Quantum with Unity Pro
Modbus Plus Network Modules
User Manual

eng June 2005

35010487_00

a brand of Schneider Electric

Telemecanique

Document Set

Presentation

This package contains the following manuals:

- Quantum and Premium Communication Architecture Reference Manual
- 140 EIA 921 00 Quantum AS-i-Bus Interface Module User Manual
- Quantum TCPIP/IP Configuration User Manual
- Quantum Modbus Plus Network Modules
- Quantum Ethernet Modules User Manual
- 140 ESI 062 10 Quantum ASCII Interface Module User Manual

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

⚡ The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

⚠ This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death, serious injury, or equipment damage.

⚠ WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

PLEASE NOTE

Electrical equipment should be serviced only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. This document is not intended as an instruction manual for untrained persons.

About the Book

At a Glance

Document Scope

This document describes the Modbus Plus networking and communication of the Quantum automation system with Unity Pro.

This document is valid for Unity Pro from version 2.0.

Validity Note

The data and illustrations found in this documentation are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be construed as a commitment by Schneider Electric.

Related Documents

Title of Documentation	Reference Number
Modbus Plus Planning and Installation Guide	890USE10000
Quantum and Premium Communication Architecture Reference Manual	Part of this package
Quantum Experts and Communication Reference Manual	UNYUSE10010V20E
Grounding and Electromagnetic Compatibility of PLC Systems User Manual	UNYUSE10010V20E
Communication Library	UNYUSE40020V20E

Note: The above mentioned documents are only available in online form at this time.

Product Related Warnings

Schneider Electric assumes no responsibility for any errors that may appear in this document. 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 Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this product related warning can result in injury or equipment damage.

User Comments

We welcome your comments about this document. You can reach us by e-mail at techpub@schneider-electric.com

Modbus Plus Network

I

At a Glance

Overview

This part of the document provides an introduction to the topic of Modbus Plus networks. It mainly deals with the communication types that can be found in Modbus Plus network application.

What's in this Part?

This part contains the following chapters:

Intoducing the Modbus Plus Network

1

Overview

Introduction

This chapter contains general information about Modbus Plus Networks.

What's in this Chapter?

This chapter contains the following topics:

Introducing the Modbus Plus Network

Overview

Modbus Plus is a local area network system for industrial control applications. Networked devices can exchange messages for the control and monitoring of processes at remote locations in an industrial plant.

The network also provides an efficient means for servicing input/output subsystems. Modbus Plus Distributed I/O (DIO) Drop Adapters and I/O Blocks (Momentum and TIO) can be placed at remote I/O sites to allow the application to control field devices over the network link.

For a detailed description of the Modbus Plus Network consult the ***Modbus Plus Network Planning and Installation Guide***.

Types of Communication

The following table shows the 4 different types of communication available on a Modbus Plus Network:

Communication Type	Parameter Setup	Remarks
Distributed I/O	During configuration	Allows connection of standard Quantum I/O to the Modbus Plus. DIO is limited to the local Modbus Plus segment
Peer Cop	During configuration	Publisher/Subscriber service, limited to the local Modbus Plus segment
Global Data	During configuration	Broadcasting service, limited to the local Modbus Plus segment
Application driven	Parameters handled through Function Blocks under the control of the User program	Allows routing and therefor not limited to the local Modbus Plus segment

Configuration of the Modbus Plus Network

The following table shows the 4 steps to configure a Modbus Plus Network

Step	Action	Configuration Tool
1	creation of the Modbus Plus logic network(s)	Project browser
2	configuration of the Modbus Plus logic network(s)	
3	adding NOM modules to the configuration (if required)	Hardware configuration window
4	association of the communication module(s) with the logic network(s).	

Benefit

This configuration allows from the second step onwards, to design your communication application (you do not have to have the hardware to start working) and use the simulator to test its operation.

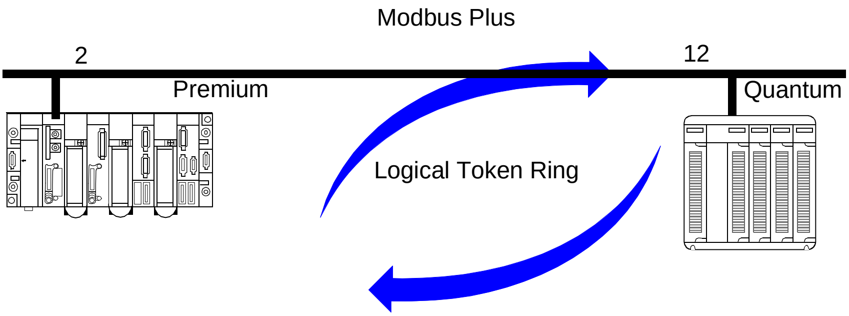
Communication in Modbus Plus Networks

Overview

Communication via Modbus Plus enables data exchange through all devices connected to the bus. Modbus Plus protocol is based on the principle of a Logical-Token-Bus (Logical Token passing). Every station in a network is identified using an address between 1 and 64, and accesses the network once a Token has been received. Double addresses are not permitted.

Communication channel

Example for a Modbus Plus communication channel



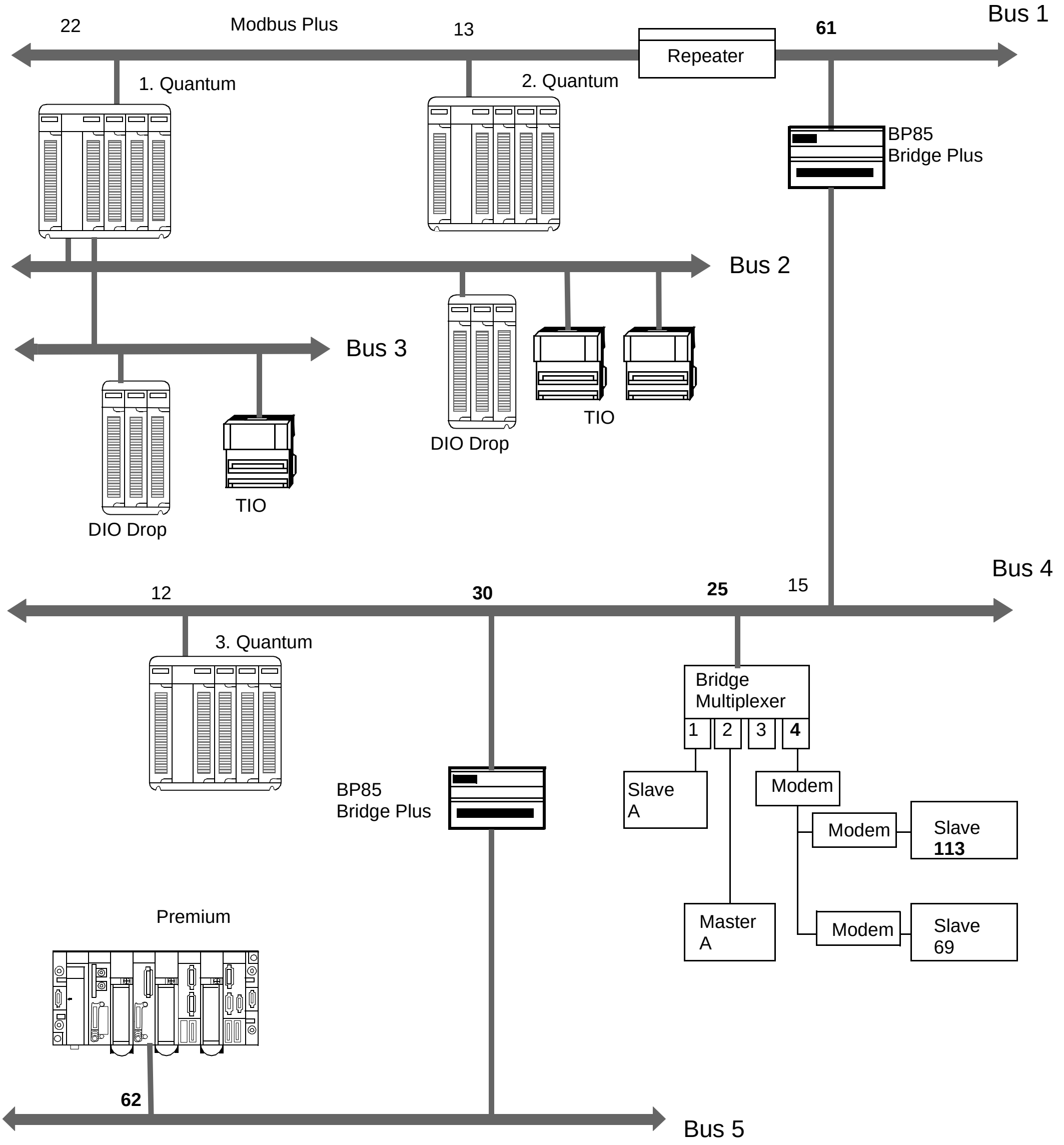
A Modbus Plus communication channel has three main functions:

- Point-to-Point exchange via message service according to Modbus protocol.
- Broadcast exchange of global data between all nodes taking part in the exchange.
- Multi-point exchange of specific data via Peer Cop.

Example for a Modbus Plus network

Overview

The example shows a segmented Modbus Plus network with 5 busses



The following table describes Modbus Plus network busses

Bus	Description
1	● connects the first Quantum via a NOM module in slot 4 ● connects the second Quantum via a CPU module in slot 2 ● contains a repeater for expansions ● contains a Bridge BP85 Plus as connection to bus segment 4
2	● connects the first Quantum via the CPU module in slot 2 with a DIO Drop station and two TIOs
3	● connects the first Quantum via the NOM module in slot 3 with a DIO Drop station and one TIO
4	● connects the third Quantum via a CPU module in slot 2 ● contains a Bridge BP85 Plus as connection to bus segment 5 ● contains a Bridge Multiplexer as connection to the serial/Modbus nodes
5	● connects a Premium via a communication module

Segment spanning data exchange

A Quantum controller can exchange data with all connected stations via a Modbus Plus network. The routing information must be entered additionally for every Bridge in the data path. This is done using the function block designed for this task. More detailed information can be found in *Data Exchange on Remote Modbus Plus Networks, p. 63*.

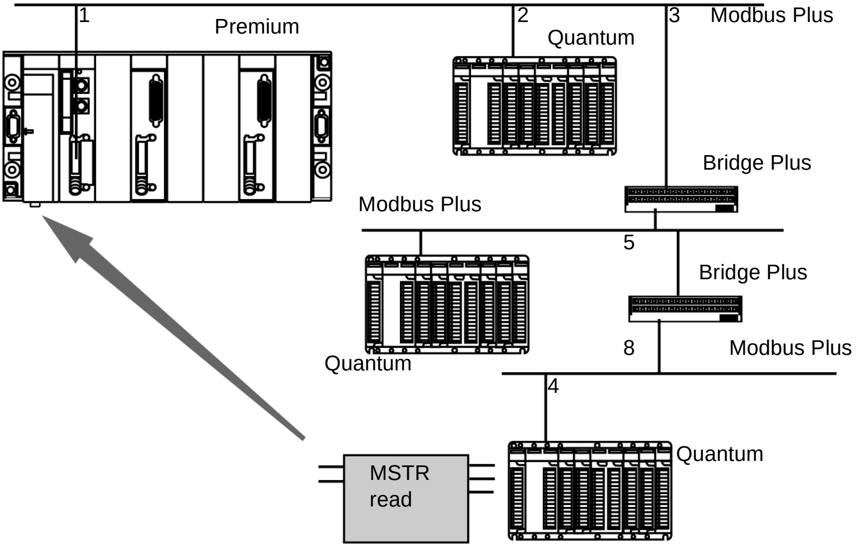
Integration in a Modbus Plus Network

Introduction

In a Modbus Plus architecture, one Quantum PLC application can communicate with a Premium or Atrium PLC and visa-versa.

Quantum with Premium

Quantum PLC communication with a Premium/Atrium PLC is available via a MSTR block. In this case, the Premium or Atrium serves as the server. Consequently all Modbus Plus stations that are connected to a network architecture, up to a maximum of 5 levels, can communicate with it.
Example



The Quantum station sends a read request to the Premium station and uses an address path for this: 8.5.1.0.0 (routing path). The MSTR function block enables the internal words of a Premium or Atrium station to be read or written. The slave register parameter of the MSTR function block gives the address of the internal words %MW directly to the PLC application. This function block also enables the read or RAZ of a Premium or Micro station statistical counter. This request is carried out by the PCMCIA card of the Premium station

Note: For Premium/Atrium PLC communication with a Quantum PLC the addressing must be offset. In order to access an address object n of a Quantum, the communication function of the Premium PLC must have the address n+1.

Modbus Plus Communication Types

2

Overview

Introduction

This chapter describes the Modbus Plus communication types.

What's in this Chapter?

This chapter contains the following sections:

2.1 DIO Drop

Overview

Introduction

This section describes the DIO Drop communication type.

What's in this Section?

This section contains the following topics:

DIO Introduction

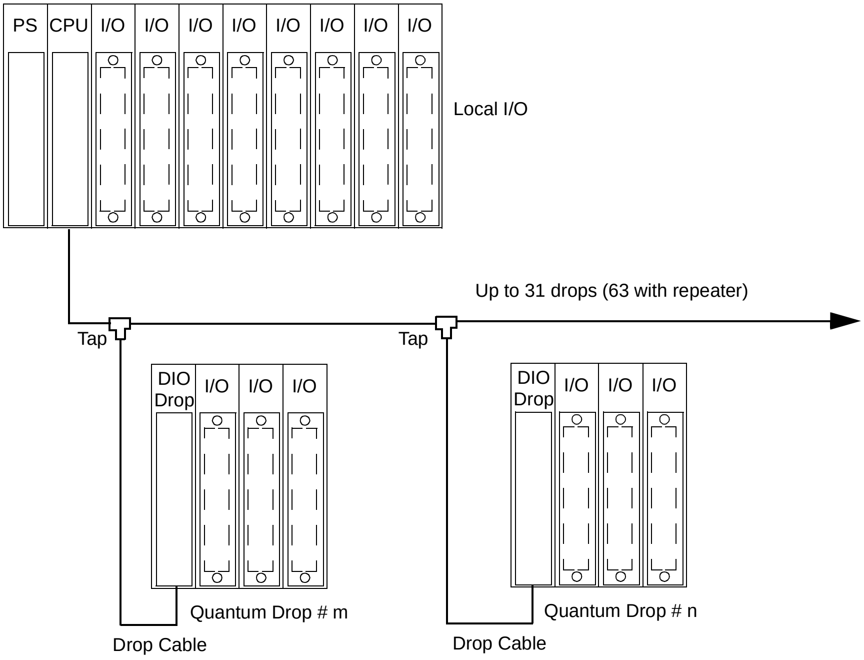
Overview

Quantum DIO is implemented over a Modbus Plus network. The CPU or NOMs module may be the network head via their Modbus Plus ports. Quantum DIO Modbus Plus drop adaptors are specifically designed to link Quantum I/O modules to the head via twisted pair shielded cable (Modbus Plus). The DIO drop modules also provide the I/O with power (maximum 3A) from a 24 VDC or a 115/230 VAC source. Each DIO network supports up to 63 distributed drops using repeaters.

Single-cable Configuration

Single-cable DIO Configuration Figure

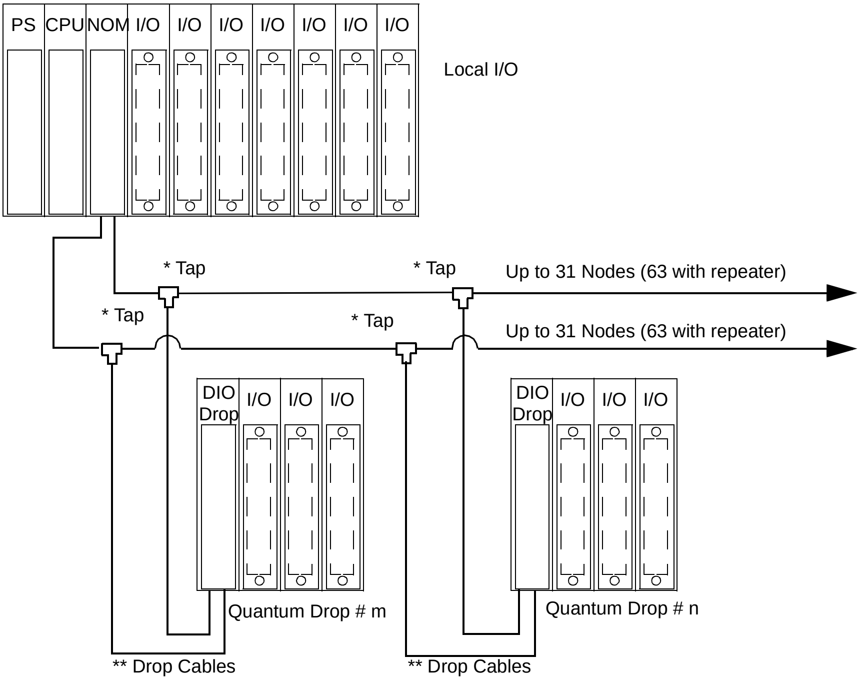
The following figure is an example of a single-cable Quantum DIO configuration.



Dual-cable Configuration

Dual-cable DIO Configuration Figure

The following figure is an example of a dual-cable Quantum DIO configuration.



Note: Dual cables provide systems with added protection against cable breaks or damaged connectors. With two cables connected between the host and each node, no single cable break will disrupt communications.

2.2 Peer Cop

Overview

Introduction

This section describes the Peer Cop communication type.

What's in this Section?

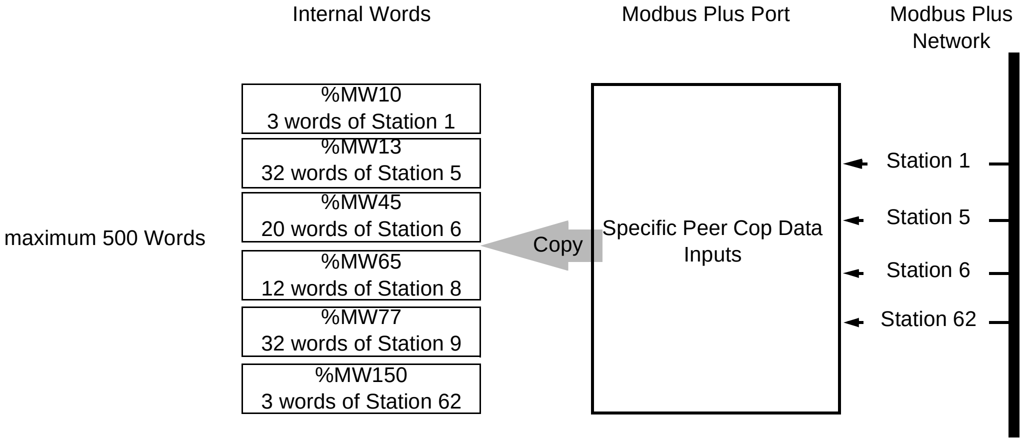
This section contains the following topics:

Specific Inputs and Outputs

Specific inputs and outputs act as a Point-to-Point-Service, that use Multicast-Protocol (Multi-stations). Every message contains one or more receive addresses for transferring the data. This function enables data to be forwarded to several stations without repetition.

Example for specific inputs

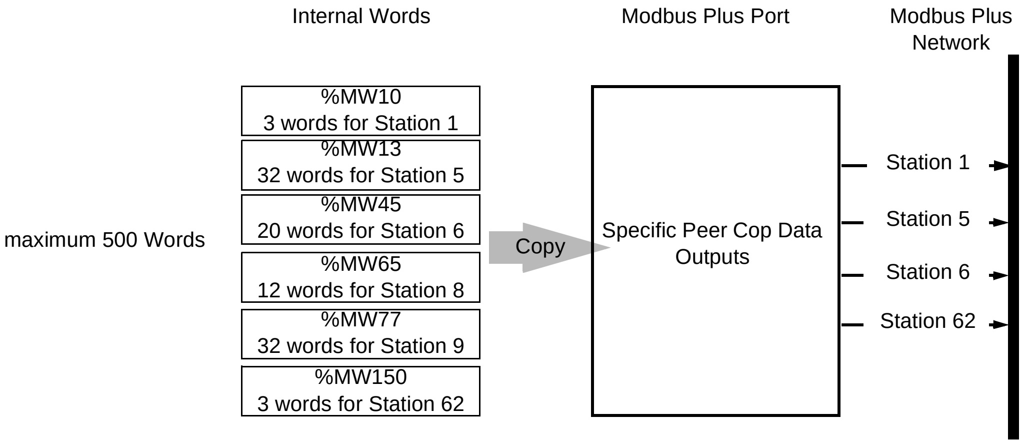
The data blocks are copied in their entirety from the Modbus Plus Port to the internal memory words.
In the following example, the address of the first internal word is %MW10:



Example for specific outputs

The data blocks are copied in their entirety from the memory words reserved in the configuration, to the Modbus Plus Port.

In the following example, the address of the first internal word is %MW10:



Global Data

Overview

When a node passes the token, it can broadcast up to 32 words (16 bits each) of global information to all other nodes on the network. The information is contained in the token frame. The process of sending global data when transmitting the token is controlled independently by the application program in each node.

Global Data table

The global data is accessible to the application programs at the other nodes on the same network. Each node maintains a table of global data sent by every other node on the network. Although only one node accepts the token pass, all nodes monitor the token transmission and read its contents. All nodes receive and store the global data into the table.The table contains separate areas of storage for each node's global data. Each node's application program can selectively use the global data from specific nodes, while other applications can ignore the data. Each node's application determines when and how to use the global data.

Characteristics

Global database applications include

- Time synchronization
- Rapid notification of alarm conditions
- Multicasting of setpoint values and constants to all devices in a common process

This allows uniform and rapid transmission of global data without having to assemble and transmit separate messages to the individual devices. The user's application can determine which data items are useful to nodes on a remote network, and forward them as necessary.

Note: Access to a network's global database is available only to the nodes on that network, because the token is not passed through bridge devices to other networks

Global Input Data

Nodes using Peer Cop can be configured to receive up to 32 words of Global Input data from each of up to 64 source nodes, up to a maximum total of 500 words. Incoming data from each source node can be indexed into up to eight fields for delivery into separate data destinations in the receiving node.

Global Output Data

Nodes using Peer Cop can be configured to send up to 32 words of Global Output data, which is globally broadcast to all active nodes on the network. Destination nodes can be configured to accept or ignore incoming data from specific source nodes..

2.3 Application Specific Communication

Introduction

Overview

Application specific communication is based on function blocks that are integrated into the application program depending on the requirements

The following six function blocks are available:

- *READ_REG: Read register, p. 129*
- *WRITE_REG: Write register, p. 137*
- *CREAD_REG: Continuous register reading , p. 69*
- *CWRITE_REG: Continuous register writing, p. 75*
- *MBP_MSTR: Modbus Plus Master, p. 81*
- *ModbusP_ADDR: Modbus Plus Address, p. 123*

Data exchange

In contrast to Peer Cop and DIO, application specific communication enables data exchange between stations in remote networks.

You can find examples for the following types of data exchange in the application specific communication section:

- *Data Exchange on a Local Segment, p. 59*
- *Data Exchange on Remote Modbus Plus Networks, p. 63*
- *Global Data - Broadcast Services, p. 66*

Modbus Plus Configuration with Unity Pro

II

At a Glance

Overview

This part of the document contains information about Modbus Plus Configuration with Unity Pro.

What's in this Part?

This part contains the following chapters:

Configuring a Logical Network

3

Overview

Introduction

This chapter describes how to configure a logical network.

What's in this Chapter?

This chapter contains the following topics:

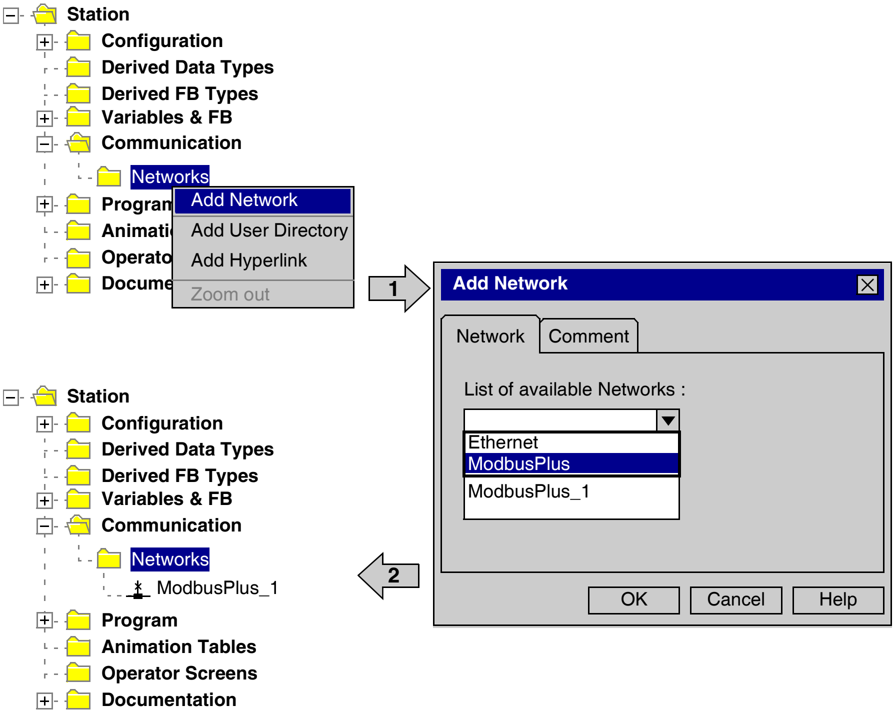
Add a new network to the Communication folder

Add a new network to the Communication folder

After starting a new application, the Communication folder under Station tree branches the Network folder. This folder is empty. Under the Network folder, the user can insert the networks by menu. A click on the right mouse-button above Network pops up a contextual menu. The user selects the type of network he wants to add. For easier use, a network name will be suggested with the prefix of the network type (Ethernet_1 or Modbus+_1). By choosing a new network the next available number for the network is chosen automatically like e.g. Modbus+_1 then Modbus+_2 and so on. At any moment, the user may rename any NetLink.

The user can also attach a comment that describes each configured network. The OK button adds the network as subfolder.

The names of network nodes are also called NetLink. These are the names of logical networks.

