460ESMM EtherNet/IP Modbus Master Device Mapping

The 460ESMM moves registers and coils between a Modbus RTU network of Slave devices and an EtherNet/IP Client like a ControlLogix® PLC. There are thousands of Modbus RTU devices in automation systems. These devices include weigh scales, drives, barcode readers and sensors of every type imaginable.

The 460ESMM deploys quickly and easily configures so you can access and integrate these Modbus devices into your EtherNet/IP network.

Protocols

Modbus RTU is an open, serial (RS-485) protocol derived from the client/server architecture. It is a widely accepted protocol due to its ease of use and reliability. Modbus RTU is widely used within Building Management Systems (BMS) and Industrial Automation Systems (IAS). This wide acceptance is due in large part to Modbus RTU's ease of use.

Modbus devices represent their data as 16-bit unsigned registers and single bit coils. The data contained in these registers varies from device to device with some register types containing field inputs (input registers), internal registers (holding registers), field coils (input status) and internal coils (coil status). One Modbus Master device sends read/write commands to read or write the registers or coils. The ease of implementation (RS485 serial port) and the simplicity of these messages ensures reliability and has made Modbus an enduring choice for industrial communication.

EtherNet/IP is an Ethernet application layer protocol that is transferred inside a TCP/IP Packet. That means that EtherNet/IP is simply a way to organize the bits and bytes transferred in a TCP or UDP packet. All devices on an EtherNet/IP network represent their data to the network as a series of data values called attributes grouped with other similar data values into sets of attributes called Objects.

EtherNet/IP is part of CIP, the Common Industrial Protocol. CIP specifies a set of Required Objects that every EtherNet/IP device must have – Identity, EtherNet/IP, TCP, Router and Connection objects to name a few. There are also EtherNet/IP Application Objects. These objects represent the data contained in an automation device as a set of Objects containing attributes. For example, an EtherNet/IP Drive device has a Motor Object with attributes for Motor Speed and Motor Size. Device profiles are sets of Objects that are used in every device of a specific type. A device supporting the EtherNet/IP Drive Profile supports the same objects as all other devices supporting the Drive profile.

There are two kinds of messages that are transferred between an EtherNet/IP Scanner Device (opens connections and initiates data transfers) and EtherNet/IP Adapter devices (provides data to Scanners). These messages are Explicit Messages (asynchronous, as needed) and I/O Messages (Data messages that are continuously transferred).

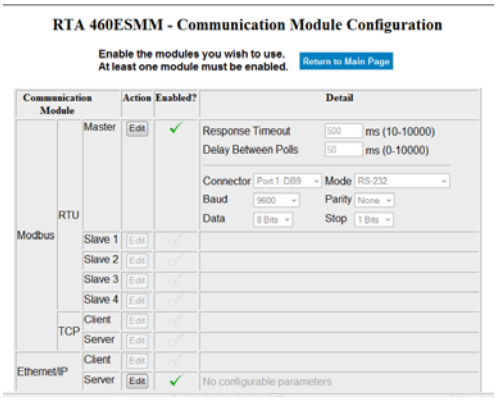
Mapping In the 460ESMM

With the 460ESMM module you can both read and write registers and coils from/to a Modbus RTU slave device. Up to 160 Registers of data can be transferred to and from up to 31 Modbus RTU Slave devices.

All the data transfer is configurable using the embedded web server. You define the number of Modbus RTU Slave devices. You set one or more configurable groups of registers to transfer from each slave and in what order that those registers are transferred to the EtherNet/IP Client.

Setting up mapping for the 460ESMM is easy.

From any browser, enter the device IP address to get to the main configuration page. Click the “Edit” button for network settings and enter your IP address, the gateway address, and set your network mask to configure your device.



From the main page, click the “Edit” button for the Communication Modules section to configure the module. Go in and set your Response Timeout, Connector, and Baud rate.

Device Data Mapping — Two data buffers are used to communicate with the 460ESMM over EtherNet/IP. One buffer contains input data transferred to an EtherNet/IP Scanner and one contains output data transferred to the 460ESMM.

Byte	Modbus Device Data to EtherNet/IP Scanner	Word
0	Input Device Data Buffers	0
-		-
-		-
-		-
-		-
319		159
320	Pass Through Data Buffers	160
-		-
-		-
-		-
395		197
396	Status Data Buffers	198
-		-
399		199

Bytes 0 through 319 are set aside for the Modbus device data defined in the web server. You set the register list that is read and written from each device and that data is mapped to the input and output device buffers of the 460ESMM where it is exchanged with the EtherNet/IP scanner.

