Master port has failed to communicate with the slave device. Communication with the slave is suspended for a user defined period based on the scanning of the command list.

Slaves are defined to the system when the gateway loads the Master Command List during start-up and initialization. Each slave defined will be set to a state value of 1 in this initial step. If the Master port fails to communicate with a slave device (timeout expired on a command, retries failed), the Master will set the state of the slave to a value of 2 in this status table. This suspends communication with the slave device for a user-specified Error Delay Count.

When the Master first suspends polling of a particular slave, it creates an Error Delay Counter for this slave address and set the value in that counter equal to the Error Delay Counter parameter in the configuration file. Then, each time a command in the list is scanned that has the address of a suspended slave, the delay counter value for that slave will be decremented. When the value reaches zero, the slave state will be set to 1. This will re-enable polling of the slave.

The first word in the defined register locations contains the status code for slave node address 1. Each successive word in the list is associated with the next node in sequence, up to slave node address 247.

Note: The values in the slave List Status tables are initialized to zero (0) at power-up, cold boot and during warm boot.

7.3.3 Error/Status Codes

These are error codes that are part of the Modbus protocol or are extended codes unique to this gateway.

Modbus Error Codes

These error codes are generated or returned on both the Master and slave ports. These codes are the standard Modbus errors (page 149).

Code	**Description**
1	Illegal Function
2	Illegal Data Address
3	Illegal Data Value
4	Failure in Associated Device
5	Acknowledge
6	Busy, Rejected Message

Module Communication Error Codes

These gateway-specific error codes are also returned from the command polling process and stored in the Command Error List memory area.

Code	**Description**
-1	CTS modem control line not set before transmit
-2	Timeout while transmitting message
-11	Timeout waiting for response after request
253	Incorrect slave address in response
254	Incorrect function code in response
255	Invalid CRC/LRC value in response

Command List Error Codes

These command-specific error codes are detected during command list loading at gateway reboot and are stored in the *Command Error List* memory area.

Code	**Description**
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (<0 or > 255)
-44	Count parameter set to 0
-45	Invalid function code
-46	All parameters set to 0
-47	All parameters set to -1

7.4 MBS Reference

7.4.1 Modbus Protocol Specification

Read Coil Status (Function Code 01)

Query

This function allows the user to obtain the ON/OFF status of logic coils used to control discrete outputs from the addressed server only. Broadcast mode is not supported with this function code. In addition to the server address and function fields, the message requires that the information field contain the initial coil address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 coils to be obtained at each request; however, the specific server device may have restrictions that lower the maximum quantity. The coils are numbered from zero; (coil number 1 = zero, coil number 2 = one, coil number 3 = two, and so on).

The following table is a sample read output status request to read coils 0020 to 0056 from server device number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data # Of Pts Ho	Data # Of Pts Lo	Error Check Field
11	01	00	13	00	25	CRC

Response

An example response to Read Coil Status is as shown in Figure C2. The data is packed one bit for each coil. The response includes the server address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each coil (1 = ON, 0 = OFF). The low order bit of the first character contains the addressed coil, and the remainder follows. For coil quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the server interface device is serviced at the end of a controller's scan, data will reflect coil status at the end of the scan. Some servers will limit the quantity of coils provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status from sequential scans.

Adr	Func	Byte Count	Data Coil Status 20 to 27	Data Coil Status 28 to 35	Data Coil Status 36 to 43	Data Coil Status 44 to 51	Data Coil Status 52 to 56	Error Check Field
11	01	05	CD	6B	B2	OE	1B	CRC

The status of coils 20 to 27 is shown as CD (HEX) = 1100 1101 (Binary). Reading left to right, this shows that coils 27, 26, 23, 22, and 20 are all on. The other coil data bytes are decoded similarly. Due to the quantity of coil statuses requested, the last data field, which is shown 1B (HEX) = 0001 1011 (Binary), contains the status of only 5 coils (52 to 56) instead of 8 coils. The 3 left most bits are provided as zeros to fill the 8-bit format.

Read Input Status (Function Code 02)

Query

This function allows the user to obtain the ON/OFF status of discrete inputs in the addressed server PC Broadcast mode is not supported with this function code. In addition to the server address and function fields, the message requires that the information field contain the initial input address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 inputs to be obtained at each request; however, the specific server device may have restrictions that lower the maximum quantity. The inputs are numbered form zero; (input 10001 = zero, input 10002 = one, input 10003 = two, and so on, for a 584).

The following table is a sample read input status request to read inputs 10197 to 10218 from server number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data #of Pts Hi	Data #of Pts Lo	Error Check Field
11	02	00	C4	00	16	CRC

Response

An example response to Read Input Status is as shown in Figure C4. The data is packed one bit for each input. The response includes the server address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each input (1=ON, 0=OFF). The lower order bit of the first character contains the addressed input, and the remainder follows. For input quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as a quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the server interface device is serviced at the end of a controller's scan, data will reflect input status at the end of the scan. Some servers will limit the quantity of inputs provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status for sequential scans.

Adr	Func	Byte Count	Data Discrete Input 10197 to 10204	Data Discrete Input 10205 to 10212	Data Discrete Input 10213 to 10218	Error Check Field
11	02	03	AC	DB	35	CRC

The status of inputs 10197 to 10204 is shown as AC (HEX) = 10101 1100 (binary). Reading left to right, this show that inputs 10204, 10202, and 10199 are all on. The other input data bytes are decoded similar.

Due to the quantity of input statuses requested, the last data field which is shown as 35 HEX = 0011 0101 (binary) contains the status of only 6 inputs (10213 to 102180) instead of 8 inputs. The two left-most bits are provided as zeros to fill the 8-bit format.

Read Holding Registers (Function Code 03)

Query

Read Holding Registers (03) allows the user to obtain the binary contents of holding registers 4xxxx in the addressed server. The registers can store the numerical values of associated timers and counters which can be driven to external devices. The addressing allows up to 125 registers to be obtained at each request; however, the specific server device may have restrictions that lower this maximum quantity. The registers are numbered form zero (40001 = zero, 40002 = one, and so on). The broadcast mode is not allowed.

The example below reads registers 40108 through 40110 from server 584 number 11.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	03	00	6B	00	03	CRC

Response

The addressed server responds with its address and the function code, followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are two bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the server interface device is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Some servers will limit the quantity of register content provided each scan; thus for large register quantities, multiple transmissions will be made using register content from sequential scans.

In the example below, the registers 40108 to 40110 have the decimal contents 555, 0, and 100 respectively.

Adr	Func	ByteCnt	Hi Data	Lo Data	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field
11	03	06	02	2B	00	00	00	64	CRC

Read Input Registers (Function Code 04)

Query

Function code 04 obtains the contents of the controller's input registers at addresses 3xxxx. These locations receive their values from devices connected to the I/O structure and can only be referenced, not altered from within the controller, The addressing allows up to 125 registers to be obtained at each request; however, the specific server device may have restrictions that lower this maximum quantity. The registers are numbered for zero (30001 = zero, 30002 = one, and so on). Broadcast mode is not allowed.

The example below requests the contents of register 3009 in server number 11.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	04	00	08	00	01	CRC

Response

The addressed server responds with its address and the function code followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are 2 bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the server interface is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Each PC will limit the quantity of register contents provided each scan; thus for large register quantities, multiple PC scans will be required, and the data provided will be form sequential scans.

In the example below the register 3009 contains the decimal value 0.

Adr	Func	Byte Count	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	04	02	00	00	E9

Force Single Coil (Function Code 05)

Query

This message forces a single coil either ON or OFF. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coil is disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 0001 = zero, coil 0002 = one, and so on). The data value 65,280 (FF00 HEX) will set the coil ON and the value zero will turn it OFF; all other values are illegal and will not affect that coil.

The use of server address 00 (Broadcast Mode) will force all attached servers to modify the desired coil.

Note: Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

The example below is a request to server number 11 to turn ON coil 0173.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/off Ind	Data	Error Check Field
11	05	00	AC	FF	00	CRC

Response

The normal response to the Command Request is to re-transmit the message as received after the coil state has been altered.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/ Off	Data	Error Check Field
11	05	00	AC	FF	00	CRC

The forcing of a coil via MODBUS function 5 will be accomplished regardless of whether the addressed coil is disabled or not (*In ProSoft products,* the coil *is only affected if the necessary ladder logic is implemented).*

Note: The Modbus protocol does not include standard functions for testing or changing the DISABLE state of discrete inputs or outputs. Where applicable, this may be accomplished via device specific Program commands (*In ProSoft products, this is only accomplished through ladder logic programming).*

Coils that are reprogrammed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function Code 5 and (even months later), an output is connected to that coil, the output will be "hot".

Preset Single Register (Function Code 06)

Query

Function (06) allows the user to modify the contents of a holding register. Any holding register that exists within the controller can have its contents changed by this message. However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller unused high order bits must be set to zero. When used with server address zero (Broadcast mode) all server controllers will load the specified register with the contents specified.

Note: Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	06	00	01	00	03	CRC

Response

The response to a preset single register request is to re-transmit the query message after the register has been altered.

Adr	Func	Data Reg Hi	Data Reg Lo	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	06	00	01	00	03	CRC

Diagnostics (Function Code 08)

MODBUS function code 08 provides a series of tests for checking the communication system between a Client device and a server, or for checking various internal error conditions within a server.

The function uses a two-byte sub-function code field in the query to define the type of test to be performed. The server echoes both the function code and sub-function code in a normal response. Some of the diagnostics cause data to be returned from the remote device in the data field of a normal response.

In general, issuing a diagnostic function to a remote device does not affect the running of the user program in the remote device. Device memory bit and register data addresses are not accessed by the diagnostics. However, certain functions can optionally reset error counters in some remote devices.

A server device can, however, be forced into 'Listen Only Mode' in which it will monitor the messages on the communications system but not respond to them. This can affect the outcome of your application program if it depends upon any further exchange of data with the remote device. Generally, the mode is forced to remove a malfunctioning remote device from the communications system.

Sub-function Codes Supported

Only Sub-function 00 is supported by the gateway.

00 Return Query Data

The data passed in the request data field is to be returned (looped back) in the response. The entire response message should be identical to the request.

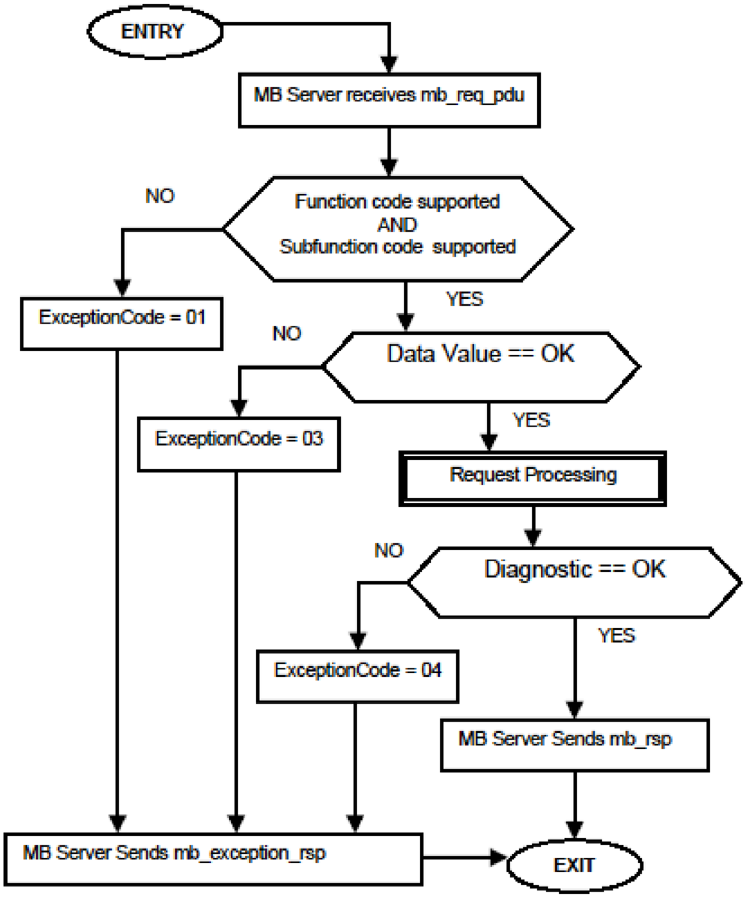
Sub-function	Data Field (Request)	Data Field (Response)
00 00	Any	Echo Request Data

Example and State Diagram

Here is an example of a request to remote device to Return Query Data. This uses a sub-function code of zero (00 00 hex in the two-byte field). The data to be returned is sent in the two-byte data field (A5 37 hex).

Request		**Response**	
Field Name	**(Hex)**	**Field Name**	**(Hex)**
Function	08	Function	08
Sub-function Hi	00	Sub-function Hi	00
Sub-function Lo	00	Sub-function Lo	00
Data Hi	A5	Data Hi	A5
Data Lo	37	Data Lo	27

The data fields in responses to other kinds of queries could contain error counts or other data requested by the sub-function code.



Modbus Exception Responses

When a Modbus Master sends a request to a server device, it expects a normal response. One of four possible events can occur from the Master's query:

- If the server device receives the request without a communication error, and can handle the query normally, it returns a normal response.
- If the server does not receive the request due to a communication error, no response is returned. The Master program will eventually process a timeout condition for the request.
- If the server receives the request, but detects a communication error (parity, LRC, CRC, etc...), no response is returned. The Master program will eventually process a timeout condition for the request.
- If the server receives the request without a communication error, but cannot handle it (for example, if the request is to read a non-existent output or register), the server will return an exception response informing the Master of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

Function Code Field: In a normal response, the server echoes the function code of the original request in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response.

With the function code's MSB set, the Master's application program can recognize the exception response and can examine the data field for the exception code.

Data Field: In a normal response, the server may return data or statistics in the data field (any information that was requested in the request). In an exception response, the server returns an exception code in the data field. This defines the server condition that caused the exception.

The following table shows an example of a Master request and server exception response.

Request		**Response**	
Field Name	**(Hex)**	**Field Name**	**(Hex)**
Function	01	Function	81
Starting Address Hi	04	Exception Code	02
Starting Address Lo	A1		
Quantity of Outputs Hi	00		
Quantity of Outputs Lo	01		

In this example, the Master addresses a request to server device. The function code (01) is for a Read Output Status operation. It requests the status of the output at address 1245 (04A1 hex). Note that only that one output is to be read, as specified by the number of outputs field (0001).

If the output address does not exist in the server device, the server will return the exception response with the exception code shown (02). This specifies an illegal data address for the server.

Modbus Exception Codes

Code	Name	Meaning
01	Illegal Function	The function code received in the query is not an allowable action for the server. This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	Illegal Data Address	The data address received in the query is not an allowable address for the server. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed; a request with offset 96 and length 5 will generate exception 02.
03	Illegal Data Value	A value contained in the query data field is not an allowable value for server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, because the Modbus protocol is unaware of the significance of any particular value of any particular register.
04	Slave Device Failure	An unrecoverable error occurred while the server was attempting to perform the requested action.
05	Acknowledge	Specialized use in conjunction with programming commands. The server has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the Master. The Master can next issue a poll program complete message to determine if processing is completed.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The server is engaged in processing a long-duration program command. The Master should retransmit the message later when the server is free.
08	Memory Parity Error	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server attempted to read record file, but detected a parity error in the memory. The Master can retry the request, but service may be required on the server device.
0a	Gateway Path Unavailable	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0b	Gateway Target Device Failed To Respond	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Force Multiple Coils (Function Code 15)

Query

This message forces each coil in a consecutive block of coils to a desired ON or OFF state. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coils are disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 00001 = zero, coil 00002 = one, and so on). The desired status of each coil is packed in the data field, one bit for each coil (1= ON, 0= OFF). The use of server address 0 (Broadcast Mode) will force all attached servers to modify the desired coils.

Note: Functions 5, 6, 15, and 16 are the only messages (other than Loopback Diagnostic Test) that will be recognized as valid for broadcast.

The following example forces 10 coils starting at address 20 (13 HEX). The two data fields, CD =1100 and 00 = 0000 000, indicate that coils 27, 26, 23, 22, and 20 are to be forced on.

Adr	Func	Hi Add	Lo Add	Quantity	Byte Cnt	Data Coil Status 20 to 27	Data Coil Status 28 to 29	Error Check Field	
11	0F	00	13	00	0A	02	CD	00	CRC

Response

The normal response will be an echo of the server address, function code, starting address, and quantity of coils forced.

Adr	Func	Hi Addr	Lo Addr	Quantity	Error Check Field	
11	0F	00	13	00	0A	CRC

The writing of coils via Modbus function 15 will be accomplished regardless of whether the addressed coils are disabled or not.

Coils that are not programmed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function code 15 and (even months later) an output is connected to that coil, the output will be hot.

Preset Multiple Registers (Function Code 16)

Query

Holding registers existing within the controller can have their contents changed by this message (a maximum of 125 registers). However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller (16-bit for the 184/384 and 584); unused high order bits must be set to zero.

Note: Function codes 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Hi Add	Lo Add	Quantity		Byte Cnt	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field
11	10	00	87	00	02	04	00	0A	01	02	CRC

Response

The normal response to a function 16 query is to echo the address, function code, starting address and number of registers to be loaded.

Adr	Func	Hi Addr	Lo Addr	Quantity		Error Check Field
11	10	00	87	00	02	56

8 ASCII Protocol

In This Chapter

This chapter contains information specific to the PLX3x-series gateway ASCII protocol driver.

8.1 ASCII Functional Overview

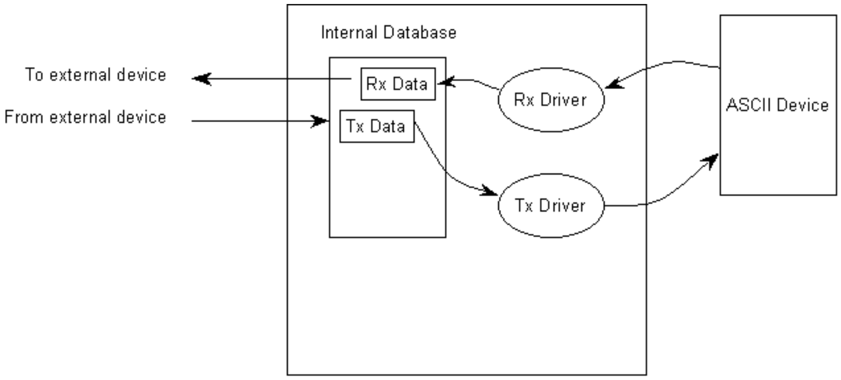
The ASCII protocol permits the PLX3x Gateway to interface any ASCII device to the many protocols and networks available. ASCII devices include barcode scanners, weigh scales, many field instruments, printers, and terminals. The protocol supports one to four serial communication ports that provide accessibility from one to four independent serial networks.

