Anybus® BY HMS NETWORKS

Linking Device

EtherNet/IP to Modbus-TCP

USER MANUAL

scm-1202–181 2.0 en-US ENGLISH

HMS Connecting Devices™

Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Networks of any inaccuracies or omissions found in this document. HMS Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Networks and is subject to change without notice. HMS Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the 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.

Table of Contents

Page

This page intentionally left blank

1 Preface

1.1 About This Document

This document describes how to install and configure the HMS-EN2MB-R EtherNet/IP to Modbus-TCP linking device.

For related documentation and file download, visit the support website at www.anybus.com/support.

1.2 Document Conventions

Numbered lists indicate tasks that should be carried out in sequence:

1. First do this
2. Then do this

Bulleted lists are used for:

- Tasks that can be carried out in any order
- Itemized information

► An action

→ and a result

User interaction elements (buttons etc.) are indicated with bold text.

```
Program code and script examples
```

Cross-reference within this document: *Document Conventions, p. 5*

External link (URL): www.hms-networks.com

WARNING
Instruction that must be followed to avoid a risk of death or serious injury.

Caution
Instruction that must be followed to avoid a risk of personal injury.

Instruction that must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.

Additional information which may facilitate installation and/or operation.

1.3 Related Documents

Document	Author	HMS Document ID
EtherNet/IP to PROFIBUS DP Linking Device User Manual	HMS	SCM-1202-026
Modbus Application Protocol Specification V1.1B	Modbus Organization	NA
CIP Specification, Vol 1 (CIP Common) & 2 (EtherNet/IP)	ODVA	NA

1.4 Document History

Version	Date	Description
2.0	2021-05–19	First release of reworked document. This document replaces SCM-1202–008.

1.5 Trademarks

2 About the Linking Device

2.1 Features

The EtherNet/IP to Modbus-TCP linking device provides a seamless connection between an Allen-Bradley Programmable Controller (PLC) by Rockwell Automation on EtherNet/IP and a Modbus-TCP network. It features:

- User interface integration in Studio 5000

  The configuration manager for the gateway is included in an add-on profile (AOP) for Studio 5000 and does not have to be installed separately. All network and device level configuration is done in Rockwell Studio 5000.

- Tag level configuration

  The user configures the tags used in the PLC when configuring the gateway. Process data is automatically mapped from Modbus-TCP to EtherNet/IP without manual work.

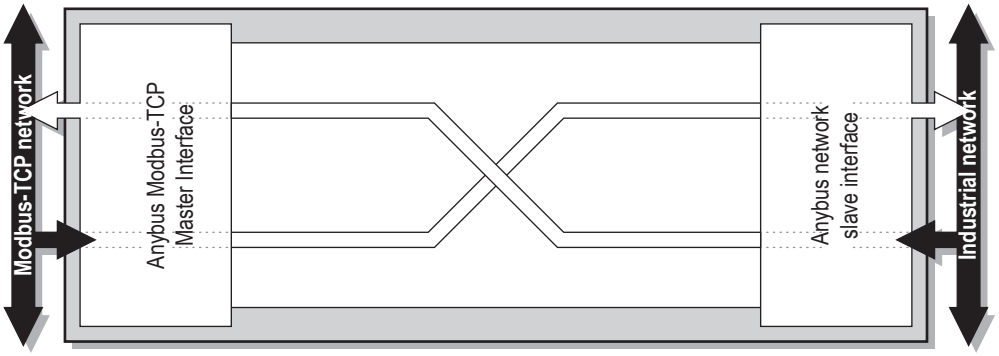
- Automatic backup of configuration

  The state of the configuration is automatically stored in the Studio 5000 project when the configuration manager is closed, and will be retrieved next time the configuration manager is opened.

The AOP is supported by RSLogix 5000 version 20 and later.

The linking device consists of an intelligent gateway platform, a Modbus-TCP network interface and an EtherNet/IP network interface. The gateway connects the two network interfaces, and forwards data from one to the other.

The linking device enables transparent data transmission between the two networks, by letting the client on the EhterNet/IP network control the Modbus-TCP network servers.

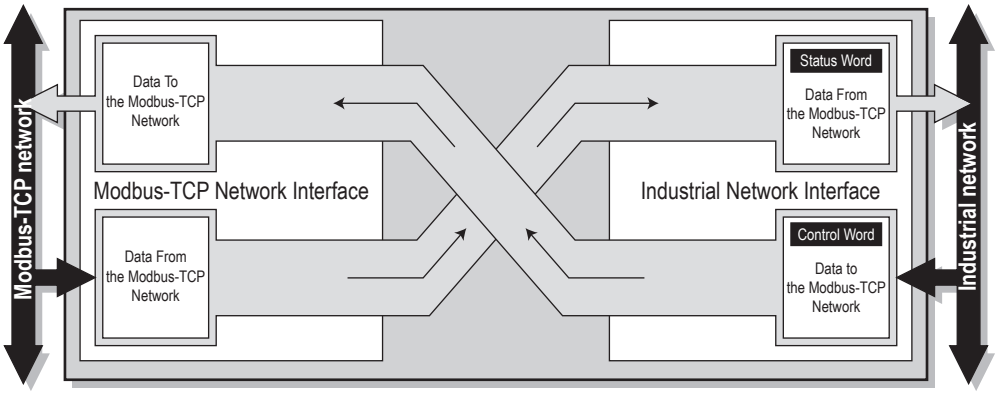


2.2 Data Consistency

The linking device ensures data consistency within all connections on the EtherNet/IP side, and within each command on the Modbus-TCP side. The EtherNet/IP side and the Modbus-TCP side are not cyclically synchronized.

There is one assembly in each direction per connection in this linking device.

Each of the two network interfaces exchanges data on its network through two buffers. The linking device forwards the data internally between these buffers as shown below.



2.2.1 Process Data I/O Buffer

Both the incoming and outgoing buffers on the EtherNet/IP side and the incoming and the outgoing buffers on the Modbus TCP side holds a maximum of 4096 bytes of data.

The buffers in the EtherNet/IP interface are represented by up to 10 separate class 1 connections with a maximum of 500/496 bytes assemblies.

There is one assembly in each direction per connection in this linking device.

See also *Data Transactions, p. 33*.

3 Installation

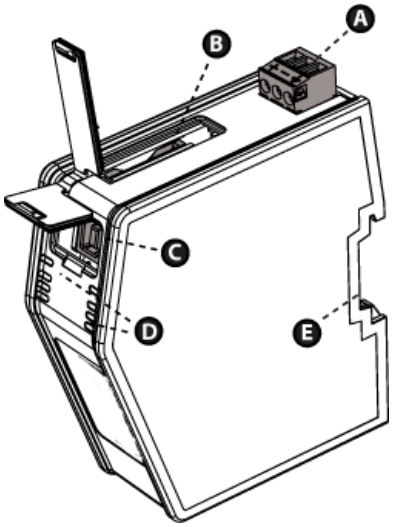
3.1 External View

> **!** Use the SD card slot with care. Do not use an SD card for configuration backups. A backup of the configuration is always stored in the Studio 5000 project. The configuration in the linking device and in Studio 5000 must match for operation. See *SD Card Functionality, p. 65*.

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, p. 11* and *Power Supply, p. 64*.

B: SD Card Slot
This slot adds the possibility to store and load configurations from an SD card.
Use with care. A backup of the configuration is always stored in the Studio 5000 project. The configuration in the linking device and in Studio 5000 must match for operation.
See *SD Card Functionality, p. 65*.

C: USB Port
This port adds the possibility to connect a PC to the linking device to perform firmware upgrades.
See *USB Connector, p. 12*.

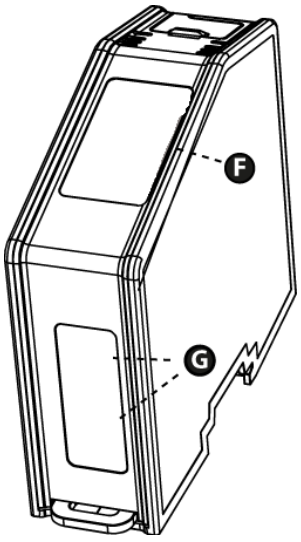


D: Status LEDs
See *Status LEDs, p. 29*.

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, p. 9*.

F: EtherNet/IP Connectors
See *EtherNet/IP & Modbus-TCP RJ45 Connectors, p. 12*.

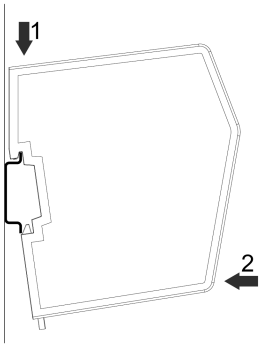
G: Modbus-TCP Connectors
2–port switch with daisy chain functionality.
See *EtherNet/IP & Modbus-TCP RJ45 Connectors, p. 12*.



3.2 Mounting the Linking Device

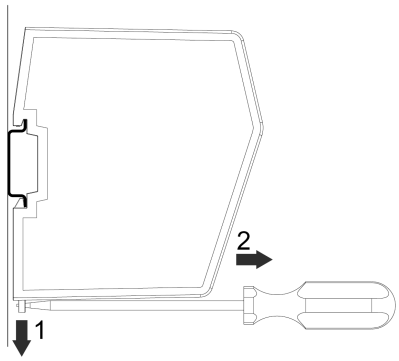
The EtherNet/IP to Modbus-TCP Linking Device can be physically installed either by mounting it onto a DIN-rail or, if installed in areas exposed to vibration, by mounting it on a wall for more stability.

3.2.1 DIN-rail Mounting

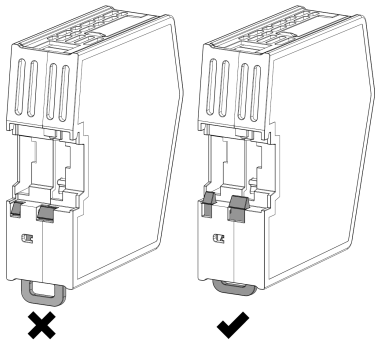


Make sure the DIN-rail fastening mechanism on the back of the module is in a fixed and closed position, i. e. pushed all the way up.

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).



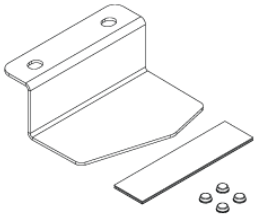
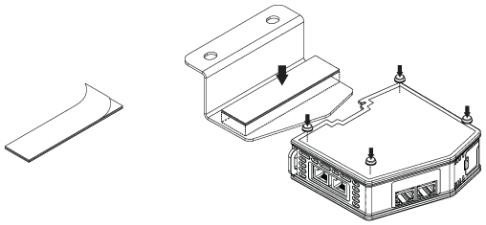
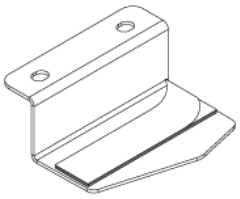
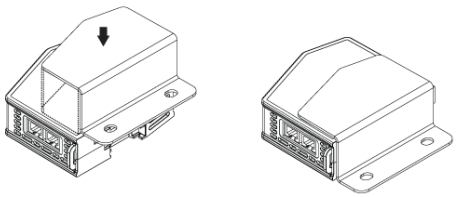
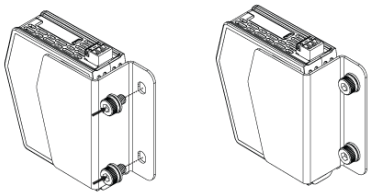
Do not leave the module with the DIN-rail fastening mechanism in a fixed and open position. This may eventually wear the fastening mechanism out so it cannot be used efficiently. Be sure to push the DIN-rail fastening mechanism back into the fixed and closed position after demounting the module.

3.2.2 Wall Mounting

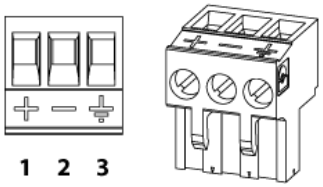
Use the wall mounting option if there is a need to place the linking device in an environment exposed to vibration. This way of mounting the module offers more stability than the traditional DIN-rail mounting.

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

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.

Step	Description	Visual Description
1	Open up the package containing the wall mounting accessories. - One metal frame - Industrial velcro - Four plastic vibration dampers	
2	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.	
3	Remove the plastic protection from the other side of the velcro.	
4	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.	
5	Attach the metal frame and the device to a wall using screws and washers (not enclosed).	

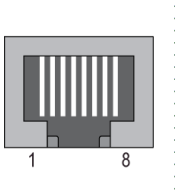
3.3 Power Connector

Pin no	Description	Connector
1	+24 V DC	
2	GND	
3	FE (Functional Earth)	

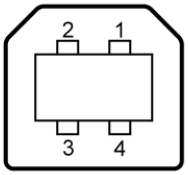
Use 60/75 or 75xC copper (CU) wire only.

