Application Note

FESTO

Commissioning of CTEU-EP in Modbus/TCP mode

This application node includes examples how to establish a Modbus/TCP communication with the CTEU-EP node.

CTEU-EP

100142

Title .....Commissioning of CTEU-EP in Modbus/TCP mode
Version 1.10
Document no. 100142
Original .....en
Author .....Festo
Last saved 28.03.2017

Copyright Notice

Legal Notice

Internet: http://www.festo.com
E-Mail: service_international@festo.com

Table of contents

1 Components/Software/ IP address used

Type/Name	Version Software / Firmware	IP address / IP Netmask
CTEU-EP	2.3.0+55c503b26bc9.20160721.9438	192.168.2.1 / 255.255.0.0
VAEM-L1-S-8-PT Interface	REV 07	--
SPAU-..-LK-..	--	--
Laptop	--	192.168.2.69 / 255.255.0.0
Codesys	V3.5 SP7 Patch 2 pbf	--

Table 1.1: 1 Components/Software used

1.1 Recommended documents as reference

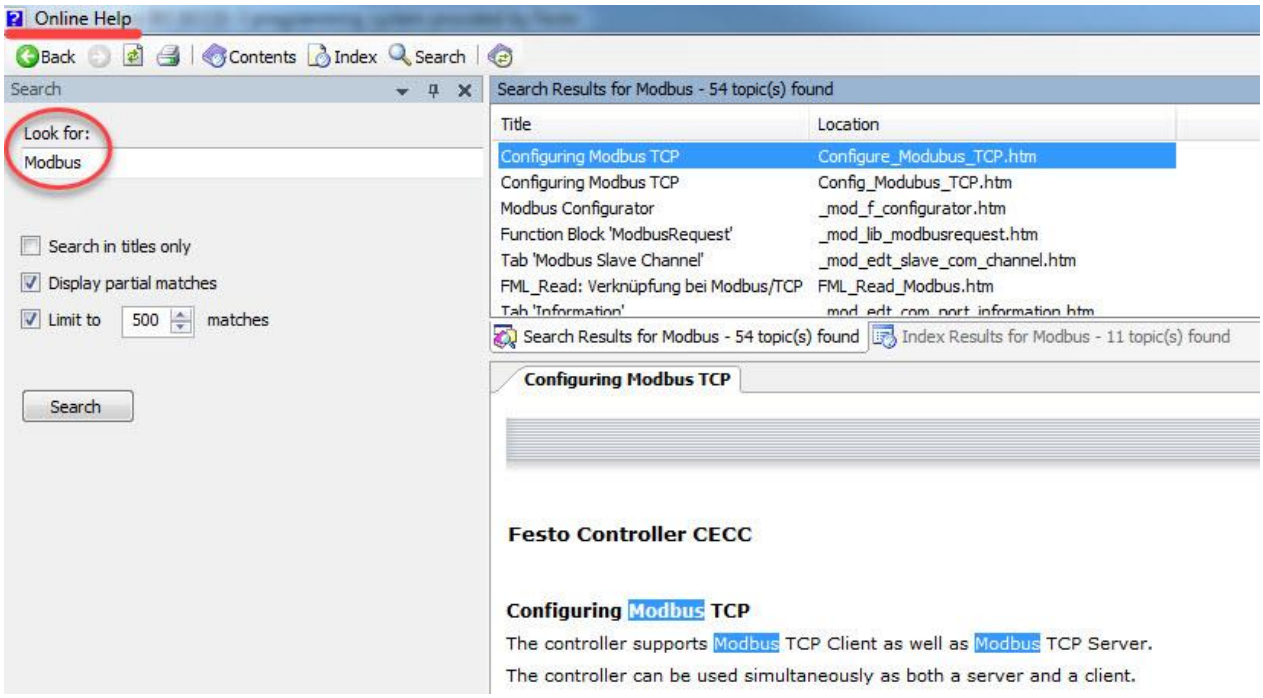
- CTEU-EP manual:
https://www.festo.com/net/SupportPortal/Files/429678/CTEU-EP_2015-10_8049309z6.pdf

- Modbus Application Protocol Specification V1.1.b3:
http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

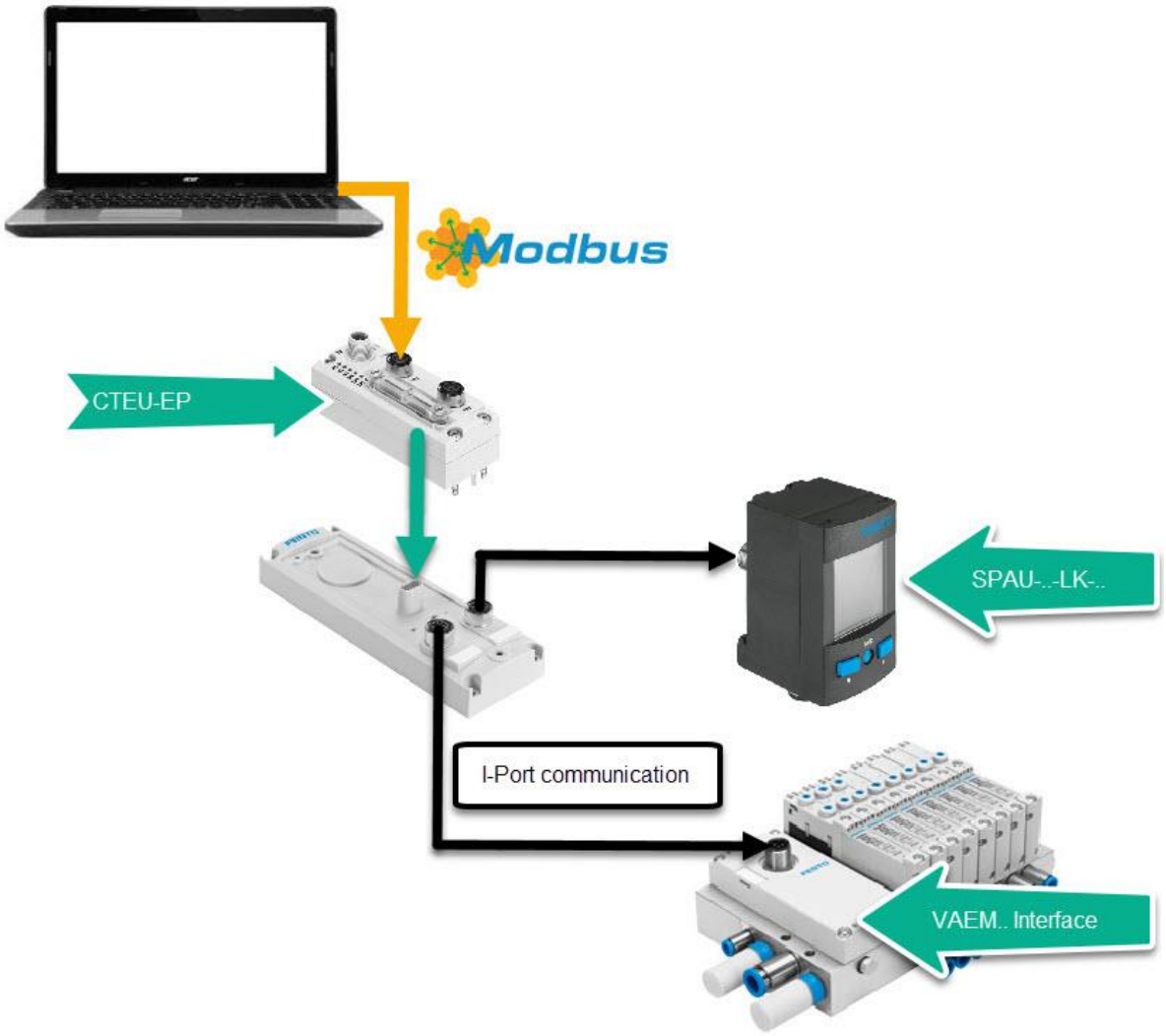
- Modbus Messaging Implementation Guide V1.0b:
http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