8.1.1 General Specifications

- **Ports:** One to four serial communication ports to receive and/or transmit data
- **Receive buffer size**: 255 bytes
- **Receive termination:** Stream mode, termination character(s), message timeout, inter-character delay, or packet size length
- **Receive database location:** -1 = disable receiver, 0 to 3896
- **Transmit buffer size:** 255 bytes
- **Transmit characters with pacing:** 0 to 65535 millisecond delay between each transmitted character
- **Transmit database location:** -1 = disable transmitter, 0 to 3896
- **Communication Configuration**
  - **Baud Rate:** 110 to 115,200
  - **Parity:** None, Odd, Even
  - **Data Bits:** 5 to 8
  - **Stop Bits:** 1 or 2
  - **RTS On and Off Timing:** 0 to 65535 milliseconds
  - **Minimum Response Delay:** 0 to 65535 milliseconds
  - **Hardware or Software Handshaking:** RTS/CTS, DTR/DSR, or XON/XOFF

8.1.2 Data Flow

The following illustration shows receive and transmit data flow of the ASCII protocol.



Data received from the ASCII device is accepted by the receive driver and placed in the receive database location configured by the user. The receive driver waits until the user-configured termination condition is recognized while receiving the data before placing the new data into the database.

For example, if the carriage-return character (ASCII 13) is used as the termination condition for a received message, this signals the end of the message. When the receive driver observes this character in the input stream, it takes all received characters and places them in the internal database.

In both receive and transmit operations, a signal is required to determine when new data is received or must be transmitted. The first word in the two data area is used for this purpose. There is new data available when the first word changes.

Example:

The sequence number in the receive data block has a value of 0 as set when the gateway initializes. The ASCII device sends a new data packet and the termination condition is present. The receive driver copies the data into the internal data area, sets the message length in the data area, and finally, sets the new sequence number. Receive and transmit data block structure is discussed in the following topics.

Receive Data

Data received by the receive driver is placed in the gateway's internal database in a fixed format at the location selected by the user. The receiver driver is disabled if the database start location is set to a value of -1. The following table shows the structure of the received data.

Word Offset	Description
0	Receive sequence number. This register is incremented by the gateway's Receive Driver for each new packet received.
1	Number of characters transmitted (0 to 255) from last transmit request
2	Number of characters (0 to 256) in receive block (9 to 136)
3	Receive State -1 = Listening for data 1 = Receiving port data
4	Total receive character count
5	Total receive message count
6	Transmit State 0 = Waiting for data to send 1 = RTS on 2 = RTS timeout 3 = Sending data 4 = Waiting for RTS off 5 = RTS turned off 30 = Intercharacter delay 31 = Intercharacter delay 32 = Intercharacter delay 100 = Message delay before transmit 101 = Message delay before transmit
7	Total transmit character count
8	Configuration error word
9 to 136	Received data on port

An incremented received sequence number in Word 0 signals that new data is available in Words 1, 2, and 9 through 136. (Words 3 through 8 are updated continuously by the gateway, whether or not the port is receiving messages.)

If the gateway is configured to swap the data bytes received, the receive driver will swap the bytes in each word received before placing the data into the data block. Because the data received may contain an odd number of bytes, the length of the message received will be incremented by 1 when an odd number of bytes are received and the swap option is utilized. This is to avoid losing the last byte of data in the message.

Transmit Data

Data to transmit by the transmit driver is placed in the gateway's internal database in a fixed format at the location selected by the user. The transmit driver is disabled if the database start location is set to a value of -1. The following table shows the structure of transmit data.

Word Offset	Description
0	Transmit sequence number. This number is incremented by the user's application for each new packet to transmit.
1	Number of characters received (0 to 256) from last receive request
2	Inter-character delay for this message (milliseconds between characters)
3	Number of characters to transmit on port (0 to 255)
4 to 131	Data to transmit on port

The first word of the data block is used to signal when new transmit data is available. Word 1 of the block may optionally contain the number of characters processed in the last receive message. Word 2 of the message is used to pace the characters during the transmission process. This may be required when interfacing with slow ASCII devices (that is, modems in command mode).

If the word is set to a value other than zero, a time delay corresponding to the number of milliseconds entered will be placed between each character transmitted. If the word is set to zero, the whole data packet will be transmitted as fast as the transmit driver can function. Word 3 of the data block contains the number of bytes present in the transmit data area to send out the ASCII serial communication port. Words 4 to 131 contain the actual data to transmit. If the swap option is utilized, the transmit driver will swap each byte in the words received before transmitting them.

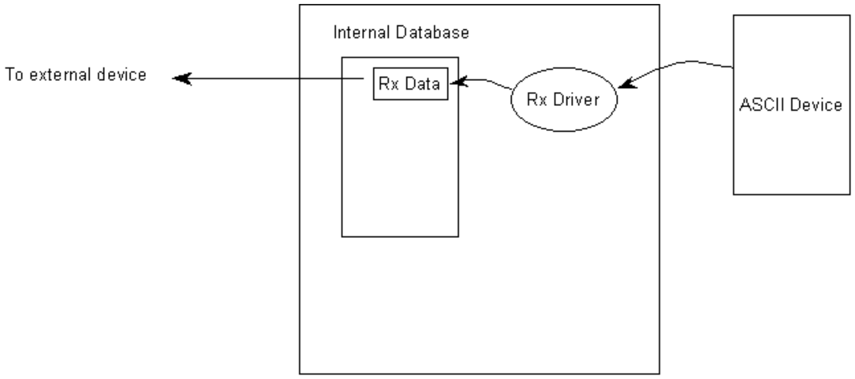
Note: If an odd number of bytes are sent by the end device when the swap option is used, then the last byte of the message may be lost.

8.1.3 Modes of Operation

The gateway can operate in several different modes with each port acting independently. The configuration of each port's driver determines its mode. The following topics describe these modes and describe the flow of data between the pieces of hardware (ASCII device and PLX3x Gateway). These topics describe the three possible types of communication devices that can be attached to application ports: receive-only, transmit-only, and transmit-receive mode.

Receive-Only Mode

A port on the gateway configured to function in receive-only mode is set up to only receive data from some sort of ASCII device. In this mode, the PLX3x gateway will never transmit data back to the ASCII device. Any data received from the ASCII device is passed from the receiver driver (Rx Driver) to the PLX3x Gateway's internal database (Rx Data). The following illustration shows the flow of data on a port configured for receive-only mode.

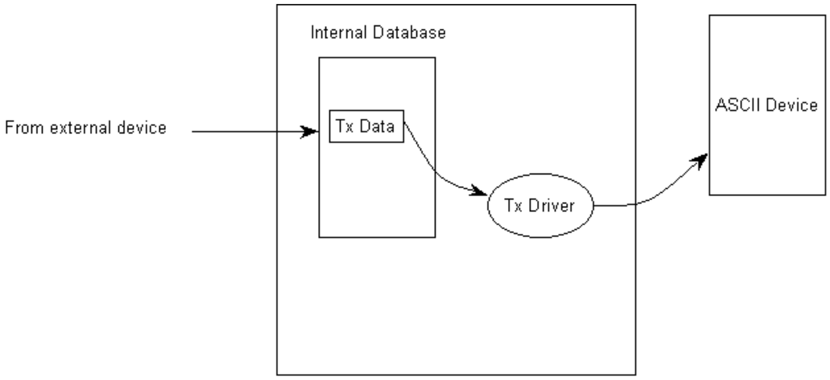


Configuring a Port for Receive-Only Mode

In order to set a port for Receive-Only mode, ensure that the **Rx DB Start** parameter in the configuration file contains the starting location of where the data will be stored. The **Tx DB Start** parameter must contain a value of **-1**. This value indicates that the port will not transmit any data.

Transmit-Only Mode

A port on the gateway configured to function in transmit-only mode is set up to only transmit data from the PLX3x gateway's internal database (received from an external source) to an ASCII device. When the transmit driver (Tx Driver) recognizes a new write block containing data (data placed in the gateway's internal database), it transmits this data out to the port. The sequence number used in the block will be different than that of the previous block, and therefore, signals that the packet is fully assembled and ready to send. The following illustration shows data flow for a transmit-only device.

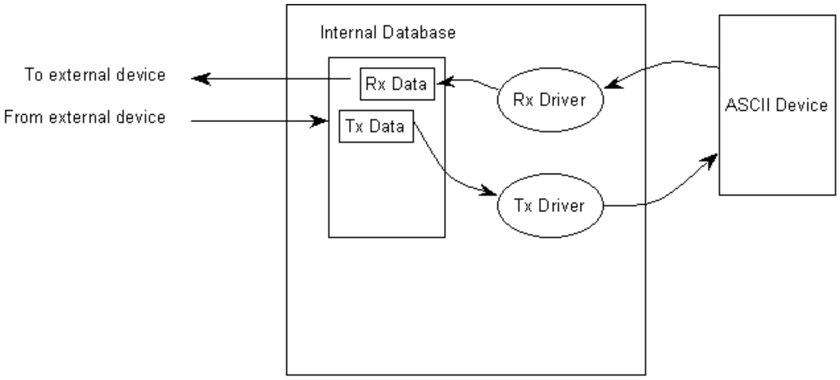


Configuring a Port for Transmit-Only Mode

In order to set a port for Transmit-Only mode, ensure that the **Tx DB Start** parameter in the configuration file contains the starting location of where the data will be stored. The **Rx DB Start** parameter must contain a value of **-1**. This value indicates that the port will not receive any data.

Transmit-Receive Mode

A port configured in transmit-receive mode can send and receive data from an ASCII device such as a terminal. This mode functions the same way as transmit-only mode or receive-only mode, but handles both the transmit and receive functions. Data flow to and from an ASCII device is handled by the gateway's transmit and receive drivers. Data received from the ASCII device is stored in the gateway's internal database until ready to be sent to an external device. Data received by an external device is also stored in the gateway's internal database until ready to be transmitted to the ASCII device. The following illustration shows the data flow when the port is configured for transmit-receive mode:



Configuring a Port for Transmit-Receive Mode

In order to set a port to both receive data and transmit data to an ASCII device, ensure that the **Rx DB Start** parameter and the **Tx DB Start** parameter both contain values that specify data storage starting locations. A **-1** value in either parameter will disable the particular function that the parameter serves.

Termination of Received Data

When data is received on the application port, the user must define in the configuration when this data will be transferred to the internal database within the gateway. This is known as the termination type for port. When the termination condition is met, the data will be sent from the port's receive buffer (data area of 255 bytes) to the internal database. This termination type is set in the bit mapped type field of the gateway object. The following illustration shows the bit map used for this parameter.

Termination Type Field

Bit(s)	**4 to 7**	**3**	**2**	**1**	**0**
Bit Value	-	8	4	2	1
Definition	Reserved	Packet size limit used	Intercharacter delay timeout used	Message timeout used	Termination character(s) used

If none of the bits are set (Type=0), the port will be configured for stream mode. Any characters received on the port are immediately sent to the processor. The processor must buffer and assemble a packet of information if this mode is selected as required by the application. If the data can be handled by the processor in this mode and it is appropriate for your application, this is the fastest method of communication between the device and the processor.

Any combination of bits is acceptable to the gateway and should be set to match the device on the specific port. An example of each termination type is given below.

Termination character(s) used

Settings:
Count = 1 (RTermCnt=1)
Termination on 0x0d (carriage return character) (RTermChar = 0d 00 00 00 …)

Data Received on port:

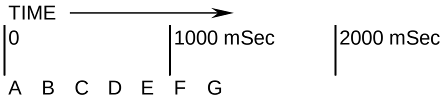
A B C 0x0d D E

Comment:
The characters "ABC" will be sent along with the 0x0d character to the controller after the 0x0d character is received. The characters "DE" will not be sent until the 0x0d character is received.

Message timeout used

Settings:
Message timeout = 1000 mSec (Rtimeout=1000)

Data Received on port:

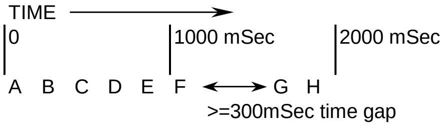


Comment:
After the 'A' character is received on the port, the message timeout is started.
The characters "ABCDE" will be sent to the controller in one block. The characters "FG" will follow in the second block one second later.

Intercharacter delay timeout used

Settings:
Intercharacter delay timeout = 300 mSec (Rdelay=300)

Data Received on port:



Comment:
After each character is received, the intercharacter delay timer is reset. The characters "ABCDEF" will be sent to the controller in one block because the delay timer expires.
The characters "GH" will follow in the second block when the next time gap is recognized.

Packet size limit used

Settings:
Packet size = 4 (RPacketLen=4)

Data Received on port:

A B C D E F G H I J

Comment:
The first block sent to the controller will contain the characters "ABCD", and the second block will contain the characters "EFGH". The characters "IJ" will not be sent until two more characters are received on the port.

8.2 ASCII Configuration

In order for the ASCII driver to function, a minimum amount of configuration data must be transferred to the gateway from the gateway's file system. Care must be taken in constructing the gateway configuration parameters. If the gateway does not function as expected, examine the configuration parameters using PCB Diagnostics.

After setting up the configuration in PCB, download it to the gateway.

8.2.1 ASCII Port [x]

Parameter	Value	Description
Enabled	YES or NO	Specifies if the port will be used. If the parameter is set to **No**, the port will not be used. If the parameter is set to **Yes**, the port will be used supporting the ASCII protocol.
RS Interface	RS-232 RS-485 RS-422	Specifies the electrical interface for the ports.
Rx DB Start	-1 or 0 to 3896	Specifies the starting location in the internal database where the received data will be stored. The buffer holds 130 words; however, the first three words of the data area define the sequence number, last write byte count and the Rx message length. If the parameter is set to -1, the port will not receive data. Refer to Receive Data (page 156) for detailed information on Rx data structure.
Tx DB Start	-1 or 0 to 3896	This parameter specifies the starting location in the internal database where the transmit data will be stored. The buffer holds 130 words; however, the first three words of the data area define the sequence number, last write byte count and the Tx message length. If the parameter is set to -1, the port will not transmit data. Refer to Transmit Data (page 157) for detailed information on Tx data structure.
Baud Rate	Various	Specifies the baud rate to be used on the port.
Parity	None Odd Even	Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use. All devices communicating through this port must use the same parity setting.
Data Bits	7 or 8	Specifies the number of data bits for each word used by the protocol. All devices communicating through this port must use the same number of data bits.
Stop Bits	1 or 2	Stop bits signal the end of a character in the data stream. For most applications, use one stop bit. For slower devices that require more time to re-synchronize, use two stop bits. All devices communicating through this port must use the same number of stop bits.
RTS On	0 to 65535	Specifies the number of milliseconds to delay data transmission after *Ready To Send* (RTS) is asserted.
RTS Off	0 to 65535	Specifies the number of milliseconds to delay after the last

Parameter	Value	Description
		byte of data is sent before the RTS modem signal is set low.
Handshaking	NO YES DTR-DSR XON-XOFF	Specifies the handshaking used on the port. **No**-No hardware or software handshaking **Yes**-RTS/CTS hardware handshaking **DTR/DSR**- DTR/DSR hardware handshaking **XON/XOFF**- software handshaking
Rx Termination Type	TERMINATION CHARACTERS MESSAGE TIMEOUT INTERCHARACTER DELAY PACKET SIZE	Defines the receive termination characteristics for the port. When the ASCII driver encounters the specified termination condition on the port, it will interpret it as end of message.
Rx Term Count	0 to 12	Used when the *Rx Termination Type* is **TERMINATION CHARACTERS**. This parameter specifies the number of different termination characters that can be used to signal the end of a received message.
Rx Term Chars	ASCII Characters	Used when the *Rx Termination Type* is **TERMINATION CHARACTERS**. This parameter specifies the termination characters used to signal the end of a received message. Enter up to 12 ASCII character codes in decimal format, with spaces between codes. The number of termination characters is set in the *Rx Term Count* parameter.
Rx Packet Length	0 to 200	Used when the *Rx Termination Type* is **PACKET SIZE**. This parameter sets the length of data required to be received on the port before considering end-of-message reached.
Rx Timeout	0 to 65535	Used when the *Rx Termination Type* is **MESSAGE TIMEOUT**. This parameter sets the number of milliseconds to wait after the first character is received on the port before considering end-of-message reached.
Rx Delay	0 to 65535	Used when the *Rx Termination Type* is **INTERCHARACTER DELAY**. This parameter sets the maximum number of milliseconds to wait between each character received on the port. When this value is exceeded, the ASCII driver considers end-of-message reached.
Swap Rx Data Bytes	YES or NO	Specifies if byte swapping of received data will occur.
Tx Timeout	0 to 65535	Specifies the timeout period to transmit a message. Message transmission will be aborted if the message is not transmitted out the port within the specified timeout period.
Tx Minimum Delay	0 to 65535	Specifies the minimum number of milliseconds to delay before transmitting a message out the port. This pre-send delay is applied before the RTS on time. This may be required when communicating with slow devices.
Swap Tx Data Bytes	YES or NO	Specifies if byte swapping of transmitted data will occur.

8.3 ASCII Diagnostics

8.3.1 PCB Diagnostics

The best way to troubleshoot this driver is to use ProSoft Configuration Builder to access the diagnostic capabilities of the gateway through the Ethernet debug port. For instructions on how to access *Diagnostics*, see *Using Diagnostics in ProSoft Configuration Builder* (page 36).

8.3.2 ASCII Status Data in Upper Memory

Each ASCII port has an associated status data area located in the gateway's upper memory. The Data Map functionality of the gateway can be used to map this data into the normal user data range of the gateway's database. See *Using the CommonNet Data Map* (page 25).

The following table lists the starting addresses in upper memory where the status data for each ASCII port can be found.

