

170 PNT Series

Modbus Plus Communication Adapters
for TSX Momentum

User Guide

870 USE 103 00

11/97



GROUPE SCHNEIDER

■ Modicon ■ Square D ■ Telemecanique

Data, Illustrations, Alterations

Data and illustrations are not binding. We reserve the right to alter products in line with our policy of continuous product development. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us using the form on one of the last pages of this publication.

Training

Schneider Automation offers suitable further training on the system.

Hotline

See addresses for the Technical Support Centers at the end of this publication.

Trademarks

All terms used in this publication to denote Schneider Automation products are trademarks of Schneider Automation Incorporated.

All other terms used in this publication to denote products may be registered trademarks and/or trademarks of the corresponding corporations.

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation,

Windows is a brandname of Microsoft Corporation in the USA and other countries.

IBM is a registered trademark of International Business Machines Corporation.

Intel is a registered trademark of Intel Corporation.

Copyright

All rights are reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying, processing or by online file transfer, without permission in writing by Schneider Automation Incorporated. You are not authorized to translate this document into any other language.

© 1997 Schneider Automation Incorporated. All rights reserved.

Contents

TSX Momentum

Modbus Plus Communication Adapters

170 PNT 110 20, 170 PNT 160 20	1
1.1 Product Overview	2
1.1.1 Function	2
1.1.2 Physical Structure	2
1.1.3 Network Compatibility	2
1.1.4 Operating Voltages and Error Control	2
1.1.5 Environmental Specification	2
1.2 Network Communication Status Indicators	3
1.3 Setting the Modbus Plus Node Address	4
1.3.1 How Addresses are Assigned	4
1.3.2 Setting the Adapter s Address	5
Communicating With the Adapter	7
2.1 Using Modbus Plus for Distributed I/O Servicing	8
2.1.1 Multi-Purpose Applications	8
2.1.2 Distributed I/O Applications	8
2.2 How Communication Adapters Handle Messages	10
2.2.1 How Messages are Defined in the Application	10
2.2.2 How Messages are Transacted on the Network	10
2.2.3 Mapping Data to I/O Bases	11
2.3 Communication Access Registers	12
2.3.1 Data Registers	13
2.3.2 Configuration Registers	13
2.3.3 Status Registers	14

TSX Momentum Modbus Plus Communication Adapters

170 PNT 110 20
170 PNT 160 20

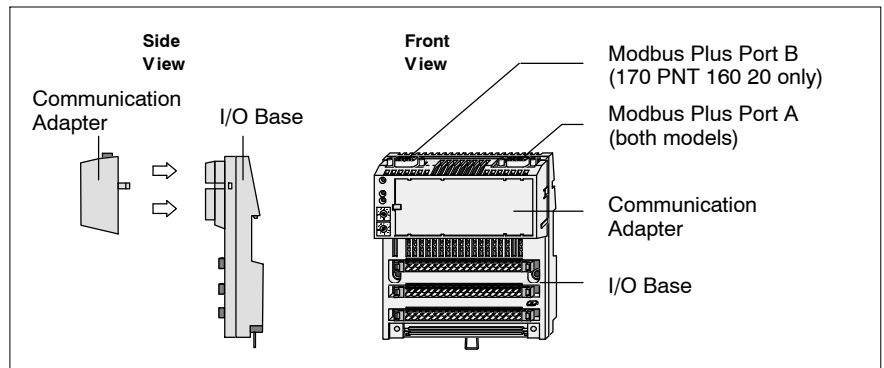
1

These Communication Adapters can be connected to any TSX Momentum I/O base to create a functional I/O module. They provide direct connection to the Modbus Plus network, enabling a programmable controller to communicate with field devices wired to the I/O base terminals. Data bits are transferred in the IEC format.

Model 170 PNT 110 20 provides one Modbus Plus port for single-cable networks. Model 170 PNT 160 20 has two ports for single-cable or dual-cable networks.

