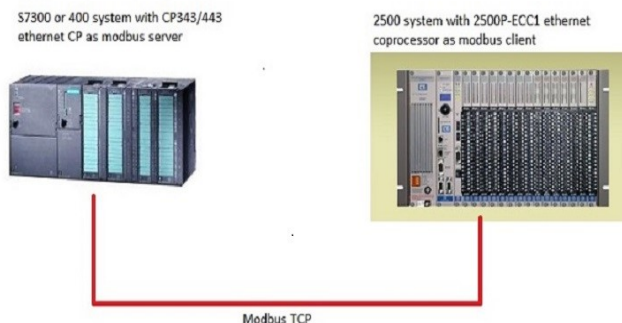


# Application Note:

*Communicating between CTI 2500 Series® PLC and Simatic S7 PLC using 2500P-ECC1 and Modbus/TCP Protocol*

## Application

A CTI 2500 Series® PLC is configured to exchange data with a Siemens S7 PLC, allowing Modbus registers to be written or read from the CTI CPU. The S7 PLC functions as a Server and the CTI PLC as a Client.



## Assumptions

- The CTI PLC uses the 2500P-ECC1 Ethernet Communications Coprocessor card as the network interface. The S7 PLC uses a CP343/443-1 card with a configured connection to the CTI PLC. The Siemens CP cards which are released for this application are 6GK7 343-1EX11-0XE0, 6GK7 443-1EX11-0XE0 or later versions. The Lean versions of the CP343/443 can also be used.
- The S7 PLC is functioning as Server and the CTI PLC as Client.
- The IP address for the CP343-1 is 10.200.64.93
- The IP address for the 2500P-ECC1 is 10.200.64.22
- In this example the CTI PLC will read 10 words from the S7 PLC using Modbus register 1-10 which corresponds to address V29851-V29860 in the CTI PLC and write 10 words to the S7 PLC using Modbus register 11-20 which corresponds to address V29701-V29710 in the CTI PLC.
- A Function block is used in the S7 PLC to map S7 addresses to Modbus registers. It can be used in Step7 V5.1 or higher.
- This Function block is freeware and does not require a software license.

## Description

The configuration has to be carried out on both the S7 side and on the CTI site at the 2500P-ECC1 using the 2500P-ECC1 configuration tool Version 1.4 or higher. No logic is needed in the user program of the CTI PLC.

A function block FB written by Lars Weiß has to be called cyclic within OB1 and in the corresponding "Register DB" S7 memory DBW(16 bit integers) and DBX (bits) are mapped to Modbus registers.

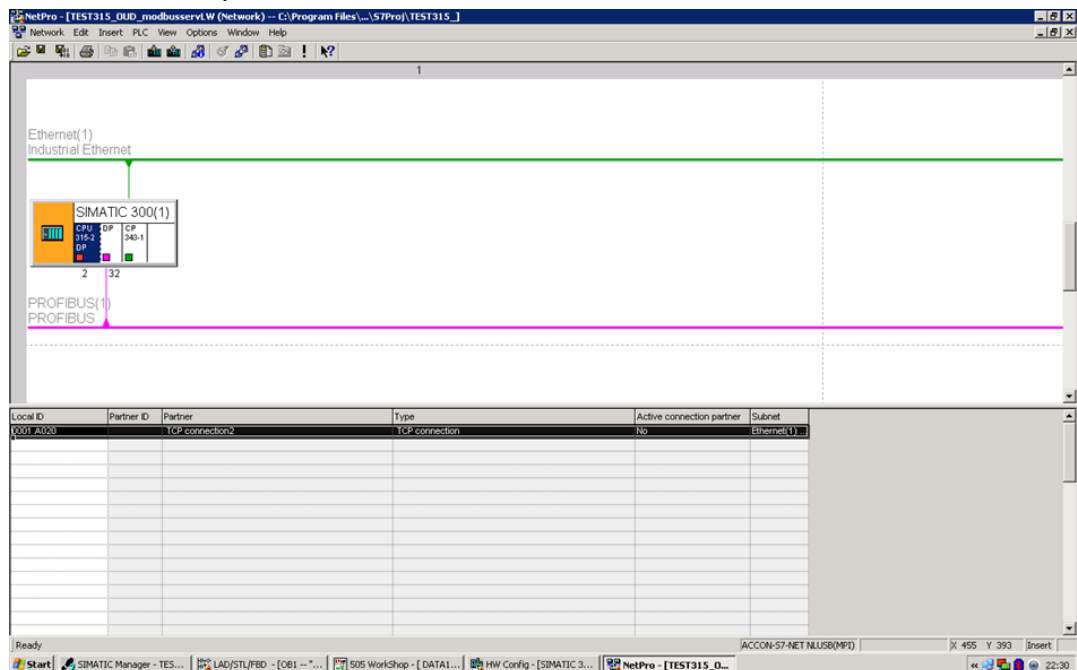
## The FB supports Modbus Functions:

01h,02h	Read n Bits
03h,04h	Read n Words
05h	Write a Bit
06h	Write a Word
10h	Write n Words
0fh	Write n Bits

The structure of this "Register DB" is not important. In this example the structure of this datablock consists of 20 integers and the first integer DBxx.DBW0 corresponds to modbusregister 0, DBxx.DBW2 corresponds to modbusregister 1 etc. But if C-memory coils had to be read or written DBxx.DBX0.0 is C1, DBxx.DBX0.1 is C2 etc. A mix of both integers and coils is also possible.

