Where Automation Connects.

PTQ-104C

Quantum Platform

IEC 60870-5-104 Client

March 4, 2013

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.

ProSoft Technology
5201 Truxtun Ave., 3rd Floor
Bakersfield, CA 93309
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

PTQ-104C User Manual

March 4, 2013

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 the enclosed DVD-ROM, and are available at no charge from our web site: http://www.prosoft-technology.com

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.

Information for ProTalk® Product Users

The statement "power, input and output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods Article 501-10(b) of the National Electrical Code, NFPA 70 for installations in the U.S., or as specified in section 18-1J2 of the Canadian Electrical Code for installations within Canada and in accordance with the authority having jurisdiction".

The following or equivalent warnings shall be included:

A Warning - Explosion Hazard - Substitution of components may Impair Suitability for Class I, Division 2;
B Warning - Explosion Hazard - When in Hazardous Locations, Turn off Power before replacing Wiring Modules, and
C Warning - Explosion Hazard - Do not Disconnect Equipment unless Power has been switched Off or the Area is known to be Nonhazardous.
D Caution: The Cell used in this Device may Present a Fire or Chemical Burn Hazard if Mistreated. Do not Disassemble, Heat above 100°C (212°F) or Incinerate.

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT - RISQUE D'EXPLOSION - AVANT DE DÉCONNECTER L'ÉQUIPEMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

Class I, Division 2 GPs A, B, C, D

II 3 G

Ex nA IIC X

0° C <= Ta <= 60° C

II - Equipment intended for above ground use (not for use in mines).

3 - Category 3 equipment, investigated for normal operation only.

G - Equipment protected against explosive gasses.

Warnings

North America Warnings

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
B Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or rewiring modules. Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C Suitable for use in Class I, Division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage:

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction.

A Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
B Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
D DO NOT OPEN WHEN ENERGIZED.

Electrical Ratings

- Backplane Current Load: 1100 mA maximum @ 5 Vdc ± 5%
- Operating Temperature: 0°C to 60°C (32°F to 140°F)
- Storage Temperature: -40°C to 85°C (-40°F to 185°F)
- Shock: 30 g operational; 50 g non-operational; Vibration: 5 g from 10 to 150 Hz
- Relative Humidity: 5% to 95% (without condensation)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings:

CSA/cUL

CSA CB Certified

ATEX

Important Notice:

⚠	CAUTION: THE CELL USED IN THIS DEVICE MAY PRESENT A FIRE OR CHEMICAL BURN HAZARD IF MISTREATED. DO NOT DISASSEMBLE, HEAT ABOVE 100°C (212°F) OR INCINERATE. Maximum battery load = 200 µA. Maximum battery charge voltage = 3.4 VDC. Maximum battery charge current = 500 µA. Maximum battery discharge current = 30 µA.

Contents

Guide to the PTQ-104C User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 10)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 93)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview	→	Reference (page 104) Product Specifications (page 106) Functional Overview (page 106)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 131) Index	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

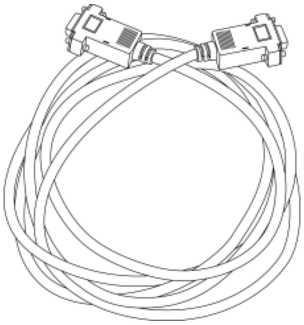
This document is intended to guide the user through the ProTalk module setup process. The user will learn how to:

- Set up the processor environment for the PTQ module
- View how the PTQ module exchanges data with the processor
- Edit and download configuration files from the PC to the PTQ module
- Monitor the operation of the PTQ module

1.1 Hardware and Software Requirements

1.1.1 Package Contents

ProTalk Module



Null Modem Serial Cable

ProSoft Solutions DVD

1.1.2 Quantum Hardware

This guide assumes the familiarity of the installation and setup of the Quantum hardware. The following should be installed, configured, and powered up before proceeding:

- Quantum Processor
- Quantum rack
- Quantum power supply
- Quantum Modbus Plus Network Option Module (NOM Module) (optional)
- Quantum to PC programming hardware
- NOM Ethernet or Serial connection to PC

1.1.3 PC and PC Software

- Windows-based PC with at least one COM port
- Quantum programming software installed on machine

  or
- Concept™ PLC Programming Software version 2.6

  or
  ProWORX PLC Programming Software
  or
  Unity™ Pro PLC Programming Software

> **Note:** ProTalk modules are compatible with common Quantum programming applications, including Concept and Unity Pro. For all other programming applications, please contact technical support.

1.2 Deployment Checklist

These steps are used for the installation of the ProTalk module.

The example programs are designed to allow the module and the processor to communicate with each other over the backplane. After this initial installation, additional steps may be needed for the specific needs of the application.

These steps must be completed in the following order:

1 Install the *ProSoft Configuration Builder* software on the PC

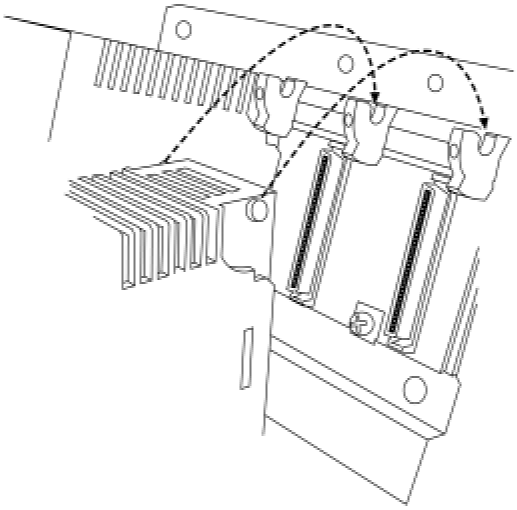
> **Important:** Earlier versions of ProSoft Configuration Builder do not support the Hot Standby (HSBY) feature on the PTQ-104C module. To make full use of the HSBY feature, please download the latest version of ProSoft Configuration Builder and review the readme files from the ProSoft Technology website at www.prosoft-technology.com/pcb.

2 Install the ProTalk module in the rack
3 Configure the module
4 Configure the 104 Client including client commands
5 Configure the processor
6 Verify communication between the processor and the module

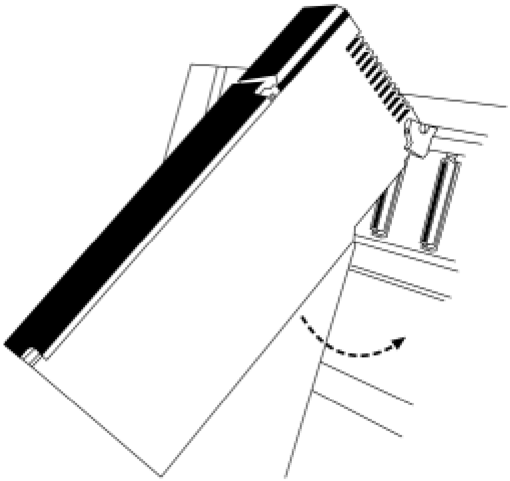
1.3 Installing the ProTalk Module in the Quantum Rack

1 Place the Module in the Quantum Rack. The ProTalk module must be placed in the same rack as the processor.

2 Tilt the module at a 45° angle and align the pegs at the top of the module with slots on the backplane.



3 Push the module into place until it seats firmly in the backplane.



Caution: The PTQ module is hot-swappable; it can be installed and removed while the rack is powered up. Do not assume that this is the case for all types of modules unless the user manual for the product explicitly states that the module is hot-swappable. Failure to observe this precaution could result in damage to the module and any equipment connected to it.

1.4 Installing ProSoft Configuration Builder Software

The ProSoft Configuration Builder (PCB) software is used to configure the module. The latest version of PCB can be found at the ProSoft Technology web site.

<u>Installing ProSoft Configuration Builder from the ProSoft Web Site</u>

1 Open the web browser and navigate to http://www.prosoft-technology.com/pcb

2 Click the **DOWNLOAD HERE** link to download the latest version of ProSoft Configuration Builder.

3 Choose **SAVE** or **SAVE FILE** when prompted.

4 Save the file to the *Windows Desktop*.

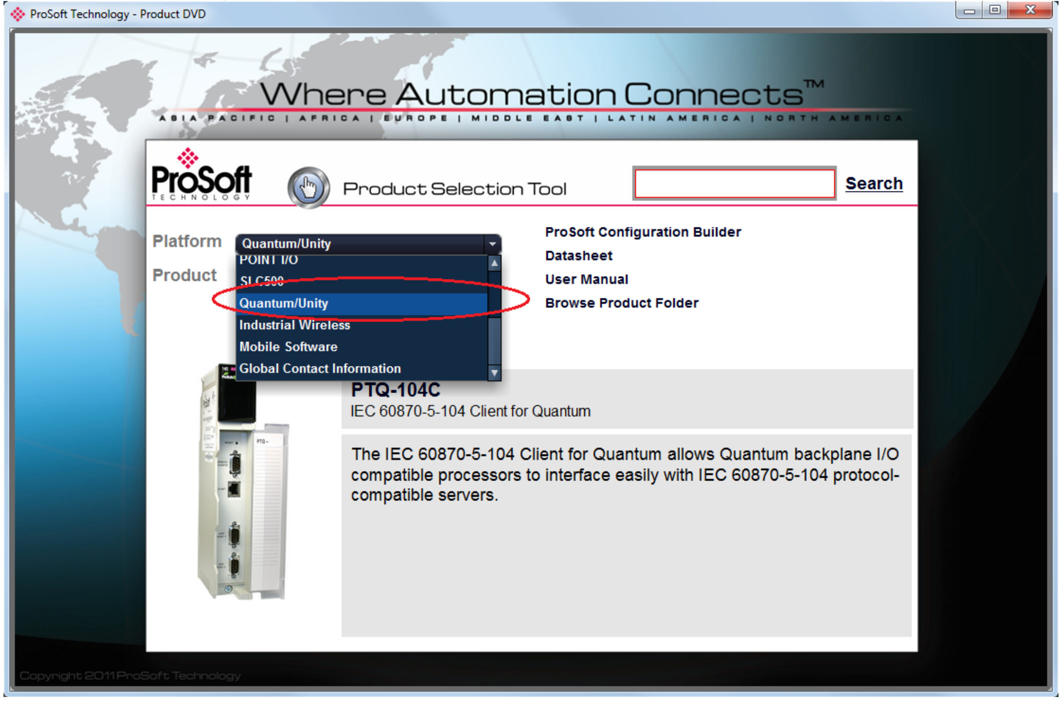
5 When the download is complete, locate and open the file, and follow the instructions on the screen to install the program.

If there is no access to the Internet, *ProSoft Configuration Builder* can be installed from the ProSoft Solutions DVD, included in the package with the module.

<u>To install ProSoft Configuration Builder from the Product DVD</u>

1 Insert the ProSoft Solutions Product DVD into the PC. Wait for the startup screen to appear.

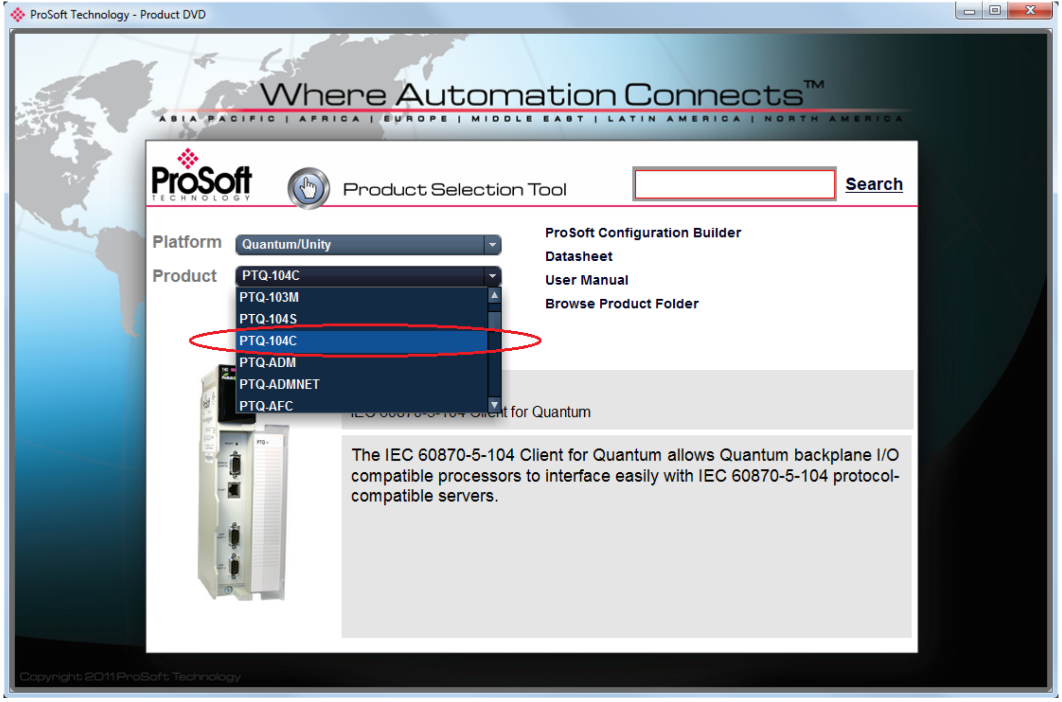
2 On the startup screen, click **PLATFORM** and select *Quantum/Unity*.


ProSoft Technology - Product DVD
Where Automation Connects™
ASIA PACIFIC | AFRICA | EUROPE | MIDDLE EAST | LATIN AMERICA | NORTH AMERICA
ProSoft TECHNOLOGY
Product Selection Tool
Search
Platform
Product
Quantum/Unity
POINT I/O
SLC500
Quantum/Unity
Industrial Wireless
Mobile Software
Global Contact Information
ProSoft Configuration Builder
Datasheet
User Manual
Browse Product Folder
Copyright 2011 ProSoft Technology

PTQ-104C
IEC 60870-5-104 Client for Quantum

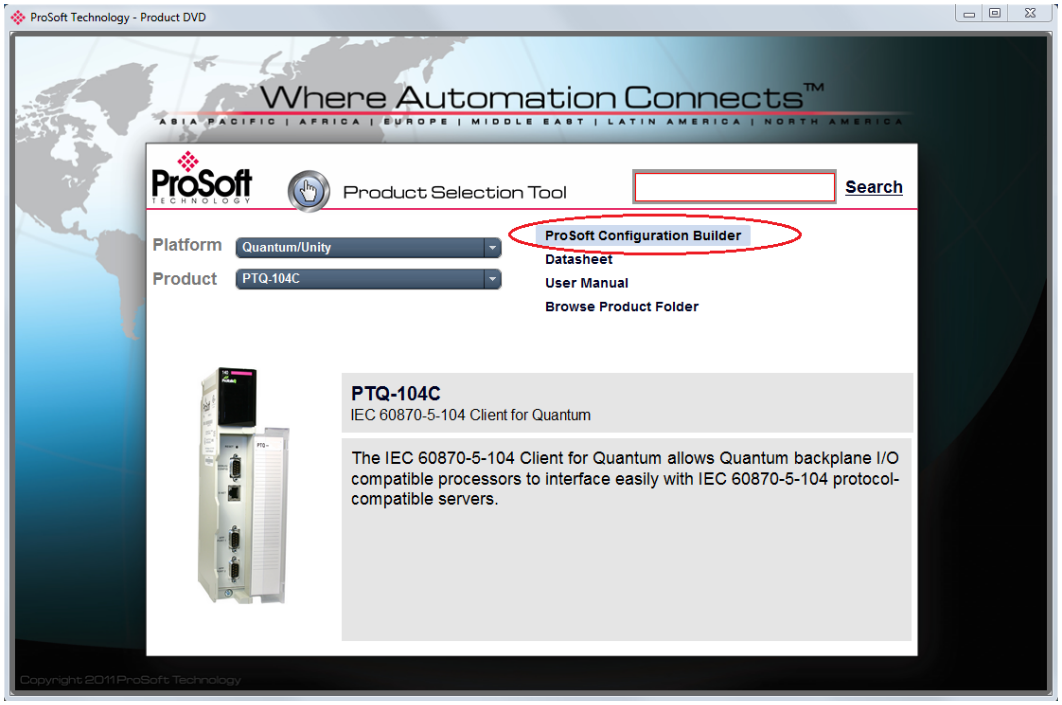
The IEC 60870-5-104 Client for Quantum allows Quantum backplane I/O compatible processors to interface easily with IEC 60870-5-104 protocol-compatible servers.

3 Click **PRODUCT** and select *PTQ-104C*.


ProSoft Technology - Product DVD
Where Automation Connects™
ASIA PACIFIC | AFRICA | EUROPE | MIDDLE EAST | LATIN AMERICA | NORTH AMERICA
ProSoft TECHNOLOGY
Product Selection Tool
Search
Platform
Quantum/Unity
Product
PTQ-104C
PTQ-103M
PTQ-104S
PTQ-104C
PTQ-ADM
PTQ-ADMNET
PTQ-AFC
ProSoft Configuration Builder
Datasheet
User Manual
Browse Product Folder
...nt for Quantum
Copyright 2011 ProSoft Technology

The IEC 60870-5-104 Client for Quantum allows Quantum backplane I/O compatible processors to interface easily with IEC 60870-5-104 protocol-compatible servers.

4 Click the **PROSOFT CONFIGURATION BUILDER** option to install the software on the PC.


ProSoft Technology - Product DVD
Where Automation Connects™
ASIA PACIFIC | AFRICA | EUROPE | MIDDLE EAST | LATIN AMERICA | NORTH AMERICA
ProSoft TECHNOLOGY
Product Selection Tool
Search
Platform
Quantum/Unity
Product
PTQ-104C
ProSoft Configuration Builder
Datasheet
User Manual
Browse Product Folder
Copyright 2011 ProSoft Technology

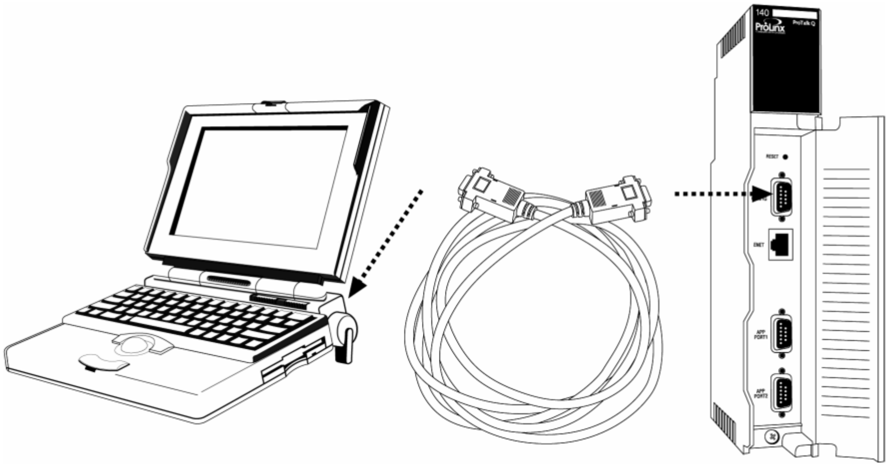
PTQ-104C
IEC 60870-5-104 Client for Quantum

The IEC 60870-5-104 Client for Quantum allows Quantum backplane I/O compatible processors to interface easily with IEC 60870-5-104 protocol-compatible servers.

1.5 Connecting the PC to the ProTalk Configuration/Debug Port

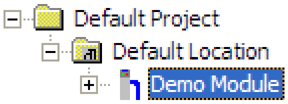
Make sure the Quantum programming software is closed before performing these steps. This action will avoid serial port conflict.

Using the supplied Null Modem cable, connect the PC to the Configuration/Debug port on the ProTalk module as shown



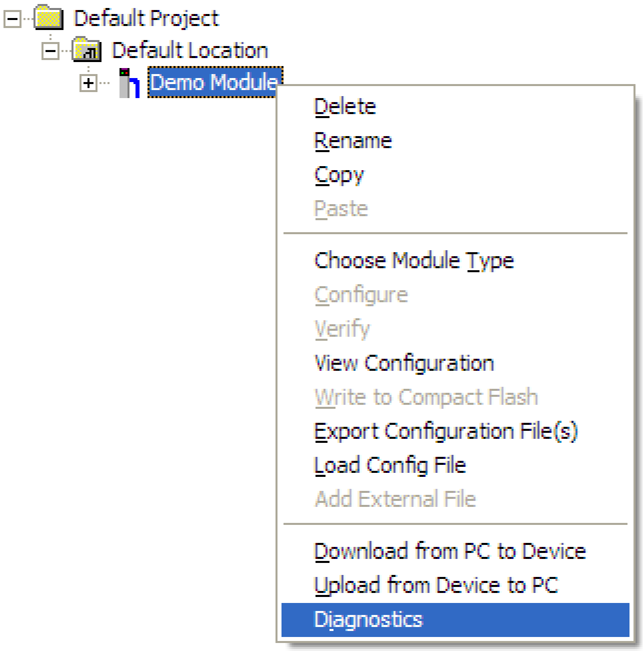
To connect to the module's Configuration/Debug serial port

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



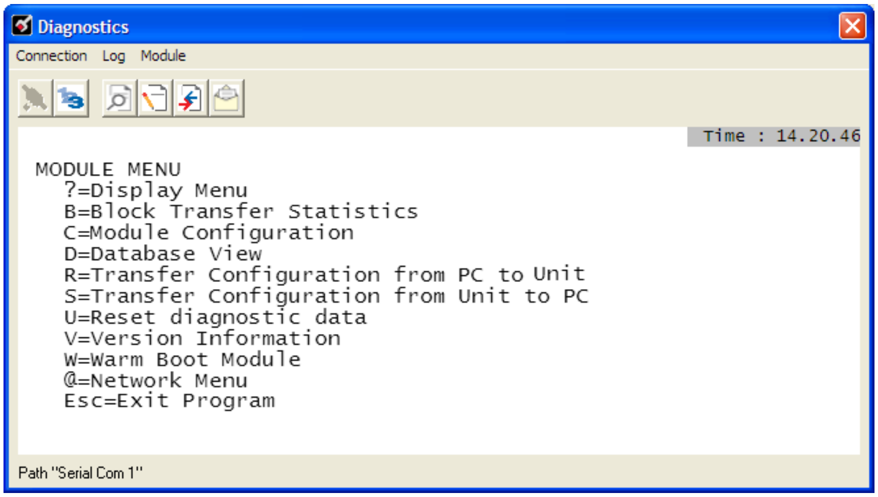
Default Project
Default Location
Demo Module

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



This action opens the **DIAGNOSTICS** dialog box.

3 Press **[?]** to open the Main Menu.

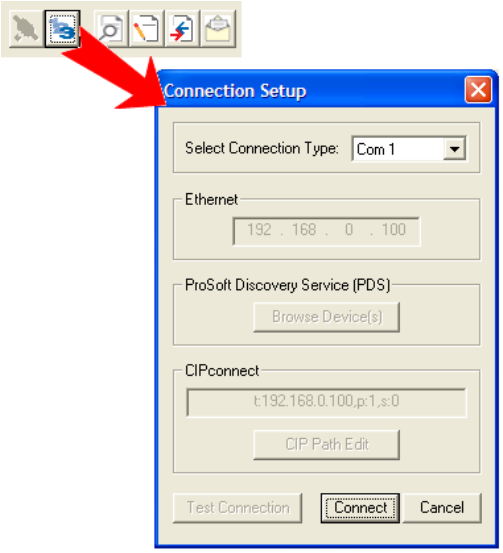


```
                                              Time : 14.20.46

MODULE MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R=Transfer Configuration from PC to Unit
  S=Transfer Configuration from Unit to PC
  U=Reset diagnostic data
  V=Version Information
  W=Warm Boot Module
  @=Network Menu
  Esc=Exit Program
```

Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in the own module.

If there is no response from the module, follow these steps:

1 Click to configure the connection. On the Connection Setup dialog box, select a valid com port or other connection type supported by the module.



Select Connection Type: Com 1

192 . 168 . 0 . 100

t:192.168.0.100,p:1,s:0

2 Verify that the null modem cable is connected properly between the computer's serial port and the module. A regular serial cable will not work.

3 On computers with more than one serial port, verify that the communication program is connected to the same port that is connected to the module.

4 If a connection is still not established, contact ProSoft Technology for assistance.

1.5.1 Ethernet Configuration

Use this procedure to configure the Ethernet settings for the module. An IP address, subnet mask and module address must be assigned. After completing this step, the module can be connected with an Ethernet cable.

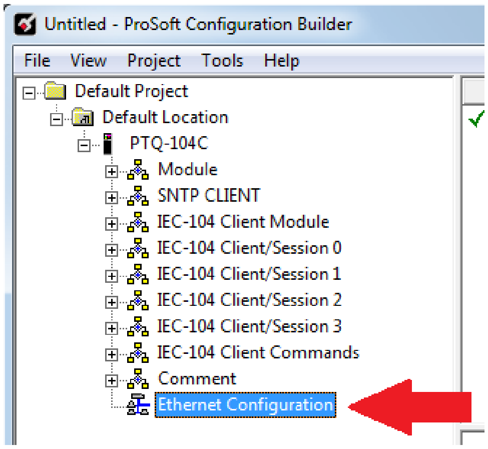
1 Determine the network settings for the module, with the help of the network administrator if necessary. The following information is needed:

o IP Address (fixed IP required) _____ . _____ . _____ . _____
o Subnet Mask _____ . _____ . _____ . _____
o Gateway Address _____ . _____ . _____ . _____

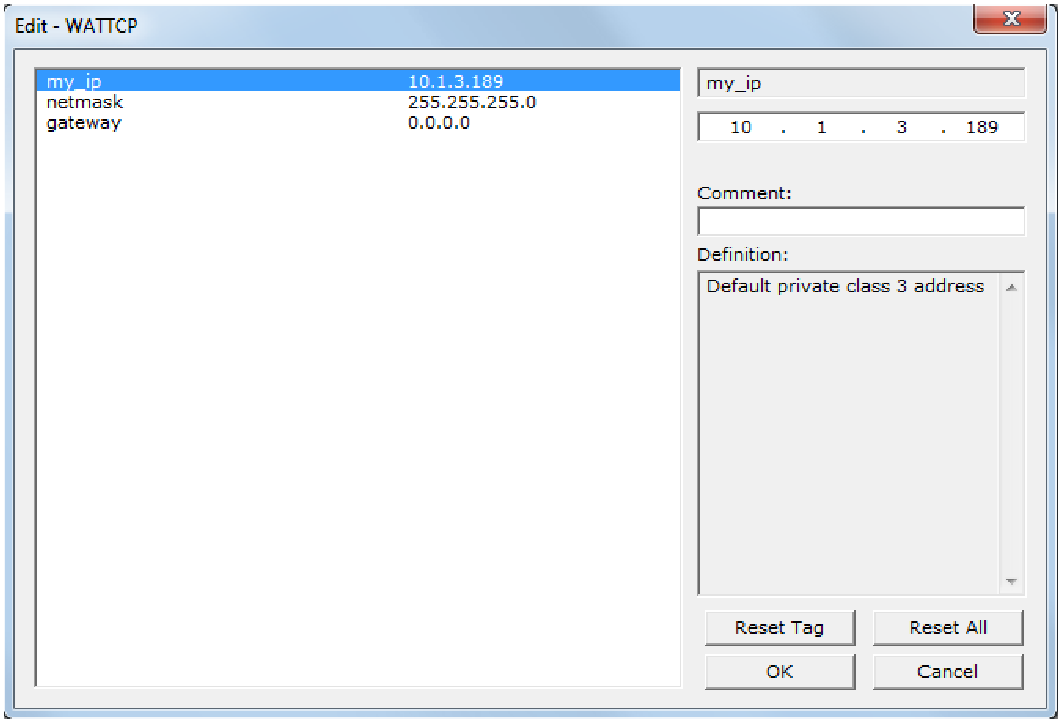
Note: The module Address is optional, and is not required for networks that do not use a default module.

HSBY Note: Hot Standby Primary IP is entered. The Standby IP address will always be the Primary IP address plus 1.

2 Double-click the **ETHERNET CONFIGURATION** icon.



3 This action opens the **EDIT** dialog box.



my_ip	10.1.3.189
netmask	255.255.255.0
gateway	0.0.0.0

my_ip

10 . 1 . 3 . 189

Comment:

Definition:

Default private class 3 address

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

5 When finished editing, click **OK** to save the changes and return to the ProSoft Configuration Builder window.

2 Configuring the Processor with Unity Pro

In This Chapter

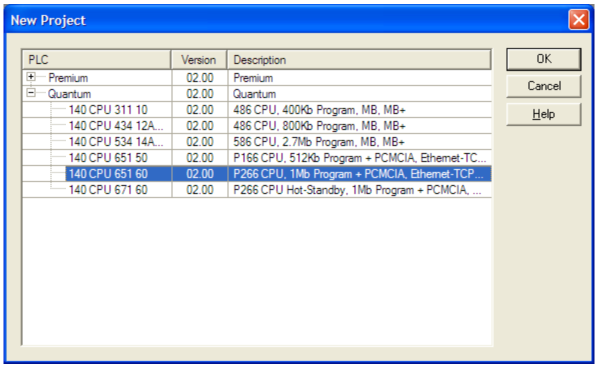
The following steps are designed to ensure the processor (Quantum or Unity) is able to transfer data successfully with the PTQ module. As part of this procedure, Unity Pro will be used to create a project, add the PTQ module to the project, set up data memory for the project, and download the project to the processor.

2.1 Creating a New Project

The first step is to open Unity Pro and create a new project.

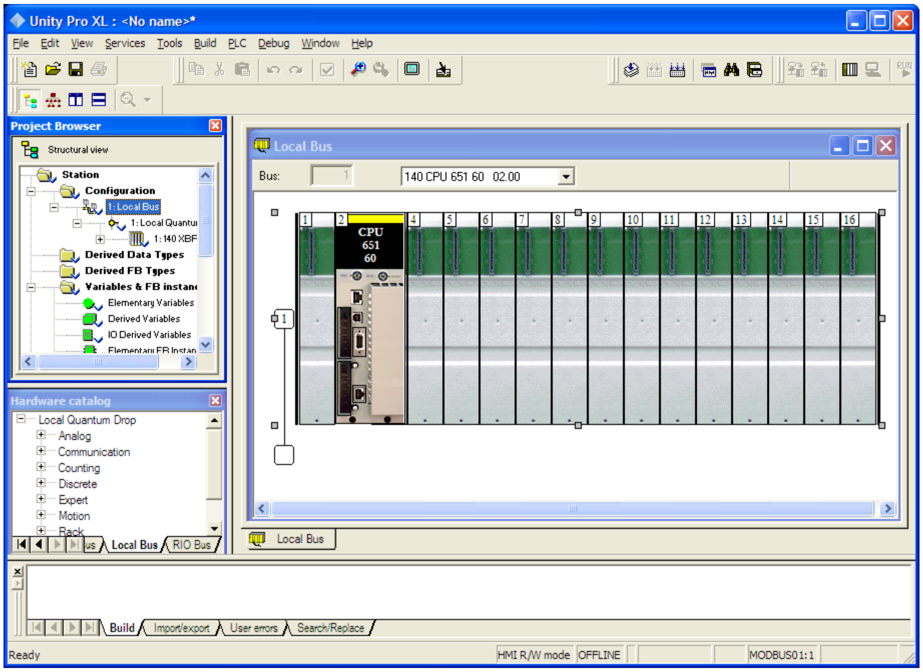
1 In the *New Project* dialog box, choose the CPU type. In the following illustration, the CPU is 140 CPU 651 60. Choose the processor type that matches the hardware configuration of the processor in the rack, if it differs from the example.