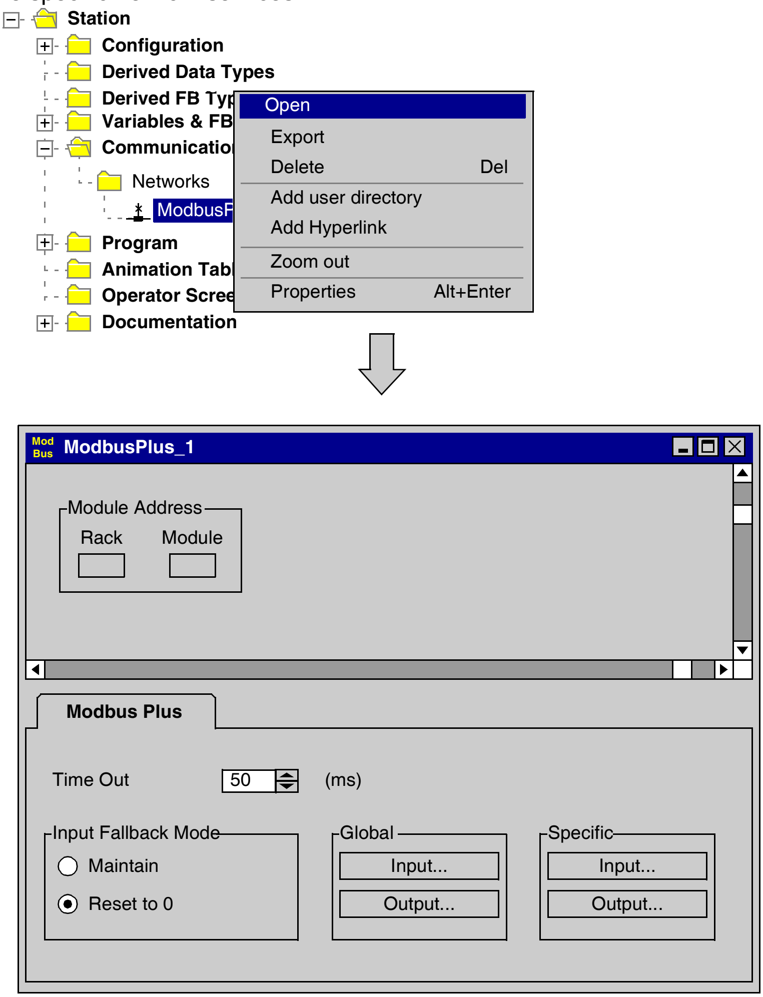


Configure Network

Configure Network

On the network folder, by a double-clicking action or by the Open item on contextual menu, the editor of the corresponding communication screen is opened in order to set the specific network services.

The figure shows the contextual menu to start network properties and the window to set the specific network services



Time Out 50 (ms)

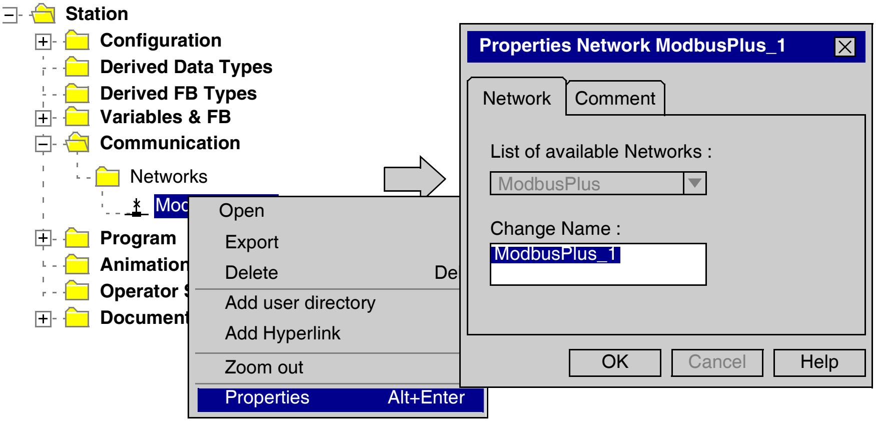
Input Fallback Mode
○ Maintain
● Reset to 0

Properties of a network

Properties of a network

The contextual menu proposes the user to see again the properties of a configured network. Here, the user can change the NetLink name and the associated comment.

The figure shows the Modbus+_1 property window



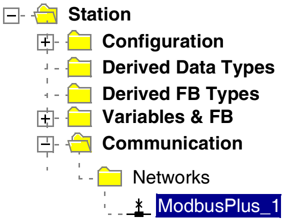
List of available Networks : ModbusPlus

Change Name : ModbusPlus_1

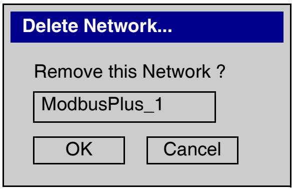
Delete an existing network folder

Delete an existing network folder

With a right-mouse-click on the network folder, a contextual menu appears. Here the user is able to delete the network configuration. In this case, the subfolder of the network will also be removed in application browser.



Station
Configuration
Derived Data Types
Derived FB Types
Variables & FB
Communication
Networks
ModbusPlus_1



Delete Network...

Remove this Network ?

ModbusPlus_1

Note: If this removed network was previously attached to a communication module, this module loses its link and it will work with its default parameters.

Link between logical and physical network

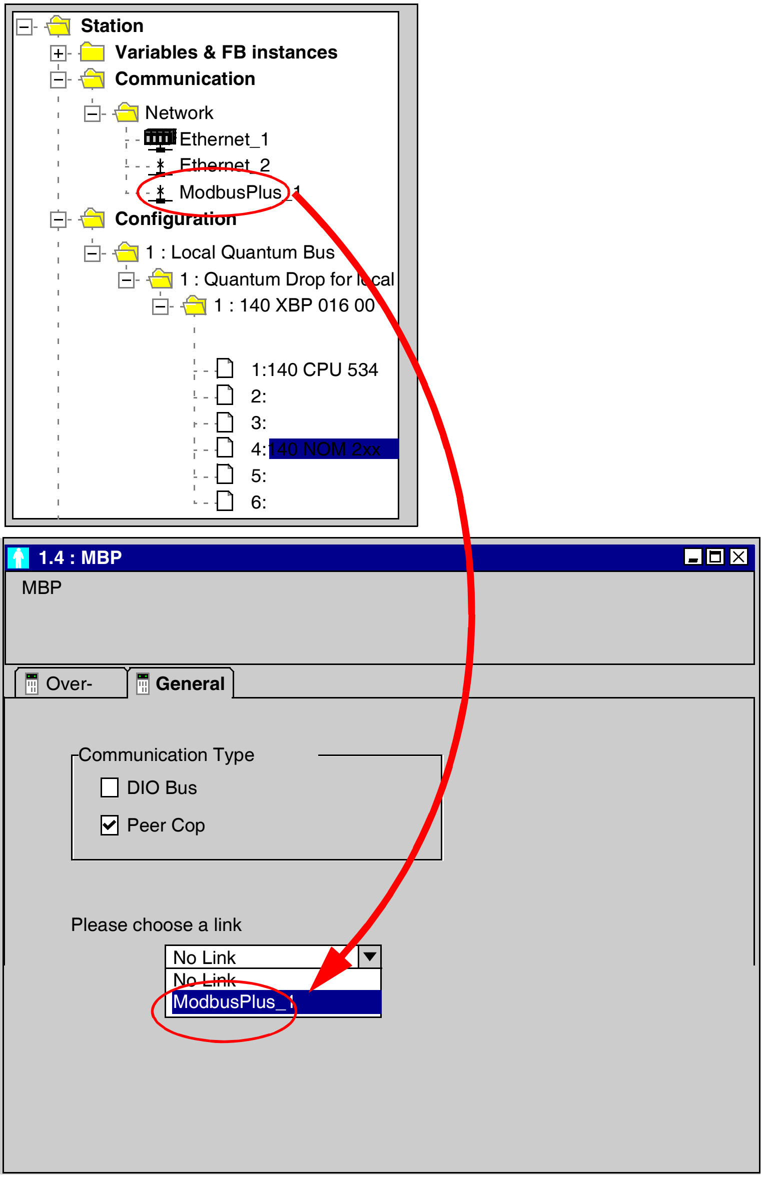
NetLinks

DuringUnity Pro application design, the NetLinks are created and inserted on sub-folder Communication under Network. These are the names of logical networks.

Under configuration folder, on the communication module node included in the current station, the list of existing NetLinks is proposed to select and attach one network to one module. Only the NetLink that can be managed by this module, are displayed in the list box on module configuration screen. No NetLink can be edited and created here (no edit box), but this list contains at least the No_Link field.

Attaching a NetLink to a Module

The following figure shows how to attach a created Modbus Plus NetLink to a NOM module .



Station
- Variables & FB instances
- Communication
  - Network
    - Ethernet_1
    - Ethernet_2
    - ModbusPlus_1
- Configuration
  - 1 : Local Quantum Bus
    - 1 : Quantum Drop for local
      - 1 : 140 XBP 016 00
        - 1:140 CPU 534
        - 2:
        - 3:
        - 4:140 NOM 2xx
        - 5:
        - 6:

1.4 : MBP

MBP

Over- | General

Communication Type

☐ DIO Bus

☑ Peer Cop

Please choose a link

No Link ▼

No Link

ModbusPlus_1

When you open the Modbus Plus configuration screen, it could take a long time (some time about 11 s). This is a normal behaviour, you just have to wait a little.

When a network is attached to a module, the icon of the corresponding node is changed and the network editor displays the address of the module in the rack .

The Icon in the Network folder indicates whether the link is attached to a module or not:

	Icon when no communication module is attached to the NetLink
	Icon when a communication module has been attached to the NetLink

Configuring a Physical Network

4

Overview

Introduction

This chapter describes how to configure a physical network.

What's in this Chapter?

This chapter contains the following topics:

Configuring a Quantum DIO Drop

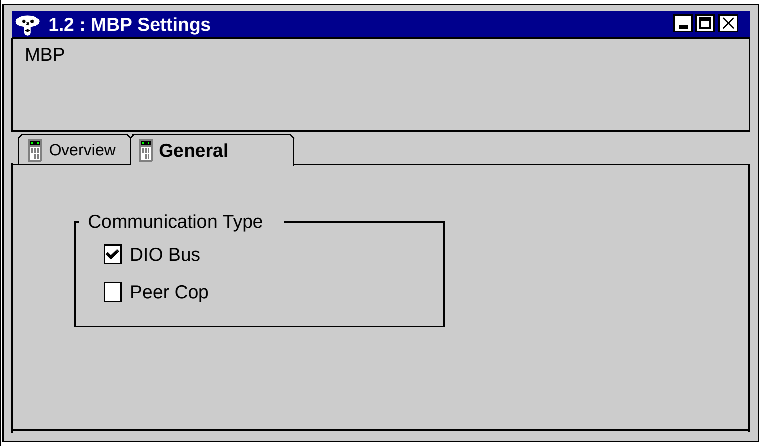
Introduction

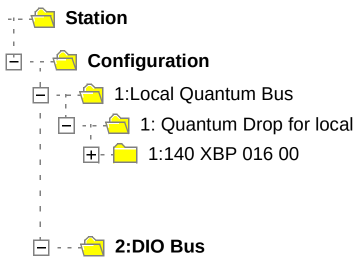
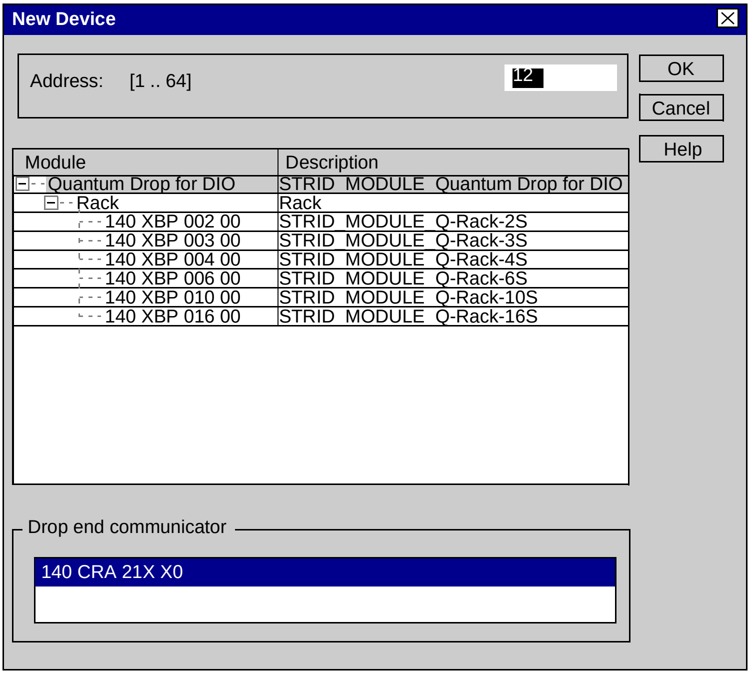
A Quantum DIO Drop consists of a standard module rack installed with I/O modules and a 140 CRA 21• •0 Modbus Plus communication module.
A DIO Bus can either be connected to the Modbus Plus connection on the CPU, or to a 140 NOM 2•• 00 communication module.

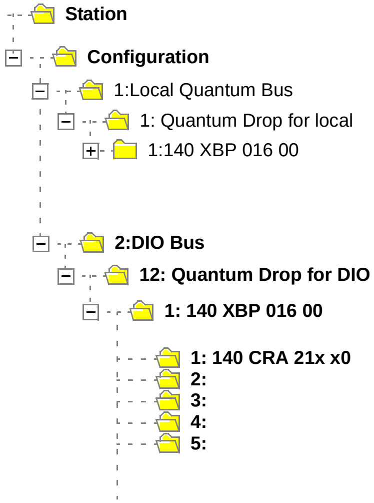
> **Note:** The Quantum DIO Drop Modules 140 CRA 2•• ••• have no health bits. For this reason the status for a properly functioning DIO Drop is always ZERO and not ONE as it is for other modules!

Adding a DIO Bus

The following table describes the procedure for adding a DIO Bus.

Step	Action
1	Select the Modbus Plus connection that you want to configure as a DIO Bus from the Project Browser or in the hardware configuration window. The following configuration window is opened:  Communication Type ☑ DIO Bus □ Peer Cop

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>Activate the <b>DIO Bus</b> check box in the configuration window and confirm your entry.
A DIO Bus is placed in the Project Browser:

<b>Station</b>
<b>Configuration</b>
1:Local Quantum Bus
1: Quantum Drop for local
1:140 XBP 016 00
<b>2:DIO Bus</b></td></tr>
<tr><td>3</td><td>Open the DIO Bus and select add module rack and select <b>New Device</b>.
The module rack selection list is opened:

<b>New Device</b>
Address: [1 .. 64] 12
OK
Cancel
Help
<table>
<tr><th>Module</th><th>Description</th></tr>
<tr><td>Quantum Drop for DIO</td><td>STRID_MODULE_Quantum Drop for DIO</td></tr>
<tr><td>Rack</td><td>Rack</td></tr>
<tr><td>140 XBP 002 00</td><td>STRID_MODULE_Q-Rack-2S</td></tr>
<tr><td>140 XBP 003 00</td><td>STRID_MODULE_Q-Rack-3S</td></tr>
<tr><td>140 XBP 004 00</td><td>STRID_MODULE_Q-Rack-4S</td></tr>
<tr><td>140 XBP 006 00</td><td>STRID_MODULE_Q-Rack-6S</td></tr>
<tr><td>140 XBP 010 00</td><td>STRID_MODULE_Q-Rack-10S</td></tr>
<tr><td>140 XBP 016 00</td><td>STRID_MODULE_Q-Rack-16S</td></tr>
</table>
Drop end communicator
140 CRA 21X X0</td></tr>
</table>

Step	Action
4	Select the desired module rack and enter the Modbus Plus address in the address field. Confirm with OK. A DIO Drop is placed in the Project Browser. The number set, 12 in our example, states the Modbus Plus address of the Drop. The Modbus Plus Coupler 140 CRA 21X X0 is automatically entered in slot 1:  **Station** **Configuration** 1:Local Quantum Bus 1: Quantum Drop for local 1:140 XBP 016 00 **2:DIO Bus** **12: Quantum Drop for DIO** **1: 140 XBP 016 00** **1: 140 CRA 21x x0** **2:** **3:** **4:** **5:**
5	To continue configuring the RIO Drop you can carry on as with configuring a local I/O.

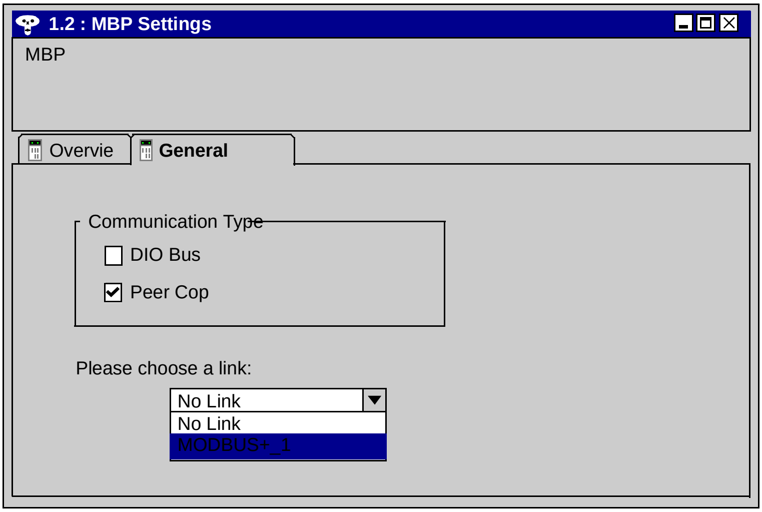
Modbus Plus Address

Ensure that the Modbus Plus Station Address that you have entered in the software configuration matches the hardware addresses of the modules used.

Peer Cop Configuration

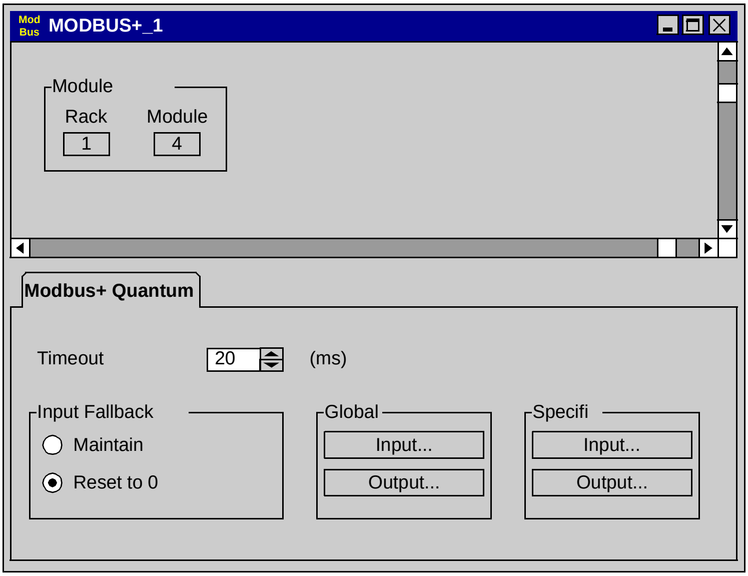
Requirements

Before configuring the **Peer Cop** communication type you must carry out the following steps:

Step	Action
1	Setup a logical Modbus Plus network in the Project Browser
2	Select the Modbus Plus connection that you want to configure as the Peer Cop from the Project Browser or in the hardware configuration window. The following configuration window is opened:  1.2 : MBP Settings MBP Overvie **General** Communication Type □ DIO Bus ☑ Peer Cop Please choose a link: No Link No Link MODBUS+_1
3	Activate the **Peer Cop** check box in the configuration window.
4	Connect the communication modules (CPU, NOM) Modbus Plus connection to the logical network (here MODBUS+_1)

Peer Cop Configuration

Carry out the following steps to configure the Peer Cop:

Step	Action
1	Open a Modbus Plus NetLink in the Project Browser by double-clicking on the symbol **Result:** The Peer Cop configuration window is opened with a predefined address corresponding to the position of the module  Mod Bus MODBUS+_1 Module Rack: 1 Module: 4 Modbus+ Quantum Timeout 20 (ms) Input Fallback ○ Maintain ● Reset to 0 Global: Input... Output... Specifi: Input... Output...
2	Make entries for **Timeout** and **Input Fallback Mode** (see Parameter description)
3	Carry out the global data configuration
4	Carry out the specific data configuration

Peer Cop parameter description

The table gives a description of the Peer Cop parameter

Parameter	Field / Button	Description
Module Address	Rack:1 Module: 2 (for example)	The topological address of the module rack and the slot number with the connected communications module are shown here
Timeout	● The default value is 500 ms. ● The values must be between 20 ms and 2 sec ● The increment is 20 ms	Refresh time for the inputs in milliseconds. It enables the maximum time to be set in which the remote drop inputs must be refreshed on the Modbus Port. If the data is not refreshed in the specified time an error is generated.
Input Fallback Mode	Maintain Reset to 0	The input values can be maintained or reset to "0"
Specific	Inputs Outputs	Buttons for specific data configuration (inputs and outputs)
Global	Inputs Outputs	Buttons for global data configuration (inputs and outputs)

Global Input Data Configuration

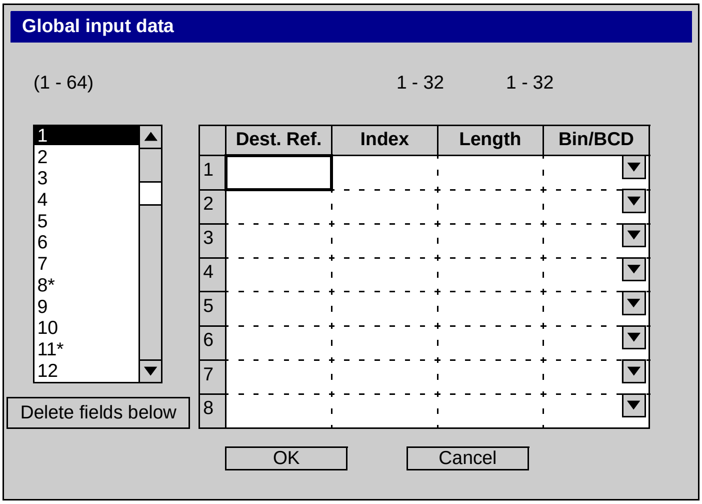
Overview

The Peer Cop configuration window contains the following buttons for global data configuration:

- **Global input data**
- Global output data

Global input data

The diagram shows the global input data configuration window



The table shows the global input data configuration parameter

Parameter	Field / Button	Description
Station window (1-64)	3	Station from which data is received
Dest. Ref.	%IW10 (for example)	Address for saving the data received
Length (max. 32)	6 (for example)	This means 6 words from station 3 are sent to all stations
Index	4 (for example)	This means that the station receives the 4th word from station 3
Bin/BCD	Bin. BCD	Received data codes

Global Output Data Configuration

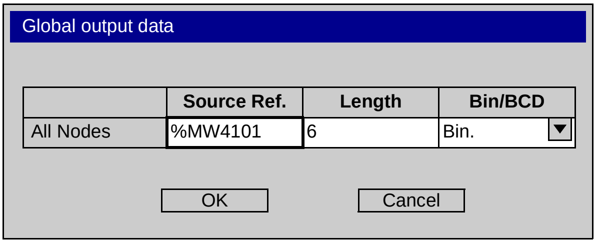
Overview

The Peer Cop configuration window contains the following buttons for global data configuration:

- Global input data
- **Global output data**

Global output data

The diagram shows the global output data configuration window



Global output data

	Source Ref.	Length	Bin/BCD
All Nodes	%MW4101	6	Bin.

The table shows the global output data configuration parameter

Parameter	Field / Button	Description
Source Ref.	%MW4101 (for example)	Address from which data is sent to all other stations
Length (max. 32)	6 (for example)	This means 6 words are sent to all stations
Bin/BCD	Bin. BCD	Received data codes

Configuring Specific Data

Overview

The Peer Cop configuration window contains the following buttons for specific data configuration:

- *Specific Input Data Configuration, p. 53*
- *Specific Output Data Configuration, p. 54*

Configuration

The specific input and output data is continuously placed as application internal words.

The user must define every local bus segment connection point as follows:

- Starting address in the table of internal words (%MW)
- Assignment of 0 to 32 words per station

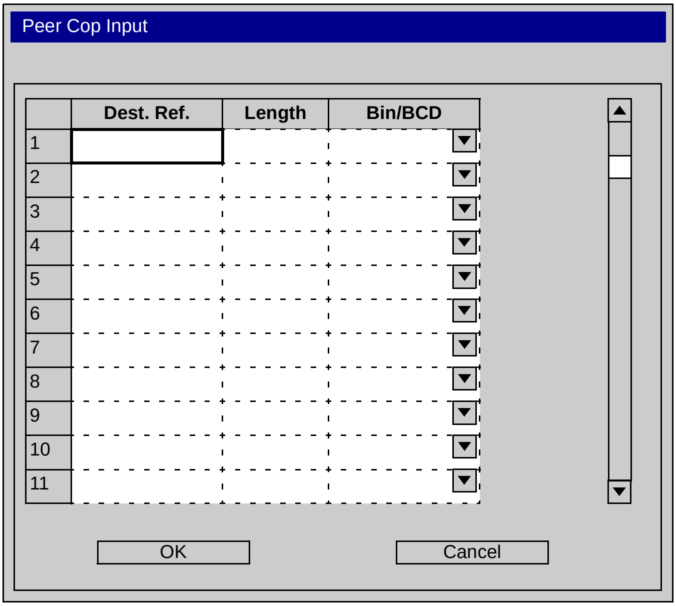
The following restrictions/rules must be observed:

- The address area for input and output words must not overlap.
- The maximum size of the specific data may not exceed 1,000 words (500 words maximum for the input words and maximum 500 words for the output words).

Specific Input Data Configuration

Specific input data

The diagram shows the specific input data configuration window



	Dest. Ref.	Length	Bin/BCD
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			

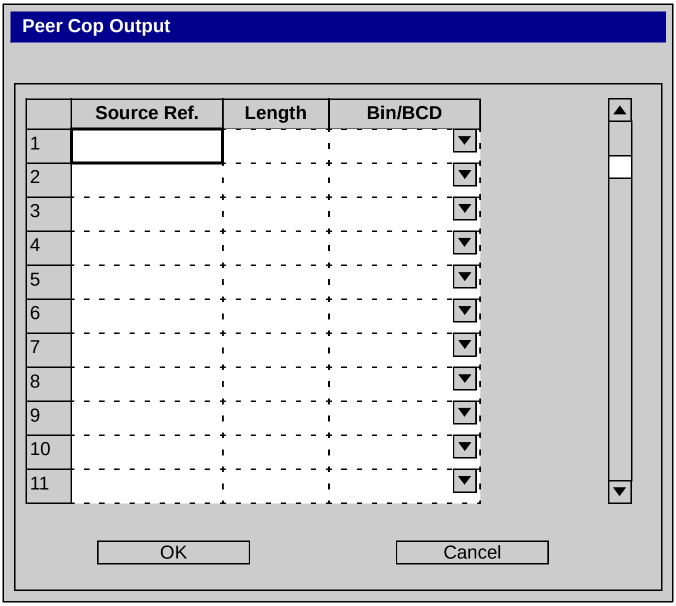
The table shows the global input data configuration parameter

Parameter	Field / Button	Description
Dest. Ref.	%IW10 (for example)	Address for saving the data received
Length (max. 32)	6 (for example)	This means 6 words from station 3 are sent to all stations
Bin/BCD	Bin. BCD	Received data codes

Specific Output Data Configuration

Specific output data

The diagram shows the specific output data configuration window



Peer Cop Output

	Source Ref.	Length	Bin/BCD
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			

The table shows the specific output data configuration parameter

Parameter	Field / Button	Description
Source Ref.	%MW4101 (for example)	Address from which data is sent to all other stations
Length (max. 32)	6 (for example)	This means 6 words are sent to all stations
Bin/BCD	Bin. BCD	Received data codes

Application Specific Communication

III

At a Glance

Overview

This part of the document contains information about Modbus Plus Application Specific Communication.