## S7 configuration

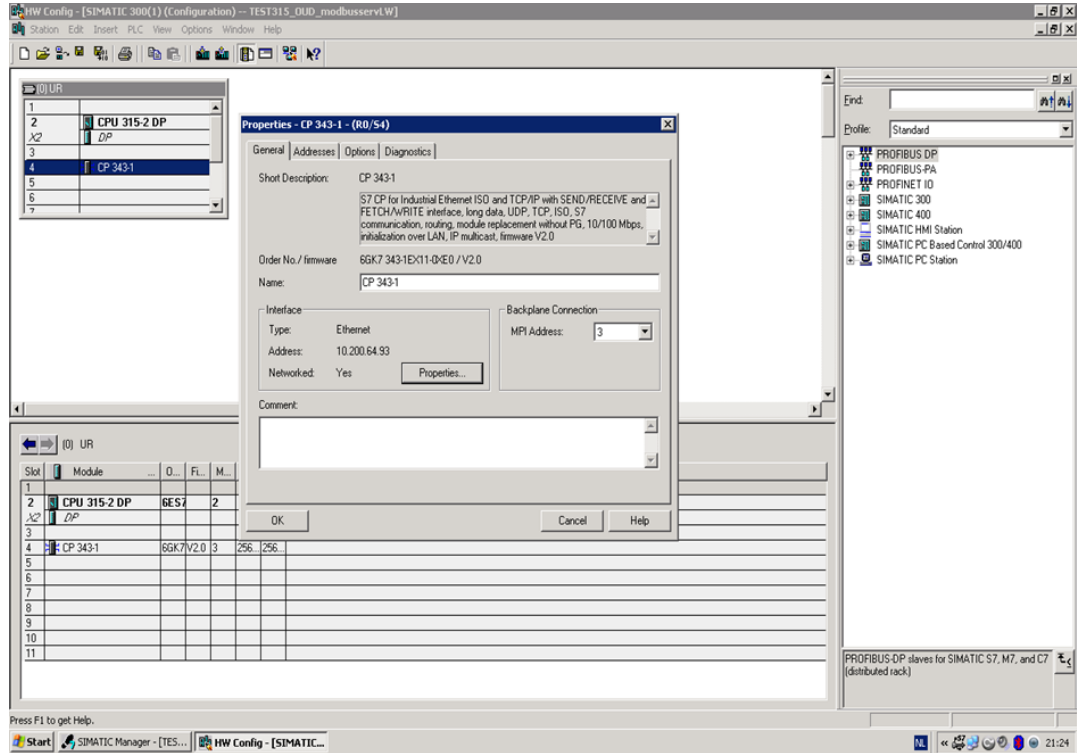
Using Step7, the hardware needs to be configured including an Ethernet network. In NetPro an unspecified connection has to be added as shown below.



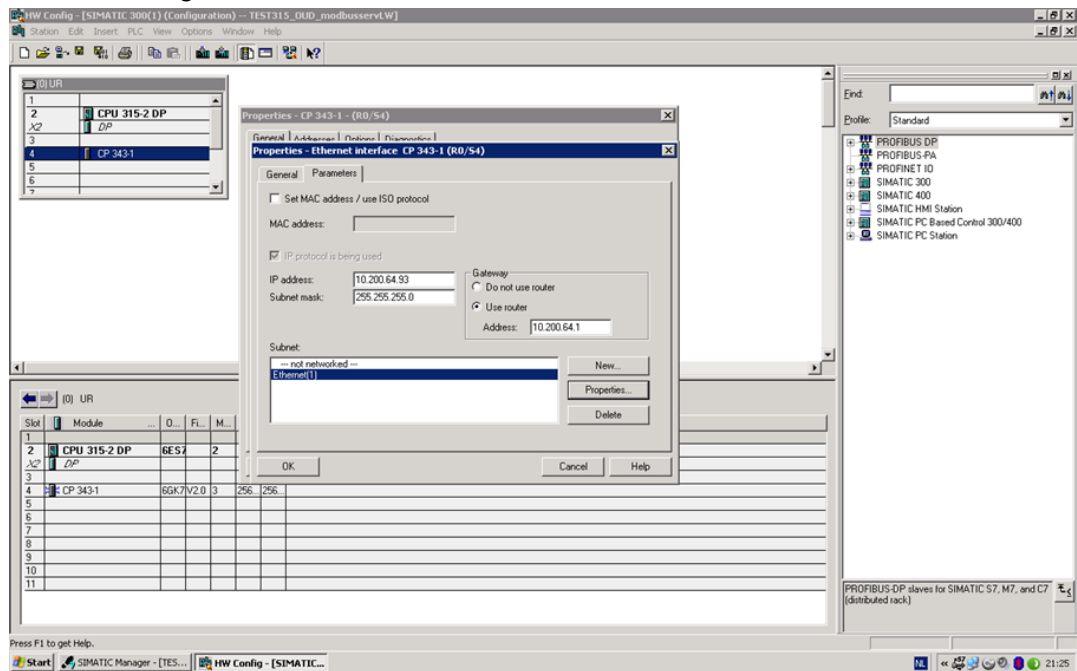
## Configuring the CP-343

The CP343-1 uses the IP address 10.200.64.93 and is connected to the Ethernet network as shown below