- IO-Link Interface and System specification V1.1.2:
http://www.io-link.com/share/Downloads/Spec-Interface/IOL-Interface-Spec_10002_V112_Jul13.pdf

- Codesys Online Help:



1.2 Topology



Note
Festo offers M12-RJ45 and M12-M12 cable for the Ethernet communication:

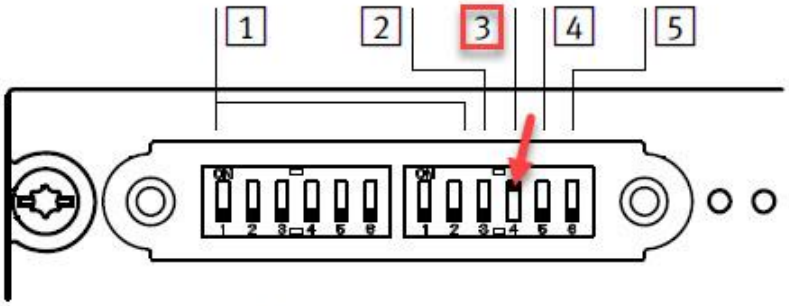
Type code	Part number	Description
NEBC-D12G4-ES-0.5-S-D12G4-ET	8040446	M12-M12 -› 0,5m
NEBC-D12G4-ES-1-S-D12G4-ET	8040447	M12-M12 -› 1m
NEBC-D12G4-ES-3-S-D12G4-ET	8040448	M12-M12 -› 3m
NEBC-D12G4-ES-5-S-D12G4-ET	8040449	M12-M12 -› 5m
NEBC-D12G4-ES-10-S-D12G4-ET	8045450	M12-M12 -› 10m
NEBC-D12G4-ES-1-S-R3G4-ET	8045451	M12-RJ45 -› 1m
NEBC-D12G4-ES-3-S-R3G4-ET	8045452	M12-RJ45 -› 3m
NEBC-D12G4-ES-5-S-R3G4-ET	8045453	M12-RJ45 -› 5m
NEBC-D12G4-ES-10-S-R3G4-ET	8040454	M12-RJ45 -› 10m

Check the Festo NEBU configurator (pn: 539052) too. It includes a lot of cable variants for e.g. IO-Link applications.
Example: NEBU-M12G5-K-0.5-M12G5 would be an IO-Link M12, 5 pin cable with a length of 0.5m

1.3 CTEU-EP DIL settings for Modbus/TCP communication

5.2 Setting the DIL switch

1. Setting IP address ➔ 5.3.
2. Perform additional settings ➔ Table, item [3] ... [5].



Item	DIL switch [1]	Function	
		ON	OFF
DIL switch 1:			
[1]	1 ... 6: Host ID of the IP address, bit 0 ... 5 [2]		
DIL switch 2:			
[1]	1 ... 2: Host ID of the IP address bit 6 and 7 [2]		
[2]	3: Reserved		
[3]	4: Network protocol	Modbus TCP	EtherNet/IP [3]
[4]	5: Status bytes ➔ 5.5 ➔ Table "Connection Parameters"	Status information in the input image	No status information in the input image [3]
[5]	6: Behaviour in case of communication errors and for control in the Idle-Mode	All outputs retain their last switching status	All outputs are reset [3]

1) Switch setting "ON" = ON, switch is to the left or on top
switch setting "OFF" = OFF, switch is to the right or underneath
2) Binary coding in ascending order from left to right 2^0, 2^1, 2^2, ...
3) Factory setting

1.4 Usable Modbus/TCP function codes and address at CTEU-EP

At Modbus /TCP following important function codes are defined within the specification:

				Function Codes code	Sub code	(hex)
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02
		Internal Bits Or Physical coils	Read Coils	01		01
			Write Single Coil	05		05
			Write Multiple Coils	15		0F
	16 bits access	Physical Input Registers	Read Input Register	04		04
		Internal Registers Or Physical Output Registers	Read Holding Registers	03		03
			Write Single Register	06		06
			Write Multiple Registers	16		10
			Read/Write Multiple Registers	23		17
			Mask Write Register	22		16

➔ **Note**
Physical means “the real process data” of a device. Internal registers include values like parameters, diagnostic etc. of a device

A) The CTEU-EP can handle maximum 64 Byte I/O data.
The usable function code(s) to access these process data are:

Function code	Modbus address range	Short description
0x01 (Hex) / 1 (Dec)	0..511 (Dec)	**Read** Bit access e.g. to check the status of a single coil
0x05 (Hex) / 5 (Dec)	0..511 (Dec)	**Write** Bit access e.g. to switch on a single coil
0x04 (Hex) / 4 (Dec)	0..31 (Dec)	**Read** 16-Bit Input register access e.g. to read the 2 Byte process data input value of a SPAU-..-LK-- sensor
0x06 (Hex) / 6 (Dec)	0..31 (Dec)	**Write** single 16-Bit register access e.g. to write the 2 Byte output data of a VPPM-..-LK pressure regulator
0x10 (Hex) / 16 (Dec)	0..31 (Dec)	**Write** multiple 16-Bit register access e.g. to write the 8 / 16 Byte output data of a CMMO-ST..-LKP motor controller
0x17(Hex) / 23 (Dec)	0..31 (Dec)	**Write / Read** multiple 16-Bit register access e.g. to write 8 / 16 Byte data of a CMMO-ST..-LK with one command and read the **output feedback**

➔ **Note**
In the next CTEU-EP release (Rev4 / FW 2.4.0) it will be possible to read the input process data from Modbus / TCP address 200 with the function codes for internal 16 Bit Registers.

B) The Modbus/TCP Idle timeout is default 100ms. If no valid Modbus/TCP frame is received during this timeout the connection gets closed. If no connection is active to CTEU-EP anymore, outputs are marked invalid (reset to zero or hold last state - depend on DIL switch settings.)

- Idle timeout is deactivated by setting it to 0.
- Idle timeout is active in range of 1..4294967296 ms.

Function code (s)	Modbus address	Short description
0x10 (Hex) / 16 (Dec) 0x17(Hex) /23 (Dec) 0x03 (Hex) / 3 (Dec)	40 (Dec)	Read / Write 16-Bit register access for Idle timeout LSW (Least significant word)
	41 (Dec)	Read / Write 16-Bit register access for Idle Timeout MSW (Most significant word)

C) ISDU (Indexed Service Data Unit) access to the connected I-Port device parameters

Function code (s)	Modbus address	Short description
0x06 (Hex) / 6 (Dec) 0x10 (Hex) / 16 (Dec) 0x17(Hex) / 23 (Dec) 0x03 (Hex) / 3 (Dec)	60 (Dec)	Read attribute = 50 (Dec) Write attribute = 51 (Dec)
	61 (Dec)	Module definition -› CTEU-EP = 0 (Dec)
	62 (Dec)	I-Port X1 = 0 (Dec) I-Port X2 = 1 (Dec)-› Only in combination with CAPC-F1-E-M12 usable
	63 (Dec)	ISDU Index -› See connected device manual / Contact Festo support
	64 (Dec)	ISDU Subindex -› See connected device manual / Contact Festo support
	65 (Dec)	Length of data in Bytes -› For Read application always 0
	66..185 (Dec)	Data -› max. 238 Byte

Note

1. Transfer is initiated by writing command (read / write), all other values has to be correctly set up before. After the command is written check input register (ISDU Status).
2. The data value (address 65…185) have a byte swap in the 16 Bit register, because of the IO-Link Big India format. In the next CTEU-EP release (Rev4 / FW 2.4.0) this is changed.

