

MPS ETHERNET INTERFACE

December 13, 2007

PRELIMINARY

Copyright © 2007 Startco Engineering Ltd.

All rights reserved.

This page intentionally left blank.



TABLE OF CONTENTS

	PAGE
1. General	1
2. Ethernet Protocol.....	1
2.1 Protocol Setup.....	1
2.2 LED Indication	1
2.3 Communication Status and Timeout.....	2
2.4 File System	2
3. Modbus TCP Protocol (Firmware > 2.50)	2
3.1 Function Codes Supported.....	3
3.2 Register Database	3
3.3 Reading Data Records	3
3.4 User-Defined Registers.....	3
3.5 Timeout Prevention	3
3.6 Error Responses	3
4. Legacy Modbus TCP and Ethernet/IP (Firmware < 2.50).....	4
4.1 Modbus/TCP Protocol	5
4.1.1 Modbus TCP Input.....	5
4.1.2 Modbus TCP Output.....	5
4.2 Ethernet/IP	5
4.2.1 Ethernet/IP Input.....	5
4.2.2 Ethernet/IP Output	6
4.3 Timeout Prevention	6
4.4 Network Configuration Files	6
5. Specifications.....	6

DISCLAIMER

Specifications are subject to change without notice. Startco Engineering Ltd. is not liable for contingent or consequential damages, or for expenses sustained as a result of incorrect application, incorrect adjustment, or a malfunction.

LIST OF FIGURES

	PAGE
2.1 MPS CTU Side View Showing LED's	2

LIST OF TABLES

	PAGE
2.1 LED 2 – Module Status	1
2.2 LED 3 – Network Status.....	1
3.1 Supported Commands.....	3
4.1 MPS Command Table.....	4
4.2 Function Codes	5
4.3 Modbus Control Command Assembly.....	5
4.4 Modbus Set-Point Assembly	5
4.5 Ethernet/IP Control Command Assembly.....	6
4.6 Ethernet/IP Set-Point Assembly	6



This page intentionally left blank.



1. GENERAL

The Ethernet interface supports Modbus TCP and Ethernet/IP. For firmware revision 2.50 and higher a new version of Modbus TCP has been added in addition to the legacy version. This new version removes MPS register-access limitations and provides full access to all MPS parameters. The new version is compatible with SE-Comm-RIS. For legacy applications, the old functionality is still provided.

Starting with version 2.50, setting the IP address and mask is done using an OPI menu selection and does not require the “IPConfig” configuration software.

Section 2. is common for all Ethernet protocols. Section 3 describes the new Modbus TCP implementation and Section 4 describes the legacy version.

2. ETHERNET PROTOCOL

This section applies to both legacy and new versions of Modbus TCP, and Ethernet/IP.

2.1 PROTOCOL SETUP

The protocol type, IP address, and IP Mask is set in the *Setup | Hardware | Network Comms* menu.

For new installations requiring Modbus TCP, select *Modbus TCP*. This provides full access to the MPS parameters as described in the MPS manual Appendix E. Multiple-register read and write instructions are supported and write requests do not require a special command sequence. This selection is compatible with SE-Comm-RIS and is recommended for new installations. This mode of operation is described in Section 3.

For legacy applications or to support Ethernet/IP, select the network type as *Anybus*. In Anybus mode, Modbus and Ethernet/IP data from the MPS is limited to User Register data with a fixed size of 64 bytes. Only a single set point can be written at a time and requires the use of a special 8-byte command sequence. Anybus-mode operation is described in Section 4.

Starting with MPS Version 2.50 IPConfig⁽²⁾ is not required because the IP Address and Mask are set using the *Ethernet IP* and *Ethernet Mask* menu settings, however, IPConfig can still be used to determine the IP Address and Mask settings for connected devices.

NOTES:

- (1) The MPS communication address is defined by the *Ethernet IP* setting and is unique for each MPS. The *Network ID* setting is ignored by the MPS. The Modbus broadcast address of 255 is not supported.
- (2) Although IPConfig software can still be used to set the IP Address and Mask, these settings are temporary and overwritten by the menu-system value when the MPS Ethernet driver is re-started during powerup or when network parameters are changed.
- (3) Communications options are mutually exclusive. Selecting *Anybus* or *Modbus TCP* disables the RS-485 interface.
- (4) Select *Anybus* or *Modbus TCP* only if the MPS has the Ethernet option installed (MPS-CTU-04-XX).