### Network configuration 1.



### Network configuration 2.

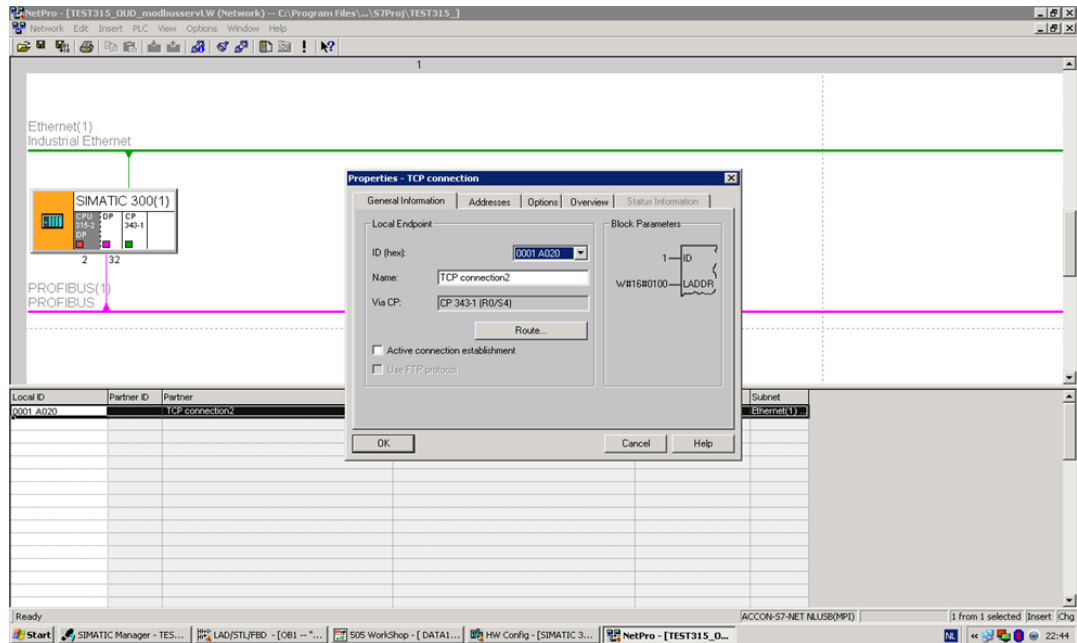


## Configuring the TCP connection for the open Modbus TCP protocol

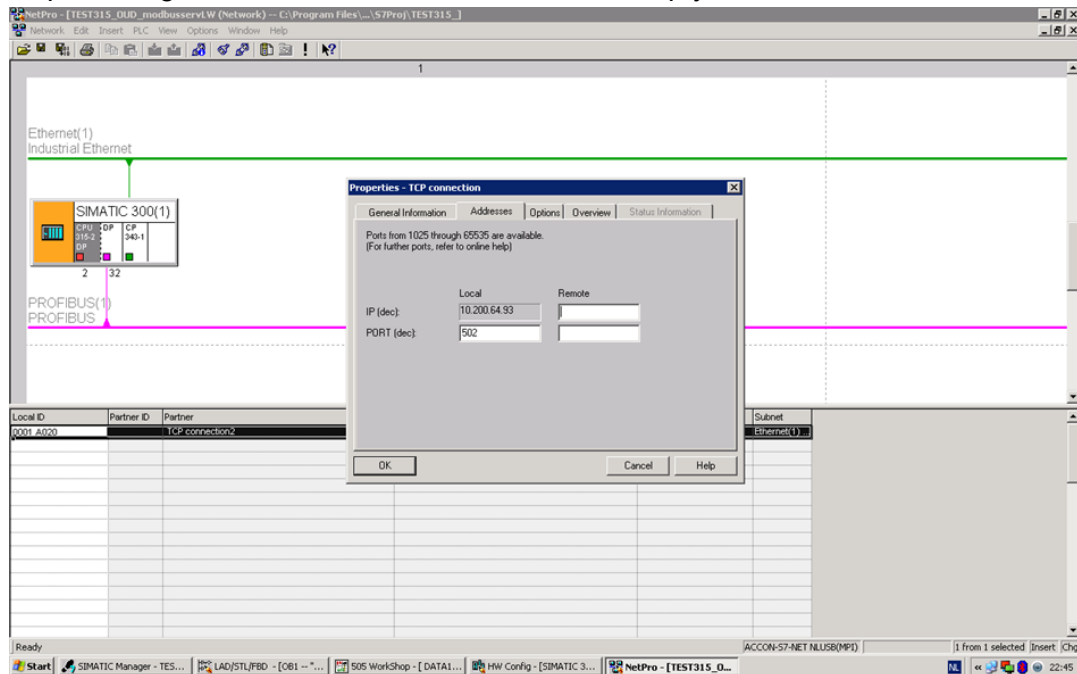
Under General Information the ID number “1” and the CP LADDR “W#16#100” can be read out. The “active connection establishment” feature must **not** be selected because the ECC will establish the connection.

For Modbus TCP server as a standard TCP port 502 is used.

Netpro configuration 1.



Netpro configuration 2. Leave the remote fields empty.



Transfer both Hardware and Netpro connections to the cpu.

## Program with the CP343-1 as server

10 words of data are read from the “Register DB” in the S7 PLC by the ECC and are stored in to V29851 – V29860 of the CTI PLC.

10 words of data V29701-V29710 from the CTI PLC are written by the ECC into the “Register DB” from the S7 PLC.

Program in the S7 plc.

