

# Weintek HMI with Built-In Modbus TCP Server



Weintek USA, Inc.  
[www.WeintekUSA.com](http://www.WeintekUSA.com)  
(425) 488-1100

Rev. 23 JUN 2021

## Weintek HMI with Built-In Modbus TCP Server

**Introduction:** This instruction manual discusses how to implement gateway functionalities using Weintek Modbus TCP server. The Gateway provides a mechanism to connect the Modbus TCP Server protocol to storage within a Weintek HMI or in a connected device. The purpose of this document is to show you how to correctly set up the Modbus tables within EasyBuilder Pro.

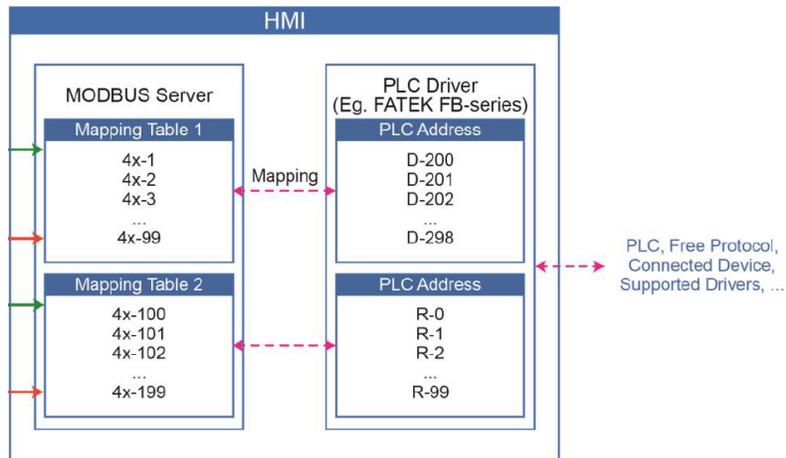
### Contents:

<b>Chapter 1. Configuration of a Modbus TCP Server in EasyBuilder Pro.....</b>	<b>3</b>
<b>Chapter 2. Mapping the HMI's Internal Memory to Modbus Memory.....</b>	<b>10</b>
<b>Chapter 3. Mapping a Tag-Based PLC to Modbus Memory.....</b>	<b>13</b>
<b>Chapter 4. Connecting Modbus RTU Devices to Modbus TCP Networks.....</b>	<b>16</b>

# Weintek HMI with Built-In Modbus TCP Server

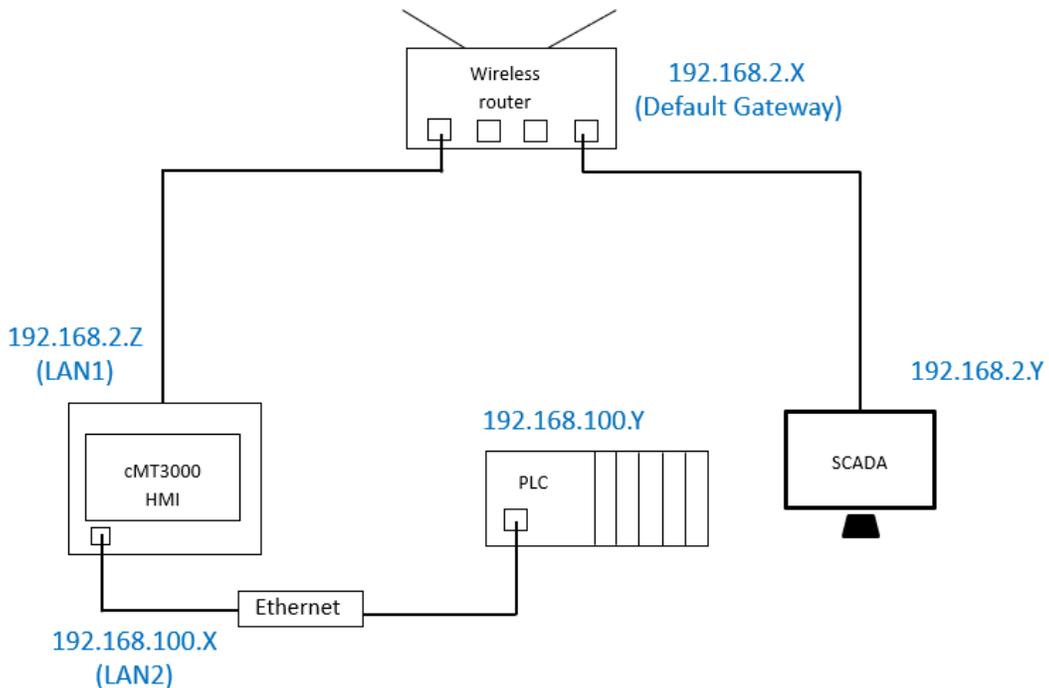
## Modbus TCP Gateway Concept:

In an HMI, the Modbus TCP server contains multiple Mapping tables. In this example, Mapping Table 1 associates MODBUS 4x-1 register with Fatek register D-200, and the number of elements is 99 words. Mapping Table 2 associates MODBUS register 4x-100 with Fatek register R-0, and the number of elements is 100 words.



