User Manual

EtherNet/IP to Modbus-TCP Linking Device

Doc.Id. SCM-1202-008
Rev. 1.1

Connecting Devices™

HALMSTAD • CHICAGO • KARLSRUHE • TOKYO • BEIJING • MILANO • MULHOUSE • COVENTRY • PUNE • COPENHAGEN

HMS Industrial Networks
Mailing address: Box 4126, 300 04 Halmstad, Sweden
Visiting address: Stationsgatan 37, Halmstad, Sweden

E-mail: info@hms-networks.com
Web: www.anybus.com

Important User Information

This document is intended to provide a good understanding of the functionality offered by the EtherNet/IP to Modbus-TCP linking device. The reader of this document is expected to be familiar with high level software design, and communication systems in general.

Liability

Every care has been taken in the preparation of this manual. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, 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 considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the US and other countries.

Trademark Acknowledgements

Anybus ® is a registered trademark of HMS Industrial Networks AB. All other trademarks are the property of their respective holders.

Warning: This is a class A product. in a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

ESD Note: This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.

Warning: DO NOT USE SD CARD OR USB CONNECTOR WHILE CIRCUIT IS LIVE UNLESS THE AREA IS KNOWN TO BE FREE OF IGNITABLE CONCENTRATIONS OF FLAMMABLE GAS OR VAPORS.

EtherNet/IP to Modbus-TCP Linking Device User Manual
Rev 1.1

Nov 2016 Doc Id SCM-1202-008

P. About This Document

For more information, documentation etc., please visit the HMS website, www.anybus.com.

P.1 Related Documents

Document	Author
Modbus Application Protocol Specification V1.1B	Modbus Organization
CIP Specification, Vol 1 (CIP Common) & 2 (EtherNet/IP)	ODVA

P.2 Document History

Summary of Recent Changes (1.0... 1.1)

Change	Page(s)
Corrected Product Code in Identity Object (01h) to 57h	51

Revision List

Revision	Date	Author(s)	Chapter(s)	Description
1.0	2016-10-27	KaD	All	First release
1.1	2016-11-15	KaD	8	Minor correction

P.3 Conventions & Terminology

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term 'linking device' refers to the EtherNet/IP to Modbus-TCP linking device
- Hexadecimal values are written in the format NNNNh, where NNNN is the hexadecimal value
- A byte always consists of 8 bits
- The terms 'master', 'scanner', 'client' and 'controller' will be used interchangeably to describe a controlling unit on the network
- The terms 'slave', 'adapter', 'server' and 'device' will be used interchangeably to describe units that are controlled by controlling units on the network

1. EtherNet/IP to Modbus-TCP Linking Device

1.1 Introduction

The EtherNet/IP to Modbus-TCP Linking Device is used to provide a seamless connection between a Modbus-TCP network and an EtherNet/IP network. The linking device enables the master of the EtherNet/IP network to control the Modbus-TCP network, and data can be transmitted transparently between the two networks.

1.1.1 Unique Integration into Studio 5000

The EtherNet/IP to Modbus-TCP linking device features a custom add-on profile (AOP) for easy integration with Studio 5000. Within this add-on profile (AOP), the HMS configuration tool can be launched. When the configuration is ready, it can automatically be translated to structured Studio 5000 controller tags.

All network and device level configuration is done within Studio 5000.

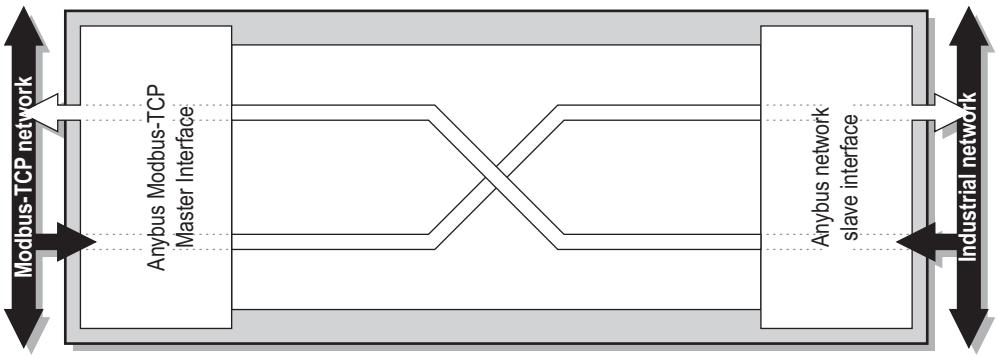
IMPORTANT:
The add-on profile is supported by RSLogix 5000, v20 and later.

1.2 Configuring the EtherNet/IP Network

The Linking Device is an EtherNet/IP adapter (slave) on the EtherNet/IP network. The general settings for the adapter interface are configured using the configuration pages. All data transfers must be configured in Studio 5000. Please note that the size of the I/O data that can be read from and written to the module is defined when configuring the linking device using the configuration pages.

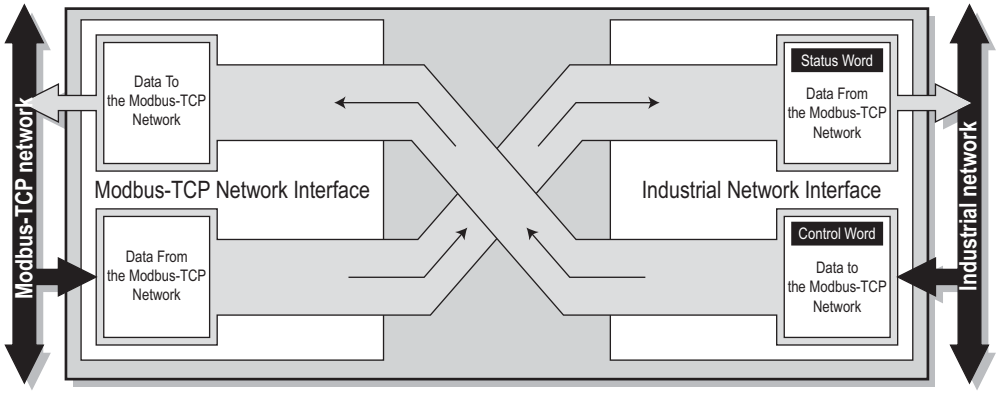
1.3 Functional Overview

Internally, the linking device consists of an intelligent gateway platform, a Modbus-TCP interface and an EtherNet/IP (slave) interface. The Modbus-TCP interface and the EtherNet/IP (slave) interface are interconnected through the intelligent gateway platform, which basically forwards data from one network to the other and vice versa as shown below.



1.4 Data Exchange

Each of the two network interfaces exchanges data on its network through two buffers. The linking device forwards the data between these buffers as shown below. Note that this process is separated from the network data exchange. While the linking device ensures data consistency (where applicable), it does not feature any built-in mechanisms for synchronization between the Modbus-TCP network and the EtherNet/IP network.



Each buffer holds a maximum of 4096 bytes of data. The first two I/O mapped bytes in either direction can be dedicated for control/status information, and another eight bytes of data coming from the Modbus-TCP network can feature a live list. Please note that the actual number of bytes that can be exchanged is highly network dependent.

Through the dedicated control word, the scanner on the EtherNet/IP network starts/stops the exchange of data on the Modbus-TCP network, and also resets the linking device if needed. The scanner on the EtherNet/IP network can see the status of the Modbus-TCP network in the corresponding status word. The live list feature gives the scanner on the EtherNet/IP network the opportunity to continuously see and monitor the status of each individual transaction on the Modbus-TCP network.

Two additional lists, transaction status and exception codes, retrievable from the module by the scanner on the EtherNet/IP network, provides detailed error information about all transactions.

The amount of data that shall be exchanged, and the use of the control/status word and the live list, is specified separately for each application. This means that even though up to 4096 bytes of data can be potentially forwarded to an interface, the amount of data that will actually be exchanged on that network is determined by the Modbus-TCP settings and the limitations of the master side fieldbus.

The available control/status functionality is described below, as well as the live list and the transaction status and exception code lists. Also note that the terminology and definitions used for different types of data vary greatly between different networking systems.

1.5 I/O Mapped Data

I/O mapped data is cyclic data, exchanged between the networks and/or devices at a high transfer rate. It is associated with implicit messaging, where data is continuously sent on the network.

1.6 Parameter Data

Parameter data is usually exchanged acyclically to set or change parameters in devices before or during normal process. Acyclical data is set up as explicit messages using CIP message instructions in Studio 5000.

Typical explicit messages that can be retrieved from the module by the scanner of the EtherNet/IP network includes the transaction status list and the exception code list.

1.7 Control/Status Word

The Control/Status word is always retrievable using acyclical access. Optionally, the Control/Status word can also be I/O mapped. If so, it is disabled by default. It can be enabled/disabled when configuring the EtherNet/IP network via the configuration web pages. See "EtherNet/IP (Adapter Interface)" on page 46.

The Control word is a 16-bit word (uint16) used by the EtherNet/IP network to control the linking device and subsequently also the Modbus-TCP network.

Bit	Value	Description
0 (Least significant bit)	0	Puts the linking device in idle state
	1	Puts the linking device in run state
1	-	A reboot of the linking device is triggered by a rising edge, i.e. a transition from 0 to 1
2-7	Set to zero	Unused
8-15	Set to zero	Unused

The Status word is a 16-bit word used by the linking device to report its current actual status to the EtherNet/IP network.

Bit	Value	Description
0 (Least significant bit)	0	The linking device is in idle state
	1	The linking device is in run state
1	-	This bit is reflecting the state of bit 1 in the control word Either 0 or 1
2-7	(reserved)	Unused
8-15	(reserved)	Unused

1.8 Live List

The live list features the possibility for the EtherNet/IP network to retrieve a list containing the status of every transaction on the Modbus-TCP network.

It is accessible using parameter access, and also I/O mapped by default. The I/O mapped live list can be enabled/disabled when configuring the EtherNet/IP network settings. See "EtherNet/IP (Adapter Interface)" on page 46. If the I/O mapped live list is enabled, it will occupy either byte 0-7 (control/status word not enabled) or byte 2-9 (control/status word enabled and mapped to the first two bytes) in the input data area.

All transactions and their places in the live list are also visible in the Transaction Monitor on the configuration web pages.

The live list consists of a bit array with 64 elements, where each bit corresponds to a transaction on the Modbus-TCP network as in the table below.

Byte 7		Byte 6-1	Byte 0			
Bit 63	Bit 62-56	Bit 55-8	Bit 7	Bit 6 - 2	Bit 1	Bit 0
Status of transaction no 63	Status of transaction no 62-56	...	Status of transaction no 7	Status of transaction no 6 - 2	Status of transaction no 1	Status of transaction no 0

- **Bit set to 1**

  Transaction successful.

- **Bit set to 0**

  Transaction not successful.

  Note: the reason for the unsuccessful transaction can be found on the corresponding index in the transaction status list.

The order of the transactions in the live list conforms to the order in which they are stored in the Modbus Server list.

Example

Consider the following configuration:

- Server 1 : a total of 2 transactions
- Server 2 : a total of 3 transactions
- Server 3 : a total of 1 transaction

This scenario will produce a live list as follows (assuming that the transactions are successful):

Bit 63	Bit 62 - 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	Server 3, transaction 1	Server 2, transaction 3	Server 2, transaction 2	Server 2, transaction 1	Server 1, transaction 2	Server 1, transaction 1
0	0	1	1	1	1	1	1

1.9 Transaction Status List

This list holds information about the transactions between the Modbus network and the linking device, from the perspective of the linking device.

It is a list available from the module, which is possibl3e to be retrieved acyclically (using parameter access) by the EtherNet/IP network. It contains a byte array with 64 elements, where each byte contains a transaction status code as in the table below.

The indexes in the transaction status list correspond completely to the indexes in the transaction live list.

Byte 0	Byte 1	Byte 2-6	Byte 7	Byte 8 - 55	Byte 56-62	Byte 63
Status of transaction no 0	Status of transaction no 1	Status of transaction no 2-6	Status of transaction no 7	...	Status of transaction no 56-62	Status of transaction no 63

Transaction status codes

Transaction Status Code	Description
0	Running ok
1	Gateway idle
2	No link
3	Modbus exception
4	Timeout
5	Linking device disconnect
6	Server disconnect
7	Cannot connect
8	Modbus header error
9	Internal device error
10	No valid data
11	Stop sending data to Modbus server
12	Unconfigured transaction

1.10 Exception Code List

If Modbus transactions fail, the slaves can respond with an exception code. These can be found in the exception code list available from the module, possible to be retrieved acyclically (using parameter access) by the EtherNet/IP network. It contains a byte array with 64 elements, where each byte contains an transaction exception code as in the table below. The indexes in the exception code list correspond completely to the indexes in the transaction live list.

Byte 0	Byte 1	Byte 2-6	Byte 7	Byte 8-55	Byte 56-62	Byte 63
Exception code for transaction no 0	Exception code for transaction no 1	Exception code for transaction no 2 - 6	Exception code for transaction no 7	Exception code for transaction no 8 - 55	Exception code for transaction no 56 - 62	Exception code for transaction no 63

Standard Modbus exception codes

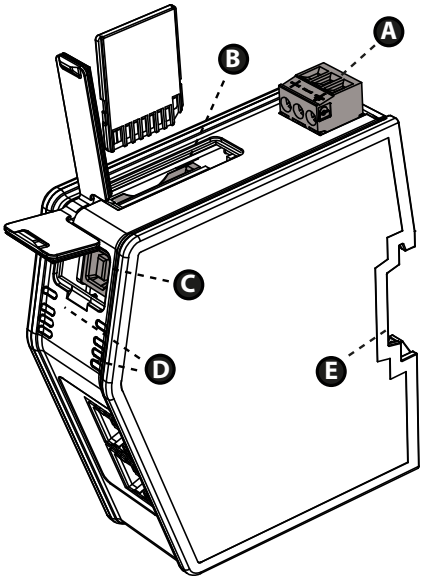
Exception Code	Description
00	No error
01	Illegal function
02	Illegal data address
03	Illegal data value
04	Slave device failure
05	Acknowledge
06	Slave device busy
08	Memory parity error
0A	Gateway path unavailable
0B	Gateway target device failed to respond

Note: The exception codes found in the exception code list are only relevant if the corresponding transaction status codes equals 3: "Modbus exception". See "Transaction Status List" on page 12 for more information.