What's in this Part?

This part contains the following chapters:

Introducing Application Specific Communication

5

Overview

Introduction

The application specific communication function enables data exchange between Modbus Plus stations to be carried out under the control of the application program. Unity Pro-Soft provides a series of function blocks for this that are described in this section.

What's in this Chapter?

This chapter contains the following topics:

Overview of Function Blocks for Modbus Plus Communication

Introduction

Unity Pro provides the following 6 function blocks for communication with Modbus Plus:

- READ_REG
- WRITE_REG
- CREAD_REG
- CWRITE_REG
- MBP_MSTR
- ModbusP_ADDR

In contrast to Peer Cop and DIO communication, application specific communication enables connections between stations that are connected from different Modbus Plus networks via Bridges.

Note: Application specific communication requires no specific configuration or programming on the respective slave. When writing registers to an other station, special attention must be made that the correct destination area is addressed to avoid unintentionally overwriting data.

Note: For Quantum PLC communication with a Premium/Atrium PLC the addressing must be made with an offset of 1. In order to access an address object **n** of a Premium PLC, the communication function of the Quantum PLC must use the **n+1** address.

The following is a brief overview of the individual function blocks. A detailed representation is found in the next chapters.

READ_REG/ WRITE_REG

A rising edge at the REQ input reads or writes a register area to this function block once. It transfers data between the PLC and an addressed slave via Modbus Plus. The address and routing information is prepared by the **ModbusP_ADDR** block.

CREAD_REG/ CWRITE_REG

This function block continuously reads or writes a register area. It transfers data between the PLC and an addressed slave via Modbus Plus.
The address and routing information is prepared by the **ModbusP_ADDR** block.

MBP_MSTR

This Modbus Plus Master block is intended for universal application. It enables both data transfer between Modbus Plus stations including global data, as well as access to diagnostics and statistical data of the Modbus Plus network.

ModbusP_ ADDR

This block prepares the Modbus Plus address and routing information for the write and read blocks.

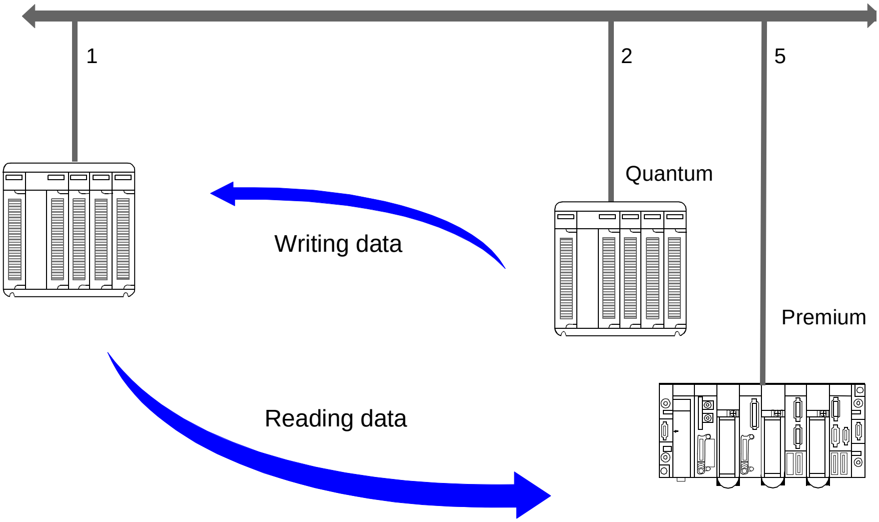
Data Exchange on a Local Segment

Overview

A Quantum controller can exchange data with all connected stations via a Modbus Plus network.
Routing information is set to 0 in the local segment.

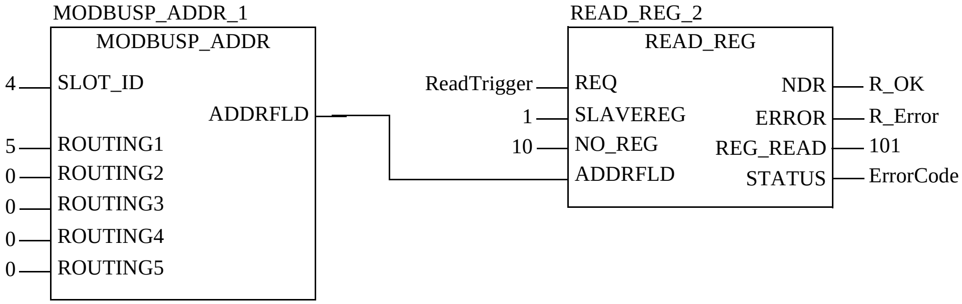
Example for a local segment

In the following configuration, a Quantum is connected with a Modbus Plus network in slot 4 via a NOM module. Data is read from a Quantum (address 2) and data is sent to a Premium (address 5).



Writing data

In the following example the 10 register %MW1 to %MW10 is read from a Quantum and placed as %MW101 to %MW110.

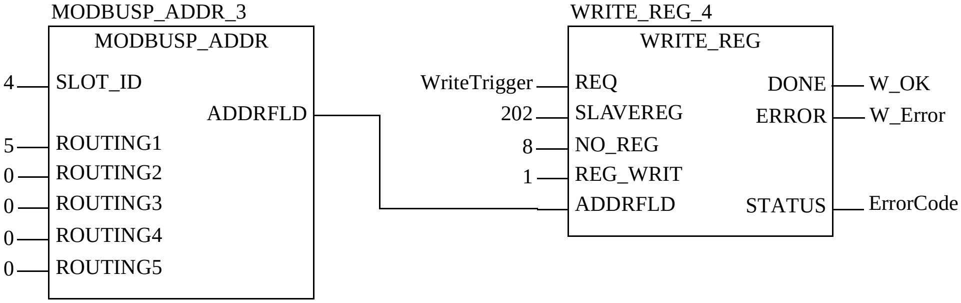


The following table describes the parameters of both blocks:

Parameter	Content/ Variable	Description
Slot_ID	4	NOM Modbus Plus communication module slot (0 for the CPU's Modbus Plus Port)
ROUTING1	2	Modbus Plus address of the destination station
ROUTING2	0	Routing byte 2, 0 local segment
ROUTING3	0	Routing byte 3, 0 local segment
ROUTING4	0	Routing byte 4, 0 local segment
ROUTING5	0	Routing byte 5, 0 local segment
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
REQ	ReadTrigger	Trigger signal to start the read process
SLAVEREG	201	Offset address of the first register in the slave to be read from.
NO_REG	10	Number of registers to read
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
NDR	R_OK	Set to "1" for one cycle after reading new data
ERROR	R_Error	Set to "1" for one cycle if an error occurs
STATUS	ErrorCode	Error code
REG_READ	1	Starting address of the destination data field

Reading data

In the following example, the 8 register %MW1 to %MW8 is read from a Premium and placed as %MW201 to %MW208.



The following table describes the parameters of both blocks:

Parameter	Content/ Variable	Description
Slot_ID	4	NOM Modbus Plus communication module slot (0 for the CPU's Modbus Plus Port)
ROUTING1	5	Modbus Plus address of the destination station
ROUTING2	0	Routing byte 2, 0 local segment
ROUTING3	0	Routing byte 3, 0 local segment
ROUTING4	0	Routing byte 4, 0 local segment
ROUTING5	0	Routing byte 5, 0 local segment
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
REQ	WriteTrigger	Trigger signal to start the write process
SLAVEREG	202	Offset address of the first register in the slave to be written. (see Information)
NO_REG	8	Number of registers to write
REG_WRIT	1	Start address of the source data field
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
DONE	W_OK	Set to "1" for one cycle after writing data
ERROR	W_Error	Set to "1" for one cycle if an error occurs
STATUS	ErrorCode	Error code

Note: For Quantum PLC communication with a Premium/Atrium PLC the addressing must be made with an offset of 1. In order to access an address object **n** of a Premium PLC, the communication function of the Quantum PLC must use the **n+1** address.

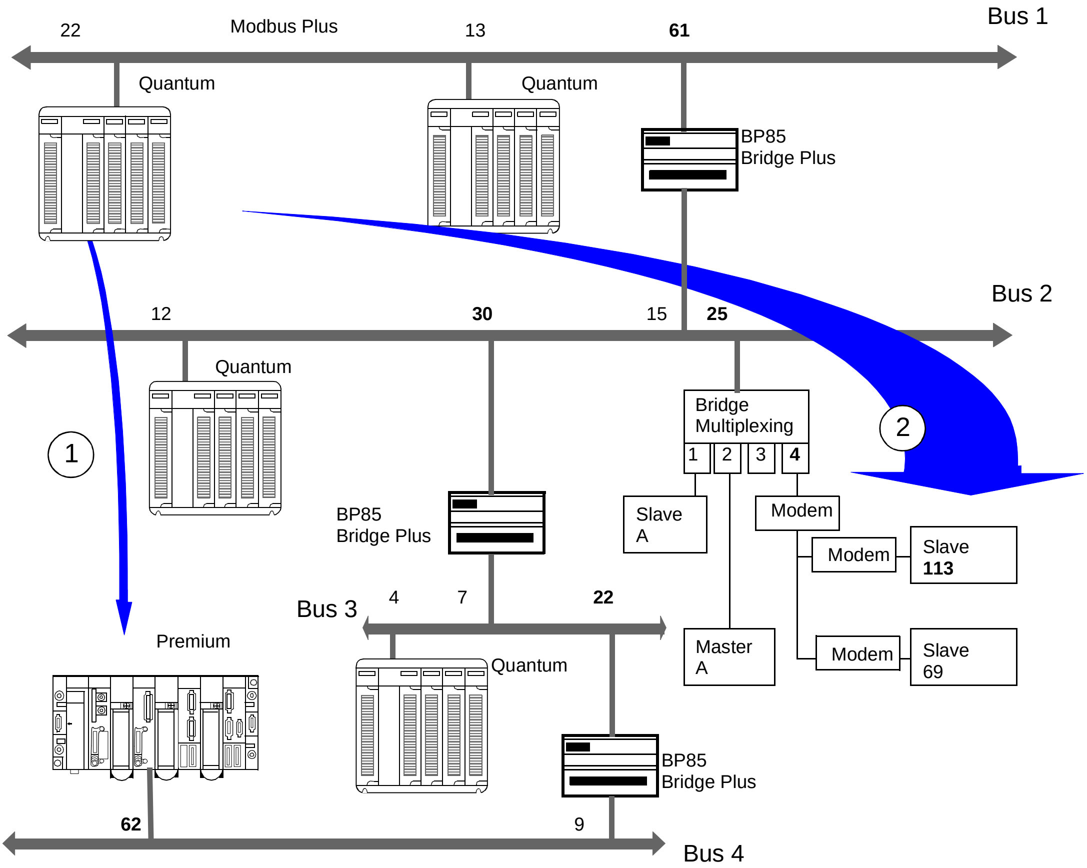
Data Exchange on Remote Modbus Plus Networks

Overview

A Quantum controller can exchange data with all connected stations via a Modbus Plus network.
In remote networks, the routing information must be entered additionally for every Bridge in the data path.

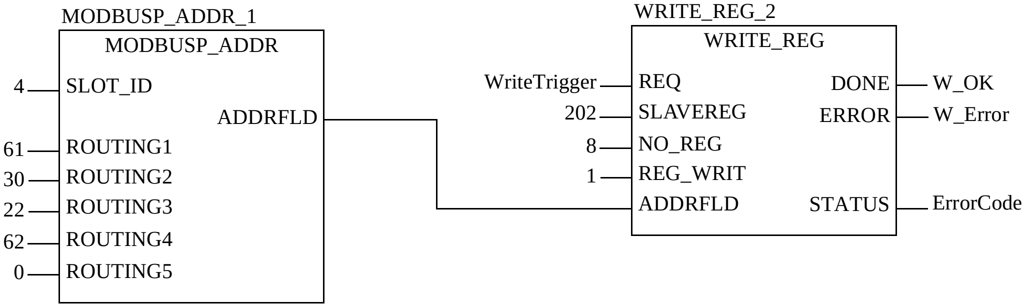
Example Segment spanning

In the following configuration, a Quantum is connected with a Modbus Plus network in slot 4 via a NOM module. Data is sent to a Premium (Bus 4, address 62) and a Modbus Slave.



Reading data (1)

In the following example, the 8 register %MW1 to %MW8 is read from a Premium and placed as %MW201 to %MW208.



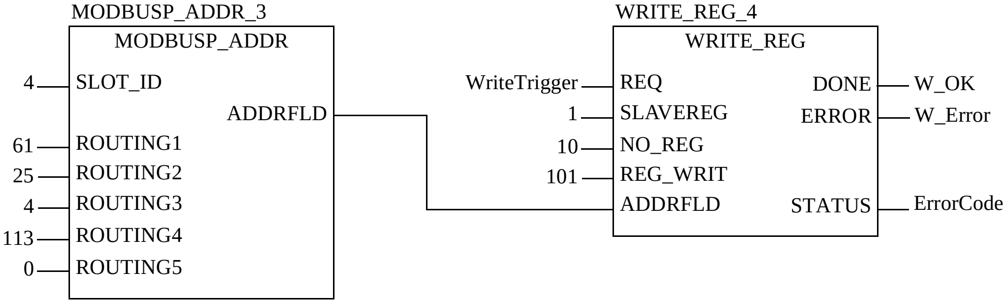
The following table describes the parameters of both blocks:

Parameter	Content/ Variable	Description
Slot_ID	4	Modbus Plus communication module slot
ROUTING1	61	Routing byte 1
ROUTING2	30	Routing byte 2
ROUTING3	22	Routing byte 3
ROUTING4	62	Routing byte 4
ROUTING5	0	Routing byte 5
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
REQ	WriteTrigger	Trigger signal to start the write process
SLAVEREG	202	Offset address of the first register in the slave to be written.
NO_REG	8	Number of registers to write
REG_WRIT	1	Start address of the source data field
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
DONE	W_OK	Set to "1" for one cycle after writing data
ERROR	W_Error	Set to "1" for one cycle if an error occurs
STATUS	ErrorCode	Error code

Note: For Quantum PLC communication with a Premium/Atrium PLC the addressing must be made with an offset of 1. In order to access an address object **n** of a Premium PLC, the communication function of the Quantum PLC must use the **n+1** address.

Reading data (2)

In the following example, the 10 registers %MW101 to %MW110 are sent to a Modbus Slave and placed as %MW1 to %MW10.



The following table describes the parameters of both blocks:

Parameters	**Content/ Variable**	**Description**
Slot_ID	4	NOM Modbus Plus communication module slot (0 for the CPU's Modbus Plus Port)
ROUTING1	61	Routing byte 1
ROUTING2	25	Routing byte 2
ROUTING3	4	Routing byte 3
ROUTING4	113	Routing byte 4
ROUTING5	0	Routing byte 5
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
REQ	WriteTrigger	Trigger signal to start the write process
SLAVEREG	1	Offset address of the first register in the slave to be written.
NO_REG	10	Number of registers to write
REG_WRIT	101	Start address of the source data field
ADDRFLD	`WordArr5`	Data structure with the Modbus Plus address and routing information
DONE	W_OK	Set to "1" for one cycle after writing data
ERROR	W_Error	Set to "1" for one cycle if an error occurs
STATUS	ErrorCode	Error code

Global Data - Broadcast Services

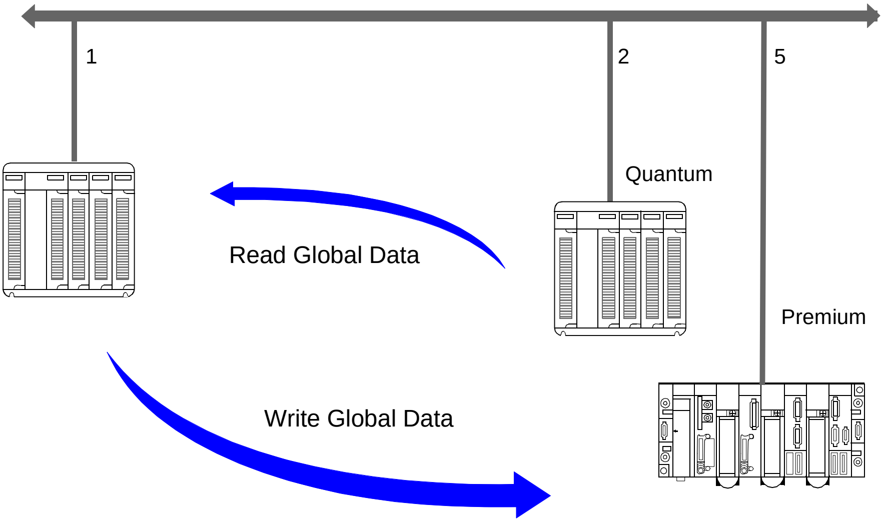
Overview

A Quantum controller can exchange global data with all connected stations via a Modbus Plus network.

Global data is a Broadcast service that enables up to 16 registers to be sent to all connected stations with the transfer of Tokens. The sending and receiving of global data with a Quantum controller can be configured through the Peer Cop framework, and can also be activated in the application program with the help of the **MSTR** block.

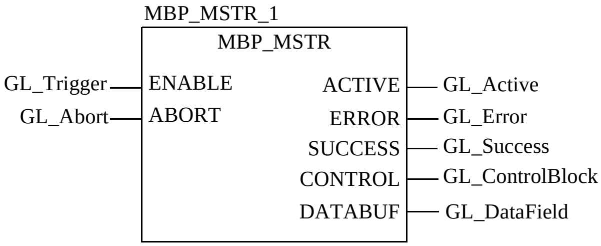
Example for Global Data

In the following configuration, a Quantum is connected with a Modbus Plus network in slot 4 via a NOM module. Global data is exchanged with the other stations in the Modbus Plus network.



Read global data

In the following example, the 10 register %MW1 to %MW10 is read from the Quantum with station address 2 as global data and placed in the GL_DataField.



The following table describes the parameters of the MSTR blocks:

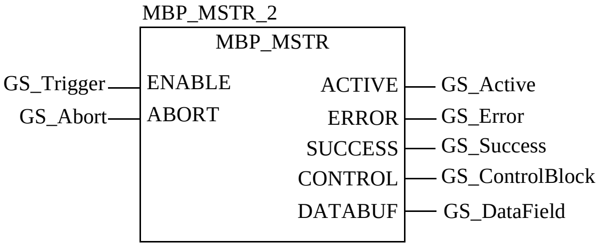
Parameters	Content/Variable	Description
ENABLE	GL_Trigger	Enable **MSTR** function
ABORT	GL_Abort	Enable active **MSTR** function
ACTIVE	GL_Active	Operation is active
ERROR	GL_Error	Operation is faulty
SUCESS	GL_Success	Operation completed successfully
CONTROL	`ARRAY [0..8] OF WORD`/ GL_ControlBlock	Field for **MSTR** control block
DATABUF	`ARRAY [0..n] OF WORD` (n ≥ 10)/ GL_DataField	Data field for the recieved data

Contents of GL_ControlBlock:

Register	Contents	Description
GL_ControlBlock[0]	6	Read global data
GL_ControlBlock[1]	-	indicates the error status
GL_ControlBlock[2]	10	Number of registers that should be read as global data
GL_ControlBlock[3]	-	Display of registers available in scanned node (will be automatically updated).
GL_ControlBlock[4]	x0402	Least significant byte: Address of the station whose global data is to be read Most significant byte: Communication module slot (0 for the CPU)

Write global data

In the following example, the 8 registers %MW101 to %MW108 are sent from the Quantum with station address 1 as global data to all nodes in the Modbus Plus network.



The following table describes the parameters of the `MSTR` block:

Parameters	Content/Variable	Description
ENABLE	GS_Trigger	Enable **MSTR** function
ABORT	GS_Abort	Enable active **MSTR** function
ACTIVE	GS_Active	Operation is active
ERROR	GS_Error	Operation is faulty
SUCESS	GS_Success	Operation completed successfully
CONTROL	`ARRAY [0..8] OF WORD` / GS_ControlBlock	Field for **MSTR** control block
DATABUF	`ARRAY [0..n] OF WORD` (n ≥ 10)/ GS_DataField	Data field of the data to be sent

Contents of the `GS_ControlBlock`:

Register	Contents	Description
GS_ControlBlock[0]	5	Write global data
GS_ControlBlock[1]	-	indicates the error status
GS_ControlBlock[2]	10	Number of registers to be sent from the State RAM as global data (1...32)
GS_ControlBlock[3]	-	Reserved
GS_ControlBlock[4]	x0400	Most significant byte: Communication module slot (0 for the CPU)

CREAD_REG: Continuous register reading

6

Overview

Introduction

This chapter describes the CREAD_REG block.

What's in this Chapter?

This chapter contains the following topics:

Description

Function description

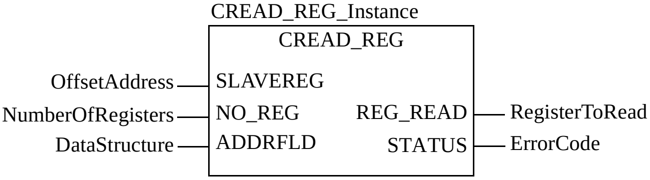
This function block reads a register area continuously. It reads data from an addressed node via Modbus Plus, TCP/IP Ethernet or SY/MAX Ethernet.
EN and ENO can be configured as additional parameters.

Note: When programming a CREAD_REG function, you must be familiar with the routing procedures used by your network. Modbus Plus routing path structures are described in detail in the *Modbus Plus Network Planning and Installation Guide*. If TCP/IP or SY/MAX Ethernet routing is implemented, standard Ethernet IP router products must be used. A full description of the TCP/IP routing is provided in the *Quantum with Unity ProTCP/IP Configuration User Guide*.

Note: Several copies of this function block can be used in the program. However, multiple instancing of these copies is not possible.

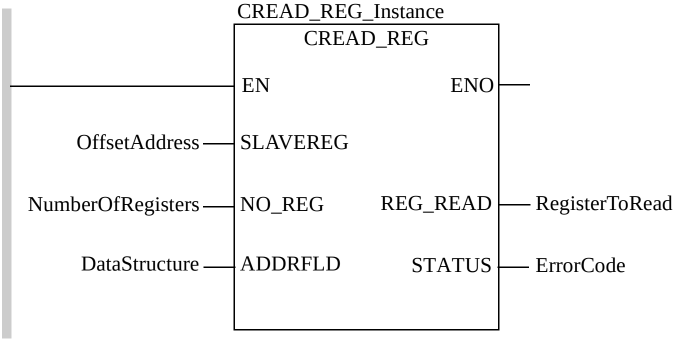
Representation in FBD

Representation:



Representation in LD

Representation:



Representation in IL

Representation:

```
CAL CREAD_REG_Instance (SLAVEREG:=OffsetAddress,
    NO_REG:=NumberOfRegisters, ADDRFLD:=DataStructure,
    REG_READ=>RegisterToRead, STATUS=>ErrorCode)
```

Representation in ST

Representation:

```
CREAD_REG_Instance (SLAVEREG:=OffsetAddress,
    NO_REG:=NumberOfRegisters, ADDRFLD:=DataStructure,
    REG_READ=>RegisterToRead, STATUS=>ErrorCode) ;
```

Parameter description

Description of input parameters:

Parameter	Data type	Description
SLAVEREG	DINT	Offset address of the first %MW register in the slave to be read from.
NO_REG	INT	Number of addresses to be read from slave.
ADDRFLD	WordArr 5	Data structure describing the Modbus Plus address, TCI/IP address or SY/MAX IP address.

Description of output parameters:

Parameter	Data type	Description
REG_READ	ANY	Data to be read For the file to be read a data structure must be declared as a located variable.
STATUS	WORD	If an error occurs while the function is being executed, an error code remains at this output for one cycle. Error code, see ● *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115* ● *SY/MAX-specific error codes, p. 117* ● *TCP/IP Ethernet error codes, p. 119*

Runtime error

For a list of all block error codes and values, see .

Derived Data Types

Derived data typeWordArr5 in Modbus Plus

Element descriptions:

Element	Data type	Description
WordArr5[1]	WORD	Least significant byte: Routing register 1 is used for address specification (routing path addresses one of five) of the destination node during network transfer. The last byte in the routing path that is not zero is the destination node. Most significant byte: Slot of the network adapter module (NOM), if any (only Quantum).
WordArr5[2]	WORD	Routing register 2
WordArr5[3]	WORD	Routing register 3
WordArr5[4]	WORD	Routing register 4
WordArr5[5]	WORD	Routing register 5

Element description for WordArr5 with TCP/IP Ethernet

Element description for WordArr5 with TCP/IP Ethernet

Element	Data type	Description
WordArr5[1]	WORD	Least significant byte: MBP on Ethernet Transporter (MET) mapping index Most significant byte: Slot of the NOE module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Element description for WordArr5 with SY/MAX Ethernet

Element description for WordArr5 with SY/MAX Ethernet

Element	Data type	Description
WordArr5[1]	WORD	Least significant byte: MBP on Ethernet Transporter (MET) mapping index Most significant byte: Slot of the NOE module
WordArr5[2]	WORD	Destination drop number (or set to FF hex)
WordArr5[3]	WORD	Terminator (set to FF hex)
WordArr5[4]	WORD	No significance
WordArr5[5]	WORD	No significance

Function mode

Function mode of the CREAD_REG block

Although a large number of CREAD_REG function blocks can be programmed; only four read operations may be active at the same time. In this case it is irrelevant whether they are the result of this function block or others (e.g. MBP_MSTR, MSTR, READ_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC (NOE 211 00) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only **one** read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

All routing information is contained in the WordArr5 data structure of the ADDRFLD input. The type function block assigned to this input is defined by the network used. Please use:

- Modbus Plus for function block ModbusP_ADDR
- TCP/IP Ethernet for function block TCP_IP_ADDR
- SY/MAX Ethernet for function block SYMAX_IP_ADDR

Note: The WordArr5 data structure can also be used with constants.

Note: This function block puts a heavy load on the network; therefore the network load must be carefully monitored. If the network load is too high, the program logic should be reorganized in order to work with the READ_REG function block, a variation of this function block that does not operate in a continuous mode, but under command control.

Parameter description

SLAVEREG

Start of the area in the addressed slave from which the source data is read. The source area always resides within the %MW register area.

> **Note:** For slaves for a **non-**Unity Pro PLC:
> The source area always resides within the 4x register area. `SLAVEREG` expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).

The parameter can be entered as an address, located variable, unlocated variable or literal.

NO_REG

Number of addresses to be read from the addressed slave (1 ... 100).
The parameter can be entered as an address, located variable or unlocated variable.

REG_READ

An `ARRAY` that is the same size as the requested transmission must be agreed upon (≥ `NO_REG`) for this parameter. The name of this array is defined as a parameter. If the array is defined too small, then only the amount of data is transmitted that is present in the array.
The parameter must be defined as a located variable.