2.2 LED INDICATION

The module contains four LED indicators as shown in Fig. 2.1.

LED 1 - Link

Indicates that the module is connected to an Ethernet network.

TABLE 2.1 LED 2 - MODULE STATUS

STATE	DESCRIPTION
Steady Off	No Power
Steady Green	Device Operational
Flashing Green	Standby
Flashing Red	Minor Fault
Steady Red	Major Fault
Flashing Green/Red	Self-Test

TABLE 2.2 LED 3 - NETWORK STATUS

STATE	DESCRIPTION
Steady Off	No Power or No IP Address
Steady Green	Connected
Flashing Green	No Connections
Flashing Red	Connection Timeout
Steady Red	Duplicate IP
Flashing Green/Red	Self-Test

LED 4 - Activity LED

Flashes green each time a packet is received or transmitted.

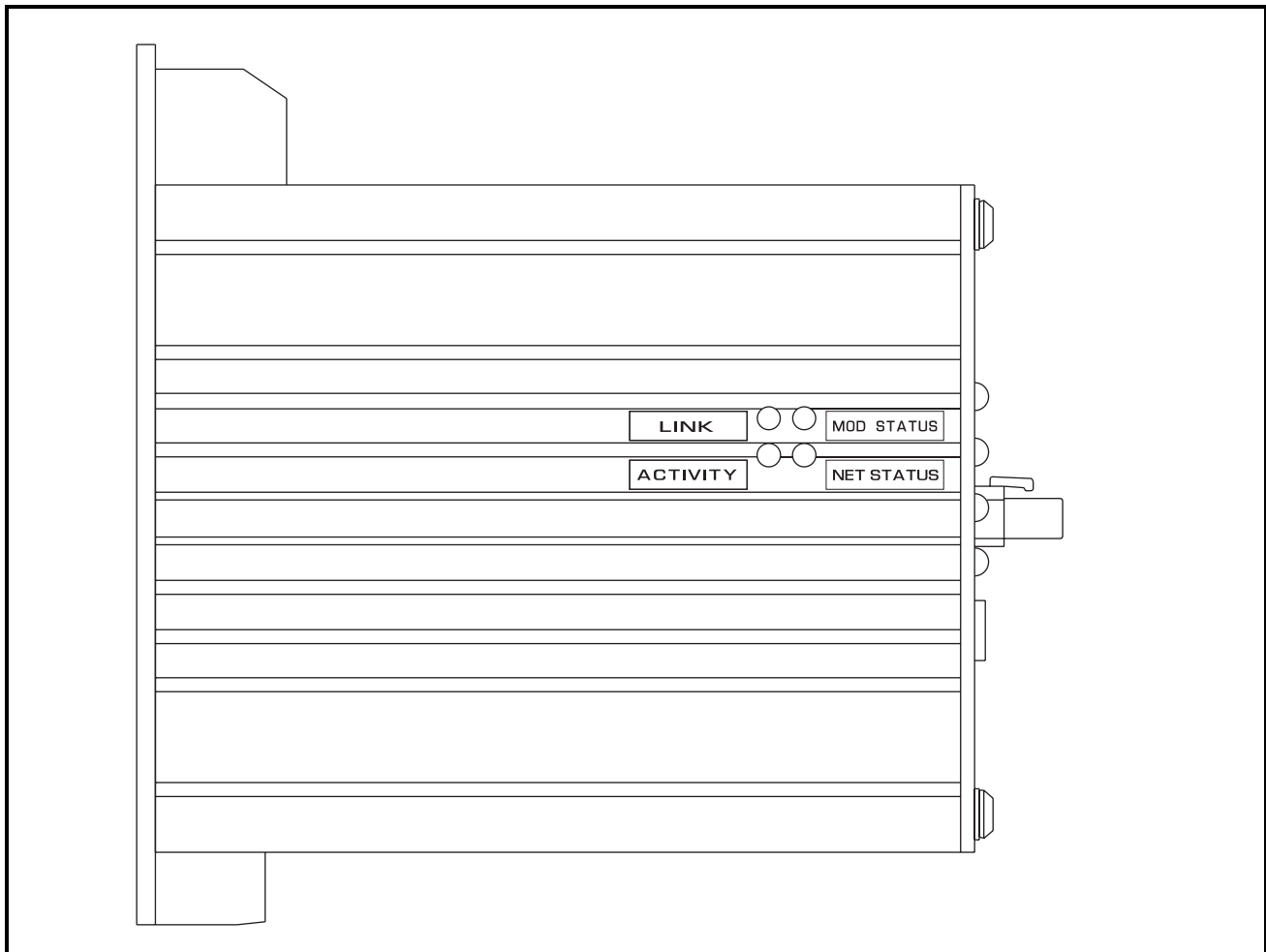


FIGURE 2.1 MPS-CTU Side View Showing LED's

2.3 COMMUNICATION STATUS AND TIMEOUT

The status of the Ethernet communication module is indicated as "Ethernet: ONLINE" when the module is operating properly, and as "Ethernet: OFFLINE" when the module is not operating properly. Module errors require the module to be reinitialised. The module is initialised on power up or can be initialised using the OPI. To initialise the module using the OPI, first disable the module by selecting *None* in the *Setup | Hardware | Network Type* menu and then select *Anybus* or *Modbus TCP* to enable the module.

In applications where the MPS start/stop functions are controlled by the network, the MPS can be configured to trip or alarm on loss of communication to the module. This feature is enabled using the *Setup | Hardware | Network Comms | Network Error* menu.

To prevent a timeout trip when using the *Modbus TCP* selection, see Section 3.5. To prevent a timeout trip when using the *Anybus* selection (legacy Modbus TCP or Ethernet/IP), see Section 4.3.

2.4 FILE SYSTEM

The module contains a file system that may be useful for storing files associated with the MPS. The file system is a fixed-size storage area with a hierarchical directory structure.

The file system is accessible via FTP, Telnet, HTTP.

The file system is case sensitive. This means that the file 'AnyBus.txt' is not identical to the file 'AnyBus.TXT'. Filenames can be a maximum of 48 characters long. Pathnames can be 256 characters in total, filename included.

3. MODBUS TCP PROTOCOL (Firmware ≥ 2.50)

The MPS implements the Modbus TCP protocol on port 502. The MPS is a slave (server) on the network. It cannot initiate communication. Up to 5 clients can be connected simultaneously to a single MPS server.



