

# TECHNICAL NOTE

## Maple Model(s)

HMI5000 Series  
cMT Series  
RMI Series

## Title

Using the Modbus Gateway

TN5099

P/N: 0907-5099

Rev. 02 Date: 11/03/2022



## Summary

Maple Systems EBPro and cMT and HMI5000 series products can bridge the gap between your SCADA system and field devices. Using the HMI's Modbus RTU/TCP Slave protocol, the Gateway provides a mechanism to connect the Modbus RTU/TCP Slave protocol to storage within the HMI or in a connected device. The SCADA system uses the ModbusTCP Master protocol to query the HMI using the HMI's own IP address<sup>1</sup>.

## Solution

### Contents

<b>1.0 Allow the SCADA System to access any HMI internal memory</b> .....	2
<b>2.0 Allow the SCADA System to access any device connected to the HMI</b> .....	3
2.1 Example #1, Animatics SmartMotor.....	3
2.2 Example #2, Allen Bradley DF1 .....	4
2.3 Example #3, Omron Hostlink .....	5
2.4 Example #4, Yaskawa MP-series over Ethernet .....	6
<b>3.0 Use the HMI as a ModbusTCP to ModbusRTU Bridge</b> .....	7
<b>4.0 Configure the Modbus Gateway</b> .....	9

<sup>1</sup> There may be several hundred milliseconds or more of latency as the ModbusTCP request is processed through the HMI and Modbus Gateway. The timeout setting on the ModbusTCP master should be adjusted accordingly.

This note provides a few specific examples of using the Modbus Gateway. However, the Modbus Gateway can connect to almost any device supported by the HMI and EBPro.

The Modbus Gateway can:

1. Allow the SCADA System to access any HMI internal memory.
2. Allow the SCADA System to access any device connected to the HMI.
3. Allow the HMI to act as a ModbusTCP-to-ModbusRTU bridge.

These 3 modes can be used simultaneously<sup>2</sup>.

See **4.0 Configure the Modbus Gateway** section for details on enabling the Modbus Gateway in EBPro.

### 1.0 Allow the SCADA System to access any HMI internal memory

This is the default setting, where Modbus addresses are mapped to HMI internal storage.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	0x <==> LB	0x-1	<==>	Local HMI	LB-0	12400 Bit(s)	Read/Write
2	1x <==> LB	1x-1	<==	Local HMI	LB-0	12400 Bit(s)	Read only
3	3x <==> LW	3x-1	<==	Local HMI	LW-0	9999 Word(s)	Read only
4	4x <==> LW	4x-1	<==>	Local HMI	LW-0	9999 Word(s)	Read/Write
5	3x <==> RW	3x-10000	<==	Local HMI	RW-0	55536 Word(s)	Read only
6	4x <==> RW	4x-10000	<==>	Local HMI	RW-0	55536 Word(s)	Read/Write

#### Modbus Request

0x1 – 0x12400

1x1 – 1x12400

3x1 – 3x9999

4x1 – 4x9999

3x10000 – 3x65536

4x10000 – 4x65536

#### Mapped To HMI Register

LB0 – LB12399

LB0 – LB12399

LW0 – LW9998

LW0 – LW9999

RW0 – RW55535

RW0 – RW55535

Be careful when using Local Bit and Word addresses 9000 and higher. Many addresses in this range are used by the HMI for specific functions.

<sup>2</sup> Serial connections are limited by the number of communication ports on the HMI. A large number of Modbus Gateway items may reduce the performance of the HMI.

## 2.0 Allow the SCADA System to access any device connected to the HMI

This is a custom setting, where Modbus addresses are mapped to a device with which the HMI is communicating. The following examples demonstrate a few of the many possible configurations.

### 2.1 Example #1, Animatics SmartMotor

This example shows an Animatics SmartMotor mapped to a series of Modbus registers.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	4x <-> AL	4x-1	<==>	Animatics Class 5 (Serial)	ArrayAL-0	10 Word(s)	Read/Write
2	0x <-> VarAZBit	0x-1	<==>	Animatics Class 5 (Serial)	VarAZBit-000	32 Bit(s)	Read/Write

#### Modbus Request

4x1 – 4x9

0x1 – 0x32

#### Mapped To Animatics Register

a1[0] – a1[9]

a.0-a.15, b.0 – b.15

The SmartMotor is connected to the HMI via an RS232 serial connection. This example shows how an RS232 device can be connected to an Ethernet SCADA system.