2 Click **OK** to continue.

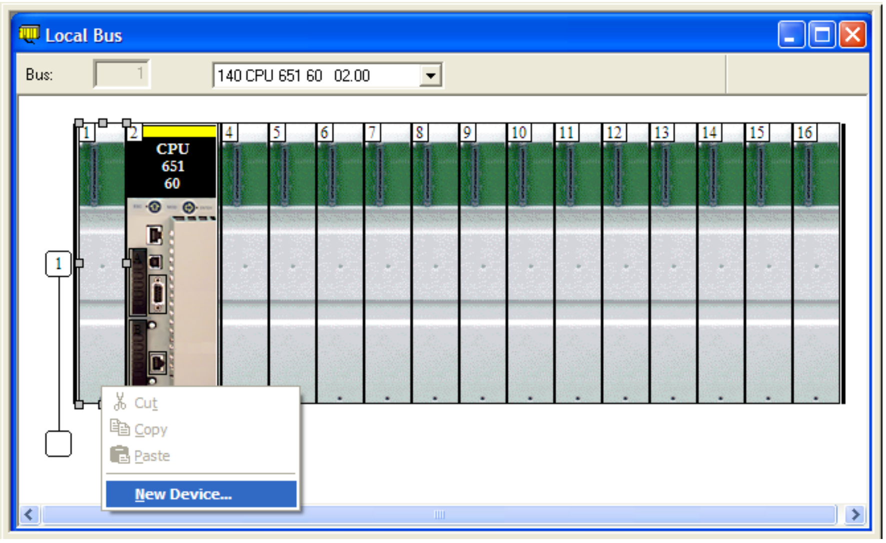

New Project

PLC	Version	Description
Premium	02.00	Premium
Quantum	02.00	Quantum
140 CPU 311 10	02.00	486 CPU, 400Kb Program, MB, MB+
140 CPU 434 12A...	02.00	486 CPU, 800Kb Program, MB, MB+
140 CPU 534 14A...	02.00	586 CPU, 2.7Mb Program, MB, MB+
140 CPU 651 50	02.00	P166 CPU, 512Kb Program + PCMCIA, Ethernet-TC...
140 CPU 651 60	02.00	P266 CPU, 1Mb Program + PCMCIA, Ethernet-TCP...
140 CPU 671 60	02.00	P266 CPU Hot-Standby, 1Mb Program + PCMCIA, ...

3 Next, add a power supply to the project. In the *Project Browser*, expand the *Configuration* folder, and then double-click the **1:LocalBus** icon. This action opens a graphical window showing the arrangement of devices in the Quantum rack.

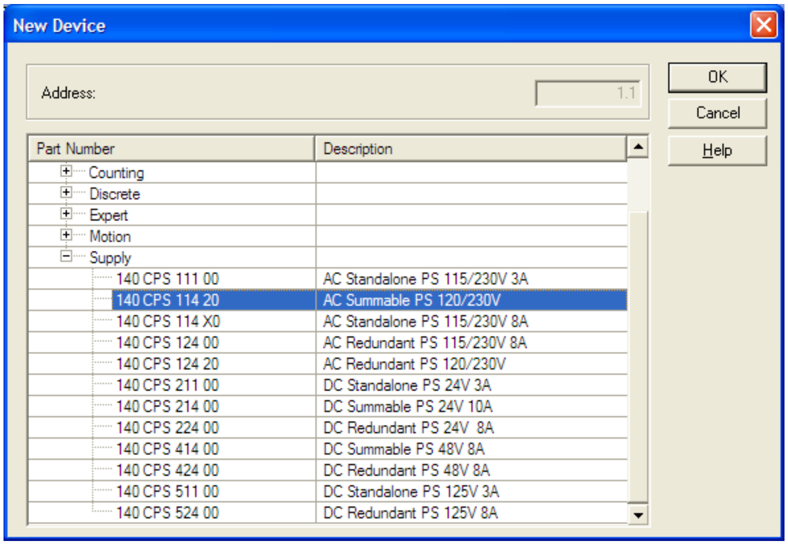


4 Select the rack position for the power supply, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW DEVICE**.



5 Expand the *Supply* folder, and select the power supply from the list.

6 Click **OK** to continue.



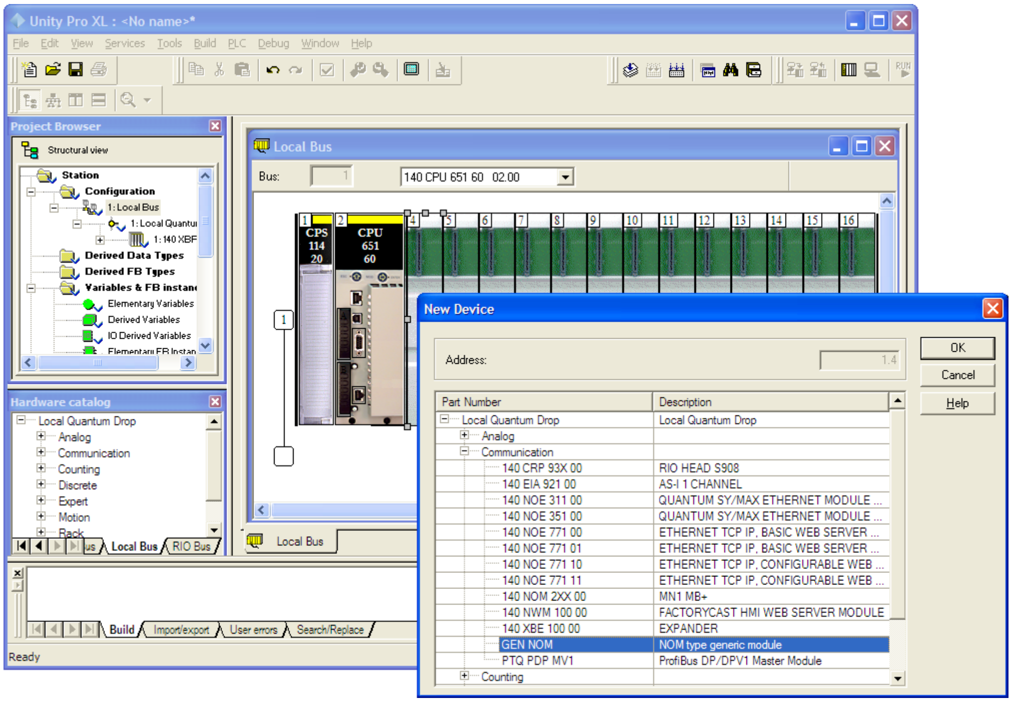
Address: 1.1

Part Number	Description
⊞ Counting	
⊞ Discrete	
⊞ Expert	
⊞ Motion	
⊟ Supply	
140 CPS 111 00	AC Standalone PS 115/230V 3A
140 CPS 114 20	AC Summable PS 120/230V
140 CPS 114 X0	AC Standalone PS 115/230V 8A
140 CPS 124 00	AC Redundant PS 115/230V 8A
140 CPS 124 20	AC Redundant PS 120/230V
140 CPS 211 00	DC Standalone PS 24V 3A
140 CPS 214 00	DC Summable PS 24V 10A
140 CPS 224 00	DC Redundant PS 24V 8A
140 CPS 414 00	DC Summable PS 48V 8A
140 CPS 424 00	DC Redundant PS 48V 8A
140 CPS 511 00	DC Standalone PS 125V 3A
140 CPS 524 00	DC Redundant PS 125V 8A

7 Repeat these steps to add any additional devices to the Quantum Rack.

2.2 Adding the PTQ Module to the Project

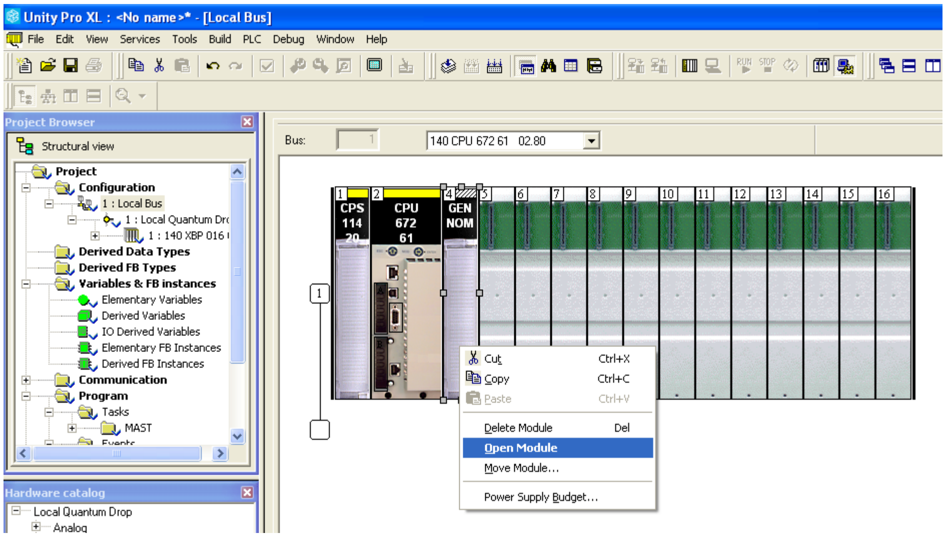
1 When adding a new device, expand the *Communication* tree, and select **GEN NOM**. This module type provides extended communication capabilities for the Quantum system, and allows communication between the PLC and the PTQ module without requiring additional programming. Click **OK**.



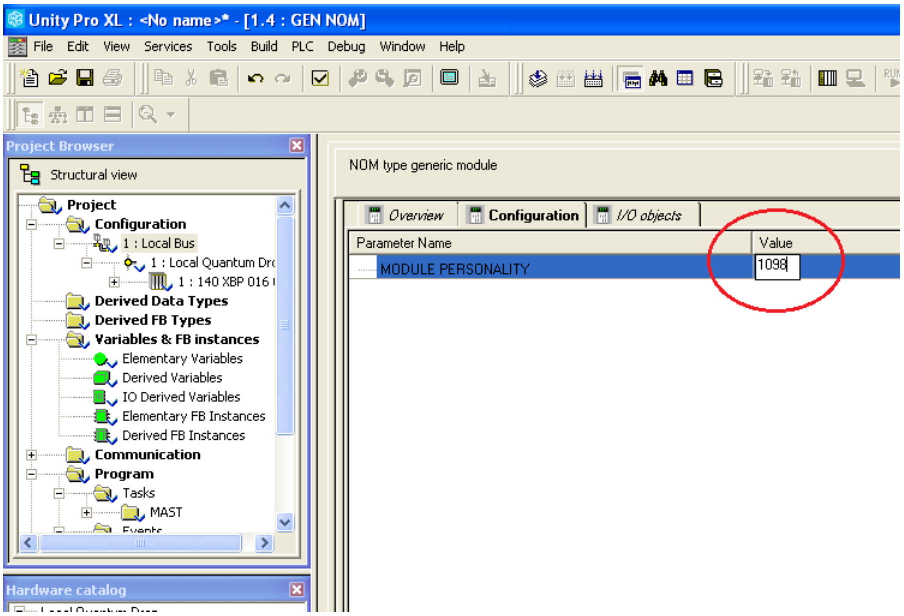
Address: 1.4

Part Number	Description
Local Quantum Drop	Local Quantum Drop
Analog	
Communication	
140 CRP 93X 00	RIO HEAD S908
140 EIA 921 00	AS-I 1 CHANNEL
140 NOE 311 00	QUANTUM SY/MAX ETHERNET MODULE ...
140 NOE 351 00	QUANTUM SY/MAX ETHERNET MODULE ...
140 NOE 771 00	ETHERNET TCP IP, BASIC WEB SERVER ...
140 NOE 771 01	ETHERNET TCP IP, BASIC WEB SERVER ...
140 NOE 771 10	ETHERNET TCP IP, CONFIGURABLE WEB ...
140 NOE 771 11	ETHERNET TCP IP, CONFIGURABLE WEB ...
140 NOM 2XX 00	MN1 MB+
140 NWM 100 00	FACTORYCAST HMI WEB SERVER MODULE
140 XBE 100 00	EXPANDER
GEN NOM	NOM type generic module
PTQ PDP MV1	ProfiBus DP/DPV1 Master Module
Counting	

2 Right click the GEN NOM icon and select Open Module.



3 Enter the module personality value. The correct value for ProTalk modules is 1098 decimal (044A hex).



Parameter Name	Value
MODULE PERSONALITY	1098

4 Before the project can be saved in Unity Pro, the modifications must be validated. Open the **EDIT** menu, and choose **VALIDATE.** If no errors are reported, the project can be saved.

5 **SAVE** the project.

2.3 Building the Project

Whenever the configuration of the PTQ module or the processor is updated, the changed configuration must be imported from the module. The project must be built (compiled) before downloading it to the processor.

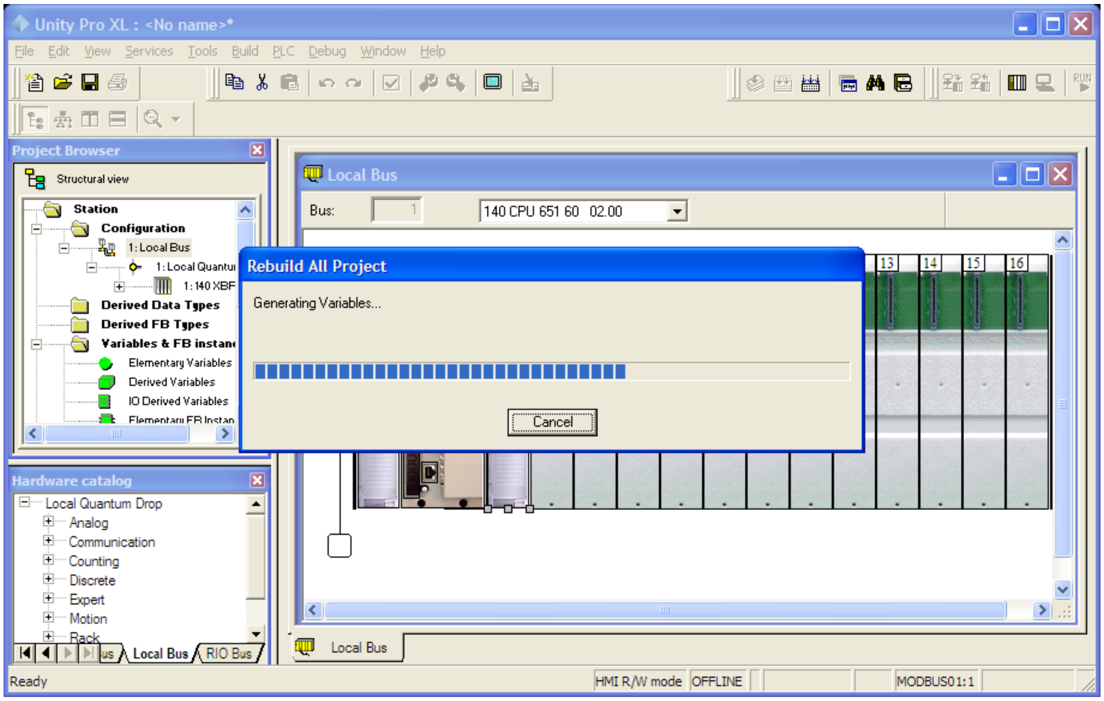
> **Note:** The following steps show how to build the project in Unity Pro. This is not intended to provide detailed information on using Unity Pro, or debugging the programs. Refer to the documentation for the processor and for Unity Pro for specialized information.

To build (compile) the project

1 Review the elements of the project in the *Project Browser*.

2 When the project is ready to be downloaded, open the **BUILD** menu, and choose **REBUILD ALL PROJECT**. This action builds (compiles) the project into a form the processor can use to execute the instructions in the project file. This task may take several minutes, depending on the complexity of the project and the resources available on the PC.

3 As the project is built, Unity Pro reports its process in a *Progress* dialog box, with details appearing in a pane at the bottom of the window. The following illustration shows the build process under way.



Unity Pro XL : <No name>*

Bus: 1 140 CPU 651 60 02.00

Rebuild All Project

Generating Variables...

Ready HMI R/W mode OFFLINE MODBUS01:1

After the build process is completed successfully, the next step is to download the compiled project to the processor.

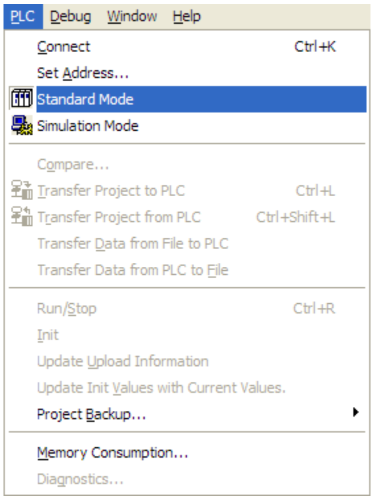
2.4 Connecting a PC to the Processor

The next step is to connect to the processor so the project file can be downloaded. The processor uses this project file to communicate over the backplane to modules identified in the project file.

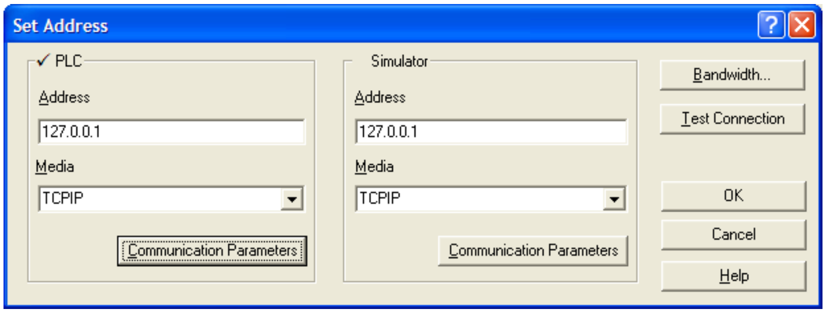
> **Note:** If a connection between the PC to the processor has not been established, verify the port drivers are installed and are available to Unity Pro.

Verifying the address and driver settings in Unity Pro

1 Open the **PLC** menu, and choose **STANDARD MODE**. This action turns off the PLC Simulator. It allows the user to communicate directly with the Quantum or Unity hardware.


PLC
Debug
Window
Help
Connect Ctrl+K
Set Address...
Standard Mode
Simulation Mode
Compare...
Transfer Project to PLC Ctrl+L
Transfer Project from PLC Ctrl+Shift+L
Transfer Data from File to PLC
Transfer Data from PLC to File
Run/Stop Ctrl+R
Init
Update Upload Information
Update Init Values with Current Values.
Project Backup...
Memory Consumption...
Diagnostics...

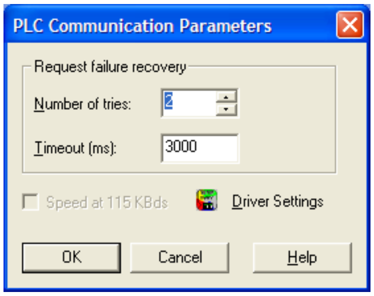
2 Open the **PLC** menu, and choose **SET ADDRESS...** This action opens the *Set Address* dialog box. Open the **MEDIA** dropdown list and choose the connection type (TCPIP or USB).


Set Address

✓ PLC
Address: 127.0.0.1
Media: TCPIP

Simulator
Address: 127.0.0.1
Media: TCPIP

3 If the **MEDIA** dropdown list does not contain the desired connection method, click the **COMMUNICATION PARAMETERS** button in the PLC area of the dialog box. This action opens the *PLC Communication Parameters* dialog box.

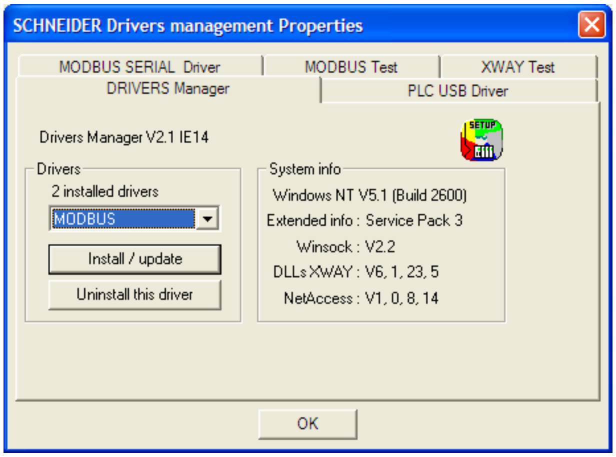


Number of tries: 2

Timeout (ms): 3000

☐ Speed at 115 KBds

4 Click the **DRIVER SETTINGS** button to open the *SCHNEIDER Drivers management Properties* dialog box.



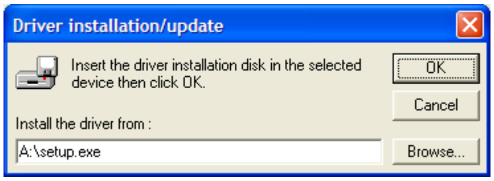
Drivers Manager V2.1 IE14

2 installed drivers

MODBUS

Windows NT V5.1 (Build 2600)
Extended info : Service Pack 3
Winsock : V2.2
DLLs XWAY : V6, 1, 23, 5
NetAccess : V1, 0, 8, 14

5 Click the **INSTALL/UPDATE** button to specify the location of the Setup.exe file containing the drivers to use. The Unity Pro installation disks will be needed for this step.



Insert the driver installation disk in the selected device then click OK.

Install the driver from : A:\setup.exe

6 Click the **BROWSE** button to locate the *Setup.exe* file to execute, and execute the setup program. After the installation, restart the PC. Refer to the Schneider Electric documentation for more information on installing drivers for Unity Pro.

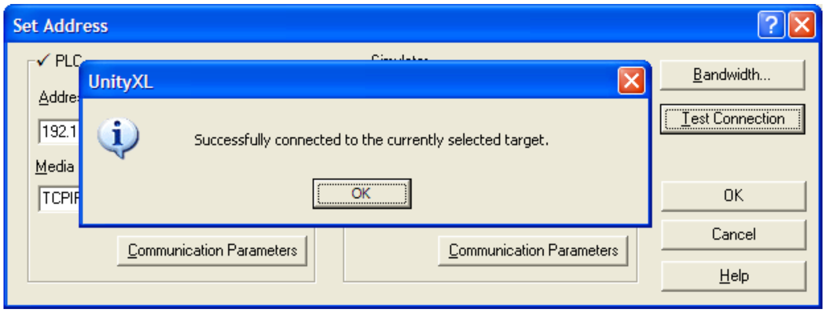
2.4.1 Connecting to the Processor with TCP/IP

The next step is to download (copy) the project file to the processor. The following steps demonstrate how to use an Ethernet cable connected from the Processor to the PC through an Ethernet hub or switch. Other connection methods may also be available, depending on the hardware configuration of the processor and the communication drivers installed in Unity Pro.

1 Connect the PC and the processor to an Ethernet hub.
2 Open the **PLC** menu, and select **SET ADDRESS**.

> **Important:** The *Set Address* dialog box is divided into two areas. Enter the address and media type in the PLC area of the dialog box, not the SIMULATOR area.

3 Enter the IP address in the address field. In the **MEDIA** dropdown list, choose 'TCPIP'.
4 Click the **TEST CONNECTION** button to verify that the settings.



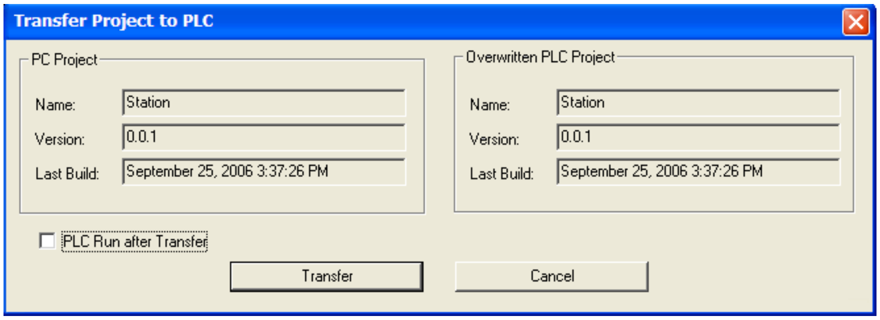
Address: 192.1

Media: TCPIP

Successfully connected to the currently selected target.

2.5 Downloading the Project to the Processor

1 Open the **PLC** menu and then choose **CONNECT.** This action opens a connection between the Unity Pro software and the processor, using the address and media type settings that were configured in the previous step.
2 On the **PLC** menu, choose **TRANSFER PROJECT TO PLC**. This action opens the **TRANSFER PROJECT TO PLC** dialog box. If the PLC needs to go to "Run" mode immediately after the transfer is complete, select (check) the **PLC RUN AFTER TRANSFER** check box.



PC Project

Name: Station

Version: 0.0.1

Last Build: September 25, 2006 3:37:26 PM

Overwritten PLC Project

Name: Station

Version: 0.0.1

Last Build: September 25, 2006 3:37:26 PM

□ PLC Run after Transfer

3 Click the **TRANSFER** button to download the project to the processor. As the project is transferred, Unity Pro reports its process in a **PROGRESS** dialog box, with details appearing in a pane at the bottom of the window.

Place the processor in Run mode when the transfer completes.

3 Configuring the Processor with Concept

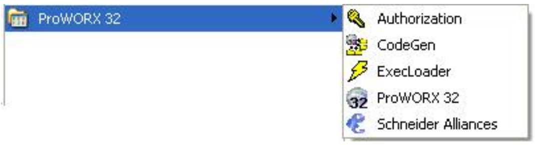
Please contact ProSoft Technical Support (+1 (661) 716-5100) for more information when configuring with Concept v2.6.

4 Configuring the Processor with ProWORX

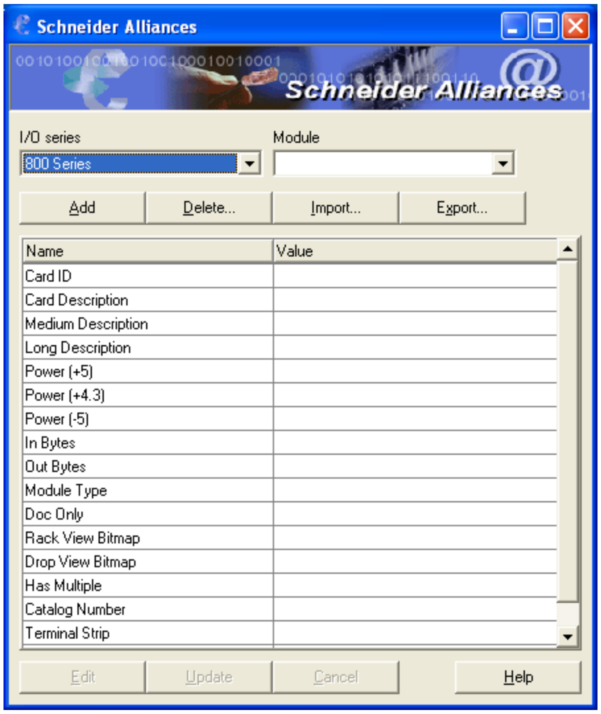
When the ProWORX 32 software is used to configure the processor, use the example SAF file provided on the ProSoft Solutions DVD.

Important Note: ProWORX software does not report whether the PTQ module is present in the rack, and therefore is not able to report the health status of the module when the module is online with the Quantum processor. Please consider this when monitoring the status of the PTQ module.

1 Run the **SCHNEIDER_ALLIANCES.EXE** application that is installed with the ProWORX 32 software:



2 Click on **IMPORT…**

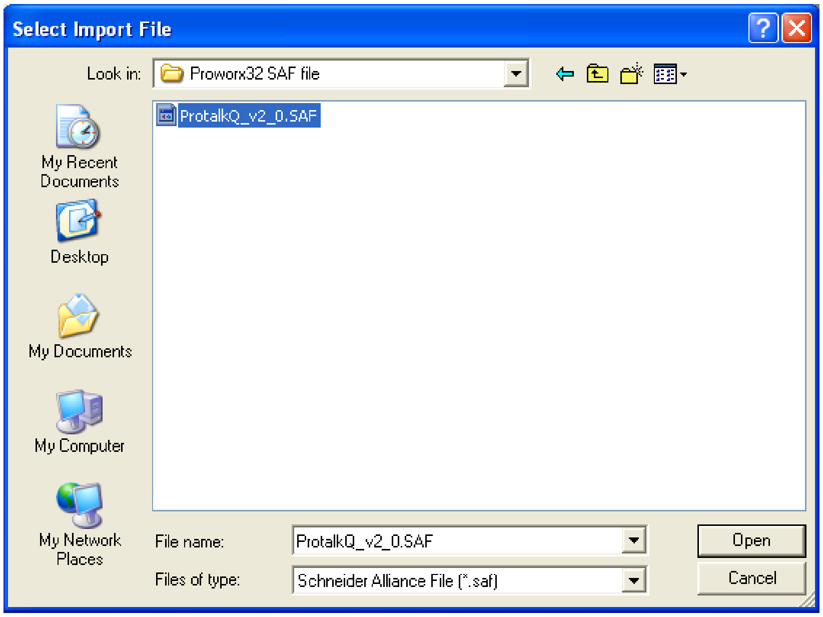


I/O series: 800 Series

Module:

Name	Value
Card ID	
Card Description	
Medium Description	
Long Description	
Power (+5)	
Power (+4.3)	
Power (-5)	
In Bytes	
Out Bytes	
Module Type	
Doc Only	
Rack View Bitmap	
Drop View Bitmap	
Has Multiple	
Catalog Number	
Terminal Strip	

3 Select the .*SAF* File that is located on the DVD shipped with the PTQ module.



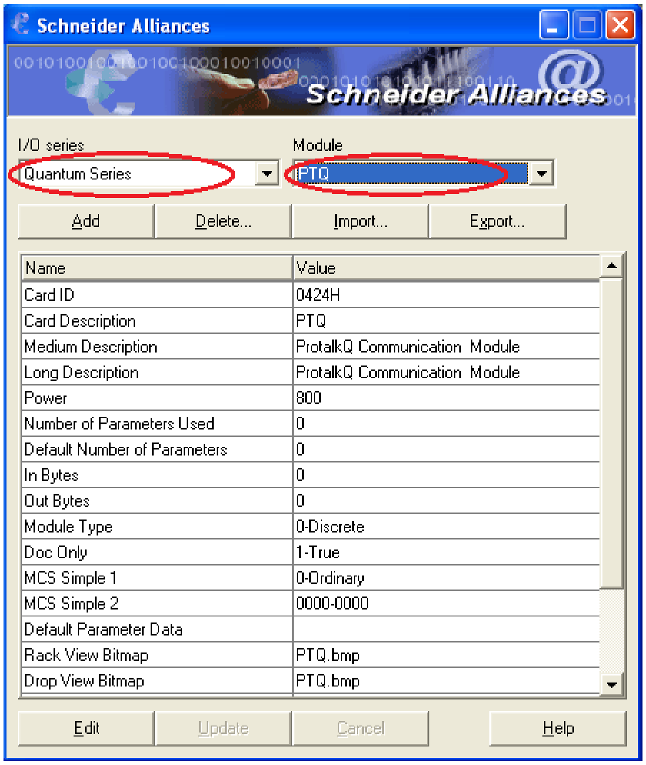
Look in: Proworx32 SAF file

ProtalkQ_v2_0.SAF

File name: ProtalkQ_v2_0.SAF

Files of type: Schneider Alliance File (*.saf)

4 After clicking **OPEN,** select the **I/O SERIES** as **QUANTUM SERIES.** Also, select the **MODULE** as **PTQ**:



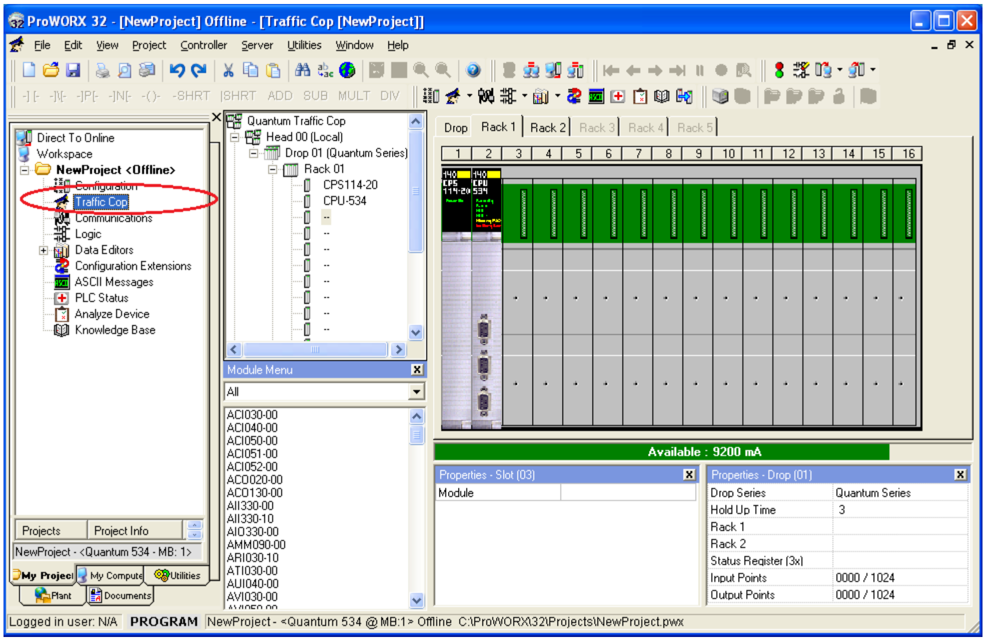
I/O series: Quantum Series

Module: PTQ

Name	Value
Card ID	0424H
Card Description	PTQ
Medium Description	ProtalkQ Communication Module
Long Description	ProtalkQ Communication Module
Power	800
Number of Parameters Used	0
Default Number of Parameters	0
In Bytes	0
Out Bytes	0
Module Type	0-Discrete
Doc Only	1-True
MCS Simple 1	0-Ordinary
MCS Simple 2	0000-0000
Default Parameter Data	
Rack View Bitmap	PTQ.bmp
Drop View Bitmap	PTQ.bmp

5 Close the Schneider Alliances application.

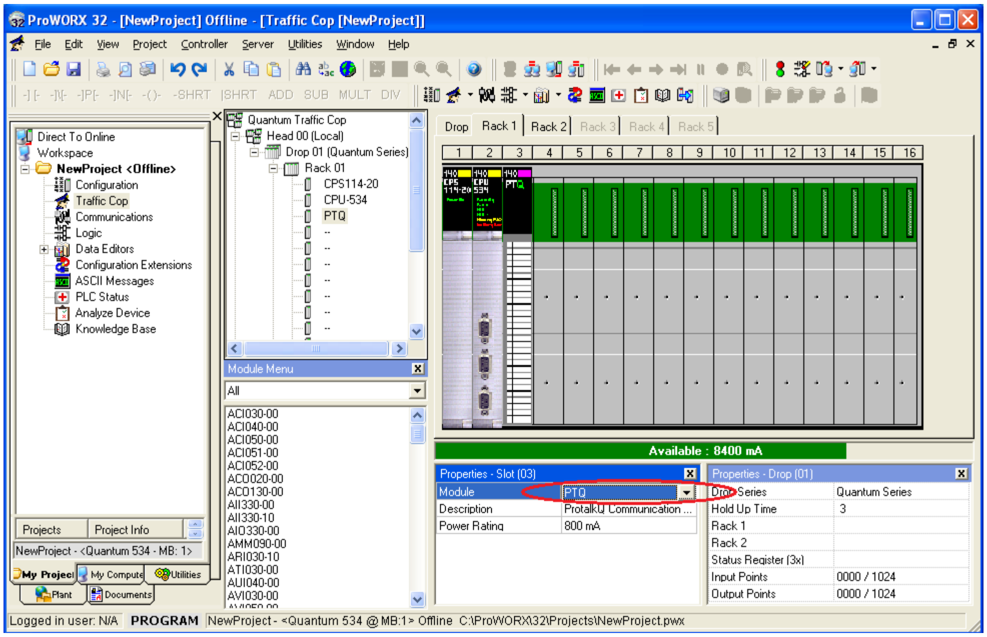
6 Run the ProWORX 32 software. Whether a new or existing project is used, click on the*Traffic Cop* section to display the rack to be edited.