The terminal tightening torque must be between 5... 7 lbs-in (0.5... 0.8 Nm)

3.4 EtherNet/IP & Modbus-TCP RJ45 Connectors

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

3.5 USB Connector

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

4 Configuration

4.1 Configuration Flow

The linking device can **only** be configured from within Studio 5000. This is done by configuring the settings of a linking device imported into the current Studio 5000 project.

A separate Modbus-TCP client configuration manager is executed to handle the configuration. It is included as an add-on profile and integrated in Studio 5000. The configuration manager is pre-installed as part of your Studio 5000 installation. No separate installation is needed.

The linking device configuration is always automatically stored in the Studio 5000 project when the Modbus-TCP client configuration manager is closed. The next time the configuration manager is opened, the stored configuration is retrieved.

The linking device is configured in the following basic steps:

1. In Studio 5000, set up and launch the HMS-EN2MB-R configuration manager (step 1–6 in *Studio 5000 Configuration Example, p. 15*).
2. Configure the Modbus-TCP client. See step 7–13 in *Studio 5000 Configuration Example, p. 15* and *Modbus-TCP Client Configuration, p. 14*.
3. Based on the client configuration, configure the tags that will be used in Studio 5000. The configuration and status is automatically sent to Studio 5000 when the tag editor is closed. See steps 14–16 in *Studio 5000 Configuration Example, p. 15* and *Tag Editor Configuration, p. 25*.

4.2 Configuration Requirements

The EtherNet/IP to Modbus-TCP linking device is configured over the EtherNet/IP port using the pre-installed add-on profile in Studio 5000.

Three requirements must be met for the configuration manager to function:

- The linking device must be powered on.
- The linking device must be connected and available from the computer with the Studio 5000 installation.
- Any switches or firewalls between the linking device and the computer with Studio 5000 must allow HTTP traffic.

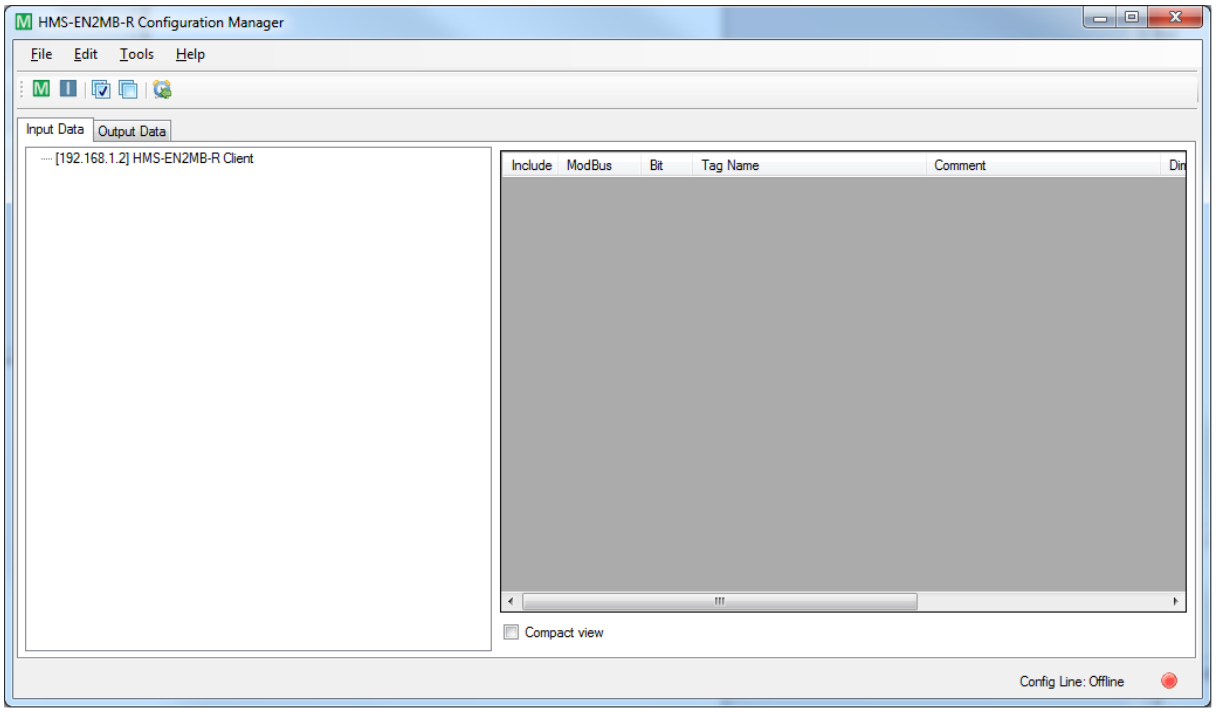
This is only required to configure the device, not during operation.

4.3 Linking Device Configuration Manager

The foundation of theEtherNet/IP to Modbus-TCP linking device configuration manager is the tag editor. It is the first view when you open the configuration manager. At first startup the editor holds no tags.

Config Line in the bottom bar shows if the EtherNet/IP connection between the PLC and the

linking device is online or offline. The linking device configuration cannot be changed when the connection is online.



See *Linking Device Configuration Manager Interface, p. 38* for a description of the menu options.

4.4 Modbus-TCP Client Configuration

The EtherNet/IP to Modbus-TCP linking device has a built-in Modbus—TCP client configuration manager. The following system requirements must be met to access the configuration manager:

- Javascript enabled
- HTTP access from the computer with Studio 5000 and the linking device

Consider the following when configuring Modbus-TCP servers and transactions:

- When you have saved data to the configuration but not yet applied it, this will appear top of the configuration manager:

  The configuration needs to be applied for changes to take effect. Go to **HMS-EN2MB-R Management** page to apply the configuration or cancel changes.

- A changed configuration must be applied in order to take effect.
- 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 limit is 4096 bytes input data and 4096 bytes output data, including optional control/status word and live list.
- Modbus transactions are assembled in up to 10 Ethernet/IP Class 1 Exclusive Owner (IO) Connections.
- Each Ethernet/IP Class 1 Exclusive Owner (IO) connection holds one input assembly of 500 bytes and one output assembly of 496 bytes.
- Only as many Ethernet/IP Class 1 Exclusive Owner (IO) connections as needed are defined and used.

- 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.

Changes made to the configuration will not be used by the linking device before they are applied and saved.

See also *Modbus-TCP Client Configuration Manager, p. 39*.

See also *Modbus-TCP Functions, p. 37*.

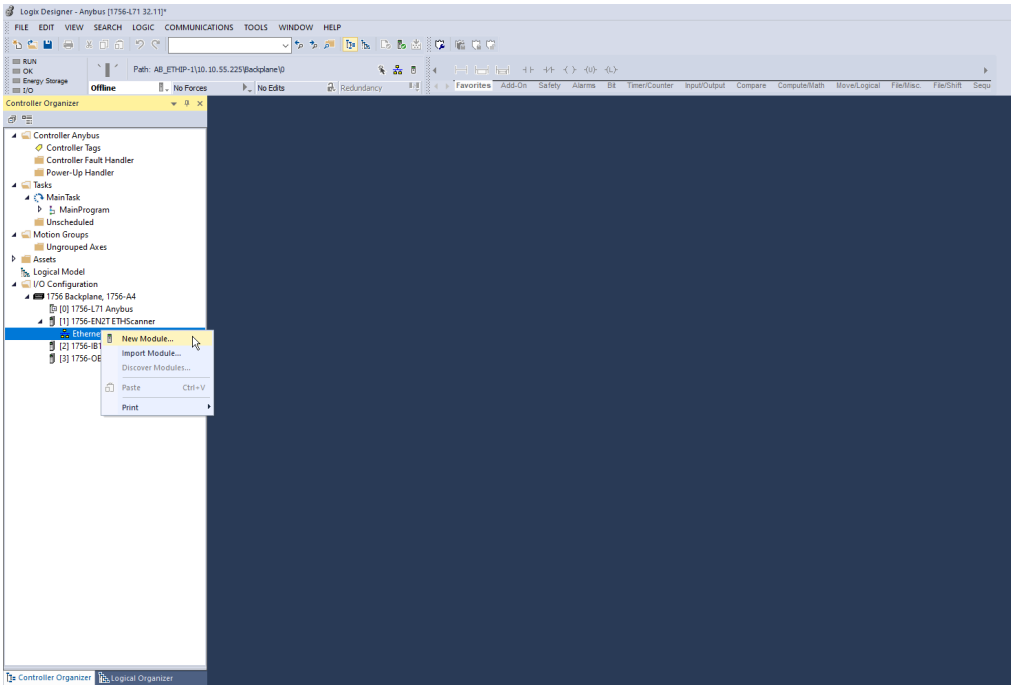
4.5 Studio 5000 Configuration Example

This example will guide you through the steps to create a basic configuration for theEtherNet/IP to Modbus-TCP linking device.

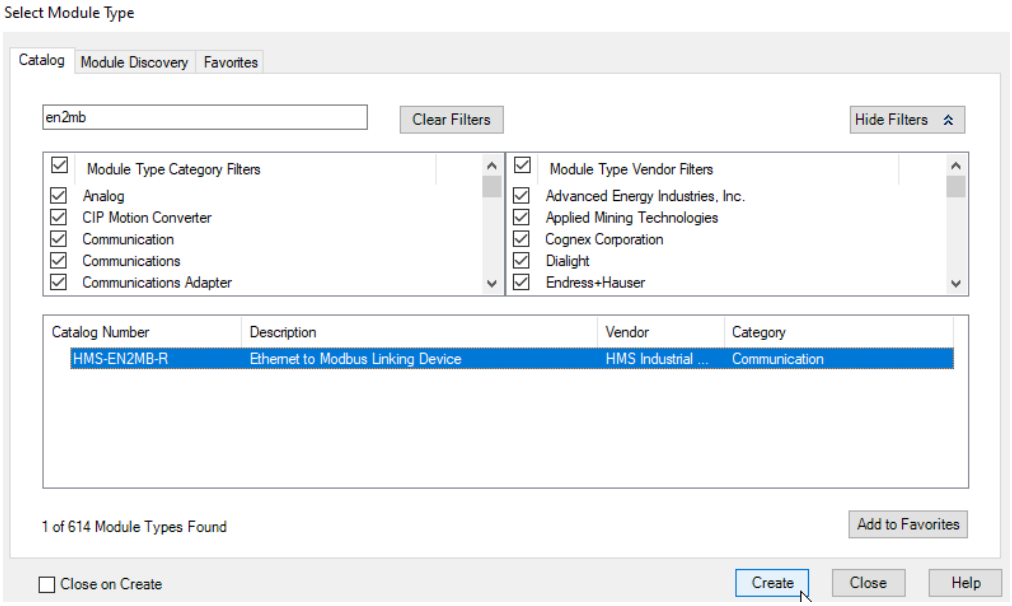
1. Start the Studio 5000 software.

   Expand the **I/O Configuration** folder in the tree view.

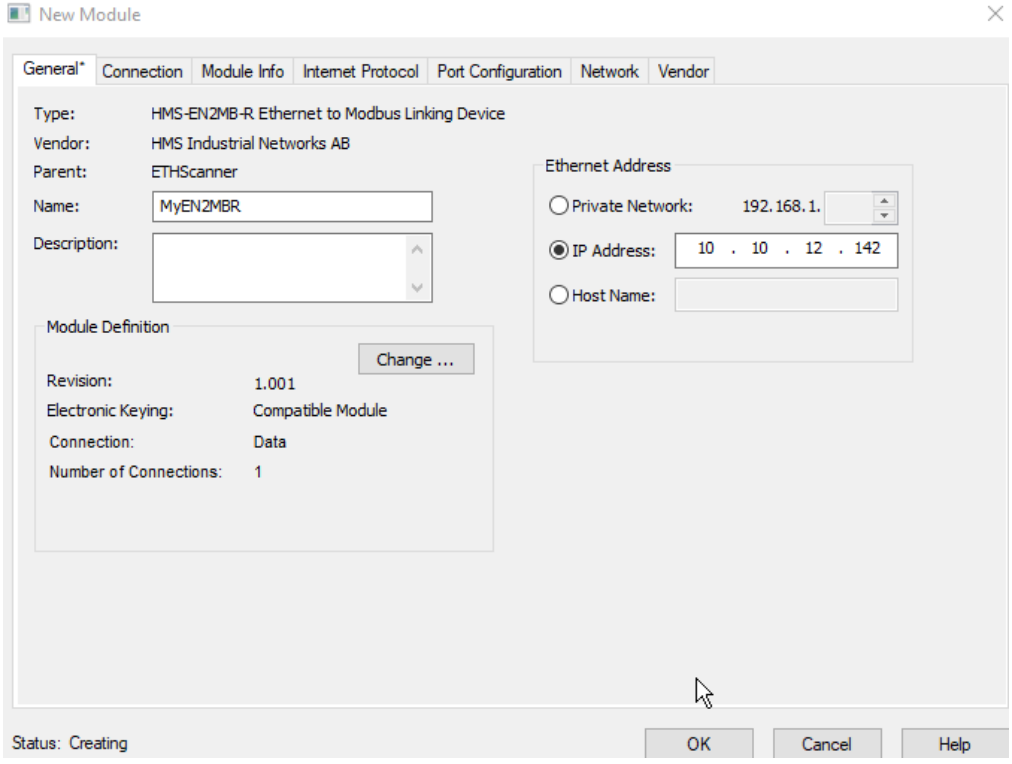
   Right-click **Ethernet** and select **New Module**.