STATUS

If an error occurs while the function is being executed, an error code remains at this output for one cycle.
Error code, see

- *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115*
- *SY/MAX-specific error codes, p. 117*
- *TCP/IP Ethernet error codes, p. 119*

The parameter can be entered as an address, located variable or unlocated variable.

CWRITE_REG: Continuous register writing

7

Overview

Introduction

This chapter describes the CWRITE_REG block.

What's in this Chapter?

This chapter contains the following topics:

Description

Function description

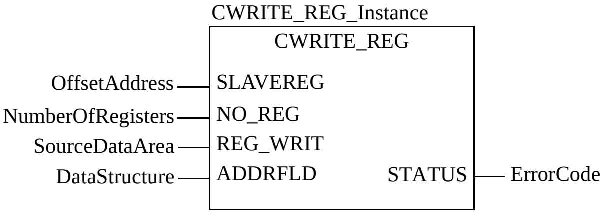
This function block writes continuously to the register area. It transfers data from the PLC via Modbus Plus, TCP/IP Ethernet or SY/MAX Ethernet to an addressed slave. EN and ENO can be configured as additional parameters.

Note: You must be familiar with the routing procedures of the network when programming a CWRITE_REG function. Modbus Plus routing path structures are described in detail in the *Modbus Plus Network Planning and Installation Guide*. If TCP/IP or SY/MAX Ethernet routing is implemented, standard Ethernet IP router products must be used. A full description of the TCP/IP routing is provided in the *Quantum with Unity ProTCP/IP Configuration User Guide*.

Note: Several copies of this function block can be used in the program. However, multiple instancing of these copies is not possible.

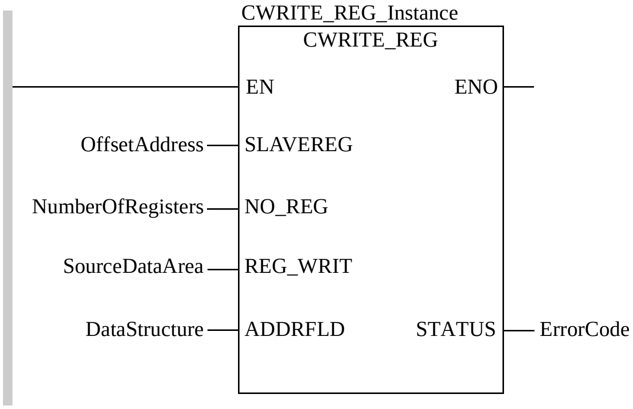
Representation in FBD

Representation:



Representation in LD

Representation:



Representation in IL

Representation:

```
CAL CWRITE_REG_Instance (SLAVEREG:=OffsetAddress,
    NO_REG:=NumberOfRegisters, REG_WRIT:=SourceDataArea,
    ADDRFLD:=DataStructure, STATUS=>ErrorCode)
```

Representation in ST

Representation:

```
CWRITE_REG_Instance ( SLAVEREG:=OffsetAddress,
    NO_REG:=NumberOfRegisters, REG_WRIT:=SourceDataArea,
    ADDRFLD:=DataStructure, STATUS=>ErrorCode) ;
```

Parameter description

Description of input parameters:

Parameter	Data type	Description
SLAVEREG	DINT	Offset address of the first %MW register in the slave to be written to.
NO_REG	INT	Number of addresses to be written from slave
REG_WRIT	ANY	Source data (A data structure must be declared as a located variable for the source file .)
ADDRFLD	WordArr5	Data structure transferring the Modbus Plus-address, TCI/IP address or SY/MAX-IP address.

Description of output parameters:

Parameter	Data type	Description
STATUS	WORD	If an error occurs while the function is being executed, an error code remains at this output for one cycle. Error code, see ● *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115* ● *SY/MAX-specific error codes, p. 117* ● *TCP/IP Ethernet error codes, p. 119*

Runtime error

For a list of all block error codes and values, see .

Derived Data Types

Element description for `WordArr5` in Modbus Plus

Element description for `WordArr5` in Modbus Plus:

Element	Data type	Description
`WordArr5[1]`	`WORD`	Least significant byte: Routing register 1 is used for address specification (routing path addresses one of five) of the destination node during network transfer. The last byte in the routing path that is not zero is the destination node. Most significant byte: Slot of the network adapter module (NOM), if any.
`WordArr5[2]`	`WORD`	Routing register 2
`WordArr5[3]`	`WORD`	Routing register 3
`WordArr5[4]`	`WORD`	Routing register 4
`WordArr5[5]`	`WORD`	Routing register 5

Element description for `WordArr5` with TCP/IP Ethernet

Element description for `WordArr5` with TCP/IP Ethernet

Element	Data type	Description
`WordArr5[1]`	`WORD`	Least significant byte: MBP on Ethernet Transporter (MET) mapping index Most significant byte: Slots of the NOE module
`WordArr5[2]`	`WORD`	Byte 4 (MSB) of the 32-bit destination IP address
`WordArr5[3]`	`WORD`	Byte 3 of the 32-bit destination IP address
`WordArr5[4]`	`WORD`	Byte 2 of the 32-bit destination IP address
`WordArr5[5]`	`WORD`	Byte 1 (LSB) of the 32-bit destination IP address

Element description for `WordArr5` with SY/MAX Ethernet

Element description for `WordArr5` with SY/MAX Ethernet

Element	Data type	Description
`WordArr5[1]`	`WORD`	Least significant byte: MBP on Ethernet Transporter (MET) mapping index Most significant byte: Slot of the NOE module
`WordArr5[2]`	`WORD`	Destination drop number (or set to FF hex)
`WordArr5[3]`	`WORD`	Terminator (set to FF hex)
`WordArr5[4]`	`WORD`	No significance
`WordArr5[5]`	`WORD`	No significance

Function mode

Function mode of the CWRITE_REG block

Although a large number of CWRITE_REG function blocks can be programmed, only four write operations may be active at the same time. In this case it is irrelevant whether they are the result of this function block or others (e.g. MBP_MSTR, MSTR, WRITE_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.
If several CWRITE_REG function blocks are used within an application, they must at least differ in the values of their NO_REG or REG_WRIT parameters.

Note: A TCP/IP communication between a Quantum PLC (NOE 211 00) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only **one** read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

All routing information is contained in the WordArr5 data structure of the ADDRFLD input. The type function block assigned to this input is defined by the network used.
Please use:

- Modbus Plus for function block ModbusP_ADDR
- TCP/IP Ethernet for function block TCP_IP_ADDR
- SY/MAX Ethernet for function block SYMAX_IP_ADDR

Note: The WordArr5 data structure can also be used with constants.

Note: This function block puts a heavy load on the network; therefore the network load must be carefully monitored. If the network load is too high, the program logic should be reorganized, in order to work with the WRITE_REG function block, a variation of this function block that does not operate in a continuous mode, but under command control.

Parameter description

SLAVEREG

Start of the area in the addressed slave to which the source data is written. The source area always resides within the %MW address area.

> **Note:** For slaves for a **non-**Unity Pro PLC:
> The destination area always resides within the 4x register area. SLAVEREG expects the target address as an offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
> The parameter can be entered as an address, located variable, unlocated variable or literal.

NO_REG

Number of registers to be written to slave processor (1 ... 100). The parameter can be entered as an address, located variable, unlocated variable or literal.

STATUS

If an error occurs while the function is being executed, an error code remains at this output for one cycle.
Error code, see

- *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115*
- *SY/MAX-specific error codes, p. 117*
- *TCP/IP Ethernet error codes, p. 119*

The parameter can be entered as an address, located variable or unlocated variable.

REG_WRIT

An ARRAY that is the same size as the requested transmission must be agreed upon (≥ NO_REG) for this parameter. The name of this array is defined as a parameter. If the array is defined too small, then only the amount of data is transmitted that is present in the array.
The parameter must be defined as a located variable.

MBP_MSTR: Modbus Plus Master

8

Overview

Introduction

This chapter describes the `MBP_MSTR` block.

What's in this Chapter?

This chapter contains the following topics:

Description

Function description

It is possible to select one of 12 available network communication operations using this function block.
Although a large number of MBP_MSTR function blocks can be programmed, only four of them can be active at the same time. All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC and a Momentum PLC is only possible when only **one**read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

Note: The function block can only be used on the program level, i.e. not in Derived Function Blocks (DFBs).

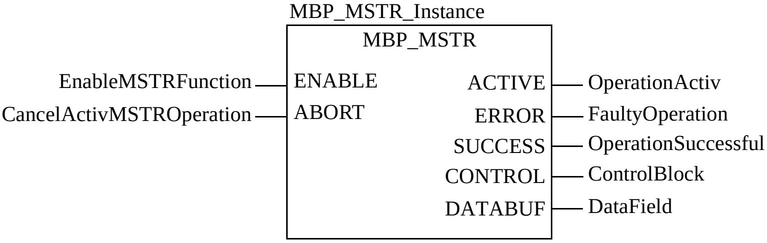
Note: Several copies of this function block can be used in the program. However, multiple instancing of these copies is not possible.

EN and ENO can be configured as additional parameters.

Note: You must be familiar with the routing procedures of your network when programming an MSTR function. Modbus Plus routing path structures are described in detail in the *Modbus Plus Network Planning and Installation Guide*. If TCP/IP or SY/MAX Ethernet routing is implemented, standard Ethernet IP router products must be used. A full description of the TCP/IP routing is provided in the *Quantum with Unity Pro TCP/IP Configuration User Guide*.

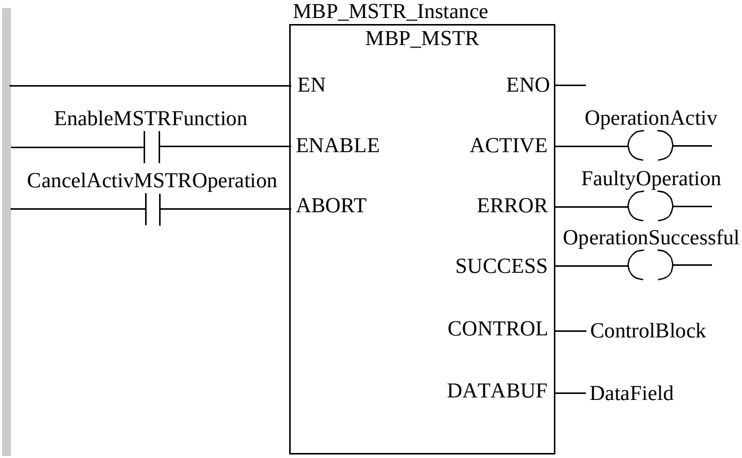
Representation in FBD

Representation:



Representation in LD

Representation:



Representation in IL

Representation:

```
CAL MBP_MSTR_Instance (ENABLE:=EnableMSTRFunction,
    ABORT:=CancelActivMSTROperation, ACTIVE=>OperationActiv,
    ERROR=>FaultyOperation, SUCCESS=>OperationSuccessful,
    CONTROL=>ControlBlock, DATABUF=>DataField)
```

Representation in ST

Representation:

```
MBP_MSTR_Instance (ENABLE:=EnableMSTRFunction,
    ABORT:=CancelActivMSTROperation, ACTIVE=>OperationActiv,
    ERROR=>FaultyOperation, SUCCESS=>OperationSuccessful,
    CONTROL=>ControlBlock, DATABUF=>DataField) ;
```

Parameter description

Description of input parameters:

Parameter	Data type	Description
ENABLE	BOOL	Enable MSTR function
ABORT	BOOL	Cancel active MSTR operation

Description of output parameters:

Parameter	Data type	Description
ACTIVE	BOOL	Operation is active
ERROR	BOOL	Faulty operation
SUCCESS	BOOL	Operation completed successfully
CONTROL	ANY	Field for MSTR control block (For the control block a data structure must be declared as a located variable.)
DATABUF	ANY	Data area (For the data area a data structure must be declared as a located variable.)

Runtime error

In the event of an error occurring during an MSTR operation, a hexadecimal error code is displayed in the CONTROL[2] register of the control block for one cycle. Function error codes are network-specific:

- Modbus Plus and SY/MAX Ethernet error codes (See *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115*)
- SY/MAX specific error codes (See *SY/MAX-specific error codes, p. 117*)
- TCP/IP Ethernet error codes (See *TCP/IP Ethernet error codes, p. 119*)
- CTE Error Codes for SY/MAX and TCP/IP Ethernet (See *CTE error codes for SY/MAX and TCP/IP Ethernet, p. 122*)

Note: For a list of all block error codes and values, see .

Function mode

Function mode of `MBP_MSTR` blocks

Using the `MBP_MSTR` block, one of 12 available network communication operations can be triggered via the network. Each operation receives a code. Whether the operations are available depends on the type of network used.

Valid function codes

Valid function codes:

Code	**Function**	**Modbus Plus**	**TCP/IP Ethernet**	**SY/MAX Ethernet**
1	Write data	X	X	X
2	Read data	X	X	X
3	Get local statistics	X	X	-
4	Clear local statistics	X	X	-
5	Write global data, Peer Cop	X	-	-
6	Read global data, Peer Cop	X	-	-
7	Get remote statistics	X	X	-
8	Clear remote statistics (See *Clear remote statistics, p. 100*)	X	X	-
9	Peer Cop Status (Peer Cop Health)	X	-	-
10	Reset optional module	-	X	X
11	Read CTE (Config extension)	-	X	X
12	Write CTE (Config extension)	-	X	X

Legend:

X	Yes
-	No

Parameter description

ENABLE

When ON, the operation specified in the first element of the CONTROL register is enabled.

ABORT

When ON, the currently active operation is aborted.

ACTIVE

ON, if the operation is active.

ERROR

ON, if the operation was aborted without success.

SUCCESS

ON, if the operation concluded successfully.

DATABUF

For operations providing data, e.g. the write operation, the data field is the data source. For operations receiving data, e.g. the read operation, the data field is the data destination.
With Ethernet CTE Read and Write operations, the data field holds the contents of the Ethernet configuration extension table. DATABUF must be defined as an ARRAY having at least 10 elements in this case.
The data field must be declared as a located variable.

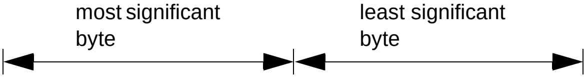
CONTROL

This field contains the control block. The first element CONTROL[1] contains a number from 1 to 12, which provides the operation code of the Modbus operation to be performed. The content of the sequence register is determined by the operation.
The data field must be declared as a located variable.
The structure of the control block differs according to the network used:

- Modbus Plus
- TCP/IP Ethernet
- SY/MAX Ethernet

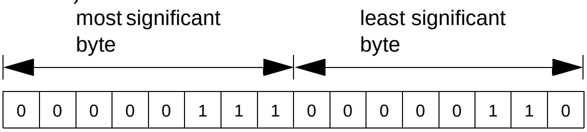
Control block for Modbus Plus

Control block for Modbus Plus:

Register	Contents
CONTROL[1]	indicates one of the Operations which are valid for Modbus Plus
CONTROL[2]	indicates the Error status.
CONTROL[3]	indicates the length (number of data transferred)
CONTROL[4]	indicates MSTR operation-dependent information
CONTROL[5]	Routing Register 1 Used for specification (routing path addresses one of five) of a destination node during network transfer. Most significant byte: Source node address This is the slot for the Modbus Plus Network Options Module (NOM). When using the Modbus Plus Port on the CPU, this byte must be set to "0" (independently of the CPU slot). Least significant byte: Destination node address This value represents a direct or a bridge address. If there is no bridge, this value contains the destination node address. If there is a bridge, this value contains the address of the bridge. If the NOM is inserted in slot 7 on the module rack, the most significant byte of routing register 1 looks as follows (value 0x0706):  0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 **Most significant byte** Slots 1 ... 16 **Least significant byte** Destination address (binary value between 1 and 64 (normal) or 65 to 255 (extended))
CONTROL[6]	Routing Register 2 This value contains the destination node address (further Bridge or Modbus Plus Modules). If addressing in the previous Routing Register has finished, the value is set to "0".
CONTROL[7]	Routing Register 3 see Routing Register 2
CONTROL[8]	Routing Register 4 see Routing Register 2
CONTROL[9]	Routing Register 5 see Routing Register 2

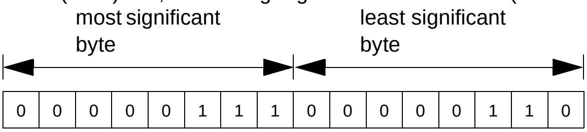
Control block for TCP/IP Ethernet

Control block for TCP/IP Ethernet:

Register	Contents
CONTROL[1]	indicates one of the Operations which are valid for TCP/IP.
CONTROL[2]	indicates the Error status (See `STATUS`, p. 74).
CONTROL[3]	indicates the length (number of data transferred)
CONTROL[4]	indicates `MSTR` operation-dependent information
CONTROL[5]	Routing Register Used for specification of a destination node during network transfer. Most significant byte: Source node address This is the NOE slot for the NOE module. When using an integrated Ethernet on the CPU, this byte must be set to "254" (0xFE) (independently of the CPU slot). Least significant byte: Destination node address The value in the least significant byte represents a direct or bridge address. If there is no bridge the value in the least significant byte is set to "0". If there is a bridge, this value contains the MBP for the Ethernet mapping index (MET). If the NOM is inserted in slot 7 on the module rack and the Ethernet mapping index (MET) is 6, the routing register looks as follows (value 0x0706)::  0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 **Most significant byte** Slots 1 ... 16 **Least significant byte** MBP on Ethernet Transporter (MET) mapping index
CONTROL[6]	Byte 4 (MSB) of the 32bit destination IP address
CONTROL[7]	Byte 3 of the 32bit destination IP address
CONTROL[8]	Byte 2 of the 32bit destination IP address
CONTROL[9]	Byte 1 (LSB) of the 32bit destination IP address

Control block for SY/MAX Ethernet

Control block for SY/MAX Ethernet:

Register	**Contents**
CONTROL[1]	indicates one of the Operations which are valid for SY/MAX.
CONTROL[2]	indicates the Error status.
CONTROL[3]	indicates the length (number of registers transferred)
CONTROL[4]	indicates MSTR operation-dependent information
CONTROL[5]	Routing Register Used for specification of a destination node during network transfer. Most significant byte: Source node address slot for the NOE module. Least significant byte: Destination node address The value in the least significant byte represents a direct or bridge address. If there is no bridge the value in the least significant byte is set to "0". If there is a bridge, this value contains the MBP for the Ethernet mapping index (MET). If NOM is inserted in slot 7 on the module rack and the Ethernet mapping index (MET) is 6, the routing register looks as follows (value 0x0706)::  most significant byte least significant byte 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 **Most significant byte** Slots 1 ... 16 **Least significant byte** MBP on Ethernet Transporter (MET) mapping index
CONTROL[6]	Destination drop number (or set to FF hex)
CONTROL[7]	Terminator (set to FF hex)

Write data

Description

The write operation transfers data to an addressed node. The transaction utilizes a master transaction path and may require several cycles.
An attempt to program the MBP_MSTR in such a way that it writes to its own drop address will generate an error in the CONTROL[2] register of the block. However, it is possible to perform a write operation to a non-existing slave register. The slave detects the status and logs it. This can last for several cycles.

Network implementation

The write operation can be performed on Modbus Plus, TCP/IP Ethernet and SY/MAX Ethernet networks.

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	1 = Write data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses sent to slave
CONTROL[4]	Determines the %MW starting register in the slave to which the data will be written (e.g. 1 = %MW1, 49 = %MW49)
CONTROL[5] ... CONTROL[9]	Routing register 1 is used to specify the address (routing path address one of five) of the node during a network transfer. The last byte in the routing path that is not zero, is the destination mode.

Use of control blocks for TCP/IP Ethernet (CONTROL)

Control block for TCP/IP Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	1 = Write data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses sent to slave
CONTROL[4]	Determines the CONTROL[ ]start address in the slave to which the data will be written.
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot Least significant byte: MBP on Ethernet Transporter (MET) mapping index
CONTROL[6] ... CONTROL[7]	Each address contains one byte of the 32bit IP address

Use of control blocks for SY/MAX Ethernet (CONTROL)

Control block for SY/MAX Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	1 = Write data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses sent to slave
CONTROL[4]	Determines the %MW starting register in the slave to which the data will be written (e.g. 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing Register Most significant byte: ´Network adapter module slot Least significant byte: Destination drop number
CONTROL[6] ... CONTROL[9]	Terminator: FF hex

Read data

Description

The read operation transfers data from a specified node on the network. The transaction utilizes a master transaction path and may require several cycles. An attempt to program the MBP_MSTR in such a way that it reads from its own station address will generate an error in the CONTROL[2] register of the block. But it is possible to perform a read operation on a non-existing register of the slave. The slave detects the status and logs it. This can last for several cycles.

Network implementation

The read operation can be performed on Modbus Plus, TCP/IP Ethernet and SY/MAX Ethernet networks.

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	2 = Read data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of registers to be read from the slave
CONTROL[4]	Determines the %MW starting register in the slave from which the data will be read (e.g. 1 = %MW1, 49 = %MW49).
CONTROL[5] ... CONTROL[6]	Routing register 1 is used to specify the address (routing path address one of five) of the node during a network transfer. The last byte in the routing path that is not zero, is the destination node.

Use of control blocks for TCP/IP Ethernet (CONTROL)

Control block for TCP/IP Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	2 = Read data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be read from the slave
CONTROL[4]	Determines the %MW starting register in the slave from which the data will be read (e.g. 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot Least significant byte: MBP on Ethernet Transporter (MET) mapping index
CONTROL[6] ... CONTROL[9]	Each address contains one byte of the 32bit IP address

Use of control blocks for SY/MAX Ethernet (CONTROL)

Control block for SY/MAX Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	2 = Read data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be read from the slave
CONTROL[4]	Determines the %MW starting register in the slave to which the data will be written (e.g. 1 = %MW1, 49 = %MW49).
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot Least significant byte: Destination drop number
CONTROL[6] ... CONTROL[9]	Terminator: FF hex

Read local statistics

Description

This operation reads the data from the local node. The operation is carried out in one cycle and does not require a master transaction path.

Network implementation

The write operation can be performed on Modbus Plus, TCP/IP Ethernet and SY/MAX Ethernet networks:

- List of available Modbus Plus network statistics (See *Modbus Plus network statistics, p. 109*)
- List of TCP/IP Ethernet network statistics (See *TCP/IP Ethernet Network Statistics, p. 114*)

Usage of control blocks for Modbus Plus (`CONTROL`)

Control block for Modbus Plus (`CONTROL`):

Register	Meaning
CONTROL[1]	3 = Read local statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be read from the local statistics (1...32)
CONTROL[4]	First address from which the statistics table must be read (Reg1=0)
CONTROL[5]	Routing register 1 is used to specify the address (routing path address one of five) of the node during a network transfer. The last byte in the routing path that is not zero, is the destination mode.

Use of control blocks for TCP/IP Ethernet (`CONTROL`)

Control block for TCP/IP Ethernet (`CONTROL`)

Register	Meaning
CONTROL[1]	3 = Read local statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be read from the local statistics (1...32)
CONTROL[4]	First address from which the statistics table must be read (Reg1=0)
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot
CONTROL[6] ... CONTROL[9]	No significance

Clear local statistics

Description

This operation deletes the statistics concerning the local node. The operation is carried out in one cycle and does not require a master transaction path.

Note: If you edit the "Clear local statistics" operation, only words 13 to 22 in the statistics table are cleared.

Network implementation

The operation can be performed on Modbus Plus and TCP/IP Ethernet networks.

- List of available Modbus Plus network statistics (See *Modbus Plus network statistics, p. 109*)
- List of TCP/IP Ethernet network statistics (See *TCP/IP Ethernet Network Statistics, p. 114*)

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	4 = Clear local statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5]	Routing register 1 is used to specify the address (routing path address one of five) of the node during a network transfer. The last byte in the routing path that is not zero, is the destination mode.

Use of control blocks for TCP/IP Ethernet (CONTROL)

Control block for TCP/IP Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	4 = Clear local statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot
CONTROL[6] ... CONTROL[9]	Reserved

Write Global Data (Peer Cop)

Description

This operation transfers data to the communication processor of the current node, so that it can be sent via the network as soon as the node receives the token. This data can be received by all nodes connected to the local network. The operation is carried out in one cycle and does not require a master transaction path.

Network implementation

The operation can only be performed on Modbus Plus networks.

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	5 = Write global data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be sent from State RAM into global data memory (comm processor) (1...32)
CONTROL[4]	Reserved
CONTROL[5]	If global data is sent via a NOM, you must enter the NOM module slot in the most significant byte of the register.

Read Global Data (Peer Cop)

Description

This operation reads data from the communications processor of any node connected to the network that sends out global data. The operation can take several cycles if the global data is not currently available with the nodes called. If global data is available, the operation is executed in one cycle. A master transaction path is not required.

Network implementation

The operation can only be performed on Modbus Plus networks.

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	6 = Read global data
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be sent from global data memory (comm processor) (1...32)
CONTROL[4]	Display of addresses available in scanned node (Is automatically updated)
CONTROL[5]	The lowest significant byte idicates the address of the node (value from 1 to 64) whose global data is to be read. If global data is received via a NOM, you must enter the NOM module slot in the most significant byte of the address.

Get remote statistics

Description

This operation reads the data referring to remote nodes on the network (see *Modbus Plus network statistics, p. 109* and *TCP/IP Ethernet Network Statistics, p. 114*). This operation can last for several cycles and does not require a master data transaction path.

With each query, the remote communications processor supplies a complete statistics table even if the query does not refer to the entire table. MBP_MSTR will then copy only those words into the identified $MW addresses that you queried.

Network implementation

The operation can be performed on Modbus Plus and TCP/IP Ethernet networks.

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	7 = Get remote statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be read from the statistics data field (1...54) The size of the data field may not be exceeded.
CONTROL[4]	First address from which the node statistics must be read. The number of available statistics registers may not be exceeded.
CONTROL[5] ... CONTROL[9]	Routing address 1 5 of the node. The last byte in the routing path that is not zero is the destination node.

Use of control blocks for TCP/IP Ethernet (CONTROL)

Control block for TCP/IP Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	7 = Get remote statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of addresses to be read from the statistics data field (1...54) The size of the data field may not be exceeded.
CONTROL[4]	First address from which the node statistics must be read. The number of available statistics registers may not be exceeded.
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot
CONTROL[6] ... CONTROL[9]	Each address contains one byte of the 32bit IP address

Clear remote statistics

Description

This operation clears the statistics concerning remote nodes on the network from the data field of the local node. This operation can last for several cycles and employs one single master data transaction path.

Note: If the "Clear remote statistics" operation is edited, only the words 13 through 22 of the statistics table (see *Modbus Plus network statistics, p. 109* and *TCP/IP Ethernet Network Statistics, p. 114*) will be deleted.

Network implementation

The write operation can be performed on Modbus Plus and TCP/IP Ethernet networks.