Note: If the slave responds with an exception code not in the list, refer to the documentation of the slave for details.

2. About the HMS-EN2MB-R Linking Device

2.1 External View

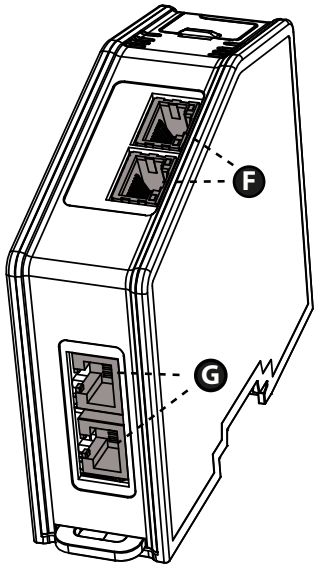


- **A: Power Connector**
  This connector is used to apply power to the linking device. It is also possible to connect protective earth (PE) to the power connector. See "Power Connector" on page 19.

- **B: SD Card Slot**
  This slot adds the possibility to store and load configurations from an SD card. See "SD Card Functionality" on page 27.

- **C: USB Port**
  This port adds the possibility to connect a PC to the linking device to perform firmware upgrades. See "USB Connector" on page 18.

- **D: Status LEDs**
  See "Status LEDs" on page 17.



- **E: DIN-rail Connector**
  The DIN-rail mechanism fastens the linking device to a DIN-rail and connects the module to protective earth (PE). See "Mounting the Linking Device" on page 15.

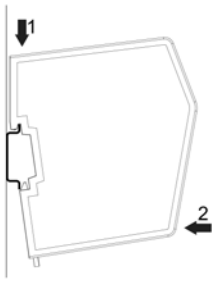
- **F: EtherNet/IP Connectors**
  See "EtherNet/IP Connectors" on page 18.

- **G: Modbus-TCP Connectors**
  2-port switch with daisy-chain functionality. See "Modbus-TCP Connectors" on page 18.

2.2 Mounting the Linking Device

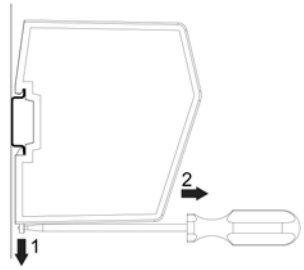
The EtherNet/IP to Modbus-TCP Linking Device can be physically installed either by mounting it onto a DIN-rail or, if installed in areas exposed to vibration, by mounting it on a wall for more stability.

2.2.1 DIN-rail Mounting

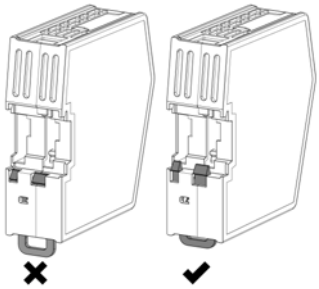


Make sure the DIN-rail fastening mechanism on the back of the module is in a fixed and closed position, i. e. pushed all the way up.

To mount the module, first hook it on to the DIN-rail (1), then push it against the DIN-rail to make it snap on (2).



To unmount the module, a screwdriver is needed. Use the screwdriver to push the DIN-rail fastening mechanism on the back of the module down until it locks in a fixed and open position (1). Then unhook the module from the DIN-rail (2).



Note: Do not leave the module with the DIN-rail fastening mechanism in a fixed and open position. This may eventually wear the fastening mechanism out so it cannot be used efficiently. Be sure to push the DIN-rail fastening mechanism back into the fixed and closed position after demounting the module.

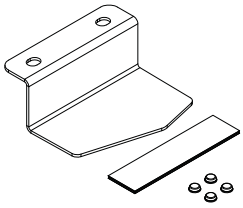
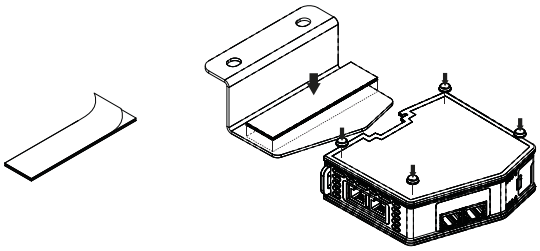
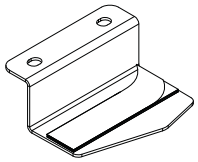
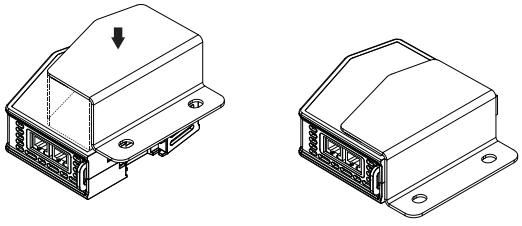
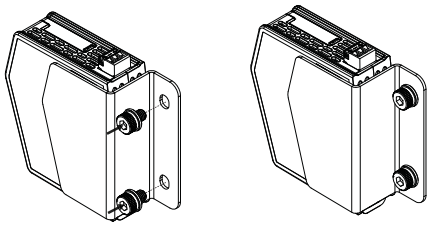
2.2.2 Wall Mounting

Use the wall mounting option if there is a need to place the linking device in an environment exposed to vibration. This way of mounting the module offers more stability than the traditional DIN-rail mounting.

Note: The device should be fastened in a standing-up position, to ensure a constant air flow.

Note: When mounting the device to a wall using the wall mount option, do not forget to connect the module to protective earth (PE) via the power connector. See "Power Connector" on page 19.

Mounting Instructions

Step	Description	Visual description
❶	Open up the package containing the wall mounting accessories. - One metal frame - Industrial velcro - Four plastic vibration dampers	
❷	Remove the plastic protection from one side of the velcro. Attach the velcro to the metal frame. Attach the four plastic vibration dampers to the linking device, on the side that will face the wall.	
❸	Remove the plastic protection from the other side of the velcro.	
❹	Turn the device around, so that the plastic vibration dampers face downwards. Fasten the metal frame to the device by pressing the frame firmly against the device, making the two velcro parts attach to each other.	
❺	Attach the metal frame and the device to a wall using screws and washers (not enclosed).	

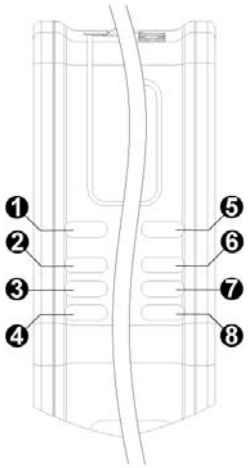
2.3 Status LEDs

Note: A test sequence is performed on all LEDs during startup.

Note: An identification LED sequence can be performed on LEDs 1, 5 and 6 by clicking the "Wink device" button in the linking device management section in the web configuration interface.

Linking Device and Modbus-TCP Network LEDs

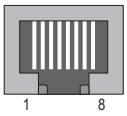
LED no	State	Status
1 - Module Status EtherNet/IP	Off	Power off
	Alternating red/green	Boot sequence
	Flashing green	Idle
	Green	Running
	Flashing red	Major or minor recoverable error
	Red	Major or minor unrecoverable error (Exception state or fatal)
2 - Network Status EtherNet/IP	Off	Power off or no IP address
	Green	Online, one or more connection established (CIP Class 1 or 3)
	Flashing green	Online, no connections established
	Red	Duplicate IP address, fatal error
	Flashing red	One or more connections timed out (CIP Class 1 or 3)
3, 4 - EtherNet/IP Ethernet Link 1 and 2	Off	No link
	Flashing green	Receiving/transmitting Ethernet packets at 100 Mbit
	Flashing yellow	Activity, receiving/transmitting Ethernet packets at 10 Mbit
5 - SD card (SD)	Green	Accessing SD card
	Red	Failure
6 - Modbus-TCP Status (MTCP)	Off	Power off
	Green	Communicating with Modbus-TCP network
	Flashing green	Idle
	Flashing red	Transaction error or timeout
	Red	Fatal error
7, 8 - Modbus-TCP Ethernet Link 1 and 2	Off	No link
	Flashing green	Receiving/transmitting Ethernet packets at 100 Mbit
	Flashing yellow	Activity, receiving/transmitting Ethernet packets at 10 Mbit



2.4 EtherNet/IP Connectors

Connectors for the EtherNet/IP network are found at the lower front of the module.

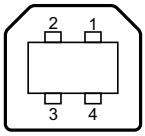
Pin no	Description
1	TX+
2	TX-
3	RX+
6	RX-
4, 5, 7, 8	Not connected
Housing	Shield



2.5 USB Connector

At the upper front of the module there is a USB connector used for firmware upgrades.

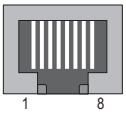
Pin no.	Description
1	+5V Input
2	USBDM (USB communication signals)
3	USBDP (USB communication signals)
4	Signal GND
Housing	Cable Shield



2.6 Modbus-TCP Connectors

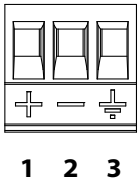
The Modbus-TCP connectors are found at the bottom of the module.

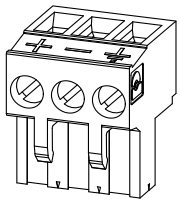
Pin no.	Description
1	TX+
2	TX-
3	RX+
4	Not connected
5	Not connected
6	RX-
7	Not connected
8	Not connected
Housing	Shield



2.7 Power Connector

Pin no.	Description
1	+24V DC
2	GND
3	PE (Protective Earth)





Notes:

- Use 60/75 or 75×C copper (CU) wire only.
- The terminal tightening torque must be between 5... 7 lbs-in (0.5... 0.8 Nm)

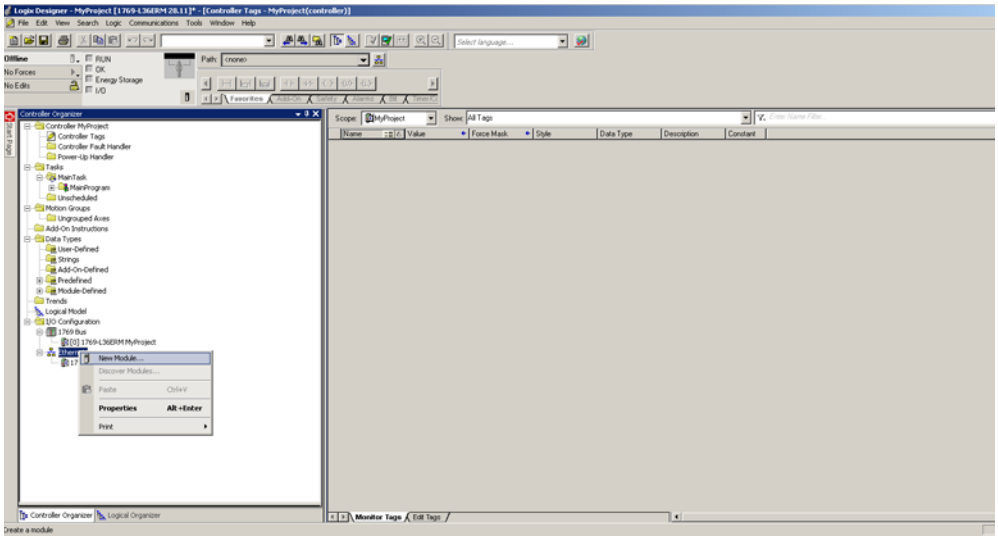
  See also...
  - "Power Supply" on page 69.

3. Studio 5000 Implementation Example

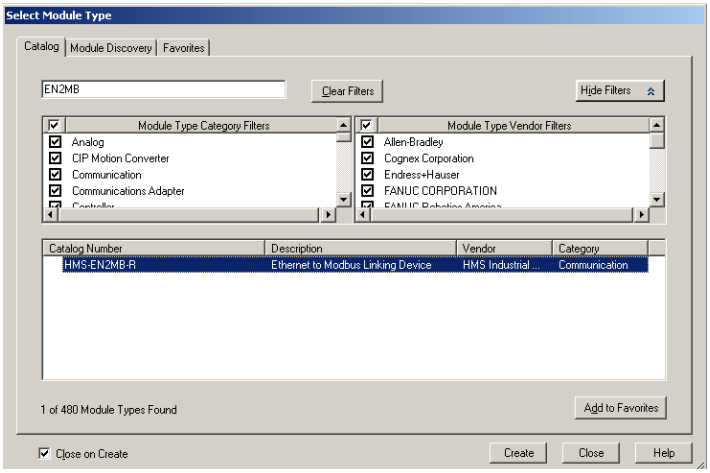
This section will guide you through all steps necessary to create a basic configuration for the HMS-EN2MB-R Linking Device.

3.1 Step by Step Guide

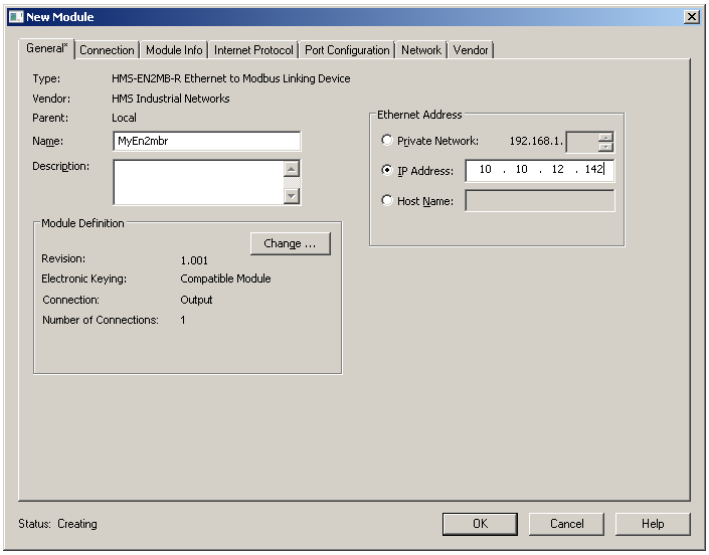
1. Start the Studio 5000 software. Expand the "I/O Configuration" folder in the tree view. Right-click "Ethernet" and select "New Module".