Properties - Drop (01)	
Drop Series	Quantum Series
Hold Up Time	3
Rack 1	
Rack 2	
Status Register (3x)	
Input Points	0000 / 1024
Output Points	0000 / 1024

7 Highlight Slot x by clicking on the slot in the rack display. In this case, the PTQ module will be placed in slot 3.

Below the rack display is the *Properties* drop-down box for slot 3, click and select **PTQ**.



Properties - Slot (03)

Module	PTQ
Description	ProtalkQ Communication ...
Power Rating	800 mA

Properties - Drop (01)

Drop Series	Quantum Series
Hold Up Time	3
Rack 1	
Rack 2	
Status Register (3x)	
Input Points	0000 / 1024
Output Points	0000 / 1024

Logged in user: N/A PROGRAM NewProject - <Quantum 534 @ MB:1> Offline C:\ProWORX\32\Projects\NewProject.pwx

NewProject - <Quantum 534 - MB: 1>

8 Save the project. It is now ready to be downloaded to the Processor.

5 Module Configuration

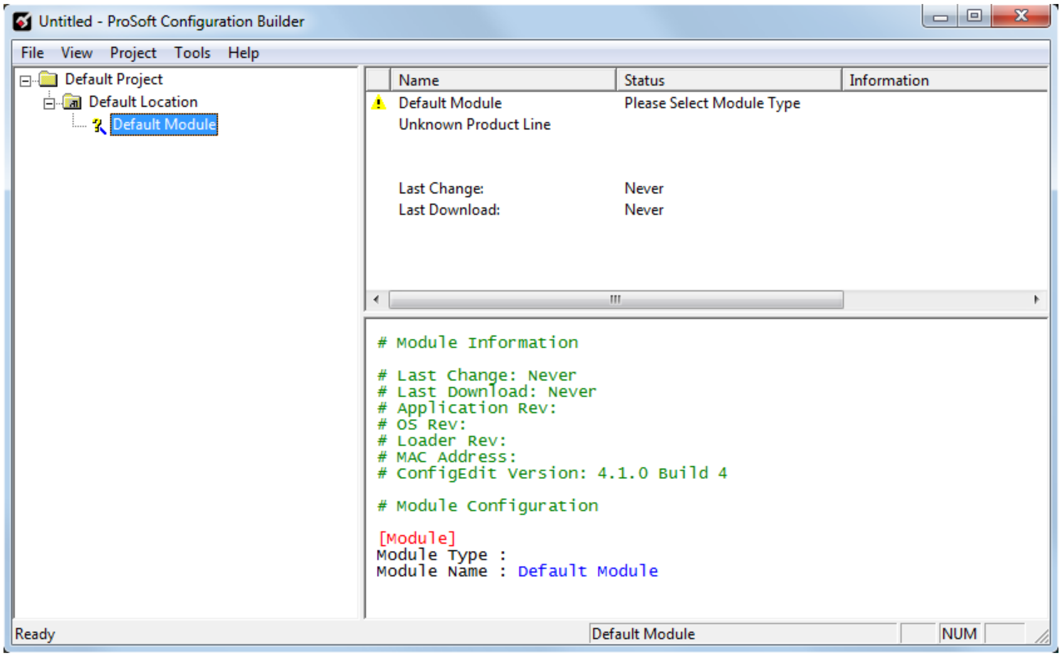
In This Chapter

5.1 Using ProSoft Configuration Builder

ProSoft Configuration Builder (PCB) provides a way to manage module configuration files customized to meet the application needs. *PCB* is not only a powerful solution for new configuration files, but also allows information to be imported from previously installed (known working) configurations to new projects.

5.1.1 Creating a Project

To begin, start ProSoft Configuration Builder. If other Windows configuration tools have been used before, the screen layout is found to be familiar. ProSoft Configuration Builder's window consists of a tree view on the left, an information pane and a configuration pane on the right side of the window. The following illustration shows the ProSoft Configuration Builder window with a new project.

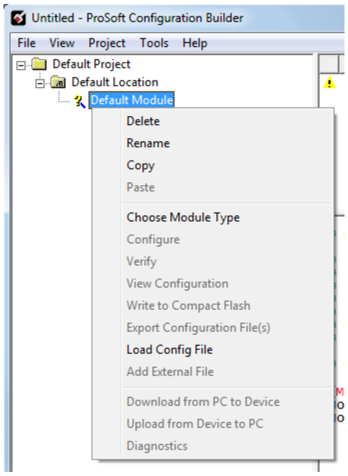


Name	Status	Information
Default Module	Please Select Module Type	
Unknown Product Line		
Last Change:	Never	
Last Download:	Never	

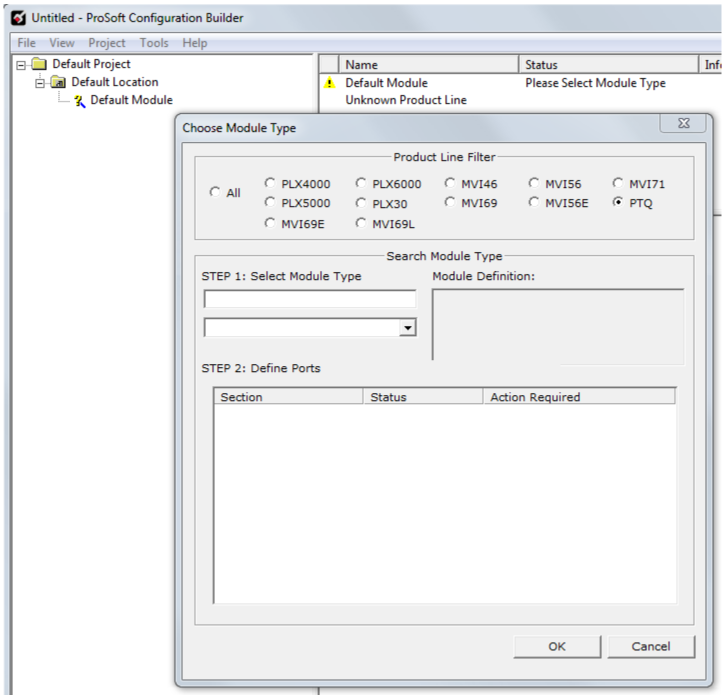
```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 4.1.0 Build 4

# Module Configuration

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

The first task is to add the PTQ-104C module to the project.

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



2 This action opens the **CHOOSE MODULE TYPE** dialog box.


Untitled - ProSoft Configuration Builder
File View Project Tools Help
Default Project
Default Location
Default Module
Name
Status
Inf

Default Module | Please Select Module Type
Unknown Product Line

Choose Module Type

Product Line Filter

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

Search Module Type

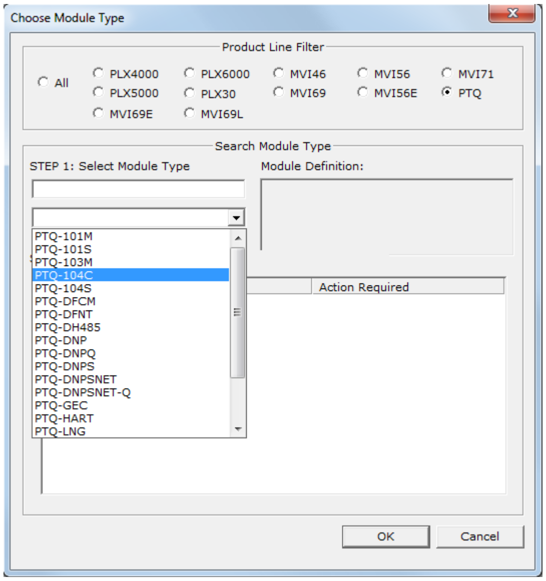
STEP 1: Select Module Type

Module Definition:

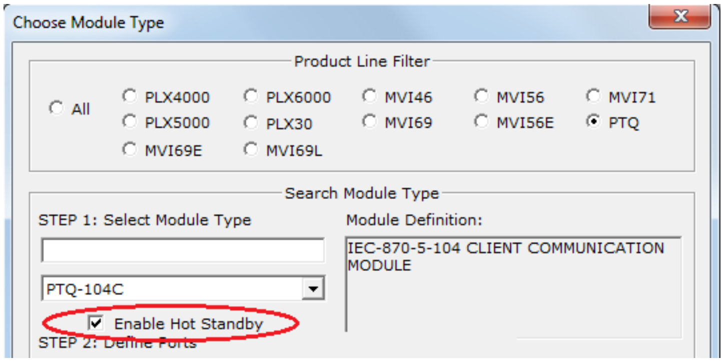
STEP 2: Define Ports

Section	Status	Action Required

3 In the **PRODUCT LINE FILTER** area of the dialog box, select **PTQ.** In the **SELECT MODULE TYPE** dropdown list, select **PTQ-104C**.



4 If Hot Standby is required, click the *Enable Hot Standby* button.

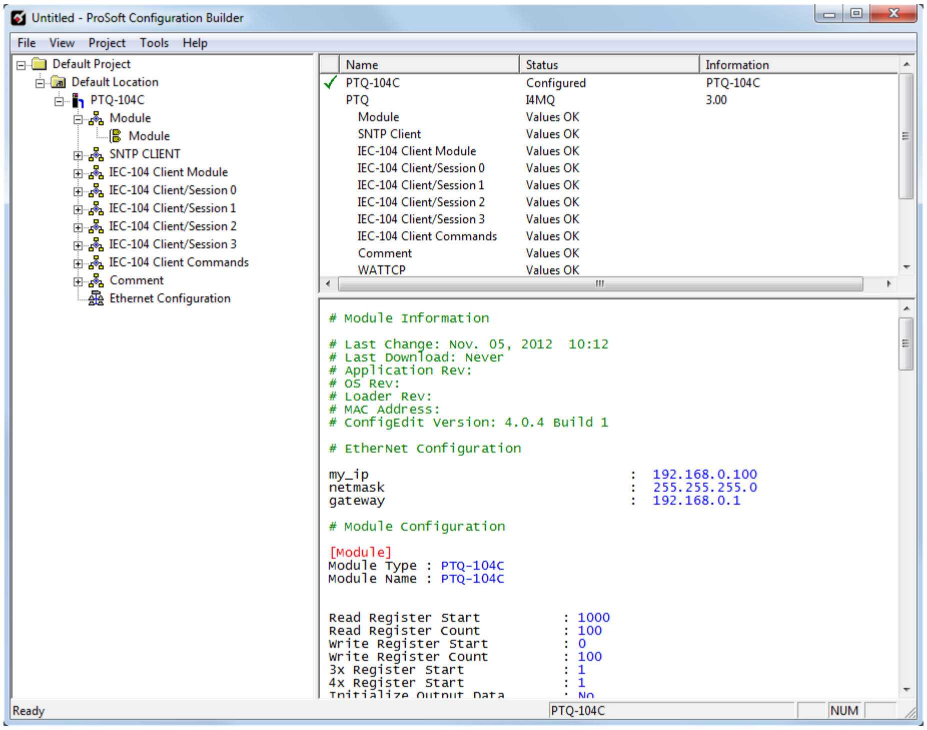


IEC-870-5-104 CLIENT COMMUNICATION MODULE

5 Click **OK** to save the settings and return to the **PROSOFT CONFIGURATION BUILDER** window.

5.1.2 Setting Module Parameters

Notice the contents of the information pane and the configuration pane updated when the PTQ-104C module was added to the project.


Untitled - ProSoft Configuration Builder
File View Project Tools Help
Ready
PTQ-104C
NUM

- Default Project
  - Default Location
    - PTQ-104C
      - Module
        - Module
      - SNTP CLIENT
      - IEC-104 Client Module
      - IEC-104 Client/Session 0
      - IEC-104 Client/Session 1
      - IEC-104 Client/Session 2
      - IEC-104 Client/Session 3
      - IEC-104 Client Commands
      - Comment
      - Ethernet Configuration

	Name	Status	Information
✓	PTQ-104C	Configured	PTQ-104C
	PTQ	I4MQ	3.00
	Module	Values OK	
	SNTP Client	Values OK	
	IEC-104 Client Module	Values OK	
	IEC-104 Client/Session 0	Values OK	
	IEC-104 Client/Session 1	Values OK	
	IEC-104 Client/Session 2	Values OK	
	IEC-104 Client/Session 3	Values OK	
	IEC-104 Client Commands	Values OK	
	Comment	Values OK	
	WATTCP	Values OK	

```
# Module Information

# Last Change: Nov. 05, 2012  10:12
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 4.0.4 Build 1

# EtherNet Configuration

my_ip                                      :  192.168.0.100
netmask                                    :  255.255.255.0
gateway                                    :  192.168.0.1

# Module Configuration

[Module]
Module Type : PTQ-104C
Module Name : PTQ-104C


Read Register Start             : 1000
Read Register Count             : 100
Write Register Start            : 0
Write Register Count            : 100
3x Register Start               : 1
4x Register Start               : 1
Initialize Output Data          : No
```

Renaming the "Default Project" and "Default Location" folders can be done in the tree view.

Renaming an object

1 Select the object and click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**
2 Type the name to assign to the object.
3 Click away from the object to save the new name.

Configuring module parameters

1 Click on the plus sign next to the icon to expand module information.
2 Double-click the icon to open the *Edit* dialog box.
3 To edit a parameter, select the parameter in the left pane and make the changes in the right pane.
4 Click **OK** to save the changes.

HSBY Note: For Hot Standby modules, a double module icon will be displayed.

Printing a configuration file

1 Select the module 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 On the *View Configuration* window, open the **FILE** menu, and choose **PRINT.** This action opens the *Print* dialog box.

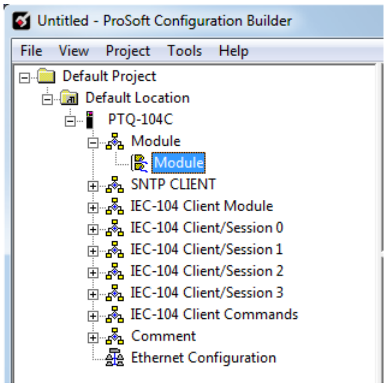
4 On the *Print* dialog box, choose the printer to use from the dropdown list, select printing options, and then click **OK.**

5.2 Backplane Data Transfer

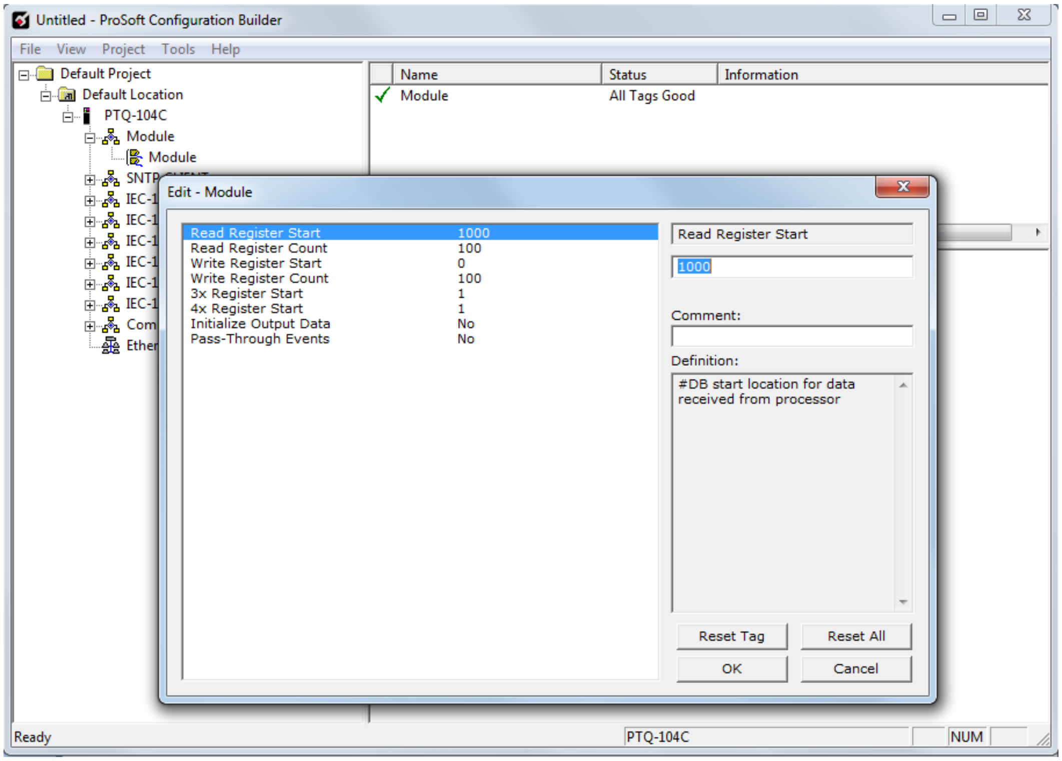
On every scan, the PTQ-104C transfers all Read and Write database values between the module and the processor.

The module will hold the processor scan for a certain period of time to complete the transfer. Therefore, the larger the read and write areas, the longer the processor scan time will be.

The [Module] section of PCB defines the starting registers for read and write operations. It also defines the length of each data area.



Double click the [Module] icon to edit.



Read Register Start	1000
Read Register Count	100
Write Register Start	0
Write Register Count	100
3x Register Start	1
4x Register Start	1
Initialize Output Data	No
Pass-Through Events	No

Read Register Start

1000

Comment:

Definition:

#DB start location for data received from processor

5.2.1 Read Register Start

0 through 3999

Database start register to move to processor

5.2.2 Read Register Count

0 through 3999

Number of words moved from module to processor

5.2.3 Write Register Start

0 through 3999

Database start register where data placed from processor

5.2.4 Write Register Count

0 through 3999

Number of words moved from processor to module

5.2.5 3X Register Start

1 through 65535

3x start register where data moved from module to processor

5.2.6 4X Register Start

1 through 65535

4x start register where data moved from processor to module

5.2.7 Initialize Output Data

Yes or No

This parameter determines if the output data for the module should be initialized with values from the processor. If the value is set to No (0), the output data will be initialized to 0. If the value is set to Yes (1), the data will be initialized with data from the processor. Use of this option requires associated ladder logic to pass the data from the processor to the module.

5.2.8 Pass-Through Events

Yes or No

This parameter specifies if event messages received on the clients will be passed to the processor. If the parameter is set to N, event messages will not be passed to the processor. If the parameter is set to Y, the module will pass all events received to the processor using block identifier 9903.

5.3 Data Exchange

The module transfers all Read and Write data between the module and the processor on every scan. Up to 4000 total words can be configured in the module database. The more data that is configured, the longer the processor scan will take.

The [Module] section of PCB defines the starting registers for read and write operations. It also defines the number of registers to use for each data area.

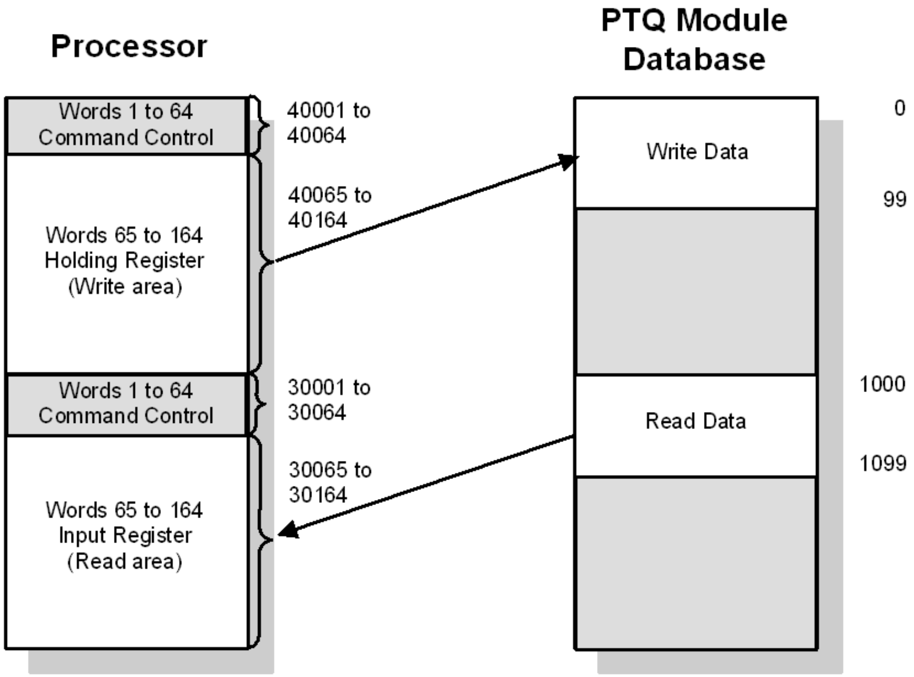
The values in the example PCB configuration file from the previous section are illustrated in the following diagram.

Read Register Start: 1000

Read Register Count: 100

Write Register Start: 0

Write Register Count: 100


Processor
PTQ Module Database
Words 1 to 64 Command Control
Words 65 to 164 Holding Register (Write area)
Words 1 to 64 Command Control
Words 65 to 164 Input Register (Read area)
40001 to 40064
40065 to 40164
30001 to 30064
30065 to 30164
Write Data
Read Data
0
99
1000
1099

Note: The diagram above shows the memory addresses for a Quantum processor. If using a Unity processor, substitute %MW for read only data, and %IW for read/write data.

Words 0 through 63 in each read/write block are reserved for special functions. Refer to Command Control (Page 50) for more information on special function blocks. The following table shows the relationship between the processor memory and the module database areas.

Module Database	Register	Unity Register	Description
Read Data	3x	%IW	Input Register
Write Data	4x	%MW	Holding Register

The data mapping in the following example shows the relationship between processor and PTQ-104C memory addresses, assuming a 4x register start value of 40001 and a PTQ-104C database start value of 0.

Processor Memory Address	Module Database Address
40065	0
40066	1
40067	2
40068	3

Processor Memory Address	Module Database Address
40069	4
…	…
40164	99

The data mapping in the following example shows the relationship between processor and PTQ-104C memory addresses, assuming a 3x register start value of 30001 and a PTQ-104C database start value of 2000.

Processor Memory Address	Module Database Address
30065	2000
30066	2001
30067	2002
30068	2003
30069	2004
…	…
30164	2099

5.3.1 Data Type Mapping and Addressing

When interfacing data in the processor to that of the IEC 60870-5-104 protocol, it is important the user understands the mapping of the data types to their corresponding representation in the modules database. The table that follows lists the data types supported by the module and their associated storage representation.

IEC-870-5-104 Data Types

Type ID	Type	Description	Data representation
1	M_SP_NA_1 (7.3.1.1)	Monitored Single-point Information: This data type stores a single binary input point. Associated time-tagged event information for this type are M_SP_TA_1 (2) and M_SP_TB_1 (30).	Single bit value (7.2.6.1) with 0=Off and 1=On.
3	M_DP_NA_1 (7.3.1.3)	Monitored Dual-point Information: This data type stores a dual-point binary input value (that is, valve status). Associated time-tagged event information for this type are M_DP_TA_1 (4) and M_DP_TB_1 (31).	Dual-bit status (7.2.6.2) with 00b (0 decimal) = indeterminate or intermediate, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = indeterminate.
5	M_ST_NA_1 (7.3.1.5)	Monitored Step-point Information: This data type is used for step position of transformers or other step position information. The value for the position ranges from -64 to 63. Associated time-tagged event information for this type are M_ST_TA_1 (6) and M_ST_TB_1 (32).	Step data (7.2.6.5) is stored in a single character value with bits 0 to 6 (-64 to +63) representing the step position and bit 7 representing the following states: 0 = Equipment is not in transient state 1 = Equipment in transient state

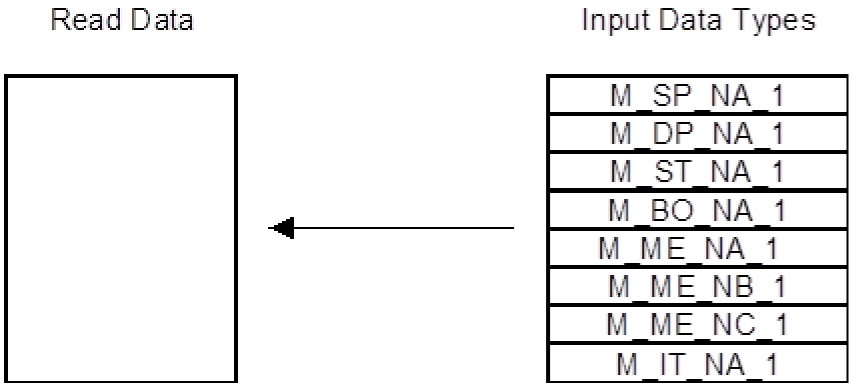
Type ID	Type	Description	Data representation
7	M_BO_NA_1 (7.3.1.7)	Monitored Bitstring of 32-bit data --This data type stores 32-bit data in binary form. Each bit in the string has a value of 0 or 1. Associated time-tagged event information for this type are M_BO_TA_1 (8) and M_BO_TB_1 (33).	Each of the 32 bits in the bitstring has a value of 0 or 1 (7.2.6.13).
9	M_ME_NA_1 (7.3.1.9)	Monitored Normalized Measured Value: This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TA_1 (10) and M_ME_TD_1 (34).	Normalized values (7.2.6.6) are stored in a word (16-bit) data area with a range of -1..+1-2^{-15}
11	M_ME_NB_1 (7.3.1.11)	Monitored Scaled Measured Value --This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TB_1 (12) and M_ME_TE_1 (35).	Scaled values (7.2.6.7) are stored in a word (16-bit) data area with a range of -2^{15}.. +2^{15}-1
13	M_ME_NC_1 (7.3.1.13)	Monitored Measured Value, Short Floating-Point Number: This data type is used for analog input data stored in floating point format according to the IEEE STD 754, QDS format. Associated time-tagged event information for this type are M_ME_TC_1 (14) and M_ME_TE_1 (36).	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign) (7.2.6.8)
15	M_IT_NA_1 (7.3.1.15)	Monitored Integrated Total-point Information --This data type stores meter or other count data. Associated time-tagged event information for this type are M_IT_TA_1 (15) and M_IT_TB_1 (37).	Binary counter data (7.2.6.9) is stored in a double-word (32-bit) value with a range of -2^{31}..+2^{31}-1.
45	C_SC_NA_1 (7.3.2.1)	Single-point Command: This command controls a single binary point such as a relay.	Single bit value (7.2.6.15) with 0 = Off and 1 = On
46	C_DC_NA_1 (7.3.2.2)	Double-point Command: This command controls a dual-point binary control device such as a trip/close relay.	Double Command (7.2.6.16) with 0 = Not permitted 1 = Off 2 = On 3 = Not permitted
47	C_RC_NA_1 (7.3.2.3)	Regulating Step Command: This command controls a stepping device such as a transformer.	Regulating Step Command (7.2.6.17) with 0 = Not permitted 1 = Next step lower 2 = Next step higher 3 = Not permitted
48	C_SE_NA_1 (7.3.2.4)	Setpoint Command, Normalized Value: This command controls an analog device.	Normalized values (7.2.6.6) are stored in a word (16-bit)data area with a range of -1..+1-2^{-15}
49	C_SE_NB_1 (7.3.2.5)	Setpoint Command, Scaled Value: This command controls an analog device.	Scaled values (7.2.6.7) are stored in a word (16-bit) data area with a range of -2^{15}.. +2^{15}-1
50	C_SE_NC_1 (7.3.2.6)	Setpoint Command, Short Floating-Point Format: This command controls an analog device accepting an IEEE STD 754 floating-point format value.	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign) (7.2.6.8)
51	C_BO_NA_1 (7.3.2.7)	Setpoint Command, 32-bit Bitstring: This command controls a bitstring in a device.	Each of the 32 bits in the bitstring has a value of 0 or 1 (7.2.6.13).
100	C_IC_NA_1	Group interrogation command	NA
101	C_CI_NA_1	Counter interrogation command	NA
102	C_RD_NA_1	Read command	NA
103	C_CS_NA_1	Clock Synchronization	NA
105	C_RP_NA_1	Reset process command	NA
107	C_TS_TA_1	Test command (104 standard)	NA
110	P_ME_NA_1	Parameter, normalized measured value	Word address

Type ID	Type	Description	Data representation
111	P_ME_NB_1	Parameter, scaled measured value	Word address
112	P_ME_NC_1	Parameter, short float value	Double-word address
113	P_AC_NA_1	Parameter activation command	NA

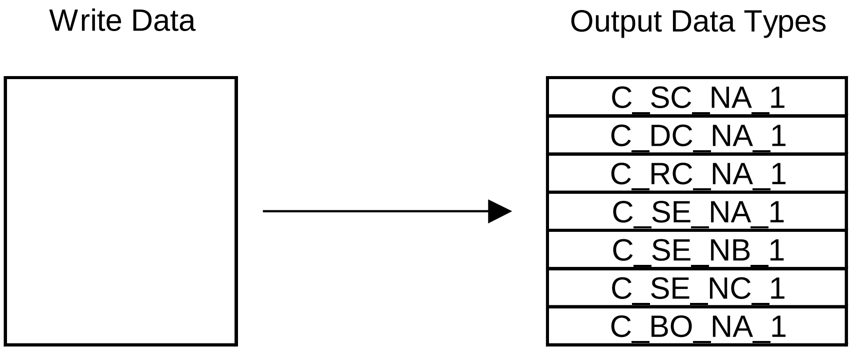
The data addressing is resumed in the following table.

Data	Size	Example
Single Point	1 bit	Address 1600 refers to word 100, bit 1 in database
Dual Point	2 bits	Address 1600 refers to word 100, bits 1 and 2 in database
Step Point	1 byte	Address 200 refers to word 100, lower byte in database
Bitstring 32 bit	2 words	Address 50 refers to word 100 and 101 in database
Normalized Measured Value	1 word	Address 100 refers to word 100 in database
Scaled Measured Value	1 word	Address 100 refers to word 100 in database
Short Float Point Measured Value	2 words	Address 50 refers to words 100 and 101 in database
Integrated Total Point	2 words	Address 50 refers to words 100 and 101 in database

Another important concept to understand is the direction of data transfer for the different data types with reference to the controller. The following illustration shows the data types (monitored data) that are transferred from the module to the processor.



The next diagram shows the movement of control data from the processor to the module. This data is then sent to the controlled devices on the serial networks.



As blocks are transferred between the module and the processor, each block contains block identification codes that define the content or function of the block of data transferred. The control blocks used by the module are displayed in the following table.

Block Range	Descriptions
9250	Status Data
9251	Client X Status Data
9901	User Constructed Command
9902	Command Control Block (Add command to Command List Queue)
9903	Event Messages
9950	Command List Error data
9970	Set PLC time using module’s time
9971	Set module’s time using PLC time
9998	Warm Boot Request from PLC (Block contains no data)
9999	Cold Boot Request from PLC (Block contains no data)

Block identification codes 9901 to 9999 are used for special control blocks to control the module. Each of these blocks is discussed in the following topics.

Normal Data Transfer Blocks

These data are transferred through read (input image) and write (output image) blocks. Refer to Module Configuration for a description of the data objects used with the blocks and the ladder logic required. The following topics discuss the structure and function of each block.

Input Data (3x Register Data)

These blocks of data transfer information from the module to the Quantum processor. The following table describes the structure of the input image.

Offset	**Description**	**Length**
0	Sequence Counter	1
1	Block ID	1
2 to 63	Command Response Data	62
64 to n	Read Data	0 to 3999

Output Data (4x Register Data)

These blocks of data transfer information from the Quantum processor to the module. The following table describes the structure of the output image.

Offset	**Description**	**Length**
0	Sequence Counter	1
1	Block ID	1
2 to 63	Command Data	62
64 to n	Write Data	0 to 3999

Command Control Blocks

Block identification codes greater than 9900 are utilized to perform special functions in the module. Each control block recognized and used by the module is defined in the following topics.

Command Codes	Descriptions
9250	Status Block
9251	Client X Status Data
9901	User Constructed Command
9902	Command Control Block (Add command to Command List Queue)
9903	Event Messages from Master port
9950	Command List Error data
9970	Set PLC time using module's time
9971	Set module's time using PLC time
9999	Cold Boot Request from PLC (Block contains no data)

Note: The command code in the I/O area is also referred to as the block ID.

Implementing Ladder to Support Special Functions

In order to use Special Functions, some form of control logic must be implemented. The following section uses structured text language to illustrate how a typical function might be implemented.

Example: Rebooting the Module.

MyTrigger is a variable that triggers this logic

*OutputContro*l variable array starts at register 4000001

The first instruction guarantees that the processor requests this block for only one scan.

The second instructions sets the Block Number (9999 = ColdBoot) and then sets the sequence number to 1.