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	8 = Clear remote statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5] ... CONTROL[9]	Routing register 1 is used to specify the address (routing path address one of five) of the destination node during a network transfer. The last byte in the routing path that is not zero, is the destination mode.

Use of control blocks for TCP/IP Ethernet (CONTROL)

Control block for TCP/IP Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	8 = Clear remote statistics
CONTROL[2]	indicates the Error status.
CONTROL[3]	Reserved
CONTROL[4]	Reserved
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot
CONTROL[6] ... CONTROL[9]	Each address contains one byte of the 32bit IP address

Peer cop health

Description

This operation reads the selected data from the peer cop communications health table and downloads the respective data into the specified %MW addresses registers of State RAM. The Peer cop communications health table is 12 words long, MBP_MSTR indexes all words with 0 through 11.

Network implementation

The operation can only be performed on Modbus Plus networks.

Usage of control blocks for Modbus Plus (CONTROL)

Control block for Modbus Plus (CONTROL):

Register	Meaning
CONTROL[1]	9 = Peer cop health
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of words wanted by the peer cop table (1...12)
CONTROL[4]	First word to be read from the peer cop table (0...11; 0=first word in peer cop table and 11=last word in peer cop table)
CONTROL[5]	Routing address 1 If this is the second of two local nodes, set the High value byte to 1.

Optional module reset

Description

The "Reset optional module" operation leads a Quantum NOE option module to start a reset cycle to reset its working environment.

Network implementation

The write operation can be performed on TCP/IP Ethernet and SY/MAX Ethernet networks.

Use of control blocks for TCP/IP Ethernet (CONTROL)

Control block for TCP/IP Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	10 = Optional module reset
CONTROL[2]	indicates the Error status.
CONTROL[3]	No significance
CONTROL[4]	No significance
CONTROL[5]	Routing Register The number shown in the High value byte in area 1 through 16 indicates the slot where the option module is located.
CONTROL[6] ... CONTROL[9]	No significance

Use of control blocks for SY/MAX Ethernet (CONTROL)

Control block for SY/MAX Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	10 = Optional module reset
CONTROL[2]	indicates the Error status.
CONTROL[3]	No significance
CONTROL[4]	No significance
CONTROL[5]	Routing Register Most significant byte: Network adapter module slot
CONTROL[6] ... CONTROL[9]	No significance

Read CTE (Config extension table)

Description

The "Read CTE" operation reads a given number of bytes from the Ethernet configuration extension table in the specified buffer in the PLC memory. The bytes to be read start with a byte offset at the start of the CTE. The contents of the Ethernet CTE table is displayed on output `DATABUF`.

Network implementation

The write operation can be performed on TCP/IP Ethernet and SY/MAX Ethernet networks.

Use of control blocks for TCP/IP Ethernet (`CONTROL`)

Control block for TCP/IP Ethernet (`CONTROL`):

Register	Meaning
CONTROL[1]	11 = Read CTE (Config extension table)
CONTROL[2]	indicates the Error status.
CONTROL[3]	No significance
CONTROL[4]	No significance
CONTROL[5]	Routing Register Least significant byte = mapping index Either a value displayed in the byte of the register or is not used. or Most significant byte = network adapter module slot
CONTROL[6] ... CONTROL[9]	The number shown in the least significant byte in the area 1 through 16 indicates the slot where the optional module is located.

Use of control blocks for SY/MAX Ethernet (`CONTROL`)

Control block for SY/MAX Ethernet (`CONTROL`):

Register	Meaning
CONTROL[1]	11 = Read CTE (Config extension table)
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of words transferred
CONTROL[4]	Byte offset in the PLC register structure, specifying from where the CTE bytes are read.
CONTROL[5]	Routing Register Most significant byte: Slot of the NOE module
CONTROL[6] ... CONTROL[9]	Terminator: FF hex

CTE indicator implementation (DATABUF)

The values in the Ethernet configuration extension table (CTE) are displayed in a field on output DATABUF when a CTE read operation is implemented. The registers display the following CTE data:
CTE indicator implementation (DATABUF):

Parameter	Register	Contents
Frame type	DATABUF[0]	1 = 802.3 2 = Ethernet
IP address	DATABUF[1]	First byte of the IP address
	DATABUF[2]	Second byte of the IP address
	DATABUF[3]	Third byte of the IP address
	DATABUF[4]	Fourth byte of the IP address
Lower netmask	DATABUF[5]	Most significant word
	DATABUF[6]	Least significant word
Gateway	DATABUF[7]	First byte of the gateway
	DATABUF[8]	Second byte of the gateway
	DATABUF[9]	Third byte of the gateway
	DATABUF[10]	Fourth byte of the gateway

Write CTE (Config extension table)

Description

The "Write CTE" operation writes the CTE configuration table from the specified data (DATABUF) to a specified Ethernet configuration extension table or to a specific slot.

Network implementation

The write operation can be performed on TCP/IP Ethernet and SY/MAX Ethernet networks.

Use of control blocks for TCP/IP Ethernet (CONTROL)

Control block for TCP/IP Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	12 = Write CTE (Config extension table)
CONTROL[2]	indicates the Error status.
CONTROL[3]	No significance
CONTROL[4]	No significance
CONTROL[5]	Routing Register Least significant byte = mapping index Either a value displayed in the byte of the address or is not used. or Most significant byte = network adapter module slot
CONTROL[6] ... CONTROL[9]	The number shown in the least significant byte in the area 1 through 16 indicates the slot where the optional module is located.

Use of control blocks for SY/MAX Ethernet (CONTROL)

Control block for SY/MAX Ethernet (CONTROL):

Register	Meaning
CONTROL[1]	12 = Write CTE (Config extension table)
CONTROL[2]	indicates the Error status.
CONTROL[3]	Number of words transferred
CONTROL[4]	Byte offset in the PLC address structure specifying where the CTE bytes are written.
CONTROL[5]	Routing Register Most significant byte = NOE module slot Least significant byte = Destination drop number
CONTROL[6]	Terminator: FF hex
CONTROL[7] ... CONTROL[9]	No significance

CTE indicator implementation (DATABUF)

The values in the Ethernet configuration extension table (CTE) are displayed in a field on output DATABUF when a CTE write operation is implemented. The registers are used to transfer the following CTE data:
CTE indicator implementation (DATABUF):

Parameter	Register	Contents
Frame type	DATABUF[0]	1 = 802.3 2 = Ethernet
IP address	DATABUF[1]	First byte of the IP address
	DATABUF[2]	Second byte of the IP address
	DATABUF[3]	Third byte of the IP address
	DATABUF[4]	Fourth byte of the IP address
Lower netmask	DATABUF[5]	Most significant word
	DATABUF[6]	Least significant word
Gateway	DATABUF[7]	First byte of the gateway
	DATABUF[8]	Second byte of the gateway
	DATABUF[9]	Third byte of the gateway
	DATABUF[10]	Fourth byte of the gateway

Peer Cop Communications Health Status

Peer Cop Communications Health Status

The table containing the Peer Cop status information fills 12 consecutive registers, which can be indexed with the numbers 0 to 11 in an `MBP_MSTR` operation. Each individual bit of the table words is used to present one aspect of communications health that refers to a specific node on the Modbus Plus network.

Relation bit network node

The bits of the words 0 to 3 represent the health at the global communications input of nodes 1 to 64. The bits of words 4 ... 7 represent the health of the output of a specific node.
The bits in words 8 to 11 represent the health of the input of a specific node.

Status type	Word index	Relation bit network node															
Global receive	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	2	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
	3	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
Send direct	4	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	5	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	6	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
	7	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
Receive direct	8	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	9	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	10	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
	11	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

Health bit status

The status of the Peer Cop Health bit indicates the current communications status of its assigned node. A health bit will be set when the associated node accepts input for its Peer Cop data block or when it receives a signal that another node has accepted specific output data from its Peer Cop output data block. A health bit will be deleted if the associated data block did not accept any communication within the configured Peer Cop health timeout period.
All health bits will be deleted when interface command "Put Peer Cop" is executed during PLC startup. The table values become valid when the Token has been completely bypassed, after the interface command "Put Peer Cop" has been carried out. The health bit of a specific node is always zero when the assigned Peer Cop entry is zero.

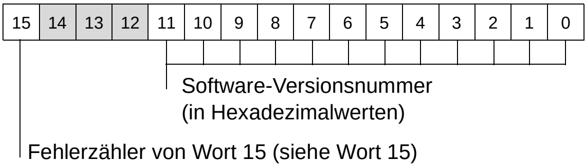
Modbus Plus network statistics

Modbus Plus network statistics

The following table shows the statistics available on Modbus Plus. You can obtain this data by running the corresponding `MBP_MSTR` operation (Modbus function code 8).

> **Note:** If you edit the "Clear local statistics" or "Clear remote statistics" operation, only words 13 to 22 in the statistics table are cleared.

Modbus Plus network statistics:

Word	Bits	Description
00		Node type ID
	0	Unknown node type
	1	PLC node
	2	Modbus bridge node
	3	Host computer node
	4	Bridge Plus node
	5	Peer I/O node
01	0 ... 11	Software version number as hexadecimal value (to read this, isolate bits 12-15 from the word)
	12 ... 14	Reserved
	15	Defines error counters from word 15. The most significant bit defines the use of error counters in word 15. The lower valued half of the most significant byte together with the least significant byte contain the software version.  15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Software-Versionsnummer (in Hexadezimalwerten) Fehlerzähler von Wort 15 (siehe Wort 15)
02		Network address of this station

Word	Bits	Description
03		MAC status variable:
	0	Startup status
	1	Offline status indicator signals
	2	Duplicated offline status
	3	Idle status
	4	Token utilization status
	5	Work response status
	6	Token transfer status
	7	Response request status
	8	Status check of transfer
	9	Token request status
	10	Response request status
04		Peer status (LED code); indicates status of this device relative to the network:
	0	Monitor connect operation
	32	Normal connect operation
	64	Never receives token
	96	Single station
	128	Duplicate station
05		Token transfer counter; increments each time this station receives the token
06		Token cycle time in ms
07	LOW	Bit representation data master fail during token ownership
	HIGH	Bit representation (bitmap) program master fail during token ownership
08	LOW	Bitmap activity token ownership of the data master
	HIGH	Bitmap activity token ownership of the program master
09	LOW	Bitmap activity token ownership of the data slave
	HIGH	Bitmap activity token ownership of the program slave
10	LOW	
	HIGH	Bitmap transfer request command data slave/slave poll
11	LOW	Bitmap response transfer request program master/master poll
	HIGH	Bitmap transfer request command program slave/slave poll
12	LOW	Bitmap connect status of the program master
	HIGH	Bitmap automatic log-off of program slave
13	LOW	Pretransfer delay error counter
	HIGH	Receive buffer DMA overrun error counter
14	LOW	Receive counter repeat command
	HIGH	Error counter data block size

Word	Bits	Description
15		If bit 15 of word 1 is not set, word 15 has the following significance:
	LOW	Error counter receiver collision abort
	HIGH	Error counter receiver alignment
		If bit 15 of word 1 is set, word 15 has the following significance:
	LOW	Data block error on cable B
	HIGH	Data block error on cable B
16	LOW	Error counter CRC receiver
	HIGH	Error counter wrong packet length
17	LOW	Error counter wrong link address
	HIGH	Error counter DMA underflow transfer buffer storage
18	LOW	Error counter wrong internal packet length
	HIGH	Error counter wrong MAC function code
19	LOW	Communication retry counter
	HIGH	Error counter communication failed
20	LOW	Counter package receipt successful
	HIGH	Error counter no response receipt
21	LOW	Error counter unexpected response receipt
	HIGH	Error counter unexpected path
22	LOW	Error counter unexpected response
	HIGH	Error counter skipped transaction
23	LOW	Bitmap active station table, nodes 1 through 8
	HIGH	Bitmap active station table, nodes 9 through 16
24	LOW	Bitmap active station table, nodes 17 through 24
	HIGH	Bitmap active station table, nodes 25 through 32
25	LOW	Bitmap active station table, nodes 33 through 40
	HIGH	Bitmap active station table, nodes 41 through 48
26	LOW	Bitmap active station table, nodes 49 through 56
	HIGH	Bitmap active station table, nodes 57 through 64
27	LOW	Bitmap token station table, nodes 1 through 8
	HIGH	Bitmap token station table, nodes 9 through 16
28	LOW	Bitmap token station table, nodes 17 through 24
	HIGH	Bitmap token station table, nodes 25 through 32
29	LOW	Bitmap token station table, nodes 33 through 40
	HIGH	Bitmap token station table, nodes 41 through 48
30	LOW	Bitmap token station table, nodes 49 through 56
	HIGH	Bitmap token station table, nodes 57 through 64

Word	Bits	Description
31	LOW	Bitmap table regarding existence of global data, nodes 1 through 8
	HIGH	Bitmap table regarding existence of global data, nodes 9 through 16
32	LOW	Bitmap table regarding existence of global data, nodes 17 through 24
	HIGH	Bitmap table regarding existence of global data, nodes 25 through 32
33	LOW	Bitmap table regarding existence of global data, nodes 33 through 40
	HIGH	Bitmap table regarding existence of global data, nodes 41 through 48
34	LOW	Bitmap table regarding existence of global data, nodes 49 through 56
	HIGH	Bitmap table regarding existence of global data, nodes 57 through 64
35	LOW	Bitmap receive buffer used, buffers 1 through 8
	HIGH	Bitmap receive buffer used, buffers 9 through 16
36	LOW	Bitmap receive buffer used, buffers 17 through 24
	HIGH	Bitmap receive buffer used, buffers 25 through 32
37	LOW	Bitmap receive buffer used, buffers 33 through 40
	HIGH	Counter of activated processed commands for station administration
38	LOW	Counter activation command output path 1 of the data master
	HIGH	Counter activation command output path 2 of the data master
39	LOW	Counter activation command output path 3 of the data master
	HIGH	Counter activation command output path 4 of the data master
40	LOW	Counter activation command output path 5 of the data master
	HIGH	Counter activation command output path 6 of the data master
41	LOW	Counter activation command output path 7 of the data master
	HIGH	Counter activation command output path 8 of the data master
42	LOW	Counter command processing input path 41 of the data slave
	HIGH	Counter command processing input path 42 of the data slave
43	LOW	Counter command processing input path 43 of the data slave
	HIGH	Counter command processing input path 44 of the data slave
44	LOW	Counter command processing input path 45 of the data slave
	HIGH	Counter command processing input path 46 of the data slave
45	LOW	Counter command processing input path 47 of the data slave
	HIGH	Counter command processing input path 48 of the data slave
46	LOW	Counter command activation output path 81 of the program master
	HIGH	Counter command activation output path 82 of the program master
47	LOW	Counter command activation output path 83 of the program master
	HIGH	Counter command activation output path 84 of the program master
48	LOW	Counter command activation output path 85 of the program master
	HIGH	Counter command activation output path 86 of the program master

Word	Bits	Description
49	LOW	Counter command activation output path 87 of the program master
	HIGH	Counter command activation output path 88 of the program master
50	LOW	Counter command processing input path C1 of the program slave
	HIGH	Counter command processing input path C2 of the program slave
51	LOW	Counter command processing input path C3 of the program slave
	HIGH	Counter command processing input path C4 of the program slave
52	LOW	Counter command processing input path C5 of the program slave
	HIGH	Counter command processing input path C6 of the program slave
53	LOW	Counter command processing input path C7 of the program slave
	HIGH	Counter command processing input path C8 of the program slave

TCP/IP Ethernet Network Statistics

TCP/IP Ethernet Network Statistics

A TCP/IP Ethernet module replies to the "Get local statistics" and "Set local statistics" commands using the following information:

Word	Meaning
00 to 02	MAC address e.g. MAC address 00 00 54 00 12 34 is displayed as follows: **Word** **Contents** 00 00 00 01 00 54 02 34 12
03	Module state ● 0x0001 = Running ● 0x4000 = APPI LED (1=ON, 0 = OFF) ● 0x8000 = LED connection
04 and 05	Number of receiver interrupts
06 and 07	Number of transfer interrupts
08 and 09	Transfer timeout error count
10 and 11	Collision detection error count
12 and 13	Omitted packets
14 and 15	Memory error count
16 and 17	Number of restarts performed by the driver
18 and 19	Receive framing error count
20 and 21	Overflow error count receiver
22 and 23	Receive CRC error counter
24 and 25	Receive buffer error counter
26 and 27	Transfer buffer error counter
28 and 29	Transfer bin underflow counter
30 and 31	Late collision counter
32 and 33	Lost carrier counter
34 and 35	Number of retries
36 and 37	IP address e.g. the IP address 198.202.137.113 (or c6 CA 89 71) is represented as follows: **Word** **Contents** 36 89 71 37 C6 CA

Modbus Plus and SY/MAX Ethernet Error Codes

Form of the function error code

Function error codes for Modbus Plus and SY/MAX Ethernet transactions appear as **Mmss**, where:

- **M** is the high code
- **m** is the low code
- **ss** is a subcode

Hexadecimal error code

Hexadecimal error code for Modbus Plus and SY/MAX Ethernet:

Hex. Error Code	Description
1001	Abort by user
2001	An operation type that is not supported has been specified in the control block
2002	One or more control block parameters were modified while the MSTR element was active (this only applies to operations which require several cycles for completion). Control block parameters my only be modified in inactive MSTR components.
2003	Invalid value in the length field of the control block
2004	Invalid value in the offset field of the control block
2005	Invalid value in the length and offset fields of the control block
2006	Unauthorized data field on slave
2007	Unauthorized network field on slave
2008	Unauthorized network routing path on slave
2009	Routing path equivalent to their own address
200A	Attempt to get more global data words than available
200E	The control block is not assigned, or parts of the control block are located outside of the %MW (4x) range.
30ss	Exceptional response by Modbus slave (See *ss hexadecimal value in 30ss error code, p. 116*)
4001	Inconsistent response by Modbus slave
5001	Inconsistent response by the network
6mss	Routing path error (See *ss hexadecimal value in 6mss error code, p. 116*) The subfield m shows where the error occurred (a 0 value means local node, 2 means 2nd device in route, etc).

ss hexadecimal value in 30ss error code

ss hexadecimal value in 30ss error code:

ss hex. Value	Description
01	Slave does not support requested operation
02	Non-existing slave registers were requested
03	An unauthorized data value was requested
05	Slave has accepted a lengthy program command
06	Function cannot currently be carried out: lengthy command running
07	Slave has rejected lengthy program command

ss hexadecimal value in 6mss error code

Note: Subfield m in error code 6mss is an `Index` in the routing information that shows where an error has been detected (a 0 value indicates the local node, 2 means the second device in the route, etc.).

The ss subfield in error code 6mss is as follows:

ss hexadecimal value	Description
01	No response reception
02	Access to program denied
03	Node out of service and unable to communicate
04	Unusual response received
05	Router-node data path busy
06	Slave out of order
07	Wrong destination address
08	Unauthorized node type in routing path
10	Slave has rejected the command
20	Slave has lost an activated transaction
40	Unexpected master output path received
80	Unexpected response received
F001	Wrong destination node was specified for the `MSTR` operation

SY/MAX-specific error codes

SY/MAX-specific error codes

When utilizing SY/MAX Ethernet, three additional types of errors may appear in the `CONTROL[1]` register of the control block ().
The error codes have the following meaning:

- 71xx Error: Errors found by the SY/MAX remote device
- 72xx Error: Errors found by the server
- 73xx Error: Errors found by the Quantum translator

SY/MAX-specific HEX error code

SY/MAX-specific HEX error code:

Hex. Error Code	Description
7101	Invalid opcode found by the SY/MAX remote device
7103	Invalid address found by the SY/MAX remote device
7109	Attempt to write to a write protected register found by the SY/MAX remote device
F710	Receiver overflow found by the SY/MAX remote device
7110	Invalid length found by the SY/MAX remote device
7111	Remote device not active, no connection (occurs when retry attempts and time-out have been used up), found by the SY/MAX remote device
7113	Invalid parameter in a read operation found by the SY/MAX remote device
711D	Invalid route found by the SY/MAX remote device
7149	Invalid parameter in a write operation found by the SY/MAX remote device
714B	Invalid drop number found by the SY/MAX remote device
7101	Invalid opcode found by the SY/MAX server
7203	Invalid address found by the SY/MAX server
7209	Attempt to write to a write protected register found by the SY/MAX server
F720	Receiver overflow found by the SY/MAX server
7210	Invalid length found by the SY/MAX server
7211	Remote device not active, no connection (occurs when retry attempts and time-out have been used up), found by the SY/MAX server
7213	Invalid parameter in a read operation found by the SY/MAX server
721D	Invalid route found by the SY/MAX server
7249	Invalid parameter in a write operation found by the SY/MAX server
724B	Invalid drop number found by the SY/MAX server
7301	Invalid opcode in an `MSTR` block request from the Quantum translator
7303	Read/Write QSE module status (200 route address out of range)
7309	Attempt to write to a write protected register when a status write is carried out (200 route)

Hex. Error Code	Description
731D	Invalid route found by the Quantum translator. Valid routes: ● dest_drop, 0xFF ● 200, dest_drop, 0xFF ● 100+drop, dest_drop, 0xFF ● All other routing values produce an error
734B	One of the following errors occurred: ● No CTE (configuration extension table) has been configured ● No CTE table entry has been made for the QSE model slot number ● No valid drop has been specified ● The QSE module has not been reset after the creation of the CTE. Note: After writing and configuring the CTE and downloading to the QSE module, the QSE module must be reset for the modifications to become effective. ● When using an `MSTR` instruction no valid slot or drop has been specified

TCP/IP Ethernet error codes

TCP/IP Ethernet error codes

An error in an MSTR routine via TCP/IP Ethernet may produce one of the following errors in the MSTR control block:
The error code appears as **Mmss**, where:

- **M** is the high code
- **m** is the low code
- **ss** is a subcode

HEX error codes TCP/IP Ethernet

HEX error codes TCP/IP Ethernet:

Hex. Error Code	Meaning
1001	Abort by user
2001	An operation type that is not supported has been specified in the control block
2002	One or more control block parameters were modified while the MSTR element was active (this only applies to operations which require several cycles for completion). Control block parameters my only be modified in inactive MSTR components.
2003	Invalid value in the length field of the control block
2004	Invalid value in the offset field of the control block
2005	Invalid value in the length and offset fields of the control block
2006	Unauthorized data field on slave
200E	The control block is not assigned, or parts of the control block are located outside of the %MW (4x) range.
3000	Generic Modbus failure code
30ss	Exceptional response by Modbus slave (See *ss hexadecimal value in 30ss error code, p. 119*)
4001	Inconsistent response by Modbus slave

ss hexadecimal value in 30ss error code

ss hexadecimal value in 30ss error code:

ss hex. Value	Meaning
01	Slave does not support requested operation
02	Non-existing slave registers were requested
03	An unauthorized data value was requested
05	Slave has accepted a lengthy program command
06	Function cannot currently be carried out: lengthy command running
07	Slave has rejected lengthy program command

HEX error codes TCP/IP Ethernet network

An error on the TCP/IP Ethernet network itself may produce one of the following errors in the `CONTROL[1]` register of the control block.

HEX error codes TCP/IP Ethernet network:

Hex. Error Code	Meaning
5004	Interrupted system invocation
5005	I/O error
5006	No such address
5009	The socket descriptor is not valid
500C	Not enough storage space
500D	Authorization denied
5011	Entry exists
5016	An argument is not valid
5017	An internal table has no more space
5020	There is interference on the connection
5023	This operation was blocked and the socket is non-blocking
5024	The socket is non-blocking and the connection cannot be closed down
5025	The socket is non-blocking and a previous connection attempt has not been concluded
5026	Socket operation on a non-socket
5027	The destination address is not valid
5028	Message too long
5029	Wrong type of protocol for the socket
502A	Protocol not available
502B	Protocol not supported
502C	Socket type not supported
502D	Operation not supported at socket
502E	Protocol family not supported
F502	Address family not supported
5030	Address is already in use
5031	Address not available
5032	Network is out of order
5033	Network cannot be reached
5034	Network shut down the connection during reset
5035	The connection was terminated by the peer
5036	The connection was reset by the peer

Hex. Error Code	Meaning
5037	An internal buffer is required, but cannot be assigned
5038	The socket is already connected
5039	The socket is not connected
503A	Cannot transmit after the socket has been shut off
503B	Too many references; cannot splice
503C	Connection timed out
503D	The connection attempt was denied
5040	Host is out of order
5041	The destination host could not be reached from this node
5042	Directory not empty
5046	NI_INIT returned -1
5047	The MTU is not valid
5048	The hardware length is not valid
5049	The route specified cannot be found
504A	Collision when invoking Select; these conditions have already been selected by another job
504B	The job ID is not valid
5050	No Network Resource
5051	Length Error
5052	Addressing Error
5053	Application Error
5054	Client cannot process request
5055	No Network Resource
5056	Non-Operational TCP connection
5057	Incoherent configuration
6003	FIN or RST not expected
F001	In reset mode
F002	Component not fully initialized

CTE error codes for SY/MAX and TCP/IP Ethernet

CTE error codes for SY/MAX and TCP/IP Ethernet

The following error codes are displayed in the `CONTROL[1]` register of the control block, if there is a problem with the Ethernet configuration extension table (CTE) in your program configuration.
CTE error codes for SY/MAX and TCP/IP Ethernet:

Hex. Error Code	**Description**
7001	There is no Ethernet configuration extension
7002	The CTE is not available for access
7003	The offset is not valid
7004	Offset + length are not valid
7005	Bad data field in the CTE

ModbusP_ADDR: Modbus Plus Address

9

Overview

Introduction

This chapter describes the ModbusP_ADDR block.

What's in this Chapter?

This chapter contains the following topics:

Description

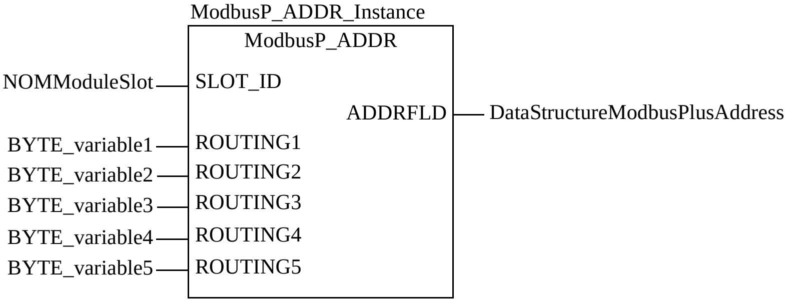
Function description

This function block allows the entry of the Modbus Plus address for function blocks READ_REG, CREAD_REG, WRITE_REG and CWRITE_REG. The address is transferred as a data structure.
EN and ENO can be configured as additional parameters.

Note: You must be familiar with your network when programming the ModbusP_ADDR function block. Modbus Plus routing path structures are described in detail in "Modbus Plus Network Planning and Installation Guide".