Port	Gateway Starting Register for Status Data
1	13000
2*	13010
3*	13020
4*	13030

*Status data for Ports 2 through 4 are only present in 4-port ASCII gateways.

<u>Port [x] Status Data</u>

Status Register Offset	Description
0	Receive state: -1 = Listening for data 1 = Receiving port data
1	Total receive character count
2	Total receive message count
3	Transmit state: 0 = Waiting for Data to Send 1 = RTS On 2 = RTS Timeout 3 = Sending data 4 = Waiting for RTS Off 5 = RTS turned off 30, 31, and 32 = Intercharacter Delay 100 and 101 = Message Delay before Transmit
4	Total transmit character count
5	Total transmit message count
6	Reserved
7 to 9	No valid data

9 SIE Protocol

In This Chapter

This chapter contains information specific to PLX3x-series gateways with a Siemens Industrial Ethernet (SIE) protocol driver.

9.1 SIE Functional Overview

The PLX3x Siemens Industrial Ethernet (SIE) protocol can be used to interface easily with multiple Siemens Industrial Ethernet server-compatible instruments and devices. The SIE protocol improves performance when controlling multiple servers on a Siemens Industrial Ethernet network, by supporting up to 20 Clients.

The gateway’s Clients interface with processors (and other server-based devices) on the SIE network using a user-constructed command list of up to 16 entries per Client. The gateway’s internal database is used as the source for write commands to the remote processors. Data collected from the processors using read commands is placed in the gateway’s database.

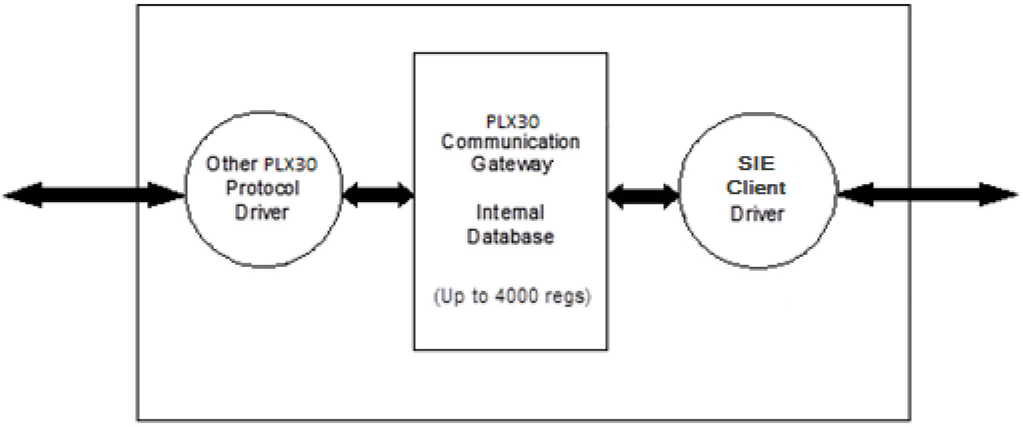
9.1.1 General Specifications

- 10/100 MB Ethernet Communication port
- Actively reads data from and writes data to Siemens Industrial Ethernet devices, using Siemens Industrial Ethernet protocol
- Siemens Industrial Ethernet data types overlap in the gateway's memory database, so the same data can be conveniently read or written as bit-level or register-level data
- Offers 20 Client connections with up to 16 commands each to talk to multiple servers
- Configurable floating-point data movement is supported
- Status and error information generated by the gateway

9.1.2 Gateway Internal Database

Central to the functionality of the gateway is the internal database. This database is shared between all the ports on the gateway and is used as a conduit to pass information from one device on one network to one or more devices on another network. This permits data from devices on one communication port to be viewed and controlled by devices on another communication port.

In addition to data from the Client, status and error information generated by the gateway can also be mapped into the internal database.



9.2 SIE Configuration

9.2.1 SIE Client x

This section defines general configuration for the SIE Client (Master).

Parameter	Value	Description
Minimum Command Delay	0 to 65535	Specifies the number of milliseconds to wait between the initial issuances of a command. This parameter can be used to delay all commands sent to servers to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.
Response Timeout	0 to 65535	Specifies the time in milliseconds that a Client will wait before re-transmitting a command if no response is received from the addressed server. The value to use depends on the type of communication network used, and the expected response time of the slowest device on the network.
Retry Count	0 to 65535	Specifies the number of times a command will be retried if it fails.

9.2.2 SIE Client x Commands

The *SIE Client x Commands* section of the configuration sets the Siemens Industrial Ethernet Client command list. This command list polls Siemens Industrial Ethernet server devices attached to the Siemens Industrial Ethernet Client port. The gateway supports numerous commands. This permits the module to interface with a wide variety of Siemens Industrial Ethernet protocol devices.

The function codes used for each command are those specified in the Siemens Industrial Ethernet protocol. Each command list record has the same format. The first part of the record contains the information relating to the gateway, and the second part contains information required to interface to the Siemens Industrial Ethernet server device.

Command List Overview

A command list needs to be constructed to interface the gateway with Siemens Industrial Ethernet server devices. The commands in the list specify the server device to be addressed, the function to be performed (read or write), the data area in the device to interface with, and the registers in the internal database to be associated with the device data. Each Client command list supports up to 16 commands.

The command list is processed from top (command #1) to bottom. A poll interval parameter is associated with each command to specify a minimum delay time in tenths of a second between the issuances of a command. If the user specifies a value of **10** for the parameter, the command will be executed no more frequently than every 1 second.

Commands Supported by the Gateway

The format of each command in the list depends on the Siemens Industrial Ethernet Function Code being executed.

The type of functions that will be supported will also depend on the server device and what it can support.

Below are examples of S7-200, S7-300 and S7-1200 functions that are supported by the gateway.

S7-300:

Data Block:

Address Type	**Function**	**Data Type**
DB	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	COUNT
	Write	COUNT

Timer:

Address Type	Function	Data Type
Timer	READ	TIME

Counter:

Address Type	Function	Data Type
Counter	READ	Count

Flag:

Address Type	Function	Data Type
Flag	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Output:

Address Type	Function	Data Type
Output	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Input:

Address Type	**Function**	**Data Type**
Input	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

S7-200:

Data Block:

Address Type	**Function**	**Data Type**
DB	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

Flag:

Address Type	**Function**	**Data Type**
Flag	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

Output:

Address Type	**Function**	**Data Type**
Output	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

Input:

Address Type	**Function**	**Data Type**
Input	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

S7-1200:

Data Block:

Address Type	Function	Data Type
DB	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	COUNT
	Write	COUNT

Flag:

Address Type	Function	Data Type
Flag	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Output:

Address Type	Function	Data Type
Output	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Input:

Address Type	Function	Data Type
Input	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Command List Parameters

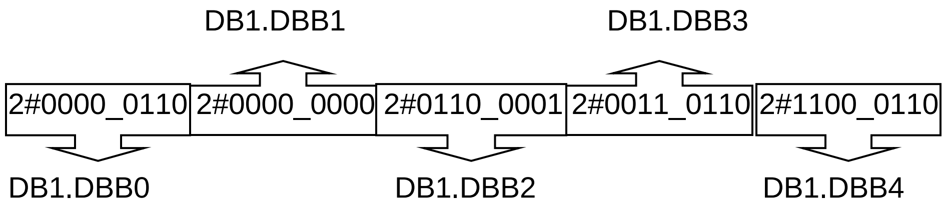
Parameter	Value	Description
Enable	DISABLED ENABLED CONDITIONAL	Specifies if the command is to be executed and under what conditions. **Disabled (0)** - the command is disabled and will not be executed in the normal polling sequence. **Continuous (1)** - the command will be executed upon each scan of the Command List if the Poll Interval is set to zero (0). If the *Poll Interval* is set to a non-zero value, the command will be executed when the interval timer for that command expires. **Conditional (2)** - the command will execute only if the internal data associated with the command changes.
Internal Address	0 to 3999 (for register-level addressing) or 0 to 63999 (for bit-level addressing)	Specifies the database address in the gateway's internal database to use as the destination for data brought in by a read command, or as the source for data to be sent out by a write command. The database address is interpreted as a bit address or a 16-bit register (word) address, depending on the Siemens Industrial Ethernet's Data Type used in the command. If Data Type – Bool is used in the command list, then the database address will be interpreted as a bit address. When any other data types are used, then the database address is interpreted as a 16-bit word (register) address.
Poll Interval	0 to 65535	Specifies the minimum interval between executions of continuous commands. The value is in tenths of a second. Therefore, if a value of 100 is entered, the command will execute no more frequently than once every 10 seconds.
Reg Count	Command-dependent	Specifies the number of 16-bit registers or binary bits to be transferred by the command. The range depends on the Siemens processor and the type of command. See Maximum Register Counts (page 188).
Swap Code	No Change Word Swap Word and Byte Swap Byte Swap	Specifies if and how the order of bytes in data received or sent is to be rearranged. This option exists to allow for the fact that different manufacturers store and transmit multi-byte data in different combinations. This parameter is helpful when dealing with floating-point or other multi-byte values, as there is no one standard method of storing these data types. The parameter can be set to rearrange the byte order of data received or sent into an order more useful or convenient for other applications. **No change (0)** - No change is made in the byte ordering (1234 = 1234) **Word Swap (1)** -The words are swapped (1234=3412) **Word and Byte Swap (2)** - The words are swapped, then the bytes in each word are swapped (1234=4321) **Byte Swap (3)** - The bytes in each word are swapped (1234=2143) These swap operations affect 4-byte (or 2-word) groups of data. Therefore, data swapping using these *Swap*

Parameter	Value	Description
		Codes should be done only when using an even number of words, such as when 32-bit integer or floating-point data is involved.
Node IP Address	xxx.xxx.xxx.xxx	IP address of the Siemens processor being addressed by the command.
PLC Type	Manual TSAP, S7-300/400/1200/1500	Type of Siemens processor being addressed by the command.
Rack	0 to 999	Rack number of the S7-300, S7-400 or S7-1200 CPU. **Note:** Rack number is not used for the S7-200 CPU.
Slot	0 to 12	Slot number of the S7-300, S7-400 or S7-1200 CPU. **Note:** Slot number is not used for the S7-200 CPU.
TSAP		TSAP of the S7-200 CPU. This can be found in the Siemens STEP 7 MicroWIN software. **Note:** TSAP is only needed for controllers that specifically need custom TSAP settings.
Func Type	Read Write	This parameter can either be **Read** or **Write**.
Data Type	BOOL BYTE DINT REAL INT TIME COUNT	This parameter can be **BOOL**, **BYTE**, **DINT**, **REAL**, **INT**, **TIME**, or **COUNT**.
Address Type	INPUT OUTPUT FLAG TIMER COUNTER DB	This parameter can be **INPUT**, **OUTPUT,FLAG**, **TIMER**, **COUNTER** or **DB** (Data Block).
DB Number	0 to 32767	Specifies the Data Block number to be used with the command. **Note:** DB Number is only used when the Address Type is set to DB.
Address		Address of the data in the Siemens device. For Read or Write operations using the INT, DINT, REAL or BYTE Data Types, the address is a byte address. For Read or Write operations using the BOOL Data Type, the address is a bit address. Please see below for notes on addressing.

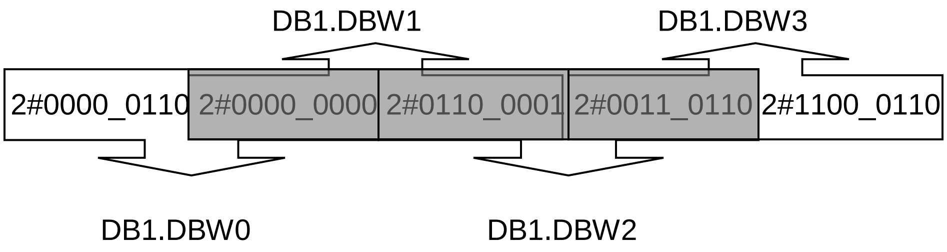
Notes on Addressing in S7 Processors

S7-300/S7-1200 Processor:

Byte Address in Data Block:



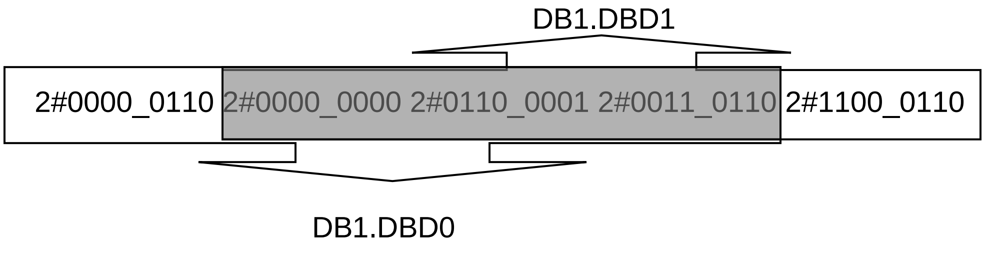
Word Address in Data Block:



The gray area above represents the byte memory locations being overlapped when word address is used consecutively (DB1.DBW0, DB1.DBW1, DB1.DBW2, etc).

If DB1.DBW0 is used as the first address in the Siemens processor, the next word address that can be used without overwriting the data would be DB1.DBW2.

Double Word Address in Data Block:



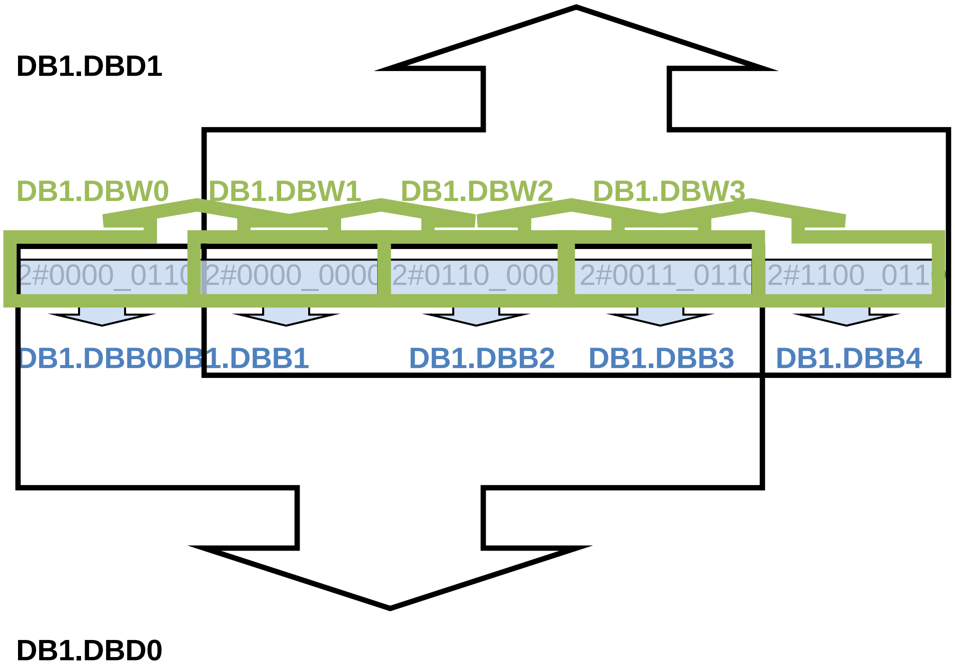
The gray area above represents the byte memory locations being overlapped when double word address is used consecutively (DB1.DBD0, DB1.DBD1, DB1.DBD2, etc).

If DB1.DBD0 is used as the first address in the Siemens processor, the next double word address that can be used without overwriting the data would be DB1.DBD4.

All of the above share the same memory locations in the processor.

Note: Incorrect memory location addressing can cause the data to be overwritten.

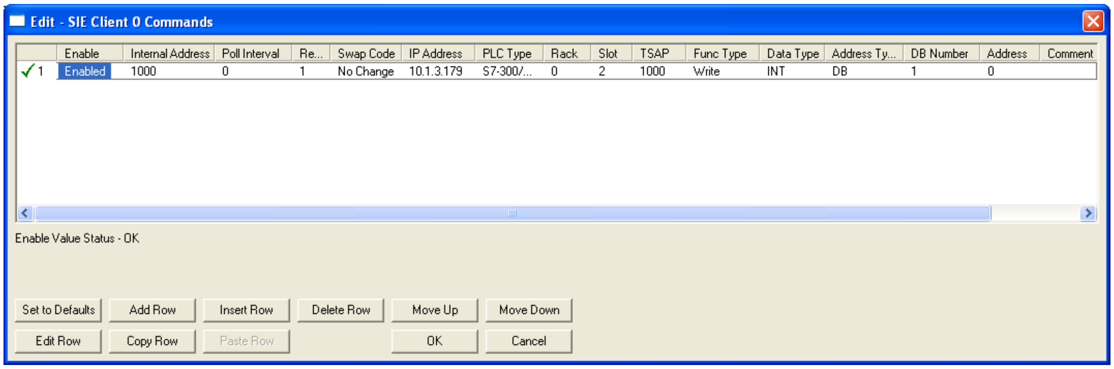
Below is a graphical representation of the addressing of the processor's memory locations.



Example:

Sending an integer value of 11733 from gateway register 1000 to a Siemens S7-300 processor demonstrates the addressing scheme in the Siemens S7-300 processor:

PCB screen shot showing Client command to send INT data to DB1 address 0:



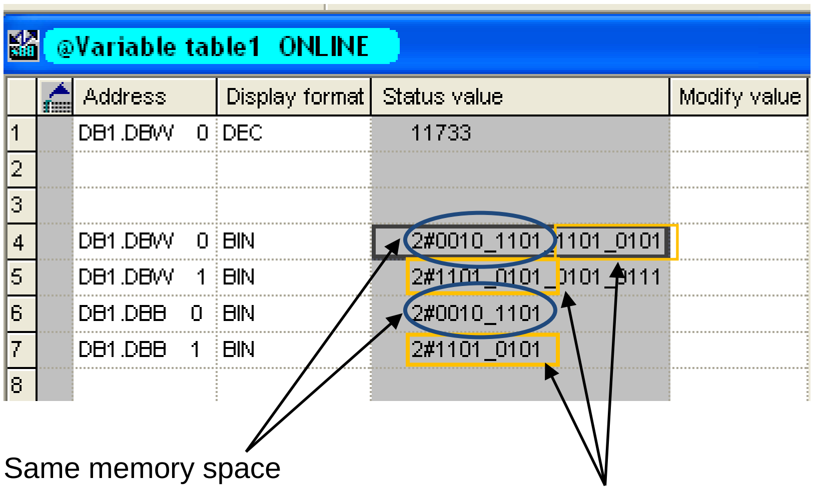
Edit - SIE Client 0 Commands

	Enable	Internal Address	Poll Interval	Re...	Swap Code	IP Address	PLC Type	Rack	Slot	TSAP	Func Type	Data Type	Address Ty...	DB Number	Address	Comment
✓ 1	Enabled	1000	0	1	No Change	10.1.3.179	S7-300/...	0	2	1000	Write	INT	DB	1	0	

Enable Value Status - OK

Row 1 in the SIMATICS Manager screen shot below shows the data transferred from the gateway.