The screenshot displays the SIMATIC Manager interface. On the left, a network editor shows a call to the 'MODBUS\_4' function block. The call parameters are: V29851 (VERB\_ID), 1 (:=1), I16100 (LADDR), and 'modbus\_data' (REGISTER\_DB). The function block is identified as FB10 / DB10, and the data register is DB111.

On the right, the 'Object Properties' window shows a list of objects in the project. A table of these objects is provided below:

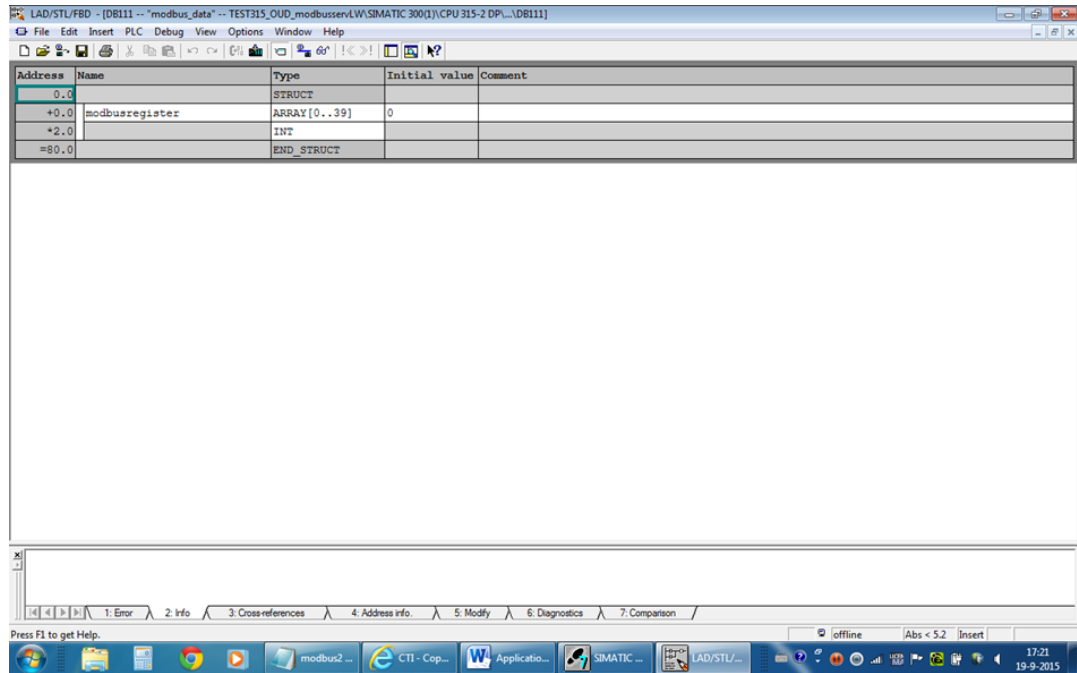
Object name	Symbolic name	Created in language	Size in the work
System data	---	---	---
OB1	CYCL_EXC	STL	
OB2	IO_FLT1	STL	
OB85	OBNL_FLT	STL	
OB86	RACK_FLT	STL	
OB121	PROG_ERR	STL	
FB10	MODBUS_4	STL	
FC5	AG_SEND	STL	
FC6	AG_RECV	STL	
FC8		STL	
DB10	IDB modbus0.4	DB	
DB100	DNTVANGDB	DB	
DB101		DB	
DB111	modbus_data	DB	
modbus_S7-ECC	modbus_S7-ECC		
modbus_S7-T1	modbus_S7-T1		
SFB4	TON	STL	
SFC58	WR_REC	STL	
SFC59	RD_REC	STL	

Red arrows point from the text 'Modbus FB and corresponding Instance DB' to the FB10 object in the tree. A green arrow points from 'Register DB' to the DB111 object. A note at the bottom right states: 'AG\_SEND, AG\_RECV and TON have to be part of your project'.

AG\_SEND and AG\_RECV can be found in “Libraries - Simatic\_NET\_CP - CP-300 - (FC5,FC6)”

TON can be found in “Libraries - Standard library - System Function Blocks - (SFB4)”

The "REGISTER\_DB" you have to create and in this example we use one that has an array of 40 integers.



At this point you have the possibility to test the Modbus communication to the S7 PLC by using a Modbus master simulator software tool like for example Modscan 32 or Modbus Poll. These simulation tools can be downloaded from the internet..

## Configuring the 2500P-ECC1

To read and write data from the S7 plc that we previously configured to emulate a Modbus TCP server we need to configure the ECC as a Modbus TCP client and add a Modbus read and write request.

Add a Modbus client connection to the S7 PLC. Use the IP address of the CP343/443 card to configure the Modbus Client connection.

The screenshot displays the '2500P-ECC1 Configuration' software interface. The main window is titled 'ECC\_TestPLC\_ModbusClient.Ecc'. The 'Protocol Selection' pane on the left shows a tree view with 'Open Modbus Client' selected, and a sub-entry '10.200.64.93: 502/TCP (Communication ECC <=> S7315 2DP)' highlighted. The 'Configure Modbus Client Connection' pane on the right shows the following configuration:

- Description: Communication ECC <=> S7315 2DP
- Modbus Device IP Address: 10.200.64.93 (circled in red)
- Port Number: 502
- Protocol: Open Modbus - TCP
- Delay between requests (ms.): 3

A red arrow points from the IP address field to a note: 'IP address of the CP343 in the S7'.