2. Select the EtherNet/IP to Modbus-TCP linking device (catalogue number: HMS-EN2MB-R) and click **Create**.

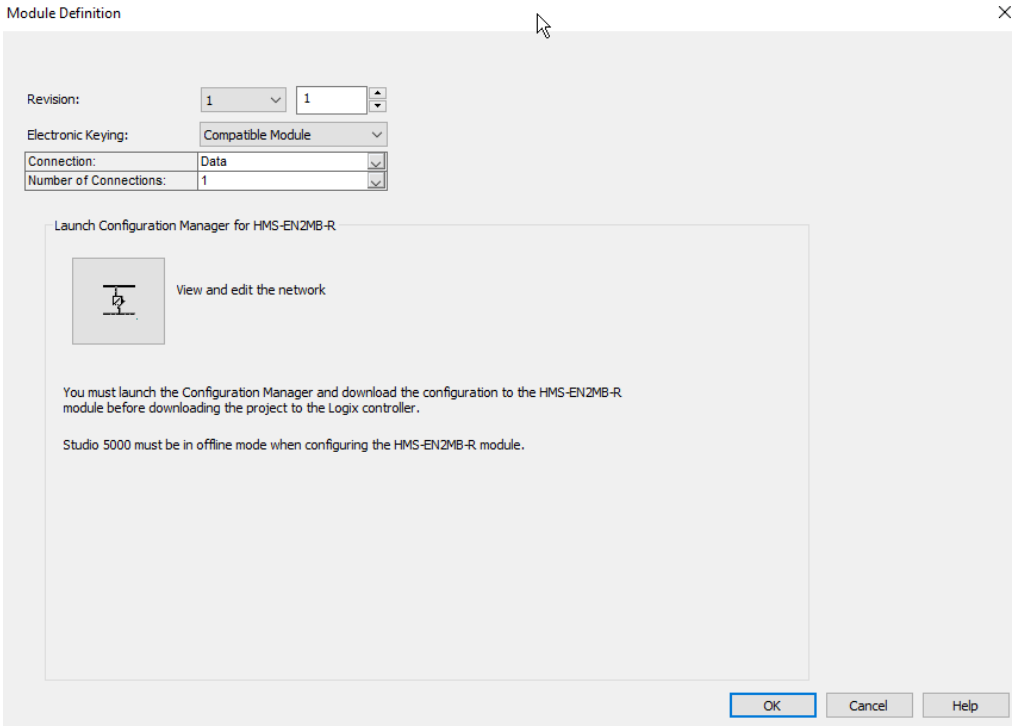


3. In the New Module window, assign a name to the module. The IP address should be set via the BOOTP-DHCP server and entered in the IP address field.



4. Click **Change** in the Module Definition section.

5. In the Module Definition window, launch the configuration manager for the HMS-EN2MB-R linking device by clicking the **View and edit the network** button.

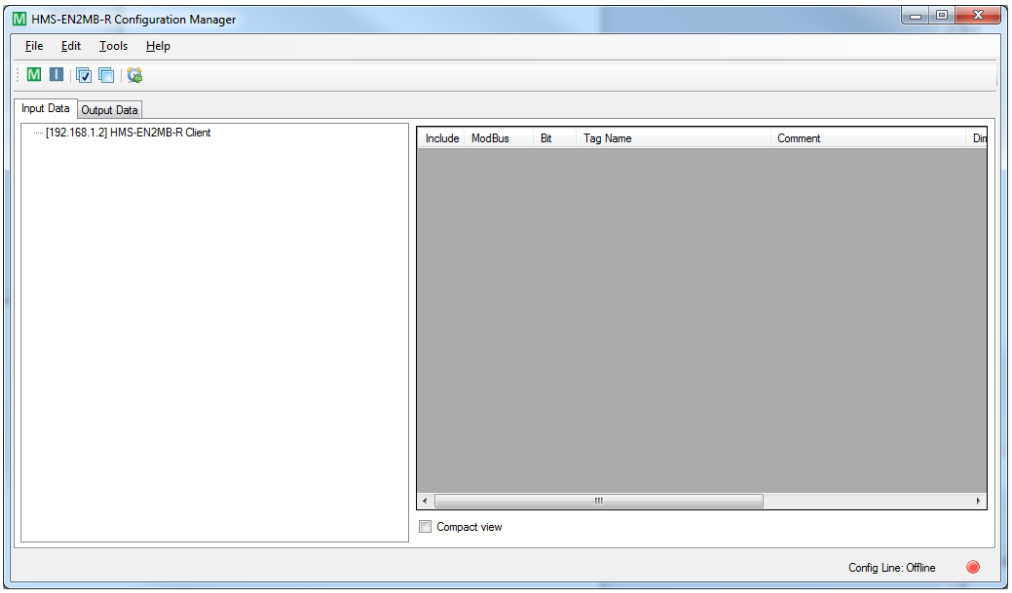


6. The foundation of the linking device configuration manager is the tag editor, and it's the first view to be presented in the configuration manager. Since this is a new configuration, the editor holds no tags. See *Linking Device Configuration Manager Interface, p. 38* for descriptions of the menu options.

   The first step is to configure the Modbus-TCP Client to define the process data to be sent through the linking device. The HMS-EN2MB-R uses a built in web interface in the actual linking device to configure the Modbus-TCP Client.

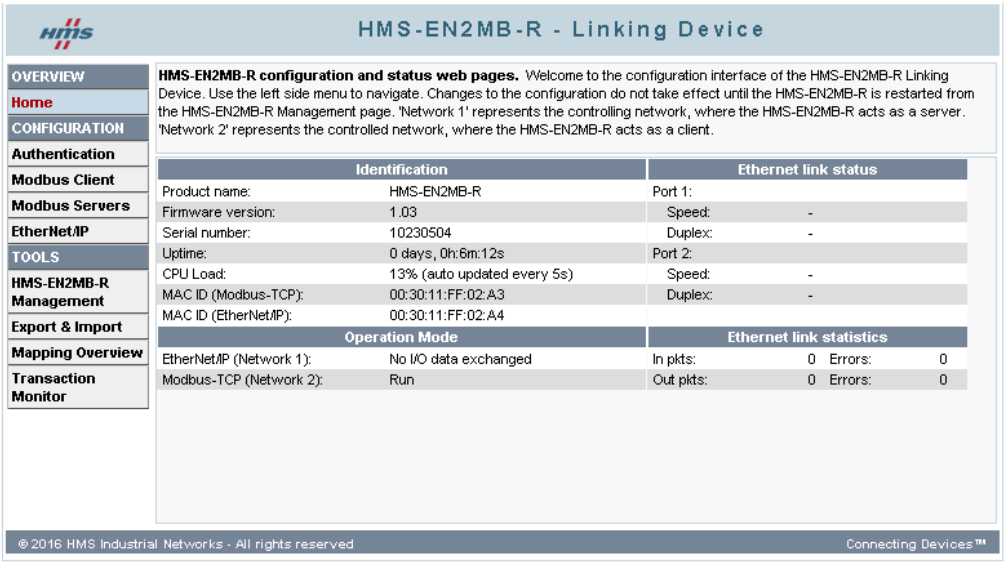
> **!** Using the built in web interface makes it mandatory to have HTTP access between the linking device and Studio 5000 during the configuration phase, as well as using monitoring features during operation.

   Open the HMS-EN2MB-R Modbus-TCP Client configuration manager by clicking the leftmost icon (M) in the toolbar to proceed.

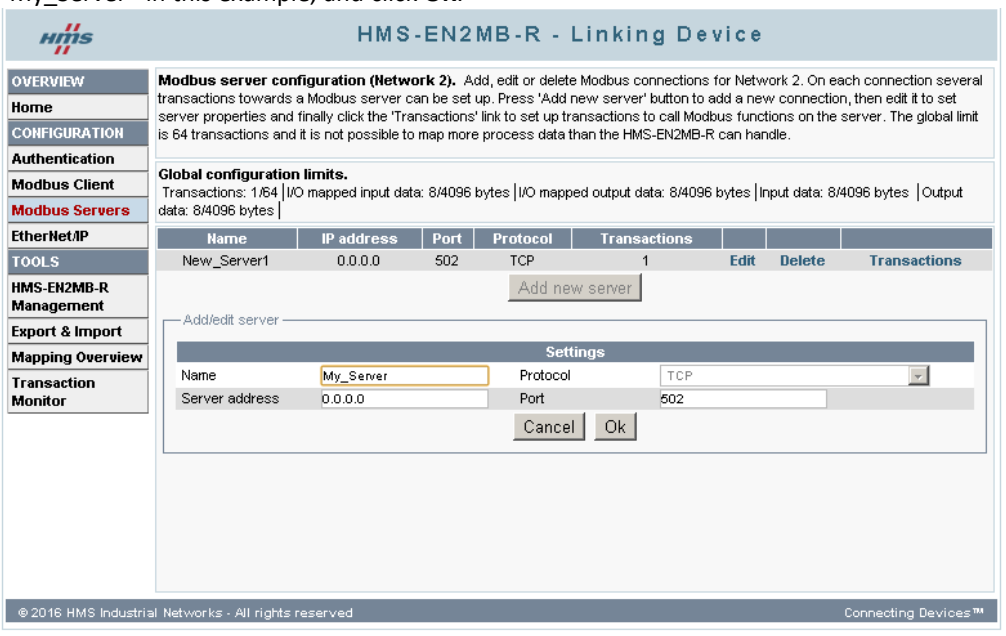


7. The first view in the Modbus-TCP Client interface is the overview. The overview presents a summary of useful information about the linking device.

   Select **Modbus Servers** in the left-hand menu to add Modbus-TCP servers to the configuration.



8. Click **Add new server** to add the configuration for a new Modbus-TCP server. Add a name, “My_Server” in this example, and click **OK**.



9. The next step is to define which transactions should be executed between the Modbus-TCP client and the Modbus-TCP server.

   See *Modbus-TCP Functions, p. 37* for information about supported Modbus-TCP functions.

   Click on **Transactions** to add transactions for this server.

10. Perform the following steps to add and configure a new transaction:

    a. Click **Add new transaction** and select **Edit**.

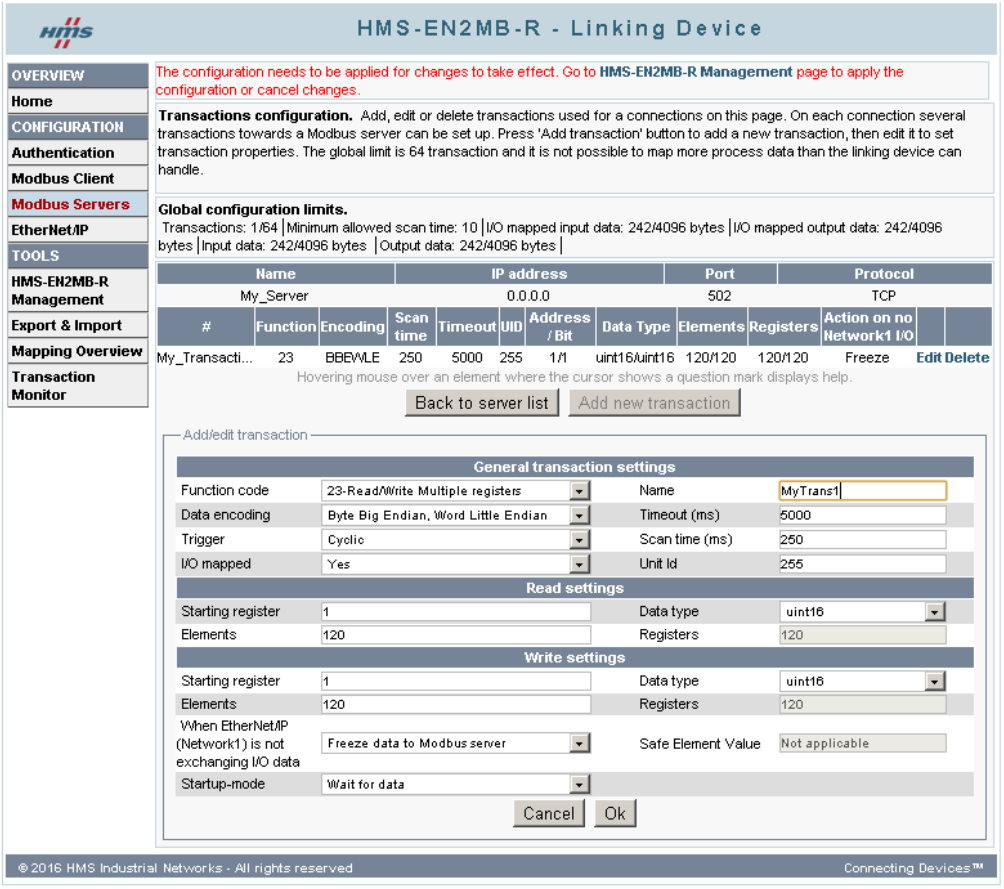
    b. Add a name of the transaction in the **Name** field in **General transaction settings**. “MyTrans1” is used in this example.

    c. Select the Modbus command function that should be executed. This example uses function code 23 (Read/Write multiple registers).

    d. Define the number of elements to be sent and received in the **Elements** field in both the **Read settings** and **Write settings** sections.