**Device Properties**

Name : Animatics Class 5 (Serial)

HMI  PLC

Location : Local [Settings...]

PLC type : Animatics Class 5 (Serial)  
V.1.02, ANIMATICS\_CL5\_SERIAL.e30

PLC I/F : RS-232

**COM Port Settings**

COM : COM 1

Baud rate : 9600

Data bits : 8 Bits

Parity : None

Stop bits : 1 Bit

Timeout (sec) : 1.5

Turn around delay (ms) : 0

Send ACK delay (ms) : 0

Parameter 1 : 0

Parameter 2 : 0

Parameter 3 : 0

The number of resending commands : 0

\* OS version 20120920 or later support 14400 baud rate

OK Cancel

OK Cancel

## 2.2 Example #2, Allen Bradley DF1

This example shows an Allen Bradley Micrologix using DF1 mapped to a series of Modbus registers.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	Outputs	0x-1	<==>	Allen-Bradley SLC500 and Micro (DF1 Serial-...	Bfn-10000000	64 Bit(s)	Read/Write
2	Inputs	1x-1	<==>	Allen-Bradley SLC500 and Micro (DF1 Serial-...	Bfn-10100000	64 Bit(s)	Read/Write
3	Analogs	3x-1	<==>	Allen-Bradley SLC500 and Micro (DF1 Serial-...	Nfn-110000	32 Word(s)	Read/Write
4	Timers	4x-1	<==>	Allen-Bradley SLC500 and Micro (DF1 Serial-...	T4SV-020000	10 Word(s)	Read/Write
5	Values	4x-11	<==>	Allen-Bradley SLC500 and Micro (DF1 Serial-...	Nfn-120000	32 Word(s)	Read/Write

### Modbus Request

0x1 – 0x64

1x1 – 1x64

3x1 – 3x32

4x1 – 4x10

4x11 – 4x42

### Mapped To AB Register

B100:0.0 – B100:3.15

B101:0.0 – B101:3.15

N110:0 – N110:31

T20:0.PRE – T30:0.PRE

N120:0 – N120:31

The AB PLC is connected to the HMI via an RS232 serial connection. This example shows how an RS232 device can be connected to an Ethernet SCADA system.

The screenshot shows the 'Device Properties' window for an Allen-Bradley SLC500 and Micro (DF1 Serial-CRC) PLC. The 'Name' field is filled with 'Allen-Bradley SLC500 and Micro (DF1 Serial-CRC)'. The 'Location' is set to 'Local'. The 'PLC type' is 'Allen-Bradley SLC500 and Micro (DF1 Serial-CRC)' with version 'V.2.70, ALLEN\_BRADLEY\_DF1.e30'. The 'PLC I/F' is 'RS-232'. A 'COM Port Settings' sub-dialog is open, showing 'COM 1', 'Baud rate: 9600', 'Data bits: 8 Bits', 'Parity: None', 'Stop bits: 1 Bit', 'Timeout (sec): 1.0', 'Turn around delay (ms): 0', 'Send ACK delay (ms): 0', 'Parameter 1: 0', 'Parameter 2: 0', 'Parameter 3: 0', and 'The number of resending commands: 0'. The dialog has 'OK' and 'Cancel' buttons at the bottom.

### 2.3 Example #3, Omron Hostlink

This example shows an Omron CS/CJ using Hostlink mapped to a series of Modbus registers.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	Outputs	0x-1	<==>	Omron CJ1 and CS1 Series	CIO_Bit-0	64 Bit(s)	Read/Write
2	Inputs	1x-1	<==	Omron CJ1 and CS1 Series	CIO_Bit-65	64 Bit(s)	Read only
3	Analogs	3x-1	<==	Omron CJ1 and CS1 Series	CIO-200	32 Word(s)	Read only
4	Holding	4x-1	<==>	Omron CJ1 and CS1 Series	H-0	32 Word(s)	Read/Write