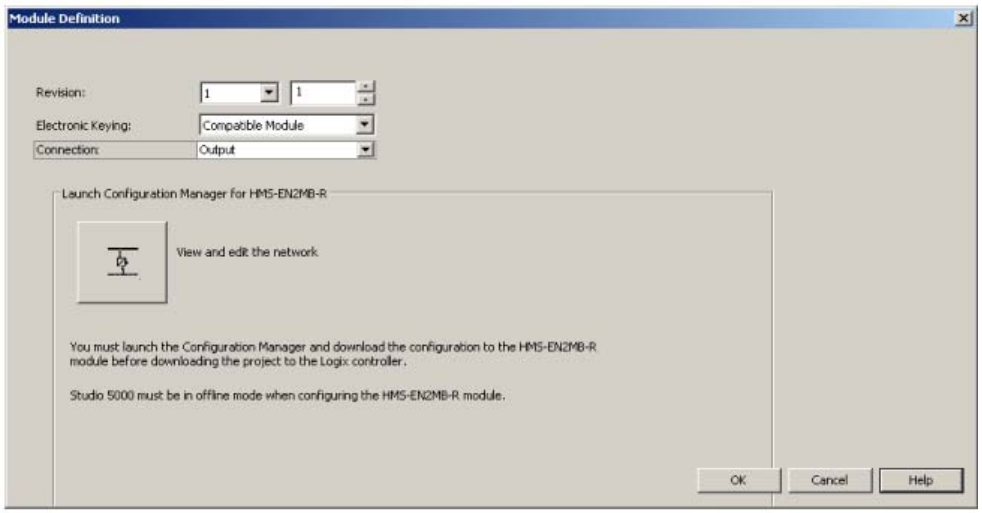
2. Select the HMS-EN2MB-R Linking Device (catalogue number: HMS-EN2MB-R) and click "Create".



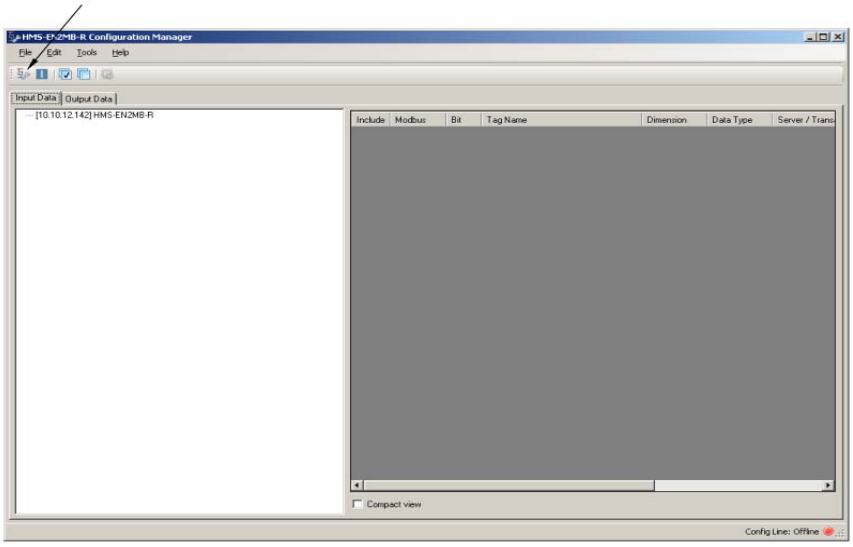
3. In the "New Module" window, assign a name to the module. The IP address should be set via the BOOTP-DHCP server and entered in the IP address field. Click "Change" in the "Module Definition" section.



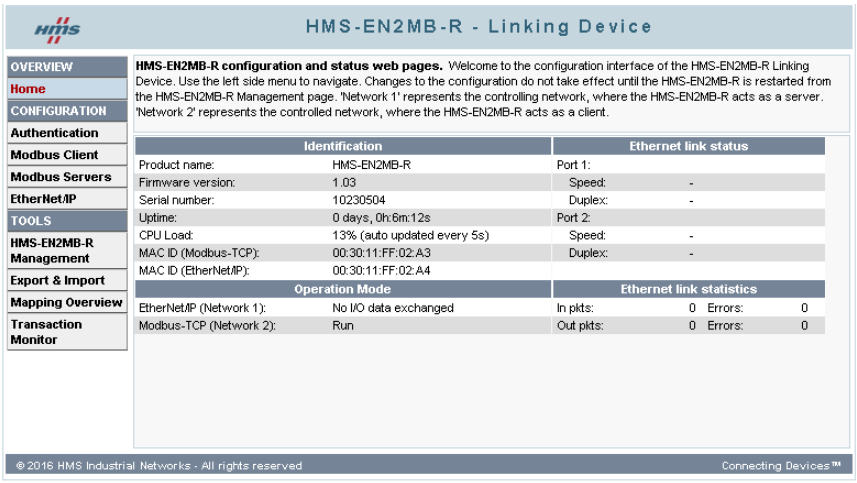
4. In the "Module Definition" window, launch the configuration manager for the HMS-EN2MB-R linking device.



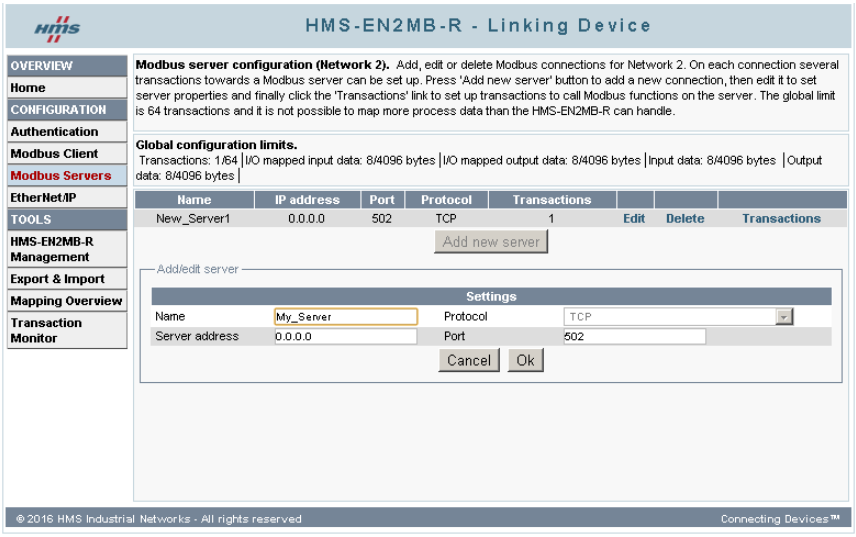
5. This part of the configuration manager is called the tag editor. Since this is a new configuration, the editor is empty. To proceed, open the HMS-EN2MB-R configuration pages by clicking the left-most icon in the tool bar.



6. The introductory page of the HMS-EN2MB-R configuration pages presents useful information about the Linking Device. In the menu to the left, choose "Modbus Servers".



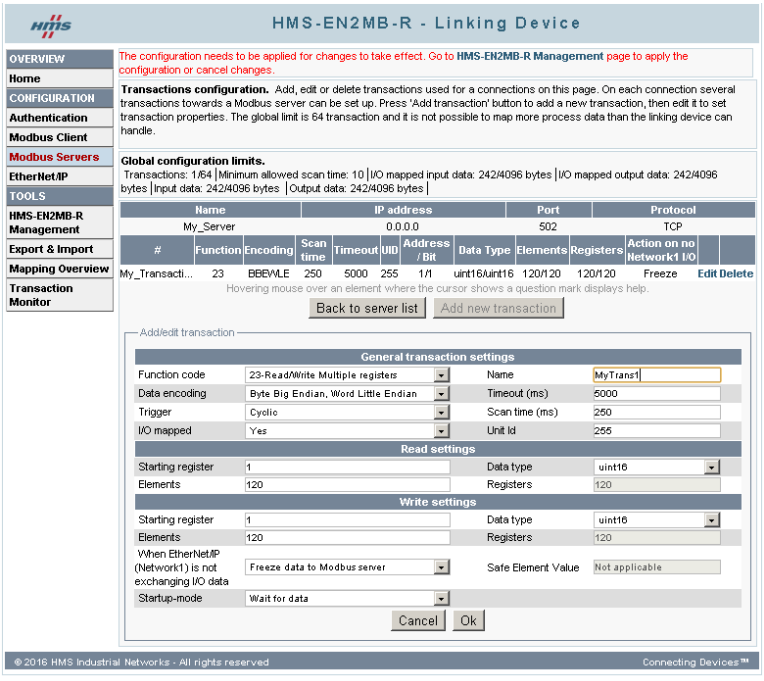
7. Create a server by clicking “Add new server”. Name it “My_Server” and click “Ok”. Click “Transactions” to create transactions for this server.



8. To create a new transaction:

- Add a new transaction, and click “Edit”.
- Name it “MyTrans1”.
- Choose Modbus function code 23 (Read/Write Multiple registers).
- In the “Elements” sections, both in “Read settings” and “Write settings”, write 120.
- Click “Ok”.

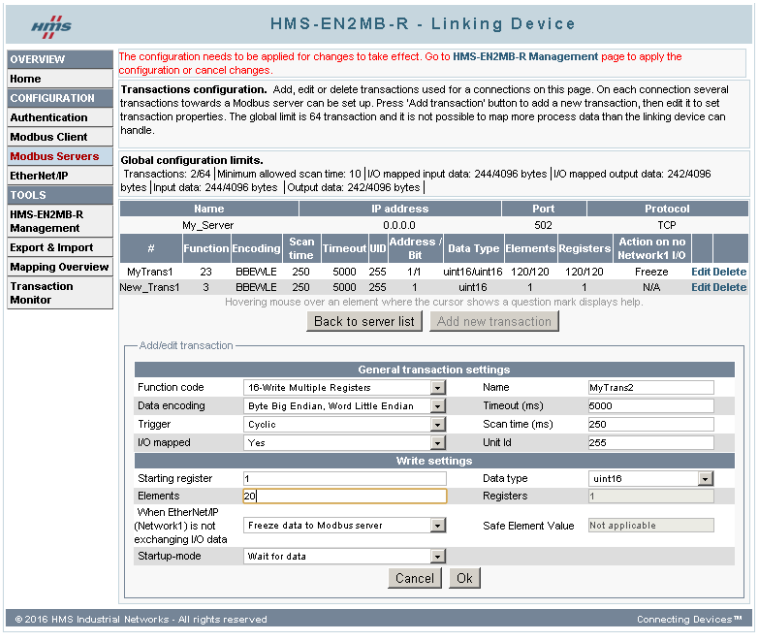
This will create a transaction which both reads and writes 120 elements (240 bytes).



9. Add another transaction:

- Name it “MyTrans2”.
- Choose Modbus function code 16 (Write Multiple Registers).
- Set the number of elements to 20.
- Click “Ok”.

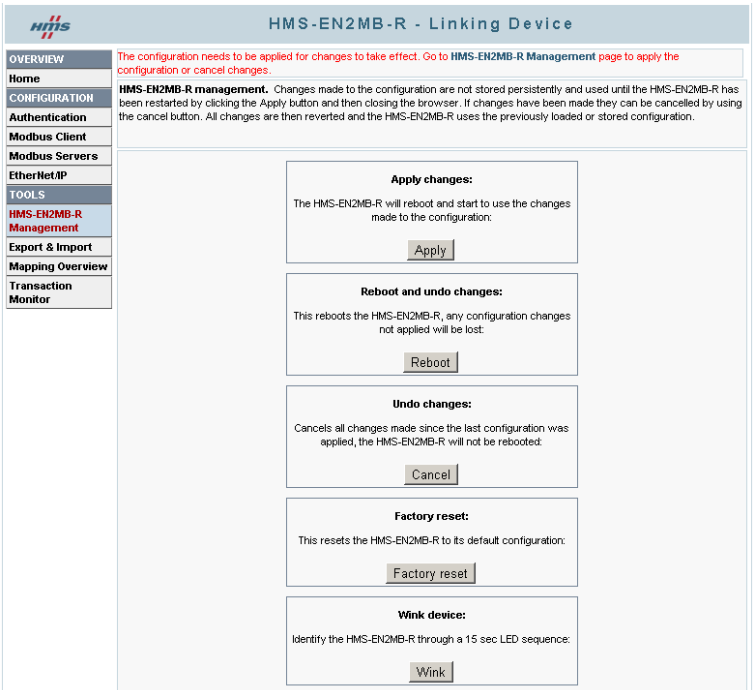
This will create a transaction which writes 20 elements (40 bytes).



10. Complete the configuration:

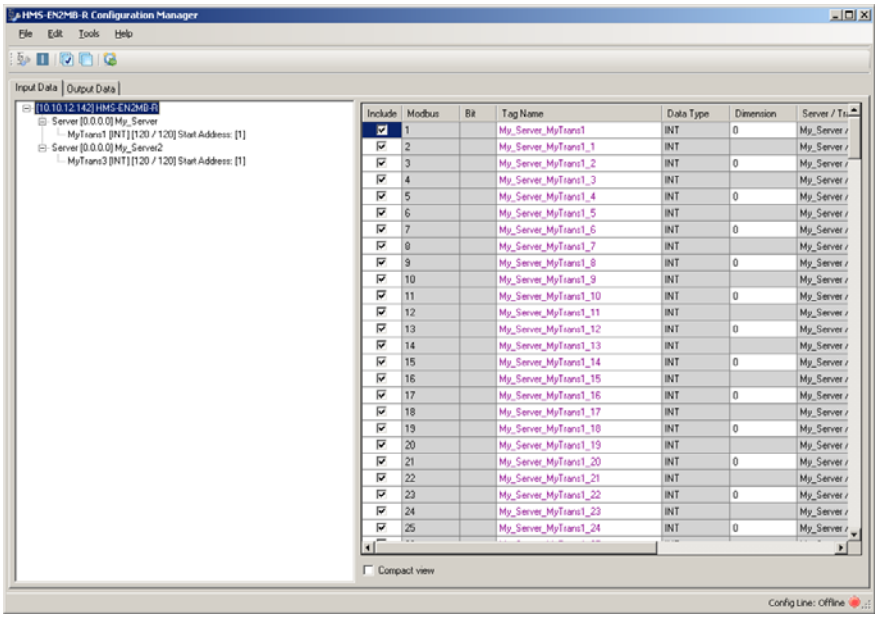
- Create another server and name it “My_Server2”.
- Add a transaction to this server, and click “Edit”.
- Name it “MyTrans3”.
- Choose Modbus function code 23 (Read/Write Multiple registers).
- In the “Elements” section, both in “Read settings” and “Write settings”, write 120.
- Click “Ok”.