```
IF MyTrigger>0 AND OutputControl1[1]> 0 THEN
 OutputControl1[0]:= InputData[0];
 OutputControl1[1]:=0;
  MyTrigger :=0;
END_IF;

IF (MyTrigger=9999)OR (MyTrigger=9998) OR (MyTrigger=9250) THEN
 OutputControl1[1] :=MyTrigger;
 Temp:=WORD_TO_INT(OutputControl1[0]);
 Temp:=Temp+1;
 OutputControl1[0]:=INT_TO_WORD(Temp);

 END_IF;
```

Example: Retrieving the time of day from the module.

This logic shows an example on how to request a block 9970 from the module (Read Module's Time) and read the response to the processor.

Assumptions:

- MyTrigger is a variable that triggers this logic
- OutputControl variable array starts at register 400,001
- InputControl variable array starts at register 300,001
- MyTime variables store the date and time values to be read from the module

Sets the Block Number (9970=Read Module's Time) and then increments the output sequence number (OutputControl[1]) by one. Once the module reads a new output sequence number from the processor it will process this request. So remember that the actual trigger is moving a new output block sequence number value to the module. Moving the block number (9970) is not the trigger to request this task from the module.

MyTrigger is set to -1 as an indication that the logic is waiting for the response from the module.

```
IF (MyTrigger=9970) THEN
 OutputControl1[1] :=MyTrigger;
 Temp:=WORD_TO_INT(OutputControl1[0]);
 Temp:=Temp+1;
 OutputControl1[0]:=INT_TO_WORD(Temp);

END_IF;
```

When the request is processed, the module will send the block response and increment the received output sequence number by 1. So the output sequence number is one less than the input sequence number the module has sent a new block. Once the block is received the processor logic copies the received data to the appropriate variables. The logic also clears the trigger for the next request.*)

```
IF (InputData[1]=9970) THEN
GetTime.MyYear :=InputData[2];
GetTime.MyMonth :=InputData[3];
GetTime.MyDay :=InputData[4];
GetTime.MyHour :=InputData[5];
GetTime.MyMinute :=InputData[6];
GetTime.MySecond :=InputData[7];
GetTime.MyMillisecond :=InputData[8];
END_IF
```

Example: Setting the time of day to the module.

This logic shows an example on how to request a block 9971 from the module (Read Module's Time).

Assumptions:

- MyTrigger is a variable that triggers this logic
- OutputControl variable array starts at register 400,001
- InputControl variable array starts at register 300,001
- MyTime variables store the date and time values to be written to the module

Sets the Block Number (9971=Write Module's Time) and then increments the output sequence number (OutputControl[1]) by one. Once the module reads a new output sequence number from the processor it will process this request. So remember that the actual trigger is moving a new output block sequence number value to the module. Moving the block number (9970) is not the trigger to request this task from the module. MyTrigger is set to -1 as an indication that the logic is waiting for the response from the module.

```
IF (MyTrigger=9971) THEN
OutputControl1[1] :=MyTrigger;
OutputControl1[2] :=SetTime.MyYear;
OutputControl1[3] :=SetTime.MyMonth;
OutputControl1[4] :=SetTime.MyDay;
OutputControl1[5] :=SetTime.MyHour;
OutputControl1[6] :=SetTime.MyMinute;
OutputControl1[7] :=SetTime.MySecond;
OutputControl1[8] :=SetTime.MyMillisecond;
Temp:=WORD_TO_INT(OutputControl1[0]);
Temp:=Temp+1;
OutputControl1[0]:=INT_TO_WORD(Temp);

END_IF;
```

Once the request was processed the module will send the block response and increment the received output sequence number by 1. So the output sequence number is one less than the input sequence number the module has sent a new block. *).

5.3.2 Read Status (Block 9250)

This block is used to request status data from the module by the processor.

Block Format from Processor (4x Register Data) sent to module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the user wishes to request a new command block.
1	Block ID	This field contains the block identification code of 9250 for the block.
2 to 63	Spare	Not used.

Block Format from Module (3x Register Data) from the module to Processor

Word Offset in Block	Data Field	Description
0	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
1 to 2	Product Name	This two-word data area contains the text values representing the product name. These words contain the text 'I4MQ' for the PTQ platform.
3 to 4	Revision	This two-word data area contains the text values for the revision number.
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
10	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
11	Reserve	This word is reserved for future use.
12	Error Blk Cnt	This word contains the total number of block transfer errors.
13	Event Msg Cnt	This word contains the number of event messages waiting to send to the processor.
14	Event Msg Overflow	This word contains a value of 0 if the event message buffer has not overflowed. If the event buffer overflows, this word will be set to a value of 1.
15	Session Count	This word contains the number of sessions configured in the module.
16	Current Cmd	This word contains the index of the current command being executed in the command list.
17	Cmd Busy Flag	This word is set to zero if no command is currently being executed and waiting on a response. If the word is set to 1, a command is currently executing.
18	Cmd Count	This word contains the count of the number of commands configured for the module.
19	Cmd Delay	This word contains the command delay counter preset. There is a fixed delay between each command to permit the module to perform other operations.
20	Cmd Queue	This word is set to zero if the command executing is from the command list. If the executing command is from the command queue, the word will be set to 1.
21	Cmd Queue Count	This word contains the number of active commands in the command queue for the module. Up to 100 commands can be buffered in this queue. These commands are transferred from the processor to the module using special command blocks.
22 to 23	Online Status	This double word value contains a bit for each of the 32 potential sessions in the module. If the bit is set for a session in the double word, the station is online. If the bit is clear, the station is offline. Use this value to determine if commands sent from the processor will have a chance of succeeding.
24	SNTP Valid	NTP time is valid (0=No, 1=Yes)

25	NTP Request	Number of requests to NTP server
26	NTP Response	Number of responses from NTP server
27	SNTP Computation	Number of times SNTP time computed
28	SNTP Set	Number of times SNTP time set
29	NTP Timeout	Number of NTP response timeouts

5.3.3 User-Constructed Command Block (9901)

Block identification code 9901 is used to issue one or more user-constructed commands. When the module receives a block 9901 identification code, it will place the included commands into the command queue.

Block Format from Processor (4x Register Data) to module

Word Offset in Block	**Data Field(s)**	**Description**
0	Sequence Counter	This field contains a new value each time the user wishes to request a new command block.
1	Block ID	This field contains the block identification code of 9901 for the block.
2	Command Count	This field defines the number of user commands contained in the block. The valid range for the field is 1 to 10.
3 to 8	Command #1	Data required to build the user defined command in the command queue.
9 to 14	Command #2	Data required to build the user defined command in the command queue.
15 to 20	Command #3	Data required to build the user defined command in the command queue.
21 to 26	Command #4	Data required to build the user defined command in the command queue.
27 to 32	Command #5	Data required to build the user defined command in the command queue.
33 to 38	Command #6	Data required to build the user defined command in the command queue.
39 to 44	Command #7	Data required to build the user defined command in the command queue.
45 to 50	Command #8	Data required to build the user defined command in the command queue.
51 to 56	Command #9	Data required to build the user defined command in the command queue.
57 to 62	Command #10	Data required to build the user defined command in the command queue.
63	Spare	Not used.

The following fields are used for each 6-word record in the command list:

Word Offset	Definitions	Description
0	Database Index	Address in module to associate with the command
1	Session Index	Session index defined in the module to associate with the command.
2	Sector Index	Sector index for session as defined in the module.
3	Data Type	ASDU data type associated with the command.
4	Point Index	Information object address for the point on which command operates.
5	Qualifier	Qualifier as defined for the command list. This parameter is data type dependent.

Block Format from Module (3x Register Data) to Processor

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9901 for the block.
2 to 63	Spare	Not used.

5.3.4 Command Control Block (9902)

The block 9902 identification code is used by the PLC to send a list of commands to be placed in the command queue. Commands placed in the queue with this method need not have their enable bit set.

Block Format from Processor (4x Register Data) to module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the user wishes to request a new command block.
1	Block ID	This field contains the value of 9902 identifying the enable command to the module.
2	Command count	This field contains the number of commands to enable in the command list. Valid values for this field are 1 to 60.
3 to 62	Command Numbers to enable	These 60 words of data contain the command numbers in the command list to enable. The commands in the list will be placed in the command queue for immediate processing by the module. The first command in the list has an index of 0.
63	Spare	Not Used

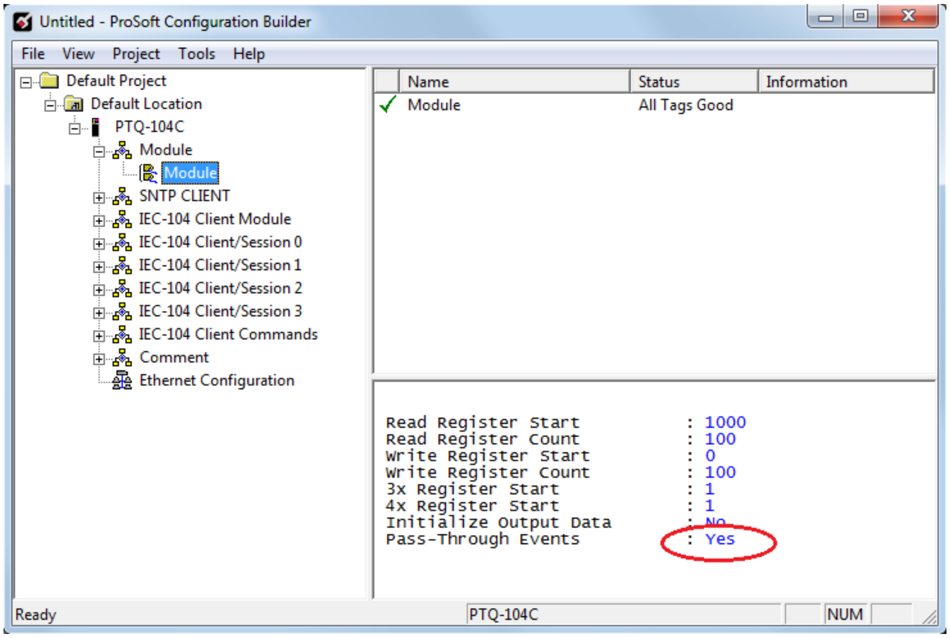
Block Format from Module (3x Register Data) to Processor

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9902 for the block.

5.3.5 Event Pass-Through Block (9903)

The event pass-through functionality allows the module to pass events to the processor after they are received from the IEC-8707-5-101 slave devices. Events are considered messages associated to supported ASDU types that contain timestamp (Hour:Minute:Seconds:Milliseconds).

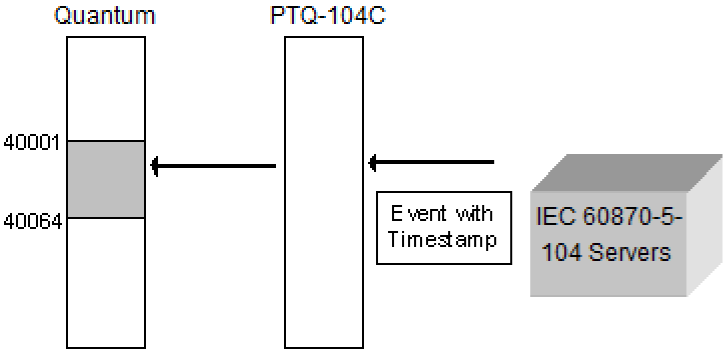
The event pass-through functionality must be initially enabled by the user in PCB. The 'Pass-Through Events' parameter will need to be set to 'Yes'.



Name	Status	Information
✓ Module	All Tags Good	

```
Read Register Start        : 1000
Read Register Count        : 100
Write Register Start       : 0
Write Register Count       : 100
3x Register Start          : 1
4x Register Start          : 1
Initialize Output Data     : No
Pass-Through Events        : Yes
```

The following illustration shows the basic idea of the event pass-through functionality. When the module receives the event from the remote device, it will build block 9903, which will be copied to the processor at the configured memory address:



Event Pass-Through Block Format

The block that is copied from the module to the processor has the following format. Each block can contain up to 4 events. The number of events per block will typically depend on the rate between how fast the module receives the events and how fast these can be passed to the processor (typically depends on the processor scan rate).

Block Format for Read

Block Format from Module (%IW or 3x Register Data)

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9903 for the block.
2	Event Count	This field contains the number of events present in the block. Values of 1 to 4 are valid.
3 to 16	Event 1	Event message
17 to 30	Event 2	Event message
31 to 44	Event 3	Event message
45 to 58	Event 4	Event message
59 to 61	Spare	Not used
62	Event count in queue	Number of messages left in event buffer
63	Event Overflow	Event buffer overflow

The format of each 14 word data region in the block is shown in the following table.

Word Offset	Definitions	Description
0	Session Index	This field contains the session index used to define the controlled unit in the module from which the event was generated.
1	Sector Index	This field contains the sector index used to define the database within the controlled unit from which the event was generated.
2	COT	This field contains the COT for the event message received from the IED. If the size of the COT is a single byte, the originator address will always be zero. The COT is in the LSB and the originator address is in the MSB.
3	Reserved	This field is reserved for future use and is added here to keep the structure double-word aligned for all platforms.
4 to 5	Point Index	This field contains the point index in the remote device that generated the event.
6	ASDU Type	This field contains the ASDU type code for the data contained in the message.
7	Milliseconds and Seconds	This word contains the seconds and milliseconds when the event occurred.
8	Minutes and Hours	This field contains the minutes and hours the event occurred.

Word Offset	Definitions	Description
9	Month and Day	This field contains the month and day of the month the event occurred.
10	Year	This field contains the year the event occurred.
11	Qualifier	This field contains the point qualifier, quality or sequence value as described in the protocol specification.
12 to 13	Value	This field contains the double word value for the point associated with the event message.

The processor logic should recognize the event count value greater than zero and read all events in the block. This value should be reset to zero to prepare the logic for the next incoming block.

5.3.6 Read Command Error List Block (9950)

Block 9950 requests the Command List Error Table from the module. The following table describes the format of the block.

Block Format from Processor (4x Register Data) to module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the user wishes to request a new command block.
1	Block ID	This field contains the value of 9950 identifying the block type to the module.
2	Number of Commands to report	This field contains the number of commands to report in the response message. The value has a range of 1 to 60.
3	Start Index of First Command	This parameter sets the index in the command list where to start. The first command in the list has a value of 0. The last index in the list has a value of MaxCommands - 1.
4 to 63	Spare	Not Used

Block Format from module to Processor:

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the value of 9950 identifying the block type to the PLC.
2	Number of Commands reported	This field contains the number of commands contained in the block that need to be processed by the PLC. This field will have a value of 1 to 60.
3	Start Index of First Command	This field contains the index in the command list for the first value in the file. This field will have a value of 0 to MaxCommands-1.
4 to 63	Command List Errors	Each word of this area contains the last error value recorded for the command. The command index of the first value (offset 4) is specified in word 3 of the block. The number of valid command errors in the block is set in word 2 of the block. Refer to the command error list to interpret the error codes reported.

5.3.7 Get Module Date and Time Block (9970)

Block 9970 requests the module's date and time. This data can be used to set the PLC clock.

5.3.8 Set Module Time Block (9971)

Block identification code 9971 is used to pass the clock time in the PLC to the module. The date and time provided will be used to set the module's clock.

Block Format from Processor (4x Register Data) to module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the user wishes to request a new command block.
1	Block ID	This field contains the block identification code of 9971 for the block.
2	Year	This field contains the four-digit year for the new time value.
3	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
4	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
5	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
6	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
8	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
9 to 63	Spare	Not Used

Block Format from Module (3x Register Data) to Processor

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9971 for the block.
2 to 63	Spare	Not Used

5.3.9 Cold Boot Block (9999)

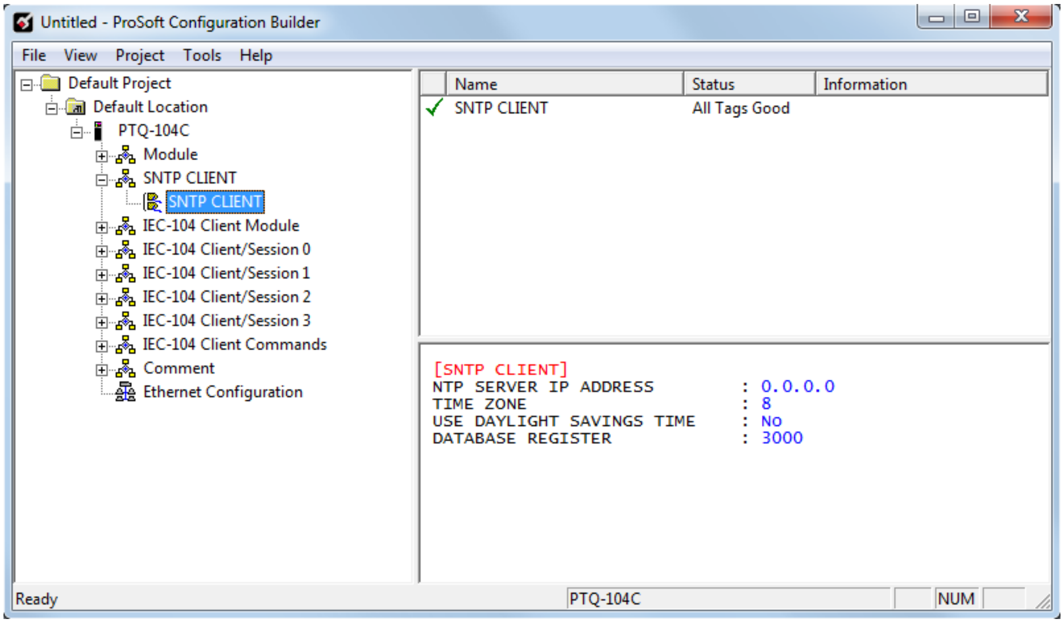
Block 9999 performs a cold-boot operation on the module. The following table describes the format of the block constructed by the processor.

Offset	Description	Length
0	9999	1
1 to 63	Spare	63

The warm and cold boot processes perform the same operation. Many of the variables that must be initialized are fixed when the module first boots. They cannot be changed after the application starts.

5.4 [SNTP CLIENT]

The [SNTP CLIENT] section of PCB is used to specify the parameters for the Simple Network Time Protocol (SNTP) Client provided with the protocol driver. This client is read in order to keep the driver's internal clock set correctly. This version of the driver supports SNTP Revision 3 and stratum between 1 and 14.



Name	Status	Information
✓ SNTP CLIENT	All Tags Good	

```
[SNTP CLIENT]
NTP SERVER IP ADDRESS        : 0.0.0.0
TIME ZONE                    : 8
USE DAYLIGHT SAVINGS TIME    : No
DATABASE REGISTER            : 3000
```

The SNTP driver will compute a new clock value every 5 minutes using the average value of 10 samples each collected over an approximate 6-second period. This new value will be used to adjust the clock maintained by the SNTP driver and used by the application.

If a valid *database register* is specified, the driver will place the time value into the module's database. The first two registers will contain the number of seconds and the next two registers will contain the number of microseconds since January 1, 1970.

A list of some of the common NTP servers can be obtained at http://www.ntp.org/

or, http://support.ntp.org/bin/view/Servers/WebHome

Other server lists can be found by searching the World Wide Web for "NTP Servers".

5.4.1 NTP Server IP Address

Enter in dotted notation

This parameter sets the IP address of the NTP server to utilize for time acquisition. Select an NTP server with the greatest accuracy that can be accessed all the time from the network. Setting this IP address to 0.0.0.0 disables SNTP server requests.

5.4.2 Time Zone

-11 to 11

This parameter specifies the time zone offset to be used from the UTC time zone. A value of zero uses UTC time. If the value entered is positive, the time zone is west of the UTC time zone (that is, Eastern Standard Time is 5). If the value entered is negative, the time zone is east of the UTC time zone (that is, Continental Europe is -1).

5.4.3 Use Daylight Savings Time

Yes or No

This parameter specifies if daylight savings time will be used in the time computation.

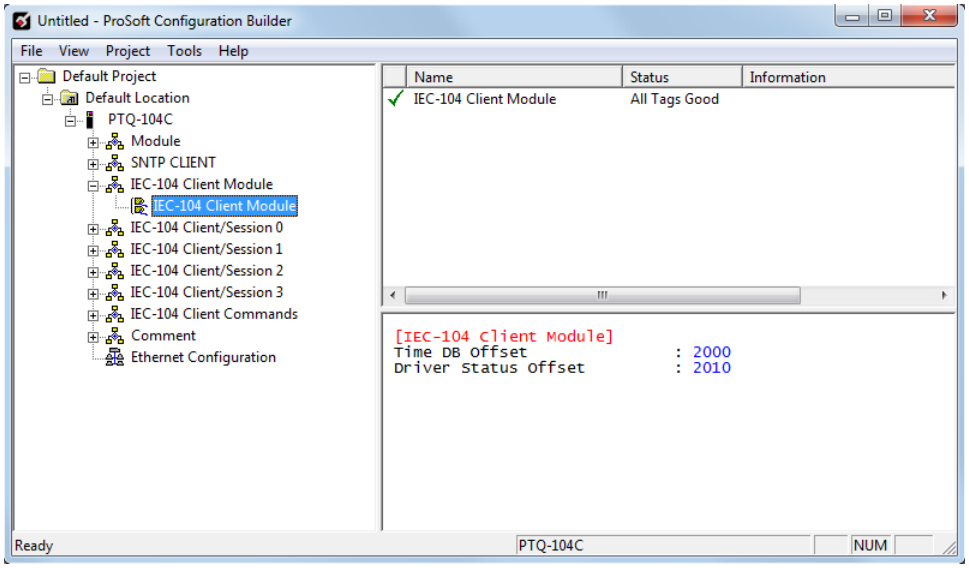
5.4.4 Database Register

-1 or 0 to 3992 as an even value

This parameter specifies if the NTP time computed by the driver is to be placed into the module’s database. If a value of -1 is specified, the time will not be placed into the database. If the value is between 0 and 3992, the time will be placed in the database. The first 4 bytes will represent the seconds since 1/1/1970, and the second 4 bytes will represent the number of microseconds. An even value should be used for the register value in order for the data to be stored correctly.

5.5 [IEC-104 Client Module]

The [IEC-104 Client Module] section of PCB includes the general configuration of the 104C driver.



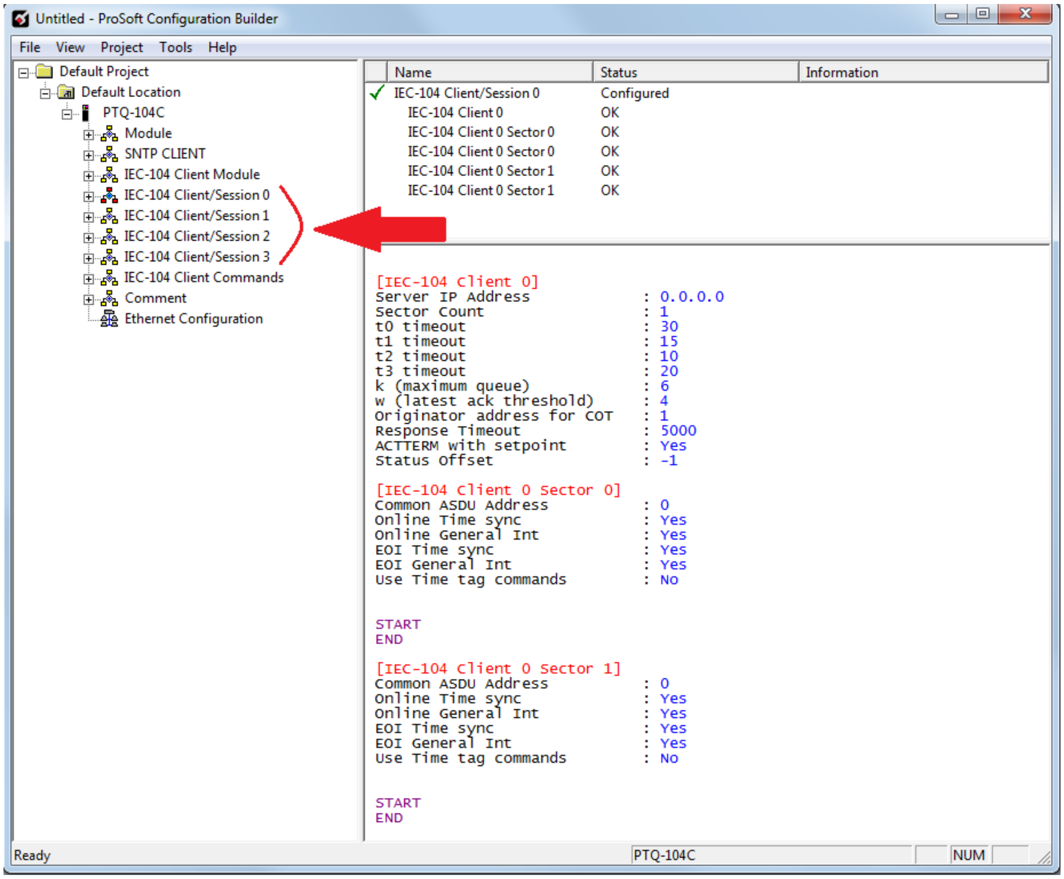
	Name	Status	Information
✓	IEC-104 Client Module	All Tags Good	

```
[IEC-104 Client Module]
Time DB Offset                  : 2000
Driver Status Offset            : 2010
```

Item	Range	Description
Time DB Offset	-1 to 3994	-1: Disable 0-3994: Database location of IEC time NOTE: Requires 6 registers
Driver Status Offset:	-1 to 3970	-1: Disable 0-3970: Database location of general client driver status data NOTE: Requires 30 registers

5.6 [IEC-104 Client/Session x]

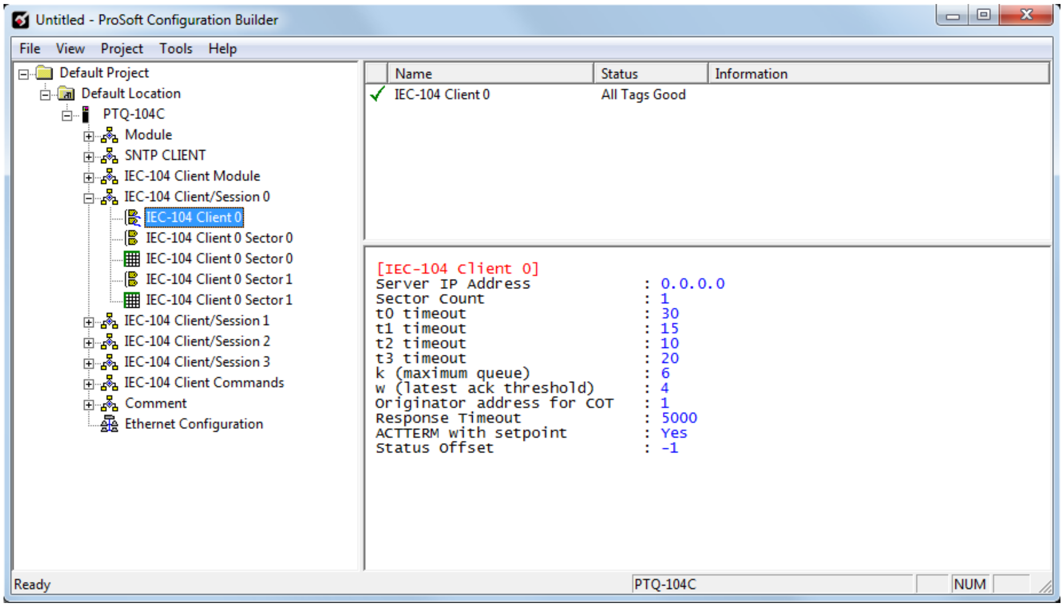
The [IEC-104 Client/Session x] section of PCB includes the client configuration for up to four IEC-104C Clients on the module.



	Name	Status	Information
✓	IEC-104 Client/Session 0	Configured	
	IEC-104 Client 0	OK	
	IEC-104 Client 0 Sector 0	OK	
	IEC-104 Client 0 Sector 0	OK	
	IEC-104 Client 0 Sector 1	OK	
	IEC-104 Client 0 Sector 1	OK	

```
[IEC-104 Client 0]
Server IP Address              : 0.0.0.0
Sector Count                   : 1
t0 timeout                     : 30
t1 timeout                     : 15
t2 timeout                     : 10
t3 timeout                     : 20
k (maximum queue)              : 6
w (latest ack threshold)       : 4
Originator address for COT     : 1
Response Timeout               : 5000
ACTTERM with setpoint          : Yes
Status Offset                  : -1

[IEC-104 Client 0 Sector 0]
Common ASDU Address            : 0
Online Time sync               : Yes
Online General Int             : Yes
EOI Time sync                  : Yes
EOI General Int                : Yes
Use Time tag commands          : No


START
END

[IEC-104 Client 0 Sector 1]
Common ASDU Address            : 0
Online Time sync               : Yes
Online General Int             : Yes
EOI Time sync                  : Yes
EOI General Int                : Yes
Use Time tag commands          : No


START
END
```

5.6.1 [IEC-104 Client/Session x > IEC-104 Client x] section



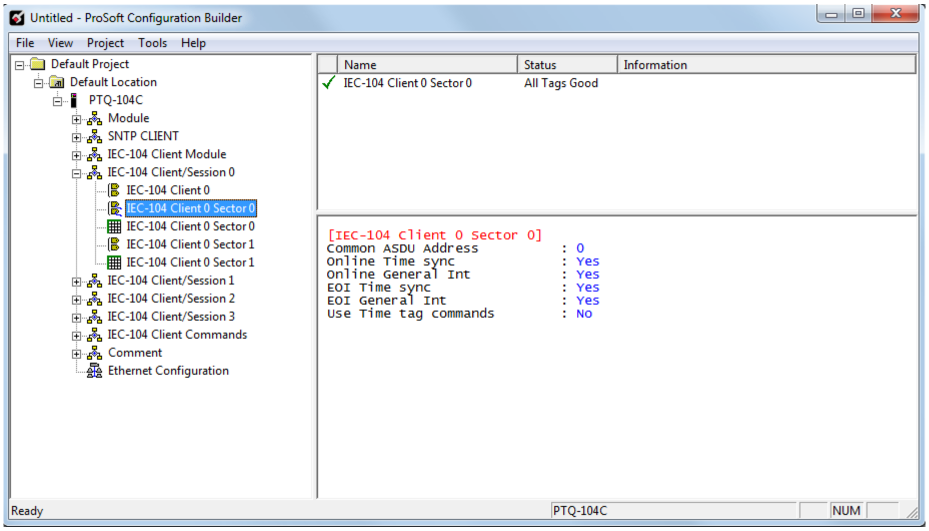
Name	Status	Information
✓ IEC-104 Client 0	All Tags Good	