#### Modbus Request

0x1 – 0x64

1x1 – 1x64

3x1 – 3x32

4x1 – 4x32

#### Mapped To Omron Register

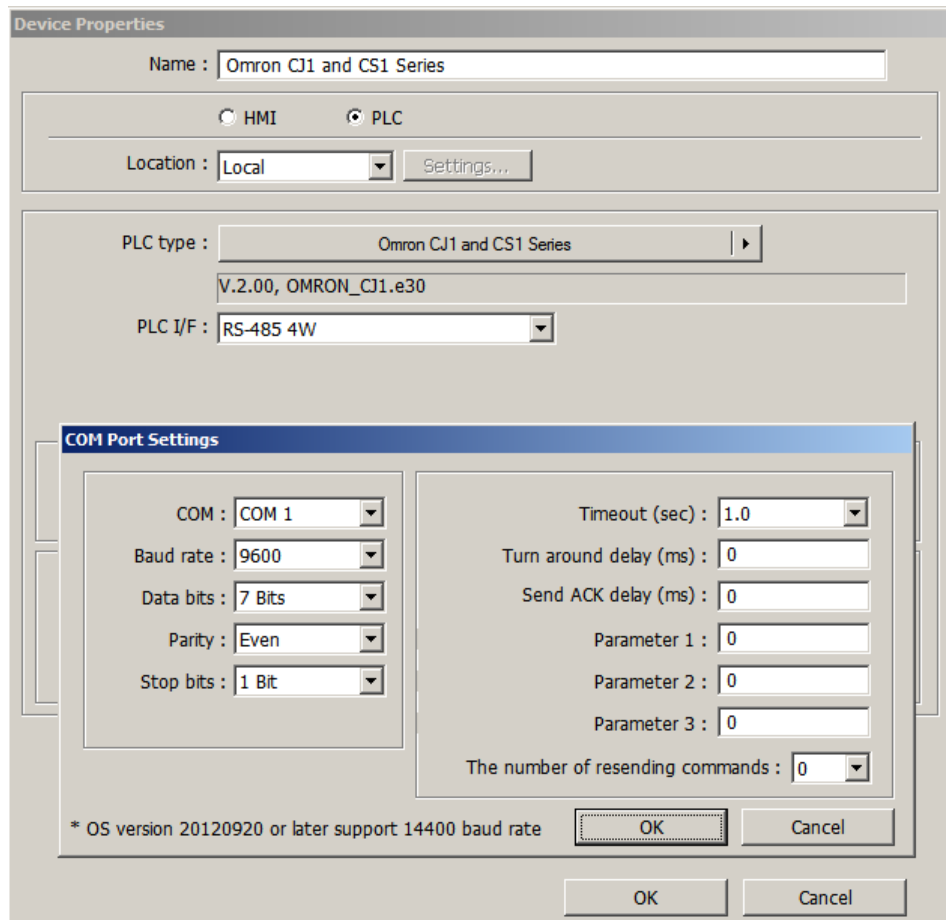
CIO0.00 – CIO3.15

CIO4.00 – CIO7.15

CIO220 – CIO251

H0 – H31

The Omron PLC is connected to the HMI via an RS485 4-wire serial connection. This example shows how an RS485 4-wire device can be connected to an Ethernet SCADA system.



## 2.4 Example #4, Yaskawa MP-series over Ethernet

This demonstrates that the Modbus Gateway can be used to connect a SCADA system to an Ethernet device that the SCADA system does not support, but that the HMI does.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	Outputs	0x-1	<==>	Yaskawa MP Series Ethernet (Extension)	OB-0	64 Bit(s)	Read/Write
2	Values	4x-1	<==>	Yaskawa MP Series Ethernet (Extension)	MW-0	64 Word(s)	Read/Write