D) Direct access to the first 8 Byte I-Port **device parameter (ISDU Index 0x43h)**

Function code (s)	Modbus address	Short description
0x06 (Hex) / 6 (Dec) 0x10 (Hex) / 16 (Dec) 0x17(Hex) /23 (Dec) 0x03 (Hex) / 3 (Dec)	50..53 (Dec)	Connected device parameter on I-Port X1
	54..57 (Dec)	Connected device parameter on I-Port X2

E) Read Diagnostic and status information of the Input register(s) via function code 0x04 (Hex)

Function code (s)	Modbus address	Short description
0x04 (Hex) / 4 (Dec)	40 (Dec)	Current diagnosis on I-Port X1 ➔ See IO-Link spec. Annex D
	41 (Dec)	Current diagnosis on I-Port X2 ➔ See IO-Link spec. Annex D
	60 (Dec)	ISDU Status -› 0 = OK ; 254 = Busy ; 255 = Error

Informations

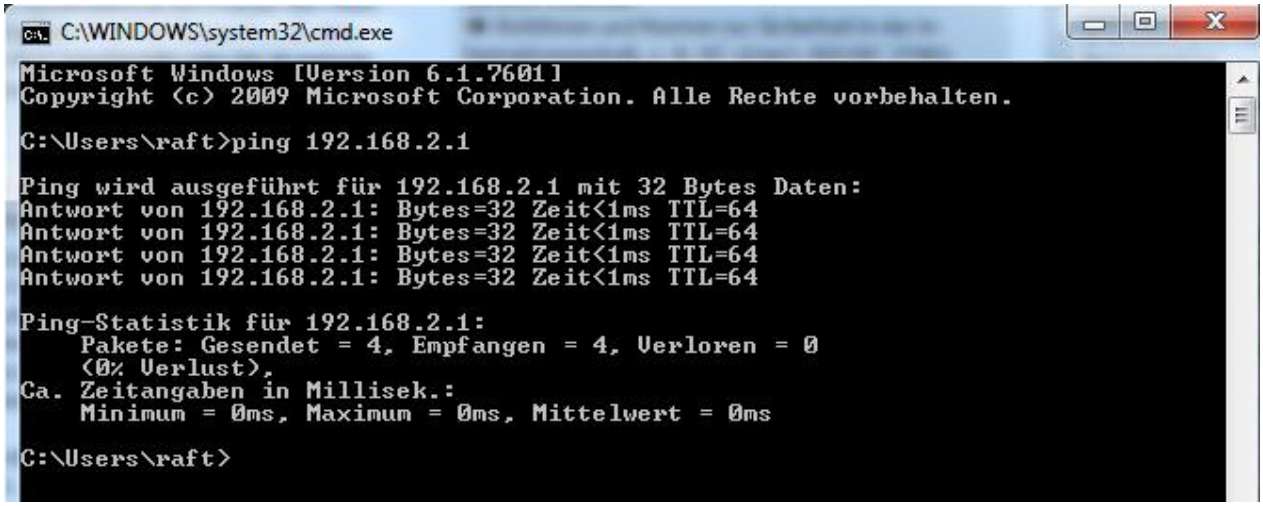
The used internal vendor error codes are:

Code 1	Code 2	Description
FF (Hex)	50 (Hex)	Device lost
FF (Hex)	51 (Hex)	Device reconnected
FF (Hex)	52 (Hex)	Device configuration failed
FF (Hex)	53 (Hex)	A device is attached to a disabled port or attached to a port that was unused at start-up
FF (Hex)	54 (Hex)	device is attached to a disabled port or when attached to a port that was unused at start-up
FF (Hex)	55 (Hex)	device is not attached to a port that is configured to EXPECT_DEVICE
FF (Hex)	56 (Hex)	Event code in tcm when input data size is greater than the configured data size
FF (Hex)	70 (Hex)	Event code when entering fatal error

2 Commissioning in Codesys

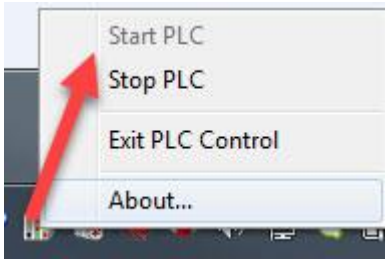
2.1 Key requirements

A) The IP address of the CTEU-EP is set and you can ping the node



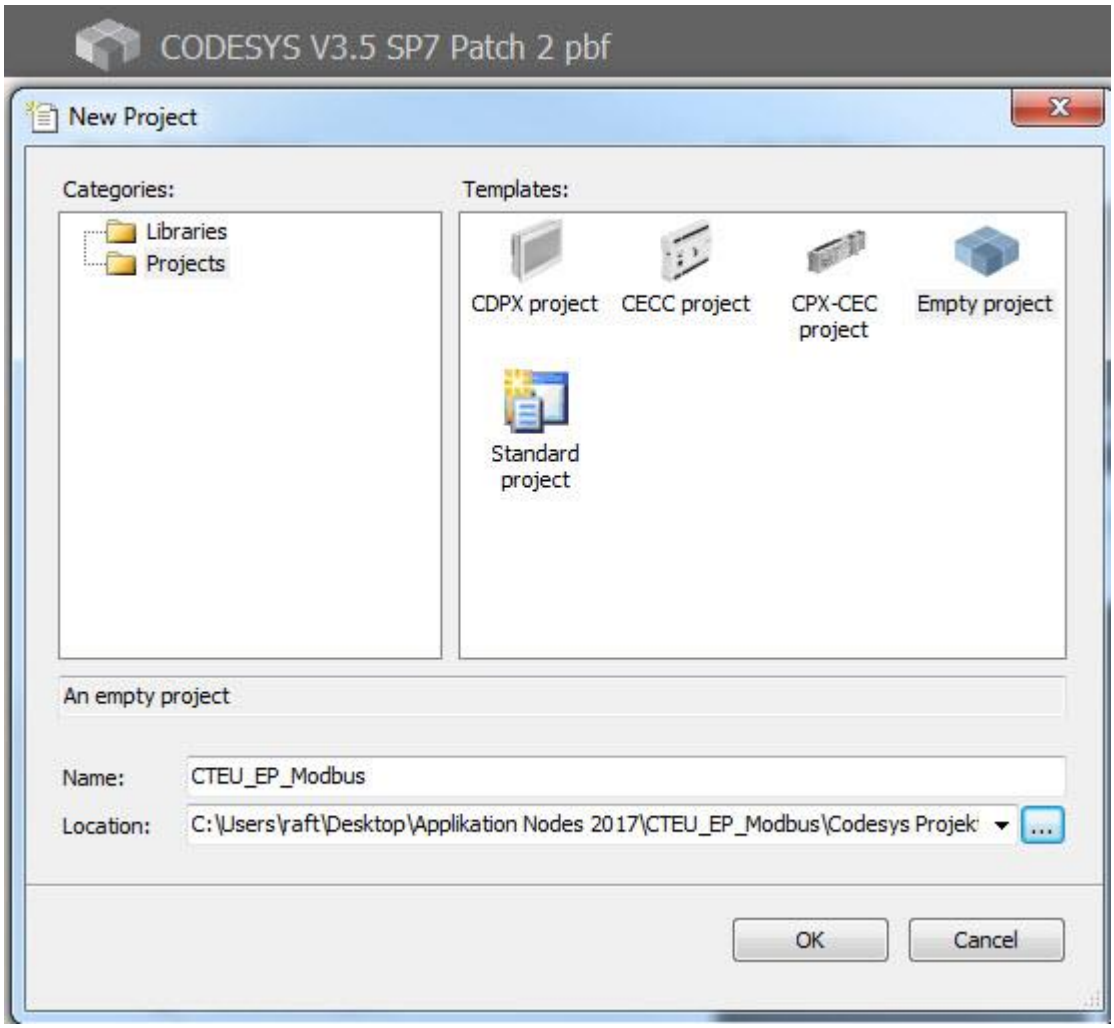
B) The LED status of CTEU-EP in combination with the connected I-Port devices look like