    This example defines 120 elements (240 bytes) in each direction.

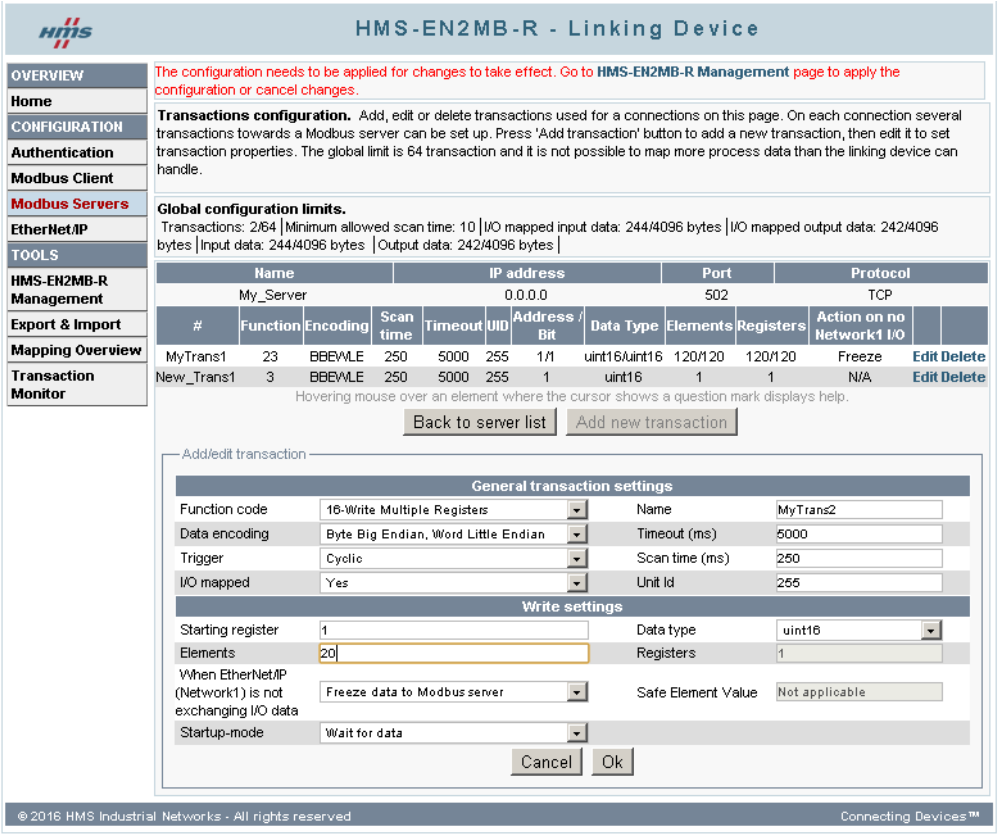
    e. Click **OK** to save the new transaction.



11. If you want to add another transaction, repeat the procedure in the previous step (step 10).

    See *Modbus-TCP Functions, p. 37* for information about supported Modbus-TCP functions.

    This example defines another transaction named “MyTrans2” that uses function code 16 (Write Multiple Registers) and writes 20 elements (40 bytes).

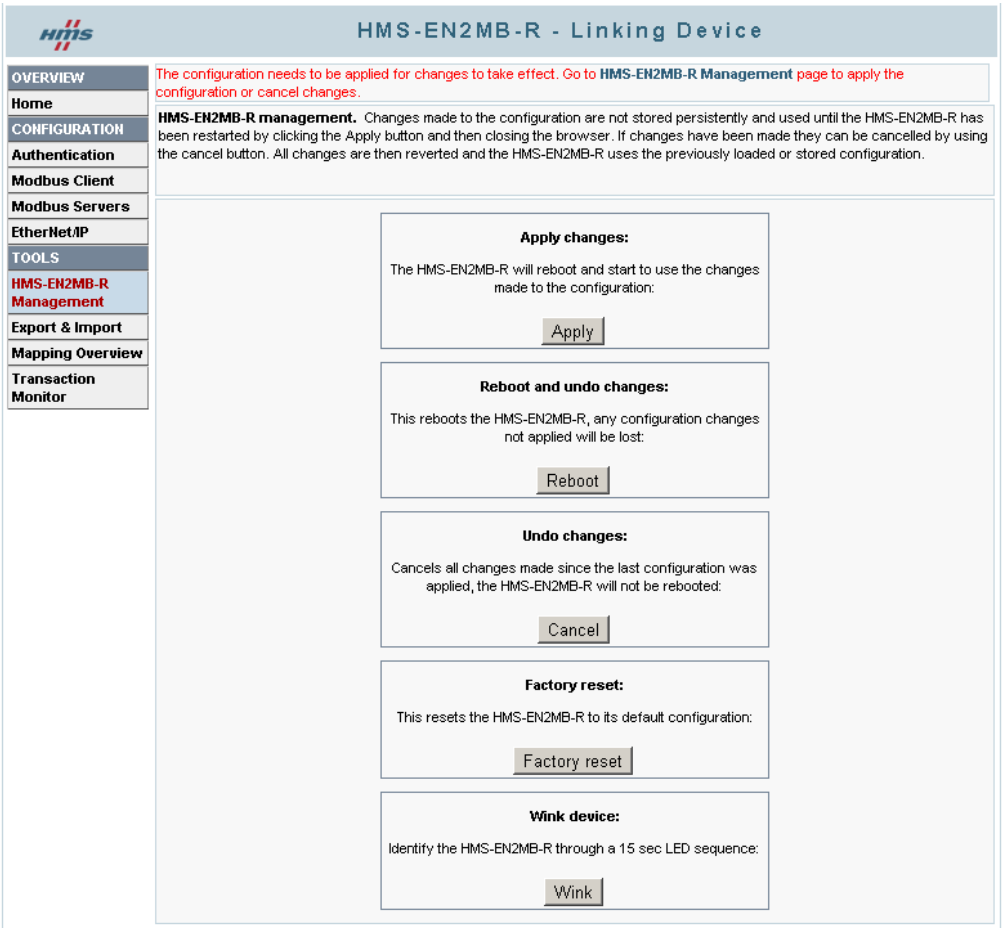


12. If you want to add another server that defines new transactions, repeat the procedures in previous steps (step 7—10).

    This example adds a second Modbus TCP Server (“My_Server2”) with one transaction (“MyTrans3”) that uses function code 23 to read and write 120 elements in each direction.

13. When the configuration of servers and transactions is complete, select **HMS-EN2MB-R Management** under Tools in the left-hand menu.

    Click **Apply** in the Apply changes section to download the configuration to the linking device.



    The configuration will now be verified.

    The process data will be placed in as many assemblies as required for the configuration, while still fulfilling data integrity within each transaction.

    If no verification issues are reported, the configuration will go active in the linking device.

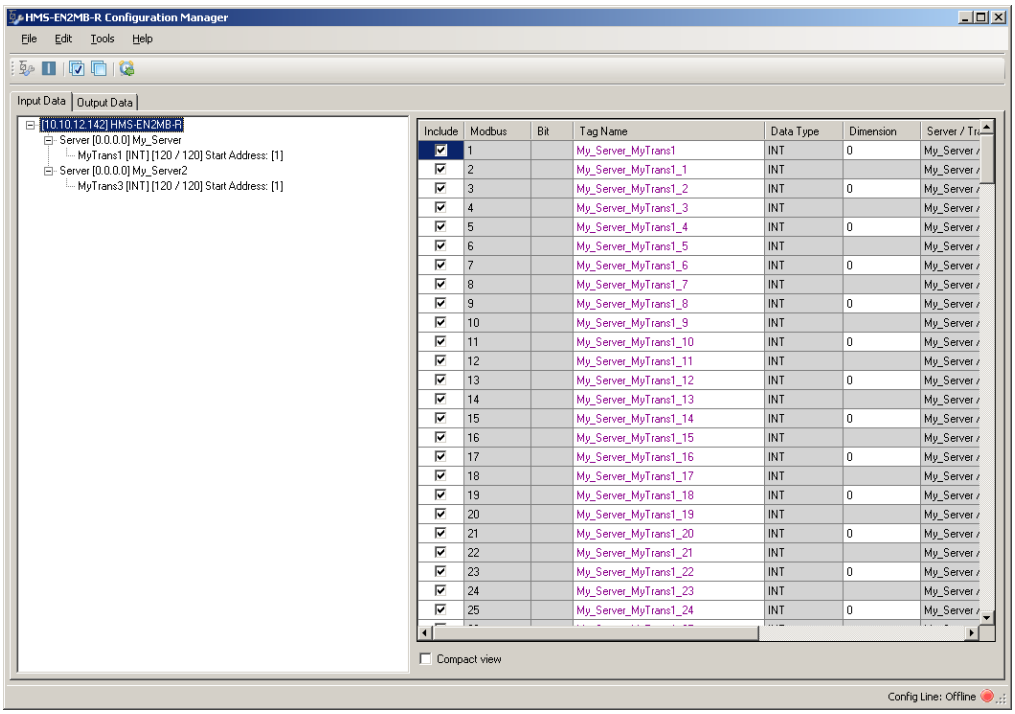
14. Close the Modbus-TCP Client configuration manager to show the tag editor in the Linking Device configuration manager again.

    The tag editor will generate tags based on the Modbus-TCP Client configuration and how the process data is placed in the assemblies.

    All new tags have a Tag Names are marked in magenta. Any changes that cannot be assumed to represent the same process data, or that use new process data, will generate new tags marked in magenta. If there are tags already defined when the Modbus-TCP Client configuration is changed, the tag editor will identify the tags that are still targeting the same Modbus-TCP server and address, and keep these.

    Some tag properties can be edited in the tag editor. The name and dimension can be changed. If **Include** is deselected, a signal will not be created for that tag.

    See *Tag Editor Configuration, p. 25* for more information about how to use the tag editor.



15. When the configuration is finished, close the tag editor window.

    The tag configuration, a checksum and the current project configuration files are sent to Studio 5000.

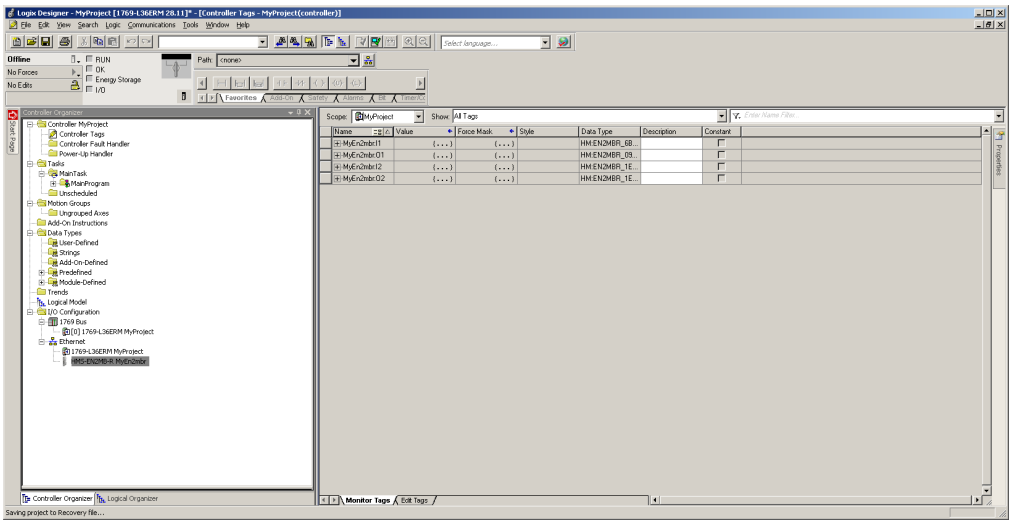
    The tag configuration is used in the Studio 5000 project to create signals and automatically map them to the linking device.