```
[IEC-104 Client 0]
Server IP Address            : 0.0.0.0
Sector Count                 : 1
t0 timeout                   : 30
t1 timeout                   : 15
t2 timeout                   : 10
t3 timeout                   : 20
k (maximum queue)            : 6
w (latest ack threshold)     : 4
Originator address for COT   : 1
Response Timeout             : 5000
ACTTERM with setpoint        : Yes
Status Offset                : -1
```

Item	Range	Description
Server IP Address	IP dotted notation	IP address for the server to connect this client to with 0.0.0.0 used to disable the client.
Sector Count	1 or 2	This parameter sets the number of sectors contained in this controlled device. The range of values is from 1 to 2. A sector section is required for each sector in a session to define its database and settings.
T0 timeout	1 to 30	Number of seconds to wait for connection to be established with server in seconds (default=30)
T1 timeout	1 to 255	Number of seconds until a timeout occurs to send or test APDU's (Application Protocol Data Units) (default = 15)
T2 timeout	1 to 255	Number of seconds until timeout for non-data acknowledgement (t2 < t1) (default=10)
T3 timeout	1 to 300	Timeout for test frame on idle state in seconds (default=20)
k (maximum queue)	1 to 12	Maximum number of messages to hold in queue
w (latest ack threshold)	1 to 8	Threshold value when to send ack. This parameter should be less than or equal to two-thirds of k.
Originator address for COT	0 to 255	Originator address in the COT (Cause of Transmission) to use in all messages from the client
Response Timeout	500 to 2^32-1	This parameter sets the maximum number of milliseconds to wait for a confirmation from the controlled station to a request from this module to application level messages.

ACTTERM with setpoint	Y or N	This parameter specifies what the last message will be in the response to a setpoint command. If the parameter is set to Y, an ACTTERM will be the last response. If set to N, ACTCON will be the last response.
Status Offset	-1 to 3956	-1: Disable 0-3956: Database location of client status data NOTE: requires 44 registers and should be placed in a location of the module memory that is being used for read data to transfer this information back to the processor.

5.6.2 [IEC-104 Client/Session x > IEC-104 Client x Sector 0] section



Name	Status	Information
✓ IEC-104 Client 0 Sector 0	All Tags Good	

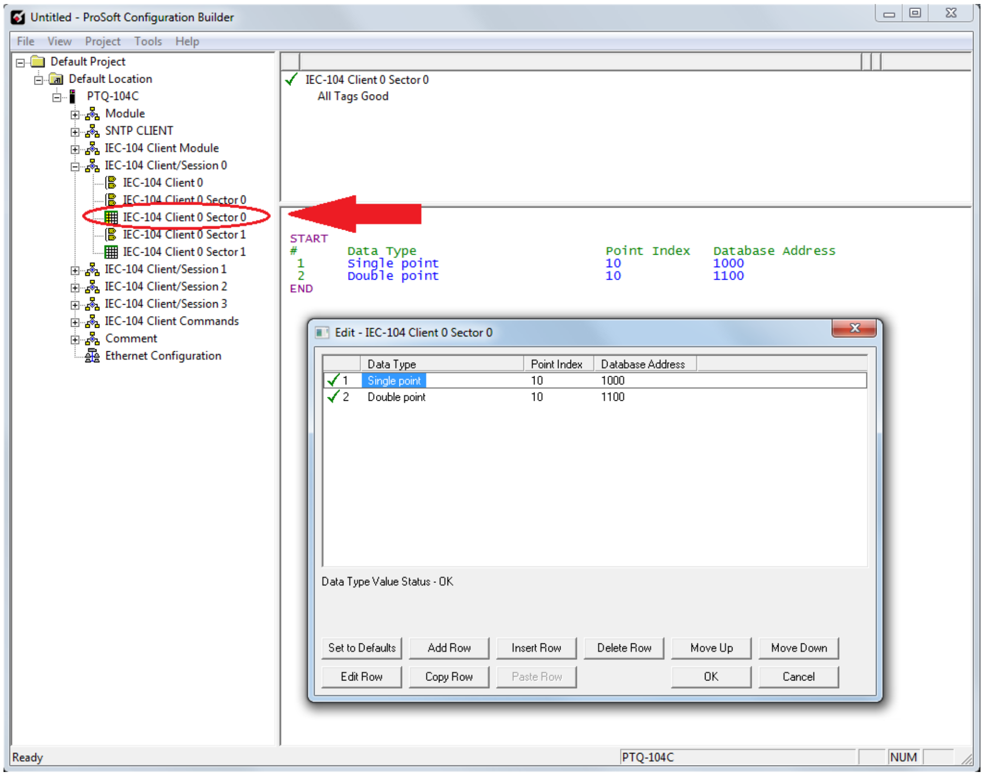
```
[IEC-104 Client 0 Sector 0]
Common ASDU Address          : 0
Online Time sync             : Yes
Online General Int           : Yes
EOI Time sync                : Yes
EOI General Int              : Yes
Use Time tag commands        : No
```

Item	Range	Description
Common ASDU Address	0 to 65535	This parameter sets the common ASDU address to associate with this sector of the specified session.
Online Time Sync	Y or N	This parameter specifies if the sector in the controlled device will be sent a time synchronization command when the unit is first recognized as being online.
Online General Int	Y or N	This parameter specifies if the sector in the controlled device will be sent a general interrogation command when the unit is first recognized as being online.
EOI Time Sync	Y or N	This parameter specifies if the sector in the controlled device will be sent a time synchronization command after this module receives an EOI message from the controlled unit.

EOI General Int	Y or N	This parameter specifies if the sector in the controlled device will be sent a general interrogation command after this module receives an EOI message from the controlled unit.
Use Time Tag Commands	Y or N	This parameter specifies if a time tag field is to be included with commands. This is as specified in the IEC-870-5-104 specification and should only be utilized if the controlled device supports these new data types. If the parameter is set to Y, a time tag will be added to all commands. If the parameter is set to N, the normal IEC-870-5-101 data type messages will be utilized

5.6.3 [IEC-104 Client/Session x > IEC-104 Client x Sector 0] section

This section defines the Data Type distribution of the module database for Client x Sector 0.



✓ IEC-104 Client 0 Sector 0
All Tags Good

```
START
#       Data Type                                   Point Index   Database Address
 1      Single point                                10            1000
 2      Double point                                10            1100
END
```

	Data Type	Point Index	Database Address
✓ 1	Single point	10	1000
✓ 2	Double point	10	1100

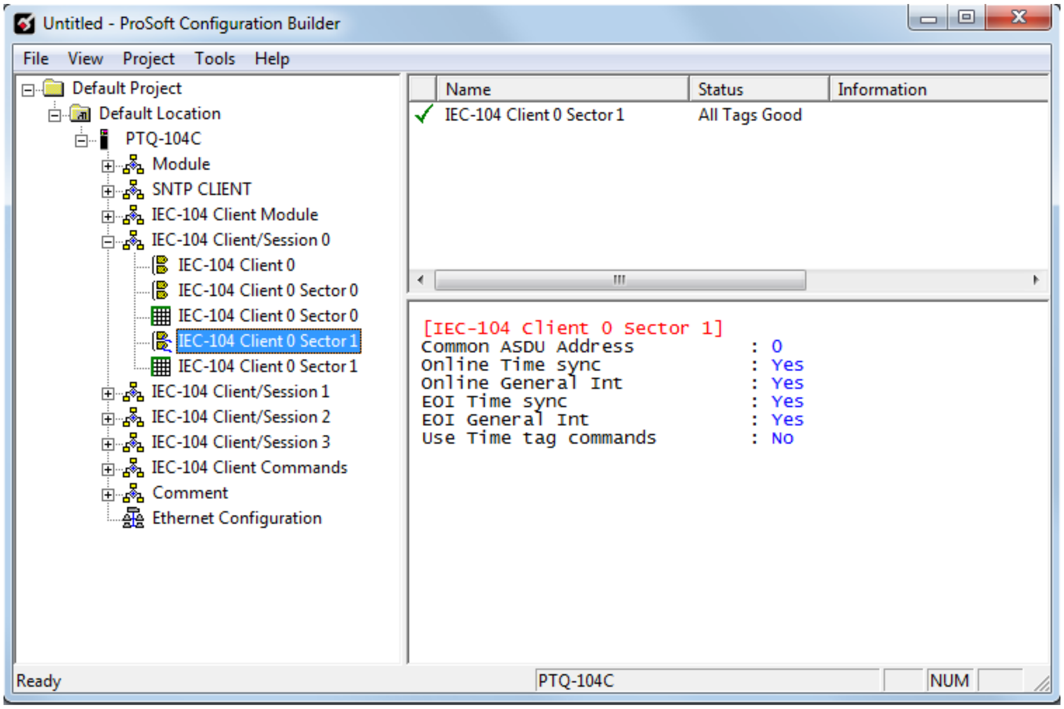
Data Type Value Status - OK

Item	Range	Description
Data Type		ASDU type code in message
Point Index	0 to 65535	Value for the Information Object Address (IOA)
Database Address	0 to 64000	Database address in the module associated to this point according to ASDU type: Single-point: bit-addressing Double-point: bit-addressing Step-point: byte-addressing

		Bitstring: double-word-addressing Setpoint normalized: word-addressing Setpoint scaled: word-addressing Setpoint floating: double-word-addressing Parameter normalized: word-addressing Parameter scaled: word-addressing Parameter float: double-word-address Integrated BCD: 3-word-addressing

5.6.4 [IEC-104 Client/Session x > IEC-104 Client x Sector 1] section

This section defines Sector 1 of Session x.



Name	Status	Information
✓ IEC-104 Client 0 Sector 1	All Tags Good	

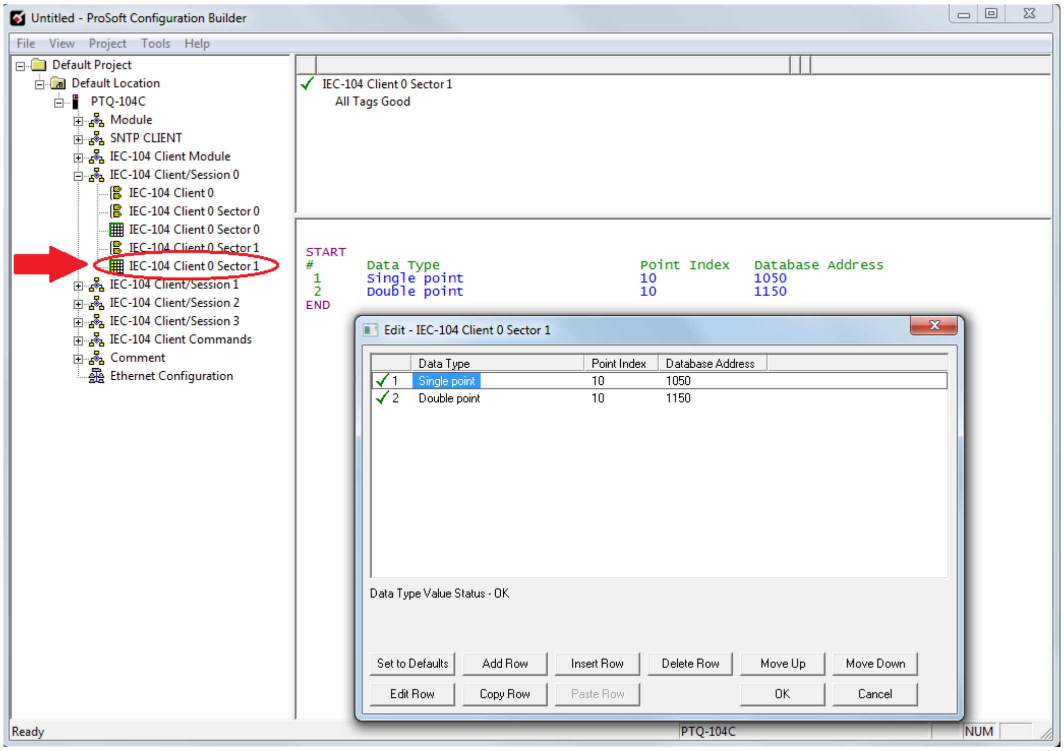
```
[IEC-104 Client 0 Sector 1]
Common ASDU Address             : 0
Online Time sync                : Yes
Online General Int              : Yes
EOI Time sync                   : Yes
EOI General Int                 : Yes
Use Time tag commands           : No
```

Item	Range	Description
Common ASDU Address	0 to 65535	This parameter sets the common ASDU address to associate with this sector of the specified session.
Online Time Sync	Y or N	This parameter specifies if the sector in the controlled device will be sent a time synchronization command when the unit is first recognized as being online.
Online General Int	Y or N	This parameter specifies if the sector in the controlled device will be sent a general interrogation command when the unit is first recognized as being online.

Item	Range	Description
EOI Time Sync	Y or N	This parameter specifies if the sector in the controlled device will be sent a time synchronization command after this module receives an EOI message from the controlled unit.
EOI General Int	Y or N	This parameter specifies if the sector in the controlled device will be sent a general interrogation command after this module receives an EOI message from the controlled unit.
Use Time Tag Commands	Y or N	This parameter specifies if a time tag field is to be included with commands. This is as specified in the IEC-870-5-104 specification and should only be utilized if the controlled device supports these new data types. If the parameter is set to Y, a time tag will be added to all commands. If the parameter is set to N, the normal IEC-870-5-101 data type messages will be utilized

5.6.5 [IEC-104 Client/Session x > IEC-104 Client x Sector 1] section

This section defines the Data Type distribution of the module database for Client x Sector 1.



✓ IEC-104 Client 0 Sector 1
All Tags Good

```
START
#        Data Type                         Point Index    Database Address
 1       Single point                      10             1050
 2       Double point                      10             1150
END
```

Edit - IEC-104 Client 0 Sector 1

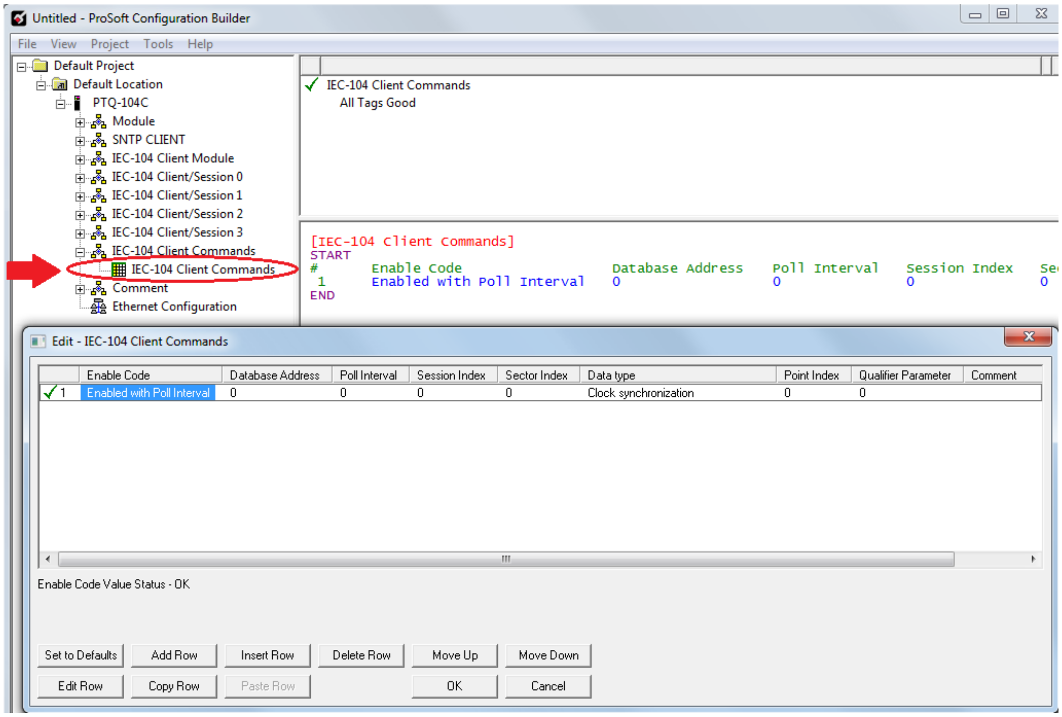
	Data Type	Point Index	Database Address
✓ 1	Single point	10	1050
✓ 2	Double point	10	1150

Data Type Value Status - OK

Item	Range	Description
Data Type		ASDU type code in message
Point Index	0 to 65535	Value for the Information Object Address (IOA)
Database Address	0 to 64000	Database address in the module associated to this point according to ASDU type: Single-point: bit-addressing Double-point: bit-addressing Step-point: byte-addressing Bitstring: double-word-addressing Setpoint normalized: word-addressing Setpoint scaled: word-addressing Setpoint floating: double-word-addressing Parameter normalized: word-addressing Parameter scaled: word-addressing Parameter float: double-word-address Integrated BCD: 3-word-addressing

5.7 [IEC-104 Client Commands] section

This section defines the 104C Client commands to send to the remote servers.



IEC-104 Client Commands
All Tags Good