11. When configuration is complete, find "HMS-EN2MB-R Management" in the menu bar to the left. "Click "Apply" in the Apply changes section, to download the configuration to the Linking Device.



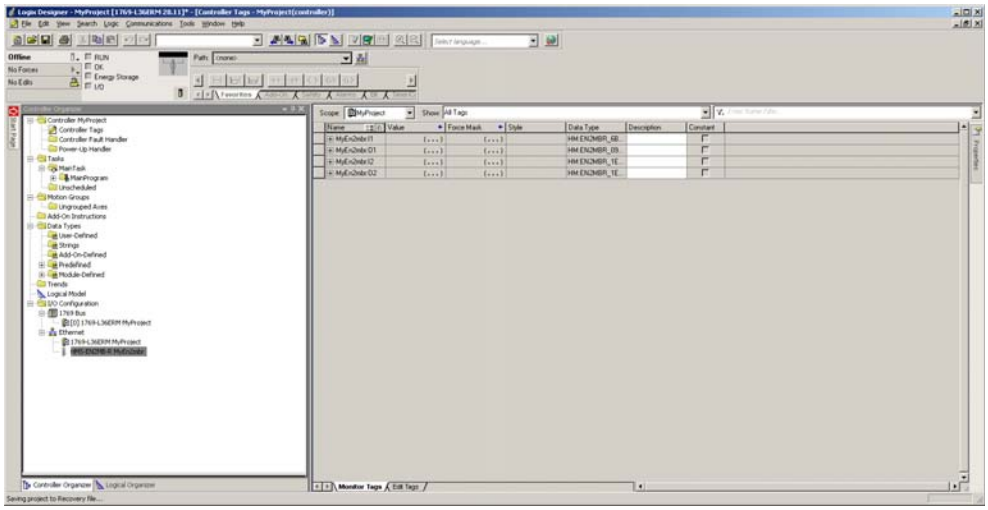
12. Close the HMS-EN2MB-R configuration pages window, to show the tag editor again.

13. The tag editor is now updated. It has automatically created named and structured tags from the Modbus configuration made in the previous steps.

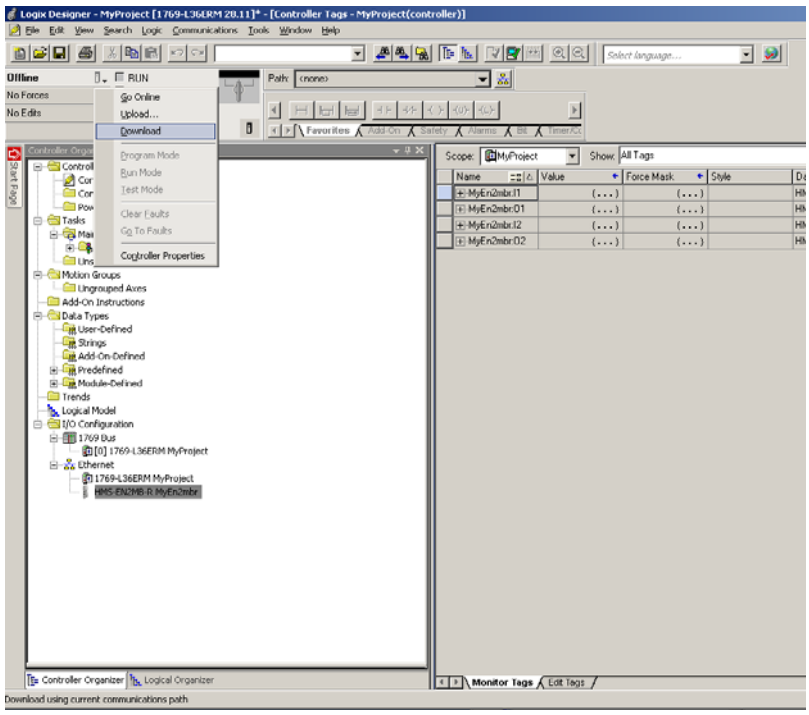


14. Close the tag editor.

15. The tags from the configuration are now imported into Studio 5000, as named and structured Studio 5000 controller tags.



16. Download the configuration to the Studio 5000 project by right-clicking the computer icon and then choosing "Download".



Chapter 4

4. SD Card Functionality

Using an SD card with the HMS-EN2MB-R linking device adds the following features:

- **Easy backup**.
  Every applied change in the configuration will automatically be saved to the linking device and the SD card. See "Easy Backup" on page 28.
- **Simple configuration copy**.
  Using the SD card, the configuration on one linking device can be copied to other linking devices. See "Simple Configuration Copy" on page 28.
- **Easy replacement**.
  If a linking device malfunctions during operation, a replacement device can easily be configured by moving the SD card to the new device. See "Easy Replacement" on page 29.

A configuration on the linking device is saved automatically to the SD card in any of these two events:

- A configuration is applied in the Management section
- A configuration is restored from a backup file

Important

The SD card acts as a master in the linking device. When an device is turned on with an SD card inserted, and that SD card contains a valid configuration file, the configuration on the SD card will always overwrite any configuration on the linking device.

4.1 General Advice and Guidelines

Turn the power off before inserting or removing an SD card from the linking device.

Do not turn the linking device off while the SD LED indicates that the SD card is being accessed. Refer to "Status LEDs" on page 17 for more information.

The linking device will not write any data to a write-protected SD card.

4.2 Starting Up

1. Format the SD card for the FAT file system using a PC. The linking device cannot use an unformatted SD card.
2. Make sure the SD card is empty and that it is not write-protected.
3. Turn the linking device off.
4. Insert the SD card into the SD card slot in the linking device.
5. Turn the linking device on.
6. Create the configuration. When finished, press the apply button in the Management section to reboot using the new configuration. During the reboot, the latest applied configuration will automatically be copied and saved to the SD card.
7. Now, the SD card is synchronized with the linking device. Both the SD card and the linking device contain the latest applied configuration.

Every time a new configuration is applied in the Management section, it is also copied to the SD card to ensure synchronization.

4.3 Easy Backup

Every time a configuration change is applied in the Management section using the configuration web pages, the configuration is saved to Studio 5000 and the Logix PLC memory, as well as to the linking device and the SD card. This is the easiest way of keeping a continuously updated configuration backup.

4.4 Simple Configuration Copy

If a configuration on one linking device needs to be copied to one or more other linking devices, it is easily done using an SD card.

1. Turn the linking device running the desired configuration off.
2. Remove the SD card from the linking device containing the desired configuration and insert it into another one.

   Note 1: The firmware version must be the same or higher in the new linking device.

   Note 2: The new linking device must support the same network type as the first linking device.
3. Turn the new linking device on. The new linking device will automatically start up using the configuration found on the SD card.

Important

If the configuration was protected by authentication information, the same information will be needed to alter the configuration in the new linking device.

4.5 Easy Replacement

If a linking device malfunctions during operation, the SD card functionality makes it easy to get the application up and running again fast.

1. Turn the malfunctioning linking device off.
2. Replace the linking device with a new one.

   Note 1: The firmware version must be the same or higher in the new linking device.

   Note 2: The new linking device must support the same network type as the old linking device.
3. Remove the SD card containing the configuration file from the old linking device and insert it into the new one.
4. Turn the new linking device on. If the SD card contains a valid configuration file, the linking device will automatically start up using the configuration found on the SD card.

Important

If the configuration was protected by authentication information, the same information will be needed to alter the configuration in the new linking device.

Depending on the settings of the master network, the communication link between the linking device and the master may no longer be valid. Linking Device settings that were configured from outside the configuration web pages will need to be set again.

4.6 SD Card Synchronization Failure

In the event of applying a configuration or restoring a configuration from a backup file, the SD card synchronization can fail. There are many possible reasons for an SD card write failure:

- The SD card is write-protected.
- The configuration file on the SD card is write-protected.
- The SD card memory is full.
- The SD card file system is corrupt.
- The SD card is damaged.

If the SD card write process fails, the reboot cycle of the linking device will halt. The GW LED will indicate "invalid configuration" and the SD LED will indicate "failure". See "Status LEDs" on page 17.

To eliminate the problem, follow the steps below:

1. Turn the linking device off.
2. Remove the SD card. Find the cause of the problem.
3. Insert an SD card.

   Note: This SD card must **not** contain a configuration file. If it does, the configuration on the SD card will overwrite the configuration on the linking device.
4. Turn the linking device on. The linking device will run the configuration that was applied or restored when the SD card write process failed.
5. Apply the configuration in the Management section to save the configuration to the SD card.
6. Now, the SD card is synchronized with the linking device. Both the SD card and the linking device contain the latest applied configuration.

Chapter 5

5. Modbus-TCP Functions

The Modbus-TCP protocol is an implementation of the standard Modbus protocol, running on top of TCP/IP. The same function codes and addressing model are used.

The HMS-EN2MB-R Linking Device supports a subset of the functions described in the Modbus-TCP specification.

Modbus-TCP transactions are normally transmitted and received on TCP port no. 502. The linking device features the possibility to set TCP ports individually for each Modbus-TCP server.

For detailed information regarding the Modbus-TCP protocol, consult the Open Modbus-TCP Specification.

The EtherNet/IP to Modbus-TCP Linking Device supports the following Modbus-TCP functions:

Modbus Function	Function Code	No. of Bits/Registers[a]	Direction	Associated with Buffer
Read Coils	1	1-2000	Modbus to linking device	Input buffer
Read Discrete Inputs	2	1-2000		
Read Holding Registers	3	1-125		
Read Input Registers	4	1-125		
Write Single Coil	5	1	Linking device to Modbus	Output buffer
Write Single Register	6	1		
Write Multiple Coils	15	1-1968		
Write Multiple Registers	16	1-123		
Read/Write Multiple Registers	23	1-125 read 1-121 write	Bidirectional	Input and output buffers

a. Please refer to the Modbus Application Protocol Specification V1.1B for more detailed information.

Modbus-TCP functions are used as important parts of transactions to Modbus-TCP servers. After configuring a server within the Modbus-TCP network, functions can be assigned to it by clicking the 'Add transaction' button in the built-in web interface.

See also...

- "Modbus Configuration Manager" on page 38
- "Modbus Servers" on page 43

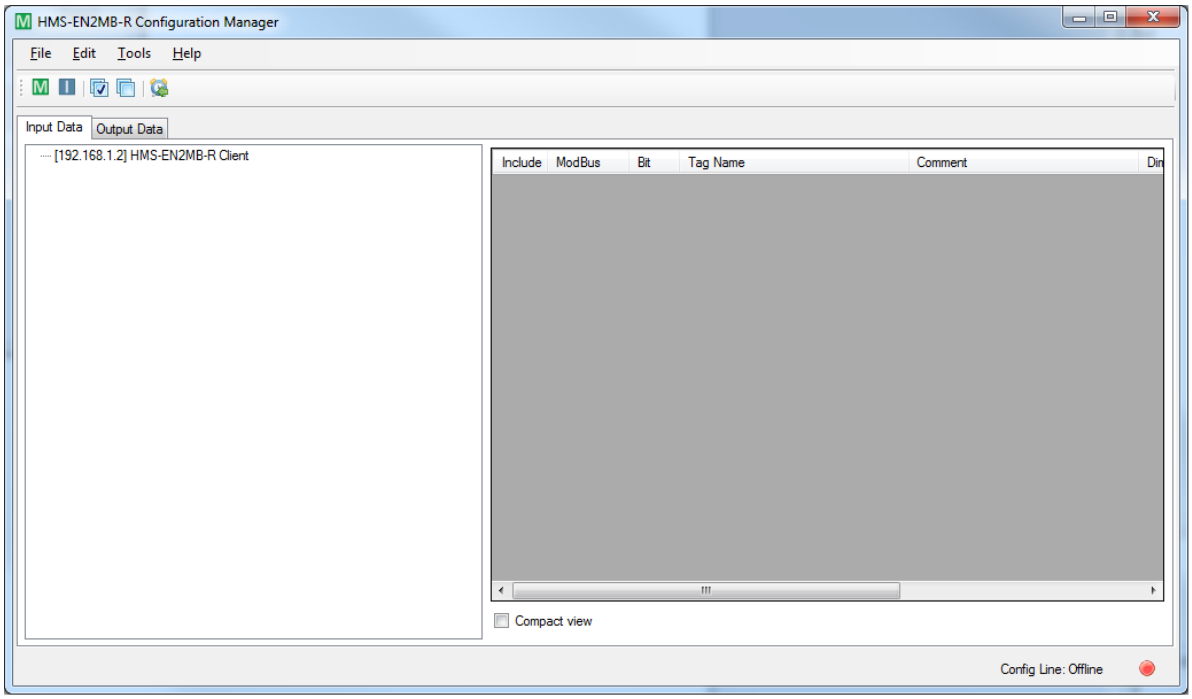
6. Tag Editor

The tag editor is supported by Windows 7 and later.

6.1 Tag Editor Overview

Initially, the tag editor looks like this.

The "Config Line" text in the lower right corner tells if the PLC is online or offline. No configuration is possible while the PLC is online.