    The checksum is used by the gateway to ensure that the tag configuration in the PLC is a correct match with the configuration in the linking device.

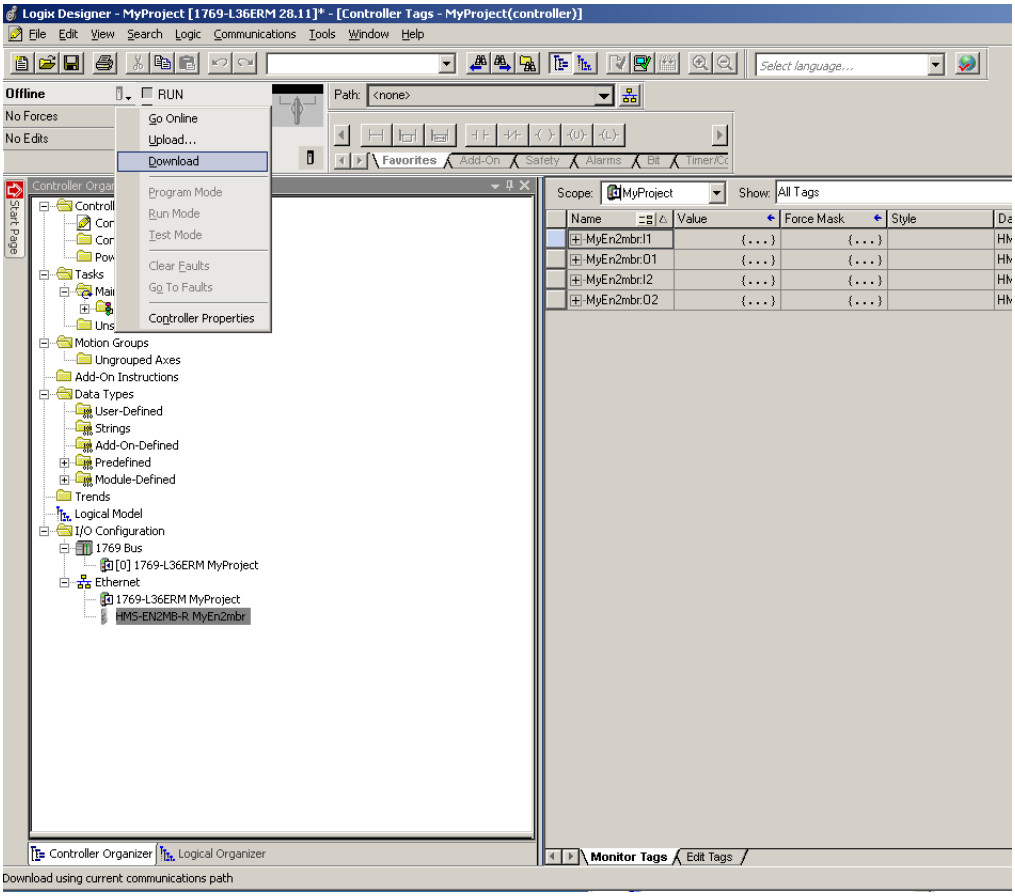
    The current project configuration files are stored in Studio 5000 and will be retrieved the next time the linking device configuration interface is opened.

> **!** There is no need to use the SD card for backup of the configuration. The configuration is always automatically stored in Studio 5000.

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



16. Right-click the computer icon and select **Download** to download the configuration from Studio 5000 to the PLC.



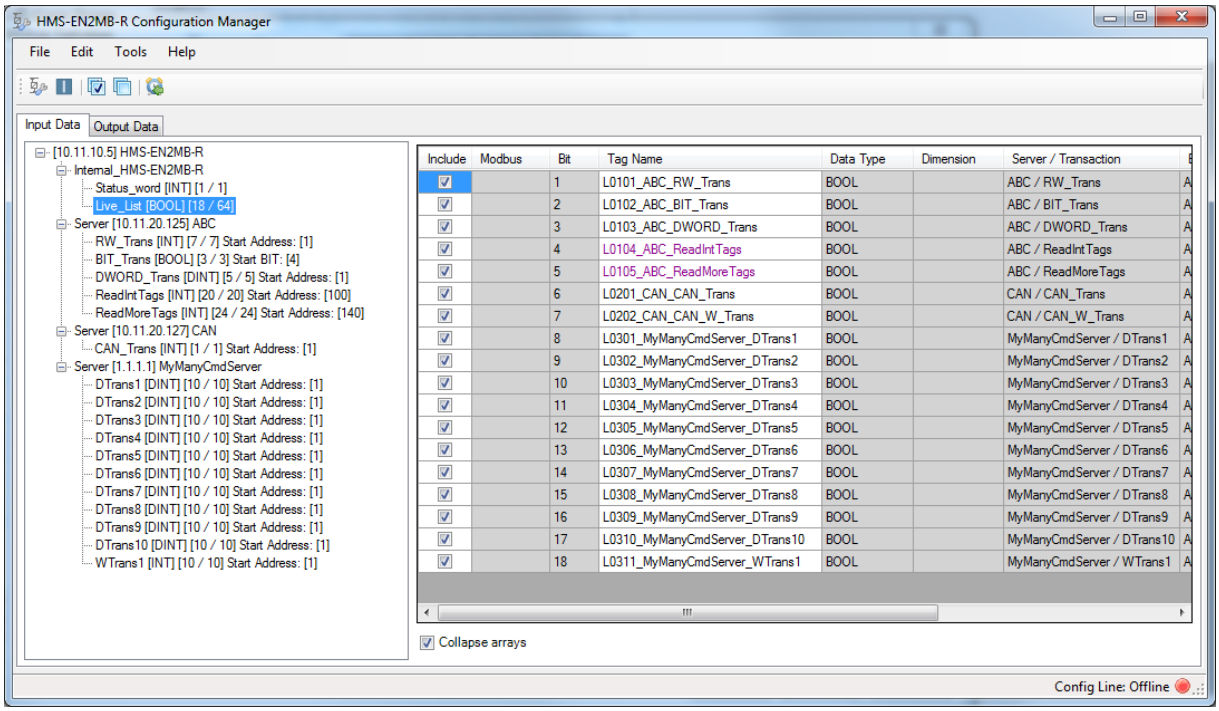
4.6 Tag Editor Configuration

The tag editor will generate tags when the Modbus-TCP Client configuration manager has been edited and is closed. The tags will be based on the Modbus-TCP Client configuration and how the process data have been placed in the assemblies.

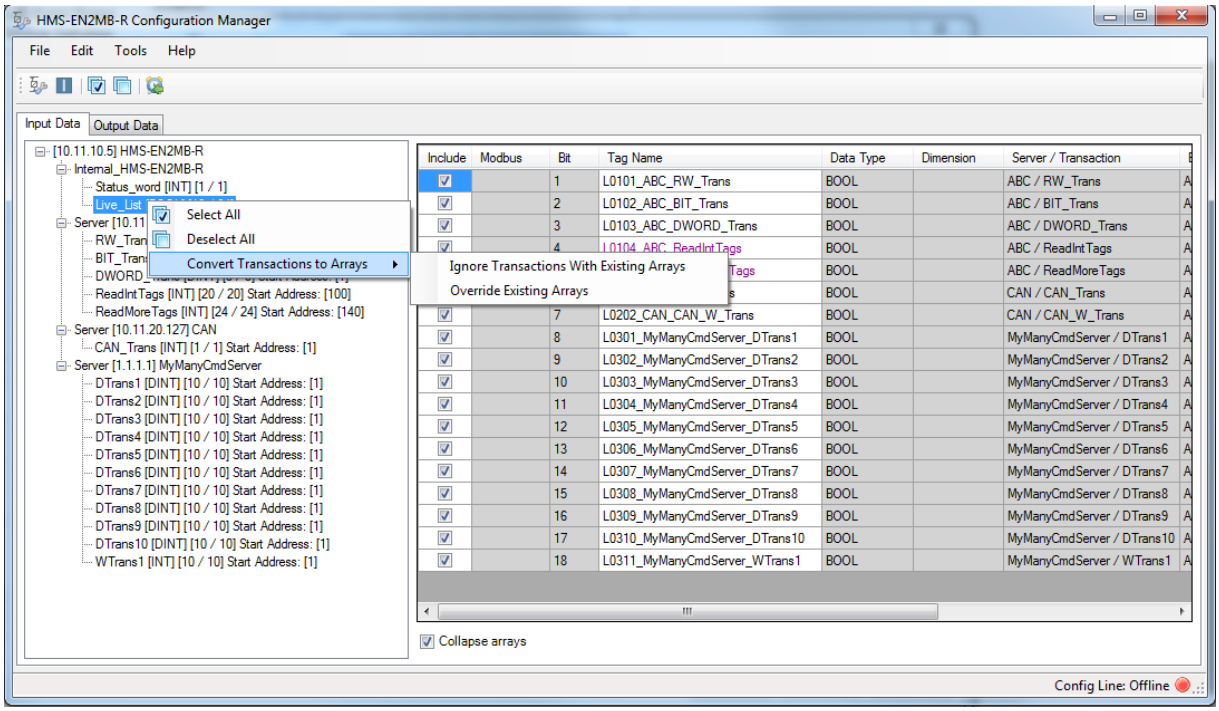
Tags that are new or edited since using the Modbus-TCP Client configuration manager will be marked in magenta.

To export tags to the Studio 5000 environment, use the checkbox in the **Include** column. Any

unchecked tags will still be available in the process data, but no tag will be created in Studio 5000.



You can make mass operations on all tags by right-clicking on a Modbus-TCP server or transaction in the tree structure to the left.



See *Tag Editor Options, p. 38* for descriptions of the options.

4.6.1 Tag Arrays

You can collect a number of consecutive tags of the same data type into a larger array tag.

To create an array tag, set a positive value in the dimension field. The tag editor will automatically allocate the following number of tags of that value and mark them blue. The tags are now uneditable.

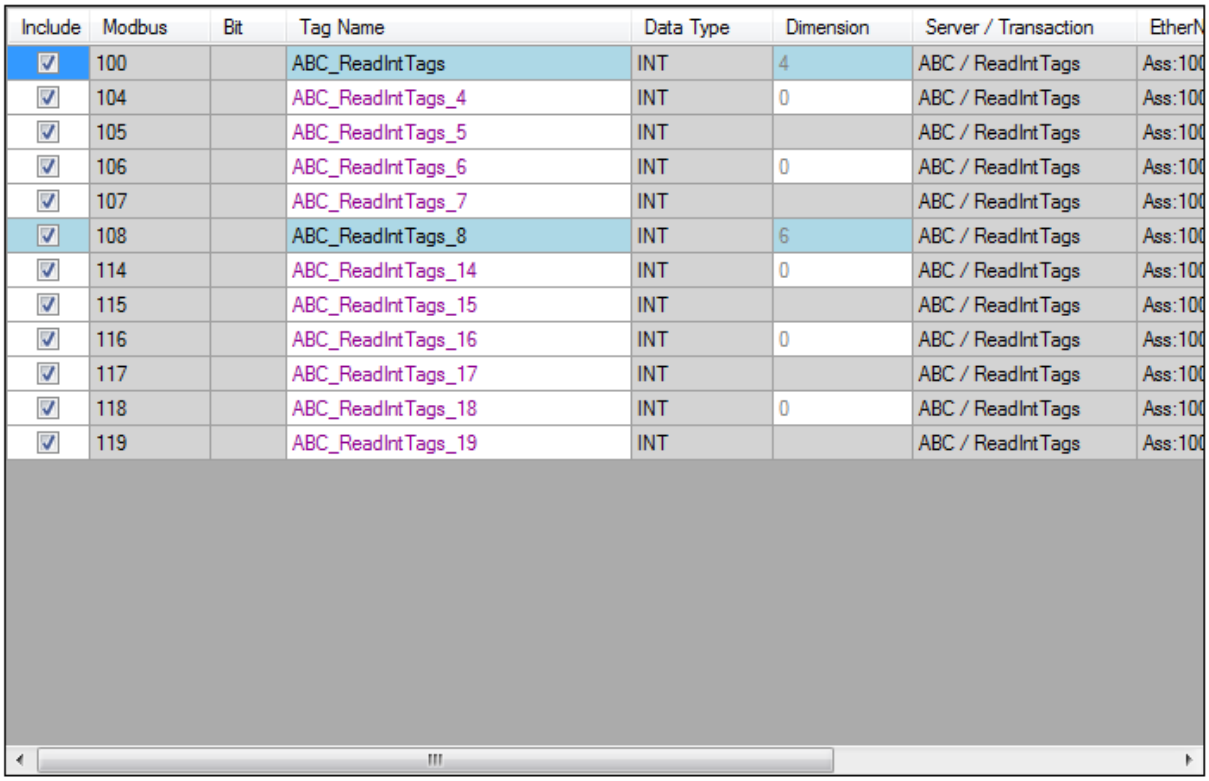
The name of the first tag will be used as the name of the array tag. Original names of the tags collected in the array will be remembered by the tag editor.