C) The Codesys Control Win V3 runtime is started

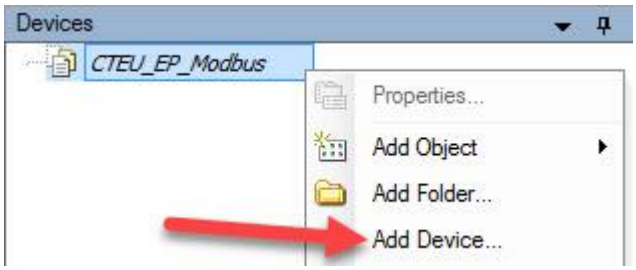


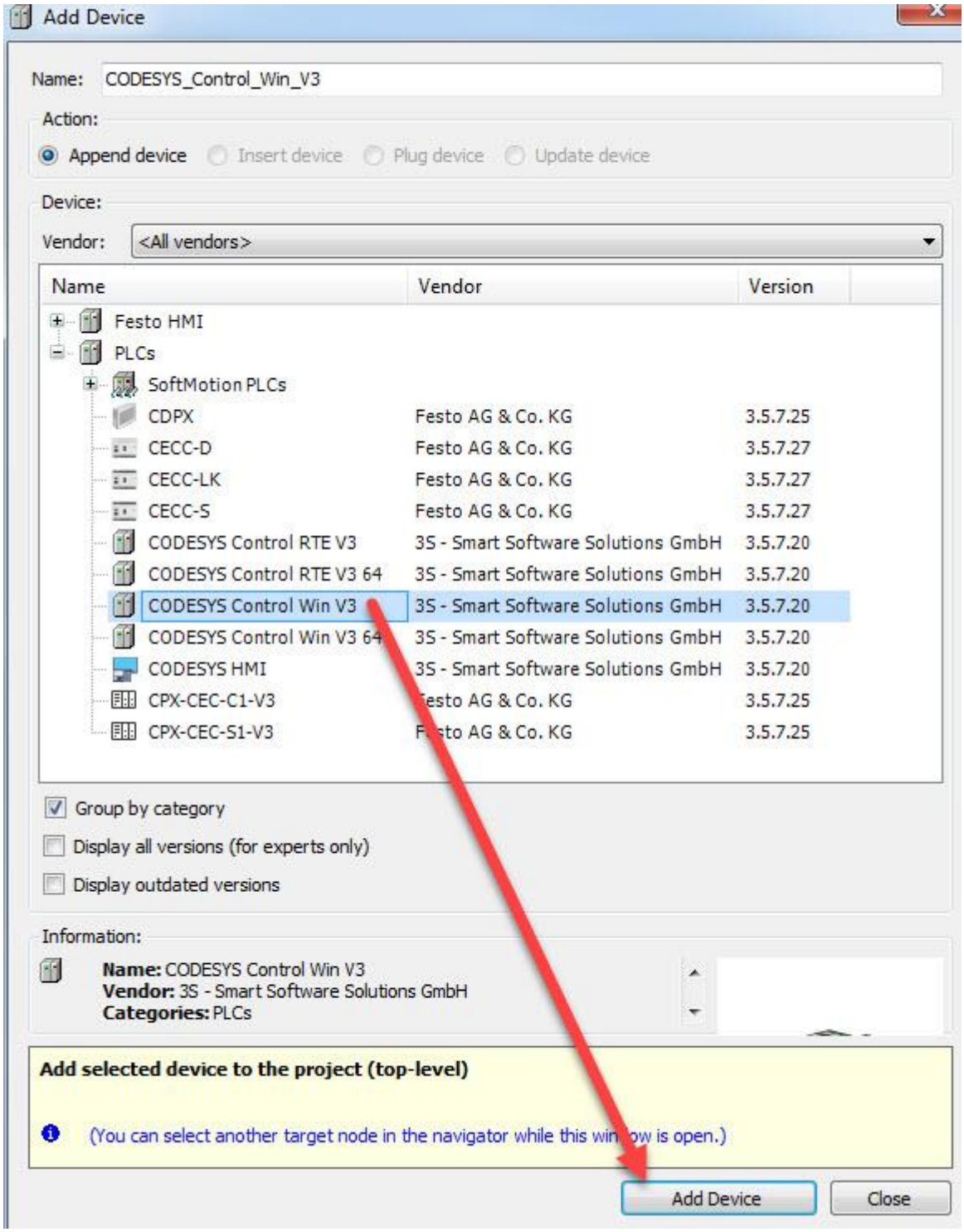
2.2 Create a project in Codesys V3

A) Create a new project

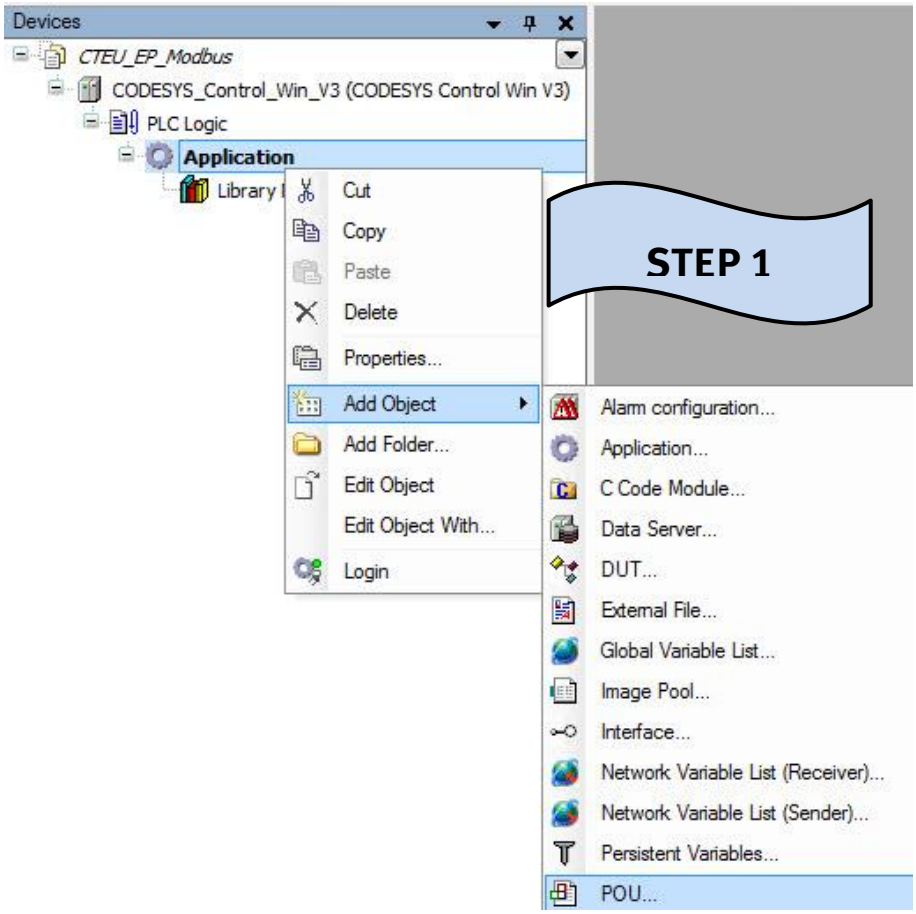


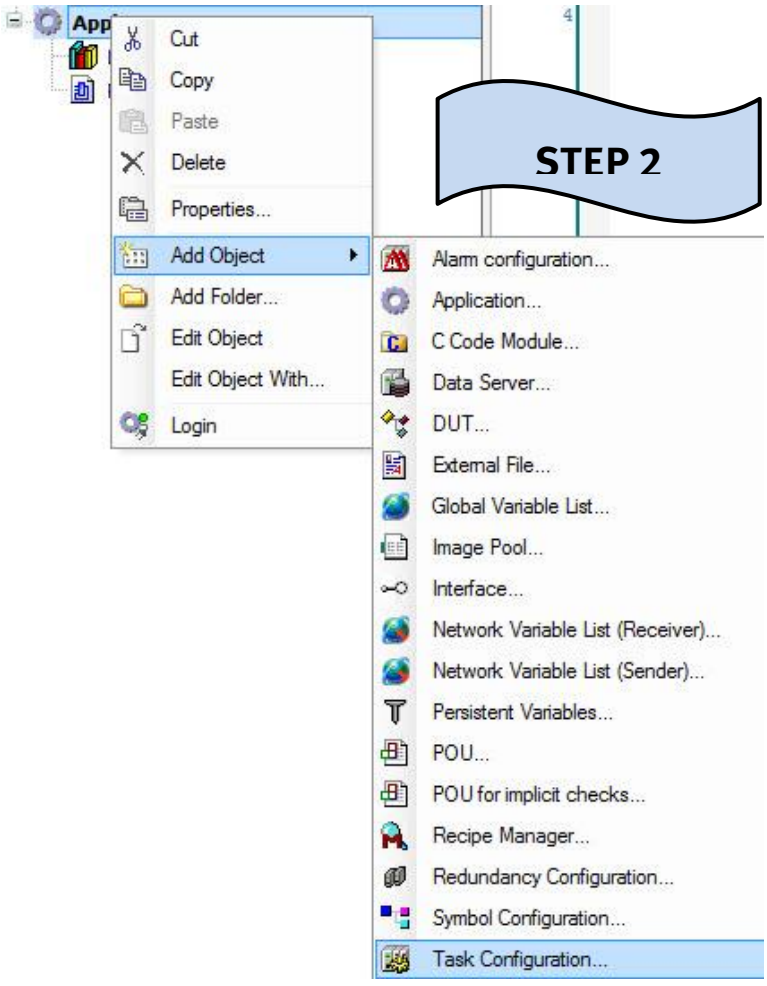
B) Add the Codesys Soft PLC