6.1.1 Menu Choices

File

- **Import**
  Import a configuration from the hard drive.
- **Export**
  Export a configuration and save for future use.
- **Exit**

Edit

- **Select All**
- **Deselect All**

Tools

- **Edit Modbus Configuration**
  This option will launch the Modbus configuration manager.
  For more information, see the "Modbus Configuration Manager" on page 38.
- **Launch IPconfig**
  With the IPconfig tool, it is possible to scan for devices and alter IP settings for the linking device.
- **Create Mapping Report**
  Saves a detailed list of all tags and their mappings on the hard drive. For documentation and error tracking.
- **Factory Reset**
  Resets the linking device to default values.
- **Reboot**
- **Restore Configuration**
  Loads the latest stored configuration from Studio 5000.
- **Wink**
  Winking the device will make all left LEDs blink for a short while.
- **Options**
  - Hide report on retrieve configuration from gateway.
  - Group tags on PLC (by Modbus server)

Help

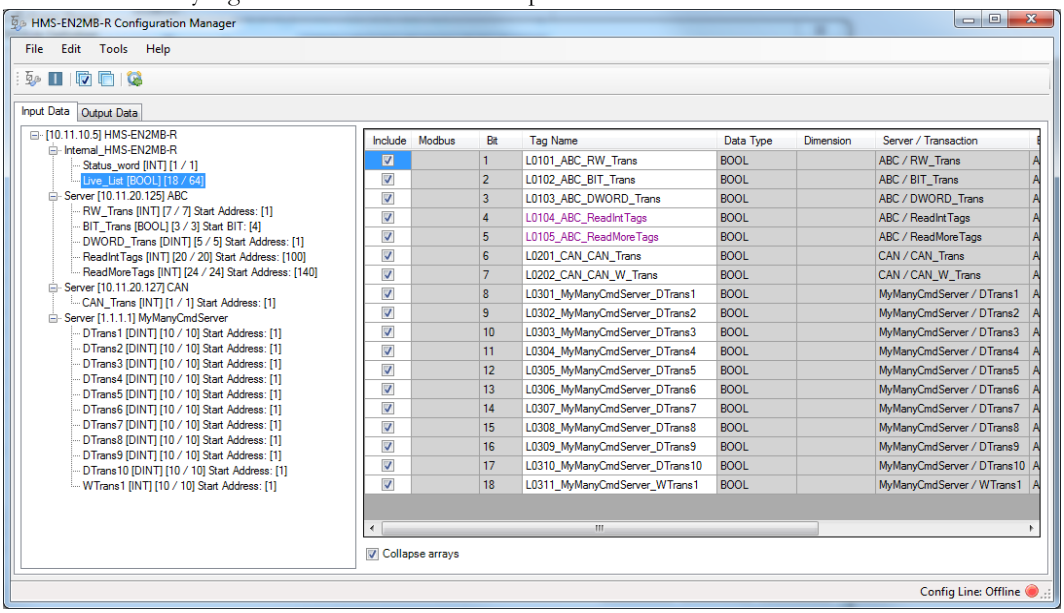
- **About**

6.2 Tag Editor Basics

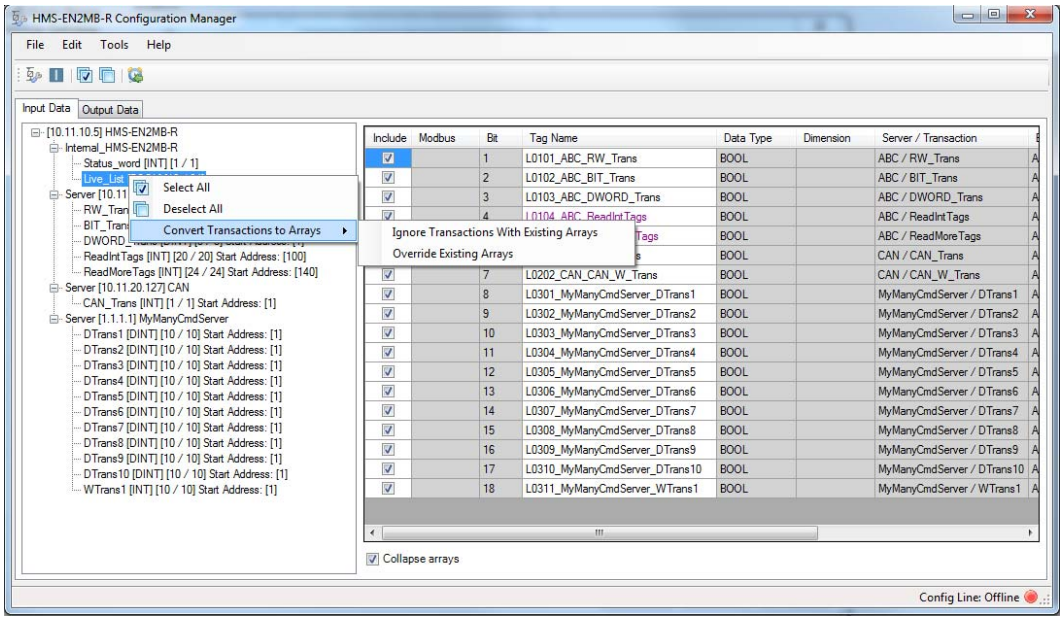
When the Modbus configuration manager is closed after configuration, the tag editor will be filled with resulting Studio 5000 tags.

Tags that are new or altered since last time, will be presented in magenta color.

The “Include” column makes it possible to choose which tags should be exported to the Studio 5000 environment. Only tags that are checked will be exported.



Right-clicking on a server or a transaction in the tree structure will present the menu below.



- **Select All**

  Checks the "Include" check box for all tags associated with that server/transaction

- **Deselect All**

  Unchecks the "Include" check box for all tags associated with that server/transaction

- **Convert Transactions to Arrays**

  Attempts to create the biggest possible array for every transaction in that tree branch

  - Ignore Transactions With Existing Arrays
    If there are already arrays in the transaction/transactions, they will not be affected.
  - Override Existing Arrays
    Existing arrays may be affected

6.2.1 Tag Arrays

Arranging tags into arrays is a good way to get a better overview of the configuration, since a big configuration may result in a large number of tags.

Creating an array is easy. Writing the array size in the dimension field of the first tag of the array, will allocate that tag and the following tags to the array. All tags except the first element of the array will then become uneditable. Original tag names will be remembered by the tag editor.

Include	Modbus	Bit	Tag Name	Data Type	Dimension	Server / Transaction	EtherN
☑	100		ABC_ReadIntTags	INT	4	ABC / ReadIntTags	Ass:100
☑	101			INT		ABC / ReadIntTags	Ass:100
☑	102			INT		ABC / ReadIntTags	Ass:100
☑	103			INT		ABC / ReadIntTags	Ass:100
☑	104		ABC_ReadIntTags_4	INT	0	ABC / ReadIntTags	Ass:100
☑	105		ABC_ReadIntTags_5	INT		ABC / ReadIntTags	Ass:100
☑	106		ABC_ReadIntTags_6	INT	0	ABC / ReadIntTags	Ass:100
☑	107		ABC_ReadIntTags_7	INT		ABC / ReadIntTags	Ass:100
☑	108		ABC_ReadIntTags_8	INT	6	ABC / ReadIntTags	Ass:100
☑	109			INT		ABC / ReadIntTags	Ass:100
☑	110			INT		ABC / ReadIntTags	Ass:100
☑	111			INT		ABC / ReadIntTags	Ass:100
☑	112			INT		ABC / ReadIntTags	Ass:100
☑	113			INT		ABC / ReadIntTags	Ass:100
☑	114		ABC_ReadIntTags_14	INT	0	ABC / ReadIntTags	Ass:100
☑	115		ABC_ReadIntTags_15	INT		ABC / ReadIntTags	Ass:100

To remove the array, just set the dimension of the first array element back to 0 (zero).

For even more overview of the configuration, there is the possibility to collapse tags. This will present a view of the configuration where all arrays are only represented by its first element, according to the picture below.

Include	Modbus	Bit	Tag Name	Data Type	Dimension	Server / Transaction	EtherN
☑	100		ABC_ReadIntTags	INT	4	ABC / ReadIntTags	Ass:100
☑	104		ABC_ReadIntTags_4	INT	0	ABC / ReadIntTags	Ass:100
☑	105		ABC_ReadIntTags_5	INT		ABC / ReadIntTags	Ass:100
☑	106		ABC_ReadIntTags_6	INT	0	ABC / ReadIntTags	Ass:100
☑	107		ABC_ReadIntTags_7	INT		ABC / ReadIntTags	Ass:100
☑	108		ABC_ReadIntTags_8	INT	6	ABC / ReadIntTags	Ass:100
☑	114		ABC_ReadIntTags_14	INT	0	ABC / ReadIntTags	Ass:100
☑	115		ABC_ReadIntTags_15	INT		ABC / ReadIntTags	Ass:100
☑	116		ABC_ReadIntTags_16	INT	0	ABC / ReadIntTags	Ass:100
☑	117		ABC_ReadIntTags_17	INT		ABC / ReadIntTags	Ass:100
☑	118		ABC_ReadIntTags_18	INT	0	ABC / ReadIntTags	Ass:100
☑	119		ABC_ReadIntTags_19	INT		ABC / ReadIntTags	Ass:100

☑ Collapse arrays

Check the check box “Collapse arrays” to collapse arrays. If there is a need to see all individual tags, uncheck the check box.

6.3 Tag Rule Definitions

IMPORTANT:
The default and automatically generated tags will follow and adhere to the rules below. They are only informative.

When adding process tags (controller tags) to the configuration the following rules must apply:

- The designated memory address of the process tag must be inside the process data area boundaries.
- The memory address of the process tag must be divisible by four, for data types DINT or REAL or a tag whose dimension is above zero.
- The memory address of the process tag must be dividable by two for data type INT.
- Two process tags can not occupy the same memory address area on the EtherNet/IP side.
- Two process tags can address the same memory address on the Modbus-TCP side.
- The designated memory address of the process tag must smaller than or equal to the corresponding designated memory address in the configuration.

7. Modbus Configuration Manager

7.1 General Information

The EtherNet/IP to Modbus-TCP Linking Device features built-in configuration pages for easy Modbus configuration. To access the configuration pages, the following system requirements need to be met:

- Javascript enabled

There are things to take into consideration when making the configuration.

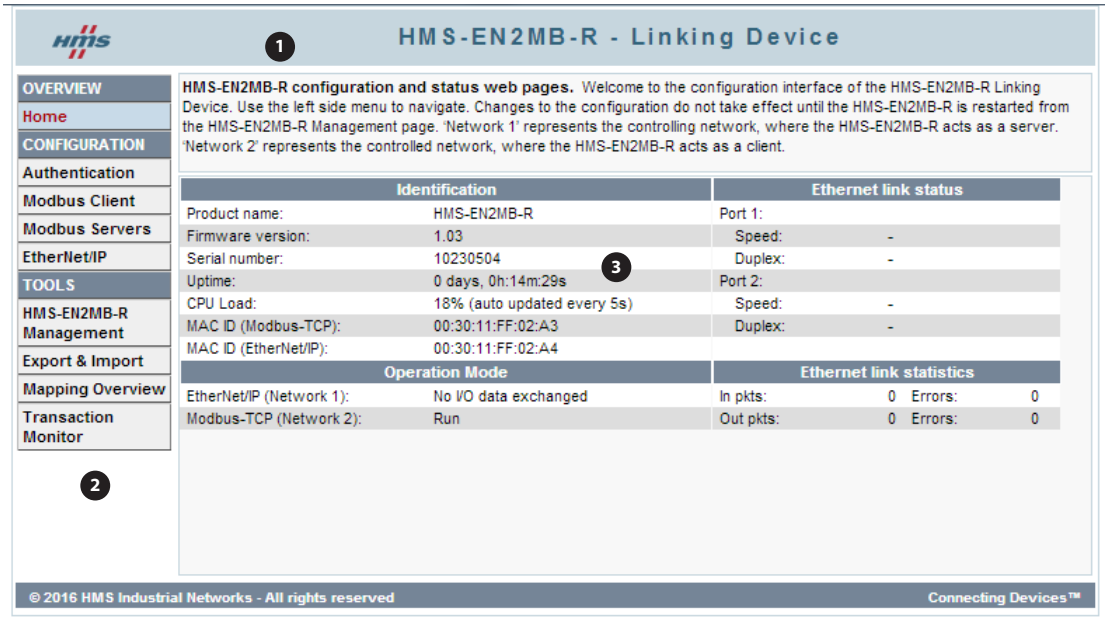
- Remember to apply the configuration in order for changes to take effect. See "HMS-EN2MB-R Management" on page 47. As soon as you have saved data to the configuration but not yet applied it, you will see the box below at the top of the web pages:

- A maximum of 64 Modbus-TCP servers can be added to the configuration.
- A maximum of 64 transactions can be set up to the servers in the configuration.
- The data limits are 4096 bytes input data and 4096 bytes output data, including optional control/status word and live list.
- Modbus transactions are assembled in multiple Class 1 Exclusive Owner (IO) Connections, on the EtherNet/IP side. There are a maximum of 10 available connections (with an associated input and output connection parameter per each) available. Connections will be made as needed. Transactions will be added to connections in the order they are created in the Modbus configuration manager. If a transaction does not fit into the remaining parts of a connection, it will be moved to the next connection. The bytes left in the previous connection will then be padded and left unused. Consequently, take care when creating the Modbus transactions, to optimize the available memory in the input and output IO areas as much as possible.

7.2 Overview

The Modbus configuration and status pages are divided into three sections:



1. **Headline Section**

   Shows the HMS logo and the name of the product.