The 'Tagname Database' table at the bottom is as follows:

Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID
Data_To_S7	V	29701	10	<input type="checkbox"/>	Data From ECC to S7315 with CP343			<input checked="" type="checkbox"/>			0
Data_From_S7	V	29851	10	<input type="checkbox"/>	Data From S7315 with CP343 to ECC			<input checked="" type="checkbox"/>			0

The status bar at the bottom right shows 'LOGIC Module Identifier: ECC\_TestPLC ECC1 IP: 10.200.64.22' and the system tray shows the date '18-9-2015' and time '16:30'.

## Add a Modbus Unit and Read request.

The screenshot shows the 'Configure Modbus Client Request' dialog box in the 2500P-ECC1 Configuration software. The 'MODBUS Request' dropdown is set to '<FC03> Read Holding Registers'. The 'No. of Trials' field is set to 1, which is circled in red with a red arrow pointing to the text 'Beware of Offset!!'. The 'Tagname Database' table is visible at the bottom.

Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID
Data_To_S7	V	29701	10	<input type="checkbox"/>	Data From ECC to S7315 with CP343						0
Data_From_S7	V	29851	10	<input type="checkbox"/>	Data From S7315 with CP343 to ECC			<input checked="" type="checkbox"/>			0

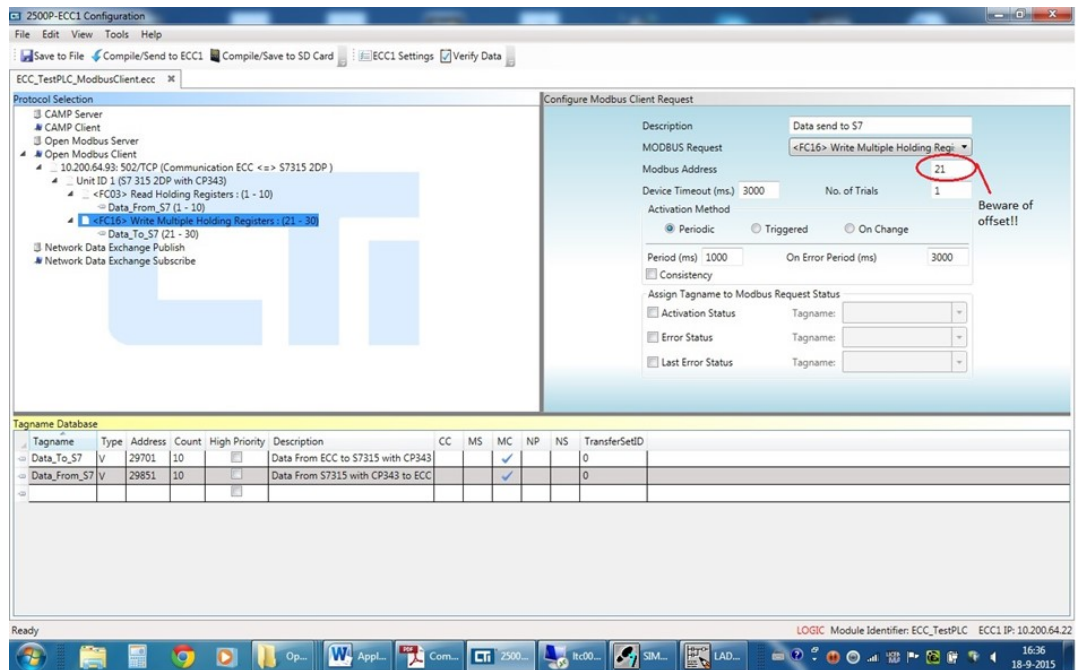
## Add a Tagname out of the "Tagname Database" in this memory area the data read from the S7 PLC will be stored.

The screenshot shows the 'Map Tagname data to Modbus Request' dialog box in the 2500P-ECC1 Configuration software. The 'Tagname' dropdown is set to 'Data\_From\_S7'. The 'Tagname Database' table is visible at the bottom.

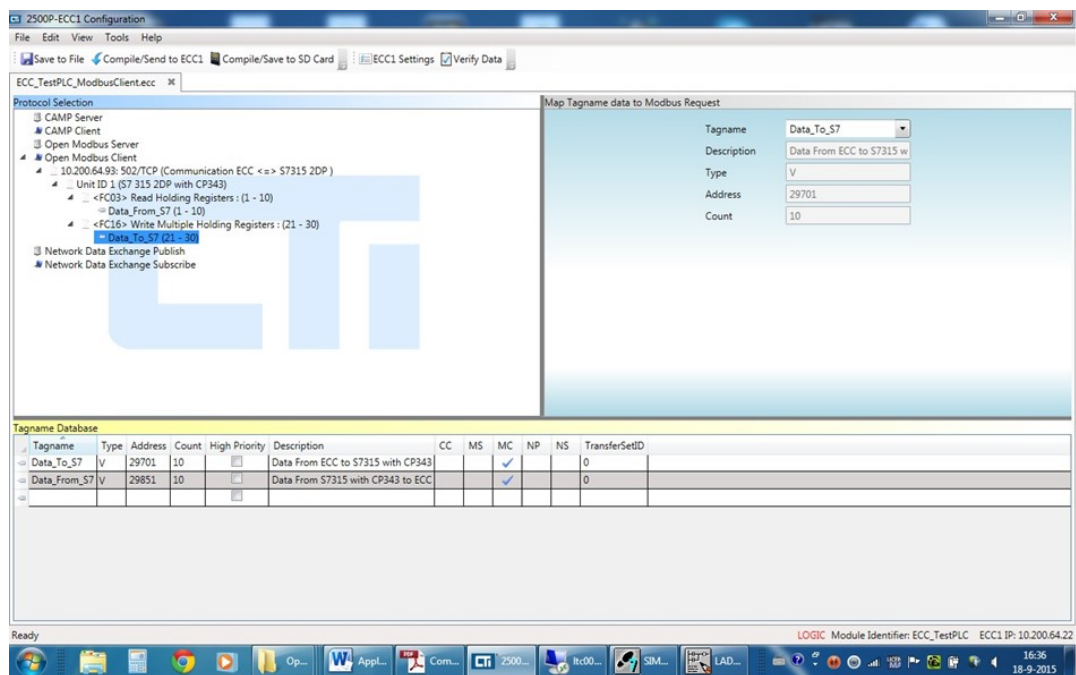
Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID
Data_To_S7	V	29701	10	<input type="checkbox"/>	Data From ECC to S7315 with CP343						0
Data_From_S7	V	29851	10	<input type="checkbox"/>	Data From S7315 with CP343 to ECC			<input checked="" type="checkbox"/>			0