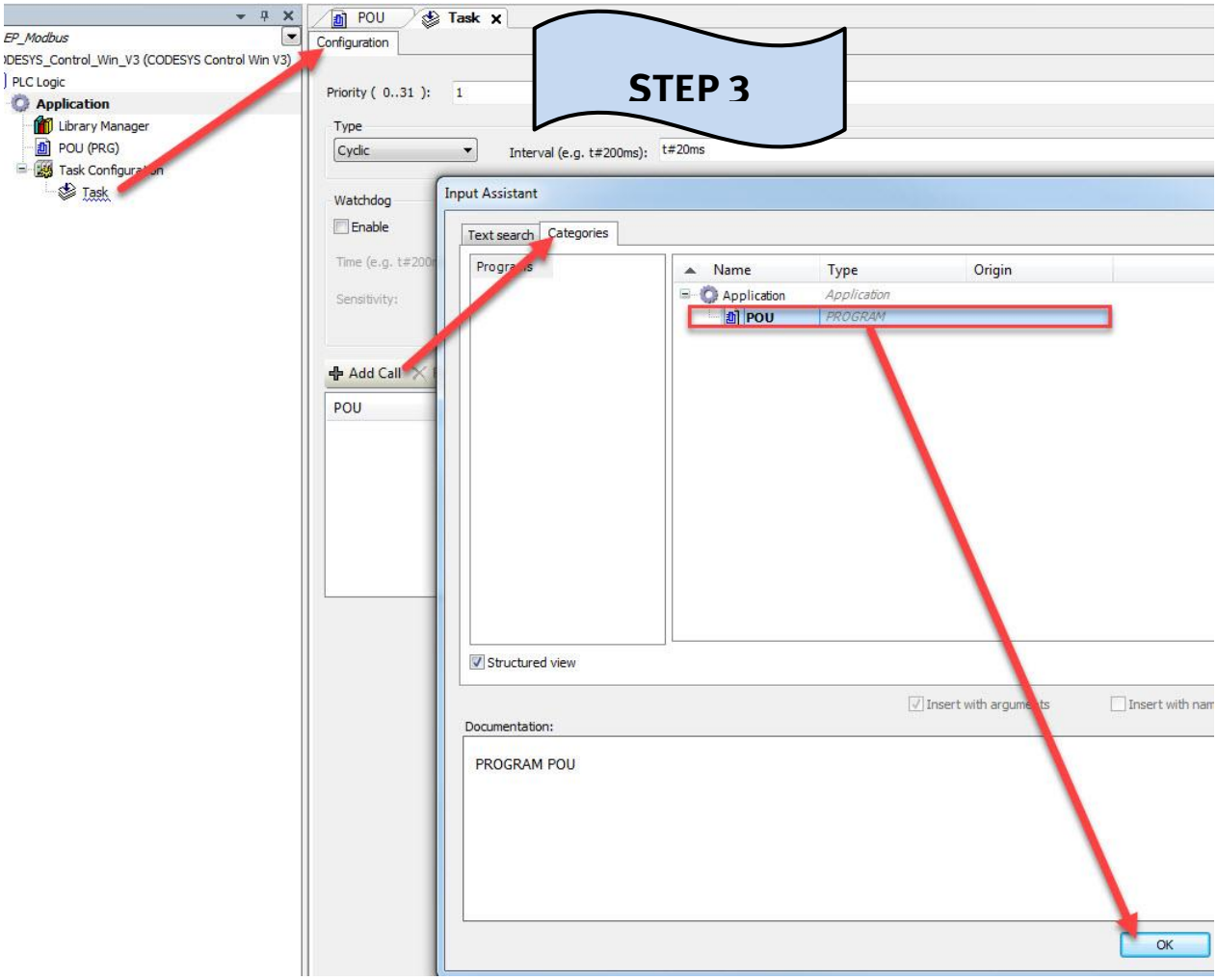




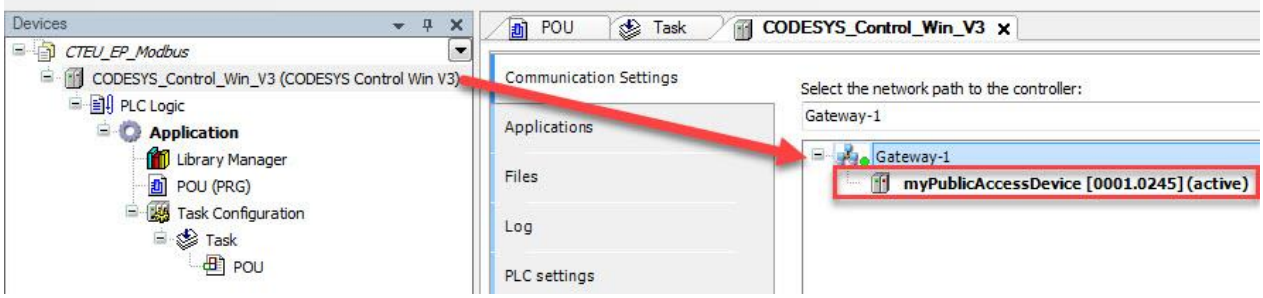
C) Add an empty program and a task



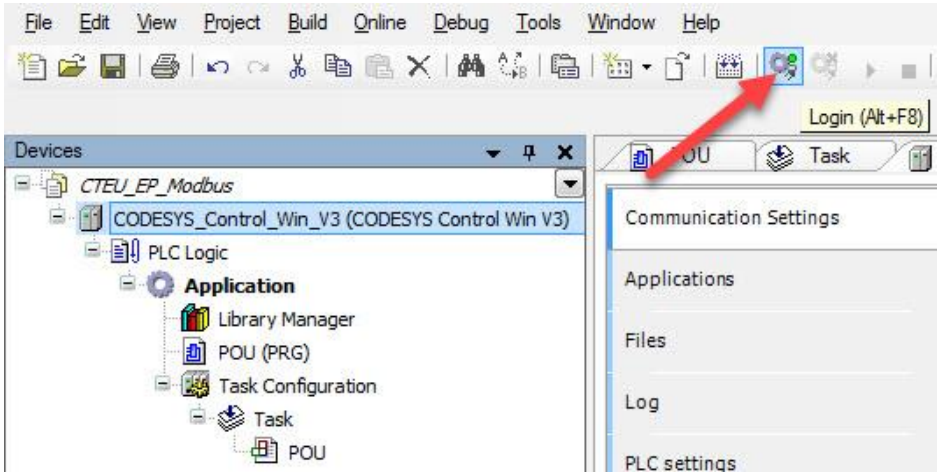




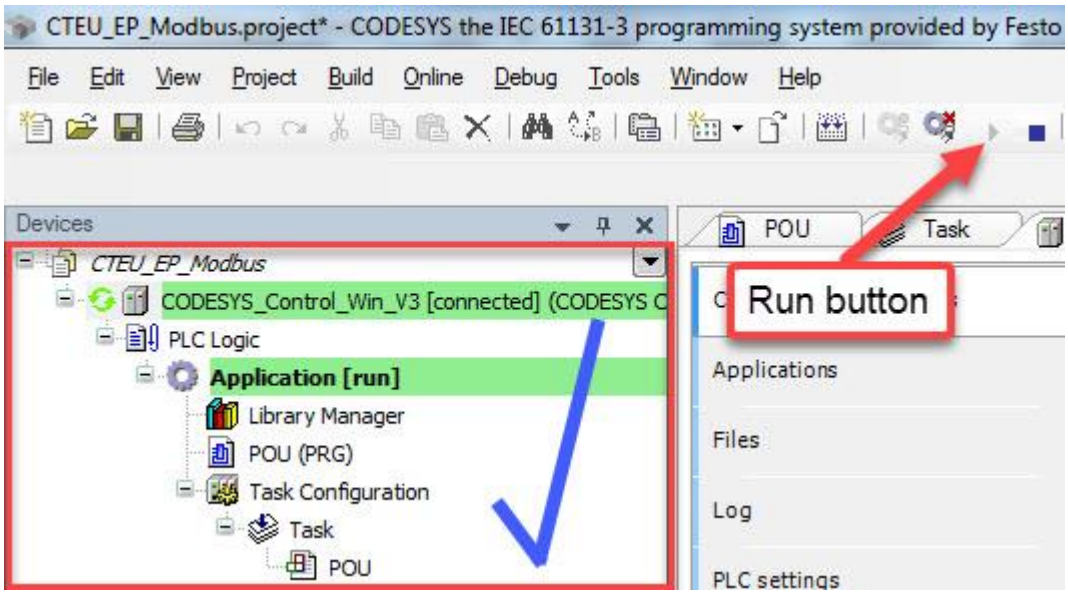
D) Connect to Codesys with your Laptop



E) Download the empty project and check online status



Result:

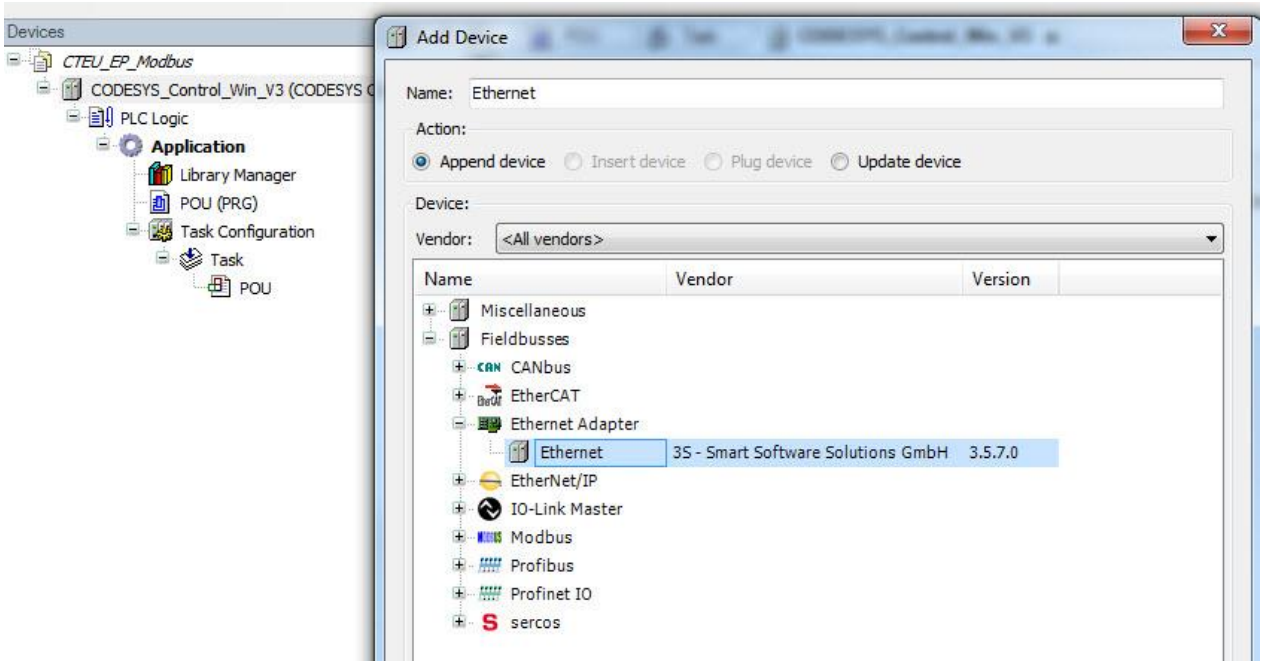


2.3 Create a Modbus TCP master/client

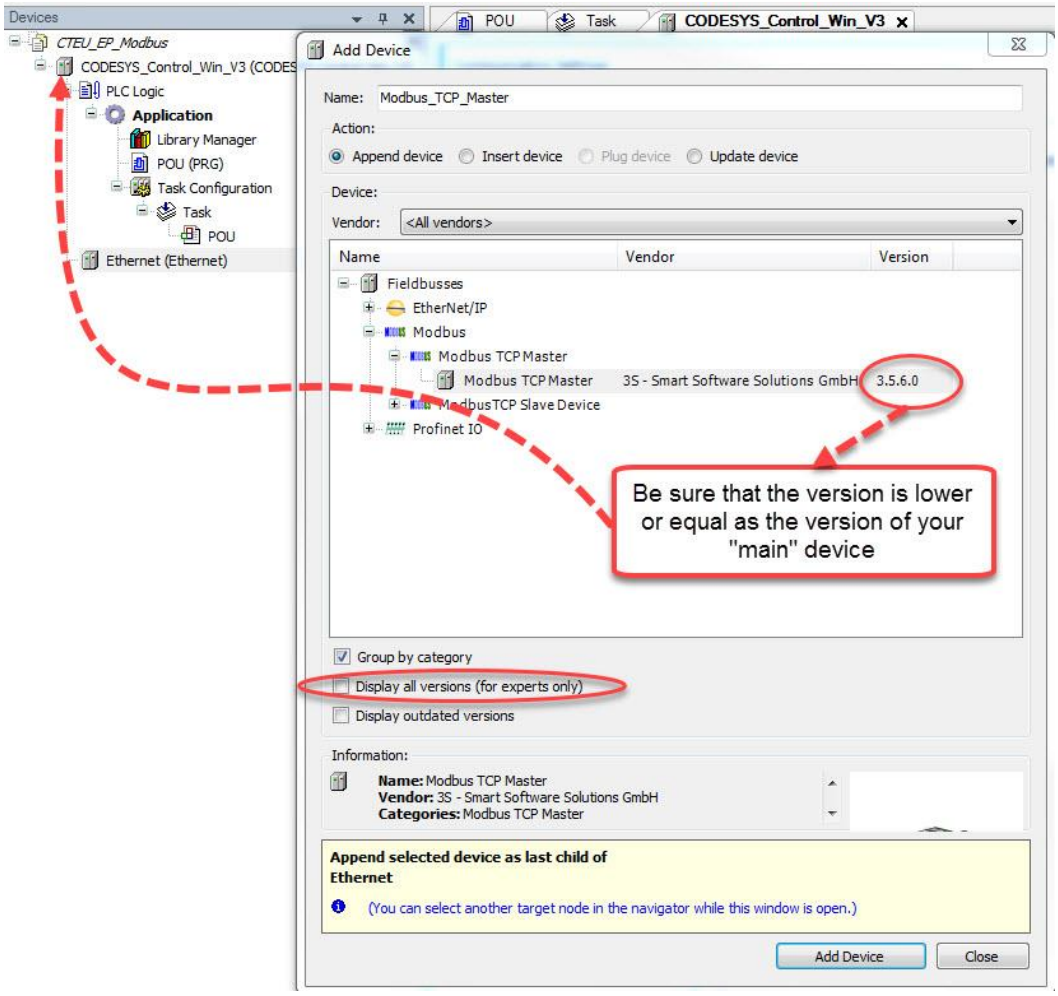
Caution

- The described procedure in chapter 2.3; 2.4; 2.5 is usable in combination with e.g. following FESTO devices too:
- - CECC-.. (PLC)
- - CPX-CEC-..-V3 (PLC)
  - CDPX (HMI) with Codesys license too!

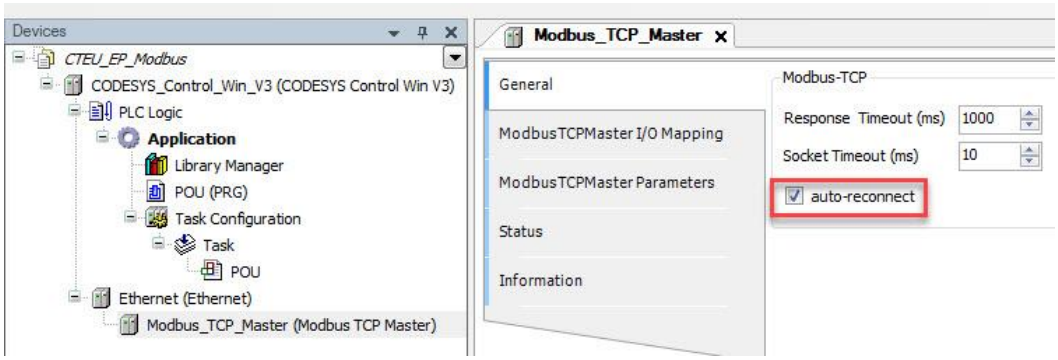
A) Add an Ethernet device to your Soft PLC



B) Add the Modbus TCP Master to the Ethernet Interface



C) Optional change the settings of the master



- Response Timeout:
The response timeout sets the time interval that the master will wait for the answer from a slave node. If the nodes do not answer within this time interval, then