## Example of Wiring Diagram:

The SCADA system uses the Modbus TCP protocol to query the HMI (Gateway).



# Weintek HMI with Built-In Modbus TCP Server

## Chapter 1. Configuration of a Modbus TCP Server in EasyBuilder Pro

In this example, a cMT-3090 HMI is connecting to a Allen Bradley MicroLogix 1100 PLC.

1. Create a new project in Easybuilder Pro and choose cMT-3090 model.
2. To get the HMI taking to a MicroLogix PLC, go to the [Home] tab» [System Parameters].
3. Add a [Rockwell EtherNet/IP (DF1)] driver to the **Device List**.
4. Click on the [Settings] button and enter the IP address of the PLC.
5. To create the MODBUS gateway, add [MODBUS Server] driver to the Device List as shown below.

Device Settings

Name : MODBUS Server

Device

Location : Local Settings...

\* Select Local for a device connected to this HMI, or Remote for a device connected through another HMI.

Device type : MODBUS Server Settings...  
Device ID : 54, V. 1.00, MODBUS\_SERVER.c30

I/F : Ethernet Open Device Connection Guide...

\* Use LB-12052 to disable MODBUS server (when bit is ON).  
\* Built-in CODESYS can use internal IP (10.255.255.1) to access local MODBUS Server.

IP : Port = 502 Settings...  
 Use UDP (User Datagram Protocol)

Station no. : 1  
 Use broadcast command  
[How to designate the station no. in object's address?..](#)

MODBUS TCP/IP Gateway (Ethernet)  
 Enable Address Mapping Tables...  
 Use station no. given by client

OK Cancel

**[I/F]- Select Ethernet.**

**[IP]- Use the default port number 502.**

**[Station no.]**- The default station number is 1. You can change it if required.

**[Modbus TCP/IP gateway]**- Check **Enable** checkbox. Click on the [Address Mapping Tables] button to configure the Modbus tables.

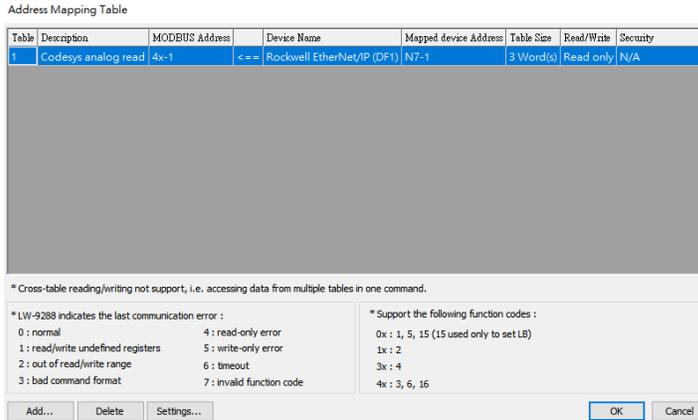
**[Use station no. given by client]**- This option depends on the Modbus TCP client.

Note: UDP won't be available when the **Modbus TCP gateway** option is used.

6. Configure Modbus tables by clicking on the [Address Mapping Tables...] button.

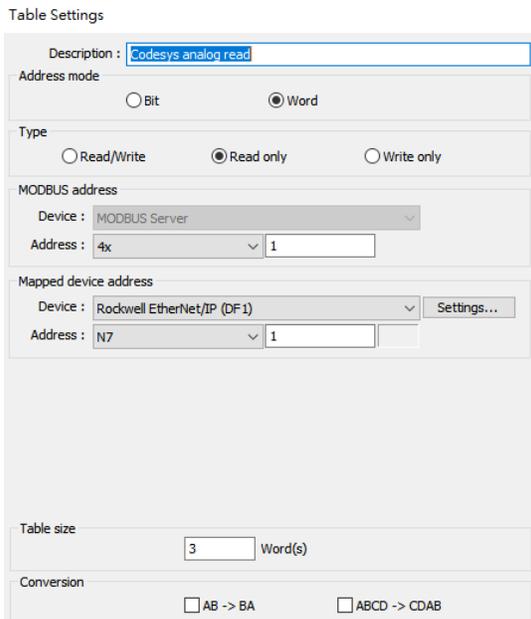
# Weintek HMI with Built-In Modbus TCP Server

## Table Editor

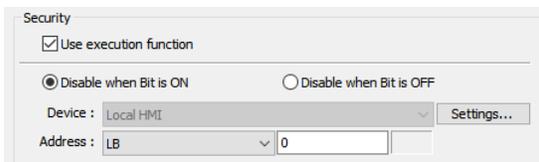


- [Add] button-** Creates a new table.
- [Delete] button-** Removes the selected table.
- [Settings] button-** Modifies the selected table.