```
[IEC-104 Client Commands]
START
#       Enable Code                 Database Address    Poll Interval    Session Index    Se
 1      Enabled with Poll Interval  0                   0                0                0
END
```

Edit - IEC-104 Client Commands

	Enable Code	Database Address	Poll Interval	Session Index	Sector Index	Data type	Point Index	Qualifier Parameter	Comment
✓ 1	Enabled with Poll Interval	0	0	0	0	Clock synchronization	0	0	

Enable Code Value Status - OK

Item	Range	Description

Enable Code	Enable/Disable	Disable = Disables command
		Enable = Enables command with Poll Interval in seconds
		Conditional = Executes command when point value in database changes

Database Address		This field specifies the location in the module's internal database to associate with the command. The Data Type (page 46) used in the command determines addressing of the index as follows:
	Data Type	**Description**
	45	Bit address, Single point command
	46	Bit address, Double point command
	47	Byte address, Regulating step point command
	48	Word address, Setpoint normalized point command
	49	Word address, Setpoint scaled point command
	50	Double-Word address, Setpoint short float point command
	51	Double-Word address, Bitstring (32-bits) point command
	100	*Word address, Group interrogation command
	101	*Word address, Counter interrogation command
	102	*Word address, Read command
	103	*Word address, Clock synchronization command
	105	*Word address, Reset process command
	107	*Word address, Test command (IEC-870-5-104 type)
	110	Word address, Normalized measured value
	111	Word address, Scaled measured value
	112	Float (double-word), Short float value
	113	*Word address, Activation command
		*Word address = Value only used to signal when to send event (Enable Code = 2)
Poll Interval	1 to 65535	This parameter is used if the Enable Code field is set to a value of 1. Delay (in seconds) to execute this command since its last execution. To be used if this command is to be executed every X seconds.
Session Index	0 to 3	This parameter is utilized to associate the command with one of the sessions defined for the module.
Sector Index	0 to 1	This parameter is used to associate the command with the proper sector of the selected session.

Item	Range	Description
Data Type	ASDU Type	This parameter is used to set the ASDU data type to be used with the message. The codes specified are those defined for the IEC-870-5-101 protocol. The following is a listing of command control data types supported in this module: Type: 45, 46, 47, 48, 49, 50, 51, 100, 101, 102, 103, 105, 107, 110, 111, 112, 113. (Full descriptions on page 46)
Point Index	1 to 65535	This parameter is the value for the Information Object Address (IOA) in the target device
Qualifier	1 to 1023	This parameter is dependent on the Data Type selected. See below for details.

Qualifier Parameter configuration

The Qualifier Parameter is based on a combination of parameters each with their own value, based on the Type used in the command. The corresponding parameters for each Data Type have a numerical code assigned to them. The Qualifier Parameter value is the sum of these numerical codes.

Data Type	Description	Qualifier Parameter
45	Bit address, Single point command	Single Point Value: 0=Off 1=On
46	Bit address, Double point command	Double Point Value: 0=Not permitted 1=Off 2=On 3=Not Permitted
47	Byte address, Regulating step point command	Regulating Point Value: 0=Not permitted 1=Next step lower if database point is set to -1 2=Next step high if database point set to +1 3=Not Permitted

The following parameters apply to each Data Type (45, 46, 47)

Qualifier Code	0=No additional definition (slave dependent) 4=Short pulse duration 8=Long pulse duration 12=Persistent output
Select/Execute	0=Direct execution without select 128=Select executed followed by execute 256=Deselect command

Use Override Flag	0=Use value in database for value 512=Use override value for state

Example - Enabling a Data Type 45 Single Point Value with:
Single Point value: 1 (On)
Qualifier Code: 4 (Short pulse duration)
Select/Execute: 0 (Direct execution without Select)
Use Override Flag: 512 (Use override value for state)

1 + 4 + 0 + 512 = 517

A value of '517' would need to be entered in the Qualifier Parameter for this command.

Data Type	Description
48	Word address, Setpoint normalized point command
49	Word address, Setpoint scaled point command
50	Double-Word address, Setpoint short float point command

The following parameter applies to each Data Type (48, 49, 50)

Qualifier Code	0=Direct execution without select 1=Select executed followed by execute 2=Deselect command

Example – Each of the Setpoint Command Data Types (48, 49, 50) can be executed with a value of 0, 1, or 2.

Data Type	Description	Qualifier Parameter
51	Double-Word address, Bitstring (32-bits) point command	No Qualifier used

Data Type	Description	Qualifier Parameter
100	*Word address, Group interrogation command	0=Not used 1 to 19 = Reserved by standard 20=Station interrogation (global) 21=Interrogation group 1 22=Interrogation group 2 23=Interrogation group 3 24=Interrogation group 4 25=Interrogation group 5 26=Interrogation group 6 27=Interrogation group 7 28=Interrogation group 8 29=Interrogation group 9 30=Interrogation group 10 31=Interrogation group 11 32=Interrogation group 12 33=Interrogation group 13 34=Interrogation group 14 35=Interrogation group 15 36=Interrogation group 16 37 to 63 = Reserved by standard 64 to 255 = Reserved for special use (private range)
101	*Word address, Counter interrogation command	**Counter Interrogation Group:** 0=No counter requested 1=Request counter group 1 2=Request counter group 2 3=Request counter group 3 4=Request counter group 4 5=Request general counter group 6 to 31 = Reserved by standard 32 to 63 = Reserved for special use (private range) **Freeze/Reset Qualifier:** 0=No freeze or reset 64=Counter freeze without reset 128=Counter freeze with reset 192=No freeze with counter reset
102	*Word address, Read command	No Qualifier used
103	*Word address, Clock synchronization command	0=Clock synchronization
105	*Word address, Reset process command	0=Not used 1=General reset of process 2=Reset pending information with time tag of the event buffer 3 to 127 = Reserved by standard 128 to 255 = Reserved for special use (private range)

Data Type	Description	Qualifier Parameter
110	Word address, Normalized measured value	The following applies to Type 110, 111, 112: **Type of parameter:** 0=Not used 1=Threshold value 2=Smoothing factor (filter time constant) 3=Low limit for transmission of measured values 4=High limit for transmission of measured values 5 to 31 = Reserved by standard 32 to 63 = Reserved for special use (private range) **Local parameter change:** 0=No change 64=Change **Parameter in operation:** 0=Operation 128=Not in operation
111	Word address, Scaled measured value	
112	Float (double-word), Short float value	
113	*Word address, Activation command	**Parameter Qualifier:** 0=Not used 1=Act/Deact of previously loaded parameters (point index = 0) 2=Act/Deact of the parameter of the point index specified 3=Act/Deact of persistent cyclic or periodic transmission of the addressed object 4 to 127 = Reserved by standard 128 to 255 = Reserved for special use (private range) **Activation Qualifier:** 0=Deactivate 256=Activate

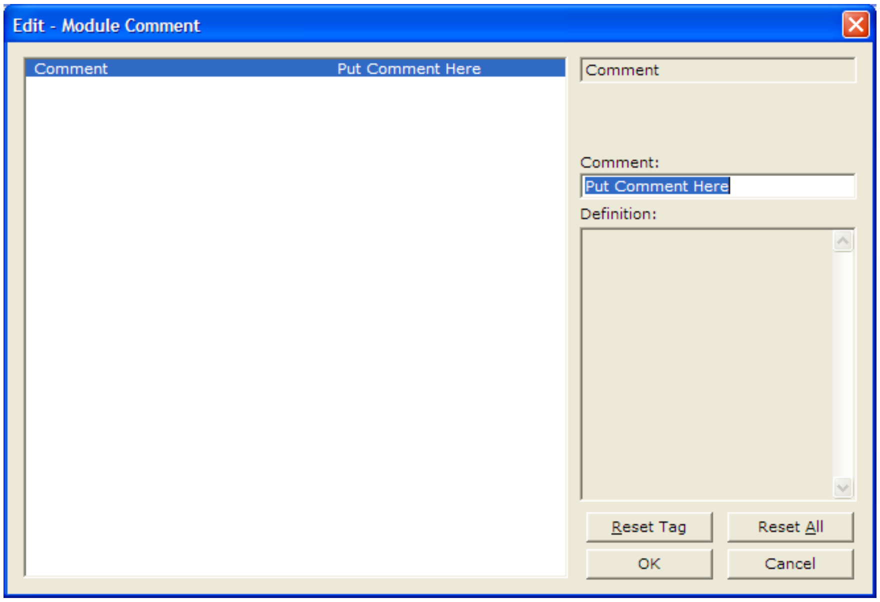
5.8 [Ethernet Configuration] section

See page 19 for more information.

5.9 To Create Optional Comment Entries

1 Click the plus sign to the left of the ⊞ Comment icon to expand the module Comments.

2 Double-click the Module Comment icon. The *Edit - Module Comment* dialog appears.



Comment	Put Comment Here

Comment

Comment: Put Comment Here

Definition:

3 Enter the comment and click **OK** to save the changes.

5.10 To print a configuration file

1 Select the module 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 On the *View Configuration* window, open the **FILE** menu, and choose **PRINT.** This action opens the *Print* dialog box.

4 On the *Print* dialog box, choose the printer to use from the dropdown list, select printing options, and then click **OK.**

6 Downloading the PCB File to the Module

In This Chapter

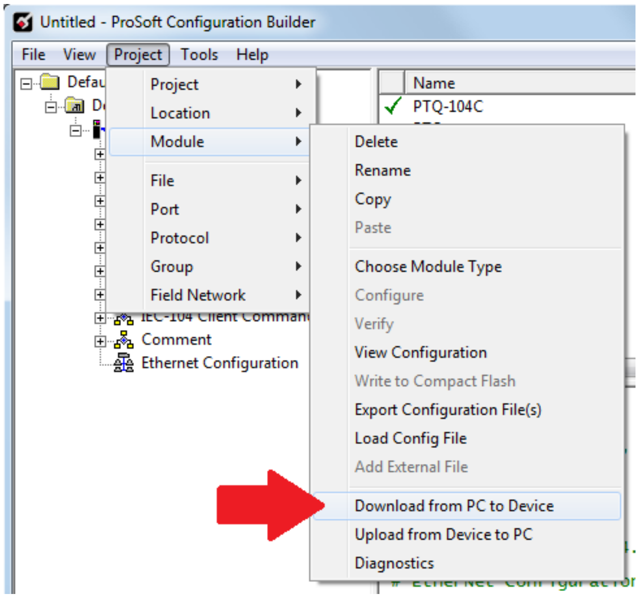
For the module to use the configured settings, download the updated configuration file from the PC to the module. This can be done over Ethernet or Serial connection.

Note: The processor (Quantum) must be in "Stop" mode before the file is downloaded to the module. Use the processor's configuration tool or the softkeys on the processor to stop the processor.

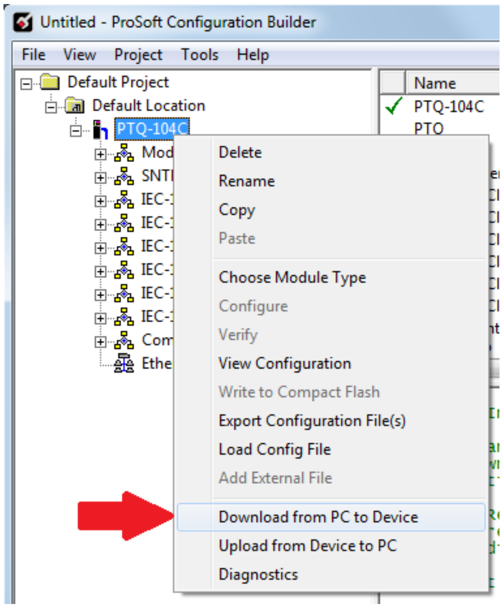
6.1 Downloading the Project via Serial Connection

1 In the tree view in ProSoft Configuration Builder, click once to select the PTQ-104C module.

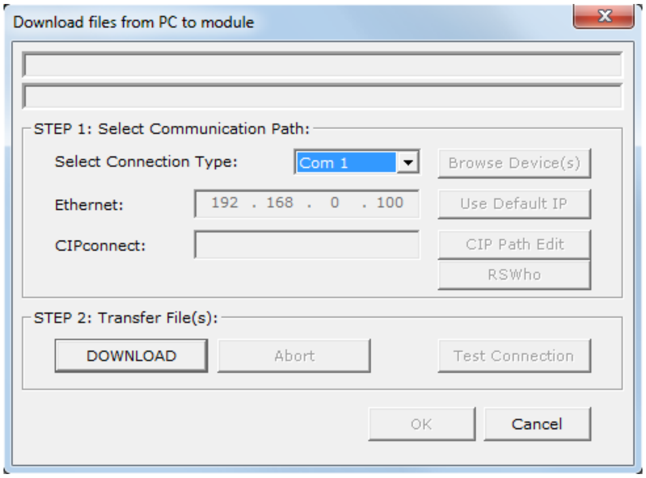
2 Open the **PROJECT** menu from the top of the PCB window, and mouse-over **MODULE.** Select **DOWNLOAD FROM PC TO DEVICE**.



Or, right-click the PTQ-104C icon in PCB and select **DOWNLOAD FROM PC TO DEVICE.**



3 The program will scan the PC for a valid com port (this may take a few seconds). When PCB has found a valid com port, the **DOWNLOAD** dialog box will open.



Download files from PC to module

STEP 1: Select Communication Path:

Select Connection Type: Com 1

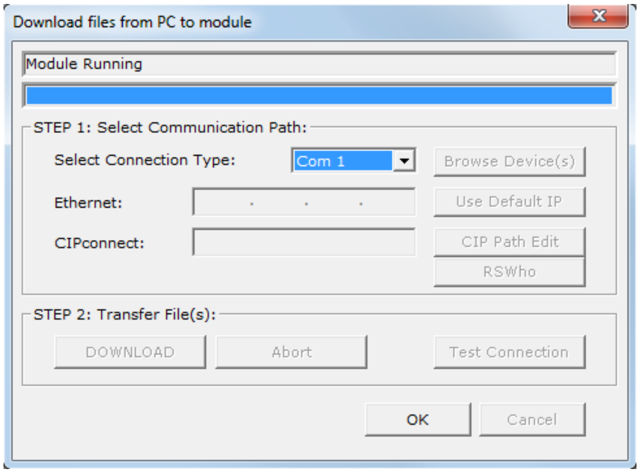
Ethernet: 192 . 168 . 0 . 100

CIPconnect:

STEP 2: Transfer File(s):

4 Choose the com port to use from the dropdown list, and then click the **DOWNLOAD** button.

The module 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: Com 1

Ethernet: . . .

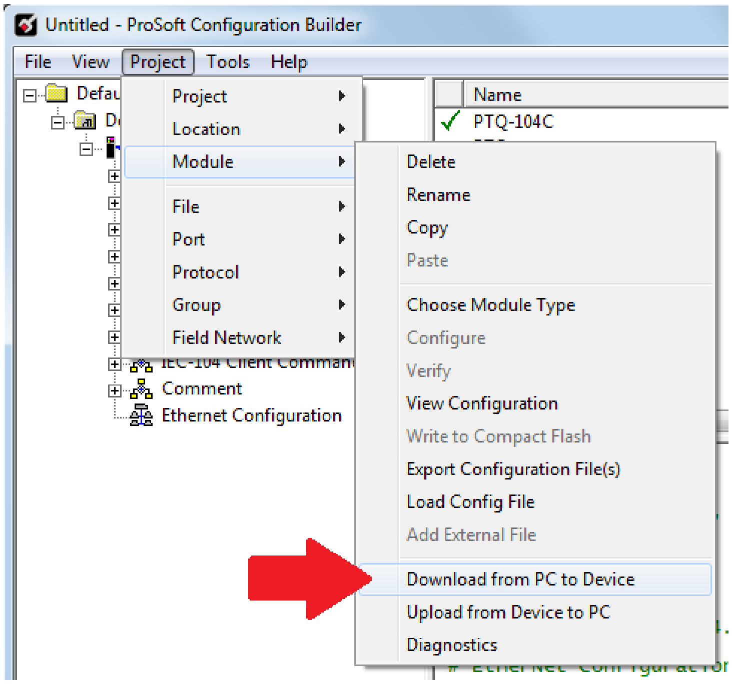
CIPconnect:

STEP 2: Transfer File(s):

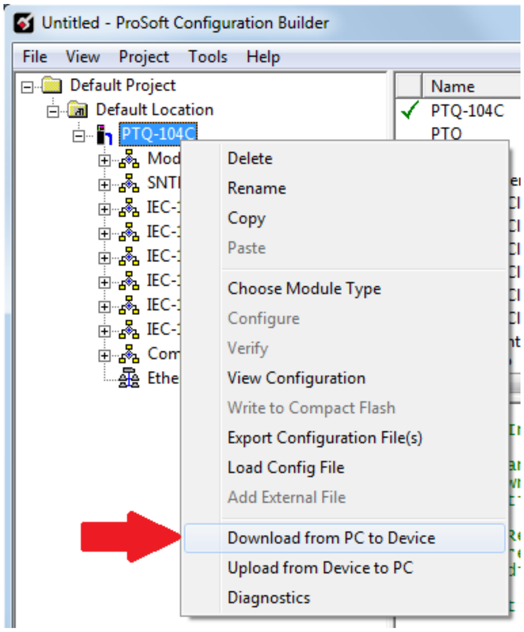
6.2 Downloading the Project via Ethernet Connection

This procedure cannot be done unless the IP address has been initially configured and downloaded serially (page 19) to the module. In the tree view in ProSoft Configuration Builder, left-click once to select the PTQ-104C module.

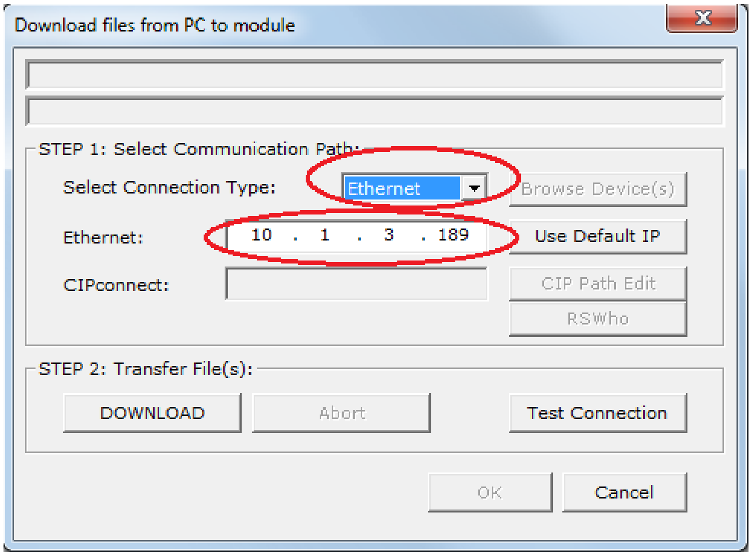
1 Open the **PROJECT** menu from the top of the PCB window, and mouse-over **MODULE.** Select **DOWNLOAD FROM PC TO DEVICE**.



Or right-click the PTQ-104C icon in PCB and select **DOWNLOAD FROM PC TO DEVICE.**



2 The **DOWNLOAD** dialog box will open. Select *Ethernet* as the Connection Type and enter the IP Address that was assigned in the [Ethernet Configuration] section of PCB.



Download files from PC to module

STEP 1: Select Communication Path:

Select Connection Type: Ethernet

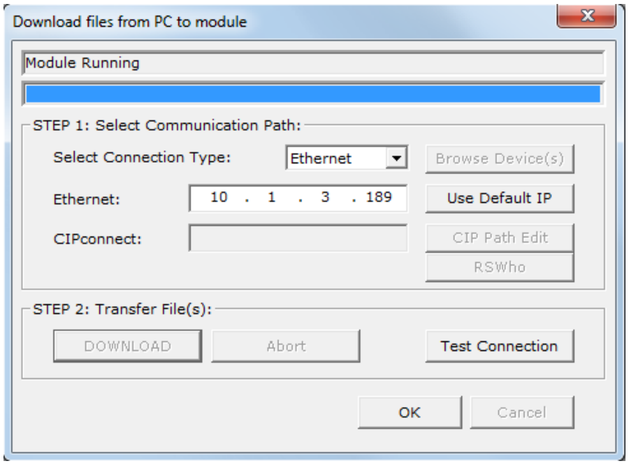
Ethernet: 10 . 1 . 3 . 189

CIPconnect:

STEP 2: Transfer File(s):

3 Click the **DOWNLOAD** button.

The module 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: 10 . 1 . 3 . 189

CIPconnect:

STEP 2: Transfer File(s):

7 Hot Standby Support

In This Chapter

7.1 Hot Standby Overview

This section describes the PTQ-104C IEC 870-5-104 Client module specifications and startup support for Modicon Quantum Hot Standby system.

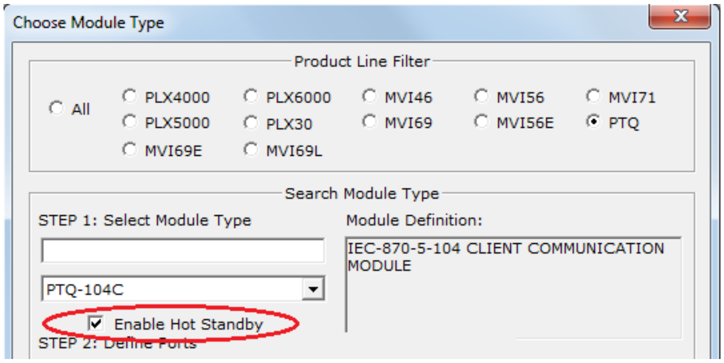
A Modicon Quantum Hot Standby with Unity system can be used when downtime cannot be tolerated. Hot standby systems deliver high availability through redundancy. A hot standby system consists of two identical configurations.

- Modicon Quantum 140 CPU 671 60
- Modicon Quantum Power Supply Module
- Modicon Quantum RIO Head
- ProSoft Technology PTQ-104C module hardware version 1.05 or higher
- Modicon Optional Modules (NOE, NOM)

One of the 140 CPU 67160s acts as the Primary controller and the other acts as the Standby controller. The Primary controller runs the application program and operates the remote I/O.

Note: The Modicon Quantum RIO Head is required even if the Remote I/O will not be used.

This feature is enabled by clicking the *Enable Hot Standby* button in the *Choose Module Type* window.



● PTQ

PTQ-104C

☑ Enable Hot Standby

IEC-870-5-104 CLIENT COMMUNICATION MODULE

7.1.1 Identical Configurations

Two backplanes are configured with identical hardware and software. One of the programmable logic controllers (PLCs) functions as the Primary controller and the other as a Standby controller, and either controller can be put in the Primary state, but the other must be in the Standby state or offline.

7.1.2 Primary and Standby Controllers

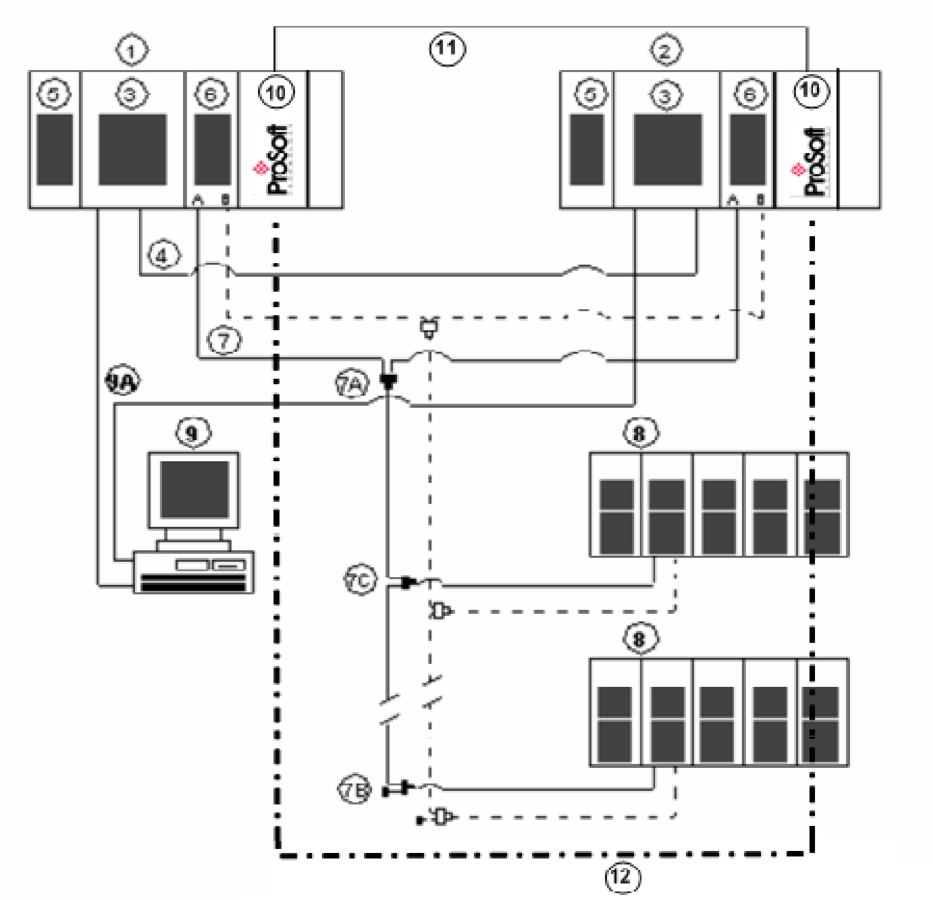
The Primary controller executes the application program, controls the remote I/O, and updates the Standby controller after every scan (program cycle). If the Primary controller fails, the Standby controller takes control within one scan. To determine if the Primary controller failed note controller’s status displayed in the HE CPU LCD screen and the RIO Head’s status displayed by the RIO Head’s LEDs.

The Standby controller does not execute the full application program but only the first section, and the Standby controller does not control the remote I/O but checks out the availability of the Modicon Quantum Hot Standby with Unity equipment.

Note: For additional information on Quantum Hot Standby support, refer to the *Unity Pro Hot Standby User Guide*.

7.1.3 System Components

The following illustration shows the components required for a Modicon Quantum Hot Standby with Unity system.



1 Primary PLC
2 Standby PLC
3 Modicon Quantum Hot Standby with Unity controller with integrated coprocessor
4 Fiber Optic Cable to connect to both controllers
5 Modicon Quantum power supply module: Install power supply in first slot for better rack layout.
6 Modicon Quantum RIO head
7 Coaxial cable with splitters (7A) (MA-0186-100), trunk terminators (7B) (52-0422-000), and tap (7C) (MA-0185-100) for connecting the RIO heads (6) with the RIO drops (8). The dashed connections represent a redundant connection in the RIO network, which is not required for the Modicon Quantum Hot Standby with Unity system.
8 Modicon Quantum RIO drop

9 Unity Pro computer connected to both controllers via Modbus or Modbus Plus (9A)
10 PTQ-104C HSBY modules
11 PTQ-104C Ethernet redundancy communication cable
12 IEC 60870-5-104 network.

Note: The 140 CRP 932 00 RIO Head unit is required for Hot Standby System to work.

7.1.4 Modicon Quantum Hot Standby with Unity and IEC Logic

Overview

A Modicon Quantum Hot Standby with Unity system requires two backplanes configured with identical hardware, software, and firmware. One of the controllers (PLC) functions as the Primary controller and the other as a Standby controller.

- The Primary updates the Standby after every scan.
- The Primary and Standby communicate constantly monitoring the health of the system.
- If the Primary fails, the Standby takes control within one scan.

7.1.5 Understanding System Scan Time in Modicon Quantum Hot Standby with Unity Systems

Effect on System Scan Time

The scan time of any Modicon Quantum Hot Standby with Unity system depends on the amount of data transferred. Because data must be transferred from Primary to Standby, any Modicon Quantum Hot Standby with Unity system always has a higher scan time than a comparable stand-alone system.

Performance Considerations

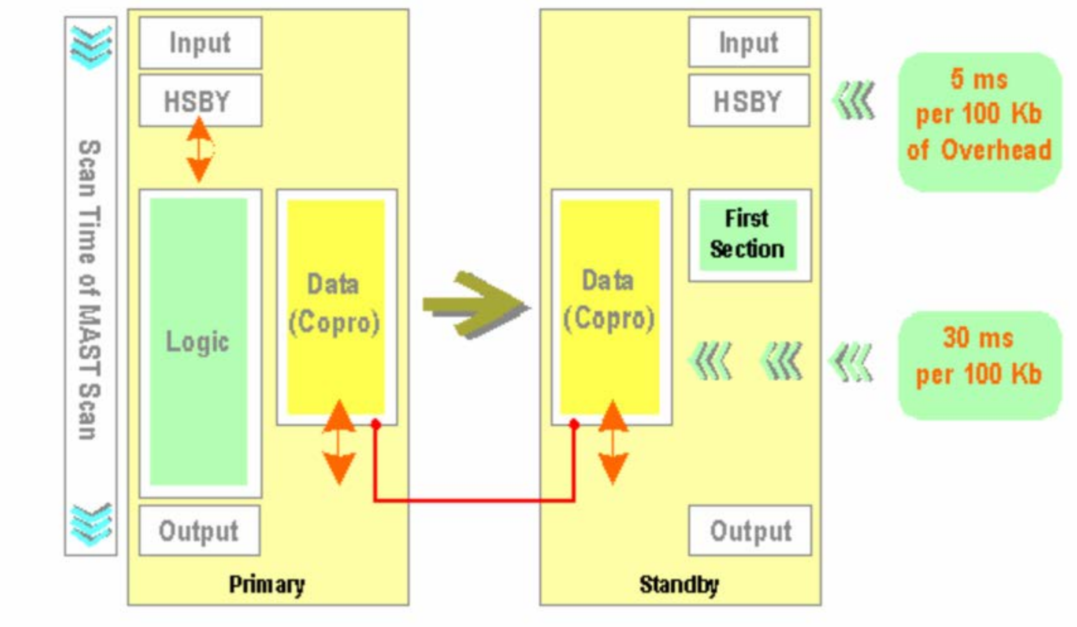
A Modicon Quantum Hot Standby with Unity system increases the length of a MAST scan, creating system overhead.

Note: System overhead is the time required to copy the application data to the communication link layer.

The network scan (communication between Primary and Standby "copros")

1 Exchanges data between both controllers
2 Runs in parallel with the application program.

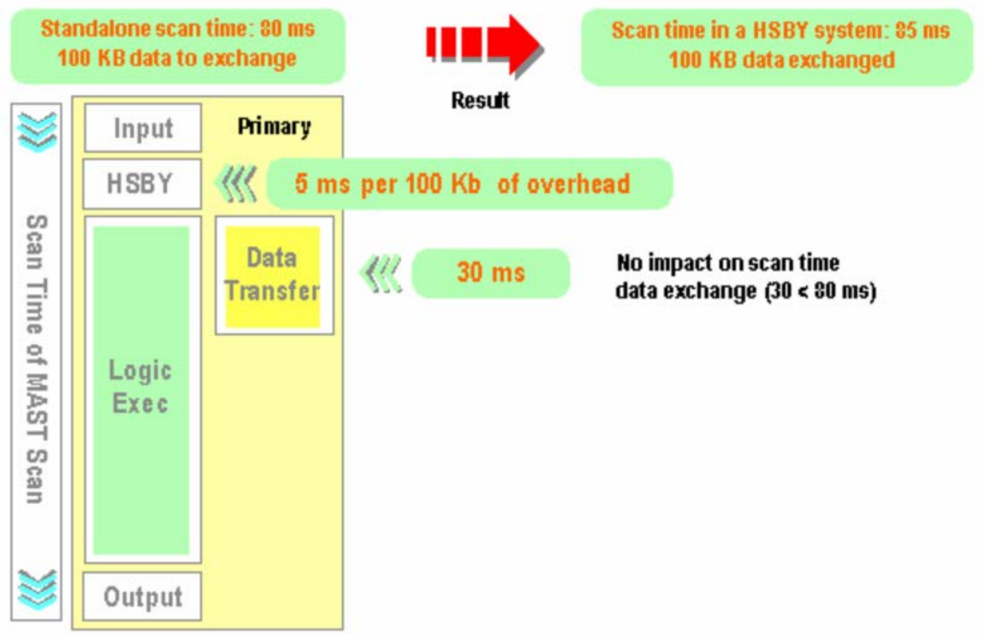
A Hot Standby system



Most of the time, the MAST scan hides the network scan. However, when some application programs are processed, additional system overhead may occur.

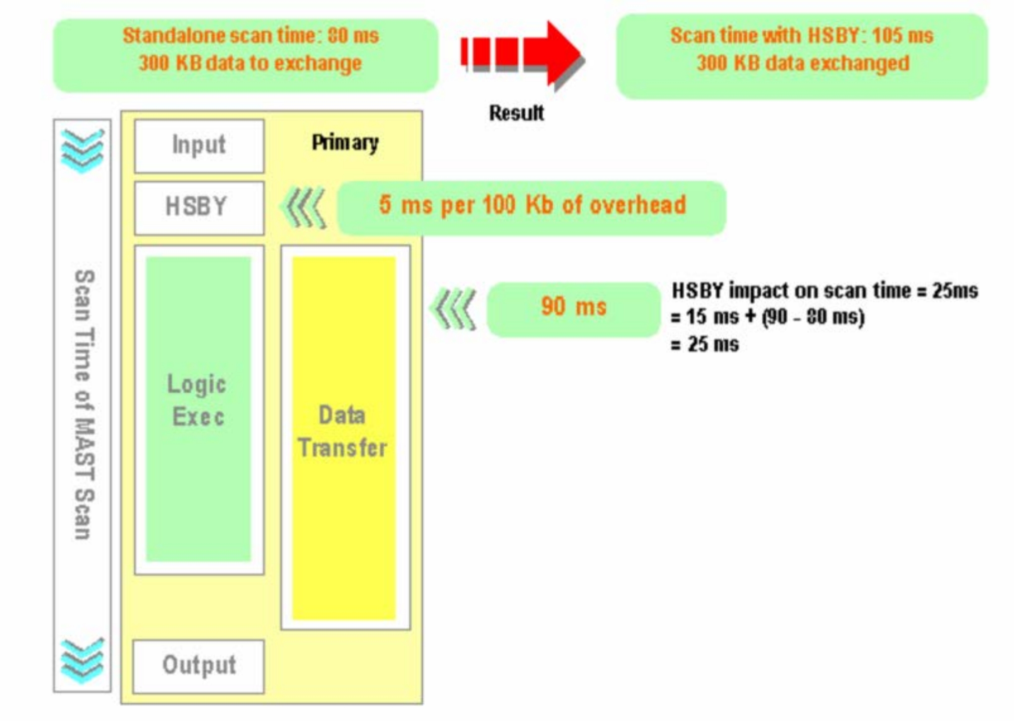
Example #1

- Stand-alone application scan time: 80 ms
- Data (state RAM + unallocated variables): 100 Kb



Example #2

- Stand-alone application scan time: 80 ms
- Data (state RAM + unlocated variables): 300 Kb



Note: In addition to the above times for system overhead, the PTQ-104C module may acquire from 100 ms to 300 ms of switch-over time. All configured data is to be updated as fast as the combined asynchronous events occur based on the processor scan time, backplane transfer time, PTQ data handling time and IEC 60870-5-104 bus cycle time. The IEC 60870-5-104 bus cycle time is based on slave reaction time, sync time, baud rate and other bus delay times for a given number of servers on the network.

7.2 Setting Up the Modicon Quantum Hot Standby with Unity System

7.2.1 Overview

Schneider Electric is a leader in offering fault-tolerant, redundant systems, and Hot Standby. Setting up a Modicon Quantum Hot Standby with Unity system involves a number of processes.

7.2.2 Mapping the Backplane Extensions

A Modicon Quantum Hot Standby with Unity requires two backplanes with at least four slots. The two backplanes must be mapped in an identical manner:

- same Modicon Quantum Hot Standby with Unity HE CPU with integrated coprocessor (Copro)
- same firmware
  - same revision level
  - same Modicon Quantum power supply module
  - same Modicon Quantum RIO Head

If other modules are used, for example local I/Os, NOMs, NOEs, these modules must be identical.

For additional information on Modicon Quantum Hot Standby Startup support, refer to the Unity Pro User Guide.

7.2.3 PTQ-104C Hot Standby Considerations

Limitations

The solution allows for up to six PTQ modules per rack (for both Primary and Standby).

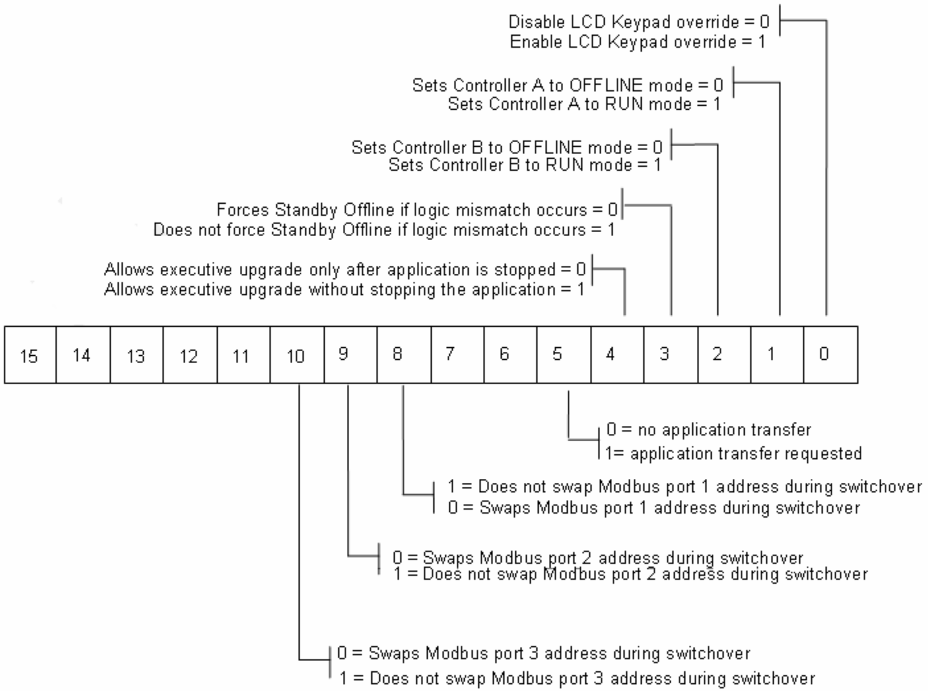
The PTQ-104C module cannot be installed in a RIO drop.

HSBY Operating Modes

Generally, the user will have full control over the switchover via the command register through application program control. This is accomplished by the user application through the SW60 command register.

PLC HSBY Command Register

The following illustration identifies the operating options provided by the Command Register (%SW60). The Command Register defines the operation of the Hot Standby application. That means both the Primary and Standby. Therefore the Command Register is transferred to the Standby PLC each scan. As a result, any changes made to the Command Register on the Standby PLC will have no effect since the value transfer from the Primary side will overwrite it.


Disable LCD Keypad override = 0
Enable LCD Keypad override = 1
Sets Controller A to OFFLINE mode = 0
Sets Controller A to RUN mode = 1
Sets Controller B to OFFLINE mode = 0
Sets Controller B to RUN mode = 1
Forces Standby Offline if logic mismatch occurs = 0
Does not force Standby Offline if logic mismatch occurs = 1
Allows executive upgrade only after application is stopped = 0
Allows executive upgrade without stopping the application = 1
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0 = no application transfer
1= application transfer requested
1 = Does not swap Modbus port 1 address during switchover
0 = Swaps Modbus port 1 address during switchover
0 = Swaps Modbus port 2 address during switchover
1 = Does not swap Modbus port 2 address during switchover
0 = Swaps Modbus port 3 address during switchover
1 = Does not swap Modbus port 3 address during switchover

%SW60 Hot Standby Command Register

%SW60.0: This bit, if set to 1, allows the Command Register RUN status of the PLC to be set through the LCD Keypad.

Warning: If the keypad override is enabled while the Hot Standby system is running, the Primary PLC will immediately read bits 14 and 15 to determine its own state and the state of the Standby. If both bits are set to 0, a switchover will occur and the former Primary will go offline. The new Primary will continue to operate.

%SW60.1: Setting this bit = 1 will put PLC A in RUN mode. Setting the bit = 0 will put PLC A in OFFLINE mode. This bit takes effect only if bit 16 is set = 1.

%SW60.2: Setting this bit = 1 will put PLC B in RUN mode. Setting the bit = 0 will put PLC B in OFFLINE mode. This bit takes effect only if bit 16 is set = 1.

%SW60.3: Setting this bit = 0 will force the Standby PLC offline if a logic mismatch is detected. Logic mismatch is defined as either the MID, LID or CID being different on Primary and Standby sides. Setting this bit = 1 will allow the Standby PLC to continue to operate normally even if the MID is different on the Primary and Standby.

%SW60.4: Setting this bit = 1 allows the executive to upgrade on the Standby without having to stop the application. This means the Hot Standby system is allowed to operate with different versions of the OS running on the Primary and Standby. This option is provided to allow upgrades to be done without shutting down the process. Clearly, the Standby PLC must be stopped to do the executive upgrade, but it will be able to operate as a valid Standby when started again.

%SW60.5: Setting this bit = 1 commands the standby station to initiate an application transfer. That function is not required in UNITY V1.

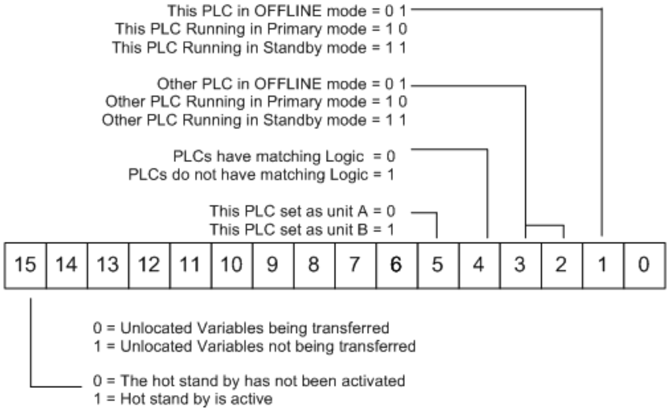
%SW60.8: If this bit is set = 1 the Modbus address on port 1 will be swapped when a switchover occurs. Swapping means to change address by ± 128 to keep the address in the range of 1 to 247. The purpose of this is to allow the P-unit of an HMI to always use the same address to connect to the Primary or Standby/Offline PLC.

PLC HSBY Status Register

The Status Register provides user information relative to the state of the two PLCs in the Hot Standby system. The Status Register is %SW61.

Both the Primary and Standby/Offline PLCs have their own copies of the Status register. The Status register is not transferred from Primary to Standby each scan. Each PLC must maintain its local Status Register based on the regular communication between the two PLCs.

The following illustration identifies the operating options provided by the Command Register.


This PLC in OFFLINE mode = 0 1
This PLC Running in Primary mode = 1 0
This PLC Running in Standby mode = 1 1
Other PLC in OFFLINE mode = 0 1
Other PLC Running in Primary mode = 1 0
Other PLC Running in Standby mode = 1 1
PLCs have matching Logic = 0
PLCs do not have matching Logic = 1
This PLC set as unit A = 0
This PLC set as unit B = 1
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0 = Unlocated Variables being transferred
1 = Unlocated Variables not being transferred
0 = The hot stand by has not been activated
1 = Hot stand by is active

%SW61 Hot Standby Status Register

The following gives additional detail on the various parts of the Status Register.

%SW61.0 to 3: These bits display the state of the local and remote Hot Standby PLCs.

%SW61.4: This bit is set = 1 whenever a logic mismatch is detected between the Primary and Standby PLCs. This means that either the MID, CID or LID is different on the two PLCs. Under this condition, if bit 13 of the Command Register is set = 0, bit 1 of the Status Register will be set = 1.

%SW61.5: This bit identifies the order reported by the Copro at start time depends on the range of the MAC addresses.

- If the A/B designation is A, then bit 5 will be set = 0.
- If the A/B designation is B, then bit 5 will be set = 1.

%SW61.14: If set = 1 it indicates that a logic mismatch has been detected that disallows Unlocated Variables to be transferred from Primary to Standby. This feature was canceled for UNITY V1.1 because it was determined that a switchover with a partial application context posed too great a hazard.

%SW61.15: If set = 1 it indicates that the Copro device is set up correctly and working.

8 Diagnostics and Troubleshooting

In This Chapter

The module provides information on diagnostics and troubleshooting in the following forms:

- LED status indicators on the front of the module provide general information on the module's status.
- Status Data can be viewed through the Configuration/Debug port using ProSoft Configuration Builider (PCB) software.
- Status data values can be transferred from the module to processor controller tags and can be monitored manually or by customer-created logic.

8.1 LED Status Indicators

LED	Color	Status	Description
PRT1	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
PRT2	Green	On	Port not used in application
		Off	Port not used in application
PRT3	Green	On	Port not used in application
		Off	Port not used in application
ERR1	Red	Off	The PTQ-104C is working normally.
		On	The PTQ-104C module program has recognized an application error.
ERR2	N/A		Not used in application
ERR3	Red	On	Configuration Error
Active	Green	On	The LED is on when the module recognizes a processor and is able to communicate if the [Backplane Data Movement] section specifies data transfer commands.
		Off	The LED is off when the module is unable to communicate with the processor. The processor either absent or not running.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or the battery is not present. The battery LED will illuminate briefly upon the first installation of the module or if the unit has been un-powered for an extended period of time. This behavior is normal, however should the LED come on in a working installation please contact ProSoft Technology.

If the module is not operating and the status LEDs are not illustrated in the table above, please call ProSoft Technology for technical assistance.

8.1.1 Ethernet LED Indicators

LED	State	Description
Data	Off	No activity on the Ethernet port.
	Green Flash	The Ethernet port is actively transmitting or receiving data.
Link	Off	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	Green Solid	Physical network connection detected. This LED must be on solid for Ethernet communication to be possible.

8.2 Diagnostics Menu

The Diagnostics menu in ProSoft Configuration Builder (PCB) for this module is arranged as a tree structure, with the *Main* Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu displayed when connected to the module is the *Main* menu.

Because this is a text-based menu system, commands are entered by typing the command letter from the computer keyboard in the Diagnostics window. The module does not respond to mouse movements or clicks.

8.2.1 Required Hardware

There are two ways to connect the module with PCB – Serial or Ethernet.

When connecting serially to the module, ProSoft Technology recommends the following minimum hardware to connect the computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.

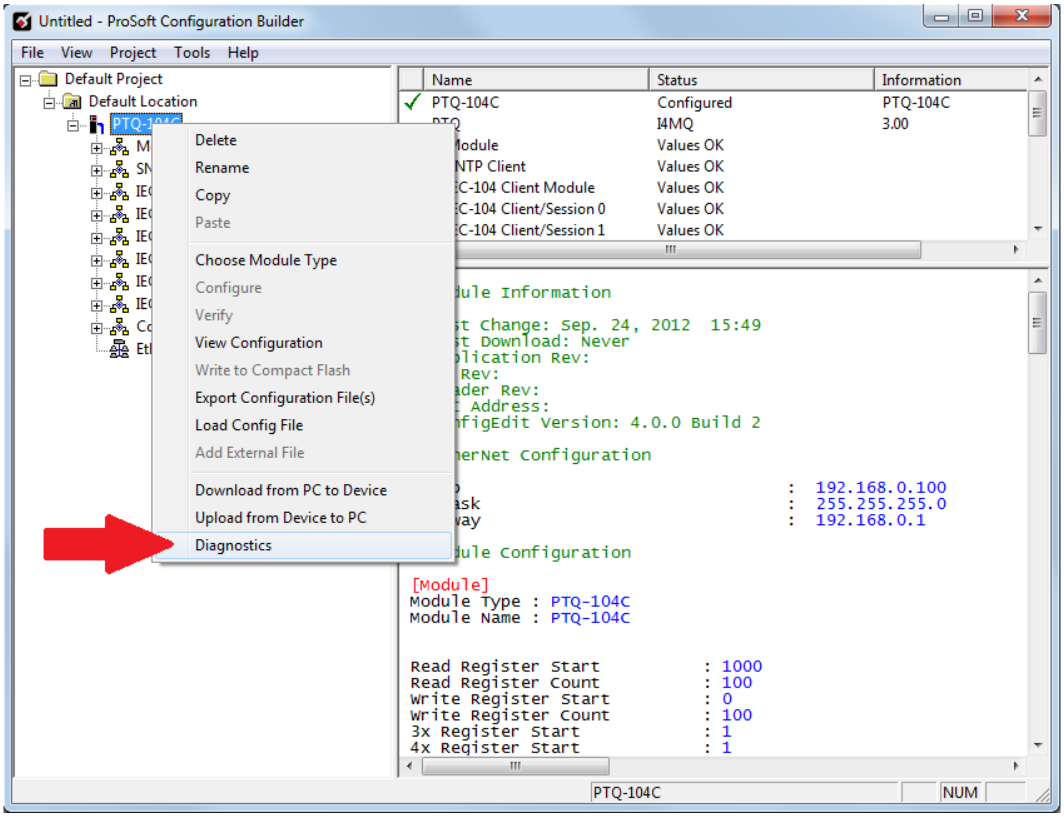
Note: When connecting for the first time, a serial connection is required since the IP address of the module has not been configured yet. Setting the IP address of the module is detailed on page 19.

Once the IP address parameters have been configured in PCB and downloaded (serially) to the module, PCB connectivity over Ethernet can be established.

8.2.2 Serial Connection to Diagnostics Menu

Connect to the module's Configuration/Debug serial port.

1 In the PCB project tree, right-click the PTQ-104C icon to open a shortcut menu.
On the shortcut menu, choose **DIAGNOSTICS.**

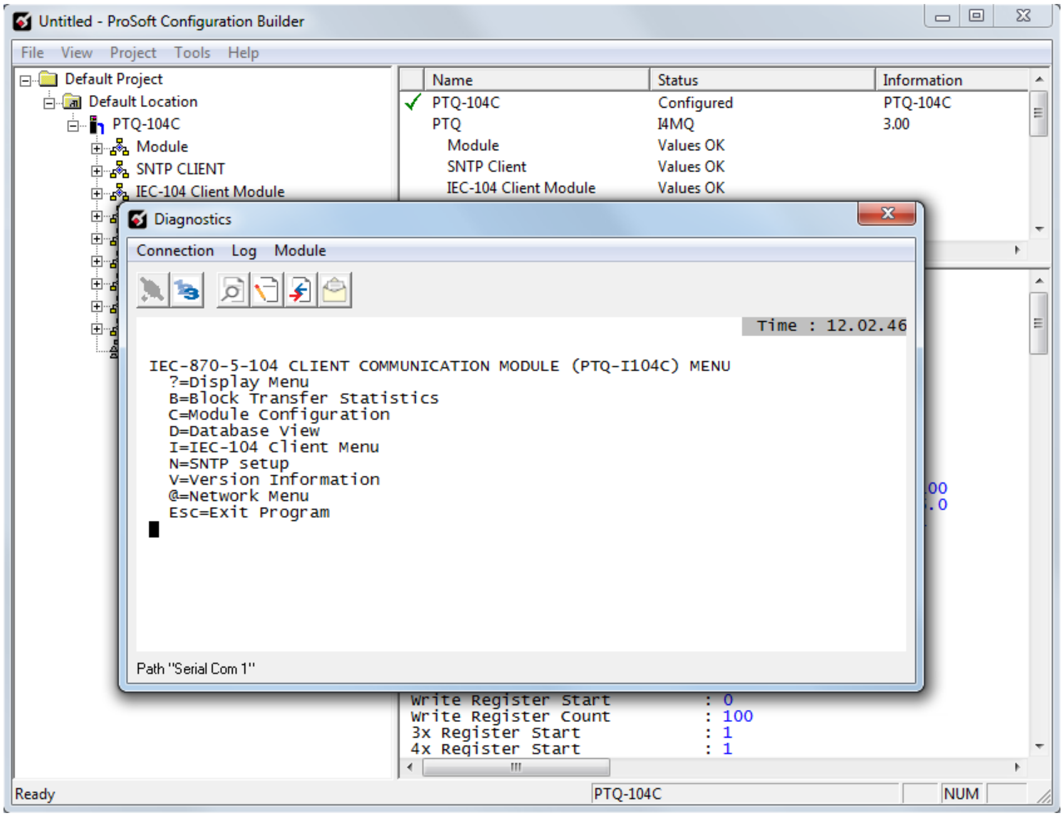


	Name	Status	Information
✓	PTQ-104C	Configured	PTQ-104C
	PTQ	I4MQ	3.00
	odule	Values OK	
	NTP Client	Values OK	
	C-104 Client Module	Values OK	
	C-104 Client/Session 0	Values OK	
	C-104 Client/Session 1	Values OK	