## Add a Modbus write request.



## Add a Tagname out of the "Tagname Database". The contents of this memory area will be written to the S7 PLC.



## Transfer the configuration to the 2500P-ECC1 module

## Test the communication

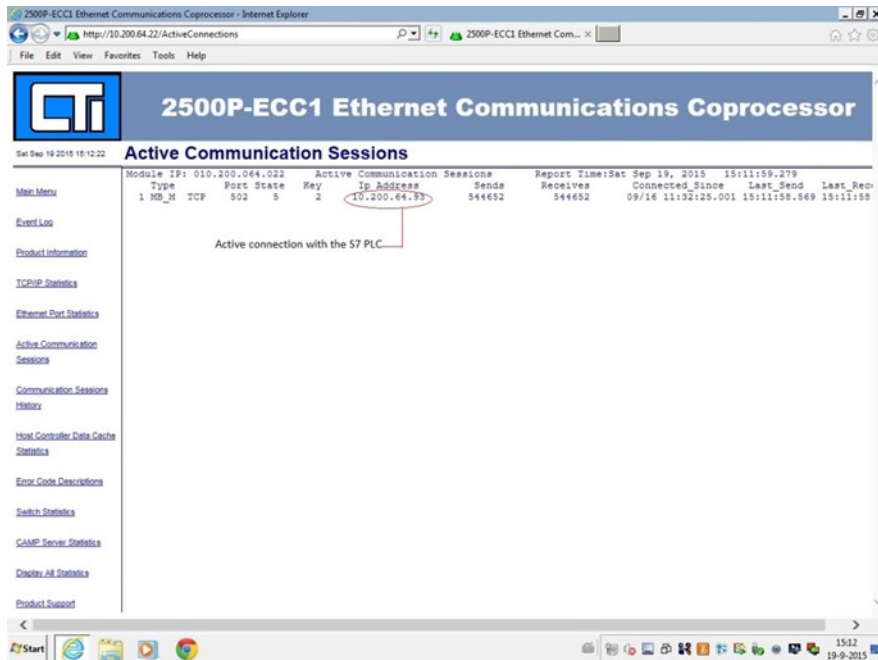
Data window in the CTI PLC with the Configured communication memory area.

Row	Address	Description	Value	Date/Time	Status
1	V29701		11 U16	02.35.33.104 PM 2015-09-19	Success
2	V29702		12 U16	02.35.33.106 PM 2015-09-19	Success
3	V29703		0 U16	02.35.33.106 PM 2015-09-19	Success
4	V29704		0 U16	02.35.33.106 PM 2015-09-19	Success
5	V29705		0 U16	02.35.33.106 PM 2015-09-19	Success
6	V29706		0 U16	02.35.33.106 PM 2015-09-19	Success
7	V29707		0 U16	02.35.33.106 PM 2015-09-19	Success
8	V29708		0 U16	02.35.33.106 PM 2015-09-19	Success
9	V29709		0 U16	02.35.33.106 PM 2015-09-19	Success
10	V29710		99 U16	02.35.33.106 PM 2015-09-19	Success
11					
12					
13					
14	V29851		16884 U16	02.35.33.106 PM 2015-09-19	Success
15	V29852		9427 U16	02.35.33.106 PM 2015-09-19	Success
16	V29853		0 U16	02.35.33.106 PM 2015-09-19	Success
17	V29854		0 U16	02.35.33.106 PM 2015-09-19	Success
18	V29855		0 U16	02.35.33.106 PM 2015-09-19	Success
19	V29856		0 U16	02.35.33.106 PM 2015-09-19	Success
20	V29857		0 U16	02.35.33.106 PM 2015-09-19	Success
21	V29858		0 U16	02.35.33.106 PM 2015-09-19	Success
22	V29859		0 U16	02.35.33.106 PM 2015-09-19	Success
23	V29860		123 U16	02.35.33.106 PM 2015-09-19	Success
24					
25					
26					
27					
28					