## Table Setting



- [Description]-** Enter a comment if needed.
- [Address mode]-** Select a data type.
- [Type]-** Select a mode to access the data in the mapped register. Modbus address 1x and 3x are **Read only**.
- [Modbus address]-** Specify the Modbus function code and starting address. The starting address starts at 1. (**one-based**)
- [Mapped device address]-** Select the connected device and specify the starting address.
- [Table size]-** The number of the bits or registers. Up to 65535 data points are available.
- [Conversion]-** This option is only available when the [Address mode] is set to **Word**.  
**AB->BA** swaps high byte and low byte in each word when checked.  
**ABCD->CDAB** swaps high word and low word in each double-word when checked.



- [Security]-** You can define a Boolean variable to prevent Modbus TCP clients from writing data in this mapping table. This option is only available when **Type** is set to Write only or Read/Write.

## Weintek HMI with Built-In Modbus TCP Server

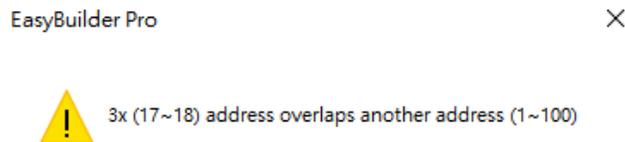
The Modbus table above maps Modbus address 4x-1 to N7:1, for a block of 3 words and read-only.

Mapping Table 1	
4x-1	← N7:1
4x-2	← N7:2
4x-3	← N7:3

A Modbus TCP client can access data using the following Modbus function codes supported in this Modbus TCP server, at the IP address assigned to the HMI.

Modbus Address in EasyBuilder Pro	Modbus Function Code	Descriptions
0x	1	Read Coil Status
	5	Force Single Coil
	15	Force Multiple Coils (LB addresses in the HMI internal memory available only)
1x	2	Read Input Status
3x	4	Read Input Registers
4x	3	Read Holding Registers
	6	Preset Single Register
	16	Preset Multiple Registers

Note: The defined **Modbus address** in a mapping table is not allowed to overlap the Modbus address in another mapping table. The warning message will be displayed as shown below.



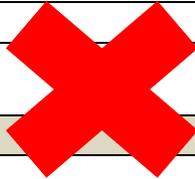
## Weintek HMI with Built-In Modbus TCP Server

For example,

Mapping Table 1	
3x-1	← N7:1
3x-2	← N7:2
3x-3	← N7:3
.....	
3x-100	← N7:100

Mapping Table 2	
3x-17	← F8:1
3x-19	← F8:2



In this demonstration, four Modbus mapping tables are created as shown below.

Address Mapping Table

Table	Description	MODBUS Address		Device Name	Mapped device Address	Table Size	Read/Write	Security
1	Codesys analog read	4x-1	<==	Rockwell EtherNet/IP (DF1)	N7-1	3 Word(s)	Read only	N/A
2	Codesys analog write	4x-17	==>	Rockwell EtherNet/IP (DF1)	N7-100	2 Word(s)	Write only	N/A
3	Codesys digital read	1x-1	<==	Rockwell EtherNet/IP (DF1)	B3-1	1 Bit(s)	Read only	N/A
4	Codesys digital write	0x-1	==>	Rockwell EtherNet/IP (DF1)	B3-10	1 Bit(s)	Write only	N/A

Mapping Table 1	
4x-1	← N7:1
4x-2	← N7:2
4x-3	← N7:3

Mapping Table 2	
4x-17	→ N7:100
4x-18	→ N7:101

## Weintek HMI with Built-In Modbus TCP Server

Mapping Table 3
1x-1 ← B3:1

Mapping Table 4
0x-1 → B3:10

### Note:

1. Some Modbus TCP clients read a group of 16 bits at a time instead of reading a single bit at a time. Bit groups are 0-15, 16-31, 32-47, 48-63, etc. All bits in the group must be available in the Modbus TCP server. Otherwise, errors will result.
2. Data is stored in four different Modbus maps. Each data point of the Coil and Discrete input objects consists of 1 bit. Each data point of the Input register and Holding register consists of 16 bits (= 1 word). The Modbus TCP server uses the extended referencing as shown below. Up to 65535 data points can be created.