an error is recorded for the implicit slave function block. The value entered for the time interval is also the default value for each node. For each node, you can still set a specific value within its Modbus TCP slave configuration.

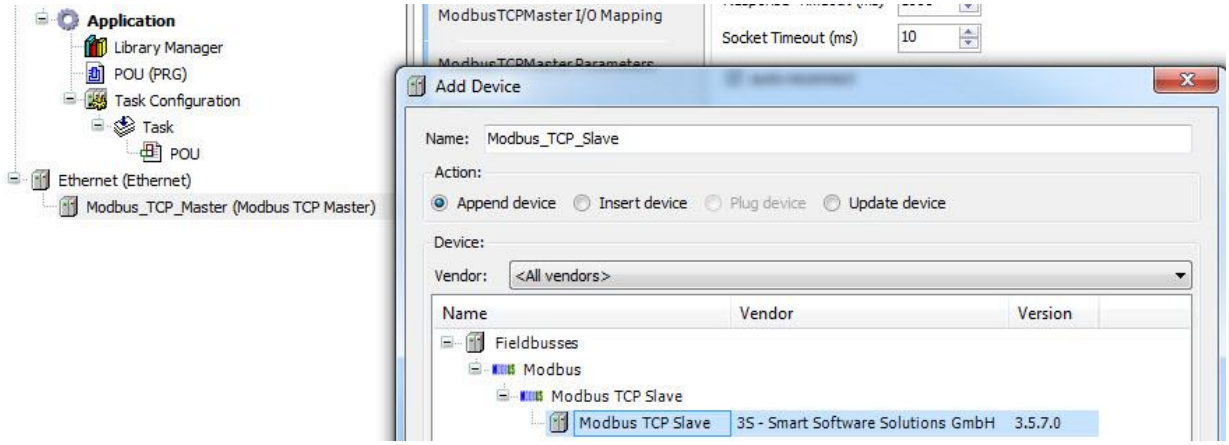
- Socket Timeout:
Maximum time to wait for incoming TCP/IP packages the bus cycle task can be blocked during this time, for example if a Modbus TCP slave is disconnected

-Auto-reconnect:
☑: After a communication error, Codesys automatically confirms the error and attempts to continue executing the Modbus command.

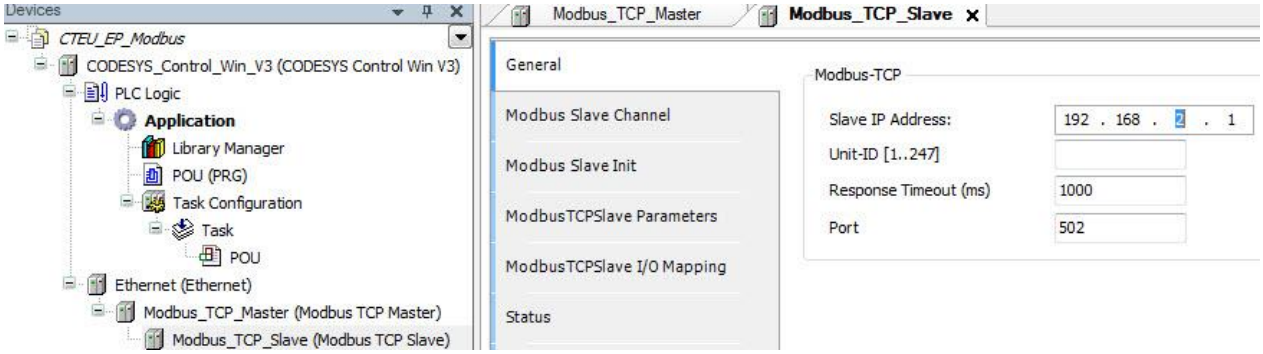
☐: The error must be explicitly confirmed in the slave function block.