Figure 1 shows the layout of a typical adapter and I/O base.

Figure 1 Communication Adapter with TSX Momentum I/O Base



This chapter describes:

- ■ Product Overview
- ■ Network Communication Status Indicators
- ■ Setting the Modbus Plus Node Address

1.1 **Product Overview**

1.1.1 **Function**

These adapters are installed on any TSX Momentum I/O base to form a complete I/O module that communicates on a Modbus Plus network. A programmable controller on the network can then read from the input terminals and write to the output terminals of the I/O base using Modbus Plus Peer Cop or MSTR Function Block messaging. Data bits are transmitted and received in the IEC format.

For information about the application and field wiring of I/O bases, refer to the *TSX Momentum I/O Bases User Manual*, part number 870 USE 002 00.

1.1.2 **Physical Structure**

Each adapter connects to the internal communication connector of its I/O base. Clips lock the adapter in place. The clips can be released with a common screwdriver to remove the adapter.

Front panel labels are supplied with each I/O base. The user can fill out the label to identify the field wiring connections and application of the I/O base terminals. The label can then be affixed to the front panel of the adapter.

1.1.3 **Network Compatibility**

Model 170 PNT 110 20 has one Modbus Plus port for connection to a network with a single trunk cable. Model 170 PNT 160 20 has two ports for connection to either a single-cable or dual-cable network.

Network port connections are compatible with standard Modbus Plus drop cables. Drop cables are available from Schneider Automation in three standard lengths: 2.4 m (8 ft), 3 m (10 ft), and 6 m (20 ft).

1.1.4 **Operating Voltages and Error Control**

The adapter receives its operating voltage from the I/O base through its internal connection to the base. The adapter monitors its voltage and goes offline to the Modbus Plus network if the voltage is not within tolerance.

1.1.5 **Environmental Specification**

The adapter conforms to the environmental specification for the I/O base upon which it is mounted. For further information refer to the *TSX Momentum I/O Bases User Manual*, part number 870 USE 002 00.

1.2 Network Communication Status Indicators

Each model has a front panel indicator showing its network communication status. The dual-cable model has two additional indicators which identify communication errors on the two cable paths.

Figure 2 Communication Status Indicators

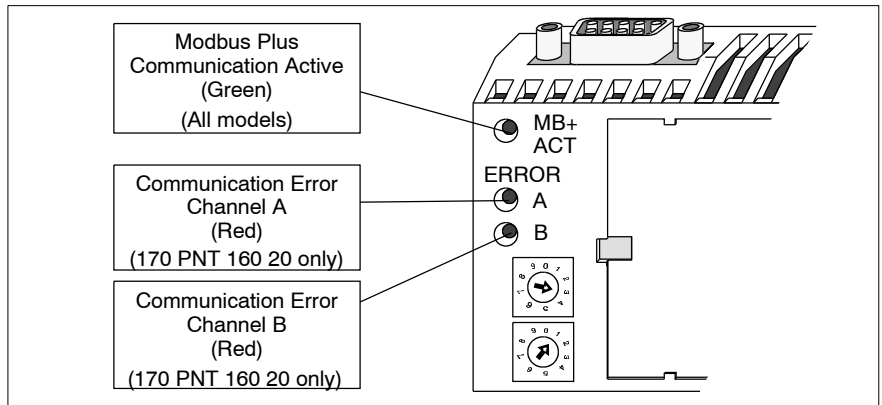


Table 1 Modbus Plus Active Indicator Patterns

Indicator Pattern (Green)	Status
Six flashes/second	Normal operating state. All nodes on a healthy network flash this pattern.
One flash/second	The node is off-line. After being in this state for 5 seconds, the node attempts to go to its normal operating state.
Two flashes, then OFF for 2 seconds	The node detects the network token being passed among other nodes, but it never receives the token.
Three flashes, then OFF for 1.7 seconds	The node does not detect any token passing on the network.
Four flashes, then OFF for 1.4 seconds	The node has detected another node using the same address.