Object Type	Access (Read-write)	Address Range
Coil (Bit)	R/W	000001-065535 (0x)
Discrete input (Bit)	R	100001-165535 (1x)
Input register (16-bits)	R	300001-365535 (3x)
Holding register (16-bits)	R/W	400001-465535 (4x)

16-bit data occupies 1 register on the Modbus map.

16-bit data
400001

32-bit data occupies 2 register on the Modbus map

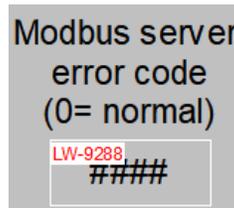
32-bit data	
400001	400002

## Weintek HMI with Built-In Modbus TCP Server

64-bit data occupies 4 register on the Modbus map

64-bit data			
400001	400002	400003	400004

7. You can create a Numeric object on the HMI screen to show the error code if communication fails. The address is defined as LW-9288, and the data type is 16 bit unsigned.

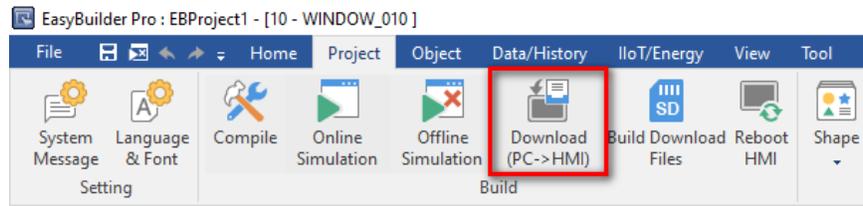


System register LW-9288 (16 bit unsigned) is used to indicate if errors exist during Modbus communication.

Error Code (Value)	Error Name	Descriptions
0	Normal	No error exists
1	Read/Write undefined registers	Reading or writing the register that is not defined in the Address Mapping Table.
2	Out of read/write range	Reading or writing a range of registers that is not within the range defined in a single Address Mapping Table.
3	Bad command format	The command format does not follow MODBUS TCP/IP protocol.
4	Read-only error	Modifying a read-only register.
5	Write-only error	Reading a write-only register.
6	Timeout	HMI cannot get the correct reply from PLC within the specified time range.
7	Invalid function code	Using a function code that is not supported by this Modbus Server.

## Weintek HMI with Built-In Modbus TCP Server

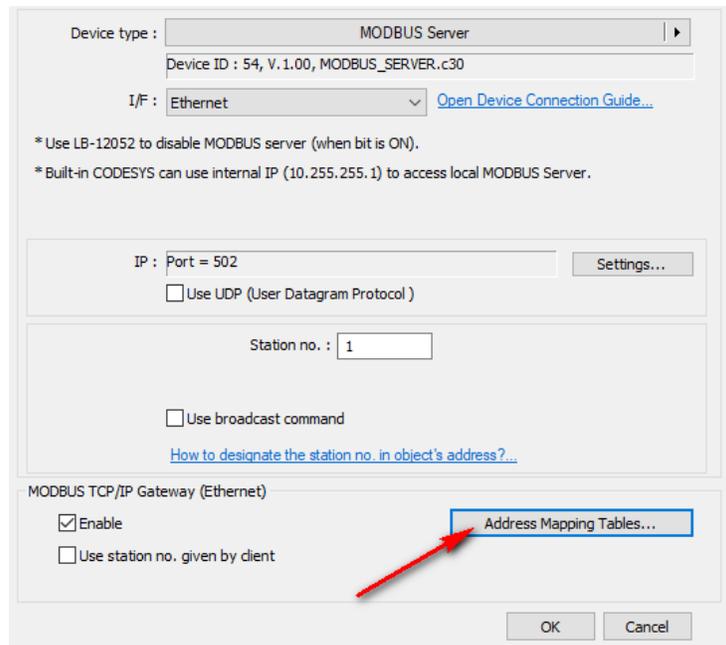
8. Transfer the HMI project to your HMI by clicking the Download button.



# Weintek HMI with Built-In Modbus TCP Server

## Chapter 2. Mapping the HMI's Internal Memory to Modbus Memory

Add a [MODBUS Server] driver to the Device List. Then click on the [Address Mapping Tables...] button.



Below is the default Modbus tables. A Modbus TCP client can access the HMI memory using Modbus TCP protocol.