If the data is broken up and displayed in binary format, it can be seen that the binary data stored in the first byte of DB1.DBW 0 is identical to that stored in byte address DB1.DBB 0. This is because the memory locations referenced by the first byte of DB1.DBW 0 and by DB1.DBB0 are one and the same, as explained previously.

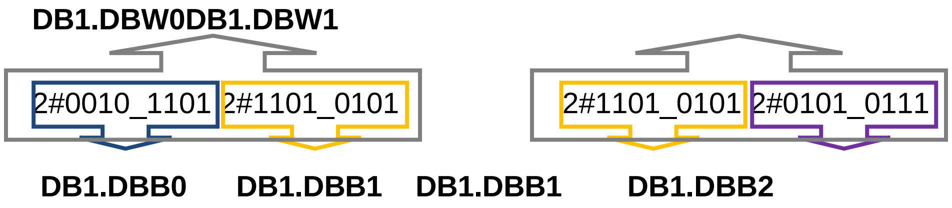


@Variable table1 ONLINE

	Address	Display format	Status value	Modify value
1	DB1.DBW 0	DEC	11733	
2				
3				
4	DB1.DBW 0	BIN	2#0010_1101_1101_0101	
5	DB1.DBW 1	BIN	2#1101_0101_0101_0111	
6	DB1.DBB 0	BIN	2#0010_1101	
7	DB1.DBB 1	BIN	2#1101_0101	
8				

Same memory space

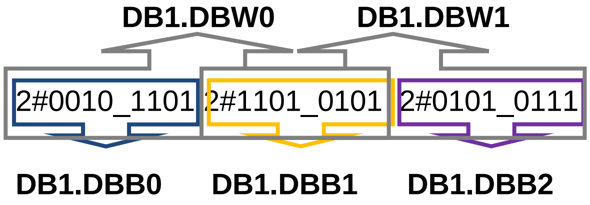
Same memory space



The first byte of DB1.DBW0 is the same as DB1.DBB0.

The second byte of DB1.DBW0 is the same as DB1.DBB1, and is the same as the first byte of DB1.DBW1.

In reality, the memory space looks like below:



Note: To access the first address of Data Block, Flag, Input, Output, Timer, and Counter memory locations in the S7-300 and S7-1200 processors, use the following address syntax.

Data Block -> DB1.DBB0, DB1.DBW0, DB1.DBD0

Flag -> MB0, MW0, MD0

Input -> IB0, IW0, ID0

Output -> QB0, QW0, QD0

Timers -> T0 – T65535

Counters -> C0 – C65535

Note: To access the first address of Data Block, Flag, Input and Output memory locations in the S7-200 processor, use the following address syntax.

Data Block -> VB0, VW0, VD0

Flag -> MB0, MW0, MD0

Input -> IB0, IW0, ID0

Output -> QB0, QW0, QD0

9.3 SIE Diagnostics

The best way to troubleshoot the SIE driver is to use ProSoft Configuration Builder to access the diagnostic capabilities of the gateway through the Ethernet debug port. For instructions on how to access *Diagnostics*, see *Using Diagnostics in ProSoft Configuration Builder* (page 36).

9.3.1 Client Command Errors

Each SIE Client has an associated status data area located in the gateway's upper memory. The Data Map functionality of the gateway can be used to map this data into the normal user data range of the gateway's database. See *Using the CommonNet Data Map* (page 25).

The following table lists the starting addresses in upper memory where the status data for each Client can be found.

Client	Gateway Starting Register for Status Data
0	21100
1	21125
2	21151
…	…
…	…
18	21568
19	21594

The following table describes the content of each Client's status data area.

Word Offset	Description
0	Command Request Count
1	Command Response Count
2	Command Error Count
3	Number of Request Packets
4	Number of Response Packets
5	Errors Sent
6	Errors Received
7	Configuration Error Word
8	Current Error
9	Last Error
10 to 25	Command List Errors (16 per Client)

For every command that has an error, the gateway automatically sets the poll delay parameter to 30 seconds. This instructs the gateway to wait 30 seconds before it attempts to issue the command again.

As the commands in the Client Command List are polled and executed, an error codeis maintained in the gateway for each command. The current error value for each command is displayed in the Command List Errors section of each Client's status data area. There is one register for each of the 16 commands in the command list. An error code of 0 means no error is currently detected for the specified command.
The following tables list the various error codes that may be displayed and their descriptions.

9.3.2 SIE Error Codes

Module Communication Error Codes

Decimal	**Hex**	**Description**
1	0x0001	No data from I/O module
3	0x0003	The desired item is not available in the PLC (200 family)
5	0x0005	The desired address is beyond limit for this PLC
6	0x0006	The CPU does not support reading a bit block of length<>1
7	0x0007	Write data size error
10	0x000a	The desired item is not available in the PLC
-123	0xff85	Cannot evaluate the received PDU
-124	0xff84	The PLC returned a packet with no result data
-125	0xff83	The PLC returned an error code not understood by this library
-126	0xff82	This result contains no data
-127	0xff81	Cannot work with an undefined result set
-128	0xff80	Unexpected function code in answer";
-129	0xff7f	PLC responds with an unknown data type
-1024	0xfc00	Short packet from PLC
-1025	0xfbff	Timeout when waiting for PLC response
-32767	0x8001	Not allowed in current operating status
-32511	0x8101	Hardware fault
-32509	0x8103	Object access not allowed
-32508	0x8104	Context is not supported. Step7 says: Function not implemented or error in telegram

Decimal	Hex	Description
-32507	0x8105	Invalid address
-32506	0x8106	Data type not supported
-32505	0x8107	Data type not consistent
-32502	0x810A	Object does not exist
-31999	0x8301	Insufficient CPU memory
-31742	0x8402	CPU already in RUN or already in STOP
-31740	0x8404	Severe error
-31488	0x8500	Incorrect PDU size
-30974	0x8702	Address invalid
-12286	0xd002	Step7: Variant of command is illegal
-12284	0xd004	Step7: Status for this command is illegal
-12127	0xd0A1	Step7: Function is not allowed in the current protection level
-11775	0xd201	Block name syntax error
-11774	0xd202	Syntax error function parameter
-11773	0xd203	Syntax error block type
-11772	0xd204	No linked block in storage medium
-11771	0xd205	Object already exists
-11770	0xd206	Object already exists
-11769	0xd207	Block exists in EPROM
-11767	0xd209	Block does not exist/could not be found
-11762	0xd20e	No block present
-11760	0xd210	Block number too big
-11712	0xd240	Coordination rules were violated
-11711	0xd241	Operation not permitted in current protection level
-11710	0xd242	Protection violation while processing F-blocks. F-blocks can only be processed after password input
-11263	0xd401	Invalid SZL ID
-11262	0xd402	Invalid SZL index
-11258	0xd406	Diagnosis: Info not available
-11255	0xd409	Diagnosis: DP error
-9215	0xdc01	Invalid BCD code or Invalid time format

SIE Client Specific Error Codes

Decimal	Hex	Description
-33	0xffdf	Failed to connect to server specified in command
-34	0xffde	Failed to create a socket
-36	0xffdc	SIE command response timeout (same as -11)
-37	0xffdb	TCP/IP connection ended before session finished

Command List Entry Error Codes

Decimal	Hex	Description
-40	0xffd8	Too few parameters
-41	0xffd7	Invalid enable code
-42	0xffd6	Internal address > maximum address
-44	0xffd4	Count parameter set to 0
-45	0xffd3	Invalid function code
-46	0xffd2	Invalid swap code
-47	0xffd1	Invalid TSAP code

9.4 SIE Reference

9.4.1 Maximum Register Counts

CPU315-2 DP

Data Block:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
DB	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	164	
	Write	BYTE		164
	READ	DINT	41	
	Write	DINT		41
	READ	REAL	41	
	Write	REAL		41
	READ	INT	82	
	Write	INT		82
	READ	TIME	82	
	Write	TIME		41
	READ	COUNT	82	
	Write	COUNT		82

Timer:

Address Type	Function	Data Type	Max Reg Cnt
Timer	READ	TIME	1

Counter:

Address Type	Function	Data Type	Max Reg Cnt
Counter	READ	Count	111

Flag:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Flag	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	222	
	Write	BYTE		212
	READ	DINT	55	
	Write	DINT		53
	READ	REAL	55	
	Write	REAL		53
	READ	INT	111	
	Write	INT		106
	READ	TIME	111	
	Write	TIME		53
	READ	Count	111	
	Write	Count		106

Output:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Output	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	128	
	Write	BYTE		128
	READ	DINT	32	
	Write	DINT		32
	READ	REAL	32	
	Write	REAL		32
	READ	INT	64	
	Write	INT		64
	READ	TIME	64	
	Write	TIME		32
	READ	Count	64	
	Write	Count		64

Input:

Address Type	**Function**	**Data Type**	**Max Reg Cnt**	**Max Reg Cnt**
Input	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	128	
	Write	BYTE		128
	READ	DINT	32	
	Write	DINT		32
	READ	REAL	32	
	Write	REAL		32
	READ	INT	64	
	Write	INT		64
	READ	TIME	64	
	Write	TIME		32
	READ	Count	64	
	Write	Count		64

CPU1212C:

Data Block:

Address Type	Function	Data Type	Max Read	Max Write
DB	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	30	
	Write	BYTE		30
	READ	DINT	7	
	Write	DINT		7
	READ	REAL	7	
	Write	REAL		7
	READ	INT	15	
	Write	INT		15
	READ	TIME	15	
	Write	TIME		15
	READ	COUNT	15	
	Write	COUNT		15

Flag:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Flag	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	212	
	Write	BYTE		212
	READ	DINT	53	
	Write	DINT		53
	READ	REAL	53	
	Write	REAL		53
	READ	INT	106	
	Write	INT		106
	READ	TIME	105	
	Write	TIME		105
	READ	Count	106	
	Write	Count		106

Output:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Output	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	212	
	Write	BYTE		212
	READ	DINT	53	
	Write	DINT		53
	READ	REAL	53	
	Write	REAL		53
	READ	INT	106	
	Write	INT		106
	READ	TIME	105	
	Write	TIME		105
	READ	Count	111	
	Write	Count		106

Input:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Input	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	222	
	Write	BYTE		212
	READ	DINT	55	
	Write	DINT		53
	READ	REAL	55	
	Write	REAL		53
	READ	INT	111	
	Write	INT		111
	READ	TIME	111	
	Write	TIME		106
	READ	Count	111	
	Write	Count		106

CPU224XP:

Data Block:

Address Type	Function	Data Type	Max Read	Max Write
DB	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	222	
	Write	BYTE		212
	READ	DINT	55	
	Write	DINT		53
	READ	REAL	55	
	Write	REAL		53
	READ	INT	111	
	Write	INT		106

Flag:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Flag	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	32	
	Write	BYTE		32
	READ	DINT	8	
	Write	DINT		8
	READ	REAL	8	
	Write	REAL		8
	READ	INT	16	
	Write	INT		16

Output:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Output	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	16	
	Write	BYTE		16
	READ	DINT	4	
	Write	DINT		4
	READ	REAL	4	
	Write	REAL		4
	READ	INT	8	
	Write	INT		8

Input:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Input	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	16	
	Write	BYTE		16
	READ	DINT	4	
	Write	DINT		4
	READ	REAL	4	
	Write	REAL		4
	READ	INT	8	
	Write	INT		8

10 PND Protocol

In This Chapter

This chapter contains information specific to the PLX3x-series gateway PND protocol driver.

10.1 PND Functional Overview

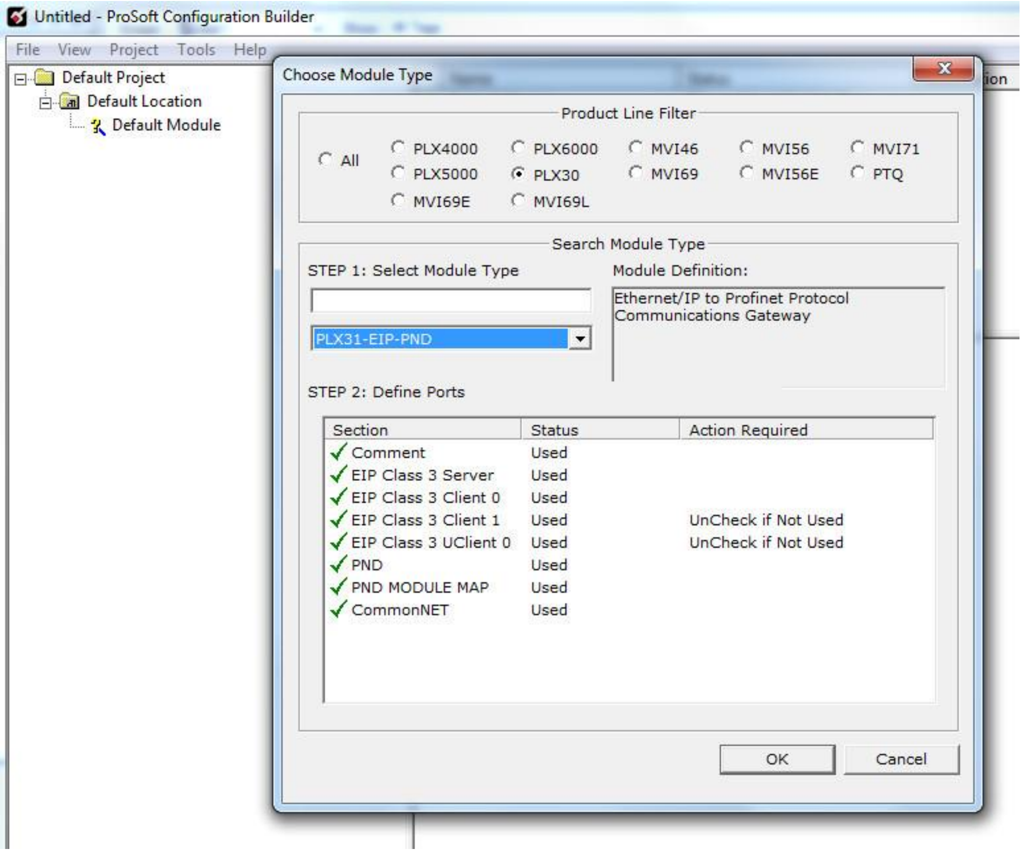
PROFINET is the communication standard for automation of PROFIBUS and PROFINET International (PI). Many years of experience with PROFIBUS and the widespread use of Industrial Ethernet are all rolled into PROFINET.

With its integrated, Ethernet-based communication, PROFINET satisfies a wide range of requirements, from data intensive parameter assignments to extremely fast I/O data transmission which enables automation in real-time.

10.2 PND Configuration

The following procedure provides the required steps to configure the PLX31-EIP-PND module using ProSoft Configuration Builder (PCB). This configuration provides the ability to communicate with a Siemens processor via the PROFINET protocol.

1. Launch ProSoft Configuration Builder.
2. Create a new PLX31-EIP-PND module.



STEP 1: Select Module Type: PLX31-EIP-PND

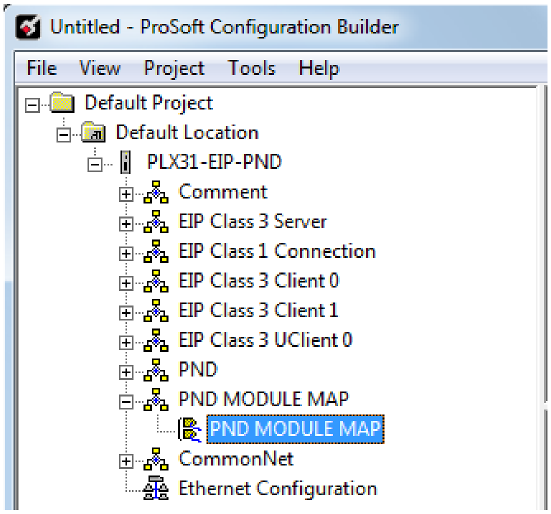
Module Definition: Ethernet/IP to Profinet Protocol Communications Gateway

STEP 2: Define Ports

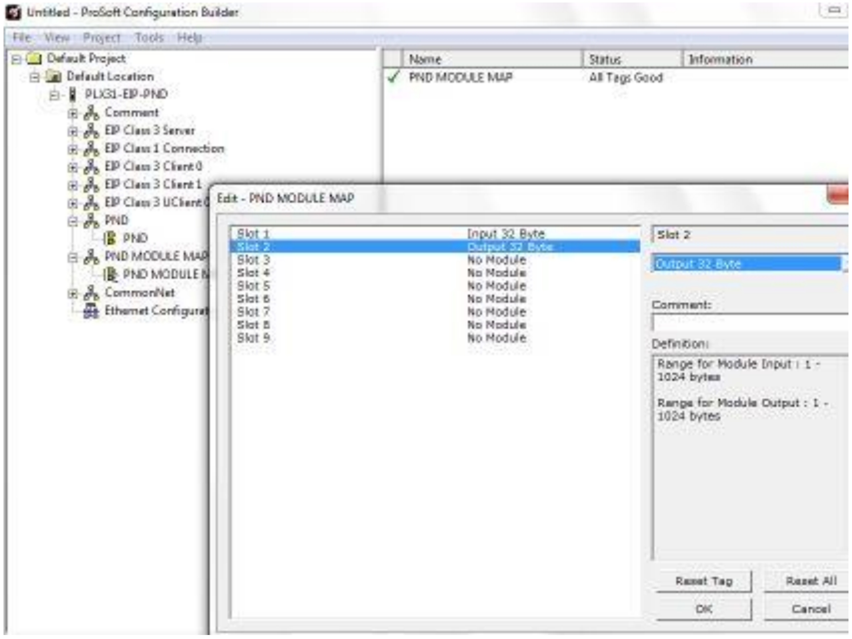
Section	Status	Action Required
✓ Comment	Used	
✓ EIP Class 3 Server	Used	
✓ EIP Class 3 Client 0	Used	
✓ EIP Class 3 Client 1	Used	UnCheck if Not Used
✓ EIP Class 3 UClient 0	Used	UnCheck if Not Used
✓ PND	Used	
✓ PND MODULE MAP	Used	
✓ CommonNET	Used	

3. Click **OK**.

4. Expand *PND MODULE MAP* and double-click **PND MODULE MAP** to edit.



5. In this example, 32-bytes of input and 32-bytes of output will be used (a drop-down list provides the range that could be used in each slot).