Table 2 Modbus Plus Channel Error Indicators (170 PNT 160 20 only)

Indicator (Red)	Status
Channel A Error	Communications error at network port A.
Channel B Error	Communications error at network port B.

1.3 Setting the Modbus Plus Node Address

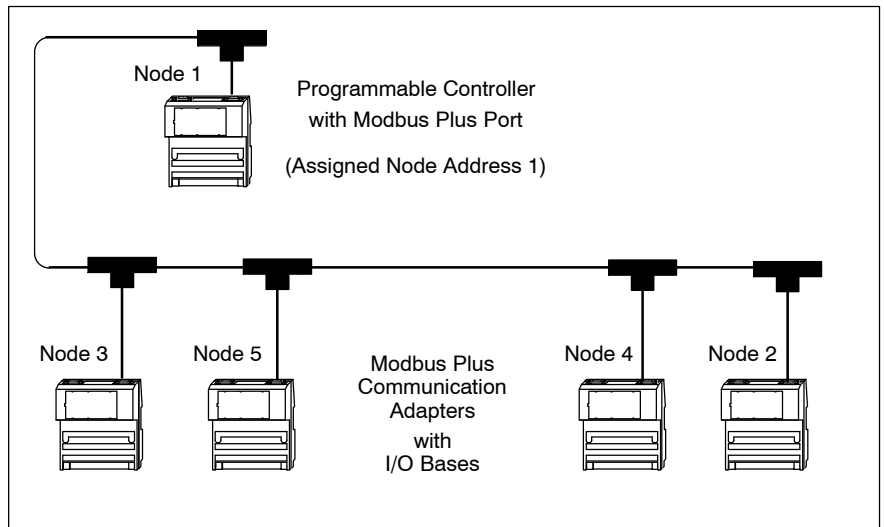
1.3.1 How Addresses are Assigned

Modbus Plus nodes are identified by addresses assigned to them by the user. Each node must have a unique address in the range 1 ... 64. Duplicate addresses are not allowed.

Starting at address 1, the lowest addresses should be assigned to programmable controllers. Communication adapters should be assigned the next addresses in direct sequence. Addresses are assigned logically and are not dependent upon the physical locations of the node devices.

Figure 3 shows typical address assignments for a network with one controller and four communication adapters.