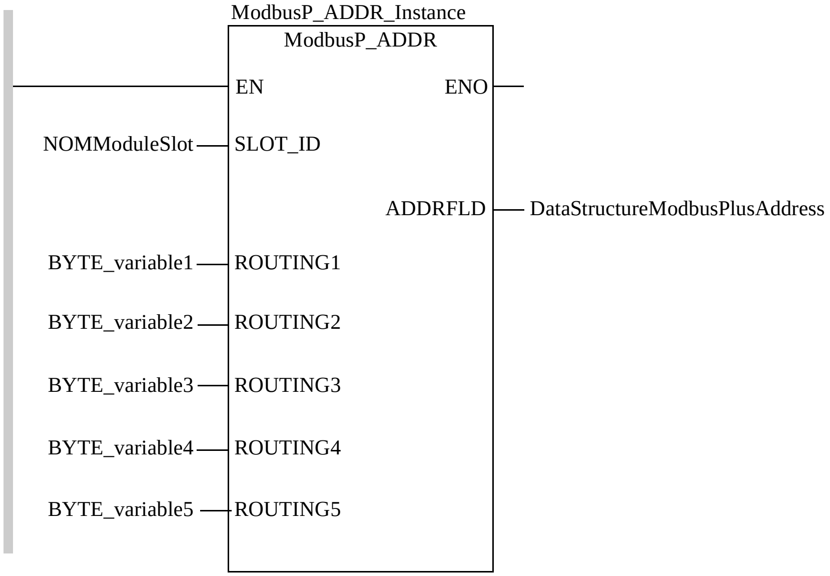
Representation in FBD

Representation:



Representation in LD

Representation:



Representation in IL

Representation:

```
CAL ModbusP_ADDR_Instance (SLOT_ID:=NOMModuleSlot,
    ROUTING1:=BYTE_variable1, ROUTING2:=BYTE_variable2,
    ROUTING3:=BYTE_variable3, ROUTING4:=BYTE_variable4,
    ROUTING5:=BYTE_variable5,
    ADDRFLD=>DataStructureModbusPlusAddress)
```

Representation in ST

Representation:

```
ModbusP_ADDR_Instance (SLOT_ID:=NOMModuleSlot,
    ROUTING1:=BYTE_variable1, ROUTING2:=BYTE_variable2,
    ROUTING3:=BYTE_variable3, ROUTING4:=BYTE_variable4,
    ROUTING5:=BYTE_variable5,
    ADDRFLD=>DataStructureModbusPlusAddress) ;
```

Parameter description

Description of input parameters:

Parameter	Data type	Description
Slot_ID	BYTE	Slot ID NOM module slot
ROUTING1	BYTE	Routing 1 is used for address specification (routing path addresses one of five) of the destination node during network transfer. The last byte in the routing path that is not zero is the destination node.
ROUTING2	BYTE	Routing 2
ROUTING3	BYTE	Routing 3
ROUTING4	BYTE	Routing 4
ROUTING5	BYTE	Routing 5

Description of output parameters:

Parameter	Data type	Description
ADDRFLD	WordArr5	Data structure used to transfer the Modbus Plus address

Detailed Description

Derived Data Types

Element description for `WordArr5`:

Element	Data type	Description
`WordArr5[1]`	`WORD`	Routing tab 1 Least significant byte: used for address specification (routing path addresses one of five) of a destination node during network transfer. Most significant byte: Slot of the network adapter module (NOM), if available.
`WordArr5[2]`	`WORD`	Routing tab 2
`WordArr5[3]`	`WORD`	Routing tab 3
`WordArr5[4]`	`WORD`	Routing tab 4
`WordArr5[5]`	`WORD`	Routing tab 5

Slot_ID

If a Modbus Plus network option module (NOM) in the rack of a Quantum controller is addressed as the destination node, the value at the `Slot_ID` input represents the physical NOM slot, i.e. if the NOM is plugged in at Slot 7 of the rack, the value appears as follows:

0	0	0	0	0	1	1	1

Routing x

The `Routing x` input is used for address specification (routing path addresses one of five) of the destination node during network transfer. The last byte in the routing path that is not zero is the destination node.

0	x	x	x	x	x	x	x

Destination address (binary value between 1 and 64 (normal) or 65 to 249 (extended))

Routing tab 1

If a Modbus Plus Network Options Module (NOM) is addressed as the destination node in a Quantum control module rack, the most significant byte represents the physical slot of the NOM. If the destination node is a CPU, the most significant byte (regardless of the CPU slot) is set to "0".
If NOM is inserted in slot 7 on the module rack, the most significant byte of routing tab 1 looks as follows:

<table>
<tr><th colspan="8">Most significant byte</th><th colspan="8">Least significant byte</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
</table>

Most significant byte Slots 1 ... 16

Least significant byte Destination address (binary value between 1 and 64 (normal) or 65 to 255 (extended))

READ_REG: Read register

10

Overview

Introduction

This chapter describes the READ_REG block.

What's in this Chapter?

This chapter contains the following topics:

Description

Function description

With a rising edge at the REQ input, this function block reads a register area from an addressed slave via Modbus Plus, TCP/IP-Ethernet or SY/MAX-Ethernet.

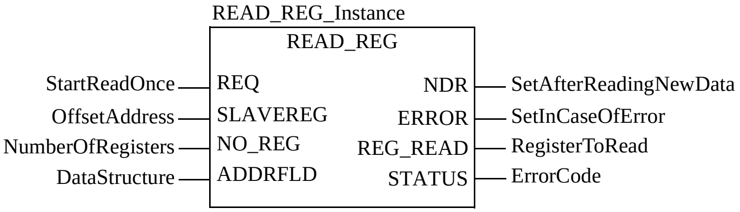
Note: When programming a READ_REG function, you must be familiar with the routing procedures used by your network. Modbus Plus routing path structures are described in detail in the *Modbus Plus Network Planning and Installation Guide*. If TCP/IP or SY/MAX Ethernet routing is implemented, standard Ethernet IP router products must be used. A full description of the TCP/IP routing is provided in the *Quantum with Unity Pro TCP/IP Configuration User Guide*.

Note: Several copies of this function block can be used in the program. However, multiple instancing of these copies is not possible.

EN and ENO can be configured as additional parameters.

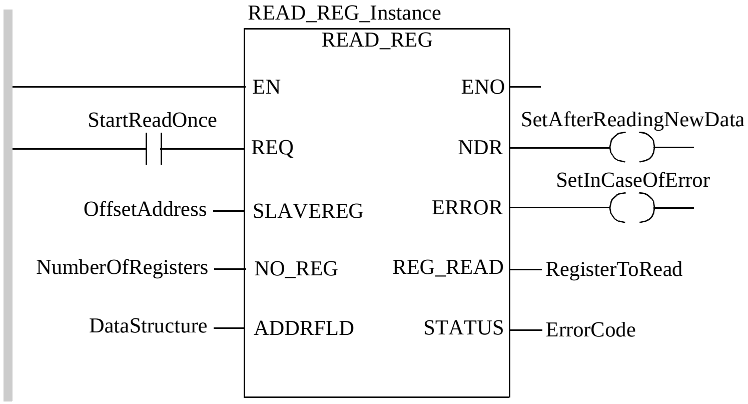
Representation in FBD

Representation:



Representation in LD

Representation:



Representation in IL

Representation:

```
CAL READ_REG_Instance (REQ:=StartReadOnce,
    SLAVEREG:=OffsetAddress, NO_REG:=NumberOfRegisters,
    ADDRFLD:=DataStructure, NDR=>SetAfterReadingNewData,
    ERROR=>SetInCaseOfError, REG_READ=>RegisterToRead,
    STATUS=>ErrorCode)
```

Representation in ST

Representation:

```
READ_REG_Instance (REQ:=StartReadOnce,
    SLAVEREG:=OffsetAddress, NO_REG:=NumberOfRegisters,
    ADDRFLD:=DataStructure, NDR=>SetAfterReadingNewData,
    ERROR=>SetInCaseOfError, REG_READ=>RegisterToRead,
    STATUS=>ErrorCode) ;
```

Parameter description

Description of input parameters:

Parameter	Data type	Meaning
REQ	BOOL,	With a rising edge at the REQ input, this function block reads a register area from an addressed slave via Modbus Plus, TCP/IP-Ethernet or SY/MAX-Ethernet.
SLAVEREG	DINT	Offset address of the first %MW register in the slave to be read from.
NO_REG	INT	Number of addresses to be read from slave
ADDRFLD	WordArr5	Data structure describing the Modbus Plus-address, TCP/IP address or SY/MAX-IP address.

Description of output parameters:

Parameter	Data type	Meaning
NDR	BOOL	Set to "1" for one cycle after reading new data
ERROR	BOOL	Set to "1" for one cycle if an error occurs
STATUS	WORD,	If an error occurs while the function is being executed, an error code remains at this output for one cycle. Error code, see ● *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115* ● *SY/MAX-specific error codes, p. 117* ● *TCP/IP Ethernet error codes, p. 119*
REG_READ	ANY	Writing data (For the file to be read a data structure must be declared as a located variable.)

Runtime error

For a list of all block error codes and values, see .

Derived Data Types

Element description for WordArr5 in Modbus Plus

Element description for WordArr5 in Modbus Plus:

Element	Data type	Description
WordArr5[1]	WORD	Least significant byte: Routing register 1 is used for address specification (routing path addresses one of five) of the destination node during network transfer. The last byte in the routing path that is not zero is the destination node. Most significant byte: Slot of the network adapter module (NOM), if any.
WordArr5[2]	WORD	Routing register 2
WordArr5[3]	WORD	Routing register 3
WordArr5[4]	WORD	Routing register 4
WordArr5[5]	WORD	Routing register 5

Element description for WordArr5 with TCP/IP Ethernet

Element description for WordArr5 with TCP/IP Ethernet

Element	Data type	Description
WordArr5[1]	WORD	Least significant byte: MBP on Ethernet Transporter (MET) mapping index Most significant byte: Slot of the NOE module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Element description for WordArr5 with SY/MAX Ethernet

Element description for WordArr5 with SY/MAX Ethernet:

Element	Data type	Description
WordArr5[1]	WORD	Least significant byte: MBP on Ethernet Transporter (MET) mapping index Most significant byte: Slot of the NOE module
WordArr5[2]	WORD	Destination drop number (or set to FF hex)
WordArr5[3]	WORD	Terminator (set to FF hex)
WordArr5[4]	WORD	No significance
WordArr5[5]	WORD	No significance

Function mode

Function mode of **READ_REG** blocks

Although a large number of READ_REG function blocks can be programmed, only four read operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or others (e.g. MBP_MSTR, CREAD_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC (NOE 211 00) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only **one** read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

All routing information is contained in the WordArr5 data structure of the ADDRFLD input. The type function block assigned to this input is defined by the network used. Please use:
- Modbus Plus for function block ModbusP_ADDR
- TCP/IP Ethernet for function block TCP_IP_ADDR
- SY/MAX Ethernet for function block SYMAX_IP_ADDR

Note: The WordArr5 data structure can also be used with constants.

Parameter description

REQ

A rising edge triggers the read transaction.
The parameter can be entered as a direct address, located variable, unlocated variable or literal.

SLAVEREG

Start of the area in the addressed slave from which the source data is read. The source area always resides within the %MW register area.

> **Note:** For slaves for a **non-**Unity Pro PLC:
> The source area always resides within the 4x register area. SLAVEREG expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
> The parameter can be entered as a direct address, located variable, unlocated variable or literal.

NO_REG

Number of addresses to be read from the addressed slave (1 ... 100).
The parameter can be entered as a direct address, located variable, unlocated variable or literal.

NDR

Transition to ON state for one program cycle signifies receipt of new data ready to be processed.
The parameter can be entered as a direct address, located variable or unlocated variable.

ERROR

Transition to ON state for one program cycle signifies the detection of a new error.
The parameter can be entered as a direct address, located variable or unlocated variable.

REG_READ

An ARRAY that is the same size as the requested transmission must be agreed upon (≥ NO_REG) for this parameter. The name of this array is defined as a parameter. If the array is defined too small, then only the amount of data is transmitted that is present in the array.
The parameter must be defined as a located variable.

STATUS

If an error occurs while the function is being executed, an error code remains at this output for one cycle.

Error code, see

- *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115*
- *SY/MAX-specific error codes, p. 117*
- *TCP/IP Ethernet error codes, p. 119*

The parameter can be entered as an address, located variable or unlocated variable.

WRITE_REG: Write register

11

Overview

Introduction

This chapter describes the WRITE_REG block.

What's in this Chapter?

This chapter contains the following topics:

Description

Function description

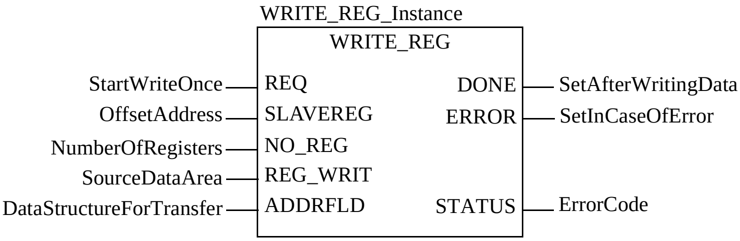
With a rising edge at the REQ input, this function block writes a register area from a register area from the PLC via Modbus Plus, TCP/IP Ethernet or SY/MAX-Ethernet to an addressed slave.
EN and ENO can be configured as additional parameters.

Note: When programming a WRITE_REG function, you must be familiar with the routing procedures used by your network. Modbus Plus routing path structures are described in detail in the *Modbus Plus Network Planning and Installation Guide*. If TCP/IP or SY/MAX Ethernet routing is implemented, standard Ethernet IP router products must be used. A full description of the TCP/IP routing is provided in the *Quantum with Unity ProTCP/IP Configuration User Guide*.

Note: Several copies of this function block can be used in the program. However, multiple instancing of these copies is not possible.

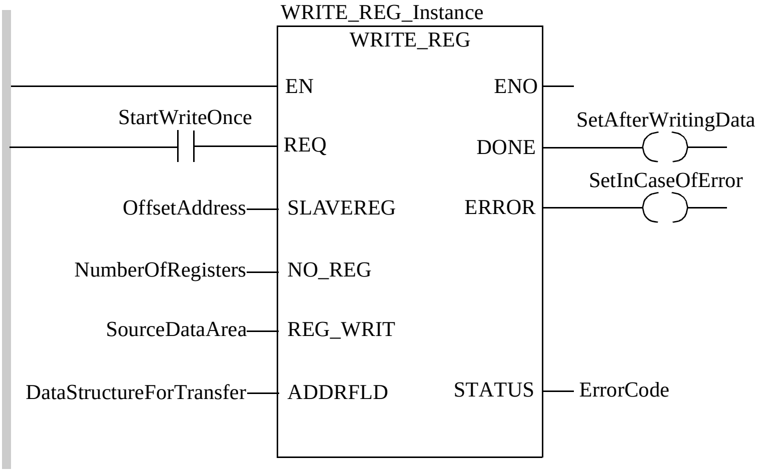
Representation in FBD

Representation:



Representation in LD

Representation:



Representation in IL

Representation:

```
CAL WRITE_REG_Instance (REQ:=StartWriteOnce,
    SLAVEREG:=OffsetAddress, NO_REG:=NumberOfRegisters,
    REG_WRIT:=SourceDataArea,
    ADDRFLD:=DataStructureForTransfer,
    DONE=>SetAfterWritingData, ERROR=>SetInCaseOfError,
    STATUS=>ErrorCode)
```

Representation in ST

Representation:

```
WRITE_REG_Instance (REQ:=StartWriteOnce,
    SLAVEREG:=OffsetAddress, NO_REG:=NumberOfRegisters,
    REG_WRIT:=SourceDataArea,
    ADDRFLD:=DataStructureForTransfer,
    DONE=>SetAfterWritingData, ERROR=>SetInCaseOfError,
    STATUS=>ErrorCode) ;
```

Parameter description

Description of input parameters:

Parameter	Data type	Meaning
REQ	BOOL	With a rising edge at the REQ input, this function block writes a register area from a register area from the PLC via Modbus Plus, TCP/IP Ethernet or SY/MAX-Ethernet to an addressed slave.
SLAVEREG	DINT	Offset address of the first %MW register in the slave to be written to.
NO_REG	INT	Number of addresses to be written from slave
REG_WRIT	ANY	Source data field (A data structure must be declared as a located variable for the source file .)
ADDRFLD	WordArr5	Data structure transferring the Modbus Plus-address, TCP/IP address or SY/MAX-IP address.

Description of output parameters:

Parameter	Data type	Meaning
DONE	BOOL	Set to "1" for one cycle after writing data.
ERROR	BOOL	Set to "1" for one cycle if an error occurs.
STATUS	WORD	If an error occurs while the function is being executed, an error code remains at this output for one cycle. Error code, see ● *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115* ● *SY/MAX-specific error codes, p. 117* ● *TCP/IP Ethernet error codes, p. 119*

Runtime error

For a list of all block error codes and values, see .

Derived Data Types

Element description for WordArr5 in Modbus Plus

Element description for WordArr5 in Modbus Plus:

Element	Data type	Description
WordArr5[1]	WORD	Least significant byte: Routing register 1 is used for address specification (routing path addresses one of five) of the destination node during network transfer. The last byte in the routing path that is not zero is the destination node. Most significant byte: Slot of the network adapter module (NOM), if any.
WordArr5[2]	WORD	Routing register 2
WordArr5[3]	WORD	Routing register 3
WordArr5[4]	WORD	Routing register 4
WordArr5[5]	WORD	Routing register 5

Element description for WordArr5 with TCP/IP Ethernet

Element description for WordArr5 with TCP/IP Ethernet

Element	Data type	Description
WordArr5[1]	WORD	Most significant byte: Slot of the NOE module Least significant byte: MBP on Ethernet Transporter (MET) mapping index
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Element description for `WordArr5` with SY/MAX Ethernet

Element description for `WordArr5` with SY/MAX Ethernet:

Element	Data type	Description
`WordArr5[1]`	`WORD`	Most significant byte: Slot of the NOE module Least significant byte: MBP on Ethernet Transporter (MET) mapping index
`WordArr5[2]`	`WORD`	Destination drop number (or set to FF hex)
`WordArr5[3]`	`WORD`	Terminator (set to FF hex)
`WordArr5[4]`	`WORD`	No significance
`WordArr5[5]`	`WORD`	No significance

Function mode

Function mode of the WRITE_REG block

Although a large number of WRITE_REG function blocks can be programmed, only four write operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or others (e.g. MBP_MSTR, CWRITE_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.
If several WRITE_REG function blocks are used within an application, they must at least differ in the values of their NO_REG or REG_WRIT parameters.

Note: A TCP/IP communication between a Quantum PLC (NOE 211 00) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only **one** read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The status signals DONE and ERROR report the function block state to the user program.
All routing information is contained in the WordArr5 data structure of the ADDRFLD input. The type function block assigned to this input is defined by the network used. Please use:

- Modbus Plus for function block ModbusP_ADDR (See *ModbusP_ADDR: Modbus Plus Address, p. 123*)
- TCP/IP Ethernet for function block TCP_IP_ADDR
- SY/MAX Ethernet for function block SYMAX_IP_ADDR

Note: The WordArr5 data structure can also be used with constants.

Parameter description

REQ

A rising edge triggers the write transaction.
The parameter can be entered as an address, located variable, unlocated variable or literal.

SLAVEREG

Start of the area in the addressed slave to which the source data is written. The source area always resides within the %MW address area.

> **Note:** For slaves for a **non-**Unity Pro PLC:
> The destination area always resides within the 4x register area. `SLAVEREG` expects the target address as an offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
> The parameter can be entered as an address, located variable, unlocated variable or literal.

NO_REG

Number of addresses to be written to slave processor (1 ... 100).
The parameter can be entered as an address, located variable, unlocated variable or literal.

REG_WRIT

An `ARRAY` that is the same size as the planned transmission must be agreed upon (≥ `NO_REG`) for this parameter. The name of this array is defined as a parameter. If the array is defined too small, then only the amount of data is transmitted that is present in the array.
The parameter must be defined as a located variable.

DONE

Transition to ON state for one program scan signifies that the data has been transferred.
The parameter can be entered as an address, located variable or unlocated variable.

ERROR

Transition to ON state for one program cycle signifies the detection of a new error.
The parameter can be entered as an address, located variable or unlocated variable.

STATUS

If an error occurs while the function is being executed, an error code remains at this output for one cycle.
Error code, see

- *Modbus Plus and SY/MAX Ethernet Error Codes, p. 115*
- *SY/MAX-specific error codes, p. 117*
- *TCP/IP Ethernet error codes, p. 119*

The parameter can be entered as an address, located variable or unlocated variable.

Hardware

IV

At a Glance

Overview

This part of the document contains information about Modbus Plus Networks

What's in this Part?

This part contains the following chapters:

Modbus Plus Network Option Modules (NOM)

12

Overview

Introduction

This chapter provides information on the following Quantum network option modules:

NOM	Communication Channels
140 NOM 211 00	1 Modbus (RS-232) serial port 1 Modbus Plus network (RS-485) port
140 NOM 212 00	1 Modbus (RS-232) serial port 2 Modbus Plus network (RS-485) port
140 NOM 252 00	1 Modbus (RS-232) serial port 2 Modbus Plus on Fiber (consisting of optical receiver and transmitter)

What's in this Chapter?

This chapter contains the following sections:

12.1 140 NOM 211 00: Modbus Plus Option Module

Overview

Introduction

This section describes the 140 NOM 211 00 Modbus Plus option module.

What's in this Section?

This section contains the following topics:

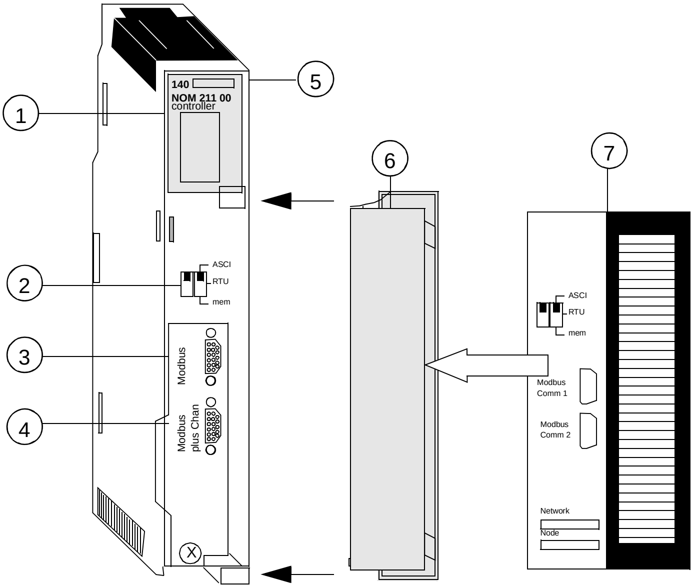
Presentation

Function

The 140 NOM 211 00 is a single channel Network Option Modul (NOM), connected via a twisted pair Modbus Plus cable network

Illustration

The following figure shows the parts of the Modbus Plus 140 NOM 211 00 modules.



1 LED Area

2 Comm Parameter Slide Switch

3 Modbus Connector

4 Modbus Plus Connector

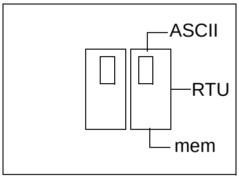
5 Model Number, Module Description, Color Code

6 Removable door

7 Customer Identification Label, (Fold label and place it inside door)

Front Panel Switches

Two, three-position slide switches are located on the front of the unit. The switch on the left is not used. The three-position slide switch on the right is used to select the comm parameter settings for the Modbus (RS-232) port provided with the Modbus Plus option module. Three options are available, as shown below.
The following figure shows the front panel switches.



Note: The NOM hardware defaults to bridge mode when the front panel switch is set to RTU or ASCII mode. When networking controllers, a panel device connected to the NOM Modbus port can communicate with the controller to which it is conected, as well as log into any nodes on the Modbus Plus network.

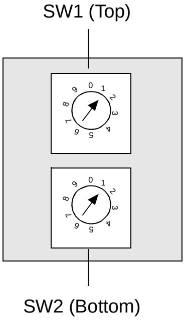
Rear Panel Switches

Two rotary switches are located on the rear panel of the modules. They are used together to set the Modbus Plus node and Modbus port address for the unit.

Note: The highest address that may be set with these switches is 64.
Rotary SW1 (top switch) sets the upper digit (tens), and rotary SW2 (bottom switch) sets the lower digit (ones) of the Modbus Plus node address. The illustration below shows the setting for an example address of 11.

SW1 and SW2 Switches Figure

The following figure shows the SW1 and SW2 switches.



Note: If "0," or an address greater than 64 is selected, the Modbus + LED will be "on" steady, to indicate the selection of an invalid address.

SW1 and SW2 Address Settings

The following table shows the address settings for the SW1 and SW2 switches.

Node Address	**SW1**	**SW2**
1 ... 9	0	1 ... 9
10 ... 19	1	0 ... 9
20 ... 29	2	0 ... 9
30 ... 39	3	0 ... 9
40 ... 49	4	0 ... 9
50 ... 59	5	0 ... 9
60 ... 64	6	1 ... 4

Note: If "0," or an address greater than 64 is selected, the Modbus + LED will be "on" steady, to indicate the selection of an invalid address.

ASCII Comm Port Parameters

The following table shows the fixed setting of the ASCII comm port parameters.

Baud	2,400
Parity	Even
Data Bits	7
Stop Bits	1
Device Address	Rear panel rotary switch setting

Setting the slide switch to the middle position assigns remote terminal unit (RTU) functionality to the port; the following comm parameters are set and cannot be changed:

RTU Comm Port Parameters

The following table shows the RTU comm port parameters.

Baud	9,600
Parity	Even
Data Bits	8
Stop Bits	1
Device Address	Rear panel rotary switch setting

Setting the slide switch to the bottom position gives you the ability to assign comm parameters to the port in software; the following parameters are valid.

Valid Comm Port Parameters

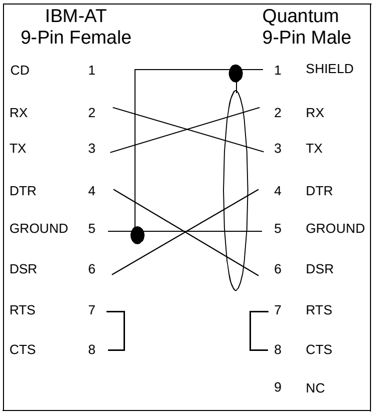
The following table shows the valid comm port parameters.

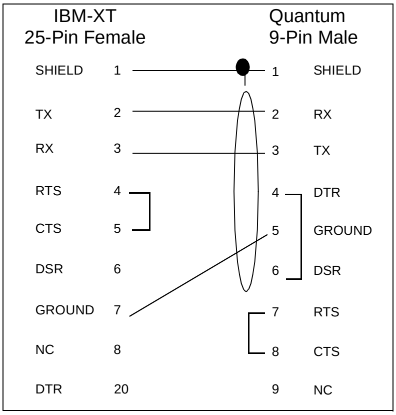
Baud	19,200	1,200
	9,600	600
	7,200	300
	4,800	150
	3,600	134.5
	2,400	110
	2,000	75
	1,800	50
Data Bits	7 / 8	
Stop Bits	1 / 2	
Parity	Enable/Disable Odd/Even	
Device Address	Rear panel rotary switch setting	

Modbus Connector Pinouts

The NOM modules are equipped with a nine-pin RS-232C connector that supports Modicon’s proprietary Modbus communication protocol. The following is the Modbus port pinout connections for 9-pin and 25-pin connections.