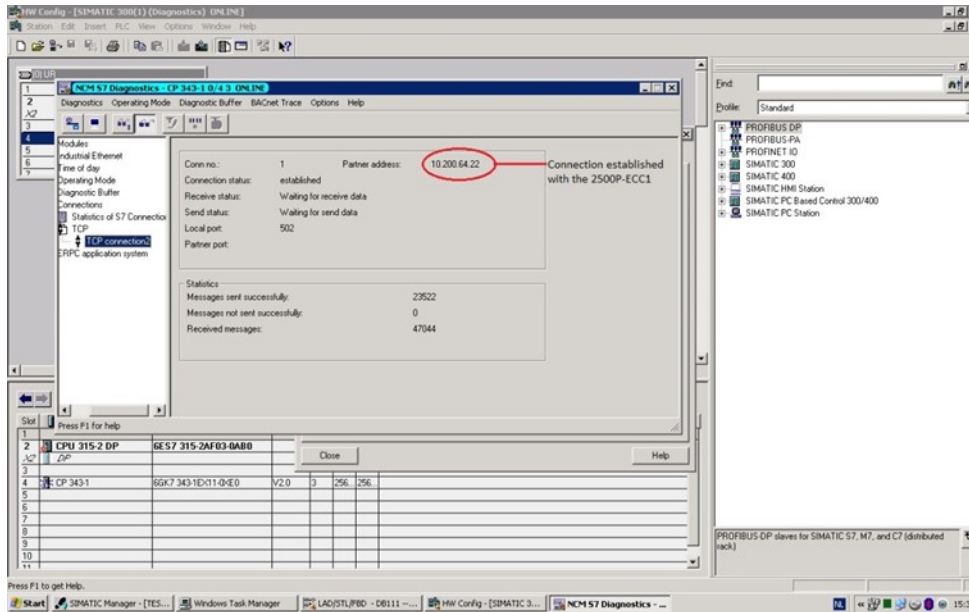
Data window in the S7 PLC with the Configured communication memory area.

Address	Symbol	Display format	Status value	Modify value
0	"modbus_data".modbusregister[0]	DEC	16694	
2	"modbus_data".modbusregister[1]	DEC	9437	
4	"modbus_data".modbusregister[2]	DEC	0	
6	"modbus_data".modbusregister[3]	DEC	0	
8	"modbus_data".modbusregister[4]	DEC	0	
10	"modbus_data".modbusregister[5]	DEC	0	
12	"modbus_data".modbusregister[6]	DEC	0	
14	"modbus_data".modbusregister[7]	DEC	0	
16	"modbus_data".modbusregister[8]	DEC	0	
18	"modbus_data".modbusregister[9]	DEC	0	
20	"modbus_data".modbusregister[20]	DEC	11	
21	"modbus_data".modbusregister[21]	DEC	12	
22	"modbus_data".modbusregister[22]	DEC	0	
23	"modbus_data".modbusregister[23]	DEC	0	
24	"modbus_data".modbusregister[24]	DEC	0	
25	"modbus_data".modbusregister[25]	DEC	0	
26	"modbus_data".modbusregister[26]	DEC	0	
27	"modbus_data".modbusregister[27]	DEC	0	
28	"modbus_data".modbusregister[28]	DEC	0	
29	"modbus_data".modbusregister[29]	DEC	99	

Webpage of the ECC showing the active Modbus client connection to the CP343 in the S7 PLC.



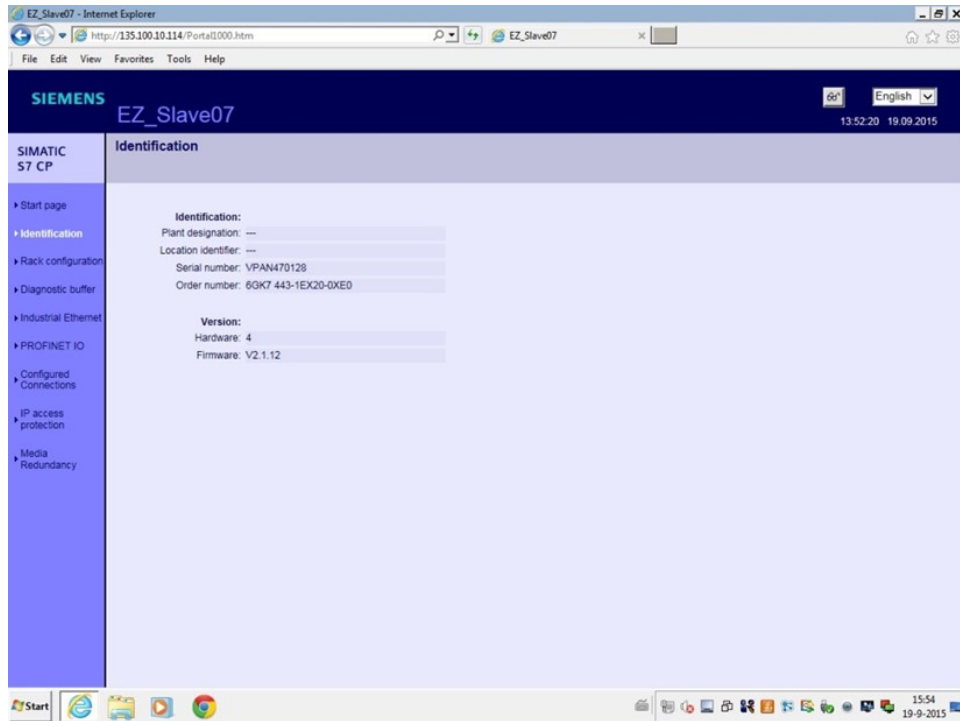
Online diagnostics of the CP343 showing the connection with the ECC.