Figure 3 Example of Node Address Assignment

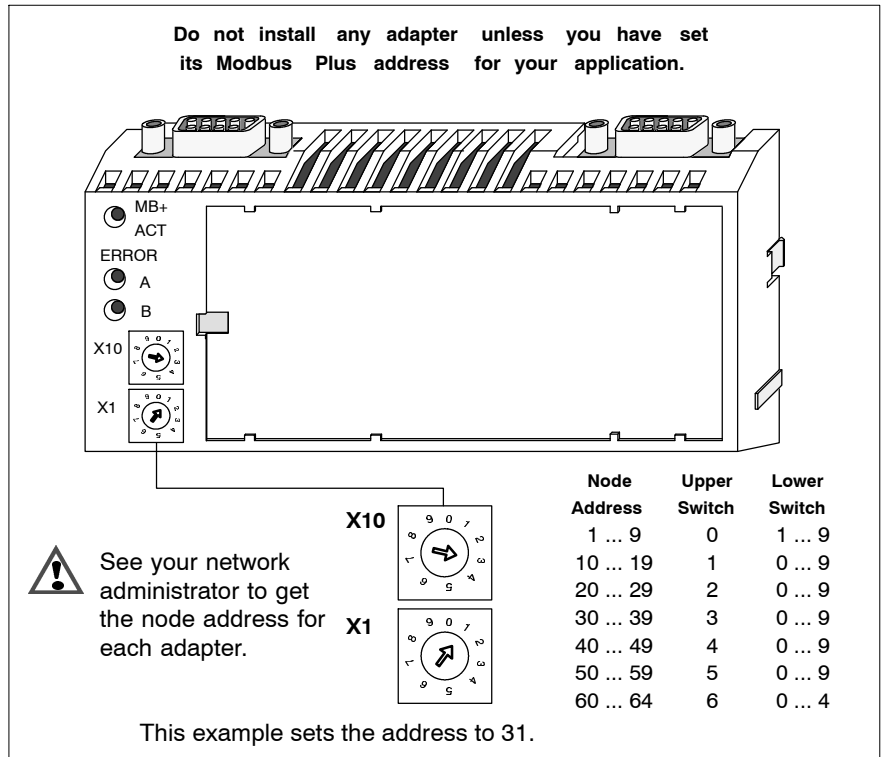


1.3.2 Setting the Adapter's Address

The communication adapter's address is assigned in two locations: in rotary switches on the adapter's front panel, and in the Peer Cop table and MSTR function blocks of the user's application program.

The address in the adapter's front panel switches must correspond to the address defined for that adapter in the application program. This is required to ensure that messages will be sent and received properly at the intended nodes across the network.

Figure 4 Setting the Modbus Plus Node Address



Communicating With the Adapter

2

-
- ■ Using Modbus Plus for Distributed I/O Servicing
 - ■ How Communication Adapters Handle Messages
 - ■ Communication Access Registers

2.1 **Using Modbus Plus for Distributed I/O Servicing**

Modbus Plus networks can be used to service multi-purpose control applications, or they can be organized for the most efficient servicing of distributed I/O devices. Both approaches are outlined below.

2.1.1 **Multi-Purpose Applications**

Modbus Plus networks can be used for applications in which programmable controllers, operator interfaces, and other kinds of devices must communicate. Message timing is determined by the amount of time each node holds its token. Timing can vary according to the current processing requirements of each node's internal program. For that reason, multi-purpose networks are not recommended for servicing I/O control applications in which I/O timing must be deterministic.

In general applications, up to five networks can be joined by Bridge Plus devices to extend the cable length to 2250 m (7500 ft) and the node count to 320 nodes.

2.1.2 **Distributed I/O Applications**

Modbus Plus networks can be designed to efficiently service I/O field devices. In such applications the timing of message transactions must be predictable to allow deterministic timing of the I/O control process. To ensure this, the network should consist of just one programmable controller node and the required group of I/O nodes. Non-I/O devices, such as additional controllers, programmers, or operator interfaces, should communicate with the I/O network controller through a separate Modbus Plus network or other type of connection.

In distributed I/O applications, messages are transacted on the local network only. Bridge Plus devices are not applicable to networks used for distributed I/O.