Address Mapping Table

Table	Description	MODBUS Address		Device Name	Mapped device Address	Table Size	Read/Write
1	0x <==> LB	0x-1	<==>	Local HMI	LB-0	12800	Read/Write
2	1x <==> LB	1x-1	<==	Local HMI	LB-0	12800	Read only
3	3x <==> LW	3x-1	<==	Local HMI	LW-0	9999	Read only
4	4x <==> LW	4x-1	<==>	Local HMI	LW-0	9999	Read/Write
5	3x <==> RW	3x-10000	<==	Local HMI	RW-0	55536	Read only
6	4x <==> RW	4x-10000	<==>	Local HMI	RW-0	55536	Read/Write

## Weintek HMI with Built-In Modbus TCP Server

Some Local Bit (LB) and Local Word (LW) addresses are reserved for System tag of the HMI. LB and LW addresses **9000** and higher must not be used on Modbus tables.

Please change the table size to avoid memory overlapping.

Address Mapping Table

Table	Description	MODBUS Address		Device Name	Mapped device Address	Table Size	Read/Write
1	0x <==> LB	0x-1	<==>	Local HMI	LB-0	8999	Read/Write
2	1x <== LB	1x-1	<==	Local HMI	LB-0	8999	Read only
3	3x <== LW	3x-1	<==	Local HMI	LW-0	8999	Read only
4	4x <==> LW	4x-1	<==>	Local HMI	LW-0	8999	Read/Write
5	3x <== RW	3x-10000	<==	Local HMI	RW-0	55536	Read only
6	4x <==> RW	4x-10000	<==>	Local HMI	RW-0	55536	Read/Write

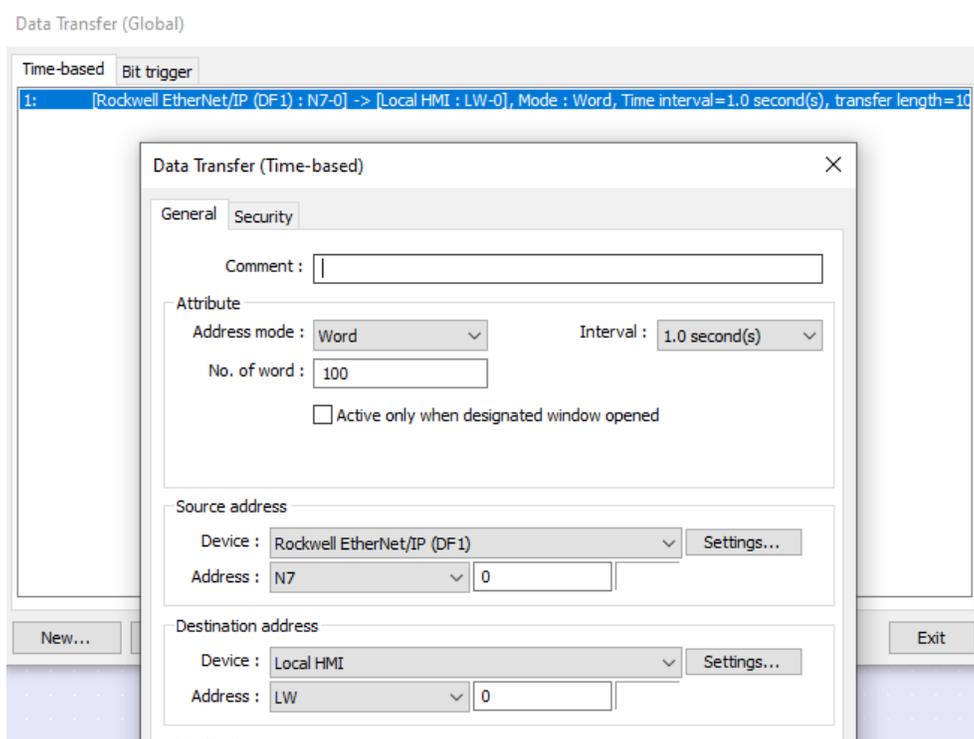
Table Number	Modbus Memory	Mapped To HMI Memory	Access (Read/ Write)
1	0x-1 to 0x-8999	LB-0 to LB-8998	R/W
2	1x-1 to 1x-8999	LB-0 to LB-8998	R
3	3x-1 to 3x-8999	LW-0 to LW-8998	R
4	4x-1 to 4x-8999	LW-0 to LW-8998	R/W
5	3x-10000 to 3x-65535	RW-0 to RW-55535	R
6	4x-10000 to 4x-65535	RW-0 to RW-55535	R/W

You can transfer the device data to the HMI memory for data exchange by the following methods.