### Modbus Request

0x1 – 0x64

4x1 – 4x64

### Mapped To Yaskawa Register

OB0 – OB40

MW0 – MW63

The screenshot shows the 'Device Properties' window for a 'Yaskawa MP Series Ethernet (Extension)' device. The 'Name' field is filled with 'Yaskawa MP Series Ethernet (Extension)'. The 'Location' is set to 'Local'. The 'PLC type' is 'Yaskawa MP Series Ethernet (Extension)' and the 'PLC I/F' is 'Ethernet'. An 'IP Address Settings' dialog is overlaid, showing an IP address of '192 . 168 . 1 . 100' and a 'Port no.' of '10000'. Below this, there are fields for 'Timeout (sec): 1.0', 'Turn around delay (ms): 0', 'Send ACK delay (ms): 0', 'Parameter 1: 0', 'Parameter 2: 0', 'Parameter 3: 0', and 'The number of resending commands: 0'. 'OK' and 'Cancel' buttons are at the bottom.

### 3.0 Use the HMI as a ModbusTCP to ModbusRTU Bridge

This example shows a number of Modbus RTU devices mapped to a series of Modbus registers. In the *Mapped PLC Address* column, the pound-sign notation is used to specify the station number followed by the starting register in the specified station. For more information on specifying station numbers, please refer to the EBPro Programming Manual Chapter 37, page 37-1.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	RTU Device #1	4x-1	<==>	Modbus RTU Master	4x-1#1	32 Word(s)	Read/Write
2	RTU Device #2	4x-33	<==>	Modbus RTU Master	4x-2#1	32 Word(s)	Read/Write
3	RTU Device #3	4x-65	<==>	Modbus RTU Master	4x-3#1	32 Word(s)	Read/Write
4	RTU Device #4	4x-97	<==>	Modbus RTU Master	4x-4#1	32 Word(s)	Read/Write
5	RTU Device #5	4x-129	<==>	Modbus RTU Master	4x-5#1	32 Word(s)	Read/Write
6	RTU Device #6	4x-161	<==>	Modbus RTU Master	4x-6#1	32 Word(s)	Read/Write
7	RTU Device #7	4x-193	<==>	Modbus RTU Master	4x-7#1	32 Word(s)	Read/Write
8	RTU Device #8	4x-225	<==>	Modbus RTU Master	4x-8#1	32 Word(s)	Read/Write
9	RTU Device #9	4x-257	<==>	Modbus RTU Master	4x-9#1	32 Word(s)	Read/Write
10	RTU Device #10	4x-289	<==>	Modbus RTU Master	4x-10#1	32 Word(s)	Read/Write