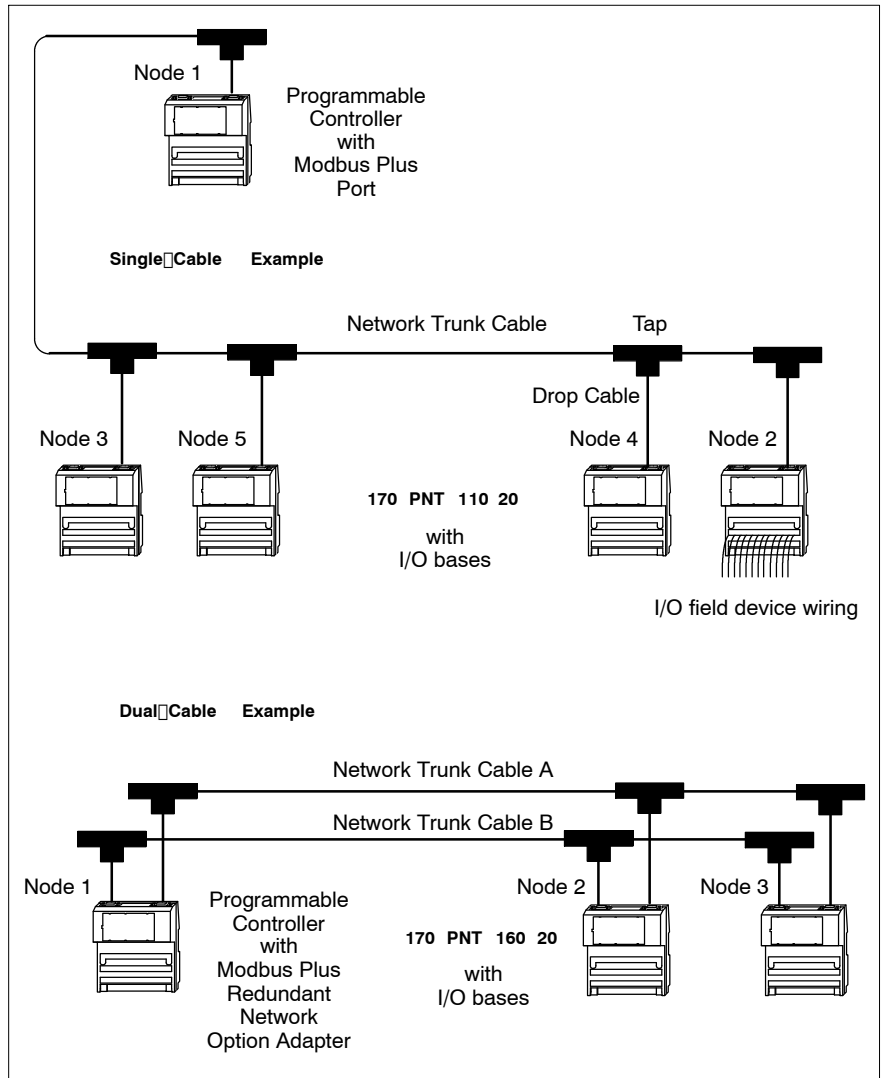
Table 3 summarizes the Modbus Plus network's maximum configuration for a distributed I/O application consisting of Momentum products.

Table 3 Maximum Distributed I/O Configuration (Momentum Nodes Only)

Parameter	Specification
max. number of nodes	64 including Controller
max. distance between two nodes	450 m (1500 ft)
min. distance between two nodes	3 m (10 ft)
max. length of network	450 m (1500 ft)
max. number of data words (16-bit words)	500 input, 500 output
max. number of I/O points (16 bits/word)	8000 input, 8000 output

Figure 5 illustrates Modbus Plus network layouts using communication adapters in a distributed I/O control application. Note that only one programmable controller and the required I/O nodes are present in this kind of application.

Figure 5 Communication Adapters on a Distributed I/O Network



2.2 **How Communication Adapters Handle Messages**

2.2.1 **How Messages are Defined in the Application**

The user defines I/O message transactions in the Peer Cop table of the controller. Entries to the table are made using panel software, such as Schneider s Concept or Modsoft software.

The Peer Cop table specifies the controller registers that are to be used for the I/O data storage. It also specifies the Communication Adapter node addresses which will handle that data.

In addition to the use of Peer Cop, I/O data messages can be transacted using Modbus Plus MSTR function blocks in the controller s application program.

The user sets each Communication Adapter s node address in switches on the adapter s front panel. The switch setting must match the address defined for that adapter in the controller s Peer Cop table. It must also match the address in MSTR function blocks intended for that adapter.

2.2.2 **How Messages are Transacted on the Network**

How Nodes Access the Network

Modbus Plus nodes access the network for transmission by acquiring a token frame that is passed from node to node in a rotating address sequence. The node currently holding the token has the sole right to transmit. All other nodes monitor the network and extract messages addressed to them.

Incoming Messages from Communication Adapters

When a Communication Adapter at an Input base module acquires the token, it transmits its message to the programmable controller node. The message data describes the current states of the signals at the base s field input terminals.

The controller reads the message and steers its contents into the data registers defined for that adapter s address in the controller s Peer Cop table.