3.1 FUNCTION CODES SUPPORTED

The MPS supports the following function codes:

- Read Holding Registers (Function Code 3)
- Read Input Registers (Function Code 4)
- Write Single Register (Function Code 6)
- Write Multiple Registers (Function Code 16)
- Force Single Coil (Function Code 5)

Function Code 3 and Function Code 4 perform the same function and are used to read data from the MPS. Function Code 6 and Function Code 16 are used to write data to the MPS. See MPS Manual Appendix E for the Modbus register list and address definitions. These functions use Modbus address group 4.

Function Code 5 is used to issue commands to the MPS. The command code and action is listed in Table 3.1. In terms of Modbus, the command code is the coil address (Modbus address group 0). The command is issued by setting the coil to ON at the specified address location. All commands use the ON request to issue the command. For example, to reset trips, the coil at Modbus coil address 00004 is set to ON. Coil commands are “one-shot” commands.

TABLE 3.1 SUPPORTED COMMANDS

COMMAND CODE	COIL ADDRESS	ACTION
0x0000	00001	STOP
0x0001	00002	START1
0x0002	00003	START2
0x0003	00004	Reset Trips
0x0004	00005	Set Real-Time Clock
0x0005	00006	Clear Data-Logging Records
0x0006	00007	Clear Trip Counters
0x0007	00008	Clear Energy Totals
0x0008	00009	Clear Running Hours
0x0009	00010	Emergency I ² t and Trip Reset
0x000A	00011	Select Local Control
0x000B	00012	De-select Local Control
0x000C	00013	Re-enable Temperature Protection

For PLCs not supporting Function Code 5, commands can be issued using Function Code 6 or 16. Commands are written to MPS register 6 (Modbus register 40007). Use the command codes listed in Table 3.1. For Function Code 16, the first data element is interpreted as the command code and subsequent bytes are ignored.

3.2 REGISTER DATABASE

Appendix E in the MPS manual contains the Modbus Register Table. The table starts at register 0 (Modbus 40001) and each register is 16-bits wide. Types “long” and “float” are 32-bit values. For both long and float types, the low-order word is transmitted first followed by the high-order word. Word values have the high byte followed by the low byte. Float types are per the IEEE 754 Floating-Point Standard. All bytes of long and

float types must be written using one message or an error will result. This does not apply for read commands.