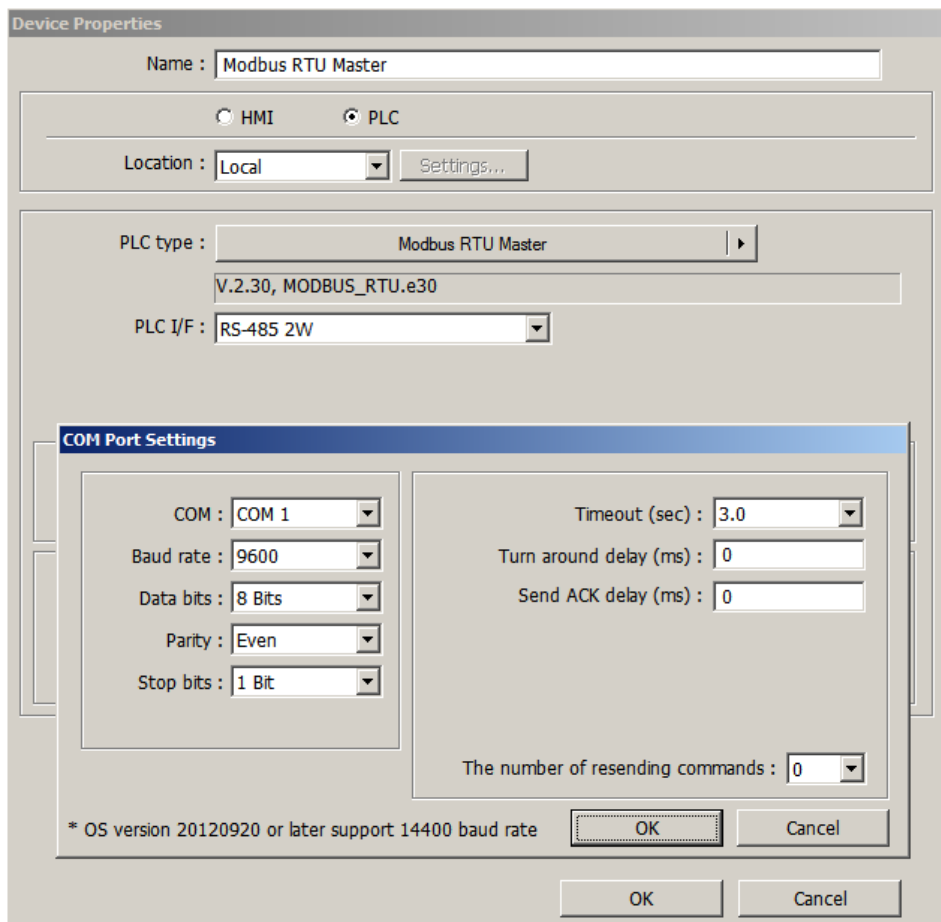
#### Modbus Request

4x1 – 4x32  
 4x33 - 4x64  
 4x65 – 4x96  
 4x97 – 4x128  
 4x129 – 4x160  
 4x161 – 4x192  
 4x193 – 4x224  
 4x225 – 4x256  
 4x257 – 4x288  
 4x289 – 4x320

#### Mapped To Modbus RTU Register

4x1 – 4x32 in Modbus RTU Station #1  
 4x1 – 4x32 in Modbus RTU Station #2  
 4x1 – 4x32 in Modbus RTU Station #3  
 4x1 – 4x32 in Modbus RTU Station #4  
 4x1 – 4x32 in Modbus RTU Station #5  
 4x1 – 4x32 in Modbus RTU Station #6  
 4x1 – 4x32 in Modbus RTU Station #7  
 4x1 – 4x32 in Modbus RTU Station #8  
 4x1 – 4x32 in Modbus RTU Station #9  
 4x1 – 4x32 in Modbus RTU Station #10

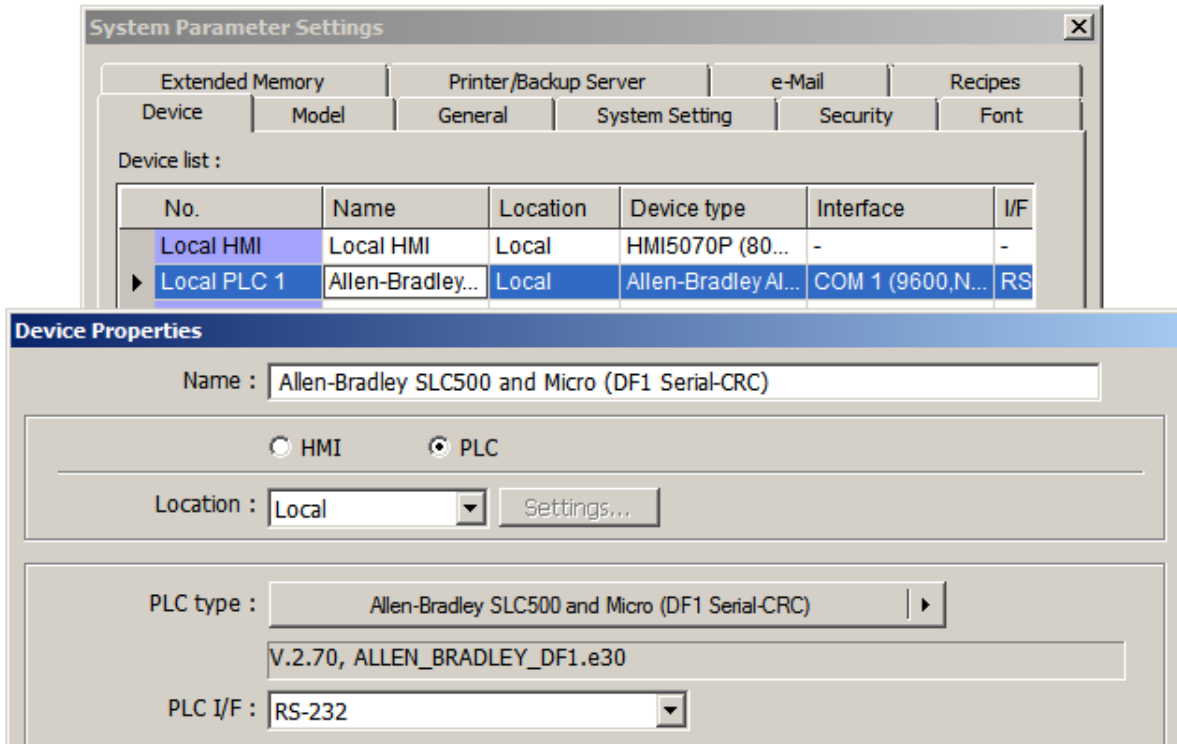
The Modbus RTU devices are connected to the HMI via an RS485 2-wire serial connection. This example shows how a network of RS485 2-wire serial devices can be connected to an Ethernet SCADA system. An RS485 4-wire network would be set up the same way.