Outgoing Messages to Communication Adapters

When the programmable controller acquires the token, it transmits its messages to the Communication Adapters. Messages are sent to the node addresses defined in the controller s Peer Cop table, with the message contents taken from the data registers defined in the table.

Each Communication Adapter at an Output base module uses its received message to control the field devices connected to the base's output terminals.

2.2.3 **Mapping Data to I/O Bases**

Mapping of data between the controller's data registers and the field terminals of I/O bases is unique to each model of I/O base. Mapping is described in the *TSX Momentum I/O Bases User Manual*, part number 870 USE 002 00.

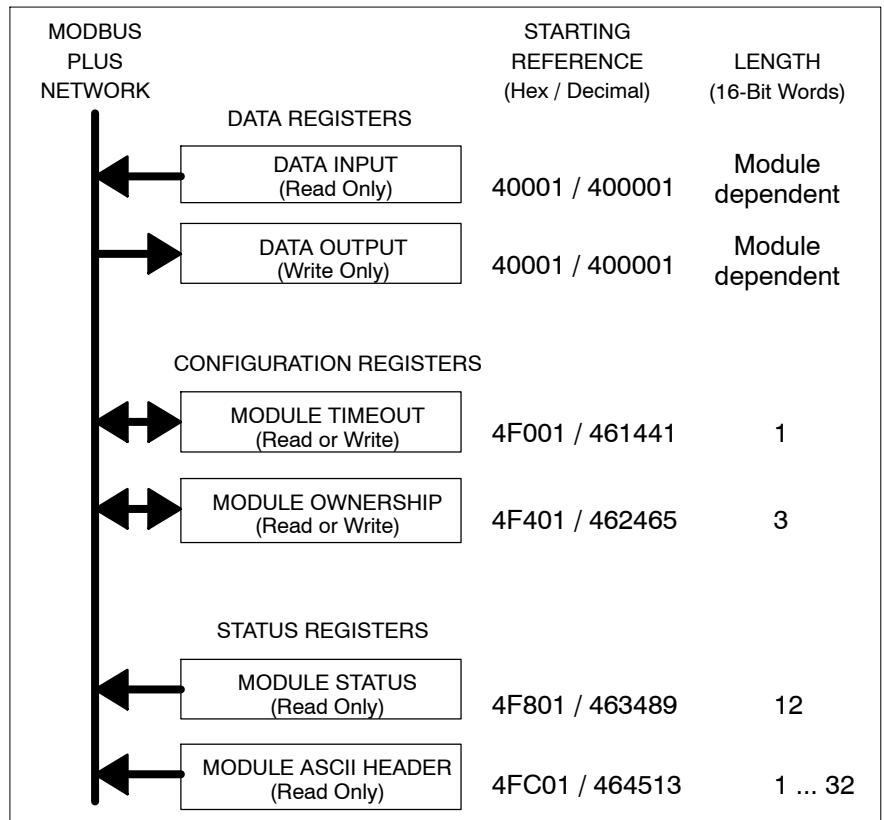
2.3

Communication Access Registers

Each adapter contains three groups of internal registers that enable the application program to communicate with the I/O base module. The application can access the registers through the network to transfer input or output data at the module's field terminals, to set or retrieve the module's configuration, or to monitor its status.

The registers are accessed as 4XXXX references in a controller's application program. Note that the Data Registers are the only ones that can be accessed by the controller's Peer Cop table. All of the registers can be accessed by MSTR function blocks.

Figure 6 Communication Adapter Access Registers



2.3.1 Data Registers

40001 hex Data Input or Output

Starting reference 40001 is used to address input data from field inputs and output data to field outputs. The data field length is determined by the specific I/O base.

This reference is the only one that is accessible through Peer Cop data transfers. All other registers can be accessed using MSTR blocks.

2.3.2 Configuration Registers

4F001 hex Outputs Holdup Timeout Value

Reference 4F001 specifies the amount of time that outputs will be held in their current state, if they are not updated by a new Modbus Plus Write command. If the module's holdup time expires before a new write command is received, all outputs are set to logical 0 (zero).