Name	Status	Information
✓ PND MODULE MAP	All Tags Good	

Slot 1	Input 32 Byte
Slot 2	Output 32 Byte
Slot 3	No Module
Slot 4	No Module
Slot 5	No Module
Slot 6	No Module
Slot 7	No Module
Slot 8	No Module
Slot 9	No Module

Slot 2

Output 32 Byte

Comment:

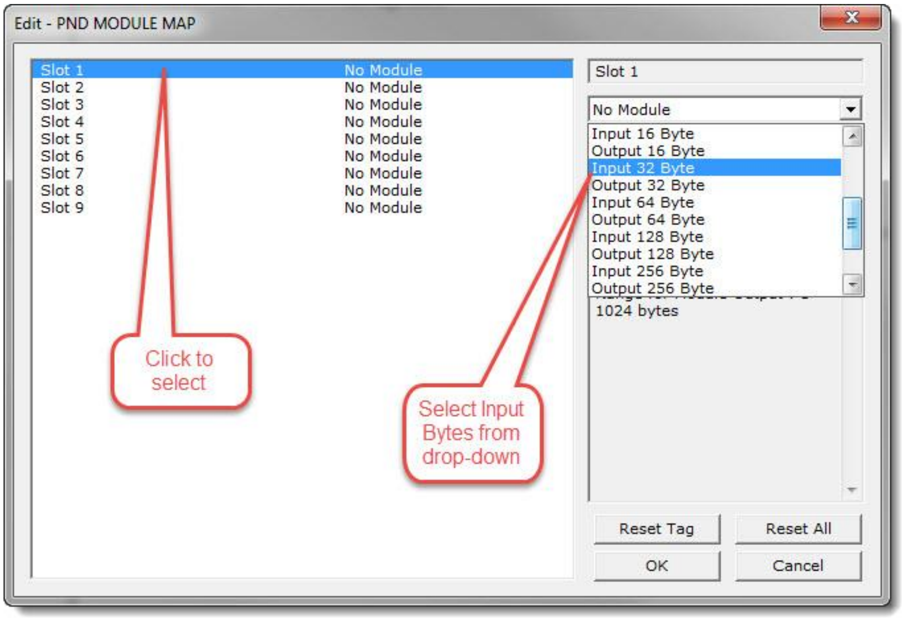
Definition:

Range for Module Input : 1 - 1024 bytes

Range for Module Output : 1 - 1024 bytes

6. Select **Slot 1**.

7. Select the number of input bytes. In this example, select **Input 32 Byte** from the drop down list.



Slot 1	No Module
Slot 2	No Module
Slot 3	No Module
Slot 4	No Module
Slot 5	No Module
Slot 6	No Module
Slot 7	No Module
Slot 8	No Module
Slot 9	No Module

Slot 1

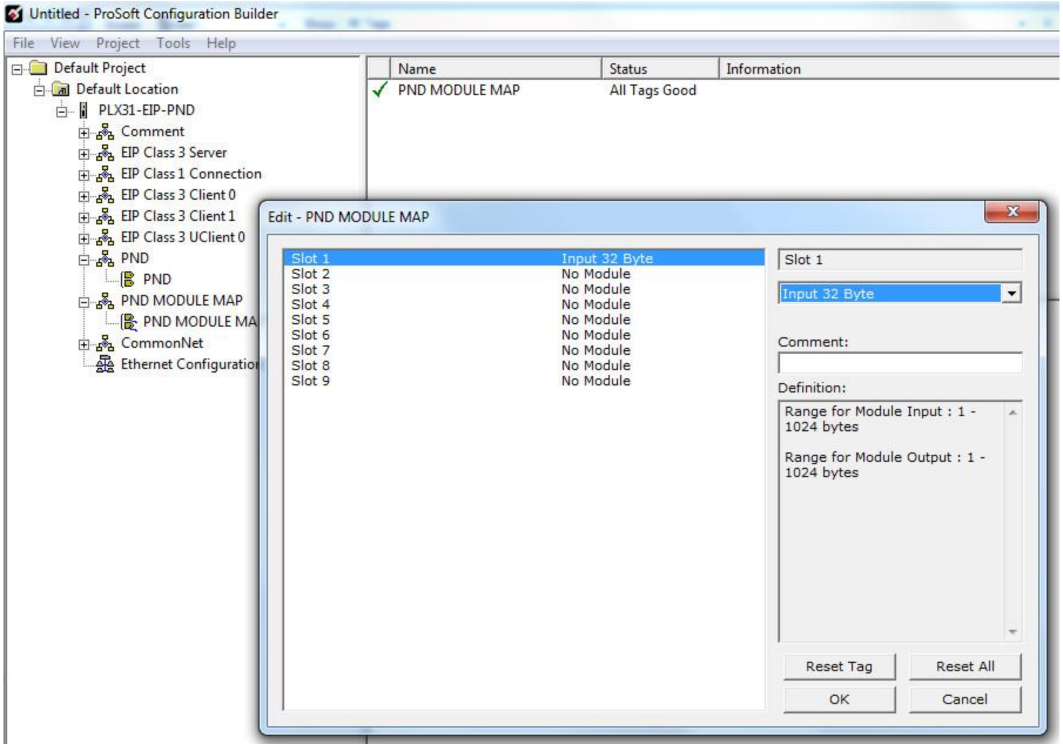
No Module

- Input 16 Byte
- Output 16 Byte
- Input 32 Byte
- Output 32 Byte
- Input 64 Byte
- Output 64 Byte
- Input 128 Byte
- Output 128 Byte
- Input 256 Byte
- Output 256 Byte

1024 bytes

8. Select **Slot 2**.

9. Select the number of output bytes. In this example, select **Output 32 Byte** from the drop down list.



Name	Status	Information
✓ PND MODULE MAP	All Tags Good	

Slot 1	Input 32 Byte
Slot 2	No Module
Slot 3	No Module
Slot 4	No Module
Slot 5	No Module
Slot 6	No Module
Slot 7	No Module
Slot 8	No Module
Slot 9	No Module

Slot 1

Input 32 Byte

Comment:

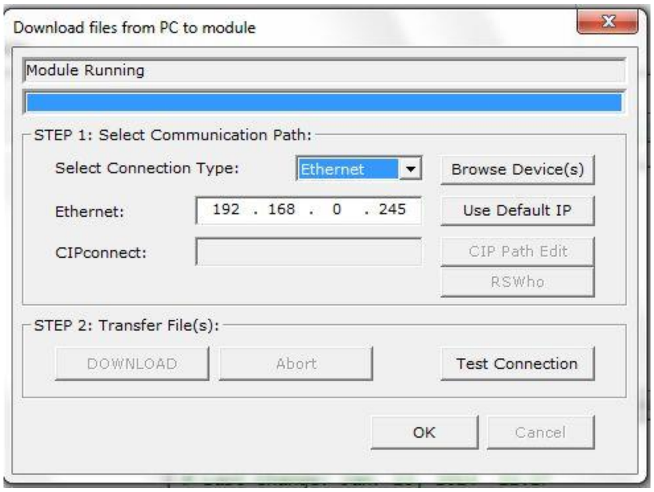
Definition:

Range for Module Input : 1 - 1024 bytes

Range for Module Output : 1 - 1024 bytes

Important: The PND module map is very important because the values entered here must match the values that are entered in the Siemens processor in order for them to communicate. The total combined Input bytes and total combined Output bytes must not be greater than 1440.

10. Save and download the configuration to the module.



Download files from PC to module

Module Running

STEP 1: Select Communication Path:

Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 245

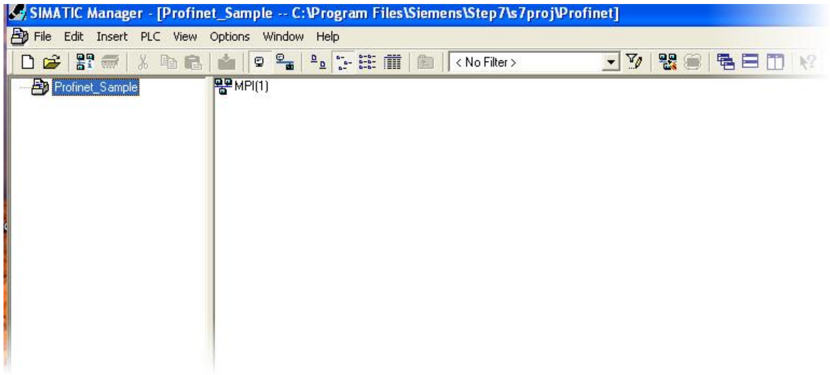
CIPconnect:

STEP 2: Transfer File(s):

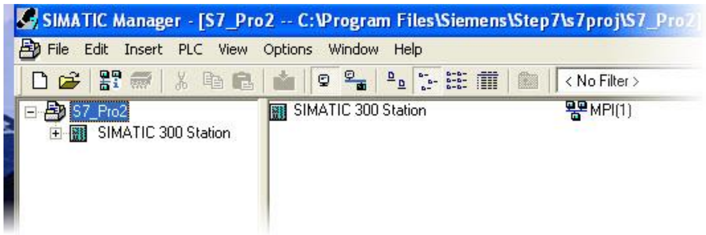
10.3 Step 7 Configuration

This procedure describes the tasks required to set up the module with Siemens Step 7. It is important to note that the configuration parameters used during PCB Configuration and download to the module must match parameters configured here within this procedure.

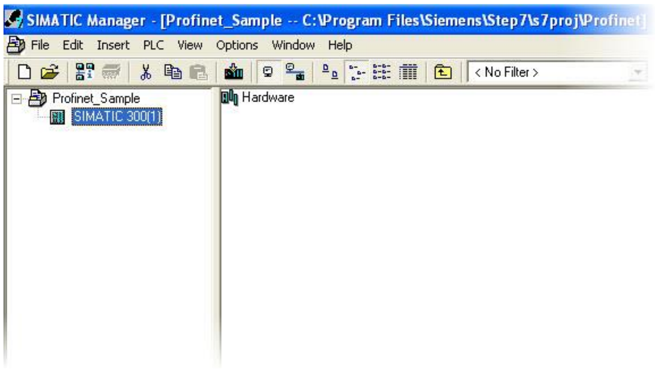
1 Start Step 7 software.
2 Create a new project. For example, Profinet_Sample.


SIMATIC Manager - [Profinet_Sample -- C:\Program Files\Siemens\Step7\s7proj\Profinet]
File Edit Insert PLC View Options Window Help
< No Filter >
Profinet_Sample
MPI(1)

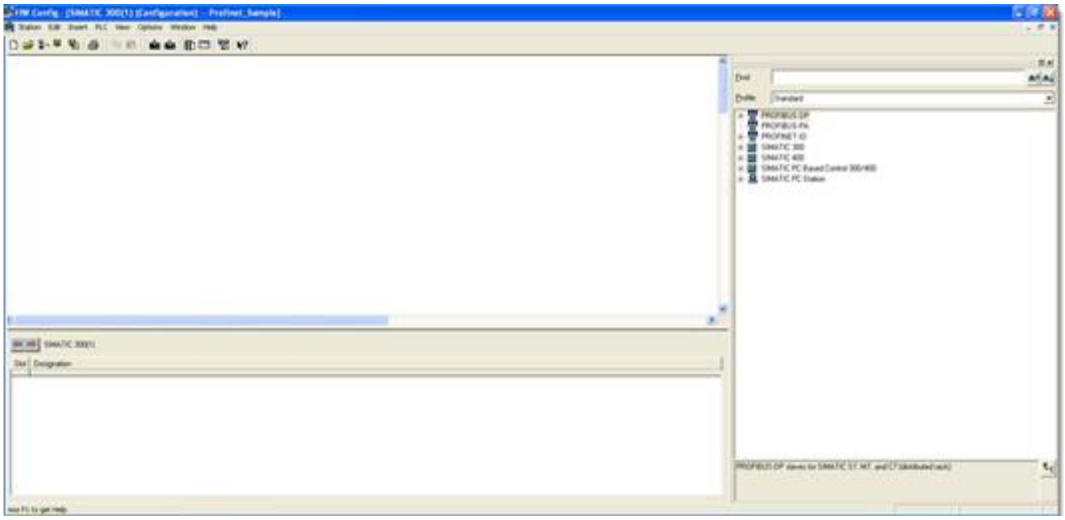
3 Add a station to the project in which the network will be configured.


SIMATIC Manager - [S7_Pro2 -- C:\Program Files\Siemens\Step7\s7proj\S7_Pro2]
File Edit Insert PLC View Options Window Help
< No Filter >
S7_Pro2
SIMATIC 300 Station
SIMATIC 300 Station
MPI(1)

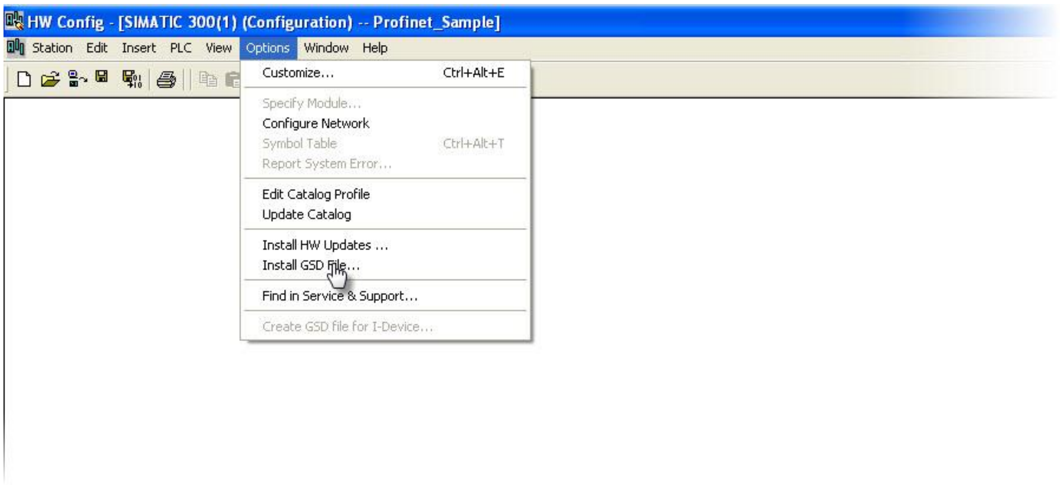
4 Click on the **SIMATIC 300 station**.
5 Configure the station by double-clicking the hardware icon under SIMATIC 300.


SIMATIC Manager - [Profinet_Sample -- C:\Program Files\Siemens\Step7\s7proj\Profinet]
File Edit Insert PLC View Options Window Help
< No Filter >
Profinet_Sample
SIMATIC 300(1)
Hardware

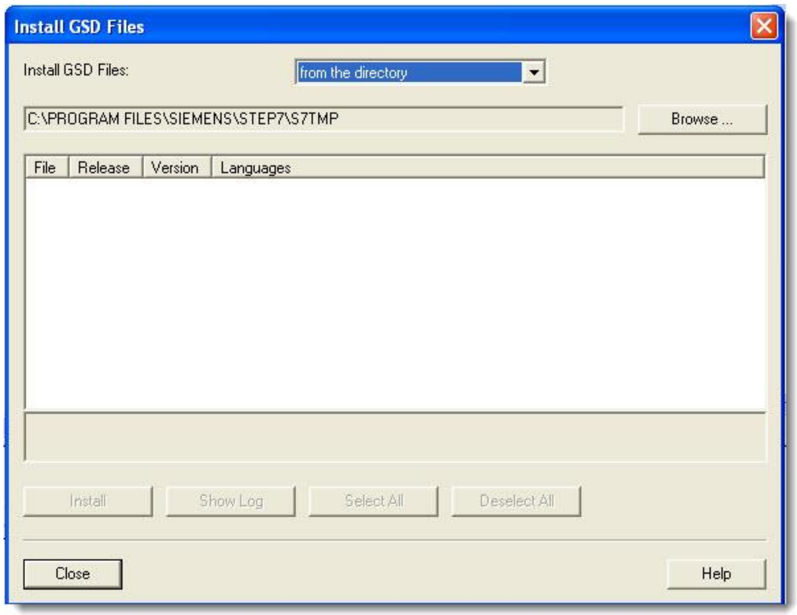
The *HW Config* window opens.



6 Click on **Options > Install GSD** file.



The Install GSD File dialog opens.