3.3 READING DATA RECORDS

Event record information is located starting at MPS register 973 (Modbus 40974).

Only one event record can be read at a time. Record data is for the record indicated by the Record Selector. To select a record, write the record number to Record Selector with the first message and then read the values in the record with a second message. Record Head points to the next available record. The last event record captured is at Record Head minus one.

The Record Selector must be in the range of 0 to 99. Values outside this range will select record 0.

3.4 USER-DEFINED REGISTERS

User-Defined Registers are used to assemble data in groups in order to minimize the amount of message requests. User-Defined Register values are entered using the *Setup | Hardware | Network Comms | User Register* menu, by using SE-Comm-RIS, or by using network communication messages.

The values entered are the MPS register numbers corresponding to the required parameter as listed in the MPS Manual Appendix E. The entered values are accessible from the menu or via communications by reading the register values starting at MPS register 1400 (Modbus 41401).

The data corresponding to these register values is retrieved by reading the values starting at registers 1432 (Modbus 41433). Data format is a function of the associated MPS register type.

Typically, for PLC communications it is desirable to define data assemblies that are grouped by data type (float or integer). A single read can then access all required float values while another read can access the integer values.

For example, to access the three phase currents enter 860, 861, 862, 863, 864, and 865 in User Register 0 to 5. The values are read by reading three float values starting at Modbus 41433. In a similar manner, the trip summary, motor status, starter status, and Message 0 can be read by entering 1096, 1097, 1098, 1104 in the next available user-register locations starting at User Register 6. These can be read starting at MPS Register 1438 (Modbus 41439).

3.5 TIMEOUT PREVENTION

To prevent a timeout trip on the MPS, a valid Modbus TCP request addressed to the specific slave is required. This can be a read or write request.

3.6 ERROR RESPONSES

Errors can originate from the hardware or communications software. When a hardware error occurs, “Anybus Error!” is displayed in the *Metering |*



Network Status menu, along with an error code. If errors persist, contact Startco Engineering Ltd.

The MPS supports the following Modbus TCP communication error responses:

- 01: Illegal Function—The function code sent to the MPS server is not supported.
- 02: Illegal Data Address—The requested address is not within the data address range in the MPS.
- 03: Illegal Data Value—Data value is not within the required range.

4. LEGACY MODBUS TCP AND ETHERNET/IP (Firmware < 2.50)

This section contains information relative to operation using the *Anybus* protocol selection. The *Anybus* selection supports Ethernet/IP and the legacy version of Modbus TCP.

The MPS implements the Modbus protocol on port 502. The MPS is a slave (server) on the network. It cannot initiate communication.

Up to 5 clients can be connected simultaneously to a single MPS server.

NOTE: The *Anybus* selection is required to support Ethernet/IP and legacy Modbus TCP.

Both Modbus TCP and Ethernet/IP use INPUT and OUTPUT assemblies to transfer data. Input assemblies are used to transfer data from the MPS to the network, and OUTPUT assemblies are used to transfer data from the network to the MPS.

The INPUT assembly consists of 64 bytes representing user-defined data. User defined data is configured by entering the required MPS register numbers in the User Defined Register block using the MPS menu system *Setup | Hardware | Network Comms | User Registers* or by using SE-Comm-RIS program. The format of the data in the assembly is a function of the associated MPS Register. For register definitions and formats see MPS manual Appendix E and F. For example, to access the first four RTD temperatures in RTD Module 1, enter register numbers 902, 903, 904, 905, 906, 907, 908, 909 in the User Defined data area. In the resulting assembly, the first 8 words (16 bytes) will contain the four float values of the RTD temperatures. The remaining values are a function of the corresponding User Register pointers and can be any type. To prevent a read error, unused User Defined data must be set to a valid MPS Register number or zero. Byte order for 16- and 32- bit values follows the convention for the specific protocol and may not be the same as indicated in MPS Appendix E, however, register numbers for float values are still entered in sequence as per the example above.

The OUTPUT area contains a 4-word (8-byte) memory buffer that is used to send control commands

and set points to the MPS. This memory has read/write access from the network. Network output is sent to this buffer and the buffer is read by the MPS.