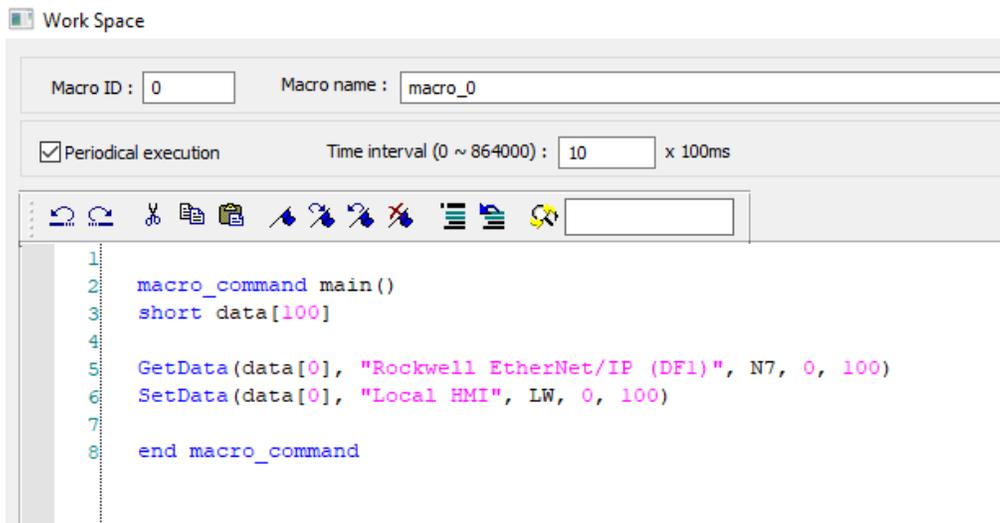
# Weintek HMI with Built-In Modbus TCP Server

These methods transfer 100 integer data points from a AB MicroLogix 1100 PLC to a HMI.

## Method 1: Using a **Data Transfer** object



## Method 2: Using a **Macro**



# Weintek HMI with Built-In Modbus TCP Server

## Chapter 3. Mapping a Tag-Based PLC to Modbus Memory

Recommended version of Easybuilder Pro: v6.04.02 or greater

This new feature within Easybuilder Pro allows you to map PLC tags, which are names you assign to addresses of your device, to the Modbus TCP server directly.

After setting up a driver for your tag-based PLC, add a [MODBUS Server] driver to the Device List. Then click on the [Address Mapping Tables...] button.



Configure the following Modbus tables as shown below.

Address Mapping Table

Table	Description	MODBUS Address		Device Name	Mapped device Address	Table Size
1	4X-1 <> INT	4x-1	<==>	CODESYS V3 (Ethernet)	INT-Application.PLC_PRG.INTarray[0]	20
2	4X-101 <> DINT	4x-101	<==>	CODESYS V3 (Ethernet)	DINT-Application.PLC_PRG.DINTarray[0]	10
3	4X-201 <> REAL	4x-201	<==>	CODESYS V3 (Ethernet)	REAL-Application.PLC_PRG.FLOATarray[0]	20
4	0X-1 <>BOOL array	0x-1	<==	CODESYS V3 (Ethernet)	BitArray-Application.PLC_PRG.BOOLarray[0]	16
5	3X-1 < WORD	3x-1	<==	CODESYS V3 (Ethernet)	WORD-Application.PLC_PRG.WORDdata	1
6	3X-101 < DWORD	3x-101	<==	CODESYS V3 (Ethernet)	DWORD-Application.PLC_PRG.DWORDdata	2
7	3X-201 < LWORD	3x-201	<==	CODESYS V3 (Ethernet)	LWORD-Application.PLC_PRG.LWORDdata	4
8	1X-1 < BOOL	1x-1	<==	CODESYS V3 (Ethernet)	BOOL-Application.PLC_PRG.BOOLdata	1

## Weintek HMI with Built-In Modbus TCP Server

Table Settings

Description: 4x-1 <> INT

Address mode:  Bit  Word

Type:  Read/Write  Read only  Write only

MODBUS address

Device: MODBUS Server

Address: 4x 1

Mapped device address

Device: CODESYS V3 (Ethernet) Settings...

Tag: Application.PLC\_PRG.INTarray[0] INT

Security

Use execution function

Table size: 20 Word(s)

Conversion:  AB -> BA  ABCD -> CDAB

*Details of Modbus Table 1*

The following PLC tags are created in the Codesys PLC program.