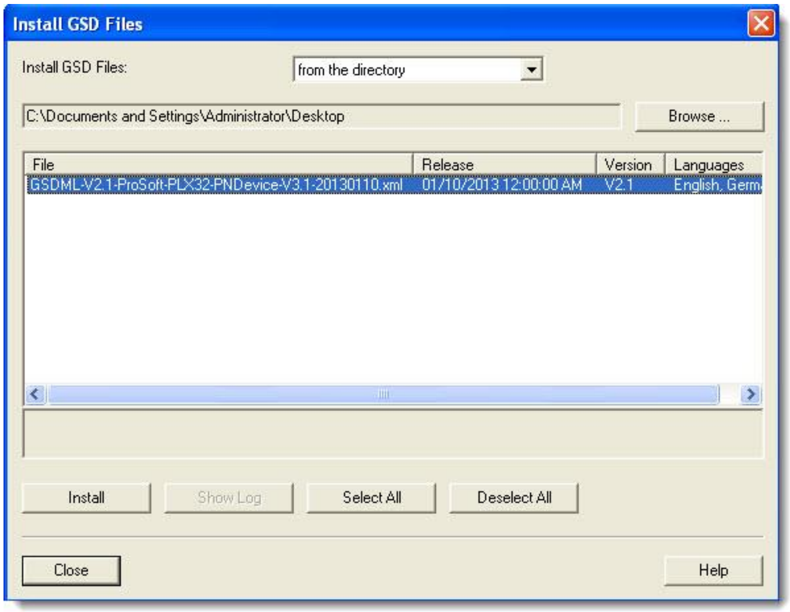


Install GSD Files: from the directory

C:\PROGRAM FILES\SIEMENS\STEP7\S7TMP

File	Release	Version	Languages

7 Click **Browse** and locate the GSD file.

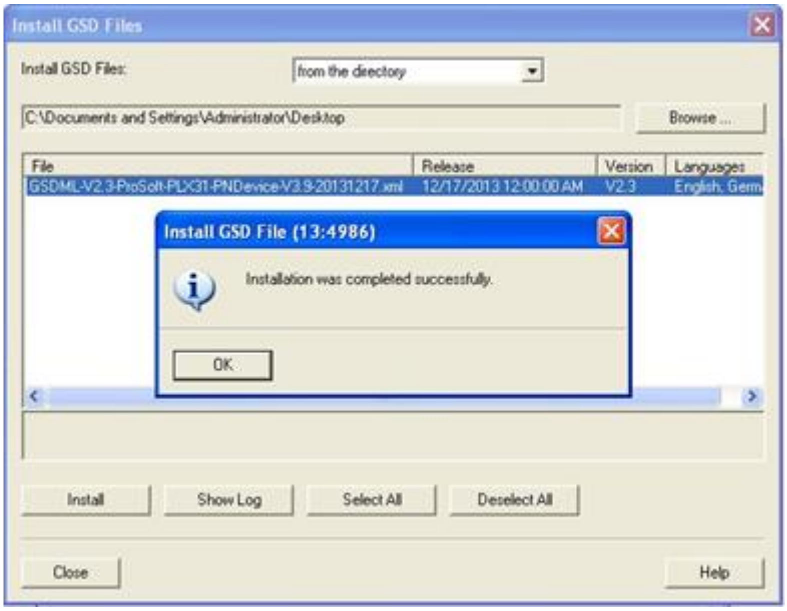


Install GSD Files: from the directory

C:\Documents and Settings\Administrator\Desktop

File	Release	Version	Languages
GSDML-V2.1-ProSoft-PLX32-PNDevice-V3.1-20130110.xml	01/10/2013 12:00:00 AM	V2.1	English, Germ

8 Select the file and click the **Install** button. The system informs you when the install is complete.



Install GSD Files

Install GSD Files: from the directory

C:\Documents and Settings\Administrator\Desktop

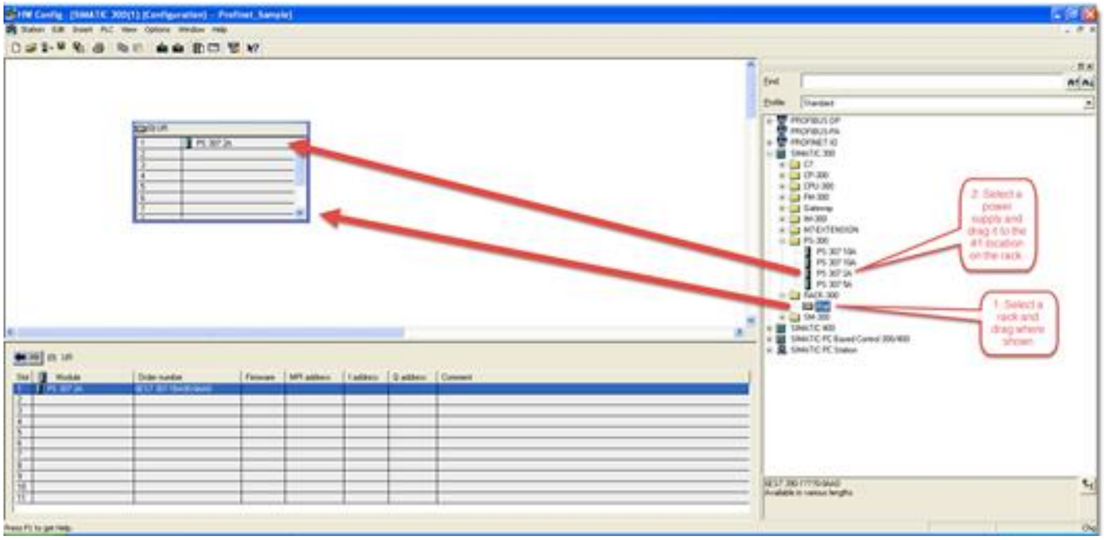
File	Release	Version	Languages
GSDML-V2.3-ProSoft-PLX31-PNDevice-V3.9-20131217.xml	12/17/2013 12:00:00 AM	V2.3	English, Germ

Install GSD File (13:4986)

Installation was completed successfully.

9 Click **OK**.

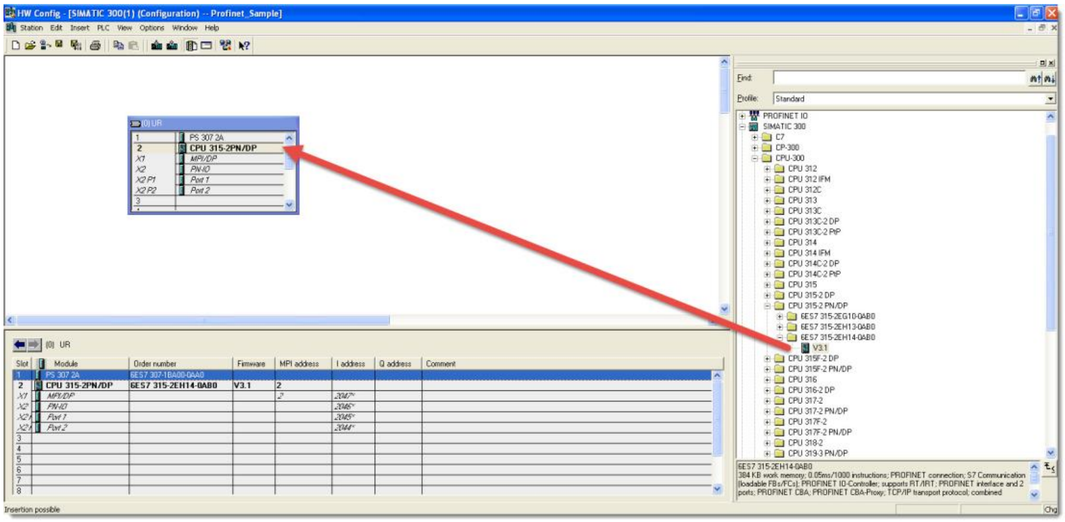
10 From the *HW Config* page, select a rack and a power supply.



1. Select a rack and drag where shown

2. Select a power supply and drag it to the #1 location on the rack

11 Drag and drop the processor to be used in the network. The example uses 315-2 PN/DP.



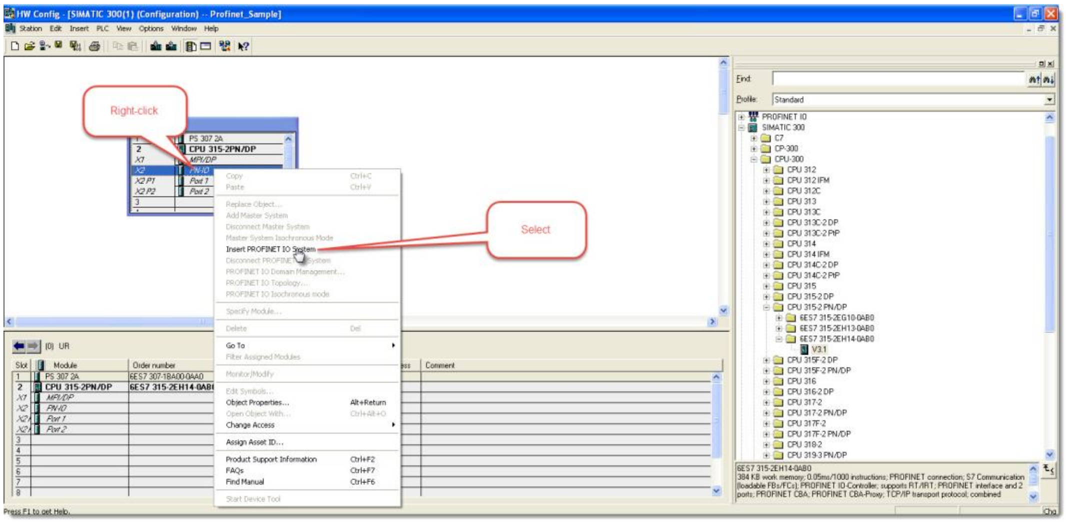
(0) UR

Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1	PS 307 2A	6ES7 307-1BA00-0AA0					
2	CPU 315-2PN/DP	6ES7 315-2EH14-0AB0	V3.1	2			
X1	MPI/DP			2	2047*		
X2	PN-IO				2046*		
X2	Port 1				2045*		
X2	Port 2				2044*		
3							
4							
5							
6							
7							
8							

6ES7 315-2EH14-0AB0
384 KB work memory; 0.05ms/1000 instructions; PROFINET connection; S7 Communication (loadable FBs/FCs); PROFINET IO-Controller; supports RT/IRT; PROFINET interface and 2 ports; PROFINET CBA; PROFINET CBA-Proxy; TCP/IP transport protocol; combined

Insertion possible

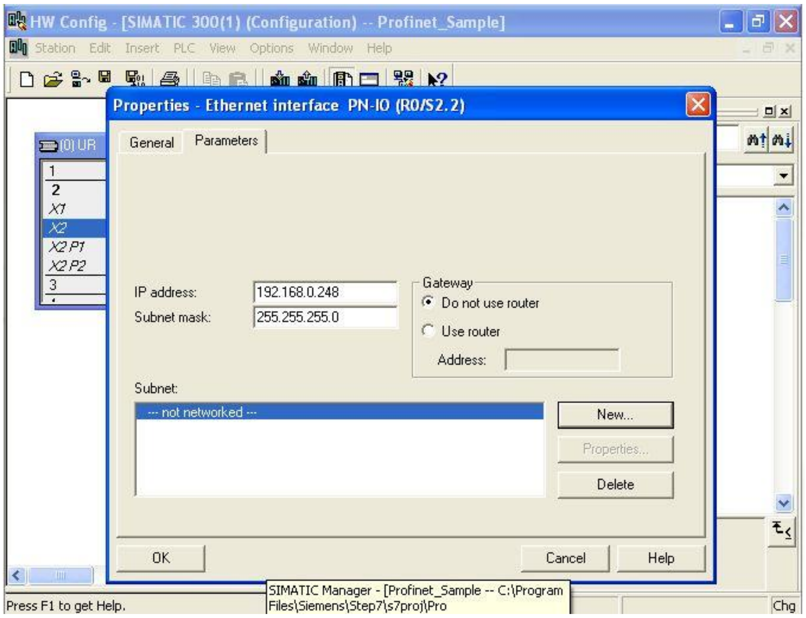
12 Right-click on the **PN-IO** option and select **Insert PROFINET IO System**.



6ES7 315-2EH14-0AB0
384 KB work memory; 0.05ms/1000 instructions; PROFINET connection; S7 Communication (loadable FBs/FCs); PROFINET IO-Controller; supports RT/IRT; PROFINET interface and 2 ports; PROFINET CBA; PROFINET CBA-Proxy; TCP/IP transport protocol; combined

Press F1 to get Help.

13 Insert the ProfiNet bus network where the ProSoft module will be added by clicking on the **PN-IO** option and select **New**. The *Properties - Ethernet Interface* dialog opens.


HW Config - [SIMATIC 300(1) (Configuration) -- Profinet_Sample]
Station Edit Insert PLC View Options Window Help
(0) UR
1
2
X1
X2
X2 P1
X2 P2
3
Properties - Ethernet interface PN-IO (R0/S2.2)
General
Parameters
New...
Properties...
Delete
OK
Cancel
Help
Chg

IP address: 192.168.0.248

Subnet mask: 255.255.255.0

Gateway

● Do not use router

○ Use router

Address:

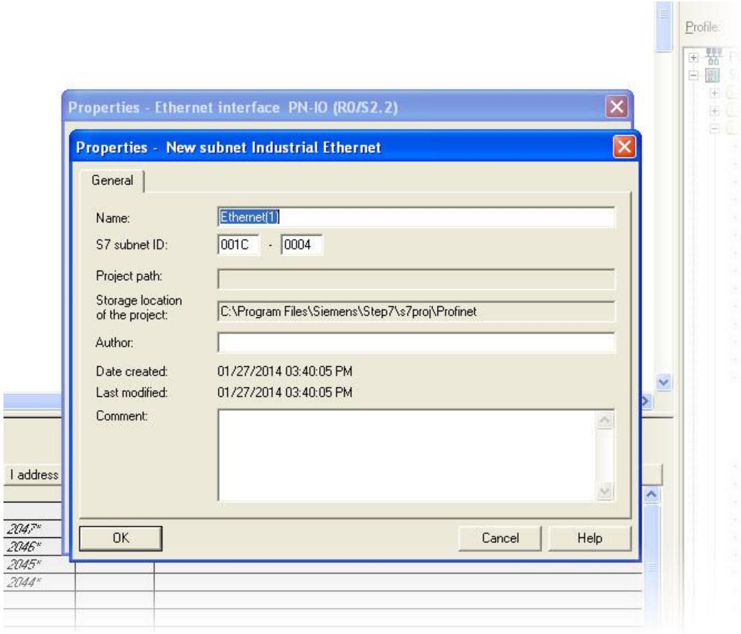
Subnet:

--- not networked ---

SIMATIC Manager - [Profinet_Sample -- C:\Program Files\Siemens\Step7\s7proj\Pro

Press F1 to get Help.

14 Click the **New** button. The *Properties - New subnet Industrial Ethernet* dialog opens.



Properties - Ethernet interface PN-IO (R0/S2.2)

Properties - New subnet Industrial Ethernet

Name: Ethernet(1)

S7 subnet ID: 001C - 0004

Project path:

Storage location of the project: C:\Program Files\Siemens\Step7\s7proj\Profinet

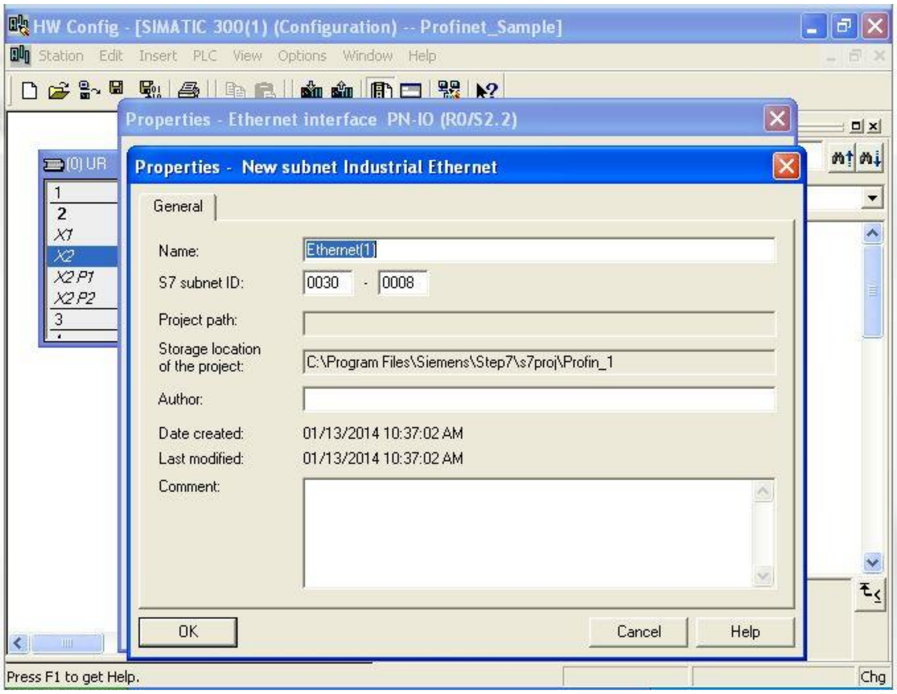
Author:

Date created: 01/27/2014 03:40:05 PM

Last modified: 01/27/2014 03:40:05 PM

Comment:

15 Click **OK** on the following prompt.



HW Config - [SIMATIC 300(1) (Configuration) -- Profinet_Sample]

Properties - Ethernet interface PN-IO (R0/S2.2)

Properties - New subnet Industrial Ethernet

Name: Ethernet(1)

S7 subnet ID: 0030 - 0008

Project path:

Storage location of the project: C:\Program Files\Siemens\Step7\s7proj\Profin_1

Author:

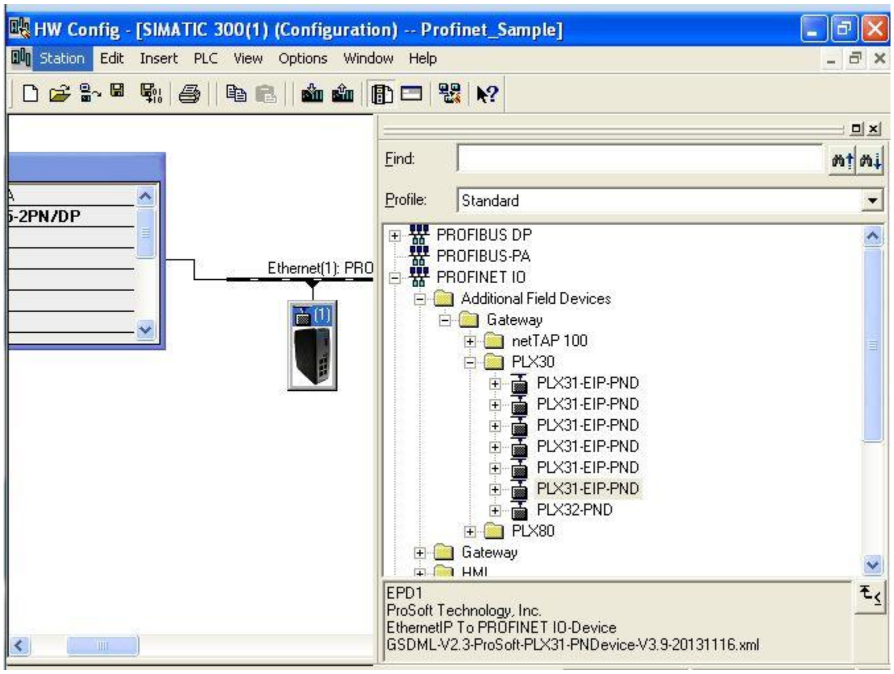
Date created: 01/13/2014 10:37:02 AM

Last modified: 01/13/2014 10:37:02 AM

Comment:

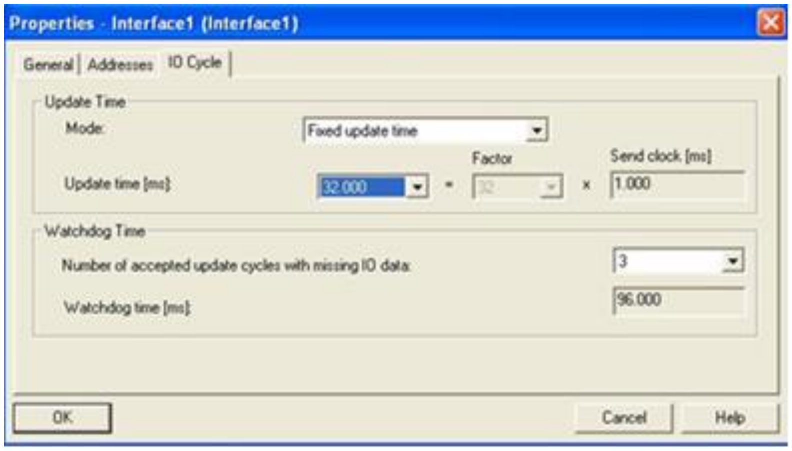
Press F1 to get Help.