2. **Navigation Section**

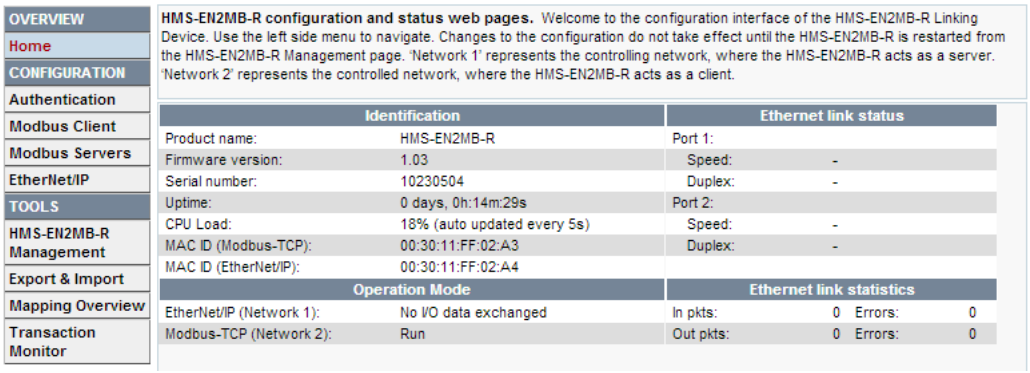
   All functionality is easily accessed from the different links. Every link and its corresponding functionality will be explained later in this chapter.

3. **Content Section**

   Clicking a link will display its contents in the content section. A short text describing the functionality of the current page will be available at the top of the section.

7.2.1 Home

The introductory window of the configuration and status pages presents important error tracking information, as well as general information and statistics.



Operation Mode

The table below shows the correlation between the operation modes of the Modbus-TCP network and the EtherNet/IP network.

		EtherNet/IP (Network 1)	
		I/O data exchanged	No I/O data exchanged
Modbus-TCP (Network 2)	Run	Data is exchanged between the two networks.	The EtherNet/IP network exchanges no data. Data to the Modbus-TCP network is in clear, freeze, safe value or stop state.
	Idle	The Modbus-TCP network exchanges no data. Data to the EtherNet/IP network is in clear or freeze state.	No data is exchanged. Both networks, independently, are in clear, freeze, safe value or stop state.

In case of an error on the EtherNet/IP network, the following additional statuses may appear:

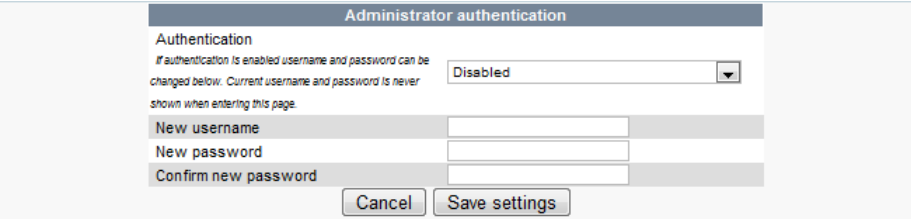
EtherNet/IP (Network 1)	Description
Error	Class 1 connection error. Duplicate IP address detected.
Shutdown	Unexpected error (the linking device needs a restart).

7.3 Configuration

Please note that changes made to the configuration will not be used by the linking device until they have been applied and saved. See "HMS-EN2MB-R Management" on page 47.

7.3.1 Authentication

Authentication can be enabled or disabled. If enabled, it is possible to set a username and password to protect the configuration.



When choosing a username and a password, use only the valid characters shown below.

Item	Valid characters
Username	A-Z, a-z, 0-9, _ (underscore). Max length: 13 characters.
Password	A-Z, a-z, 0-9, _ (underscore). Max length: 12 characters.

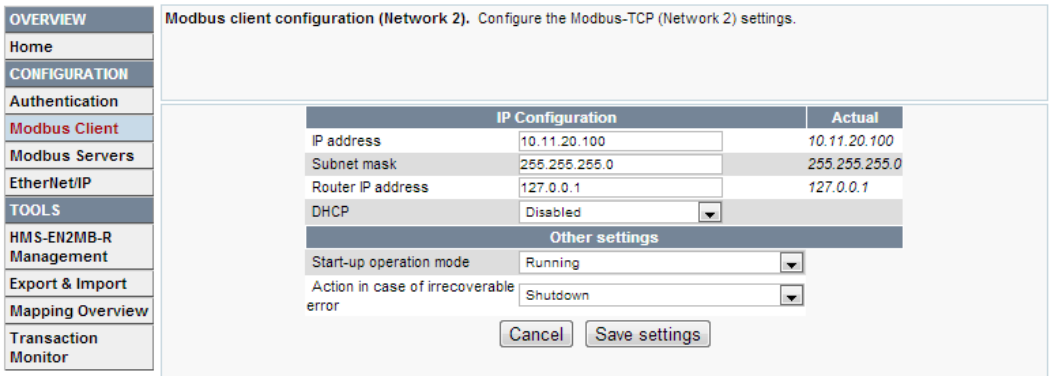
Important Notice

Note that when restoring a configuration from Studio 5000, or when importing a configuration (using either the tag editor or the configuration pages), authentication has to be disabled. Restore or import is only possible when authentication is disabled.

Note also that it is very important to save the authentication information. There is no way to retrieve a lost username or password. If the authentication information is lost, the only way to restore the linking device is to download new firmware via the USB interface. This will erase any configuration currently on the module.

7.3.2 Modbus Client

Configuration of the client side of the Modbus-TCP network. On this side, the linking device will act as a Modbus-TCP client. To the right, in the "Actual" column, the currently used values can be seen.



Available IP Configuration Settings

Item	Description
IP address	If not set by DHCP (or HICP), set these values manually.
Subnet mask	
Router IP address	
DHCP	Enabled by default. When enabled, the linking device can obtain the TCP/IP settings dynamically from the DHCP server of the Modbus-TCP network.

Start-up Operation Mode

Value	Description
Running	The Modbus-TCP client starts to exchange data with the servers as soon as possible after start-up.
Idle	The Modbus-TCP client does not exchange any data with the servers and waits for instructions via the control word.

Action in Case of Irrecoverable Error

If the linking device encounters an irrecoverable error, there are two possible options.

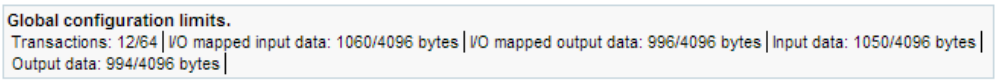
Value	Description
Shutdown	The linking device will shut down. All LEDs will display red.
Restart	The linking device will restart.

When finished configuring the Modbus-TCP client, click 'Save settings'. Note that the changes will not take effect until they are applied in the HMS-EN2MB-R management section. See "HMS-EN2MB-R Management" on page 47.

7.3.3 Modbus Servers

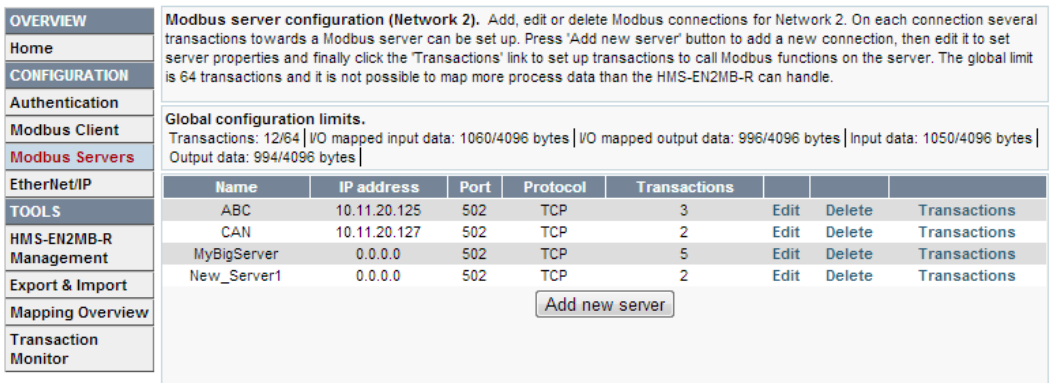
The configuration of the servers on the Modbus-TCP network is made here. The linking device can handle up to 64 different servers, and a maximum of 64 transactions distributed among those servers. It is possible to map up to 4096 bytes of data in either direction, including control/status word and live list.

The global configuration limits box keeps track of the number of added transactions and the current amount of I/O mapped input and output data. It also keeps track of the total amount of data in the configuration (both I/O mapped and not I/O mapped data).



Add Server

Click 'Add server' to add a server to the configuration. Click 'Edit' to see and edit the settings:



Available editable settings:

Setting	Description
Name	While not required, renaming the server makes the configuration easier to comprehend. Note that it is only possible to use uppercase and lowercase characters, numerals and underscore (_). Default name is 'New_Server', followed by an incremental suffix. Max length: 32 characters.
Server address	The IP address of the server.
Protocol	TCP.
Port	Default Modbus-TCP port is 502. If the server requires it, it is possible to change. Value range: 0 - 65535.

When the server is configured, click 'Ok'.

Note: When the server and its settings are configured, transactions must be added to the server. See "Add Transactions" on page 44. At any time, it is possible to have only one server without specified transactions.

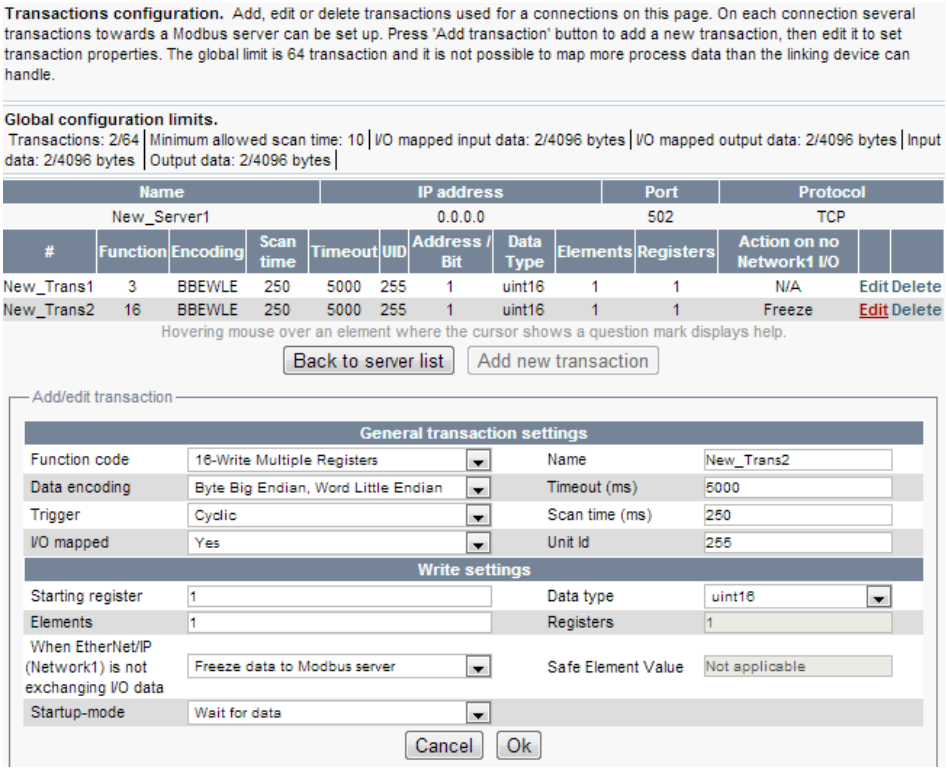
Add Transactions

Transactions represent the data that is read from/written to the servers of the Modbus-TCP network.

The global configuration limits box keeps track of the number of added transactions, the current minimum allowed scan time, and the current amount of I/O mapped data as well as total amount of data (both I/O mapped and not I/O mapped data).

Global configuration limits.
Transactions: 12/64 | I/O mapped input data: 1060/4096 bytes | I/O mapped output data: 996/4096 bytes | Input data: 1050/4096 bytes | Output data: 994/4096 bytes |

To add transactions, find the server in the server list and click 'Transactions'. This presents a list of all transactions configured for that server. Click 'Add transaction' to add a new default transaction to the list and click 'edit'.



Transactions configuration. Add, edit or delete transactions used for a connections on this page. On each connection several transactions towards a Modbus server can be set up. Press 'Add transaction' button to add a new transaction, then edit it to set transaction properties. The global limit is 64 transaction and it is not possible to map more process data than the linking device can handle.

Global configuration limits.
Transactions: 2/64 | Minimum allowed scan time: 10 | I/O mapped input data: 2/4096 bytes | I/O mapped output data: 2/4096 bytes | Input data: 2/4096 bytes | Output data: 2/4096 bytes |

Name	IP address	Port	Protocol
New_Server1	0.0.0.0	502	TCP

#	Function	Encoding	Scan time	Timeout	UID	Address / Bit	Data Type	Elements	Registers	Action on no Network1 I/O		
New_Trans1	3	BBEWLE	250	5000	255	1	uint16	1	1	N/A	Edit	Delete
New_Trans2	16	BBEWLE	250	5000	255	1	uint16	1	1	Freeze	Edit	Delete

Hovering mouse over an element where the cursor shows a question mark displays help.

Back to server list | Add new transaction

Add/edit transaction

General transaction settings			
Function code	16-Write Multiple Registers	Name	New_Trans2
Data encoding	Byte Big Endian, Word Little Endian	Timeout (ms)	5000
Trigger	Cyclic	Scan time (ms)	250
I/O mapped	Yes	Unit Id	255
Write settings			
Starting register	1	Data type	uint16
Elements	1	Registers	1
When EtherNet/IP (Network1) is not exchanging I/O data	Freeze data to Modbus server	Safe Element Value	Not applicable
Startup-mode	Wait for data		

Cancel | Ok

See a description of the settings on the next page.