BOOLdata: BOOL;

WORDdata: WORD;

DWORDdata: DWORD;

LWORDdata: LWORD;

BOOLarray: ARRAY[0..31] OF BOOL;

INTarray: ARRAY[0..19] OF INT;

DINTarray: ARRAY[0..4] OF DINT;

FLOATarray: ARRAY[0..9] OF REAL;

## Weintek HMI with Built-In Modbus TCP Server

Table Number	Modbus Memory	Mapped To Modbus RTU	Access (Read/Write)	The Number of Data Points	Table Size
1	4x-1 to 4x-20	INTarray	R/W	20	20 words
2	4x-101 to 4x-110	DINTarray	R/W	5	10 words
3	4x-201 to 4x-220	FLOATarray	R/W	10	20 words
4	0x-1 to 0x-16	BOOLarray	R/W	16	16 bits
5	3x-1 to 3x-1	WORDdata	R	1	1 words
6	3x-101 to 3x-102	DWORDdata	R	1	2 words
7	3x-201 to 4x-204	LWORDdata	R	1	4 words
8	1x-1 to 1x-1	BOOLdata	R	1	1 bit

# Weintek HMI with Built-In Modbus TCP Server

## Chapter 4. Connecting Modbus RTU Devices to Modbus TCP Networks

This example demonstrates how to connect a number of Modbus RTU devices on an RS-485 network to a Modbus TCP network.

In this case, three Modbus RTU devices are connected to the HMI via an RS485 2-wire serial connection, so we will need to add a Modbus RTU master to the Device List. The [Device default station no.] can be left by default because this parameter for each device will be specified on the next step.

The screenshot shows a configuration dialog box for a Modbus RTU device. The 'Name' field is set to 'MODBUS RTU'. The 'Device' radio button is selected. The 'Location' is set to 'Local'. The 'Device type' is 'MODBUS RTU, RTU over TCP' with a 'Device ID' of '4, V.4.40, MODBUS\_RTU.e30'. The 'I/F' is 'RS-485 2W'. The 'COM' port is 'COM3 (19200,N,8,1)'. The 'Device default station no.' is '1'. There are checkboxes for 'Default station no. use station no. variable' and 'Use broadcast command'. At the bottom, there are dropdown menus for 'Interval of block pack (words): 5', 'Max. read-command size (words): 120', and 'Max. write-command size (words): 120'. There are also buttons for 'Address Range Limit...', 'Data Conversion...', 'OK', and 'Cancel'.

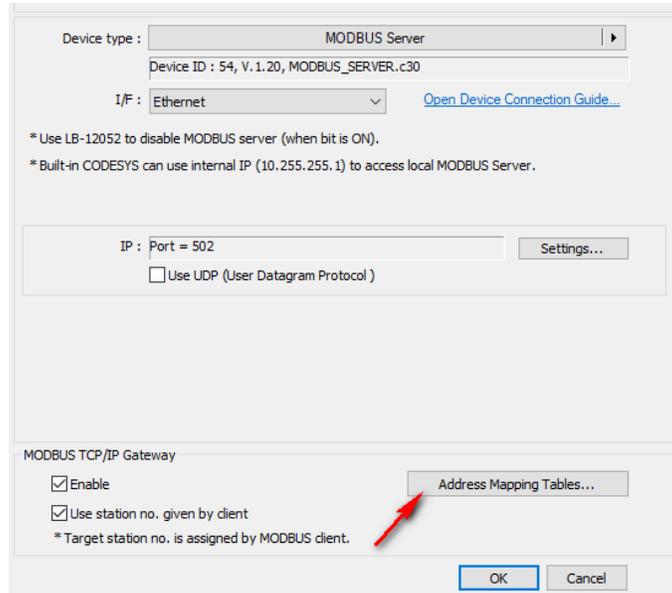
Add a [MODBUS Server] driver to the Device List. There are two methods to configure the Modbus server to meet your application's requirements.

# Weintek HMI with Built-In Modbus TCP Server

## Method 1:

Check the [Use station no. given by client] checkbox. The station number (or "unit identifier") is the slave address of the Modbus RTU device behind a Modbus/TCP to Modbus RTU gateway.

Then click on the [Address Mapping Tables...] button.



Configure the following Modbus tables as shown below. Adjust the "Table Size" to 65535 for each table.