16 Add the PLX31-EIP-PND module to the network.



EPD1
ProSoft Technology, Inc.
EthernetIP To PROFINET IO-Device
GSDML-V2.3-ProSoft-PLX31-PNDevice-V3.9-20131116.xml

17 Configure the interface port for the ProSoft module as shown.



Update Time

Mode: Fixed update time

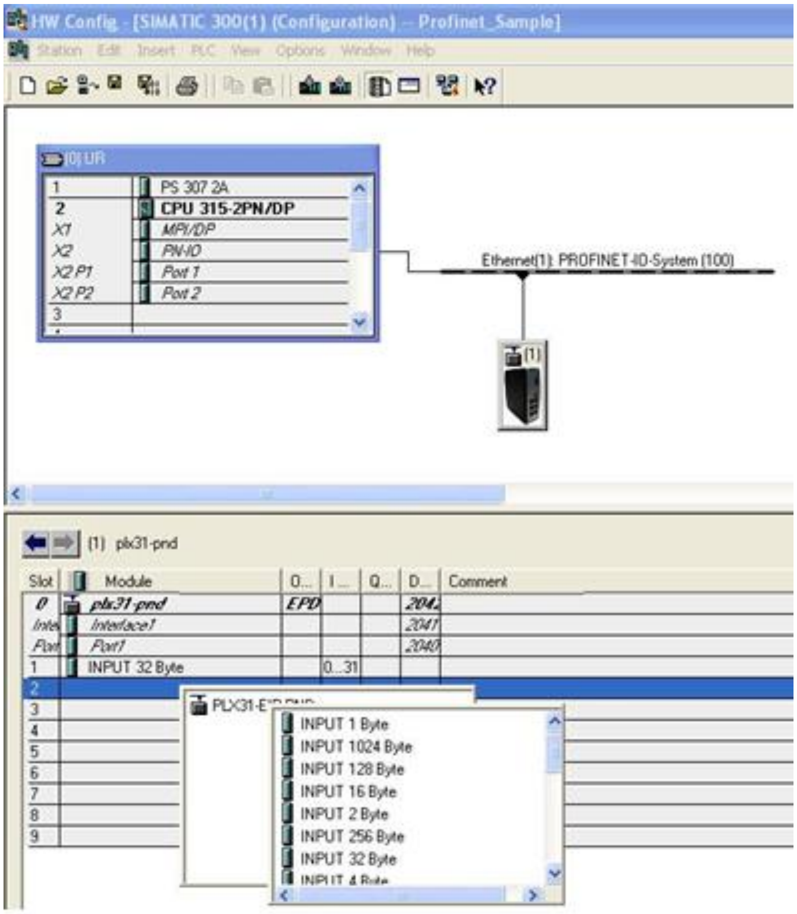
Update time [ms]: 32.000 = Factor 32 x Send clock [ms] 1.000

Watchdog Time

Number of accepted update cycles with missing IO data: 3

Watchdog time [ms]: 96.000

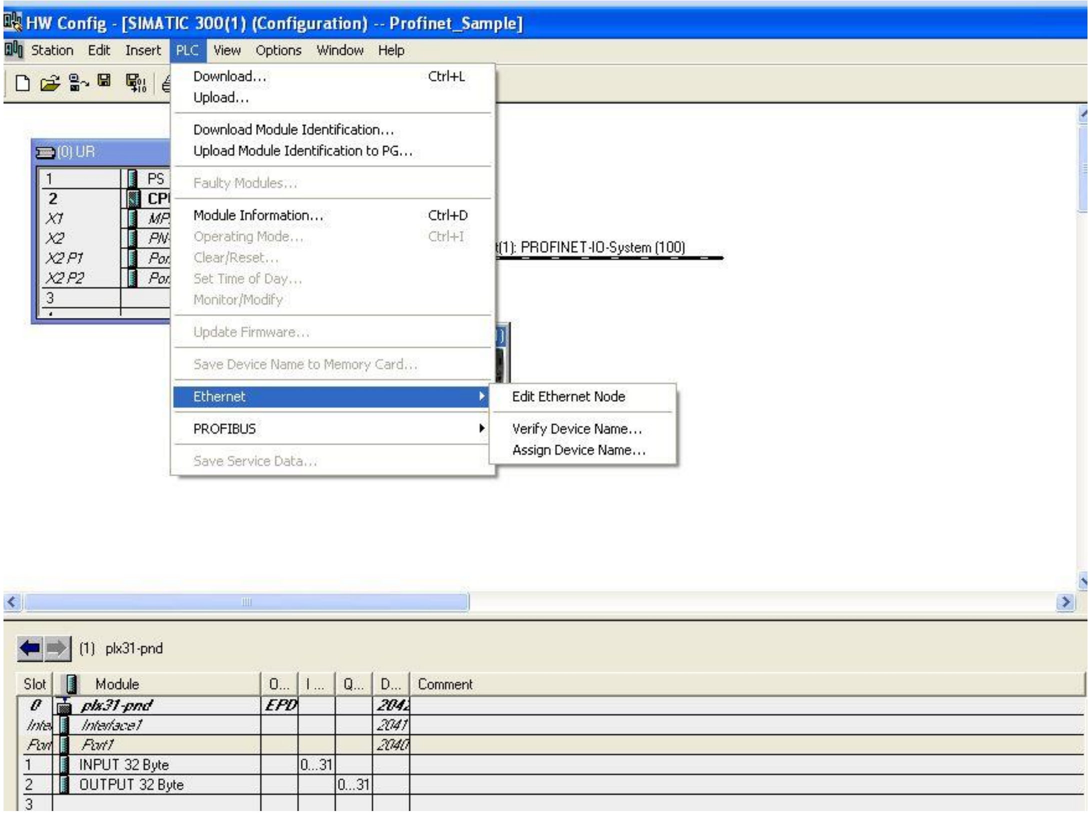
18 Configure the inputs and outputs to match the PCB configuration. The example uses 32 inputs and 32 outputs.


HW Config - [SIMATIC 300(1) (Configuration) -- Profinet_Sample]
Station Edit Insert PLC View Options Window Help
(0) UR
Ethernet(1): PROFINET-IO-System (100)
(1)
(1) plx31-pnd
PLX31-EIP-PND
INPUT 1 Byte
INPUT 1024 Byte
INPUT 128 Byte
INPUT 16 Byte
INPUT 2 Byte
INPUT 256 Byte
INPUT 32 Byte
INPUT 4 Byte

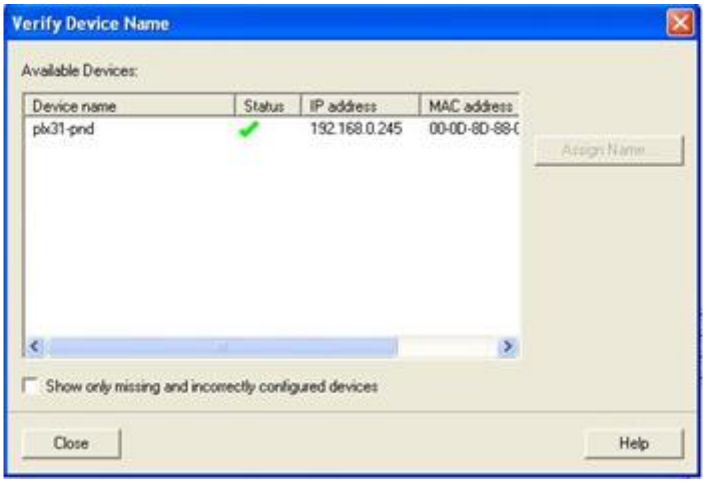
1	PS 307 2A
2	**CPU 315-2PN/DP**
X1	*MPI/DP*
X2	*PN-IO*
X2 P1	*Port 1*
X2 P2	*Port 2*
3	

Slot	Module	O...	I...	Q...	D...	Comment
0	***plx31-pnd***	***EPD***			***2042***	
Inte	*Interface1*				*2041*	
Port	*Port1*				*2040*	
1	INPUT 32 Byte		0...31			
2						
3						
4						
5						
6						
7						
8						
9						

19 Verify the name and IP address by clicking **PLC** > **Ethernet** > **Verify Device Name…**



Slot		Module	O...	I ...	Q...	D...	Comment
0		plx31-pnd	EPD			2042	
Inte		Interface1				2041	
Port		Port1				2040	
1		INPUT 32 Byte		0...31			
2		OUTPUT 32 Byte			0...31		
3							



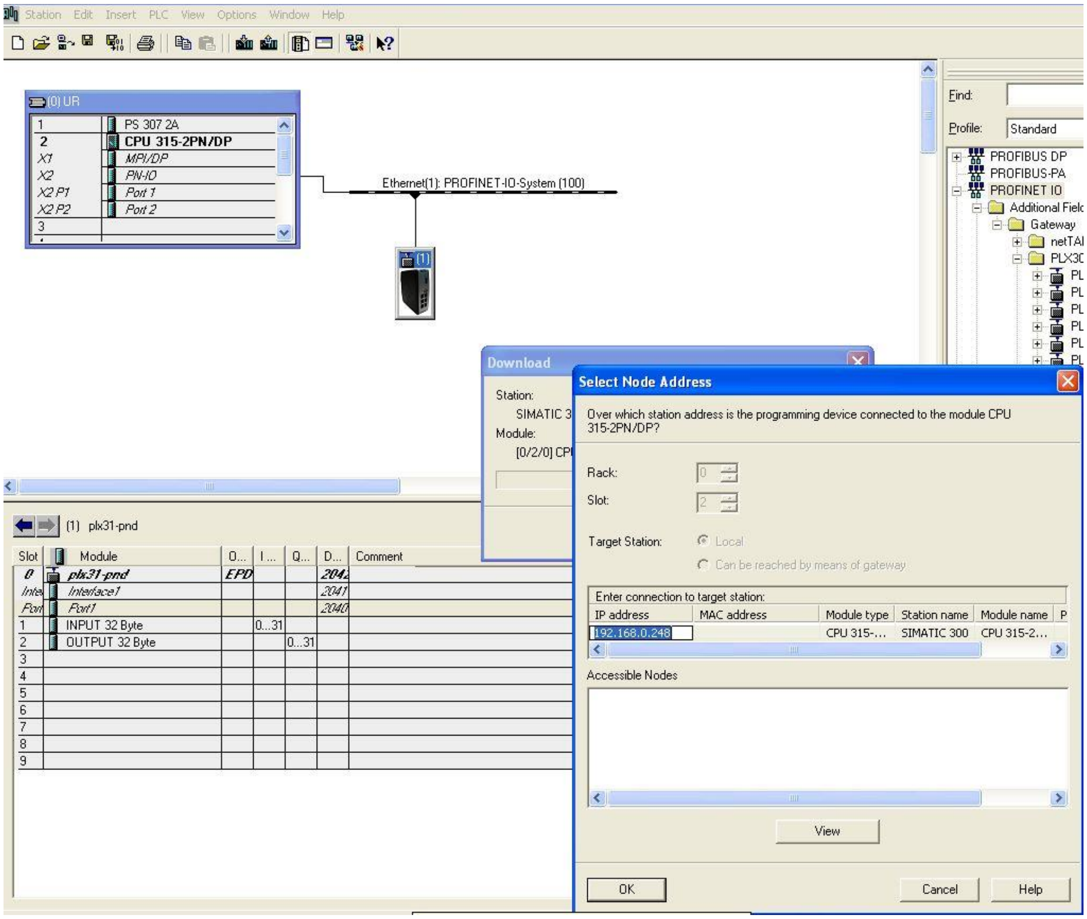
Available Devices:

Device name	Status	IP address	MAC address
plx31-pnd	✓	192.168.0.245	00-0D-8D-88-(

□ Show only missing and incorrectly configured devices

If the status is not green, the module needs the name and IP to be assigned. This is done by clicking **PLC** > **Ethernet** > **Assign Device Name**.

20 Once the status is green, save and download the program.



(0) UR

1	PS 307 2A
2	CPU 315-2PN/DP
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
X2 P2	Port 2
3	

(1) plx31-pnd

Slot	Module	O...	I...	Q...	D...	Comment
0	plx31-pnd	EPD			2042	
Inte	Interface1				2041	
Port	Port1				2040	
1	INPUT 32 Byte		0...31			
2	OUTPUT 32 Byte			0...31		
3						
4						
5						
6						
7						
8						
9						

Select Node Address

Over which station address is the programming device connected to the module CPU 315-2PN/DP?

Rack: 0

Slot: 2

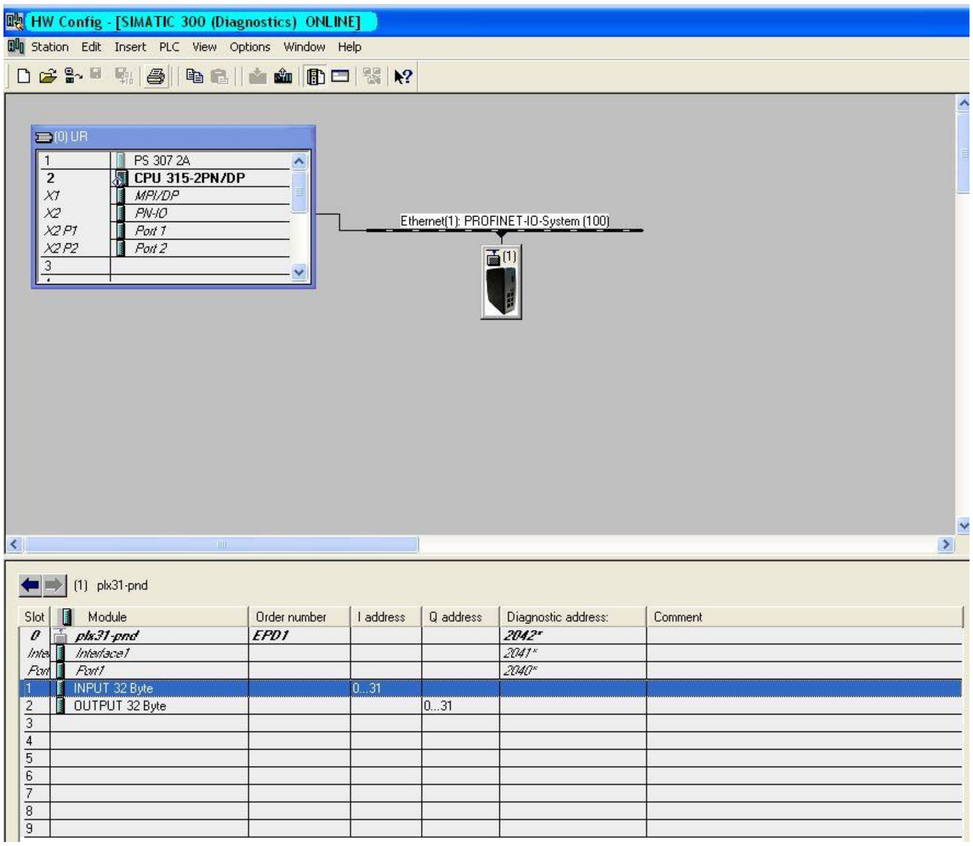
Target Station: ● Local ○ Can be reached by means of gateway

Enter connection to target station:

IP address	MAC address	Module type	Station name	Module name	P
192.168.0.248		CPU 315-...	SIMATIC 300	CPU 315-2...	

Accessible Nodes

Once you download the program, the RUN LED and the DC5V on the processor should be green. There should be no red LEDs.