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 release the tags from the array, change the dimension to a lower value.

To remove the array tag, set the dimension to zero.

You can collapse array tags into single lines to get a more compact overview of the configuration. Check the **Collapse arrays** checkbox under the tag list view.



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

4.6.2 Tag Rule Definitions

> **!** The default and automatically generated tags will adhere to the rules below.

All process tags (controller tags) in the configuration must follow these rules:

- The start address offset must be within the assembly of the process data.
- The start address offset of the process tag must be divisible by 4, for data types DINT or REAL or tag arrays (when dimension is above zero).
- The memory address of the process tag must be divisible by 2 for data type INT.
- Two process tags can not occupy the same address area in an Ethernet/IP assembly.
- Two process tags can address the same address on the Modbus-TCP side.
- The designated address allocation of the process tag must be smaller than or equal to the corresponding designated address range in the configuration.

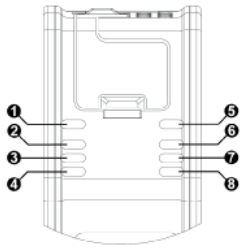
5 Verify Operation

5.1 Status LEDs

A test sequence is performed on all LEDs during startup.

At power up, LED 1 - 2 will indicate solid orange and LED 3 - 4 will indicate solid yellow for 15 - 30 seconds. This will be followed by a LED test sequence, performed on the (MS EN) Module Status and (NS) Network Status LEDs.

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.

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

6 Maintenance

6.1 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.

6.2 Replace a Linking Device

> **!** The recommended procedure to back up a configuration is to use the backup function in Studio 5000. It is NOT recommended to use the SD card for backups.

The configuration is backed up in the Studio 5000 project every time the configuration manager is closed, and recalled the next time it is opened.

To replace a linking device:

1. Make sure that the configuration manager is closed.
2. Remove the old linking device.
3. Connect the new linking device.
4. Set the IP-address of the gateway in Studio 5000.
5. Open the gateway configuration manager from within Studio 5000
6. Before doing anything else, download the configuration to the gateway.
7. Close the configuration manager.

6.3 SD Card

> **!** The recommended procedure to back up a configuration is to use the backup function in Studio 5000. It is NOT recommended to use the SD card for backups.

> **!** The SD card acts as client in the linking device. When a 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. This may result in a checksum missmatch, and the Linking Device will reject any connection attempts from the PLC.

Using an SD card with the EtherNet/IP to Modbus-TCP linking device adds the following functionality:

- Backup

  The recommended procedure to back up a configuration is to use the backup function in Studio 5000. See *Backup and Restore, p. 46*.

  Every applied change in the configuration will automatically be saved to the linking device and the SD card.

- Configuration copy

  The configuration on one linking device can be copied to other linking devices using the SD card.

- Replacement

  See *Replace a Linking Device, p. 30* for the recommended linking device replacement procedure.

  If a linking device malfunctions during operation, a replacement device can be configured by moving the SD card to the new device.

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

See also *SD Card Functionality, p. 65*.

This page intentionally left blank

A Data Transactions

A.1 I/O Mapped Data

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

A.2 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 are the transaction status list and the exception code list.

A.3 Control/Status Word

In the direction from the PLC to the linking device, a control word can be enabled.

In the direction from the linking device to the PLC, a status word and a Modbus-TCP server live list can be enabled.

The control/status word can always be retrieved using acyclic access. The control/status word can also be I/O mapped as cyclic process data as a configuration parameter in the user interface. See ..(add link to Ethernet IP Adapter Interface).

The control word is a 16-bit word (uint16) used by the PLC on the EtherNet/IP network to control the linking device and thereby 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 PLC on 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

A.4 Live List

The PLC on the EtherNet/IP network can retrieve a live list containing the status of every transaction on the Modbus-TCP network.

The live list is always accessible to the PLC on the EtherNet/IP network using acyclic parameter access. By default, the live list is also mapped as the first eight bytes after the status word (if enabled) to the cyclic process data in the direction from the linking device to the PLC. It is possible to enable/disable the process data mapping of the live list. See .... "EtherNet/IP (Adapter Interface)".

If both the status word and live list are enabled, then the status word is mapped to byte 0–1 and the live list is mapped to byte 2–9.

If the status word is disabled and the live list is enabled, then the live list is mapped to byte 0-7.

All transactions and their positions in the live list are also visible in the Transaction Monitor in the configuration interface.

The live list consists of a bit array with 64 elements, where each bit corresponds to a transaction on the Modbus-TCP network:

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.

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.

A.4.1 Example

Consider the follow 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 the following live list when 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, transac-tion 2	Server 1, transaction 1
0	0	1	1	1	1	1	1

A.5 Transaction Status List

The transaction status list holds information about the transactions between the Modbus-TCP servers and the linking device, from the perspective of the linking device in the role as Modbus-TCP client.

The transaction status list is always accessible to the PLC on the EtherNet/IP network using acyclic parameter access.

The transaction status list contains a byte array with 64 elements. Each byte contains a transaction status code according to 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

A.6 Exception Code List

If a failure occurs during a transaction between the linking device and a Modbus-TCP server, the Modbus-TCP server may respond with an exception code.

The exception code list is always accessible to the PLC on the Ethernet/IP network using acyclic parameter access .

The exception code list contains a byte array with 64 elements. Each byte contains a transaction exception code defined in the table below. The indexes in the exception code list correspond 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

Standard Modbus Exception Codes (continued)

Exception Code	Description
08	Memory parity error
0A	Gateway path unavailable
0B	Gateway target device failed to respond

The exception codes found in the exception code list are only relevant if the corresponding transaction status codes equals 3: "Modbus exception".

If the exception code sent by the Modbus-TCP server is missing in the standard Modbus-TCP exception code list, refer to the Modbus-TCP server documentation for details.

B 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 EtherNet/IP to Modbus-TCP 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 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, see 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	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

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.

C Linking Device Configuration Manager Interface

C.1 Menu Options

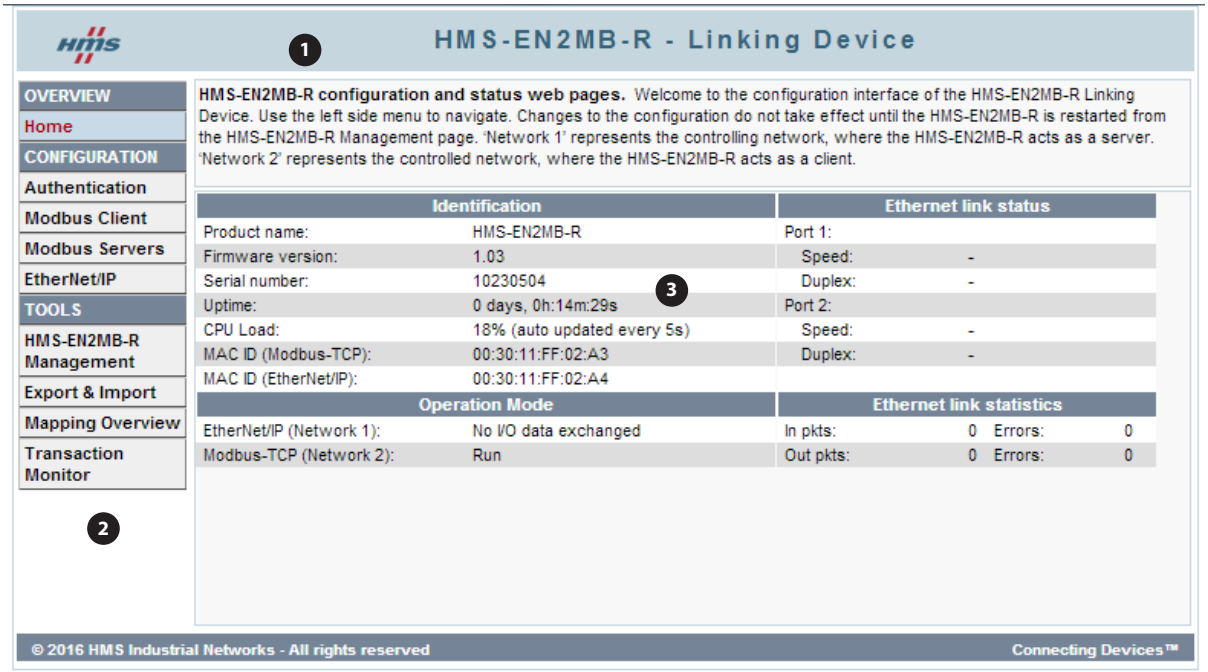
File	Import	Imports a configuration.
	Export	Exports a configuration.
	Exit	
Edit	Select All	
	Deselect All	
Tools	Edit Modbus Configuration	Launches the Modbus-TCP Master user interface.
	Launch IPconfig	Scans for devices and configure IP settings for the linking device. **Caution** Changing the IP address of the linking device without changing the IP address in Studio 5000 will make the linking device unreachable for the PLC and Studio 5000. It is recommended to change the IP address through the Studio 5000 BOOTP tools to guarantee synchronization.
	Create Mapping Report	Exports a detailed list of all tags, including the address offset in each assembly where the tag is located in the Ethernet/IP process data. Use for documentation or error tracking. Not required for function.
	Factory Reset	Resets the linking device to default values.
	Reboot	Sends reboot command to the linking device.
	Restore Configuration	Reverts to the latest stored configuration from Studio 5000.
	Wink	Makes all left LEDs blink. Use to identify the particular linking device among multiple installations.
	Options	Hides report on retrieve configuration from gateway. Groups tags on PLC by Modbus server.
Help	About	

C.2 Tag Editor Options

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.

D Modbus-TCP Client Configuration Manager

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**

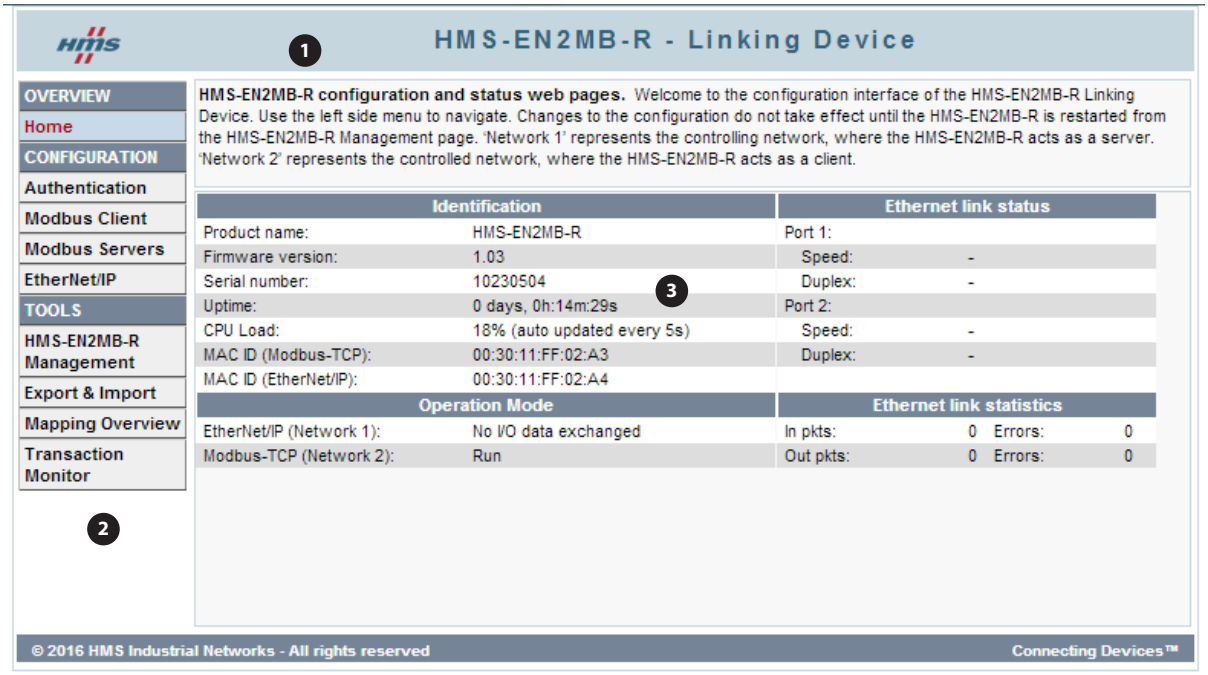
   All functionality is easily accessed from the different links. Every link and its corresponding functionality will be explained later in this chapter.