#### 4.0 Configure the Modbus Gateway

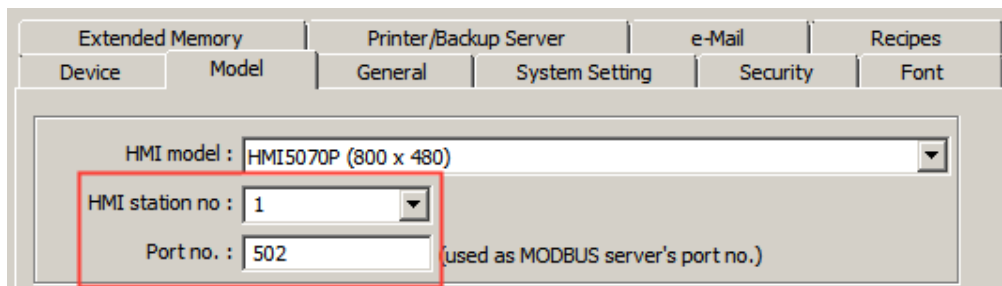
First, add the protocol to which Modbus registers will be mapped. Please refer to the EZwarePlus Programming manual or the software help system for more information. In the example below, the *Allen-Bradley SLC500 and Micro (DF1 Serial-CRC)* protocol will be used.



Next, configure the Modbus RTU/TCP Slave protocol in the HMI.

On the Model tab of System Parameter dialog:

- Change the HMI Station Number to the Modbus Unit Number desired for the HMI.
- Change the Port Number to the port used by the ModbusTCP Master. Typically, Port 502 is used.



Add the **Modbus RTU/TCP Slave** protocol to the project, and select the **PLC I/F** as *Ethernet*.

The screenshot shows the 'Device Properties' dialog box for a 'Modbus RTU/TCP Slave' device. The 'Name' field is set to 'Modbus RTU/TCP Slave'. The 'Device Type' is set to 'PLC'. The 'Location' is set to 'Local'. The 'PLC type' is set to 'Modbus RTU/TCP Slave' with version 'V.1.00, MODBUS\_SERVER.e30'. The 'PLC I/F' is set to 'Ethernet'. The 'IP' is set to 'Local,Port=502(=HMI Port)'. The 'Use UDP (User Datagram Protocol)' checkbox is unchecked. The 'Station no.' is set to '1'. The 'Use broadcast command' checkbox is unchecked. The 'MODBUS TCP/IP Gateway' section has the 'Enable' checkbox checked and the 'Address Mapping Tables...' button visible. 'OK' and 'Cancel' buttons are at the bottom.

Check the *Enable* checkbox in the MODBUS TCP/IP Gateway section, and click the *Address Mapping Tables* button to set up the Modbus Gateway.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	0x <==> LB	0x-1	<==>	Local HMI	LB-0	12400 Bit(s)	Read/Write
2	1x <==> LB	1x-1	<==	Local HMI	LB-0	12400 Bit(s)	Read only
3	3x <==> LW	3x-1	<==	Local HMI	LW-0	9999 Word(s)	Read only
4	4x <==> LW	4x-1	<==>	Local HMI	LW-0	9999 Word(s)	Read/Write
5	3x <==> RW	3x-10000	<==	Local HMI	RW-0	55536 Word(s)	Read only
6	4x <==> RW	4x-10000	<==>	Local HMI	RW-0	55536 Word(s)	Read/Write

\*Note: No cross-table reading/writing, i.e. accessing data from multiple tables in one command.