```
dule Information

t Change: Sep. 24, 2012  15:49
t Download: Never
lication Rev:
 Rev:
ader Rev:
 Address:
figEdit Version: 4.0.0 Build 2

erNet Configuration

                                  :  192.168.0.100
ask                               :  255.255.255.0
ay                                :  192.168.0.1

dule Configuration

[Module]
Module Type : PTQ-104C
Module Name : PTQ-104C


Read Register Start          : 1000
Read Register Count          : 100
Write Register Start         : 0
Write Register Count         : 100
3x Register Start            : 1
4x Register Start            : 1
```

This action opens the **DIAGNOSTICS** dialog box.

2 Press **[?]** to open the Main Menu.



Name	Status	Information
✓ PTQ-104C	Configured	PTQ-104C
PTQ	I4MQ	3.00
Module	Values OK	
SNTP Client	Values OK	
IEC-104 Client Module	Values OK	

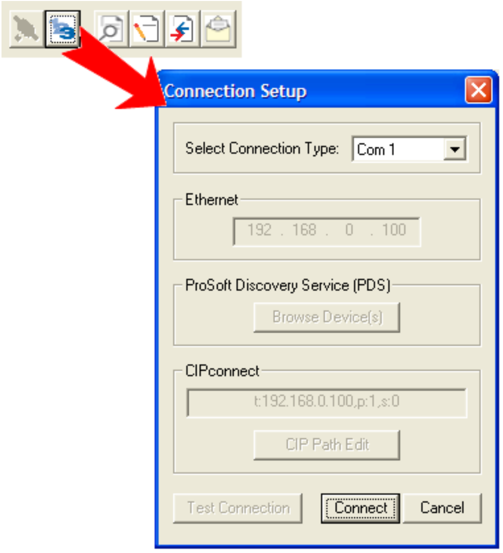
```
                                                   Time : 12.02.46

IEC-870-5-104 CLIENT COMMUNICATION MODULE (PTQ-I104C) MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=IEC-104 Client Menu
  N=SNTP setup
  V=Version Information
  @=Network Menu
  Esc=Exit Program
█
```

Path "Serial Com 1"

```
Write Register Start          : 0
Write Register Count          : 100
3x Register Start             : 1
4x Register Start             : 1
```

If there is no response from the module, follow these steps when trying to connect serially:

1 Click to configure the connection. On the Connection Setup dialog box, select a valid com port or other connection type supported by the module.



Select Connection Type: Com 1

192 . 168 . 0 . 100

t:192.168.0.100,p:1,s:0

2 Verify that the null modem cable is connected properly between the computer's serial port and the module. A regular serial cable will not work.

3 On computers with more than one serial port, verify that the communication program is connected to the same port that is connected to the module.

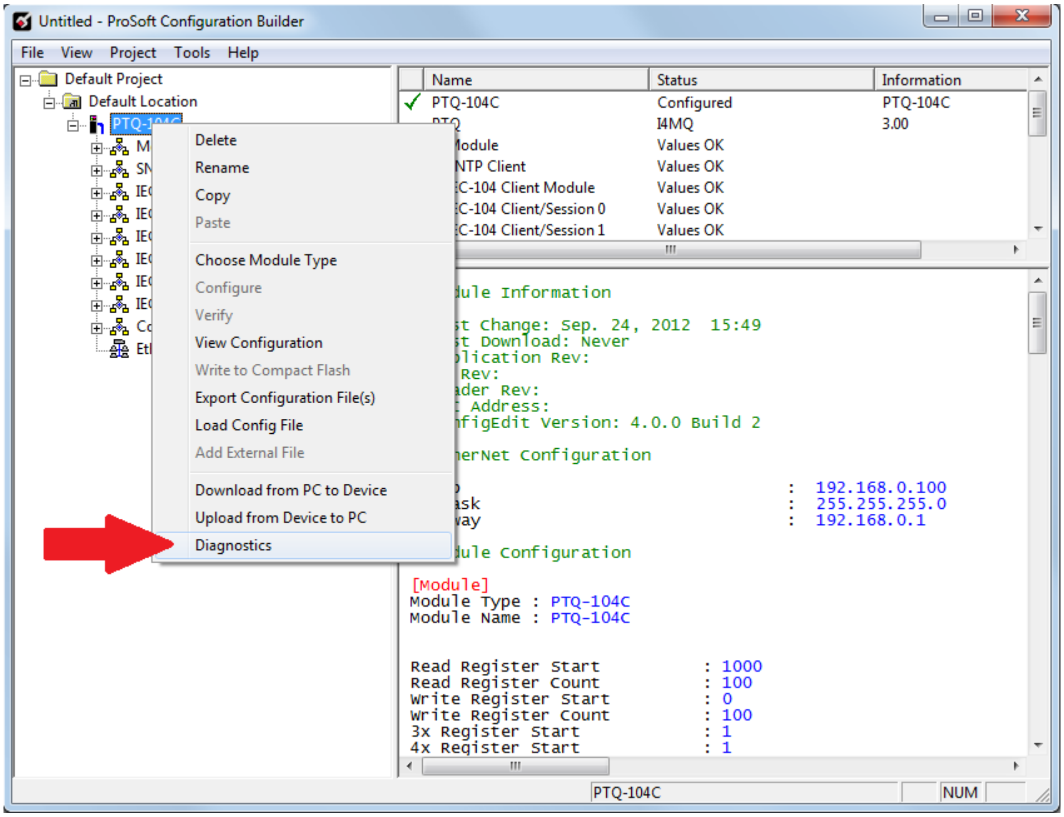
If a connection cannot be established, contact ProSoft Technology for assistance.

8.2.3 Ethernet Connection to Diagnostics Menu

This connection can only be made if an IP address has been configured and downloaded to the module.

Connect the module's Ethernet port to the Ethernet network.

1 In the PCB project tree, right-click the PTQ-104C icon to open a shortcut menu.
On the shortcut menu, choose **DIAGNOSTICS.**

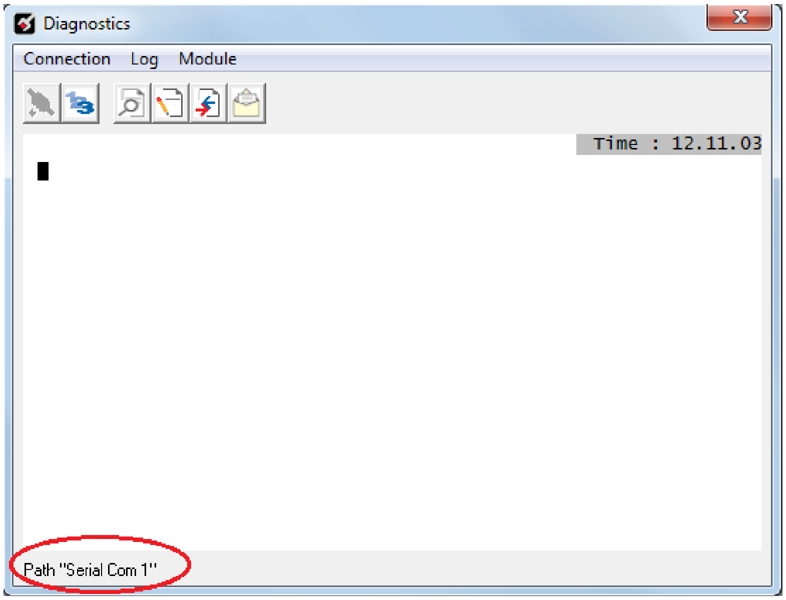


Name	Status	Information
✓ PTQ-104C	Configured	PTQ-104C
PTQ	I4MQ	3.00
Module	Values OK	
NTP Client	Values OK	
C-104 Client Module	Values OK	
C-104 Client/Session 0	Values OK	
C-104 Client/Session 1	Values OK	

```
dule Information

t Change: Sep. 24, 2012  15:49
t Download: Never
plication Rev:
 Rev:
ader Rev:
 Address:
nfigEdit Version: 4.0.0 Build 2

erNet Configuration

                                 :  192.168.0.100
ask                              :  255.255.255.0
way                              :  192.168.0.1

dule Configuration

[Module]
Module Type : PTQ-104C
Module Name : PTQ-104C

Read Register Start          : 1000
Read Register Count          : 100
Write Register Start         : 0
Write Register Count         : 100
3x Register Start            : 1
4x Register Start            : 1
```

This action opens the **DIAGNOSTICS** dialog box.

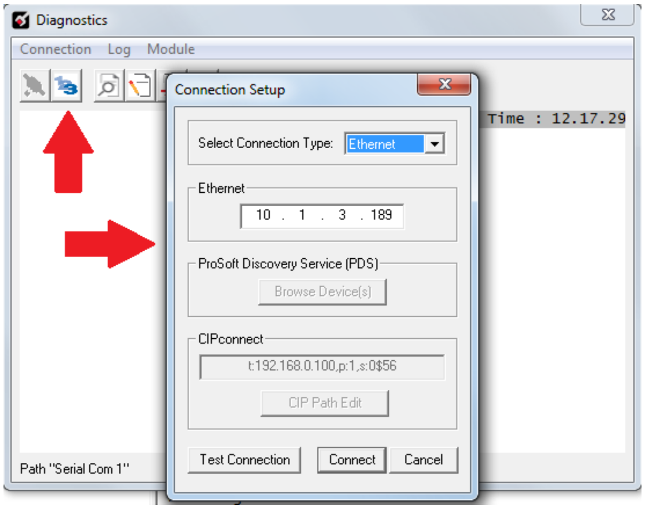
2 If the bottom of the window reveals a Serial connection, the connection path will need to be adjusted to an Ethernet connection.



Time : 12.11.03

Path "Serial Com 1"

3 Click the *Setup Connection* icon or select *Connection>Setup Connection* on the window. Select Ethernet as the Connection Type and enter the IP address in the field below.



Time : 12.17.29

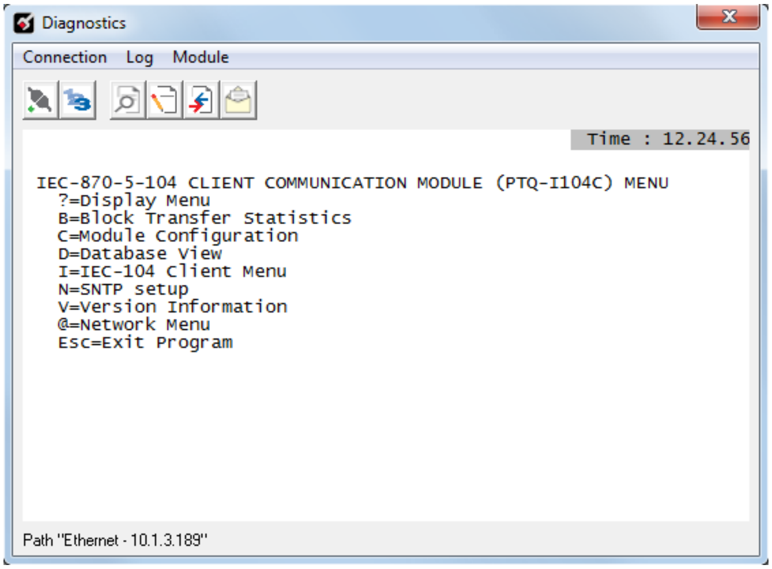
Select Connection Type: Ethernet

Ethernet: 10 . 1 . 3 . 189

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

Path "Serial Com 1"

4 Click *Connect* and the Diagnostics window will appear with the correct path. Press **[M]** and menu options will appear.



```
                                                     Time : 12.24.56

 IEC-870-5-104 CLIENT COMMUNICATION MODULE (PTQ-I104C) MENU
   ?=Display Menu
   B=Block Transfer Statistics
   C=Module Configuration
   D=Database View
   I=IEC-104 Client Menu
   N=SNTP setup
   V=Version Information
   @=Network Menu
   Esc=Exit Program
```

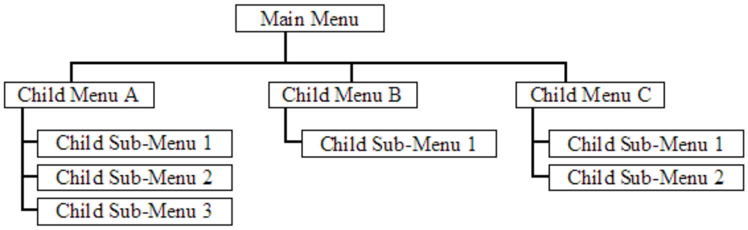
Path "Ethernet - 10.1.3.189"

If a connection cannot be established, contact ProSoft Technology for assistance.

8.2.4 Navigation in Diagnostics Window

All sub-menus in PCB contain commands to redisplay the menu or return to the previous menu. Pressing **[M]** on the keyboard will bring up the Main Menu.

The organization of the menu structure is represented in simplified form in the following illustration:

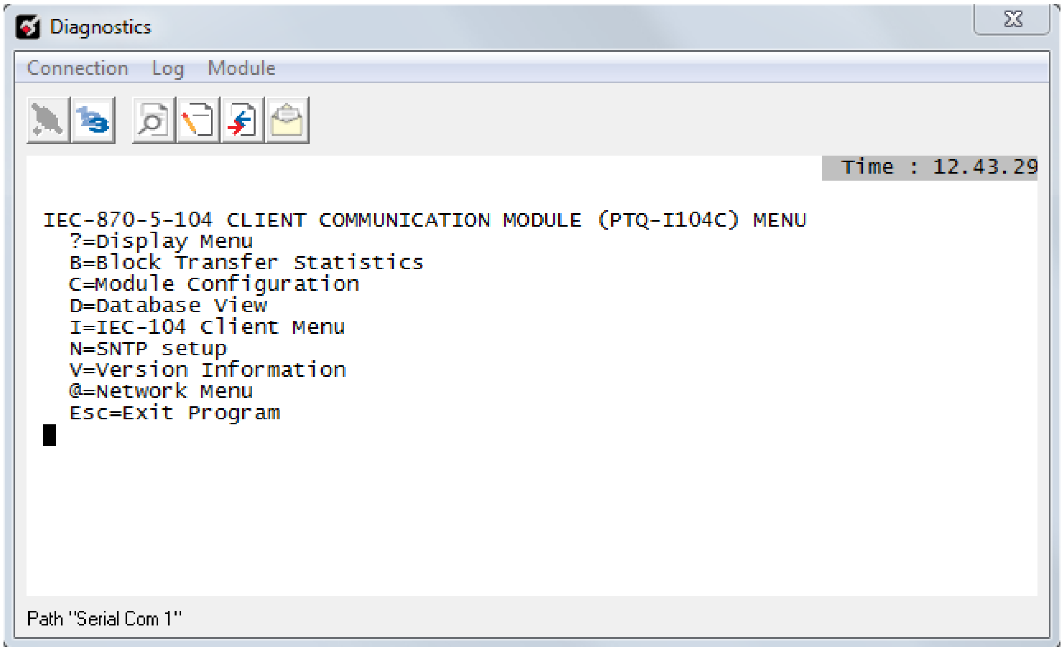


Keystrokes

The keyboard commands on these menus are not case sensitive.

The menus use a few special characters (**[?], [-], [+], [@]**) that must be entered exactly as shown. Some of these characters will require the **[SHIFT], [CTRL]** or **[ALT]** keys. For example, on US English keyboards, enter the **[?]** command as **[SHIFT][/].**

Also, take care to distinguish capital letter **[I]** from lower case letter **[L]** and number **[1]**. Likewise for capital letter **[O]** and number **[0].**