HW Config - [SIMATIC 300 (Diagnostics) ONLINE]
Station Edit Insert PLC View Options Window Help
(0) UR
Ethernet(1): PROFINET-IO-System (100)
(1)

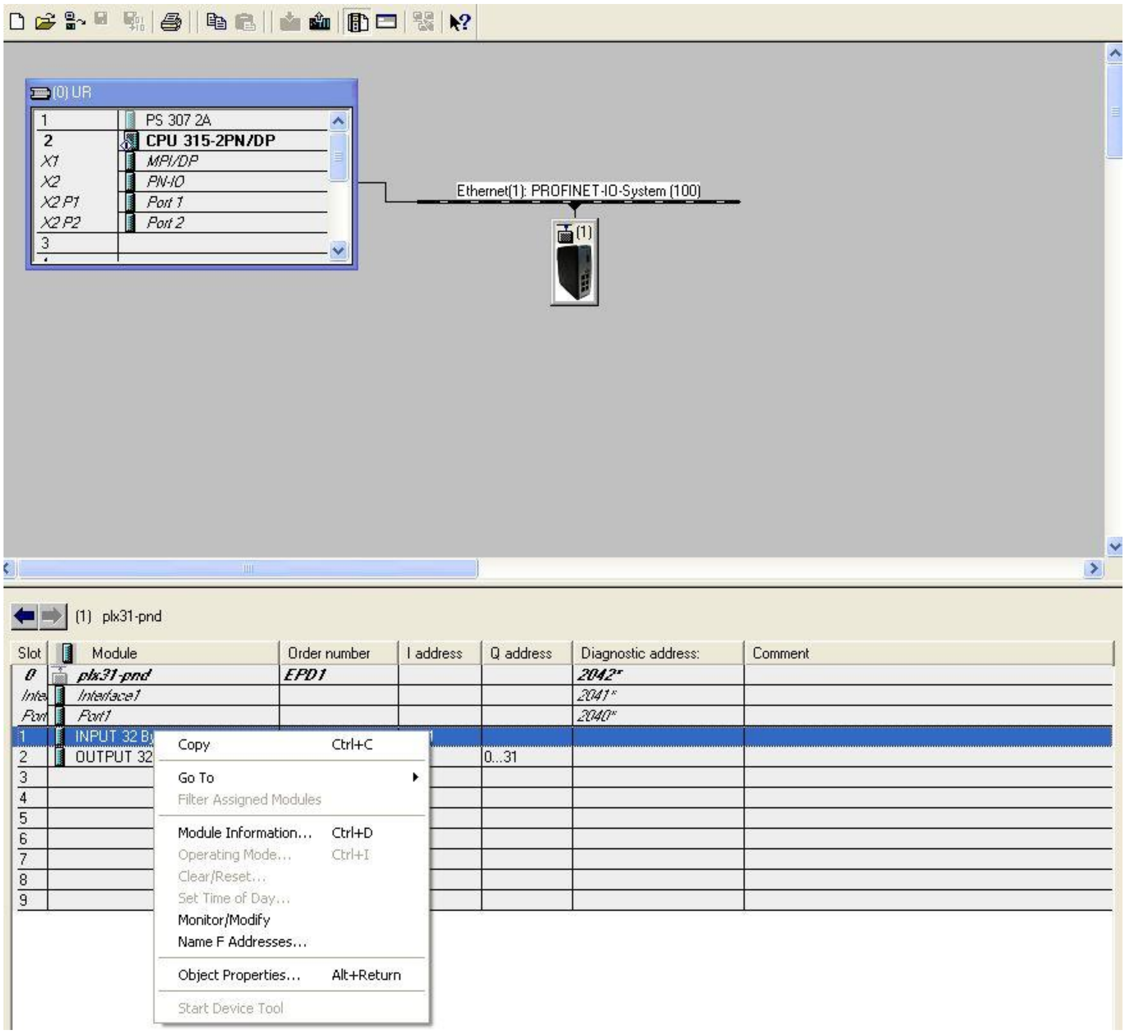
1	PS 307 2A
2	CPU 315-2PN/DP
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
X2 P2	Port 2
3	

(1) plx31-pnd

Slot	Module	Order number	I address	Q address	Diagnostic address:	Comment
0	plx31-pnd	EPD1			2042*	
Inte	Interface1				2041*	
Port	Port1				2040*	
1	INPUT 32 Byte		0...31			
2	OUTPUT 32 Byte			0...31		
3						
4						
5						
6						
7						
8						
9						

10.3.1 Monitoring Data in Step 7

1 To monitor the values coming out of the processor, click on the **Input** and choose the **Monitor/Modify** option.

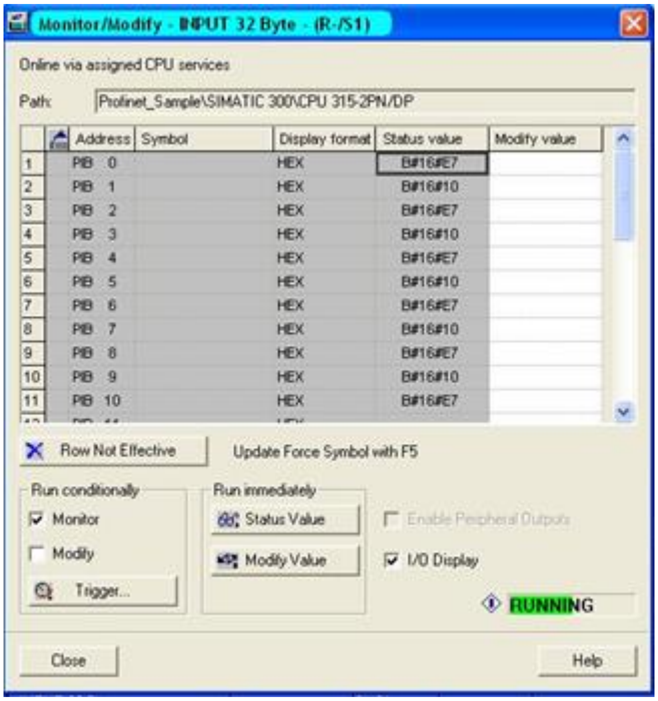


1	PS 307 2A
2	CPU 315-2PN/DP
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
X2 P2	Port 2
3	

(1) plx31-pnd

Slot	Module	Order number	I address	Q address	Diagnostic address:	Comment
0	***plx31-pnd***	***EPD1***			***2042****	
Inte	*Interface1*				*2041**	
Port	*Port1*				*2040**	
1	INPUT 32 By		1			
2	OUTPUT 32			0...31		
3						
4						
5						
6						
7						
8						
9						

The values coming into the processor are displayed at the input section.



Monitor/Modify - INPUT 32 Byte - (R-/S1)

Online via assigned CPU services

Path: Profinet_Sample\SIMATIC 300\CPU 315-2PN/DP

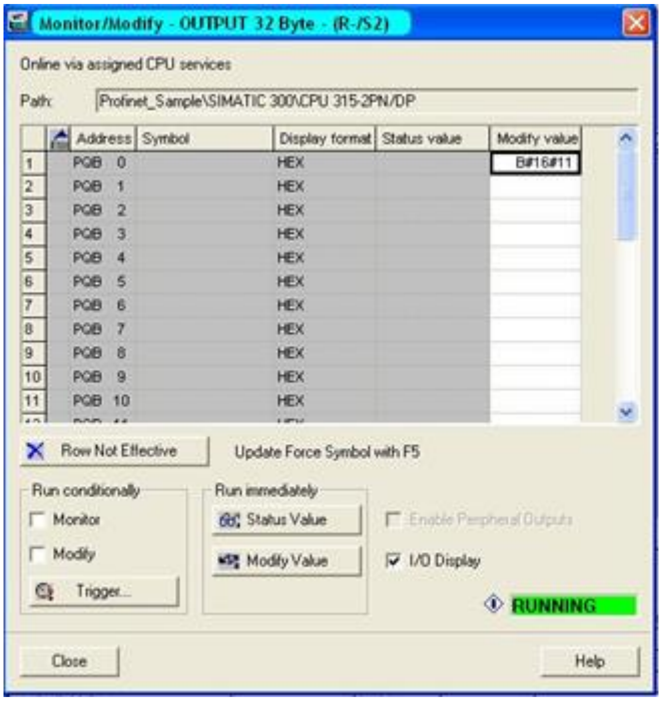
	Address	Symbol	Display format	Status value	Modify value
1	PIB 0		HEX	B#16#E7	
2	PIB 1		HEX	B#16#10	
3	PIB 2		HEX	B#16#E7	
4	PIB 3		HEX	B#16#10	
5	PIB 4		HEX	B#16#E7	
6	PIB 5		HEX	B#16#10	
7	PIB 6		HEX	B#16#E7	
8	PIB 7		HEX	B#16#10	
9	PIB 8		HEX	B#16#E7	
10	PIB 9		HEX	B#16#10	
11	PIB 10		HEX	B#16#E7	

Run conditionally: ☑ Monitor □ Modify

□ Enable Peripheral Outputs

☑ I/O Display

2 To modify the values that the processor sends to the module, choose the *Output*, select the desired value, and select **Modify Value**.



Monitor/Modify - OUTPUT 32 Byte - (R-/S2)

Online via assigned CPU services

Path: Profinet_Sample\SIMATIC 300\CPU 315-2PN/DP

	Address	Symbol	Display format	Status value	Modify value
1	PQB 0		HEX		B#16#11
2	PQB 1		HEX		
3	PQB 2		HEX		
4	PQB 3		HEX		
5	PQB 4		HEX		
6	PQB 5		HEX		
7	PQB 6		HEX		
8	PQB 7		HEX		
9	PQB 8		HEX		
10	PQB 9		HEX		
11	PQB 10		HEX		

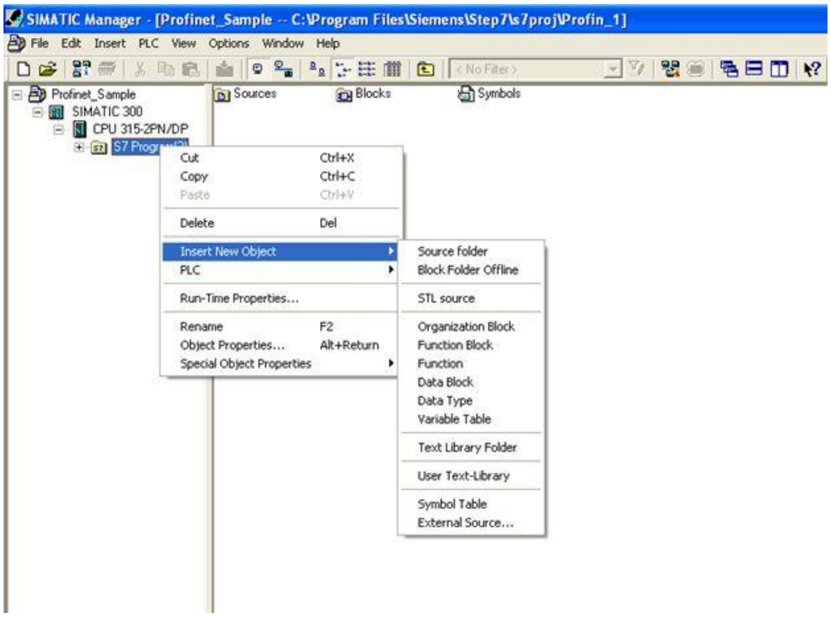
Run conditionally: □ Monitor □ Modify

□ Enable Peripheral Outputs

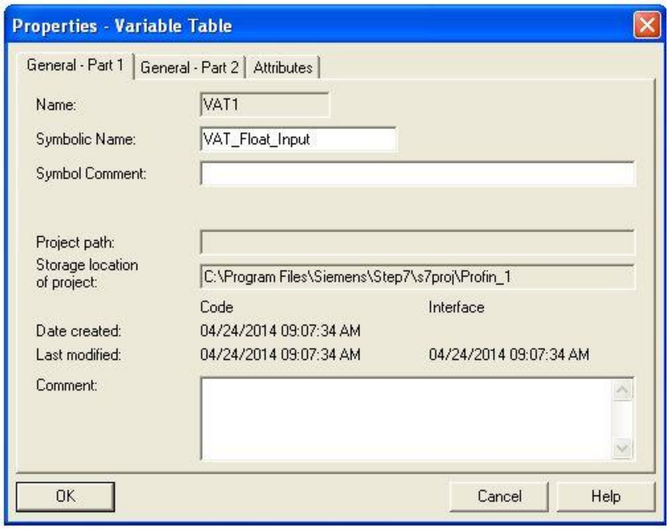
☑ I/O Display

10.3.2 Creating a Variable Table to Display Floating Point Input Values

1 Select **Insert New Object > Variable Table**.



2 Assign a name to the table and click **OK**.


Properties - Variable Table

General - Part 1 | General - Part 2 | Attributes

Name: VAT1

Symbolic Name: VAT_Float_Input

Symbol Comment:

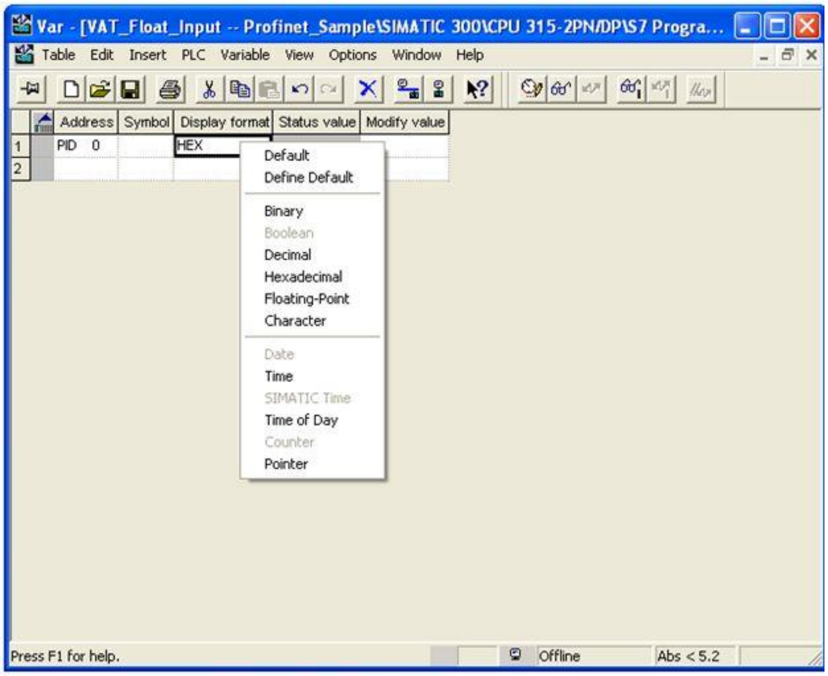
Project path:

Storage location of project: C:\Program Files\Siemens\Step7\s7proj\Profin_1

	Code	Interface
Date created:	04/24/2014 09:07:34 AM	
Last modified:	04/24/2014 09:07:34 AM	04/24/2014 09:07:34 AM

Comment:

3 Double-click on the table to monitor.



Var - [VAT_Float_Input -- Profinet_Sample\SIMATIC 300\CPU 315-2PN/DP\S7 Progra...

	Address	Symbol	Display format	Status value	Modify value
1	PID 0		HEX		
2					

Press F1 for help. Offline Abs < 5.2

10.4 PND Diagnostics

10.4.1 PCB Diagnostics

The best way to troubleshoot the PND driver is to use ProSoft Configuration Builder to access the diagnostic capabilities of the gateway through the Ethernet debug port. For instructions on how to access *Diagnostics*, see *Using Diagnostics in ProSoft Configuration Builder* (page 36).

10.4.2 PND Status Data in Upper Memory

The PND driver has an associated status data area located in the gateway's upper memory. The Data Map functionality of the gateway can be used to map this data into the normal user data range of the gateway's database. See *Using the CommonNet Data Map* (page 25).

The following table lists the PND driver's general status and error data that is located in the gateway's upper memory.

Note:
The status data starts at upper memory 6000 for the following gateways:

- PLX31-PND-MBS
- PLX31-PND-MBS4

The status data starts at upper memory 13000 for the following gateways:

- PLX31-EIP-PND
- PLX32-EIP-PND
- PLX31-MBTCP-PND
- PLX32-MBTCP-PND

Offset	Description
6000 / 13000	Total number of write messages to PLC
6001 / 13001	Total number of write message errors
6002 / 13002	Input Error message status. See below for error code descriptions.
6003 / 13003	Total number of read messages from PLC
6004 / 13004	Total number of read message errors
6005 / 13005	Output Error message status. See below for error code descriptions.
6006 / 13006	Connection Status
6007 / 13007	Driver Enabled? 1=Yes, 0=No

Input Error Message Status

Error code	Description
0	No error
0x0604	Pointer to data memory is NULL
0x0503	Write buffer is already locked or unlocked
0x060A	Data length to be written is invalid

Output Error Message Status

Error code	Description
0	No error
0x0001	No new data was copied into the buffer
0x0604	Pointer to data memory is NULL, or pointer to variable to receive the APDU Data Status is NULL
0x060B	IOCR ID is 0
0x0803	IOCR ID could not be found, or IOCR ID is invalid
0x060A	Data length to be read is invalid

10.4.3 Configuration Error Codes

The ERR LED is lit when no connection from the PLC is established.

Error Codes

Error code	Description
0	No error
1	Input Swap Error
2	Output Swap Error
18	Get Network Data Adapter
19	Get IP Address Error
20	Get NetMask Error
21	Get Gateway Error
22	Get MAC Address Error
23	PROFINET Mapping Error

11 Support, Service and Warranty

In This Chapter

11.1 Contacting Technical Support

ProSoft Technology, Inc. (ProSoft) is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

1. Product version number
2. System architecture
3. Network details

If the issue is hardware related, we will also need information regarding:

1. Gateway configuration and associated ladder files, if any
2. Gateway operation and any unusual behavior
3. Configuration/Debug status information
4. LED patterns
5. Details about the serial, Ethernet or fieldbus devices interfaced to the Gateway, if any.

Note: *For technical support calls within the United States, ProSoft's 24/7 after-hours phone support is available for urgent plant-down issues. Detailed contact information for all our worldwide locations is available on the following page.*

Internet	Web Site: www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com
Asia Pacific (location in Malaysia)	Tel: +603.7724.2080, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Asia Pacific (location in China)	Tel: +86.21.5187.7337 x888, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Europe (location in Toulouse, France)	Tel: +33 (0) 5.34.36.87.20, E-mail: support.EMEA@prosoft-technology.com Languages spoken include: French, English
Europe (location in Dubai, UAE)	Tel: +971-4-214-6911, E-mail: mea@prosoft-technology.com Languages spoken include: English, Hindi
North America (location in California)	Tel: +1.661.716.5100, E-mail: support@prosoft-technology.com Languages spoken include: English, Spanish
Latin America (Oficina Regional)	Tel: +1-281-2989109, E-Mail: latinam@prosoft-technology.com Languages spoken include: Spanish, English
Latin America (location in Puebla, Mexico)	Tel: +52-222-3-99-6565, E-mail: soporte@prosoft-technology.com Languages spoken include: Spanish
Brasil (location in Sao Paulo)	Tel: +55-11-5083-3776, E-mail: brasil@prosoft-technology.com Languages spoken include: Portuguese, English

11.2 Warranty Information

For complete details regarding ProSoft Technology’s TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS, please see the documents at www.prosoft-technology/legal

Documentation is subject to change without notice.

Index