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.

D.1 Home

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



D.1.1 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:

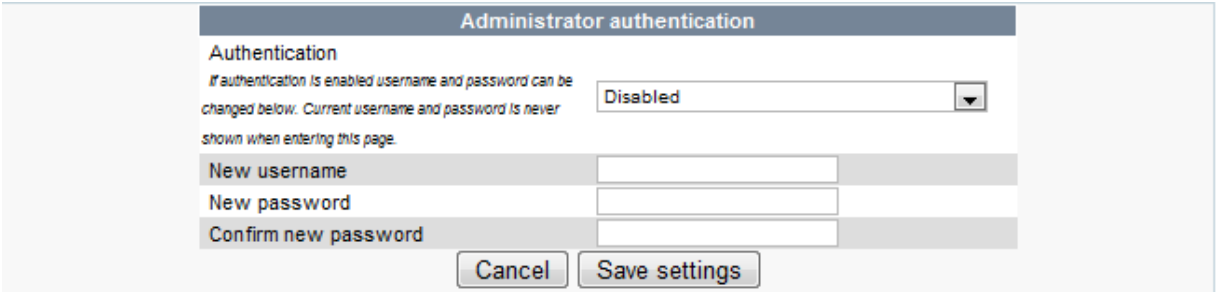
EtherNet/IP (Network 1)	**Description**
Error	Class 1 connection error. Duplicate IP address detected.
Shutdown	Unexpected error (the linking device needs a restart).

D.2 Configuration

D.2.1 Authentication

Select **Authentication** in the left-hand menu in the configuration manager.

Authentication can be enabled or disabled. If enabled, you can 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.

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

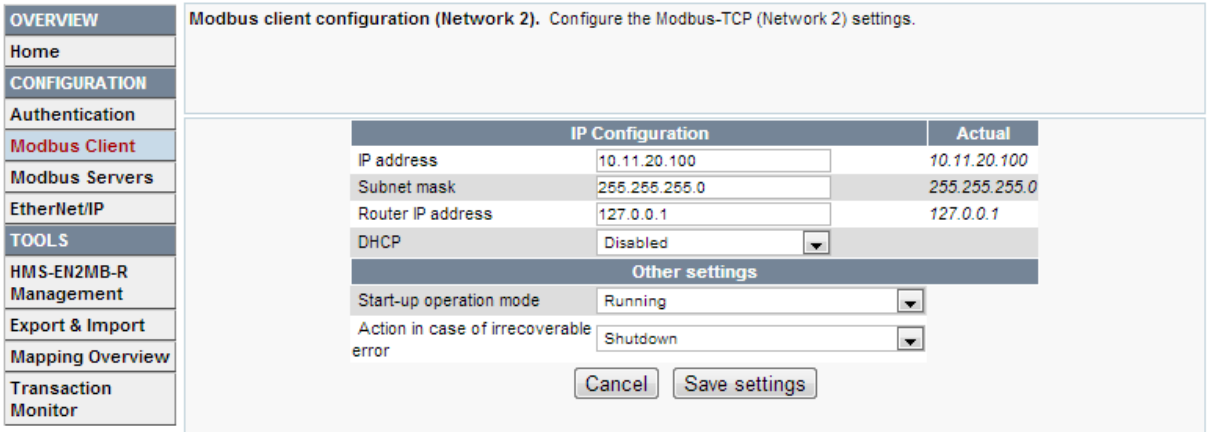
Save the authentication information. There is no way to retrieve a lost username or password. If the authentication information is lost, the only way to restore the linking device is to download new firmware via the USB interface. This will erase any configuration currently on the module.

D.2.2 Modbus Client

Select **Modbus Client** to configure 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.

Click **Save settings** when you are finished configuring the Modbus-TCP client.

Changes made to the configuration will not be used by the linking device before they are applied and saved.



IP Configuration Settings	
Item	**Description**
IP address	If not set by DHCP (or HICP), set these values manually.

IP Configuration Settings (continued)

Item	Description
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

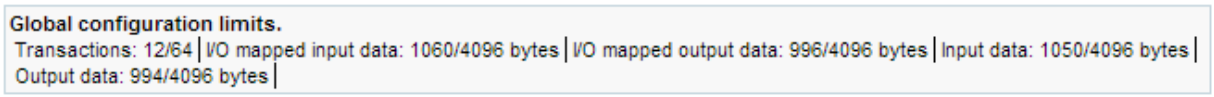
Value	Description
Shutdown	The linking device will shut down. All LEDs will display red.
Restart	The linking device will restart.

D.2.3 Modbus Servers

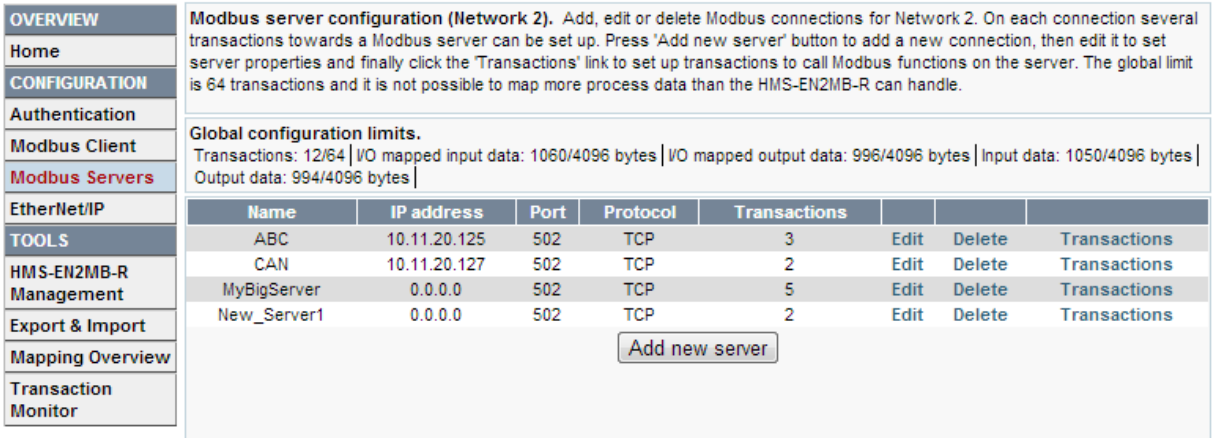
Select **Modbus Servers** to configure servers on the Modbus-TCP network.

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).



Click **Add new server** to add a server to the configuration. Click **Edit** to see and edit the settings. Click **OK** when the server is configured.



Server 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.

Transactions should be added to the server when the server is configured. It's only possible to have one server without specified transactions.

Add Transactions

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

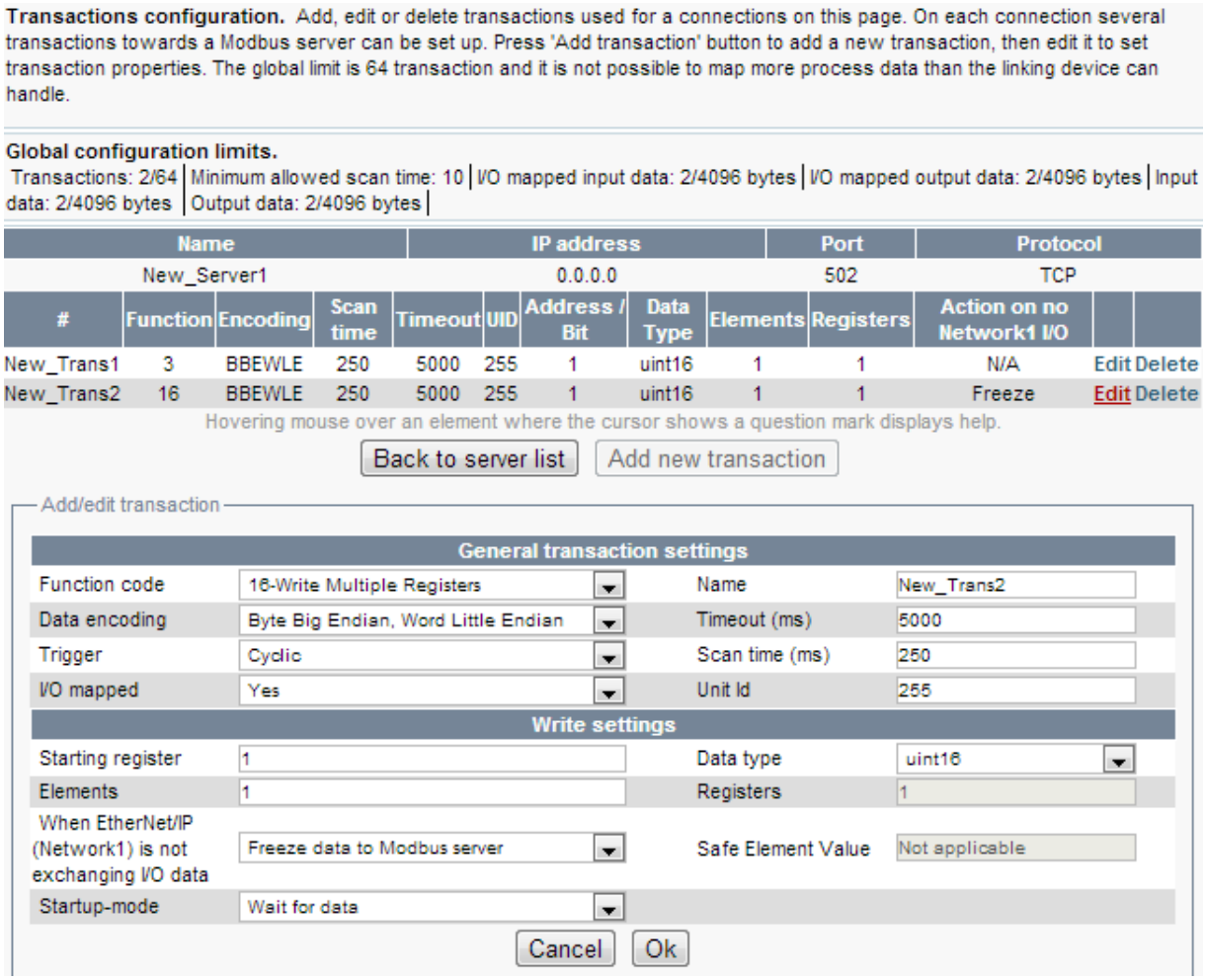
The global configuration limits box keeps track of the number of added transactions, the current minimum allowed scan time, and the current amount of I/O mapped data as well as total amount of data (both I/O mapped and not I/O mapped data).

To add transactions, select the server in the server list and click **Transactions**. This shows a list of all transactions configured for that server.

Click **Add new transaction** to add a new default transaction to the list and click **Edit**.

Click **OK** when the transaction is configured. 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.

The linking device needs to be restarted before any changes will take effect.



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

Transaction Settings

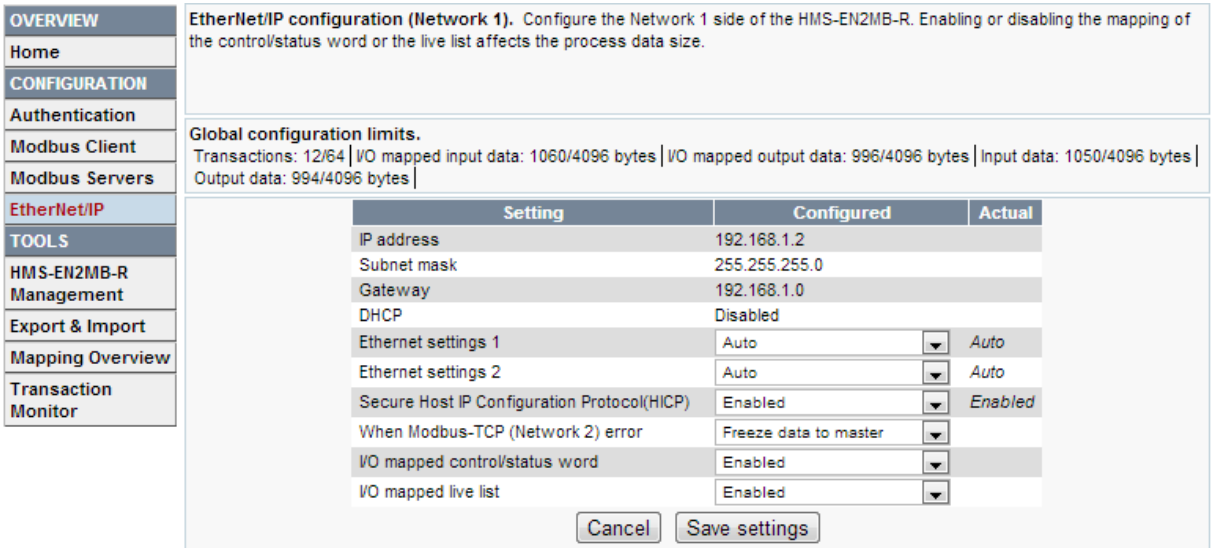
Setting	Description
Function code	The function code defines the purpose of the transaction. Choose from the available different Modbus functions.
Data encoding	Decides in what order the different bytes of the received/transmitted data shall be sent on the Modbus network.
Trigger	Only applicable for write 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.