Available settings

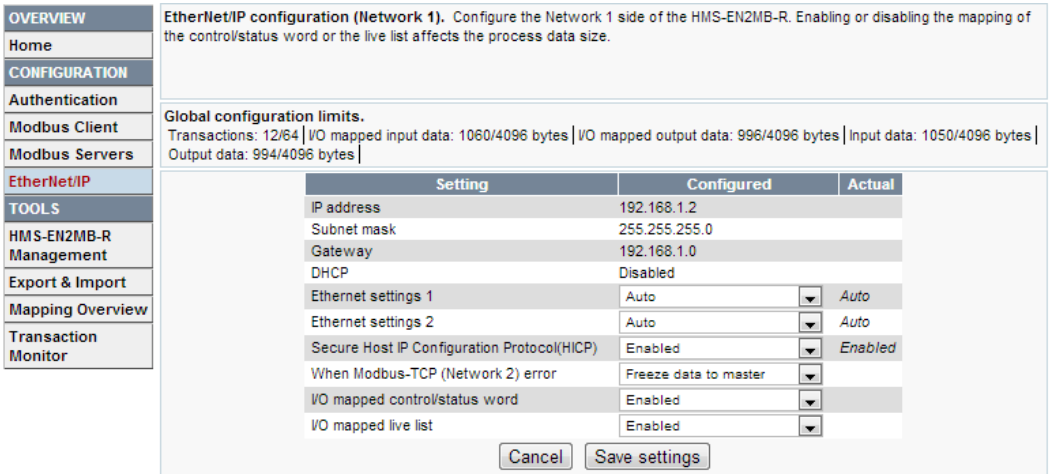
Setting	Description
Function code	The function code defines the purpose of the transaction. Choose from the available different Modbus functions, see “Modbus-TCP Functions” on page 31.
Data encoding	Decides in what order the different bytes of the received/transmitted data shall be sent on the Modbus network.
Trigger	Only applicable for write transactions. Cyclic. On data change.
I/O mapped	Decides whether to map the data to the memory that is cyclically exchanged between the EtherNet/IP network and the Modbus-TCP network (I/O mapped data). Note: data that is not I/O mapped will be exchanged acyclically.
Name	While not required, renaming the transaction makes the configuration easier to comprehend. On the EtherNet/IP side, this name will also propagate to tag name, for identification purposes. Note that it is only possible to use uppercase and lowercase characters, numerals and underscore (_). Default name is ‘New_Trans’, followed by an incrementing suffix. Max length: 32 characters.
Timeout (ms)	The time span within which the server must return a response to the transaction. If no response is received within the timeout period, the connection to the server will be closed. If the connection to the server is closed, all transactions to that server will be affected. Value range: 10 - 65535 (ms).
Scan time (ms)	The scan time defines how often the transaction shall be resent, e.g. the time cycle of a repeating transaction. Minimum scan time (ms) is calculated by multiplying the total number of transactions by three. The minimum scan time will increase by adding more transactions. Value range: 10 - 10000 (ms).
Unit ID	Only applicable for Modbus RTU servers. If the Modbus-TCP server work as a router to Modbus RTU servers, it is possible to send transactions to a single Modbus RTU server using the unit ID. Value range: 0 - 247; 255. If not communicating with a Modbus RTU server, use the value 255 (default).
Starting register	The starting Modbus server register or bit to write to/read from. Value range: 1 - 65536.
Elements	The number of elements to write/read. Value range: See “Modbus-TCP Functions” on page 31.
When EtherNet/IP (Network 1) is not exchanging I/O data	Note: Only available for I/O mapped write transactions. Clear data to Modbus server: only zeros will be transmitted. Freeze data to Modbus server: the data that was stored last will be repeated. Write safe value: choose a specific value to transmit for every element (See safe element value below). Stop: no data will be transmitted to the Modbus server.
Data type	Write/read data either as two byte integers (uint16) or four byte integers (uint32).
Registers	The resulting amount of registers to write/read. The calculation is based on the number of elements to read/write and the chosen data type.
Safe Element Value	Note: Only available for write transactions. A numeric value to send for every element if network 1 (EtherNet/IP) is not exchanging I/O data.
Startup-mode	Wait for data: all data for the transaction must have been sent from the EtherNet/IP network and received by the linking device before the transaction is carried out. Directly: the data is sent as soon as possible after start-up.

When finished editing the transaction, click ‘Ok’. All data resulting from configured transactions will be mapped to the internal memory of the linking device. Read transactions will be mapped to the input area, and write transactions will be mapped to the output area. See “Mapping Overview” on page 48 for more information.

Note: The linking device needs to be restarted before any changes will take effect. See “HMS-EN2MB-R Management” on page 47.

7.3.4 EtherNet/IP (Adapter Interface)

Configuration of the EtherNet/IP adapter interface of the linking device.



What is shown is the currently stored configuration, provided that all changes are saved and applied to the linking device.

The column 'Actual' presents the settings that are currently used.

It is possible to override the TCP/IP and Ethernet settings set from the network by entering new values in the Configured column and pressing "Save settings".

Note that no changes will take effect until the configuration has been applied. See "HMS-EN2MB-R Management" on page 47.

Available settings for the EtherNet/IP network

Setting	Description
IP address	If not detected automatically, set these items manually.
Subnet mask	
Gateway	
DHCP	If enabled, the linking device can obtain TCP/IP settings from a DHCP server.
Ethernet settings 1	Network speed and communication model.
Ethernet settings 2	
Secure Host IP Configuration Protocol (HICP)	If enabled, the linking device can be found on the network using the IPconfig tool.
When Modbus-TCP (Network 2) error	The "Freeze data to master" option instructs the linking device to keep sending the latest received data from the Modbus-TCP network to the EtherNet/IP scanner.
	The "Clear data to master" option instructs the linking device to clear the input data area and send only zeros to the EtherNet/IP scanner.
I/O mapped control/status word	If enabled, the control/status word is mapped to the output/input area respectively. See "I/O Mapped Data" on page 10.
I/O mapped live list	If enabled, the live list is mapped to the input area. See "Live List" on page 11.

7.4 Tools

7.4.1 HMS-EN2MB-R Management

Apply changes

Permanently store changes made to the configuration and reboot, using the new configuration.

No changes made in the configuration will be permanently stored or used by the linking device until they are applied by clicking 'Apply'.

Before storing and rebooting, the linking device will validate the not yet stored configuration. If errors are found, the linking device will produce an information message with instructions to correct the errors. The linking device will not store an invalid configuration.

Reboot and undo changes

The linking device will be restarted. All changes made since the last configuration was loaded will be undone.

Undo changes

Undo all changes made since the last configuration was loaded.

Factory reset

Reset the linking device to completely remove the configuration currently stored in the module.

Wink device

Clicking the "Wink device" button will start a 15 second LED sequence on LEDs 1, 5 and 6 on the linking device. For identification purposes.

7.4.2 Backup and Restore

Backup the configuration that is currently used to file, or restore a previously saved configuration from file.

It is not possible to backup or restore the configuration until all changes are either applied or undone. See "HMS-EN2MB-R Management" on page 47.

Two things can happen when loading an old configuration:

- **Configuration valid:**
  The linking device will reboot and automatically use the previously stored configuration.
- **Configuration not valid:**
  The linking device will produce an error message. The chosen configuration will not be accepted or loaded into memory.

Important Notice

Before loading a previously stored configuration, locate any authentication information associated with it. If a valid configuration is loaded that is protected by a password, the linking device can not be reconfigured until the correct authentication information has been provided.

7.4.3 Mapping Overview

This page provides a description of all data resulting from the transactions of the currently applied configuration. It is divided into two parts. The first part describes the linking device interface to the EtherNet/IP network, and the second part all applied transactions on the Modbus-TCP network.

If needed, it is possible to print the configuration to paper. Click the printer symbol to the right on the mapping overview page to access a printer friendly version of the mapping overview.

EtherNet/IP

The I/O mapped data will always be presented according to the following priority order:

- **Input data**

  Data from the Modbus-TCP network to the EtherNet/IP network.
  - Status word (optional)
  - Live list (optional)
  - Input data (bit transactions will always be mapped first)

- **Output data**

  Data from the EtherNet/IP network to the Modbus-TCP network.
  - Control word (optional)
  - Output data (bit transactions will always be mapped first)

The parameter section data presents a detailed list of all data, including both the I/O mapped and the not I/O mapped data, available acyclically from the linking device to the EtherNet/IP network. This list also includes the transaction status and exception code lists, available for error identification.

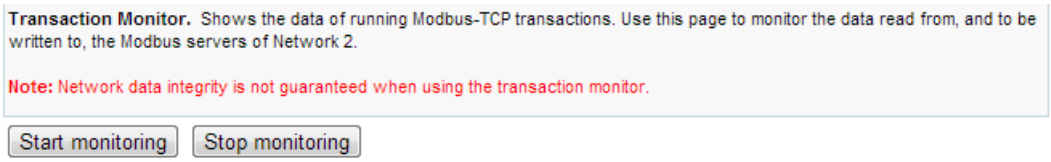
- "Exception Code List" on page 13
- "Transaction Status List" on page 12

Modbus-TCP network

A detailed list of all Modbus servers and transactions in the configuration.

7.4.4 Transaction Monitor

The transaction monitor interface presents a detailed list of all transactions currently operating on the Modbus-TCP network.



To start or stop the transaction monitor, press the desired button.

The data in the transaction monitor is automatically updated, and it is possible to choose to view the data either in decimal or in hexadecimal values. The time that has passed since the last update is visible at the top of the transaction list. Every post in the list contains the following transaction information:

- Server name and transaction name
- The type of Modbus function chosen for the transaction
- The size of the data read from or written to the Modbus-TCP network
- The actual data read from or written to the Modbus-TCP network
- The bit position of the transaction in the live list (also presented as byte.bit)

If there is a transaction error, an error message will appear instead of the data.

A red frame around the list indicates that the web browser has lost connection to the web server of the linking device. If this happens, try reloading the page by clicking on "Transaction Monitor" in the menu to the left.

Note: Viewing the transaction monitor may affect performance.

8. CIP Objects

8.1 General Information

This chapter specifies the CIP-object implementation in the linking device. These objects can be accessed from the network.

Mandatory Objects:

Manufacturer Specific Objects:

8.2 Identity Object (01h)

Object Description

-

Supported Services

Class:	Get_Attribute_Single Get_Attributes_All
Instance:	Get_Attribute_Single Set_Attribute_Single Get_Attributes_All Reset

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0001h (Object revision)

Instance #1 Attributes

#	Name	Access	Type	Value
1	Vendor ID	Get	UINT	005Ah (HMS Industrial Networks AB)
2	Device Type	Get	UINT	000Ch (Communications Adapter)
3	Product Code	Get	UINT	0057h (EtherNet/IP to Modbus-TCP Linking Device)
4	Revision	Get	Struct of: {USINT, USINT}	Major and minor firmware revision
5	Status	Get	WORD	
6	Serial Number	Get	UDINT	Unique serial number (assigned by HMS)
7	Product Name	Get	SHORT_STRING	“EtherNet/IP to Modbus-TCP Linking Device” (Name of product)
11	Active language	Set	Struct of: USINT USINT USINT	
12	Supported Language List	Get	Array of: Struct of: USINT USINT USINT	

Device Status

Bit(s)	Name
0	Module Owned
1	(reserved)
2	Configured[a]
3	(reserved)
4... 7	Extended Device Status: Value:Meaning: 0000b Unknown 0010b Faulted I/O Connection 0011b No I/O connection established 0100b Non-volatile configuration bad 0110b Connection in Run mode 0111b Connection in Idle mode (other) (reserved)
8	Set for minor recoverable faults
9	Set for minor unrecoverable faults
10	Set for major recoverable faults
11	Set for major unrecoverable faults
12... 15	(reserved)

a. This bit shows if the product has other settings than “out-of-box”.

Service Details: Reset Service

There are two types of network reset requests on EtherNet/IP:

- **Type 0: ‘Power Cycling Reset’**

  This service emulates a power cycling of the module, and corresponds to Anybus reset type 0 (Power cycling).

- **Type 1: ‘Out of box reset’**

  This service sets a “out of box” configuration and performs a reset, and corresponds to Anybus reset type 2 (Power cycling + factory default).

8.3 Message Router (02h)

Object Description

-

Supported Services

Class: -

Instance: -

Class Attributes

-

Instance Attributes

-

8.4 Assembly Object (04h)

Object Description

The Assembly object uses static assemblies and holds the Process Data sent/received by the host application. The default assembly instance IDs used are in the vendor specific range.

Supported Services

Class: Get_Attribute_Single

Instance: Get_Attribute_Single
Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0002h (Object revision)
2	Max Instance	Get	UINT	(Highest instance number)

Instance C6h Attributes (Heartbeat, Input-Only)

This instance is used as heartbeat for Input-Only connections.

#	Name	Access	Type	Value
3	Data	Set	N/A	- (The data size of this attribute is zero)

Instance C7h Attributes (Heartbeat, Listen-Only)

This instance is used as heartbeat for listen-only connections.

#	Name	Access	Type	Value
3	Data	Set	N/A	- (The data size of this attribute is zero)

Instance 05h Attributes (Configuration Data)

Configuration Data that is sent through the 'Forward_Open'-service will be written to this instance.

#	Name	Access	Type	Value
3	Data	Get/Set	N/A	- (Configuration data written to the application when the forward open command has the configuration data included)

Instance C3h Attributes (Heartbeat, Input-Only Extended)

This instance is used as heartbeat for input-only extended connections, and does not carry any attributes. If the connection times out, the module does not switch to the Error state.

Instance C4h Attributes (Heartbeat, Listen-Only Extended)

This instance is used as heartbeat for listen-only extended connections, and does not carry any attributes.

Instance 64h Attributes (Producing Instance)

#	Name	Access	Type	Value
3	Produced Data	Get	Array of BYTE	This data corresponds to the Write Process Data

Instance 96h Attributes (Consuming Instance)

#	Name	Access	Type	Value
3	Consumed Data	Set	Array of BYTE	This data corresponds to the Read Process Data

8.5 Connection Manager (06h)

Object Description

-

Supported Services

Class:	-
Instance:	Forward_Open Forward_Close

Instance Descriptions

(No supported instance attributes)

Class 1 Connection Details

General

Class 1 connections are used to transfer I/O data, and can be established to instances in the Assembly Object. Each Class 1 connection will establish two data transports; one consuming and one producing. The heartbeat instances can be used for connections that shall only access inputs. Class 1 connections use UDP transport.

- Total number of supported Class 1 connections: 4
- Max input connection size: 256 bytes
- Max output connection size: 256 bytes
- Supported API: 2... 3200 ms
- T $\Rightarrow$ O Connection type: Point-to-point, Multicast
- O $\Rightarrow$ T Connection type: Point-to-point
- Supported trigger types: Cyclic, COS

Connection Types

- **Exclusive-Owner connection**

  This type of connection controls the outputs of the Anybus module and does not depend on other connections.