\* LW-9288 indicates the last communication error :

0 : normal	4 : read-only error
1 : read/write undefined registers	5 : write-only error
2 : out of read/write range	6 : timeout
3 : bad command format	7 : invalid function code

\* Support the following function codes :

0x : 1, 5, 15 (15 used only to set LB)

1x : 2

3x : 4

4x : 3, 6, 16

The default settings are shown. Click the *Delete* button to delete the selected item. Click the *Add* button to add a new item, or the *Setting* button to modify the selected item.

Click the *Add* button to set up a new map.

**Description:** Text that describes this item.

**Device Type:** Specify whether this item is a *Bit* or *Word* device. This selection determines the options available in the *MODBUS Address* and *Mapped PLC Address* fields.

**MODBUS Address:** Specify the starting MODBUS address that will represent the data in the device being mapped. This is the address from which the SCADA system will ask for data, in this case 4x11.

**Mapped PLC Address:** Select the controller to which the Modbus address will be mapped, and specify the starting address in that controller. The Allen Bradley protocol and address N120:0 have been selected above.

**Table Size:** Specify the number of elements to map. Since a Word transfer is being configured, this represents 32 words.<sup>3</sup>

**Conversion, AB -> BA:** When checked, each byte of each word value is swapped. Available only when the *Device Type* is set to *Word*.

**Conversion, ABCD -> CDAB:** When checked, each word of each double-word value is swapped. If both conversion options are checked, *AB -> BA* occurs first. Available only when the *Device Type* is set to *Word*.

**OK:** Close the dialog and add this item to the Modbus Gateway map.

<sup>3</sup> The table size defines the maximum number of elements to be mapped. The HMI will issue one request to the mapped device for each request received from the ModbusTCP master. If a request is received requesting an address outside of the mapped range, the Modbus error register LW9288 will set to 2.

**Address Mapping Table**

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	Values	4x-11	<==>	Allen-Bradley SLC500 and Micro (DF1 Serial-...	Nfn-120000	32 Word(s)	Read/Write

\*Note: No cross-table reading/writing, i.e. accessing data from multiple tables in one command.

\* LW-9288 indicates the last communication error :

0 : normal	4 : read-only error
1 : read/write undefined registers	5 : write-only error
2 : out of read/write range	6 : timeout
3 : bad command format	7 : invalid function code

\* Support the following function codes :

0x : 1, 5, 15 (15 used only to set LB)

1x : 2

3x : 4

4x : 3, 6, 16

Modbus address 4x11 is mapped to AB address N120:0, for a block of 32 words.

Modbus Address	Micrologix Address
4x11	N120:0
4x12	N120:1
4x13	N120:2
4x14	N120:3
4x15	N120:4
4x...	N120:...
4x41	N120:30
4x42	N120:31

The ModbusTCP Master in the SCADA system would be configured to read/write address 4x11, at the IP address assigned to the HMI. The HMI will read/write from the Micrologix at address N120:0, and send the appropriate Modbus response to the ModbusTCP Master.

The following Modbus function codes are supported:

<b>Code</b>	<b>Definition</b>
1	Read Coil Status
2	Read Input Status
3	Read Holding Registers
4	Read Input Registers
5	Force Single Coil
6	Preset Single Register
15	Force Multiple Coils (LB addresses in the HMI <i>only</i> , not compatible with bits in a PLC)
16	Preset Multiple Registers

Modbus communication errors are reported in HMI Local Word 9288:

<b>Value</b>	<b>Error</b>
0	No error
1	Attempt to read/write an undefined register
2	Attempt to read/write an address outside the defined map
3	Bad command format from the Modbus Master
4	Attempt to write to a Read-only register
5	Attempt to read to a Write-only register
6	Timeout from the mapped device
7	Invalid function code received from the Modbus Master

To debug comms from the HMI to the mapped device, place a numeric display object on the HMI referencing an address in the mapped device.