The following figures show the Modbus port pinout connections for 9-pin (left) and 25-pin (right).



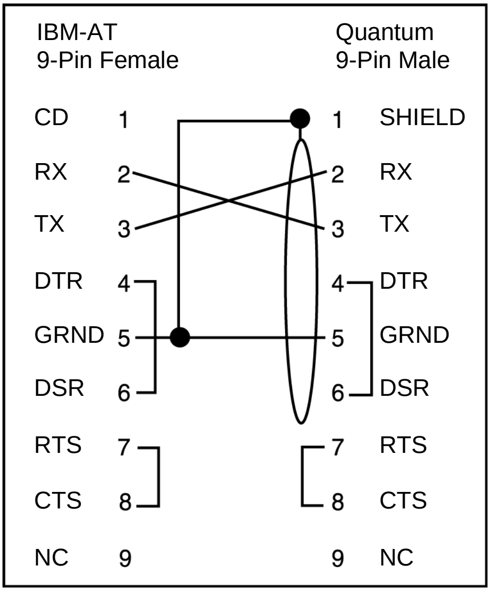


The following is the abbreviation key for the above figure.

TX: Transmitted Data	DTR: Data Terminal Ready
RX: Received Data	CTS: Clear to Send
RTS: Request to Send	N/C: No Connection
DSR: Data Set Ready	CD: Carrier Detect

Modbus Ports Pinout Connections for Portable Computers

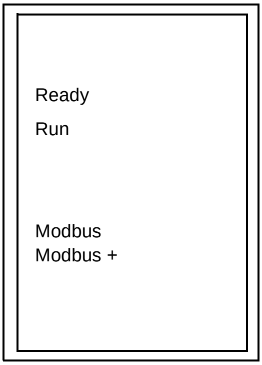
The following figure shows the Modbus port pinout connections for 9-pin portable computers.



Indicators

Illustration

The following figure shows the Modbus Plus NOM LED indicators.



Description

The following table shows the Modbus Plus NOM LED Descriptions.

LEDs	Color	Indication when On
Ready	Green	The module has passed powerup diagnostics.
Run	Green	Indicates that the unit is in kernel mode–should always be OFF during normal operations.
Modbus	Green	Indicates communication is active on the single RS-232 serial port.
Modbus+	Green	Indicates communication is active on the Modbus Plus port.

Error Codes

Error Codes Table

The blinking run LED error codes for the NOM module shows the number of times the Run LED on the NOM module blinks for each type of error and the crash codes for each (all codes are in hex).

The following table shows the blinking run LED error codes for the NOM module.

Number of Blinks	Code	Error
Steady	014H	normal power down event
2	815	RAM sequence error
3	49H	illegal data command received by bypass code
	4BH	diagnostics test pattern invalid in the icb block
	4CH	diagnostics test pattern invalid in the page 0
	4DH	icb address not the same as found in hcb
	4EH	bad code selected for mstrout_sel proc
	52H	config table exec_id is different than the sys table exec_id
	53H	got a pupinit hook for neither S985 nor S975 addr
	56H	did not get bus ack form 984 interface within 400 ms
	59H	unexpected modbus port state in send command to 680 proc
	5AH	system table missing
	5BH	bad DPM critical byte write
4	616H	bad or unexpected interrupt
	617H	loopback error on modbus port 1
	618H	parity error
	619H	set port greater than 21
	61AH	controller ram size is less than 8k
	621H	modbus cmd-buffer overflow
	622H	modbus cmd-length is zero
	623H	modbus abort command error
	624H	bad modbus state trn-int
	625H	bad modbus state rcv-int
	626H	bad comm state trn_asc
	627H	transmit underflow error
	628H	bad comm state trn_tru
	629H	bad comm state rcv_asc
	62AH	bad comm state rcv_rtu
	62BH	bad transmit comm state
	62CH	bad receive comm state

	62DH	bad modbus state tmr0_evt
	62EH	bad uart interrupt
	631H	UPI timeout error
	632H	bad UPI response opcode
	633H	UPI bus diagnostic error
	634H	mbp bus interference error
	635H	bad mbp response opcode
	636H	timeout waiting for mbp
	637H	mbp out of synchronization
	638H	mbp invalid path
	639H	peer did not respond with complement of the opcode
	63AH	peer unable to come out of transitions at powerup
	681H	bad master state
	682H	bad slave state
	683H	unknown routing failure to send
	684H	bad port number in set () proc
	685H	bad port number in reset () proc
	686H	bad port number in getport () proc
	687H	bad port number in bitpos () proc
	688H	bad port number in enable_transmit_interrupt () proc
	689H	bad port number in enable_receive_interrupt () proc
	68AH	bad port number in disable_transmit_interrupt () proc
	68BH	bad port number in
	691H	privilege flag is not reset in the session timeout proc
	692H	bad port number in chkmst_hdw () proc
	6A1H	unknown controller type in reset busy flag
	6A2H	unknown function code in generate_poll_cmd () proc
	6A3H	unknown function code in generate_logout_msg () proc
	6A4H	slave link timeout on port other than port #9
	6A5H	illegal bypass command received by bypass code
5	513H	RAM address test error
6	412H	RAM data test error
7	311H	PROM checksum error

Specifications

General Specifications

General Specifications

Power Dissipation	4 W
Bus Current required	750 mA (max.)

Communication Ports

Communication Ports

1 Modbus Plus network (RS-485) port (9-pin connector)	
1 Modbus (RS-232) serial port (9-pin connector)	A bridge mode capability in the module permits a panel device connected to this port to access nodes on the Modbus Plus network or to access the local PLC directly without having to go out onto the network.

Diagnostics

Diagnostics

Power Up	RAM RAM Address Executive Checksum Processor
Runtime	RAM RAM Address Executive Checksum Processor

12.2 140 NOM 212 00: Modbus Plus Option Module

Overview

Introduction

This section describes the 140 NOM 212 00 Modbus Plus option module.

What's in this Section?

This section contains the following topics:

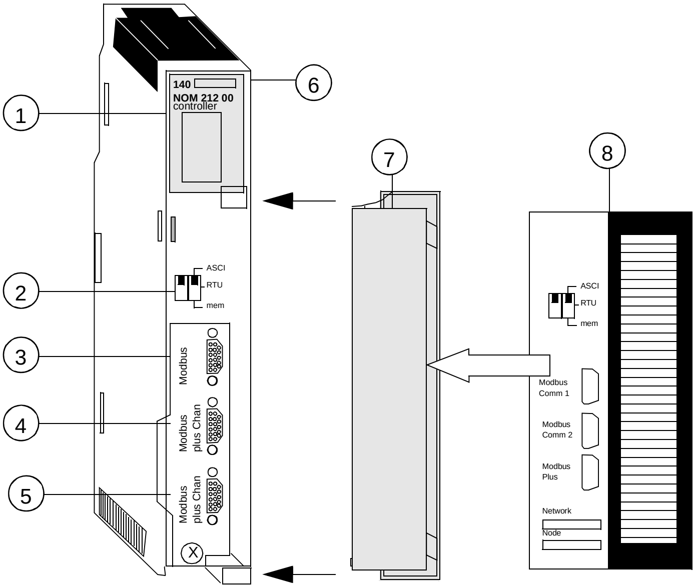
Presentation

Function

The 140 NOM 212 00 is a dual channel Network Option Modul (NOM), connected via a twisted pair Modbus Plus cable network

Illustration

The following figure shows the parts of the Modbus Plus 140 NOM 212 00 modules.



1 LED Area

2 Comm Parameter Slide Switch

3 Modbus Connector

4 Modbus Plus Connector (Chan A)

5 Modbus Plus Connector (Chan B)

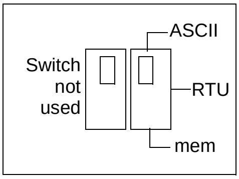
6 Model Number, Module Description, Color Code

7 Removable door

8 Customer Identification Label, (Fold label and place it inside door)

Front Panel Switches

Two, three-position slide switches are located on the front of the unit. The switch on the left is not used. The three-position slide switch on the right is used to select the comm parameter settings for the Modbus (RS-232) port provided with the Modbus Plus option module. Three options are available, as shown below.
The following figure shows the front panel switches.



Note: The NOM hardware defaults to bridge mode when the front panel switch is set to RTU or ASCII mode. When networking controllers, a panel device connected to the NOM Modbus port can communicate with the controller to which it is conected, as well as log into any nodes on the Modbus Plus network.

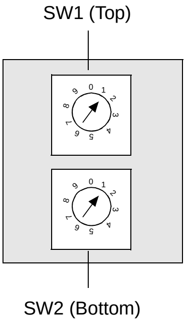
Rear Panel Switches

Two rotary switches are located on the rear panel of the modules. They are used together to set the Modbus Plus node and Modbus port address for the unit.

Note: The highest address that may be set with these switches is 64.
Rotary SW1 (top switch) sets the upper digit (tens), and rotary SW2 (bottom switch) sets the lower digit (ones) of the Modbus Plus node address. The illustration below shows the setting for an example address of 11.

SW1 and SW2 Switches Figure

The following figure shows the SW1 and SW2 switches.



Note: If "0," or an address greater than 64 is selected, the Modbus + LED will be "on" steady, to indicate the selection of an invalid address.

SW1 and SW2 Address Settings

The following table shows the address settings for the SW1 and SW2 switches.

Node Address	SW1	SW2
1 ... 9	0	1 ... 9
10 ... 19	1	0 ... 9
20 ... 29	2	0 ... 9
30 ... 39	3	0 ... 9
40 ... 49	4	0 ... 9
50 ... 59	5	0 ... 9
60 ... 64	6	1 ... 4

Note: If "0," or an address greater than 64 is selected, the Modbus + LED will be "on" steady, to indicate the selection of an invalid address.

ASCII Comm Port Parameters

The following table shows the fixed setting of the ASCII comm port parameters.

Baud	2,400
Parity	Even
Data Bits	7
Stop Bits	1
Device Address	Rear panel rotary switch setting

Setting the slide switch to the middle position assigns remote terminal unit (RTU) functionality to the port; the following comm parameters are set and cannot be changed:

RTU Comm Port Parameters

The following table shows the RTU comm port parameters.

Baud	9,600
Parity	Even
Data Bits	8
Stop Bits	1
Device Address	Rear panel rotary switch setting

Setting the slide switch to the bottom position gives you the ability to assign comm parameters to the port in software; the following parameters are valid.

Valid Comm Port Parameters

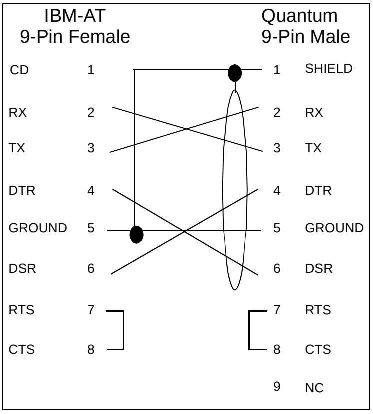
The following table shows the valid comm port parameters.

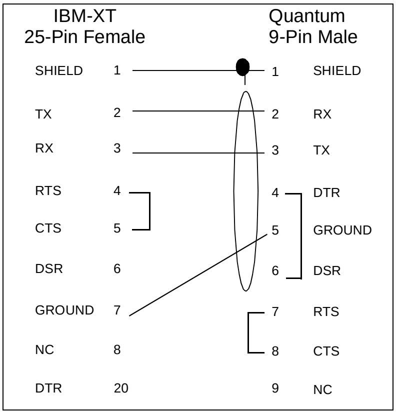
Baud	19,200	1,200
	9,600	600
	7,200	300
	4,800	150
	3,600	134.5
	2,400	110
	2,000	75
	1,800	50
Data Bits	7 / 8	
Stop Bits	1 / 2	
Parity	Enable/Disable Odd/Even	
Device Address	Rear panel rotary switch setting	

Modbus Connector Pinouts

The NOM modules are equipped with a nine-pin RS-232C connector that supports Modicon's proprietary Modbus communication protocol. The following is the Modbus port pinout connections for 9-pin and 25-pin connections.

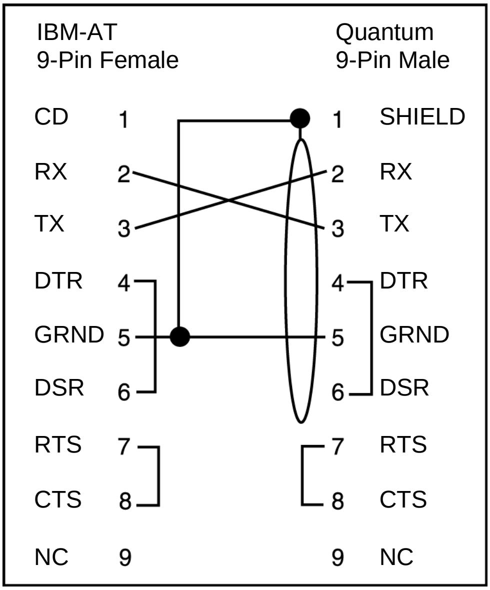
The following figures show the Modbus port pinout connections for 9-pin (left) and 25-pin (right).





Modbus Ports Pinout Connections for Portable Computers

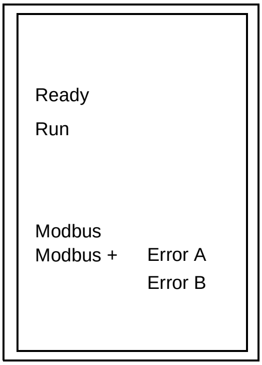
The following figure shows the Modbus port pinout connections for 9-pin portable computers.



Indicators

Illustration

The following figure shows the Modbus Plus NOM LED indicators.



Description

The following table shows the Modbus Plus NOM LED Descriptions.

LEDs	Color	Indication when On
Ready	Green	The module has passed powerup diagnostics.
Run	Green	Indicates that the unit is in kernel mode–should always be OFF during normal operations.
Modbus	Green	Indicates communication is active on the single RS-232 serial port.
Modbus+	Green	Indicates communication is active on the Modbus Plus port.
Error A	Red	There is an error condition on Cable A
Error B	Red	There is an error condition on Cable B

Error Codes

Error Codes Table

The blinking run LED error codes for the NOM module shows the number of times the Run LED on the NOM module blinks for each type of error and the crash codes for each (all codes are in hex).

The following table shows the blinking run LED error codes for the NOM module.

Number of Blinks	Code	Error
Steady	014H	normal power down event
2	815	RAM sequence error
3	49H	illegal data command received by bypass code
	4BH	diagnostics test pattern invalid in the icb block
	4CH	diagnostics test pattern invalid in the page 0
	4DH	icb address not the same as found in hcb
	4EH	bad code selected for mstrout_sel proc
	52H	config table exec_id is different than the sys table exec_id
	53H	got a pupinit hook for neither S985 nor S975 addr
	56H	did not get bus ack form 984 interface within 400 ms
	59H	unexpected modbus port state in send command to 680 proc
	5AH	system table missing
	5BH	bad DPM critical byte write
4	616H	bad or unexpected interrupt
	617H	loopback error on modbus port 1
	618H	parity error
	619H	set port greater than 21
	61AH	controller ram size is less than 8k
	621H	modbus cmd-buffer overflow
	622H	modbus cmd-length is zero
	623H	modbus abort command error
	624H	bad modbus state trn-int
	625H	bad modbus state rcv-int
	626H	bad comm state trn_asc
	627H	transmit underflow error
	628H	bad comm state trn_tru
	629H	bad comm state rcv_asc
	62AH	bad comm state rcv_rtu
	62BH	bad transmit comm state
	62CH	bad receive comm state

	62DH	bad modbus state tmr0_evt
	62EH	bad uart interrupt
	631H	UPI timeout error
	632H	bad UPI response opcode
	633H	UPI bus diagnostic error
	634H	mbp bus interference error
	635H	bad mbp response opcode
	636H	timeout waiting for mbp
	637H	mbp out of synchronization
	638H	mbp invalid path
	639H	peer did not respond with complement of the opcode
	63AH	peer unable to come out of transitions at powerup
	681H	bad master state
	682H	bad slave state
	683H	unknown routing failure to send
	684H	bad port number in set () proc
	685H	bad port number in reset () proc
	686H	bad port number in getport () proc
	687H	bad port number in bitpos () proc
	688H	bad port number in enable_transmit_interrupt () proc
	689H	bad port number in enable_receive_interrupt () proc
	68AH	bad port number in disable_transmit_interrupt () proc
	68BH	bad port number in
	691H	privilege flag is not reset in the session timeout proc
	692H	bad port number in chkmst_hdw () proc
	6A1H	unknown controller type in reset busy flag
	6A2H	unknown function code in generate_poll_cmd () proc
	6A3H	unknown function code in generate_logout_msg () proc
	6A4H	slave link timeout on port other than port #9
	6A5H	illegal bypass command received by bypass code
5	513H	RAM address test error
6	412H	RAM data test error
7	311H	PROM checksum error

Specifications

General Specifications

General Specifications

Power Dissipation	4 W (typical)
Bus Current required	780 mA

Communication Ports

Communication Ports

2 Modbus Plus network (RS-485) port (9-pin connector)	For dual connectivity on a single Modbus Plus network. These ports handle identical versions of all inbound and outbound transactions and keep track of the data paths used for these transactions
1 Modbus (RS-232) serial port (9-pin connector)	A bridge mode capability in the module permits a panel device connected to this port to access nodes on the Modbus Plus network or to access the local PLC directly without having to go out onto the network.

Diagnostics

Diagnostics

Power Up	RAM RAM Address Executive Checksum Processor
Runtime	RAM RAM Address Executive Checksum Processor

12.3 140 NOM 252 00: Modbus Plus Option Module

Overview

Introduction

This section describes the 140 NOM 252 00 Modbus Plus option module.

What's in this Section?

This section contains the following topics:

Presentation

Overview

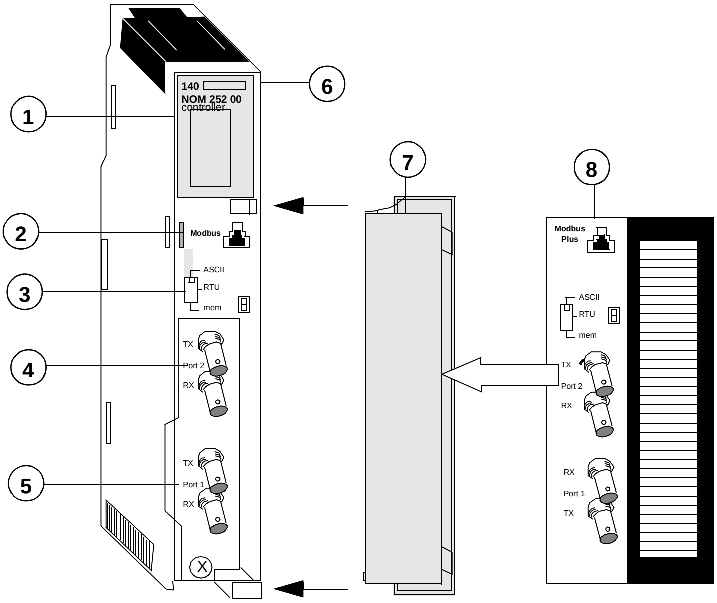
The Modbus Plus on Fiber module provides connectivity to Modbus Plus nodes by fiber cable.

There are many benefits that result from the use of fiber optics. Some of these benefits include:

- Longer distances between nodes (up to 3 km), thereby, increasing the total length of the network.
- Fiber optic medium is not susceptible to the effects of electromagnetic interference, RF interference, and lightning.
- Intrinsically safe links that are required in many hazardous industrial environments.
- Total electrical isolation between terminal points on the link

Illustration

The following figure shows the parts of the Modbus Plus 140 NOM 252 00 module.



1 LED Area

2 Modbus Connector

3 Comm Parameter Slide Switch

4 Port 2 TX and RX Connectors

5 Port 1 TX and RX Connectors

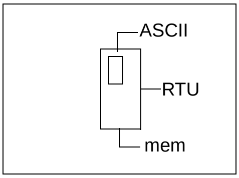
6 Model Number, Module Description, Color Code

7 Removable door

8 Customer Identification Label, (Fold label and place it inside door)

Front Panel Switch

A three-position slide switch is located on the front of the unit. This switch is used to select the comm parameter settings for the Modbus (RS-232) port. Three options are available, as shown below.
The following figure shows the front panel switch.



Setting the slide switch to the top position assigns ASCII functionality to the port; the following comm parameters are set and cannot be changed.

ASCII Comm Port Parameters

The following table shows the fixed setting of the ASCII comm port parameters.

Baud	2,400
Parity	Even
Data Bits	7
Stop Bits	1
Device Address	Rear panel rotary switch setting

Setting the slide switch to the middle position assigns remote terminal unit (RTU) functionality to the port; the following comm parameters are set and cannot be changed:

RTU Comm Port Parameters

The following table shows the RTU comm port parameters.

Baud	9,600
Parity	Even
Data Bits	8
Stop Bits	1
Device Address	Rear panel rotary switch setting

Setting the slide switch to the bottom position gives you the ability to assign comm parameters to the port in software; the following parameters are valid.

Valid Comm Port Parameters

The following table shows the valid comm port parameters.

Baud	19,200	1,200
	9,600	600
	7,200	300
	4,800	150
	3,600	134.5
	2,400	110
	2,000	75
	1,800	50
Data Bits	7 / 8	
Stop Bits	1 / 2	
Parity	Enable/Disable Odd/Even	
Device Address	Rear panel rotary switch setting	

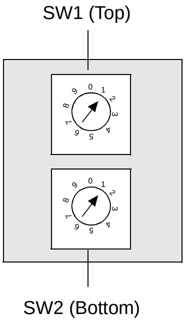
Rear Panel Switches

Two rotary switches are located on the rear panel of the modules. They are used together to set the Modbus Plus node and Modbus port address for the unit.

Note: The highest address that may be set with these switches is 64.
Rotary SW1 (top switch) sets the upper digit (tens), and rotary SW2 (bottom switch) sets the lower digit (ones) of the Modbus Plus node address. The illustration below shows the setting for an example address of 11.

SW1 and SW2 Switches Figure

The following figure shows the SW1 (top) and SW2 (bottom) switches.



SW1 and SW2 Address Settings

The following figure shows the node address settings for the SW1 and SW2 switches.

Node Address	SW1	SW2
1 ... 9	0	1 ... 9
10 ... 19	1	0 ... 9
20 ... 29	2	0 ... 9
30 ... 39	3	0 ... 9
40 ... 49	4	0 ... 9
50 ... 59	5	0 ... 9
60 ... 64	6	1 ... 4

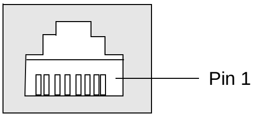
Note: If "0" or an address greater than 64 is selected, the Modbus + LED will be "on" steady, to indicate the selection of an invalid address.

Modbus Connector

The NOM 252 00 module is equipped with an RS-232 port (see below) located on the front of the module. This port uses an eight-position RJ45 (phone jack-type) connector.

Modbus pin 1 Figure

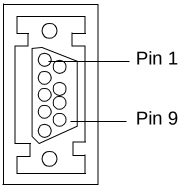
The following figure shows the NOM 252 00 Pin 1 connector.



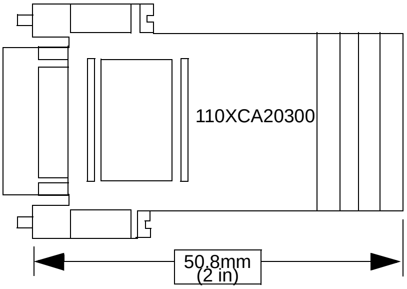
Note: A D-shell adapter is available from Modicon for NOM 252 00-to-computer connections: a (110 XCA 20 300) 9-pin adapter for PC-AT type computers (see the illustration pinout table below).

Pinouts Figures

The following figures show the 9-pin adapter front view (left) and side view (right).



Front View

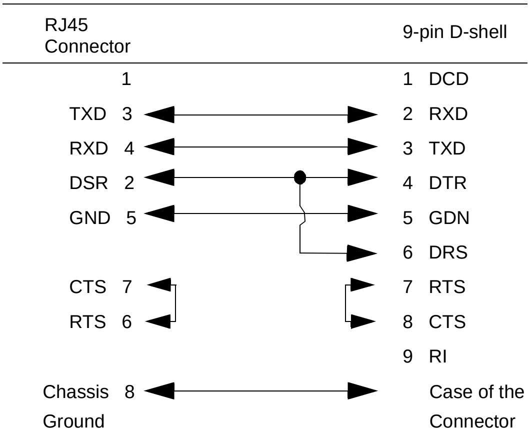


Side View

Connector Pinouts Figure

The following figure shows the 9-pin RJ45 connector schematic.

Connectors Pinouts

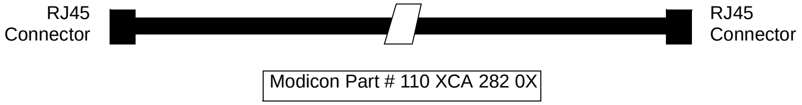


BJ45 Cable Types

This following shows an example of the 110 XCA 282 0X cable. A table is also provided which includes part numbers and cable lengths.

RJ45 Connector Figure

The following figure shows the RJ45 connector (Modicon Part # 110 XCA 282 OX).



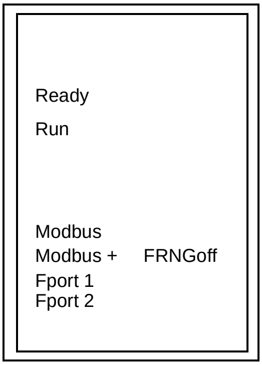
BJ45 Cable Part Numbers Table

Cable Part Numbers	Cable Lengths
110 XCA 282 01	3 ft. (0.91 m)
110 XCA 282 02	10 ft. (3 m)
110 XCA 282 03	20 ft. (6 m)

Indicators

Illustration

The following figure shows the Modbus Plus on Fiber LED indicators.



Description

The following table shows the Modbus Plus on fiber LED descriptions.

LEDs	Color	Indication when On
Ready	Green	The module has passed powerup diagnostics.
Run	Green	Indicates that the unit is in kernel mode – should always be OFF during normal operations. Note: The table for the NOM 21X 00 shows the number of times the RUN LED on the Modbus Plus on Fiber Module blinks for each type of error and the crash codes for each (all codes are in hex).
Modbus	Green	Indicates communication is active on the single RS-232 serial port.
Modbus+	Green	Indicates communication is active on the Modbus Plus port.
Fport1	Green	Indicates an optical signal has been received on fiber optic Port 1.
Fport2	Green	Indicates an optical signal has been received on fiber optic Port 2.
FRNGoff	Red	Indicates the first break in a self healing ring.