  - Max. no. of Exclusive-Owner connections: 1
  - Connection point O ⇒ T: Assembly Object, instance 64h (Default)
  - Connection point T ⇒ O: Assembly Object, instance 96h (Default)

- **Input-Only connection**

  This type of connection is used to read data from the Anybus module without controlling the outputs. It does not depend on other connections.

  - Max. no. of Input-Only connections: Up to 4[1]
  - Connection point O ⇒ T: Assembly Object, instance 03h (Default)
  - Connection point T ⇒ O: Assembly Object, instance 96h (Default)

  Note: If an Exclusive-Owner connection has been opened towards the module and times out, the Input-Only connection times out as well. If the Exclusive-Owner connection is properly closed, the Input-Only connection remains unaffected.

- **Input-Only Extended connection**

  This connections functionality is the same as the standard Input-Only connection. However when this connection times out it does not affect the state of the application.

  - Max. no. of Input-Only connections: Up to 4[a]
  - Connection point O ⇒ T: Assembly Object, instance 05h (Default)
  - Connection point T ⇒ O: Assembly Object, instance 96h (Default)

  Note: If an Exclusive-Owner connection has been opened towards the module and times out, the Input-Only connection times out as well. If the Exclusive-Owner connection is properly closed, the Input-Only connection remains unaffected.

- **Listen-Only connection**

  This type of connection requires another connection in order to exist. If that connection (Exclusive-Owner or Input-Only) is closed, the Listen-Only connection will be closed as well.

  - Max. no. of Listen-Only connections: Up to 4[2]
  - Connection point O ⇒ T: Assembly Object, instance 04h (Default)
  - Connection point T ⇒ O: Assembly Object, instance 96h (Default)

- **Listen-Only Extended connection**

  The functionality of this connection is the same as the standard Listen-Only connection. However when this connection times out it does not affect the state of the application.

  - Max. no. of Listen-Only connections: Up to 4[b]
  - Connection point O ⇒ T: Assembly Object, instance 07h (Default)
  - Connection point T ⇒ O: Assembly Object, instance 96h (Default)

- **Redundant-Owner connection**

  This connection type is not supported by the module.

1. Shared with Exclusive-Owner and Listen-Only connections
2. Shared with Exclusive-Owner and Input-Only connections

Class 3 Connection Details

- **Explicit message connection**

  Class 3 connections are used to establish connections towards the message router. Thereafter, the connection is used for explicit messaging. Class 3 connections use TCP transport.

  - No. of simultaneous Class 3 connections: 16
  - Supported API: 2 - 10000 ms
  - T ⇒ O Connection type: Point-to-point
  - O ⇒ T Connection type: Point-to-point
  - Supported trigger type: Application

8.6 DLR Object (47h)

Object Description

-

Supported Services

Class:	Get_Attribute_Single Get_Attributes_All
Instance:	Get_Attribute_Single Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0002h (Object revision)

Instance #1 Attributes

#	Name	Access	Type	Value
1	Network Topology	Get	USINT	Bit:Contents: 0 “Linear” 1 “Ring”
2	Network Status	Get	USINT	Bit:Contents: 0 “Normal” (N/A) 1 “Ring Fault” 2 “Unexpected Loop Detected” 3 “Partial Network Fault” 4 “Rapid Fault/Restore Cycle”
10	Active Supervisor Address	Get	Struct of: {UDINT, Array of 6 USINTs}	This attribute holds the IP address (IPv4) and/or the Ethernet Mac address of the active ring supervisor
12	Capability Flags	Get	DWORD	01h (Announce-based ring node)

8.7 QoS Object (48h)

Object Description

-

Supported Services

Class:	Get_Attribute_Single Get_Attributes_All
Instance:	Get_Attribute_Single Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0001h (Object revision)

Instance #1 Attributes

#	Name	Access	Type	Value
1	802.1Q Tag Enable	Set	USINT	Enables or disables sending 802.1Q frames Bit:Contents: 0 Disabled (Default) 1 Enabled
4	DSCP Urgent	Set	USINT	CIP transport Class 1 messages with priority Urgent Default: 55
5	DSCP Scheduled	Set	USINT	CIP transport Class 1 messages with priority Scheduled Default: 47
6	DSCP High	Set	USINT	CIP transport Class 1 messages with priority High Default: 43
7	DSCP Low	Set	USINT	CIP transport Class 1 messages with priority Low Default: 31
8	DSCP Explicit	Set	USINT	CIP UCMM and CIP Class 3 Default: 27

8.8 ADI Object (A2h)

Object Description

-

Supported Services

Class:	Get_Attribute_Single
Instance:	Get_Attribute_Single Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	Object revision (Current value = 0001h)
2	Max Instance	Get	UINT	
3	Number of instances	Get	UINT	

Instances Attributes

Each instance corresponds to an instance within the Application Data Object.

#	Name	Access	Type	Description
1	Name	Get	SHORT_STRING	Parameter name (including length)
2	ABCC Data type	Get	USINT	Data type of instance value
3	No. of elements	Get	USINT	Number of elements of the specified data type
4	Descriptor	Get	USINT	Bit field describing the access rights for this instance Bit:Meaning: 0 Set = Get Access 1 Set = Set Access
5	Value[a]	Get/Set	Determined by attribute #2	Instance value
6	Max value[a]	Get		The maximum permitted parameter value
7	Min value[a]	Get		The minimum permitted parameter value
8	Default value[a]	Get		The default parameter value

a. Converted to/from CIP standard by the module

8.9 Port Object (F4h)

Object Description

-

Supported Services

Class: Get_Attributes_All
Get_Attribute_Single

Instance: Get_Attributes_All
Get_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	Object revision (Current value = 0001h)
2	Max Instance	Get	UINT	Max. instance number
3	Number of instances	Get	UINT	Number of ports
8	Entry Port	Get	UINT	Returns the instance of the Port Object that describes the port through which this request entered the device.
9	Port Instance Info	Get	Array of:	Array of structures containing instance attributes 1 and 2 from each instance. The array is indexed by instance number, up to the maximum number of instances. The value at index 1 (offset 0) and any non-instantiated instances will be zero.
			Struct of: UINT (Type) UINT (Number)	 Enumerates the type of port CIP port number associated with this port

Instance Attributes (Instance #1)

This instance reflects the properties associated with the Ethernet interface.

#	Name	Access	Type	Value
1	Port Type	Get	UINT	0h (default)
2	Port Number	Get	UINT	2h
3	Link Object	Get	Struct of: UINT Padded EPATH	- 2h 20 F5 24 01h
4	Port Name	Get	SHORT_STRING	“EtherNet/IP”
7	Node Address	Get	Padded EPATH	-
8	Port Node Range	Get	Struct of: UINT (Min.) UINT (Max.)	-

8.10 TCP/IP Interface Object (F5h)

Object Description

The object groups TCP/IP-related settings.

Supported Services

Class services: Get_Attribute_All
Get_Attribute_Single

Instance services: Get_Attribute_All
Get_Attribute_Single
Set_Attribute_Single[1]

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 2

1. Support for this service can be disabled by implementing attribute #9 in the EtherNet/IP Host Object.

Instance Attributes

#	Access	Name	Type	Value	Comments
1	Get	Status	DWORD	0000 0001h 0000 0002h 0000 0010h	Bit #: Meaning: 0: Attribute #5 contains valid information from DHCP or nonvolatile storage 1: Attribute #5 contains valid information from hardware settings 4: Mcast pending 6: AcdStatus. Set to 1 if an address conflict is detected. Address conflict detection is enabled/disabled in attribute #10.
					Attribute #5 contains valid information from hardware settings.
					Mcast pending
2	Get	Configuration Capability	DWORD	0000 0016h - or - 0000 0006h - or - 0000 0026h - or - 0000 0036h	The 'Configuration Settable'-bit (bit 4) in this attribute reflects the value of instance attribute #9 in the EtherNet/IP Host Object. The module is capable of resolving host names by querying a DNS server
3	Get/Set	Configuration Control	DWORD	-	Value:Meaning: 0 Configuration from nonvolatile memory 2 Configuration from DHCP
4	Get	Physical Link Object	Struct of: UINT (Path size) Padded EPATH	- 0002h 20 F6 24 03h	- 2 words Path to Ethernet Link Object, Instance 3
5	Get/Set	Interface Configuration	Struct of: UDINT (IP) UDINT (Mask) UDINT (GW) UDINT (DNS1) UDINT (DNS2) STRING (Domain)		- IP address Subnet mask Default gateway Primary DNS Secondary DNS Default domain
6	Get/Set	Host Name	STRING	-	Host name of Anybus module
8	Get/Set	TTL Value	USINT	1	TTL value for EtherNet/IP multicast packets
9	Get/Set	Mcast Config	Struct of:		
		Alloc Control	USINT	0	Value:Meaning: 0 Use default allocation algorithm to generate multicast addresses 1 Allocate multicast addresses according to the values in the 'Num Mcast'- and 'Mcast Start Addr'-fields
		(reserved)	USINT	0	Set to zero. Do not change
		Num Mcast	UINT	1	Number of multicast addresses to allocate for EtherNet/IP
		Mcast Start Addr	UDINT	-	Starting multicast address from which to begin allocation
10	Set	SelectAcd	Bool	1	Value: Meaning: 0: Disable ACD 1: Enable ACD (default) If ACD (address conflict detection) is enabled, bit 6 in attribute #1 will be set if an ACD conflict is detected. The Network Status LED will also indicate a detected conflict.

#	Access	Name	Type	Value	Comments
11	Set	LastConflictDetected	Struct of:		ACD Diagnostic parameters related to the last conflict detected.
		AcdActivity	USINT	-	State of ACD activity when last conflict detected
		RemoteMAC	ARRAY of 6 USINT	-	MAC address of remote node form the ARP PDU in which a conflict was detected
		ArpPdu	ARRAY of 28 USINT	-	Copy of the raw ARP PDU in which a conflict was detected

8.11 Ethernet Link Object (F6h)

Object Description

This object groups diagnostic information for the Ethernet interface.

Supported Services

Class services: Get_Attribute_All
Get_Attribute_Single

Instance services: Get_Attribute_All
Get_Attribute_Single
Set_Attribute_Single
Get_And_Clear

Class Attributes

#	Access	Name	Type	Value	Comments
1	Get	Revision	UINT	3	Revision 3
2	Get	Max Instance	UINT	3	Instance 3 is the max instance
3	Get	Number of instances	UINT	3	3 instances

Instance Attributes

#	Access	Name	Type	Value	Comments
1	Get	Interface Speed	UDINT	10 or 100	Actual Ethernet interface speed
2	Get	Interface Flags	DWORD	-	See “Interface Flags” on page 68.
3	Get	Physical Address	Array of 6 USINTS	(MAC ID)	Physical network address
6	Get/Set	Interface Control	Struct:		
		Control Bits	WORD	-	Interface control bits
		Forced Interface Speed	UINT	-	Speed at which the interface shall be forced to operate. Returns ‘Object state Conflict’ if auto-negotiation is enabled.
7	Get	Interface Type	USINT	-	See “Interface Type” on page 68
10	Get	Interface Label	SHORT_STRING	-	See “Interface Label” on page 68

Interface Flags

Bit	Name	Description
0	Link status	Indicates whether or not the Ethernet 802.3 communications interface is connected to an active network. Value:Meaning: 0 Inactive link 1 Active link
1	Half/full duplex	Indicates the duplex mode currently in use. Value:Meaning: 0 Half duplex 1 Full duplex
2 - 4	Negotiation Status	Indicates the status of link autonegotiation Value:Meaning: 0 Autonegotiation in progress. 1 Autonegotiation and speed detection failed (using default values) 2 Autonegotiation failed but detected speed (using default duplex value) 3 Successfully negotiated speed and duplex. 4 Autonegotiation not attempted. Forced speed and duplex.
5	Manual Setting requires Reset	Value:Meaning: 0 Interface can activate changes to link parameters during runtime 1 Reset is required in order for changes to have effect
6	Local Hardware Fault	Value:Meaning: 0 No local hardware fault detected 1 Local hardware fault detected
7-31	(reserved)	(ignore)

Interface Type

Instance	Value	Description
1	2	Twisted-pair
2	2	Twisted-pair
3	1	Internal interface

Interface Label

Instance	Value
1	Port 1
2	Port 2
3	Internal

Appendix A

A. Technical Specification

A.1 Protective Earth (PE) Requirements

In order to achieve proper EMC behavior, the product must be connected to protective earth (PE) via the DIN-rail connector. If the DIN-rail cannot be used, PE must be connected to the power connector.

HMS Industrial Networks does not guarantee proper EMC behavior unless these PE requirements are fulfilled.

Note: Make sure the DIN-rail is properly connected to PE.

A.2 Power Supply

Supply Voltage

The HMS-EN2MB-R linking device requires a regulated 24 V (20.4 V to 28.8 V) DC power source.

Power Consumption

The typical power consumption is 150 mA at 24 V.

A.3 Environmental Specification

A.3.1 Temperature

Operating

-25° to +70° Celsius

Non-operating

-40° to +85° Celsius

A.3.2 Relative Humidity

The product is designed for a relative humidity of 5% to 95% noncondensing.

A.4 EMC (CE) Compliance

EMC compliance testing has been conducted according to the Electromagnetic Compatibility Directive 2004/108/EC. For more information please consult the EMC compliance document, see product/support pages for EtherNet/IP to Modbus-TCP Linking Device at www.anybus.com.

Appendix B

B. Copyright Notices

This product includes software developed by Carnegie Mellon, the Massachusetts Institute of Technology, the University of California, and RSA Data Security:

****

Copyright 1986 by Carnegie Mellon.

****

Copyright 1983,1984,1985 by the Massachusetts Institute of Technology

****

Copyright (c) 1988 Stephen Deering.

Copyright (c) 1982, 1985, 1986, 1992, 1993

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Stephen Deering of Stanford University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' ANDANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

****

Copyright (C) 1990-2, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD4 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD4 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

****

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.