Bytes 320 through 395 are used for Pass through Messaging (PTM). PTM messages are passed directly from the EtherNet/IP Scanner to your device. Responses from the Modbus device are passed directly back to the Scanner..

Bytes 396 through 399 in the Input buffer provide Status data for each Modbus RTU device.

Byte	EtherNet/IP Scanner Data to Modbus Device	Word
0	Output Device Data Buffers	0
-		-
-		-
-		-
-		-
319		159
320	Pass Through Data Buffers	160
-		-
-		-
-		-
395		197
396	Reserved Buffers	198
-		-
399		199

Input/Output Device Data Buffers

An EtherNet/IP Scanner moves data from its buffer to the 460ESMM Output Buffer and data in the 460ESMM Input Buffer to the Scanner. You configure the address and the number of words and bytes of each Modbus device that is included in each of these buffers.

Byte	Modbus Device Data to EtherNet/IP Scanner (EtherNet/IP Input)	Word
0	Modbus Device # 1	0
n		w
	Modbus Device # 2	
n1		w1
	Modbus Device # 3	
n2		w2
	Modbus Device # n	
319		159

Byte	EtherNet/IP Scanner Data to Modbus Device (EtherNet/IP Output)	Word
0	Modbus Device # 1	0
n		w
	Modbus Device # 2	
n1		w1
	Modbus Device # 3	
n2		w2
	Modbus Device # n	
319		159

Device Data Buffer Example

Modbus Device Data to EtherNet/IP Input Data Buffer

Data Read from Modbus Devices

Device 1	55	21	7	6
Device 2	15	1	17	

Byte	EtherNet/IP Input Data Buffer		Word
0 - 1	55	21	0
2 - 3	7	6	1
4 - 5	15	1	2
6 - 7	17		3

EtherNet/IP Input i/O Message

EtherNet/IP Scanner

EtherNet/IP Output Data to Modbus Input Data Buffer

EtherNet/IP Scanner

Byte	EtherNet/IP Output Data Buffer		Word
0 - 1	9	23	0
2 - 3	33	13	1
4 - 5	2		2

EtherNet/IP Output I /O Message

Data Written to Modbus Devices

9	23		Device 1
33	13	2	Device 2

Pass Through Data Buffer

Bytes 320 to 395 are reserved for Pass Through Messaging (PTM). PTM is a mechanism for sending any Modbus command from an EtherNet/IP Scanner to a Modbus device through the 460ESMM. Any response message received from the Modbus network is returned to the EtherNet/IP Scanner in the Output Buffer. The 460ESMM performs no validation or inspection on the Pass Through Message or the response message.

Byte	Modbus Device Data to EtherNet/IP Scanner (Output Assembly 112)	Word
-		-
-		-
-		-
320	Pass Through Message Data	160
-		-
-		-
-		-
395		197

Byte	EtherNet/IP Scanner Data to Modbus Device (Input Assembly 100)	Word
-		-
-		-
-		-
320	Pass Through Response Data	160
-		-
-		-
-		-
395		197

Pass Through Buffer Example

Modbus Slave Device = 5
Function Code = 3 (Read Holding Registers)
Starting Address = 500
Number of Registers = 1
Data = 55

Byte 320 - 321 = Length Field
Byte 322 - 323 = Sequence Number
Bye 324 -395 = Modbus Message

Byte	Pass Through Message	Word
-		-
-		-
320 - 321	6	160
322 - 323	1	161
324	5	162 (Hi Byte)
325	3	162 (Low Byte)
326 - 327	500	163
328 - 329	1	164
-		-
-		-
395		197

Byte	Pass Through Response	Word
-		-
-		-
320 - 321	5	160
322 - 323	1	161
324	5	162 (Hi Byte)
325	3	162 (Low Byte)
326	2	163 (Hi Byte)
327 - 328	55	163 (Low Byte) 164 (Hi Byte)
-		-
-		-
395		197

Status Data Buffers

The last two input registers are reserved for Device Status Data. Users can look at the data in these registers to monitor the status of all 31 Modbus devices with your Ethernet/IP Client status at Bit 0 of word 198 .

Example 1:

Byte	EtherNet/IP Input Buffers	Words
0		0
-		-
-		-
-		-
-		-
-		-
-		-
395		197
396	1111111111111101	198
398	1111101111111111	199

Device 2 and 27 may or may not be mapped, but currently not connected.

Example 2:

Byte	EtherNet/IP Input Buffers	Words
0		0
-		-
-		-
-		-
-		-
-		-
-		-
395		197
396	1111111111111111	198
398	1111111111111111	199

All mapped devices are connected and communicating properly.