Transaction Settings (continued)

Setting	Description
	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.
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.

D.2.4 EtherNet/IP (Adapter Interface)

Select **EtherNet/IP** in the left-hand menu to configure the EtherNet/IP adapter interface of the linking device.



The currently used settings are shown in the **Actual** column.

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

The configuration must be applied for any changes to take effect.

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.
I/O mapped live list	If enabled, the live list is mapped to the input area.

D.3 Tools

D.3.1 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 current active configuration will be undone.

Undo Changes

Undo all changes made since current active configuration.

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.

D.3.2 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.

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.

> **!** 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.

D.3.3 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 Network

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)

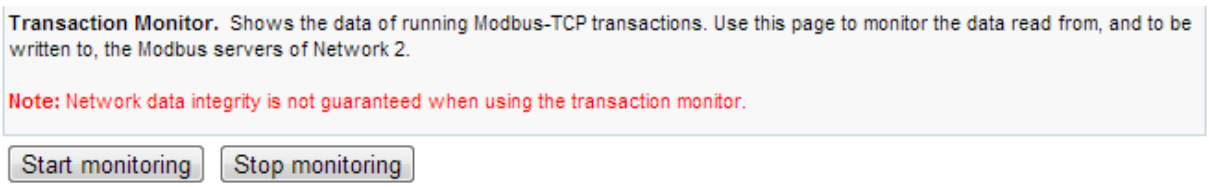
The parameter section data presents a detailed list of all data, including both the I/O mapped and the not I/O mapped data, available acyclically from the linking device to the EtherNet/IP network. This list also includes the transaction status and exception code lists, available for error identification.

Modbus-TCP Network

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

D.3.4 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.

Viewing the transaction monitor may affect performance.

E CIP Objects

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

Mandatory Objects:

Manufacturer Specific Objects:

E.1 Identity Object (01h)

Category

Extended

Object Description

The Identity Object provides identification of and general information about the module.

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 Attributes

#	Name	Access	Type	Value/Description
1	Vendor ID	Get	UINT	005Ah (HMS Industrial Networks AB)
2	Device Type	Get	UINT	000Ch (Communications Adapter)

#	Name	Access	Type	Value/Description
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	See Device Status table below
6	Serial Number	Get	UDINT	Unique serial number (assigned by HMS)
7	Product Name	Get	SHORT_STRING	“EtherNet/IP to Modbus-TCP Linking Device”
11	Active language	Set	Struct of: USINT USINT USINT	Only English supported.
12	Supported Language List	Get	Array of: Struct of: USINT USINT USINT	Only English supported.

Device Status

Bit(s)	Name
0	Module Owned
1	(reserved)
2	Configured This bit shows if the product has other settings than "out-of-box".
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)

Service Details: Reset

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

Type 0: Power Cycling Reset	This service power cycles the module.
Type 1: Out of box reset	This service sets an “out of box” configuration (all configuration data will be erased) and performs a reset.

E.2 Message Router (02h)

Category

Extended

Object Description

The Message Router Object provides a messaging connection point through which a client may address a service to any object class or instance residing in the physical module.

In the HMS-EN2MB-R module it is used internally to direct object requests.

Supported Services

Class:	-
Instance:	-

Class Attributes

-

Instance Attributes

-

E.3 Assembly Object (04h)

Category

Extended

Object Description

The Assembly object uses static assemblies and holds the Process Data sent/received by the host application. It allows data to and from each object to be sent or received over a single connection. The default assembly instance IDs used are 100-109 and 150-159.

The terms “input” and “output” are defined from the network’s point of view. An input will produce data on the network and an output will consume data from the network.

See also ....

- Assembly Mapping Object (see Anybus CompactCom 40 Software Design Guide)

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	Maximum instance number

Instance 03h Attributes (Heartbeat, Input-Only)

This instance is used as heartbeat for Input-Only connections. The data size of the Heartbeat instance in the Forward_Open-request should be 0 bytes, however other values are also permitted.

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance 04h Attributes (Heartbeat, Listen-Only)

This instance is used as heartbeat for listen-only connections. The data size of the Heartbeat instance in the Forward_Open-request should be 0 bytes, however other values are also permitted.

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance 05h Attributes (Configuration Data)

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

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (Configuration data written to the application when the forward open command has the configuration data included)- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance C6h Attributes (Heartbeat, Input-Only)

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

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance C7h Attributes (Heartbeat, Listen-Only)

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

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance C3h Attributes (Heartbeat, Input-Only)

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 Error state.

Instance C4h Attributes (Heartbeat, Listen-Only)

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/Description
3	Produced Data	Get	Array of BYTE	This data corresponds to the Write Process Data.

Instance 96h Attributes (Consuming Instance)

#	Name	Access	Type	Value/Description
3	Produced Data	Get	Array of BYTE	This data corresponds to the Read Process Data.

E.4 Connection Manager (06h)

Category

Extended

Object Description

-

Supported Services

Class:	-
Instance:	Forward_Open
	Forward_Close

Class Attributes

(No supported class attributes)

Instance Attributes

(No supported instance attributes.)

Class 0 Connection Details

Not supported.

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 RPI (Requested Packet Interval):	2... 3200 ms
T→O Connection type:	Point-to-point, Multicast
O→-T Connection type:	Point-to-point
Supported trigger types:	Cyclic, CoS (Change of State)

T	Target, in this case the module
O	Origin, in this case the master

Connection Types

- **Exclusive-Owner connection**

  This type of connection controls the outputs of the HMS-EN2MB-R 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**

  Max. no. of Input-Only connections: Up to 4

  Connection point O →T: Assembly Object, instance 03h (Default)

  Connection point T →O: Assembly Object, instance 96h (Default)

  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

  Connection point O →T: Assembly Object, instance 05h (Default)

  Connection point T →O: Assembly Object, instance 96h (Default)

- **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 Input-Only connections: Up to 4

  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 Input-Only connections: Up to 4

  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 gateway.

Class 3 Connection Details

General

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 RPI (Requested Packet Interval):	2... 10000 ms
T→O Connection type:	Point-to-point
O→-T Connection type:	Point-to-point
Supported trigger type:	Application

E.5 DLR Object (47h)

Category

Extended

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 Attributes

Attributes #1–4 and #6–7 an be customized by implementing the EtherNet/IP Host Object.

#	Name	Access	Type	Value/Description
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”

#	Name	Access	Type	Value/Description
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)

E.6 QoS Object (48h)

Category

Extended

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 Attributes

#	Name	Access	Type	Value/Description
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

E.7 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	0001h (Object revision)

Instance 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: 0 1	Meaning: Get access Set access
5	Value	Get/Set	Determined by attribute #2	Instance value	
6	Max value	Get		The maximum permitted parameter value	
7	Min value	Get		The minimum permitted parameter value	
8	Default value	Get		The default parameter value	

E.8 Port Object (F4h)

Object Description

-

Supported Services

Class:	Get_Attribute_Single
	Get_Attributes_All
Instance:	Get_Attribute_Single
	Get_Attributes_All

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0001h (Object revision)
2	Max instance	Get	UINT	Max. instance number
3	Number of instances	Get	UINT	Number of ports

#	Name	Access	Type	Value
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)	-

E.9 TCP/IP Interface Object (F5h)

Category

Extended

Object Description

This object provides the mechanism to configure the TCP/IP network interface of the module. It groups the TCP/IP-related settings.

Supported Services

Class:	Get_Attribute_All
	Get_Attribute_Single
Instance:	Get_Attribute_All
	Get_Attribute_Single
	Set_Attribute_Single

Class Attributes

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

Instance Attributes

#	Name	Access	Type	Value	Comments
1	Status	Get	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	Configuration Capability	Get	DWORD	0000 0016h - or - 0000 0006h - or - 0000 0026 - 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	Configuration Control	Get/Set	DWORD	-	Value: Meaning 0: Configuration from non-volatile memory 2: Configuration from DHCP
4	Physical Link Object	Get	Struct of: UINT (Path size) Padded EPATH	- 0002h 20 F6 24 03h	- - Path to Ethernet Link Object, Instance #3
5	Interface Configuration	Get/Set	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	Host Name	Get/Set	STRING	-	Host name of the HMS-EN2MB-R
8	TTL Value	Get/Set	USINT	1	TTL value for EtherNet/IP multicast packets
9	Mcast Config	Set	Struct of:		IP multicast configuration
	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	SelectAcd	Set	Bool	1	Value: Meaning: 0: Disable ACD

#	Name	Access	Type	Value	Comments
					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.
11	LastConflictDetected	Set	Struct of:		ACD Diagnostic parameters Related to the last conflict detected.
	AcdActiviity		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.

E.10 Ethernet Link Object (F6h)

Category

Extended

Object Description

This object maintains link specific counters and status information for an IEEE 802.3 communications interface. Exactly one instance for each communications interface on the module is supported. Instances for internally accessible interfaces can also be supported.

Supported Services

Class:	Get_Attributes_All
	Get_Attribute_Single
Instance:	Get_Attributes_All
	Get_Attribute_Single
	Set_Attribute_Single
	Get_And_Clear

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0003h (Object revision)
2	Max Instance	Get	UINT	3 (Maximum instance number)
3	Number of Instances	Get	UINT	3 (Number of instances)

Instance Attributes

#	Name	Access	Type	Value	Comments
1	Interface Speed	Get	UDINT	10 or 100	Actual Ethernet interface speed.
2	Interface Flags	Get	DWORD	-	See table “Interface Flags” below.
3	Physical Address	Get	Array of 6 USINTs	(MAC ID)	Physical network address, i.e. assigned MAC address.

#	Name	Access	Type	Value	Comments

#	Name	Access	Type	Value	Comments
6	Interface Control	Get/Set	Struct of:		
	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	Interface Type	Get	USINT	-	See table "Interface Type" below.
10	Interface Label	Get	SHORT_STRING	—	See table "Interface Label" below.

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 auto-negotiation. Value: Meaning: 0 Auto-negotiation in progress. 1 Auto-negotiation and speed detection failed (using default values) (Recommended default values are 10 Mbps, half duplex) 2 Auto negotiation failed but detected speed (using default duplex value) 3 Successfully negotiated speed and duplex. 4 Auto-negotiation 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)	Set to 0.

Interface Label

Instance	Value
1	Port 1
2	Port 2
3	Internal

Interface Type

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

F Technical Specification

F.1 Functional Earth (FE) Requirements

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

HMS Networks AB does not guarantee proper EMC behavior unless these FE requirements are fulfilled.

Make sure the DIN-rail is properly connected to FE.

F.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.

F.3 Environmental Specification

Temperature	Operating	-25º to +70º Celsius
	Non-operating	-40º to +85º Celsius
Relative humidity		The product is designed for a relative humidity of 5% to 95% noncondensing.

G SD Card Functionality

> **!** The recommended procedure to back up a configuration is to use the backup function in Studio 5000. It is NOT recommended to use the SD card for backups.

> **!** The SD card acts as client in the linking device. When a 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. This may result in a checksum missmatch, and the Linking Device will reject any connection attempts from the PLC.

G.1 General 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.

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

G.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.

G.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.

G.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.

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

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.

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

G.5 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 cad 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”.

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.

*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.

H 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.

Copyright (c) 2008-2013 Marshall A. Greenblatt. Portions Copyright (c)

2006-2009 Google Inc. All rights reserved.

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 Google Inc. nor the name Chromium Embedded Framework 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 COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY 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 COPYRIGHT OWNER 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.

This page intentionally left blank

Box 4126
300 04 Halmstad, Sweden

info@hms.se

scm-1202–181 2.0 en-US / 2021-05-19 / 22485