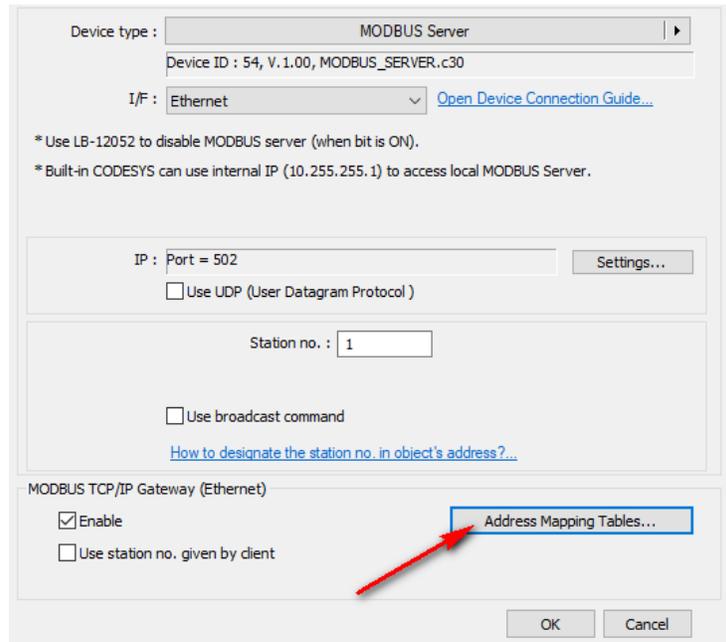
Table	Comment	MODBUS Address		Device Name	Mapped device Address	Table Size	Read/Write	Security
1	0x	0x-1	< == >	MODBUS RTU	0x-1	65535	Read/Write	N/A
2	1x	1x-1	< ==	MODBUS RTU	1x-1	65535	Read only	N/A
3	3x	3x-1	< ==	MODBUS RTU	3x-1	65535	Read only	N/A
4	4x	4x-1	< == >	MODBUS RTU	4x-1	65535	Read/Write	N/A

In this scenario, a Modbus TCP client sends a message to the HMI's Modbus TCP server. The Modbus TCP server passes the message to the internal Modbus RTU master in order to trigger the internal Modbus TCP master to send the message to the external Modbus TCP slave. If the Modbus TCP client reads 3x registers from the starting register 1 to 16 and tells the Modbus server that the target slave is 2, it represents that the Modbus TCP client will retrieve 16 registers data stored in RTU slave ID 2.

# Weintek HMI with Built-In Modbus TCP Server

## Method 2:

Leave [Use station no. given by client] **unchecked**. Then click on the [Address Mapping Tables...] button.



Configure the following Modbus tables as shown below. The address format of Modbus RTU slaves is **ST#Addr**. The **ST** stands for the Modbus slave number, and the **Addr** stands for the Modbus starting register. The **#** sign notation is used to specify the station number followed by the Modbus starting register in the specified station.

Table	Description	MODBUS Address		Device Name	Mapped device Address	Table Size	Read/Write	Security
1	st1	4x-1	<==>	MODBUS RTU	4x-1#1	16	Read/Write	N/A
2	st2	4x-100	<==>	MODBUS RTU	4x-2#1	16	Read/Write	N/A
3	st3	4x-200	<==>	MODBUS RTU	4x-3#1	16	Read/Write	N/A
4	st1_bit	0x-1	<==>	MODBUS RTU	0x-1#1	16	Read/Write	N/A
5	st2_bit	0x-100	<==>	MODBUS RTU	0x-2#1	16	Read/Write	N/A
6	st3_bit	0x-200	<==>	MODBUS RTU	0x-3#1	16	Read/Write	N/A

## **Weintek HMI with Built-In Modbus TCP Server**

In this scenario, a Modbus TCP client sends a message to the HMI's Modbus TCP server. The Modbus TCP server passes the message to the internal Modbus RTU master in order to trigger the internal Modbus TCP master to send the message to the external Modbus TCP slave. If the Modbus TCP client reads 4x registers from the starting register 1 to 16, it represents that the Modbus TCP client will retrieve 16 registers data stored in RTU slave ID 1.

## Weintek HMI with Built-In Modbus TCP Server

“Modbus” is a registered trademark of Schneider Electric.

CODESYS® is a trademark of 3S-Smart Software Solutions GmbH.

Windows is a trademark or a registered trademark of Microsoft Corporation in the United States and/or other countries.

Other company names, product names, or trademarks in this document are the trademarks or registered trademarks of their respective companies.

This document is subject to change without prior notice.



Founded in 1996, WEINTEK LABS is a global-leading HMI manufacturer and is dedicated to the development, design, and manufacturing of practical HMI solutions. WEINTEK LAB's mission is to provide quality, customizable HMI-solutions that meet the needs of all industrial automation requirements while maintaining customer satisfaction by providing “on-demand” customer service. WEINTEK LABS brought their innovative technology to the United States in 2016, WEINTEK USA, INC., to provide quality and expedient solutions to the North American industrial market.

6219 NE 181s Street STE 120  
Kenmore, WA 98028  
425-488-1100