The first word in the buffer (byte 0,1) is the Request Header. This value is used to identify the message as a control-command request or a 1-word or 2-word set-point request. A transition from 0 to 3 indicates a control command, a transition from 0 to 19 (0x0013 hex) indicates a 1-word set point, and a transition from 0 to 35 (0x0023) indicates a 2-word or float set-point command. For control commands, the second word is one of the values listed in Table 4.1.

TABLE 4.1 MPS COMMAND TABLE

COMMAND CODE	ACTION
0x0000	STOP
0x0001	START1
0x0002	START2
0x0003	Reset Trips
0x0004	Set Real-Time Clock
0x0005	Clear Data-Logging Records
0x0006	Clear Trip Counters
0x0007	Clear Energy Totals
0x0008	Clear Running Hours
0x0009	Emergency I ² t and Trip Reset
0x000A	Select Local Control
0x000B	De-select Local Control
0x000C	Re-enable Temperature Protection

For set-point commands, the second word (byte 2,3) is the MPS Register number of the set point to be changed. This must be a valid set point with write access (MPS Appendix E 'ACCESS' column should indicate R/W). The remaining bytes contain the set-point data.

The command buffer should be written using a single message, however, in some cases, where a configuration tool is used, individual bytes may be sent as they are entered. For these applications, the low byte of the Request Header must be sent last and only after all other bytes have been entered.

For PLC applications where mixing of words and floats is not allowed, there are three options:

- Convert the float⁽¹⁾ data to a 2-word format and write the message using word format.
- Convert the Request Header and MPS Register to a single float¹ containing the same two word values. Write the resulting 2 words to the buffer.
- In the case of Modbus TCP, write the float to word address 1030 (Modbus 41,031), then write the Request Header and MPS Register number as two words starting at address 1028 (Modbus 41,029). The set point is updated by the MPS on a Request Header transition from 0 to 35 (0x0023).

⁽¹⁾ There is a HEX calculator program available from Startco Engineering Ltd to calculate the required float or word values for this option.



The following sections outline the specific operation of Modbus TCP and Ethernet/IP.

4.1 MODBUS/TCP PROTOCOL

The MPS Ethernet module can handle up to 5 simultaneous connections. It supports all of the Modbus/TCP class 0 and class 1 and some of the class 2 commands.

Although word addressing (4xxxx) is recommended for operation, both bit and word addressing is supported.

A 16-bit word can be read as a single word using word addressing and can also be read as bits by using bit addressing. The bit address is 16 times the word address. For example, a value at word address 34 has its most significant bit located at bit address 16x34 or 544. The use of the Modbus prefix 0, 1, 3, and 4 is a function of the master and may not be required. Check the master's documentation

TABLE 4.2 FUNCTION CODES

FUNCTION CODE	FUNCTION NAME	CLASS	AFFECTS AREA	ADDRESSING METHOD	MODBUS GROUP
1	Read coils	1	IN/OUT	Bit	0xxxx
2	Read input discretes	1	IN/OUT	Bit	1xxxx
3	Read multiple registers	0	IN/OUT	Word	4xxxx
4	Read input registers	1	IN/OUT	Word	3xxxx
5	Write coil	1	OUT	Bit	0xxxx
6	Write single register	1	OUT	Word	4xxxx
7 *	Read exception status	1	-	-	-
15	Force multiple coil	2	OUT	Bit	0xxxx
16	Force multiple registers	0	OUT	Word	4xxxx
22	Mask write register	2	OUT	Word	-
23	Read/Write registers	2	IN/OUT	Word	-

- * Exception Code 01: Illegal Function
- Exception Code 02: Illegal Data Address
- Exception Code 03: Illegal Data Value

4.1.1 MODBUS TCP INPUT

INPUT data from the MPS is read starting at word address 32 (Modbus 40,033). Although the assembly size is 64 bytes, the total number of words or floats that make up the read request can be less than the 64-byte total. In the RTD temperature example, only four floats (16 bytes) need to be read. Byte order for 16- and 32- bit values follows the convention for the Modbus and is the same as indicated in MPS Appendix E.

NOTE: For Modbus TCP, user data is read starting at 40,033 not 41,433.

4.1.2 MODBUS TCP OUTPUT

The OUTPUT assembly (memory buffer) is located at word address 1028 (Modbus 41,029), and has read/write capability.