```
                                                        Time : 12.43.29

IEC-870-5-104 CLIENT COMMUNICATION MODULE (PTQ-I104C) MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=IEC-104 Client Menu
  N=SNTP setup
  V=Version Information
  @=Network Menu
  Esc=Exit Program
```

Path "Serial Com 1"

Displaying the Current Menu

Press **[?]** at any time in any menu to display the current menu options.

Block Transfer Statistics

Press **[B]** to display the block transfer statistics menu. Use this command to view the module's status in the rack including block counts, I/O transfer settings, Hot Standby settings, etc.

Viewing Module Configuration

Press **[C]** to display the current configuration and statistics for the module including IEC 60870-5-104 protocol-specific settings.

Database View

Press **[D]** to view the data of the module's database. Press **[S]** to 'show' the IEC 60870-5-104 values coming in/out of the IEC 60870-5-104 network.

To navigate in this menu, press **[N]** to display the next 100 values of the data base. Press **[P]** to display the previous 100 values of the data base.

Note: The values do not refresh automatically. The same keystroke (**[D], [P],** or **[N]**) is needed to refresh the values.

IEC-104 Client Menu

Press **[I]** (Upper case 'i') to open the IEC-104 Client menu. Press **[S]** to 'show' the first Session configuration.

To navigate in this menu, press **[N]** to display the next Session configuration. Press **[P]** to display the previous Session configuration.

SNTP Setup Menu

Press **[N]** to view the SNTP setup. Refer to SNTP Support for more information on configuring and using this function.

Version Information Menu

Press **[V]** to view the module's firmware information.

Network Menu

Press **[@]** to view the module's IP address information including the subnet mask and gateway. Press **[S]** to 'show' the information.

Exiting the Program

Caution: Some of the commands available from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures.
Use these commands when specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. To prevent an unwanted command to be executed, please be careful when pressing keys.

Press **[ESC]** to restart the module and force all drivers to be loaded. The module will use the configuration stored in the module's Flash memory to configure the module.

9 Reference

In This Chapter

9.1 Product Specifications

9.1.1 *Standards*

The standards used in developing the product are listed in the following table.

Publication	Title
IEC 870-5-101	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-101 Amendment 1	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-101 Second Edition 2003-02	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Link Transmission Procedures
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Application Information Elements
IEC 60870-5-5	Basic Application Functions

The PTQ-104C IEC 60870-5-104 Client Communication Module allows Quantum compatible processors to interface easily with IEC 60870-5-104 protocol-compatible servers.

The PTQ-104C module acts as an input/output module between the IEC-60870-5-104 Ethernet network and the Quantum processor. Data transfer between the module and the processor is asynchronous from the actions on the network. Databases are defined by the user for the module to hold data as required by the protocol.

The PTQ-104C module is a powerful communication interface for Quantum processors. Developed under license from Schneider Electric, the module incorporates proprietary backplane technology that enables powerful data access to the Quantum processor.

The module supports the Quantum 140CPU67160 Hot Standby processor in the Unity Pro programming environment.

9.1.2 *Hot Standby Support*

The module provides support for 140CPU6716000 Hot Standby processor with Unity Pro programming software. Look for the HSBY (Hot Standby) icon for special notes relating to the support and configuration of the module.

HSBY Note: For detailed understanding of HSBY specification, refer to Chapter 6.

9.1.3 *General Specifications*

- Single Slot - Quantum backplane compatible
- The module is recognized as an Options module and has access to PLC memory for data transfer
- Configuration data is stored in non-volatile memory in the ProTalk® module
- Up to six modules can be placed in a rack

- Local rack - The module must be placed in the same rack as processor
- Compatible with all common Quantum programming packages, including Concept (version 2.6 or higher), Unity Pro (version 2.2 or higher), ProWORX (version 2.20 or later), and ModSoft
- Quantum data types supported: 0x, 1x, 3x, 4x
- High speed data transfer across backplane provides quick data update times

9.1.4 Hardware Specifications

Specification	Value
Backplane Current Load	1100 mA maximum @ 5 Vdc ± 5%
Operating Temperature	0°C to 60°C (32°F to 140°F)
Storage Temperature	-40°C to 85°C (-40°F to 185°F)
Relative Humidity	5% to 95% (without condensation)
Vibration	Sine vibration 4-100 Hz in each of the 3 orthogonal axes
Shock	30G, 11 mSec. in each of the 3 orthogonal axes
Dimensions (HxWxD), Approx.	250 x 40.34 x 103.85 mm 9.84 x 1.59 x 4.09 in
LED Indicators	Active Status Ethernet port Status Battery Status Debug/Config Port Activity Module Status
Debug/Configuration Port (Debug)	
CFG Port (DEBUG)	DB-9M PC Compatible RS-232 only No hardware handshaking
Application Ports	
Application Ethernet port	RJ45 Connector Link and Activity LED indicators Electrical Isolation 1500 V rms at 50 Hz to 60 Hz for 60 s, 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

9.1.5 Functional Specifications

The PTQ-104C module sends data read/write commands to a server/slave on the network.

The module has 4000 words of user defined internal register space that are accessible to the protocol driver and to the Quantum processor memory. Any of the supported database types can be individually located (within the total database size limit of 4000 words) and each database point is mapped within the module and can be assigned to one or more Groups. The supported IEC-870-5-104 database point types are:

Database Type	Description	Event Generation	Point Size
M_SP_NA	Monitored single-point database	State Change	1 bit

Database Type	Description	Event Generation	Point Size
M_DP_NA	Monitored dual-point database	State Change	2 bits
M_ST_NA	Monitored step-point database	State Change	1 byte
M_BO_NA	Monitored Bitstring database	Dead-band	2 wprds
M_ME_NA	Monitored normalized-point database	Dead-band	1 word
M_ME_NB	Monitored scaled-point database	Dead-band	1 word
M_ME_NC	Monitored short-float point database	Dead-band	2 words
M_IT_NA	Monitored integrated total database	N/A	2 words
C_SC_NA	Command single-point database	N/A	1 bit
C_DC_NA	Command dual-point database	N/A	2 bit
C_RC_NA	Command step-point database	N/A	1 byte
C_SE_NA	Command normalized-point database	N/A	1 word
C_SE_NB	Command scaled-point database	N/A	1 word
C_SE_NC	Command short-float point database	N/A	2 words
C_BO_NA	Command 32-bit Bitstring database	N/A	2 words

IEC 60870-5-104 Client Specifications

The IEC 60870-5-104 Client functionality supported by the module includes:

- The IEC 60870-5-104 communication driver is built in accordance to the approved IEC specification
- The module functions as a Client on the network supporting data read/write commands to IEC 60870-5-104 servers on the network
- Four TCP sockets
- Supports clock synchronization commands from client or from the Quantum
- Supports Group interrogation and Counter interrogation commands.
- Optional user defined list of acceptable client host IP addresses
- Configurable Common ASDU address (sector) and Information Object Address
- An IEC Interoperability Document for the ProTalk is available which fully documents data types supported by the module

Hot Standby

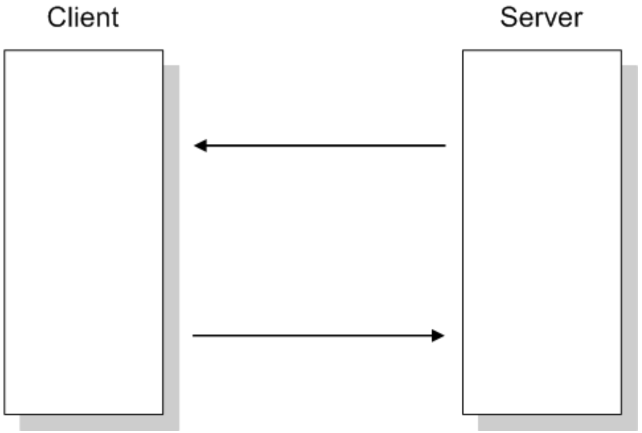
- Hot Standby features support the SE 140 671 CPU
- Supports up to six PTQ-104C Hot Standby modules per rack
- Module automatically detects Hot Standby system

9.2 IEC-60870-5-104 Protocol Implementation

The intent of this section is to provide a quick understanding of how the PTQ-104C module implements the IEC-60870-5-104 protocol, without going into complex details of the specification.

The IEC-60870-5-104 protocol applies to Telecontrol equipment and systems with data transmission for monitoring and controlling geographically widespread processes. This protocol consists of the IEC-60870-5-101 protocol, with the addition of TCP/IP as the transport mechanism.

Any application with the IEC-60870-5-104 protocol consists of a client (Controlling Station) and one or more servers (Controlled Stations). The client constantly monitors and controls the data from each server in the TCP/IP network.



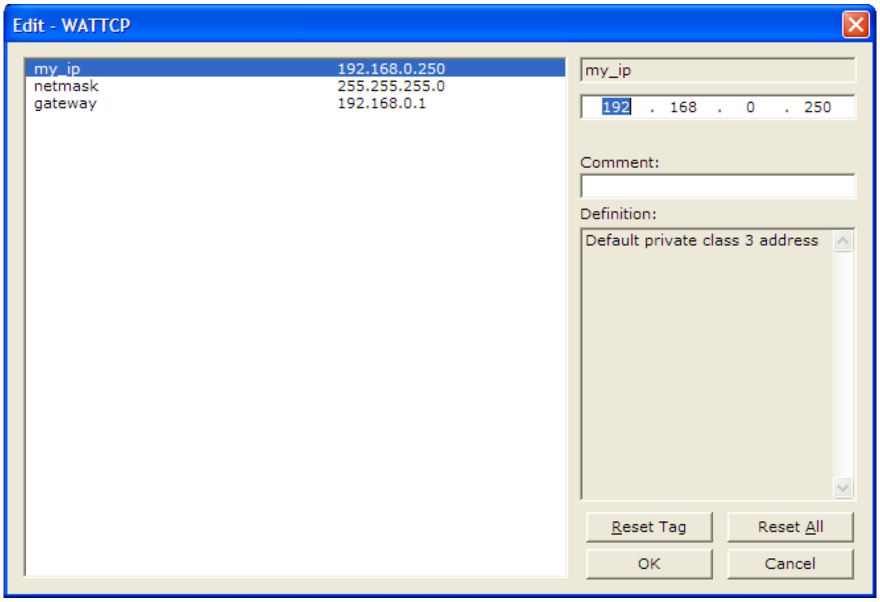
The PTQ-104C operates as an IEC-60870-5-104 client; sending read or write commands to remote server(s).

9.2.1 Module Address

The PTQ-104C module is identified at transport level using the IP Address.

IP Address

The PTQ-104C module is identified by a unique IP address on the TCP/IP network. Edit the WATTCP.CFG configuration file (or use the configuration tool) to enter a valid IP address. The following example lists the default contents of the WATTCP.CFG file:



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

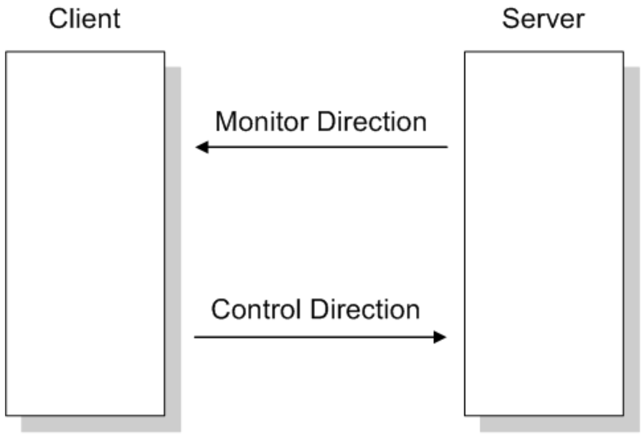
In this example, the PTQ-104C module is identified by IP address 192.168.0.250 in the IEC-60870-5-104 network, with a netmask (subnet mask) of 255.255.255.0 and a default gateway address of 192.168.0.1.

9.2.2 Monitor Direction and Control Direction: Point Definition

The protocol specification defines two directions of data: monitor direction and control direction.

Monitor Direction: The direction of transmission from a server to a Client

Control Direction: The direction of transmission from a Client to a server

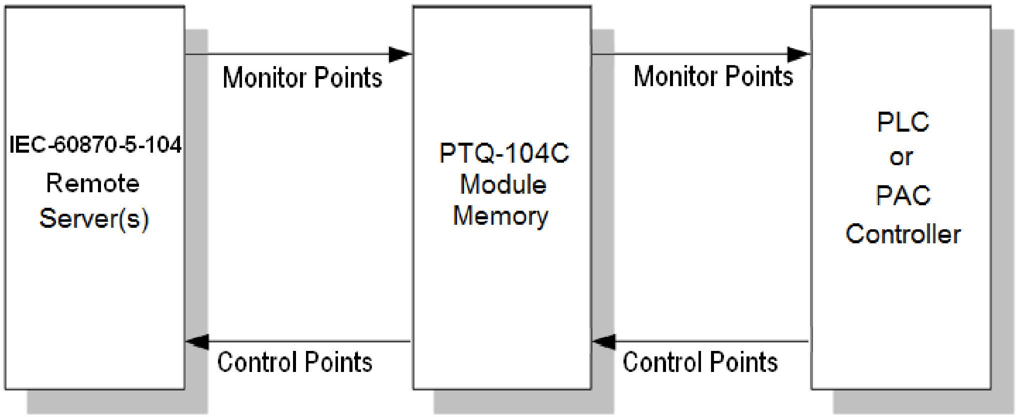


The points that are typically transferred from a server to a Client are also known as **Monitor Points** (or Monitor Information Objects). The points that are typically transferred from a Client to a server are also known as **Control Points** (or Command Information Objects).

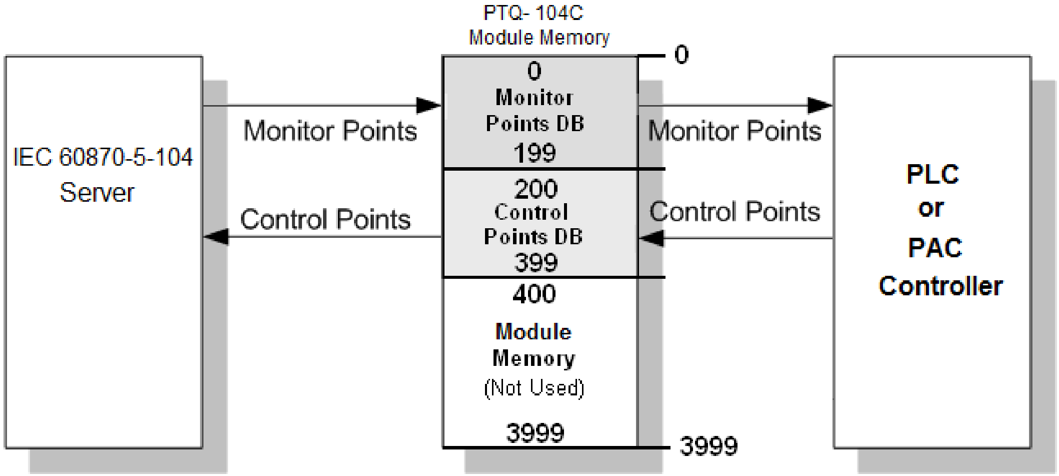
The PTQ-104C contains an internal database of 4000 16-bit words. The monitor and control points must be associated to the database addresses in the PTQ-104C. To configure the points for the PTQ-104C, follow these steps:

1. Calculate the number of monitor and control points for the application.
2. Calculate the PTQ-104C database regions that are required for the application, based on the number of monitor and control points. Define two separate regions. Remember that each data type stores a different quantity of data (for example, M_SP_NA uses one bit, M_ST_NA uses one byte, and so on).
3. Configure each point within its PTQ-104C database region.

4 Make sure the other parts of the application correctly update the module database regions associated with the configured 104C data types, as shown in the following illustration.



All points must be configured in the correct location in the PTQ-104C database in order to be properly updated by other parts of the application. Make sure to configure the control points and monitor points in separate areas of the PTQ-104C database. The following illustration shows an example configuration:

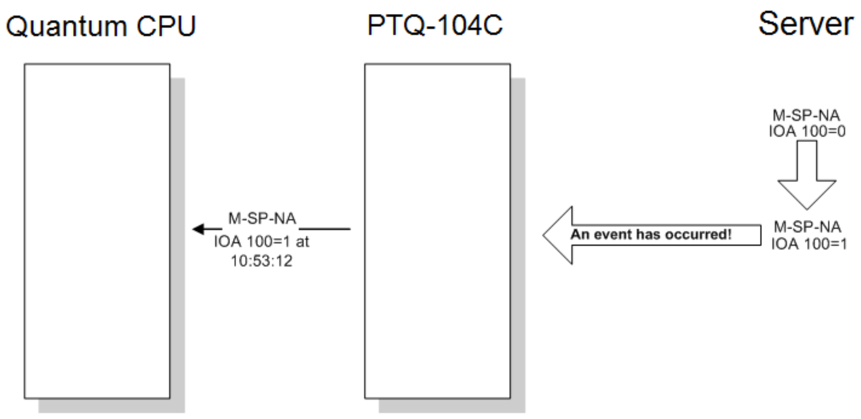


In this example, all monitor points are located between database addresses 0 and 199, and all control points are located between address 200 and 399.

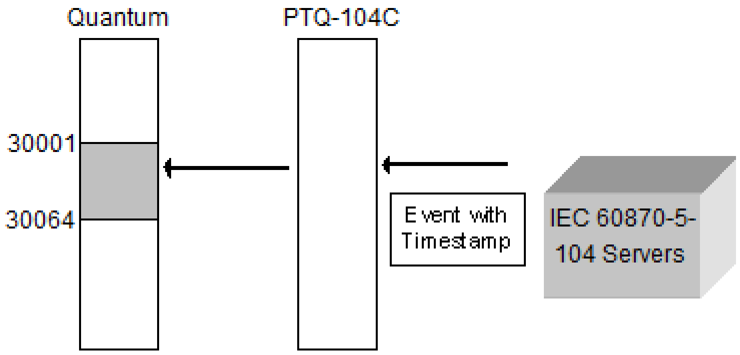
9.2.3 Events

The event pass-through functionality must be initially enabled by the user in PCB. The 'Pass-Through Events' parameter will need to be set to 'Yes' in the [Module] section.

In order to improve communication efficiency, most applications will require the Client to periodically poll for data changes with a higher priority than polling for other monitored data. Every time a data point changes, the server can send this information as an *event*, typically with date and time information indicating when the change occurred.



The following illustration shows the basic idea of the event pass-through functionality. When the module receives the event from the remote device, it will build block 9903, which will be copied to the processor at the configured memory address:



The module supports a buffer queue of 99 events per data type. When the queue is full, the module will delete the oldest event in the queue, replacing it with the newest event received for that data type.

Event Pass-Through Block Format

The block that is copied from the module to the processor is structured in the format below. The user must write controller logic to process this data.

Each block can contain up to 4 events. The number of events per block will typically depend on the rate between how fast the module receives the events and how fast these can be passed to the processor (typically depends on the processor scan rate).

Block Format for Read

Block Format from Module (%IW or 3x Register Data)

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9903 for the block.
2	Event Count	This field contains the number of events present in the block. Values of 1 to 4 are valid.
3 to 16	Event 1	Event message
17 to 30	Event 2	Event message
31 to 44	Event 3	Event message
45 to 58	Event 4	Event message
59 to 61	Spare	Not used
62	Event count in queue	Number of messages left in event buffer
63	Event Overflow	Event buffer overflow

The format of each 14 word data region in the block is shown in the following table.

Word Offset	Definitions	Description
0	Session Index	This field contains the session index used to define the controlled unit in the module from which the event was generated.
1	Sector Index	This field contains the sector index used to define the database within the controlled unit from which the event was generated.
2	COT	This field contains the COT for the event message received from the IED. If the size of the COT is a single byte, the originator address will always be zero. The COT is in the LSB and the originator address is in the MSB.
3	Reserved	This field is reserved for future use and is added here to keep the structure double-word aligned for all platforms.
4 to 5	Point Index	This field contains the point index in the remote device that generated the event.

Word Offset	Definitions	Description
6	ASDU Type	This field contains the ASDU type code for the data contained in the message.
7	Milliseconds and Seconds	This word contains the seconds and milliseconds when the event occurred.
8	Minutes and Hours	This field contains the minutes and hours the event occurred.
9	Month and Day	This field contains the month and day of the month the event occurred.
10	Year	This field contains the year the event occurred.
11	Qualifier	This field contains the point qualifier, quality or sequence value as described in the protocol specification.
12 to 13	Value	This field contains the double word value for the point associated with the event message.

The processor logic should recognize the event count value greater than zero and read all events in the block. This value should be reset to zero to prepare the logic for the next incoming block.

9.3 Cable Connections

The PTQ-104C module has the following communication connections on the module:

- One Ethernet port (RJ45 connector)
- One RS-232 Configuration/Debug port (DB9 connector)

9.3.1 Ethernet Connection

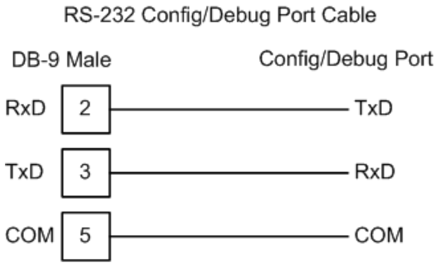
The PTQ-104C module has an RJ45 port located on the front of the module, labeled *Ethernet*, for use with the TCP/IP network. The module is connected to the Ethernet network using an Ethernet cable between the module's Ethernet port and an Ethernet switch or hub.

Warning: The PTQ-104C module is NOT compatible with Power Over Ethernet (IEEE802.3af / IEEE802.3at) networks. Do NOT connect the module to Ethernet devices, hubs, switches or networks that supply AC or DC power over the Ethernet cable. Failure to observe this precaution may result in damage to hardware, or injury to personnel.

Important: The module requires a static (fixed) IP address that is not shared with any other device on the Ethernet network. Obtain a list of suitable IP addresses from the network administrator BEFORE configuring the Ethernet port on this module.

9.3.2 RS-232 Configuration/Debug Port

This port permits the ProSoft Configuration Builder software to view configuration and status data within the module. The (null modem) cable for communications on this port is shown in the following diagram:



9.4 General Module Status (Block 9250) Table

This section contains the block structure of the PTQ-104C module's general status data from Block 9250. This block is copied automatically to the database starting at address 4000. To request the complete status block, refer to Read Status (9250) (page 52).

Word Offset in Block	Data Field(s)	Description
0	Block ID Code	Block Identification Code 9250
1	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
2 to 3	Product Name	This two-word data area contains the text values representing the product name.
4 to 5	Revision	This two-word data area contains the text values for the revision number.
6 to 7	Op Sys #	This two-word data area contains the text values for the operating system number.
8 to 9	Run Number	This two-word data area contains the text values for the run number.
10	Read Block Count	This word contains the total number of block read operations successfully executed.
11	Write Block Count	This word contains the total number of block write operations successfully executed.
12	Reserved	Reserved
13	Error Block Count	This word contains the total number of block transfer errors.
14	Event Message Count	This word contains the number of event messages waiting to send to the processor.
15	Event Message Overflow	This word contains a value of 0 if the event message buffer has not overflowed. If the event buffer overflows, this word will be set to a value of 1.
16	Session Count	This word contains the number of sessions configured in the module.

Word Offset in Block	Data Field(s)	Description
17	Current Command	This word contains the index of the current command being executed in the command list.
18	Command Busy Flag	This word is set to zero if no command is currently being executed and waiting on a response. If the word is set to 1, a command is currently executing.
19	Command Count	This word contains the count of the number of commands configured for the module.
20	Command Delay	This word contains the command delay counter preset. There is a fixed delay between each command to permit the module to perform other operations.
21	Command Queue	This word is set to zero if the command executing is from the command list. If the executing command is from the command queue, the word will be set to 1.
22 to 23	Command Queue Count	This word contains the number of active commands in the command queue for the module. Up to 100 commands can be buffered in this queue. These commands are transferred from the processor to the module using special command blocks.
24	Online Status	This double word value contains a bit for each of the 32 potential sessions in the module. If the bit is set for a session in the double word, the station is online. If the bit is clear, the station is offline. Use this value to determine if commands sent from the processor will have a chance of succeeding.
25	SNTP Valid	NTP time is valid (0=No, 1=Yes)
26	NTP Request	Number of requests to NTP server
27	NTP Response	Number of responses from NTP server
28	SNTP Computation	Number of times SNTP time computed
29	SNTP Set	Number of times SNTP time set
30	NTP Timeout	Number of NTP response timeouts

9.5 Client Status (Block 9251) Table

This section contains the block structure of the PTQ-104C module's Client status data from Block 9251.

Word Offset in Block	Data Field(s)	Description
0	Block ID Code	Block Identification Code 9251
1	Active	Active status of client 0 = Not active 1 = Waiting on StartDT connection 2 = Online (StartDT connection received)

Word Offset in Block	Data Field(s)	Description
2	State	State of Client Socket: -255 = Idle until processor in run -1 = Ready for socket open 0 = Waiting for socket establish 1 = Obsolete (not used) 2 = Read and write data to socket (process data) 3 = Need more data for response so wait for rest of data 50 = Sending test frame 60 = Sending S-frame 70 = Sending StartDT act 80 = Sending StopDT act 100 = End program/close socket 1000 = Close socket (start 2-second timeout) 1001 = Wait for close & abort if timeout 2000 = ARP request/response 2001 = Open Socket
3	Open Count	Number of times socket open attempted
4	Close Count	Number of times socket closed
5	Connect Count	Number of times socket established
6-15	Host IP	ASCII string of remote server IP address (10 words)
16	t0 Timeouts	Number of t0 timeouts
17	t1 Timeouts	Number of t1 timeouts
18	t2 Timeouts	Number of t2 timeouts
19	t3 Timeouts	Number of t3 timeouts
20	Sequence Errors	Number of sequence errors
21	Bad Address Errors	Number of bad address errors
22	Length Errors	Number of length errors
23	Rx Frame Count	Number of frames received on socket
24	Tx Frame Count	Number of frames transmitted on socket
25	Cmd Requests	Number of commands issued from command list and queue
26	Cmd Responses	Number of responses to commands issued from command list and queue.
27	Cmd Error Count	Number of errors recognized when issued from command list or queue
28	Requests	Number of requests from command driver
29	Responses	Number of messages received by command driver
30	Errors Sent	Number of errors sent by command driver (not used)
31	Errors Received	Number of errors received by command driver
32	Configuration Error	Configuration error word for client
33	Current Error	Current error recognized
34	Last Error	Last error recognized
35	Send Number	Send sequence number
36	Rec Number	Received sequence number
37	Ack Number	Last acknowledged sequence number
38	Queue Max	Maximum number of queue (k)
39	Queue Threshold	Queue threshold before S-Frame sent (w)
40	Rec Packets	Number of I-frames received but not acknowledged
41	Queue Now	Number of messages in queue

Word Offset in Block	Data Field(s)	Description
42	Queue First	First index of message in queue
43	Queue Index	Current index of message in queue

9.6 IEC 60870-5-104 Client Interoperability Statement

This companion standard presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, such as the number of octets in the COMMON ADDRESS of ASDUs represent mutually exclusive alternatives. Only one value of the defined parameters is admitted per system.

Other parameters, such as the listed set of different process information in command and in monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming from different manufacturers it is necessary that all partners agree on the selected parameters.

Note: In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should be marked in the white boxes as follows:

□	Function or ASDU is not used
☒	Function or ASDU is used as standardized (default)
R	Function or ASDU is used in reverse mode
B	Function or ASDU is used in standard and reverse mode

The possible selection (blank, X, R, B) is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

9.6.1 System or device

□ System definition

☒ Controlling station definition (Client)

□ Controlled station definition (Server)

9.6.2 Application Layer

Transmission mode for application data

Mode 1 (Least significant octet first), as defined in clause 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common Address of ASDU

(System-specific parameter)

☐ One octet ☒ Two octets

Information object address

(System-specific parameter)

☐ One octet ☐ Structured
☐ Two octets ☐ Unstructured
☒ Three octets

Cause of transmission

(System-specific parameter)

☐ One octet ☒ Two octets (with originator address)

Length of APDU

(System-specific parameter, specify the maximum length of the APDU per system)

The Maximum length of the APDU is 253 (default). The maximum length may be reduced by the system.

Maximum length of 253 per APDU system

9.6.3 Selection of standard ASDUs

Process information in monitor direction

(Station-specific parameter; mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <1>	Single-point information	M_SP_NA_1
☐ <2>	Single-point information with time tag	M_SP_TA_1
☒ <3>	Double-point information	M_DP_NA_1
☐ <4>	Double-point information with time tag	M_DP_TA_1
☒ <5>	Step position information	M_ST_NA_1
☐ <6>	Step position information with time tag	M_ST_TA_1
☒ <7>	Bitstring of 32 bit	M_BO_NA_1
☐ <8>	Bitstring of 32 bit with time tag	M_BO_TA_1
☒ <9>	Measured value, normalized value	M_ME_NA_1
☐ <10>	Measured value, normalized value with time tag	M_ME_TA_1
☒ <11>	Measured value, scaled value	M_ME_NB_1
☐ <12>	Measured value, scaled value with time tag	M_ME_TB_1
☒ <13>	Measured value, short floating point value	M_ME_NC_I

Type ID	Description	Type
☐ <14>	Measured value, short floating point value with time tag	M_ME_TC_1
☒ <15>	Integrated totals	M_IT_NA_1
☐ <16>	Integrated totals with time tag	M_IT_TA_1
☐ <17>	Event of protection equipment with time tag	M_EP_TA_1
☐ <18>	Packed start events of protection equipment with time tag	M_EP_TB_1
☐ <19>	Packed output circuit information of protection equipment with time tag	M_EP_TC_1
☐ <20>	Packed single-point information with status change detection	M_PS_NA_1
☐ <21>	Measured value, normalized value without quality description	M_ME_ND_1
☒ <30>	Single-point information with time tag CP56Time2a	M_SP_TB_1
☒ <31>	Double-point information with time tag CP56Time2a	M_DP_TB_1
☒ <32>	Step position information with time tag CP56Time2a	M_ST_TB_1
☒ <33>	Bitstring of 32 bit with time tag CP56Time2a	M_BO_TB_1
☒ <34>	Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
☒ <35>	Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
☒ <36>	Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
☒ <37>	Integrated totals with time tag CP56Time2a	M_IT_TB_1
☐ <38>	Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
☐ <39>	Packed start events of protection equipment with time tag CP56time2a	M_EP_TE_1
☐ <40>	Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Process information in control direction

(Station-specific parameter, mark each Type ID '**X**' if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <45>	Single command	C_SC_NA_1
☒ <46>	Double command	C_DC_NA_1
☒ <47>	Regulating step command	C_RC_NA_1
☒ <48>	Set point command, normalized value	C_SE_NA_1
☒ <49>	Set point command, scaled value	C_SE_NB_1
☒ <50>	Set point command, short floating point value	C_SE_NC_1
☒ <51>	Bitstring of 32 bit	C_BO_NA_1
☒ <58>	Single command with time tag	C_SC_TA_1

Type ID	Description	Type
	CP56Time2a	
☒ <59>	Double command with time tag CP56Time2a	C_DC_TA_1
☒ <60>	Regulating step command with time tag CP56Time2a	C_RC_TA_1
☒ <61>	Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
☒ <62>	Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
☒ <63>	Set point command, short float value with time tag CP56Time2a	C_SE_TC_1
☒ <64>	Bitstring of 32 bit with time tag CP56Time2a	C_BO_TA_1

Either the ASDUs of the set <45>-<51> or of the set <58>-<64> are used.

System information in monitor direction

(Station-specific parameter, mark "**X**" if used)

Type ID	Description	Type
☒ <70>	End of initialization	M_EI_NA_1

System information in control direction

(Station-specific parameter, mark each Type ID '**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <100>	Interrogation command	C_IC_NA_1
☒ <101>	Counter interrogation command	C_CI_NA_1
☒ <102>	Read command	C_RD_NA_1
☒ <103>	Clock synchronization command	C_CS_NA_1
☒ <104>	Test command	C_TS_NB_1
☒ <105>	Reset process command	C_RP_NC_1
☒ <106>	Delay acquisition command	C_DVD_NA_1

Parameter in control direction

(Station-specific parameter, mark each Type ID '**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

Type ID	Description	Type
☒ <110>	Parameter of measured value, normalized value	P_ME_NA_1
☒ <111>	Parameter of measured value, scaled value	P_ME_NB_1
☒ <112>	Parameter of measured value, short floating point value	P_ME_NC_1
☒ <113>	Parameter activation	P_AC_NA_1

File transfer

(Station-specific parameter, mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

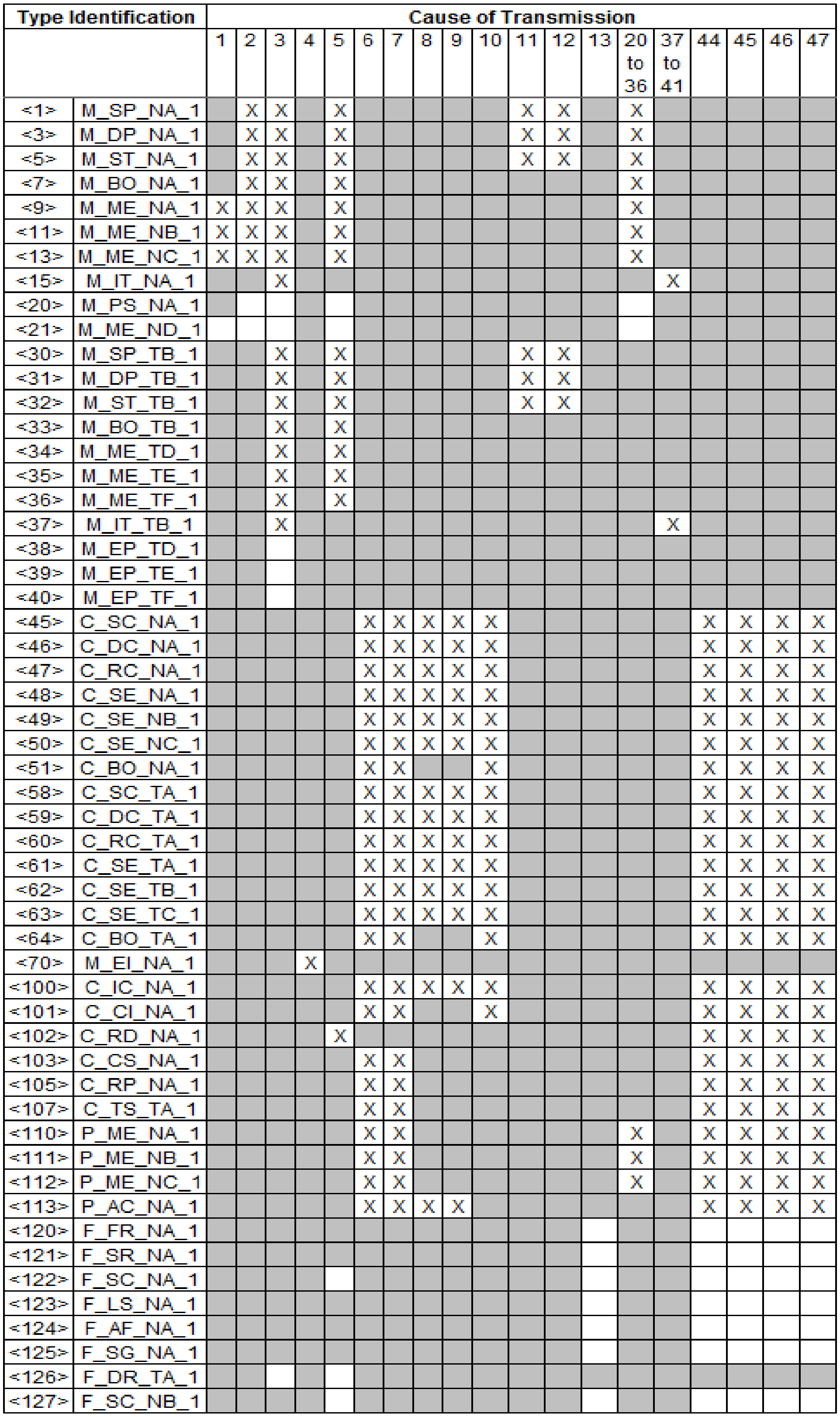
Type ID	**Description**	**Type**
□ <120>	File ready	F_FR_NA_1
□ <121>	Section ready	F_SR_NA_1
□ <122>	Call directory, select file, call file, call section	F_SC_NA_1
□ <123>	Last section, last segment	F_LS_NA_1
□ <124>	Ack file, ack section	F_AF_NA_1
□ <125>	Segment	F_SG_NA_1
□ <126>	Directory	F_DR_TA_1

9.6.4 Type identifier and cause of transmission assignments

(Station-specific parameters)

See the table on the next page.

- Shaded boxes: option not required
- Blank boxes: functions or ASDU not used
- Mark Type Identification/Cause of Transmission combinations: '**X**" if only used in standard direction.



Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1		X	X		X						X	X		X					
<3>	M_DP_NA_1		X	X		X						X	X		X					
<5>	M_ST_NA_1		X	X		X						X	X		X					
<7>	M_BO_NA_1		X	X		X									X					
<9>	M_ME_NA_1	X	X	X		X									X					
<11>	M_ME_NB_1	X	X	X		X									X					
<13>	M_ME_NC_1	X	X	X		X									X					
<15>	M_IT_NA_1			X												X				
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			X		X						X	X							
<31>	M_DP_TB_1			X		X						X	X							
<32>	M_ST_TB_1			X		X						X	X							
<33>	M_BO_TB_1			X		X														
<34>	M_ME_TD_1			X		X														
<35>	M_ME_TE_1			X		X														
<36>	M_ME_TF_1			X		X														
<37>	M_IT_TB_1			X												X				
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1						X	X	X	X	X						X	X	X	X
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1						X	X			X						X	X	X	X
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X
<59>	C_DC_TA_1						X	X	X	X	X						X	X	X	X
<60>	C_RC_TA_1						X	X	X	X	X						X	X	X	X
<61>	C_SE_TA_1						X	X	X	X	X						X	X	X	X
<62>	C_SE_TB_1						X	X	X	X	X						X	X	X	X
<63>	C_SE_TC_1						X	X	X	X	X						X	X	X	X
<64>	C_BO_TA_1						X	X			X						X	X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1					X											X	X	X	X
<103>	C_CS_NA_1						X	X									X	X	X	X
<105>	C_RP_NA_1						X	X									X	X	X	X
<107>	C_TS_TA_1						X	X									X	X	X	X
<110>	P_ME_NA_1						X	X							X		X	X	X	X
<111>	P_ME_NB_1						X	X							X		X	X	X	X
<112>	P_ME_NC_1						X	X							X		X	X	X	X
<113>	P_AC_NA_1						X	X	X	X							X	X	X	X
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1																			
<127>	F_SC_NB_1																			

9.6.5 Basic Application Functions

Station initialization

(Station-specific parameter, mark "**X**" if function is used)

☒ Remote initialization

Cyclic data transmission

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Cyclic data transmission

Read procedure

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Read procedure

Spontaneous transmission

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous

(Station-specific parameter, mark each information type "**X**" where both a Type ID without time and corresponding Type ID with time are issued in response to a single spontaneous change of a monitored object)

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular Information Object Addresses for which double transmission is enabled are defined in a project-specific list.

□ Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 and M_PS_NA_1

□ Double-point information M_DP_NA_1, MDP_TA_1 and M_DP_TB_1

□ Step position information M_ST_NA_1, M_ST_TA_1 and M_ST_TB_1

□ Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1 and M_BO_TB_1

□ Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 and M_ME_TD_1

□ Measured value, scaled value M_ME_NB_1, M_ME_TB_1 and M_ME_TE_1

□ Measured value, short floating point number M_ME_NC_1, M_ME_TC_1 and M_ME_TF_1

Station Interrogation

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ global

☒ group 1

☒ group 2

☒ group 3

☒ group 4

☒ group 5

☒ group 6

☒ group 7

☒ group 8

☒ group 9

☒ group 10

☒ group 11

☒ group 12

☒ group 13

☒ group l4

☒ group 15

☒ group 16

Clock synchronization

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Clock synchronization

□ Day of week used

□ RES1, GEN (time tag substituted/not substituted) used

□ SU-bit (summertime) used (Optional)

Command transmission

(Object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Direct command transmission

☒ Direct set point command transmission

☒ Select and execute command

☒ Select and execute set point command

☒ C_SE_ACTTERM used

☒ No additional definition

☒ Short pulse duration (duration determined by a system parameter in the outstation)

☒ Long pulse duration (duration determined by a system parameter in the outstation)

☒ Persistent output

☒ Supervision of maximum delay in command direction of commands and set point commands

60 seconds Maximum allowable delay of commands and set point commands

Transmission of Integrated totals

(Station- or object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Mode A: Local freeze with spontaneous transmission

☒ Mode B: Local freeze with counter interrogation

☒ Mode C: Freeze and transmit by counter-interrogation commands

☒ Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously

☒ Counter read

☒ Counter freeze without reset

☒ Counter freeze with reset

☒ Counter reset

☒ General request counter

☒ Request counter group 1

☒ Request counter group 2

☒ Request counter group 3

☒ Request counter group 4

Parameter loading

(Object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Threshold value

☒ Smoothing factor

☒ Low limit for transmission of measured value

☒ High limit for transmission of measured value

Parameter activation

(Object-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Activation/Deactivation of persistent cyclic or periodic transmission of the addressed object

Test procedure

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

B Test procedure

File transfer
*(Station-specific parameter, mark "**X**" if function is used)*

File transfer in monitor direction

- □ Transparent file
- □ Transmission of disturbance data of protection equipment
- □ Transmission of sequence of events
- □ Transmission of sequence of recorded analogue values
- □ File transfer in control direction
- □ Transparent file

Background scan

(Station-specific parameter, mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Background scan

Definition of time outs

Parameter	**Default Value**	**Remarks**	**Range**
t_0	30 seconds	Time-out of connection establishment	1 to 1000 seconds
t_1	15 seconds	Time-out of send or test APDUs	1 to 255 seconds
t_2	10 seconds	Time-out for acknowledges in case of no data messages ($t_2 < t_1$)	1 to 255 seconds
t_3	20 seconds	Time-out for sending test frames in case of a long idle time	1 to 300 seconds

Maximum range of values for configurable time-outs: accuracy 1 second.

Maximum number of outstanding I format APDUs k and latest acknowledge APDUs (w)

Parameter	**Default Value**	**Remarks**	**Range**
k	6 APDUs	Maximum number of message to hold in queue	1 to 12
w	4 APDUs	Threshold value when to send ack	1 to 8

Maximum range of values *k*: 1 to 32767 (2^{15}-1) APDUs, accuracy 1 APDU

Maximum range of values *w*: 1 to 32767 (2^{15}-1) APDUs, accuracy 1 APDU

(Recommendation: *w* should not exceed two-thirds of *k*).

Port number

Parameter	Value	Remarks
Port number	2404	In all cases

Redundant connections

0	Number N of redundancy group connections used

RFC 2200 suite

RFC 2200 is an official Internet Standard which describes the state of standardization of protocols used in the Internet as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

☒ Ethernet 802.3

□ Serial X.21 interface

□ Other selection from RFC 2200

9.7 Frequently Asked Questions

9.7.1 How do I configure the module?

The ProTalk module requires a configuration file to be edited and downloaded using ProSoft Configuration Builder (PCB). An overview of the backplane communications with the PLC can be found in the Module Configuration section of this manual (page 62). For more in-depth information, refer to Backplane Data Exchange (page 44).

9.7.2 What kind of data transfer rates can I expect between the PLC and the module?

Data transfer rates between the PLC and the module depend on a number of variables, among them are the number of words being transferred per command, the amount of other network traffic at the time data is being transferred, and overall processor scan times.

9.7.3 Does the module work in a remote rack?

The module is designed to be located in the chassis with the PLC and will not operate in a remote chassis.

9.7.4 Can I use the module in a hot backup system?

The PTQ-104C module supports the 140CPU67160 Hot Standby processor. Refer to Hot Standby Support (page 81) for setup and configuration instructions.

10 Support, Service & Warranty

In This Chapter

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. Module configuration and associated ladder files, if any
2. Module 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 module, if any.

Note: *For technical support calls within the United States, an after-hours answering system allows 24-hour/7-days-a-week pager access to one of our qualified Technical and/or Application Support Engineers. 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

10.1 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 on the Product CD/DVD or go to www.prosoft-technology/warranty

Documentation is subject to change without notice

Index