The field length is one word. The timeout value is expressed in units of 10 milliseconds, with a minimum register value of 30 (300 milliseconds) and maximum value of 6000 (60 seconds). The default value is 100 (1 second).

The register's contents can be read using a Modbus Plus Read command.

4F401 hex Ownership of Write Privilege

Starting reference 4F401 specifies the addresses of up to three nodes which may concurrently own write privilege to the adapter. The field length is three words.

When the adapter first receives power, it will give sole write privilege to the first node that writes to it. The adapter maintains an internal 60-second timer for handling the write privilege, and will reserve sole privilege to that node as long as the node continues to write within 60-second intervals to the adapter.

A node which currently owns the write privilege may write up to three words to the adapter starting at reference 4F401. Each of the three words must correspond to a valid node address in the range 1 ... 64 decimal. With those addresses stored in the adapter, any of those three nodes may then write to the adapter. This allows up to three nodes to concurrently own write privilege to the adapter.

If writes continue to occur within the 60-second interval from any of the three privileged nodes, no other node may write to the adapter. If the timer is allowed to expire, any node may write to the adapter.

Note that this 60-second Write Privilege timer is separate from the Outputs Holdup timer, and applies only to the write privilege. Any node may read the input data or status information from the adapter. The 60-second time is a fixed value and is not accessible to the application.

2.3.3 Status Registers

4F801 hex Module Status Block

These registers provide information about the module's revision level and current operating parameters.

The block's length is 12 words. The registers can be read, but cannot be written into.

Table 4 Module Status Block Layout

Reference (hex)	Purpose	Contents
4F801	Length of status block (words)	12 decimal
4F802	I/O module quantity of input bytes	Module dependent
4F803	I/O module quantity of output bytes	Module dependent
4F804	I/O module ID number	Module dependent
4F805	I/O module revision number	Format: XR where: X = upper 4 bits, always 0000 R = lower 12 bits, defining the revision as 3 hex characters. Example: 100 hex = Rev. 1..00 200 hex = Rev. 2.00
4F806	ASCII header block length (words)	Module dependent
4F807	Last node address to communicate	1 ... 64 decimal
4F808	Remaining ownership reservation time	30 ... 6000 decimal, in units of 10 ms (300 ms ... 60 s)
4F809	Remaining outputs holdup time	30 ... 6000 decimal, in units of 10 ms (300 ms ... 60 s)
4F80A	I/O module health	8000 hex = healthy 0000 hex = not healthy
4F80B	I/O module last error value	Module dependent
4F80B	I/O module error counter	Error count 0000 ... FFFF hex

4FC01 hex Module ASCII Header Block

These registers contain an ASCII text description of the module. The registers can be read, but cannot be written into.

The block length depends upon the type of I/O base to which the adapter is connected. The maximum length is 64 bytes of ASCII characters, corresponding to a length of 8 ... 32 words as specified in word 6 of the module status block (at reference 4F806).

The following table shows the header block layout as a string of ASCII characters as they are positioned from the starting reference 4FC01.

Table 5 Module ASCII Header Block Layout

4FC01 + Byte Offset	ASCII Characters	Meaning
0 ... 10	M O D B U S P L U S	Modbus Plus network device
11	20 hex (32 decimal)	space
12	20 hex (32 decimal)	space
13 14 15	I E C	IEC data mode (Data bit order per IEC standard)
16	20 hex (32 decimal)	space
17 18 19	D I G E X P A N A	Digital module (ID range: XX00 ... XX7F hex) Expert module (ID range: XX80 ... XXBF hex) Analog module (ID range: XXC0 ... XXFE hex)
20 21	H H L L	Module ID code (HH = high byte, LL = low byte)
22 23	I I O O	Module I/O words (II = input words, OO = output words)
24 ... 63	□ □	Reserved

Figure 7 Examples: ASCII Header Block