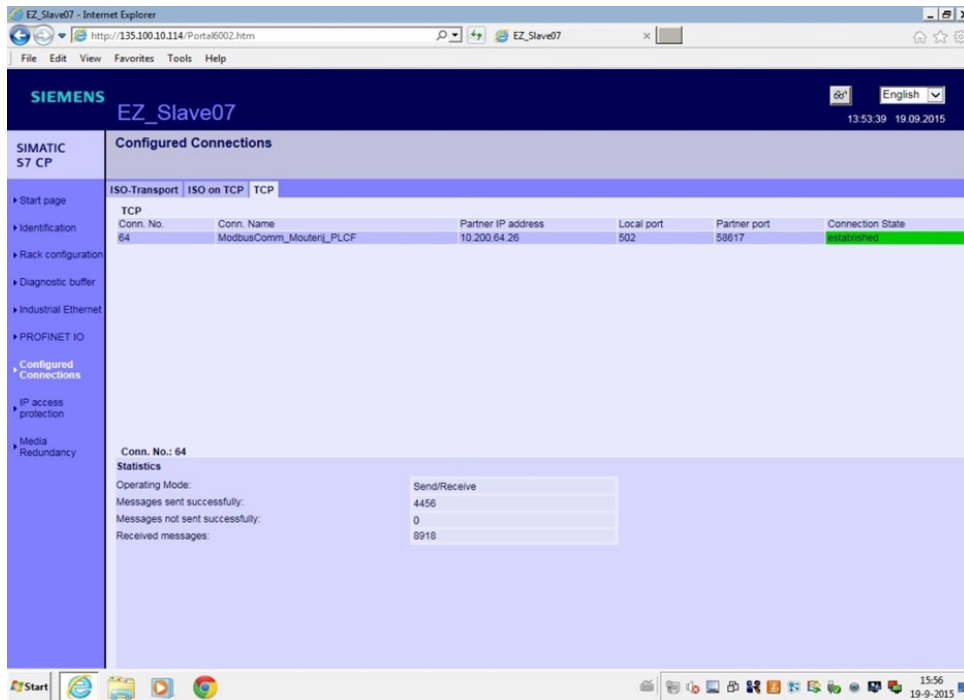
## Example of this communication implemented on the factory floor of a malting plant

The communication has also been tested and running live in a malting plant with 2500P-ECC1 and S7416 with CP443 as shown in the next screenshots. The S7 PLC is situated in the utility area and is on a different VLAN than the ECC1.

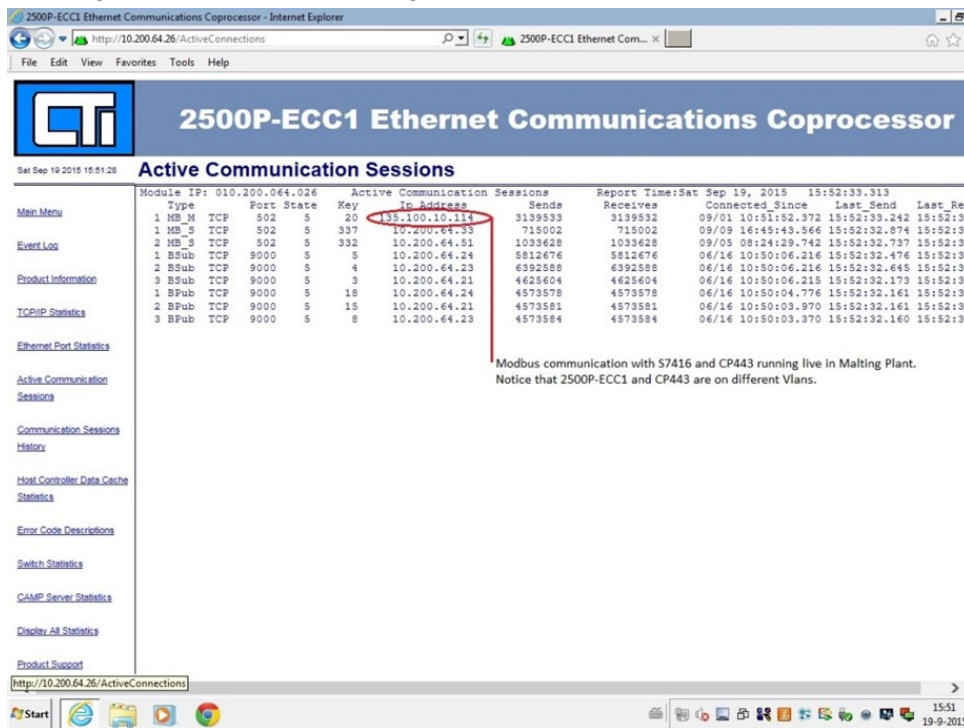
Webpage CP443 with the same modbus configuration.



Webpage CP443 with the same modbus configuration IP from CP443 = 135.100.10.114, 2500P-ECC1 = 10.200.64.26



Webpage 2500P-ECC1 showing Modbus communication with the CP443.



**Control Technology Inc.**

5734 Middlebrook Pike, Knoxville, TN 37921-5962  
 Phone: +1.865.584.0440 Fax: +1.865.584.5720  
 www.controltechnology.com