Fiber Optic Cable Connections

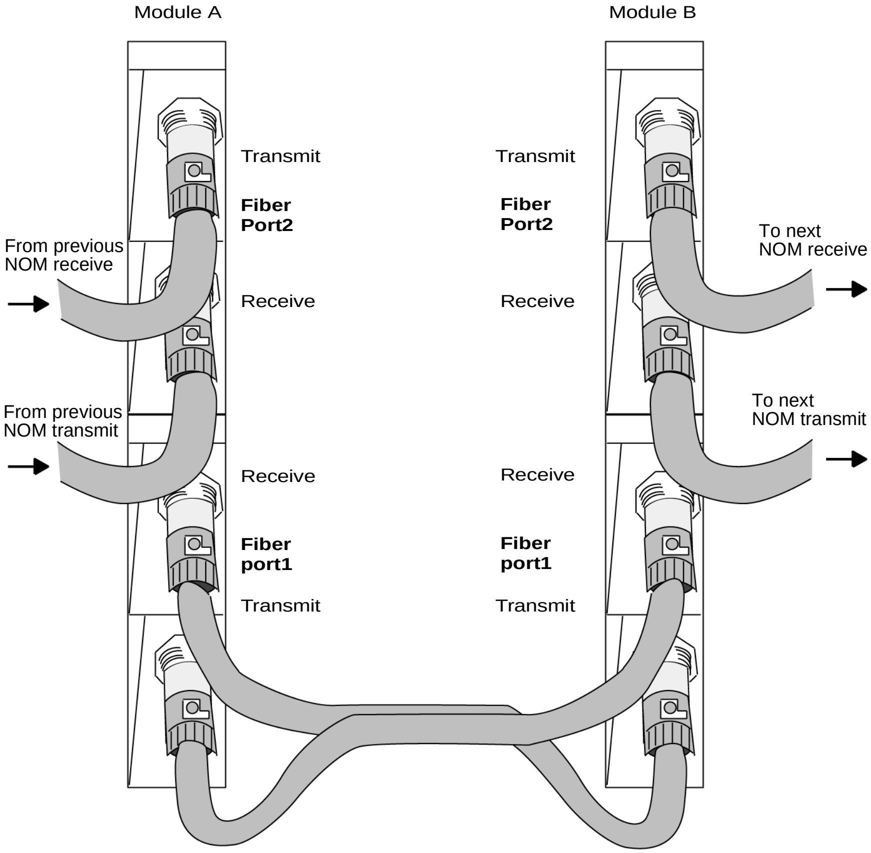
Fiber Optic Cable Connections

The NOM 252 00 module is connected in the Quantum system by a fiber optic cable (see below). The cable has two strands. Each module transmits a uni-directional signal. For this reason, each strand must be connected to the transmit port on one module and the receive port on the other.

One strand of the fiber optic cable is marked at 10-inch (25 cm) intervals with the manufacturer’s name and the cable specifications. This is the only way to distinguish the two strands.

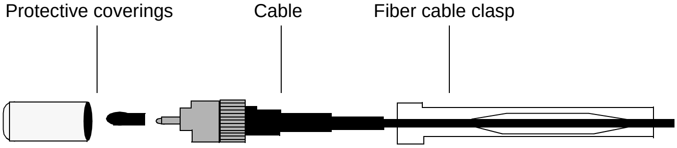
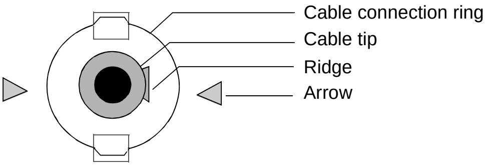
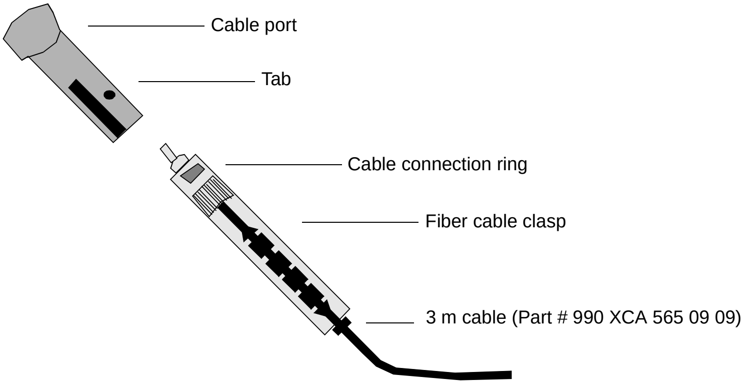
Fiber Optic Cable Connections Figure

The following figure shows the fiber optic cable connections.



Connecting the Fiber Optic Cable

The following steps show how to connect the fiber optic cable.

Step	Action
1	Remove the protective plastic coverings from the cable ports and the tips of the cable. Snap one of the fiber cable clasps (shipped with the module) over the cable so that the wider end of the tool is closest to the cable end. 
2	Turn the connection ring so that one of the arrows on the side of the ring lines up with the ridge inside. 
3	a. Slide the tool up to the connection ring. b. Gripping the cable with the plastic cable clasp, slide the cable end onto the lower cable port. The arrow and the ridge on the connection ring should line up with the slot on the left of the cable port. c. Use the clasp to push the cable over the tab on top of the port. d. Turn the cable to the right, so that the tab locks securely e. Remove the clasp. f. Repeat this process with the remaining strand of cable.  Cable port Tab Cable connection ring Fiber cable clasp 3 m cable (Part # 990 XCA 565 09 09)

Fiber Optic Configurations

Here are four typical configurations that show the wide range of the network architecture:

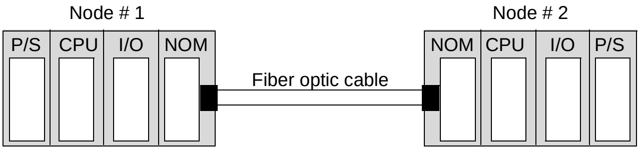
- Point-to-point connection
- Bus configuration
- Tree configuration
- Self-healing ring configuration

Point-to-Point Configuration

This type of configuration (see below) allows communication over the distance of up to 3 km through harsh industrial environments.

Point-to-Point Configuration Example Figure

The following figure shows the point-to-point configuration.



Bus Configuration

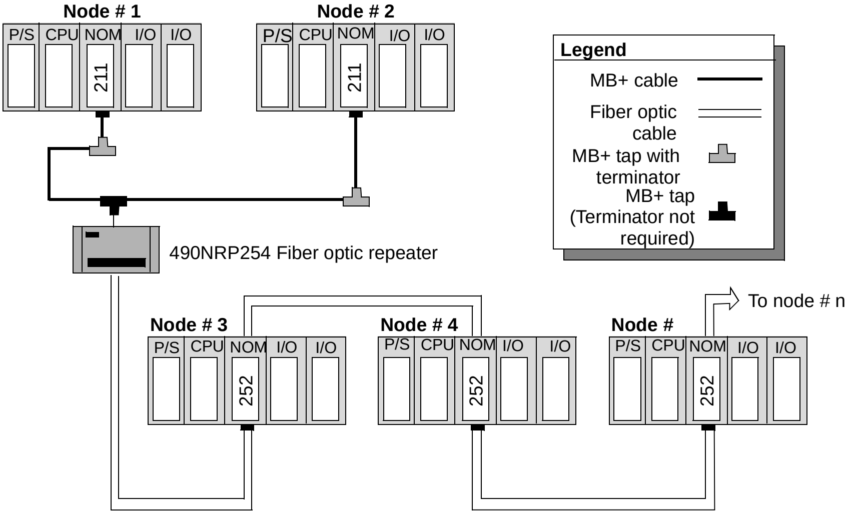
This type of configuration is used when it is required to connect a number of fiber nodes and can be used to increase the distance of a standard Modbus Plus network by changing to a fiber medium. This kind of network allows the connection of up to 32 Quantum NOM 252 nodes over the distance of 5 km.

The following illustrations show the NOM 252 00 module in a mixed fiber optic/ twisted pairs bus configuration network and a straight fiber optic bus configuration network.

	CAUTION
⚠	**Equipment Failure** The loss of a single node in this configuration disables the rest of the network. **Failure to follow this precaution can result in injury or equipment damage.**

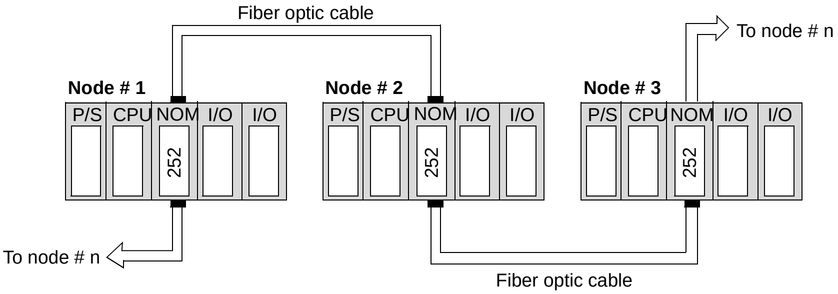
Bus Configuration Example 1

The following figure shows the mixed fiber optic/copper network.



Bus Configuration Example 2

The following figure shows the straight fiber optic network.



Note: The distance between nodes on fiber is limited by the maximum allowable power loss from end-to-end (3 km over 62.5 mm fiber). Power loss includes the fiber optic cable attenuation, connector losses at the Fiber Optic Receiver and Transmitter ports, and the system margin of 3 dB.

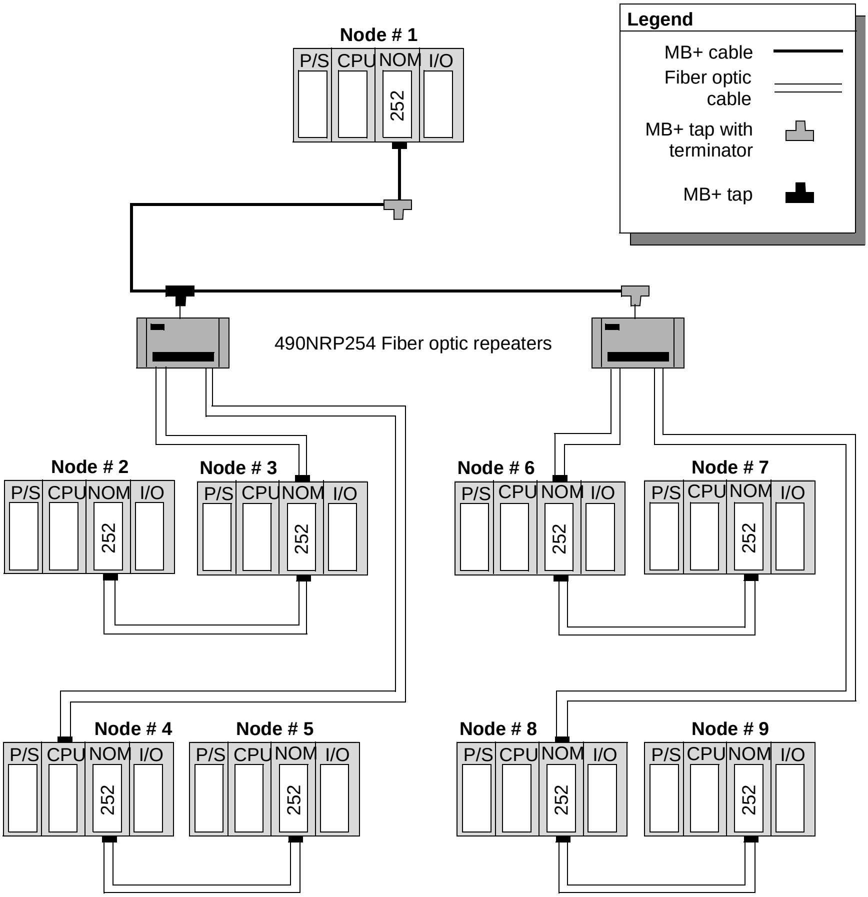
In this configuration, the end NOM 252 00 in this configuration will have the FRNGoff LED active. It also displays the Cable B Framing error in the MBPSTAT (in ladder logic).

Tree Configuration

Using tree configurations allows for greater flexibility in the layout of Modbus Plus and NOM 252 00 networks. The following illustrations are samples tree configurations. Additional repeaters may be connected in order to extend communication between electrical links.

Tree Configuration Example

The following figure shows the tree configuration.

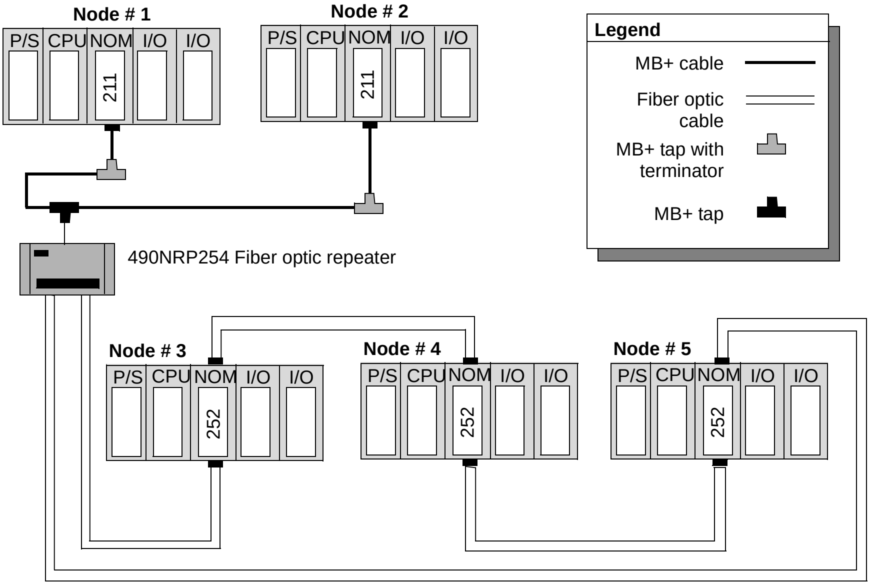


Self-healing Ring Configuration

This configuration can be achieved by connecting the unused fiber optic ports of the first and last NOM 252 00 directly or through the fiber optic repeater, if a mixed fiber optic/twisted pairs network is used. This type of connection has all the advantages of the previously described configurations, along with built-in redundancy. A broken connection between any two Quantum modules in the ring will automatically reconfigure the network to the bus configuration, and maintain communication.

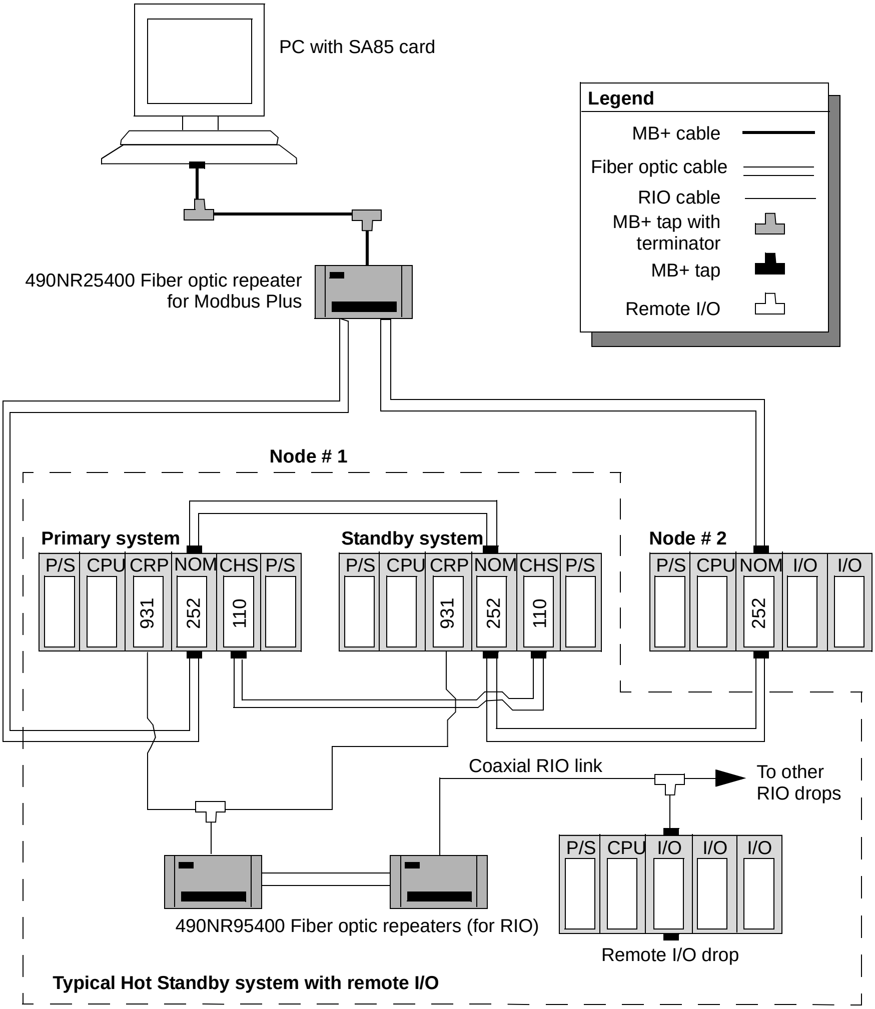
Self-healing Ring Configuration Example

The following figure shows a self-healing ring configuration example.



Hot Standby Systems Figure

The following figure shows the self-healing ring configuration for hot standby systems.



Network Status

The information about the condition of the network is presented in the form of Network Status. This information indicates the loss of connection (the first break in the self-healing ring) and is similar to the way the existing 140 NOM 212 00 reports the loss of the redundant cable.

The break in the fiber cable will be detected by the module not receiving the signal from the cable break side. The incident will be reported by MBPSTAT as a Cable B Framing error. This condition also activates the FRNGoff LED on the module front.

Recommended Materials for Fiber Optic Links

Modicon does not manufacture fiber optic products such as cables, connectors, or special tools. However, we have experience with third party suppliers of materials, and are able to provide guidelines on product compatibility.

Connectors

The following table shows the connector types

Connector type	Part number	Operating temperature
ST bayonet (epoxy)	3M 6105	-40 ... +80 °C
ST bayonet (hot melt)	3M 6100	-40 ... +60 °C
ST bayonet (epoxy)	AMP 501380-5 series	-30 ... +70 °C
ST bayonet (epoxy)	AMP 503415-1 series	-20 ... +75 °C
Light crimp ST-style	AMP 503453-1 series	-20 ... + 60 °C
Mechanical line splice (one size fits all)	3M 2529 Fiberlok1 II	-40 ... +80 °C

Note: All connectors must have a short boot for strain relief.

Termination Kits

The following table shows the termination kits.

Kit type	Part number	Description
Bayonet ST (eoxy)	AMP 503746-1	For all epoxy type ST style
Light crimp XTC	AMP 50330-2	For all light crimp
Mechanical line splice	3M 2530	Fiber splice prep kit, complete with cleaving tool
3M hot melt	3M 05-00185 3M 05-00187	110 V termination kit 220 V termination kit

Other Tools Table

The following table shows other tools that may be needed for fiber optic links.

Product	Part number	Description/use
3M (Photodyne) optical source driver	9XT	Hand-held optical source driver (requires a light source)
3M (Photodyne) optical light source	1700-0850-T	850 nm Light Source, ST Connectors for 9XT
3M (Photodyne) power meter	17XTA-2041	Hand-held fiber optic power meter
3M optical light source, 660 nm, visible	7XE-0660-J	Use with 9XT to troubleshoot raw fiber, requires FC/ST patch cord
3M FC/ST patch cord	BANAV-FS-0001	Connects FC connector on 7XE to ST
3M bare fiber adapter, ST-compatible	8194	Allows the use of above source and meter to test raw fiber (two required)

Cables

It is recommended that you use 62.5/125 mm cable (such as AMP 503016-1, AMP 502986-1, or equivalent) with a maximum attenuation of 3.5 dB/km in most of the configurations.

Note: Modicon recommends using the 990 XCA 656 09 cable.

Note: All cables must have a maximum cable diameter of not more than 3 mm at the terminal side.

Connections

The following information discusses connecting the NOM 252 00 on fiber cable, adding a new mode to the network, and repairing the break in the cable.

Note: When a new network is assembled, it is recommended that you connect all cables before powering up the system. Connect fiber optic cables as described previously in this section.

Adding a New Node to the Network

If a new node is added to an existing network in order to extend the network (at the end of any configuration), then a new node may be connected first by fiber cable and then hot-swapped to the backplane to avoid errors to the existing network.
If a new node is added to the middle of the network, disconnect the fiber optic cables from one side of the existing NOM 252 module, and connect to port 1 or 2 of the new node. Additional fiber optic cable then needs to be connected to the second port of the new NOM 252 and to the next NOM 252 in the network. Finally, hot-swap the new NOM 252 to the backplane.

Repairing the Break in the Cable

Because the NOM 252 00 will stop transmitting in the direction from which it receives no signal, replaceing a broken fiber optic cable and reconnectioning do not suffice to re-establish communication over that segment. Hot-swapping only one NOM 252 at the repaired connections is required to complete the connection.

Note: Breakage of any fiber connectors or fiber optic cables is the equivalent to breaking the trunk cable in a copper-based Modbus Plus network.

For the self-healing ring configuration, repairing the first break in the fiber optic network has to be scheduled when one of the units on either side of the repaired break can be hot-swapped, without creating further problems by disconnecting the node.

Note: Self-healing configurations are not considered as redundant networks. Redundant networks yield a high system availability.

Calculations

Use the following formula to calculate the number of NOM 252 00 modules in a fiber network:

Step	Action
1	The total allowable pulse width distortions and jitter are limited to 20% of the bit period and is 200 nsec for the full fiber optic network.
2	The jitter contributed by the NOM 252 is 5 nsec max.
3	Jitter contributed by fiber optic repeaters (if used) is 40 nsec.
4	Use the following formula to determine the number (N) of chained repeaters: $N = \frac{200\text{nsec} - X(L)\text{nsec} - 40\text{nsec}}{5\text{nsec}} + 1$ where "L" is the total cable length (km), and "X" is the jitter (added by the fiber optic cable) in nsec/km: X = 3 ns/km for 50/125 micron meters 5 ns/km for 62.5/125 micron meters 7.5 ns/km for 100/140 micron meters

Specifications

General Specifications

General Specifications

Power Dissipation	4 W (typical)
Bus Current required	780 mA
External Power	Not required

Communication Ports

Communication Ports

Optical Ports	2 (consisting of an optical receiver and transmitter)
Modbus Port	1 RJ45 (phone jack-type) connector

Diagnostics

Diagnostics

Power Up	RAM RAM Address Executive Checksum Processor
Runtime	RAM RAM Address Executive Checksum

Optical Transmission

Optical Transmission

Interface	ST-Type connector
PulseWidth Disstortion and Jitter	5 ns or better
Wavelength	820 nm
Power Loss Budget (includes 3 dB of system margins)	50/125 micron fiber -6.5 dB 62.5/125 micron fiber -11 dB 100/140 micron fiber -16.5 dB
Maximum distance for point-to- point connection	2 km over 50 micron fiber 3 km over 62.5 micron fiber 3 km over 100 micron fiber
Maximum System Length in Self Healing Ring Configuration	10 km over 62.5 micron fiber

Optical Transmitter Specifications

Optical Transmitter Specifications

Optical Power (Measured with 1 m test fiber)	-12.8 ... -19.8 dBm average power in 50/125 micron fiber cable -9.0 ... -16 dBm average power in 62.5/125 micron fiber cable -3.5 ... -10.5 dBm average power in 100/140 micron fiber cable
Rise/Fall Time	20 ns or better
Silence (OFF leakage)	-43 dBm

Optical Receiver Specifications

Optical Receiver Specifications

Receiver Sensitivity	-30 dBm average power
Dynamik range	-20 dB
Detected Silence	-36 dBm

Hardware Installation

13

Mounting Quantum Communication Modules

Overview

Quantum communication modules (NOMs, HE-CPUs) can be inserted into any slot of any backplane. Although Power supply modules should be installed in the first or last slots, to have a cooling effect. The modules can be removed under power (hot swapped) without damaging modules or the backplane.
Refer to the following figures and procedure when mounting modules.

⚠ **CAUTION**

Possible danger to personnel or equipment.

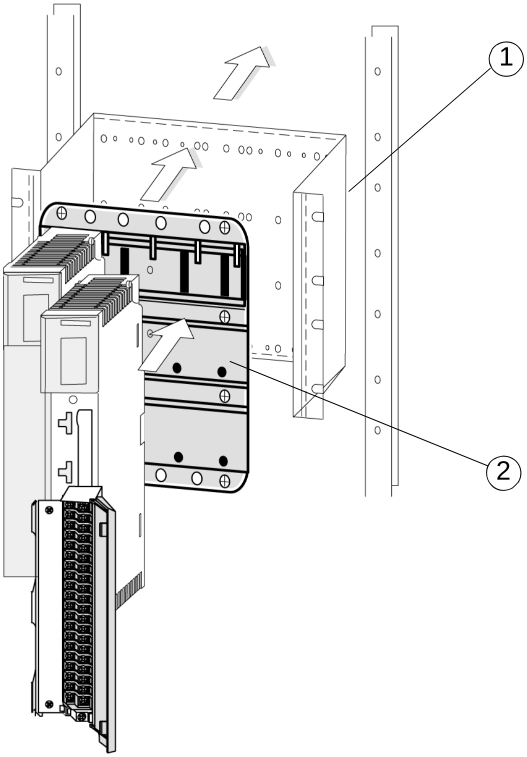
An I/O module can only be hot swapped with the field side terminal strip removed.
It is possible, that a hot swapped controller module stops with an error code.

Failure to follow this precaution can result in injury or equipment damage.

Note: To ensure EMC level , the mounting area of the CPU has to provide metallic contact. Therefore remove any labels in the affected area and clean the surface using solvent.

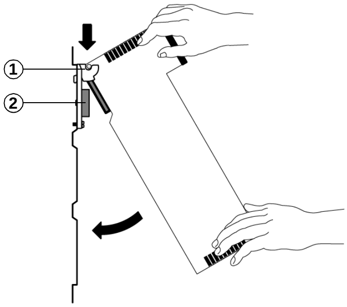
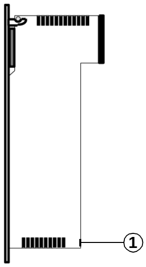
Mounting Bracket and Backplane

The following steps describe the mounting of bracket and backplane

Step	Action
1	If required for the application, select and install a 20 mm or 125 mm mounting bracket to the rack using standard hardware. Front view:  **1** Mounting bracket **2** Backplane
2	Select and install the appropriate backplane to the mounting bracket using standard hardware and remove the plastic backplane connector dust covers.

Mounting a Module

The following steps describe the mounting of a module

Step	Illustration	Action
1	Side view:  **1** Module Hooks **2** I/O Bus Connector	Mount the module at an angle on to the two hooks located near the top of the backplane.
2		Swing the module down to make an electrical connection with the backplane I/O bus connector.
3	Side view:  **1** Mounting screw	Tighten the screw at the bottom of the module to fasten it to the backplane. **Note:**The maximum tightening torque for this screw is 2-4 in-lbs (0.23 - 0.45 Nm).

Index