The control-command assembly is shown in Table 4.3 and the set-point assembly is shown in Table 4.4.

TABLE 4.3 MODBUS CONTROL COMMAND ASSEMBLY

BYTE NUMBER	DESCRIPTION
0	Request Header (High)
1	Request Header (Low)
2	MPS Command (High)
3	MPS Command (Low)
4-7	Not used

TABLE 4.4 MODBUS SET-POINT ASSEMBLY

BYTE NUMBER	DESCRIPTION
0	Request Header (High)
1	Request Header (Low)
2	Destination Register (High)
3	Destination Register (Low)
4-7	Set-Point Data. 2 bytes for words, 4 bytes for floats

For word set points (short, 16-bit), byte 4 is the high byte, byte 5 is the low byte, and byte 6 and 7 are not used. For float set points, standard Modbus byte order is used to represent the IEEE 32-bit float. Byte 4 is the high byte and byte 5 is the low byte of the low-order word of the float. Byte 6 is the high byte and byte 7 is the low byte of the high-order word of the float. Character values can use the 1-word or 2-word format, each consisting of a pair of characters where the high byte is the first character and the low byte is the second character.

4.2 ETHERNET/IP

4.2.1 ETHERNET/IP INPUT

INPUT data is read by a GetSingleAttribute service to Class 4, Instance 100, Attribute 3. Byte order for 16- and 32- bit values follows the convention for the EtherNet/IP protocol and is not the same as indicated in MPS



Appendix E. In any case, float value registers should be listed as two registers in sequence (902, 903...).

4.2.2 ETHERNET/IP OUTPUT

The OUTPUT assembly (memory buffer) is written using the SetSingleAttribute to assembly Class 4, Instance 150, Attribute 3. The Control Command Assembly is shown in Table 4.5 and the set-point assembly is shown in Table 4.6.

TABLE 4.5 ETHERNET/IP CONTROL COMMAND ASSEMBLY

BYTE NUMBER	DESCRIPTION
0	Request Header (Low)
1	Request Header (High)
2	MPS Command (Low)
3	MPS Command (High)
4-7	Not used

TABLE 4.6 ETHERNET/IP SET-POINT ASSEMBLY

BYTE NUMBER	DESCRIPTION
0	Request Header (Low)
1	Request Header (High)
2	Destination Register (Low)
3	Destination Register (High)
4-7	Set-Point Data. 2 bytes for words, 4 bytes for floats

For word set points (short, 16-bit), byte 4 is the low byte, byte 5 is the high byte, and byte 6 and 7 are not used. For float set points (IEEE 32-bit), byte 4 is the low byte and byte 7 is the high byte of the 4-byte float value. Character values use the 1-word or 2-word format, each consisting of a pair of characters where the high byte is the first character and the low byte is the second character.

4.3 TIMEOUT PREVENTION

To prevent a trip or alarm, new data must be written to the OUTPUT memory buffer at an interval less than the MPS trip time of 3 seconds. The suggested method is to use the Control Command format. At regular intervals write an incrementing value to the Command word (bytes 2,3) while keeping the Request Header word (bytes 0,1) at 0. Keeping the Request Header at 0 prevents the MPS from interpreting the Command data as a valid control command. Incrementing the Command word ensures that a “changed data” event is posted to indicate valid communications. Reading data from the module is not sufficient to satisfy the time-out timer.

Communication status is displayed in the *Metering | Comm State* menu.

Writes to the OUPUT memory buffer are indicated by “Output: NO” or “Output: YES”. If the module is receiving output from the network, then “Output YES” will be displayed.

4.4 NETWORK CONFIGURATION FILES

For Ethernet/IP, an eds file is available from www.startco.ca, or www.anybus.com. As part of the network configuration, the input and output assembly size may need to be specified. The INPUT assembly size is fixed at 64 bytes and the OUTPUT assembly size is 8 bytes.

This document does not describe the specific objects required to support the Ethernet/IP protocol. For details on these objects, see the Anybus-S documentation at www.anybus.com.

5. SPECIFICATIONS

Interface10BaseT, 100BaseT,
Cat. 3, 4, 5, UTP, STP
ProtocolModbus TCP
Baud Rate.....10/100 Mbps.
Number of Slaves Connected.....Up to 254 units
Number of Connections/SlaveUp to five (5)
Bus length100 m per segment