2.4 Create a Modbus TCP slave/server

A) Add the Modbus TCP Slave to the master



B) Set the IP address of the Modbus TCP slave / server



- Unit-ID:

No impact on Modbus TCP devices

- Response Timeout (ms):

Time interval for the master to wait for the response from the slave. This is especially configured for this slave node and **overwrites** the general response timeout setting of the respective master.

➔ **Note**

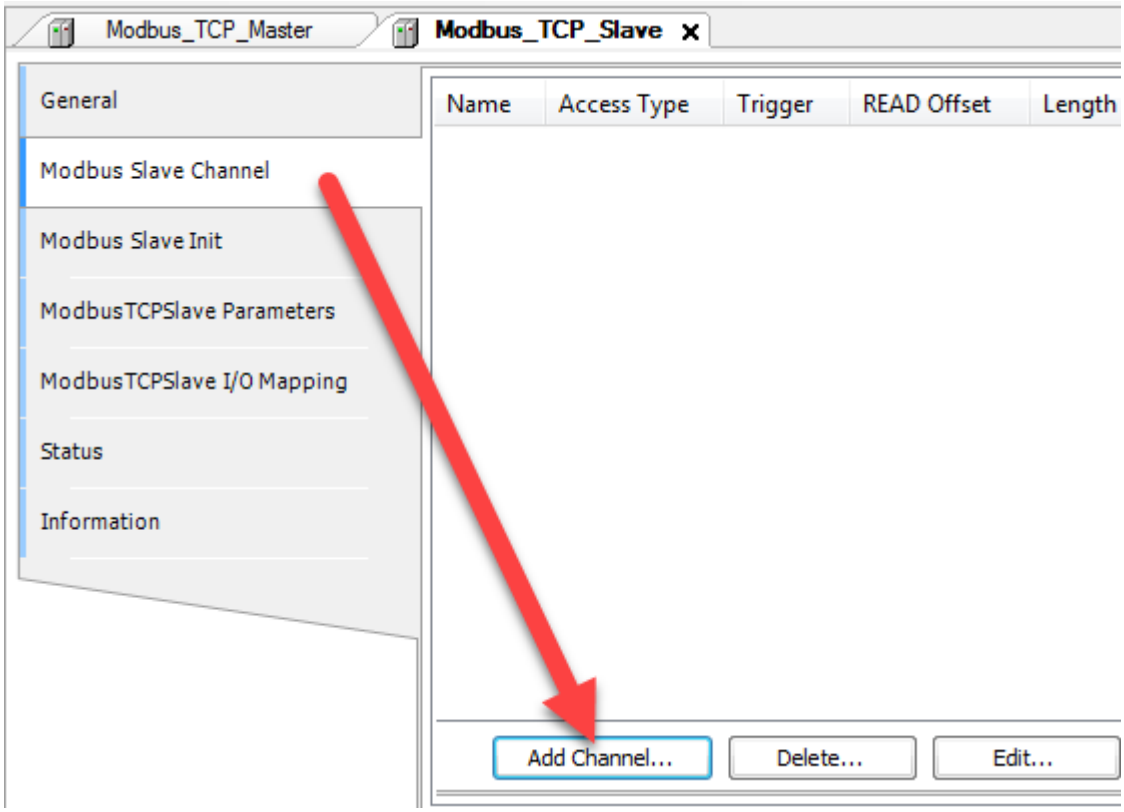
The CTEU-EP Modbus/TCP Idle timeout is default 100ms and defined in Modbus address 40 (Dez) and 41 (Dez). The setting in Codesys has only an influence on the master/client behavior!

- Port:

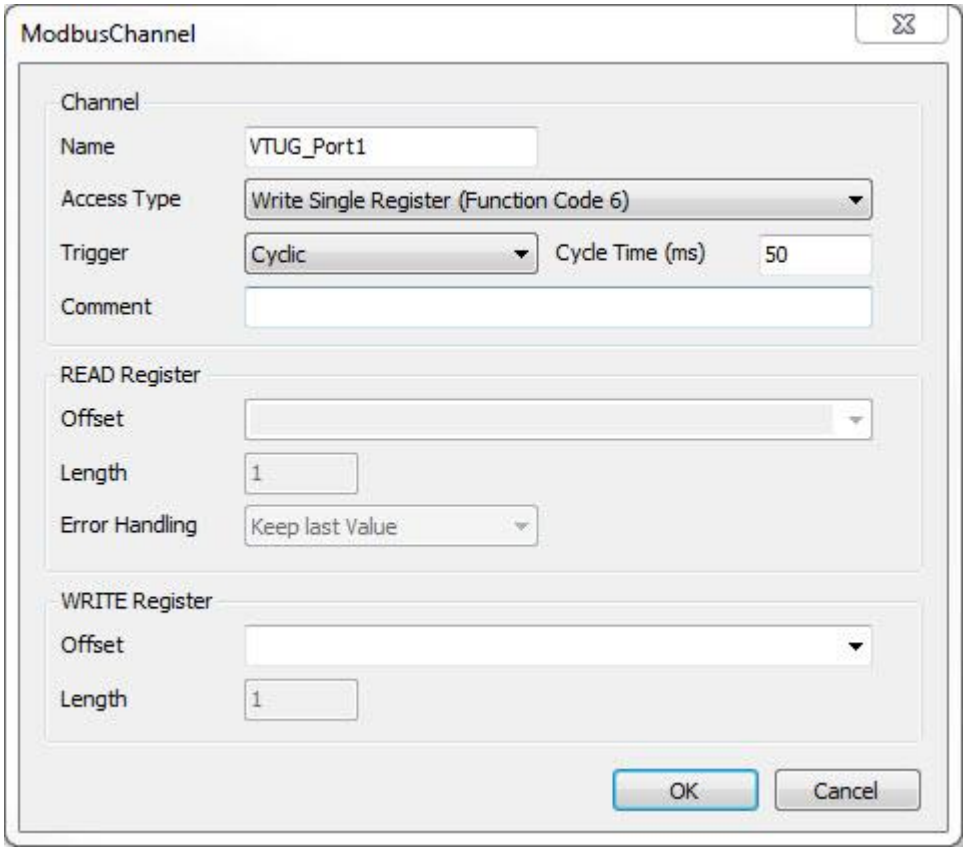
Port number (TCP/IP) of the slave 502 is default for Modbus

C) Define Modbus communication commands

Step1: Add a Channel



Step2: Define how the master/client communicates with the slave/server.



[Example screenshot for cyclic communication with a VTUG 2 Byte (16 coil) valve terminal]

Informations

A) Access Type defines the function code

-> The slave/server has to support such code otherwise no data exchange is possible

Access Type	▪ *Read Coils (Function Code 1)* ▪ *Read Discrete Inputs (Function Code 2)* ▪ *Read Holding Registers (Function Code 3)* ▪ *Read Input Registers (Function Code 4)* ▪ *Read Single Coil (Function Code 5)* ▪ *Write Single Register (Function Code 6)* ▪ *Write Multiple Coils (Function Code 15)* ▪ *Write Multiple Registers (Function Code 16)* ▪ *Read/Write Multiple Registers (Function Code 23)*

B) Trigger define how you send the Modbus/TCP message

Trigger	▪ *Cyclic*: The request occurs periodically. ▪ *Rising edge*: The request occurs as a reaction to a rising edge of the Boolean trigger variables.

C) Read register options

Offset	Start address where reading should start (value range 0-65535)
Length	Number of registers to be read (for word access) or number of discrete inputs to be read (for bit access)
Error Handling	Defines what should happen to the data in case of a communication error ▪ *Set to ZERO* ▪ *Keep last Value*

D) Write register options

Offset	Number of the register to be written to (value range 0-65535)
Length	Number of registers to be written to (= Words) Value range of the parameter depends on function code:

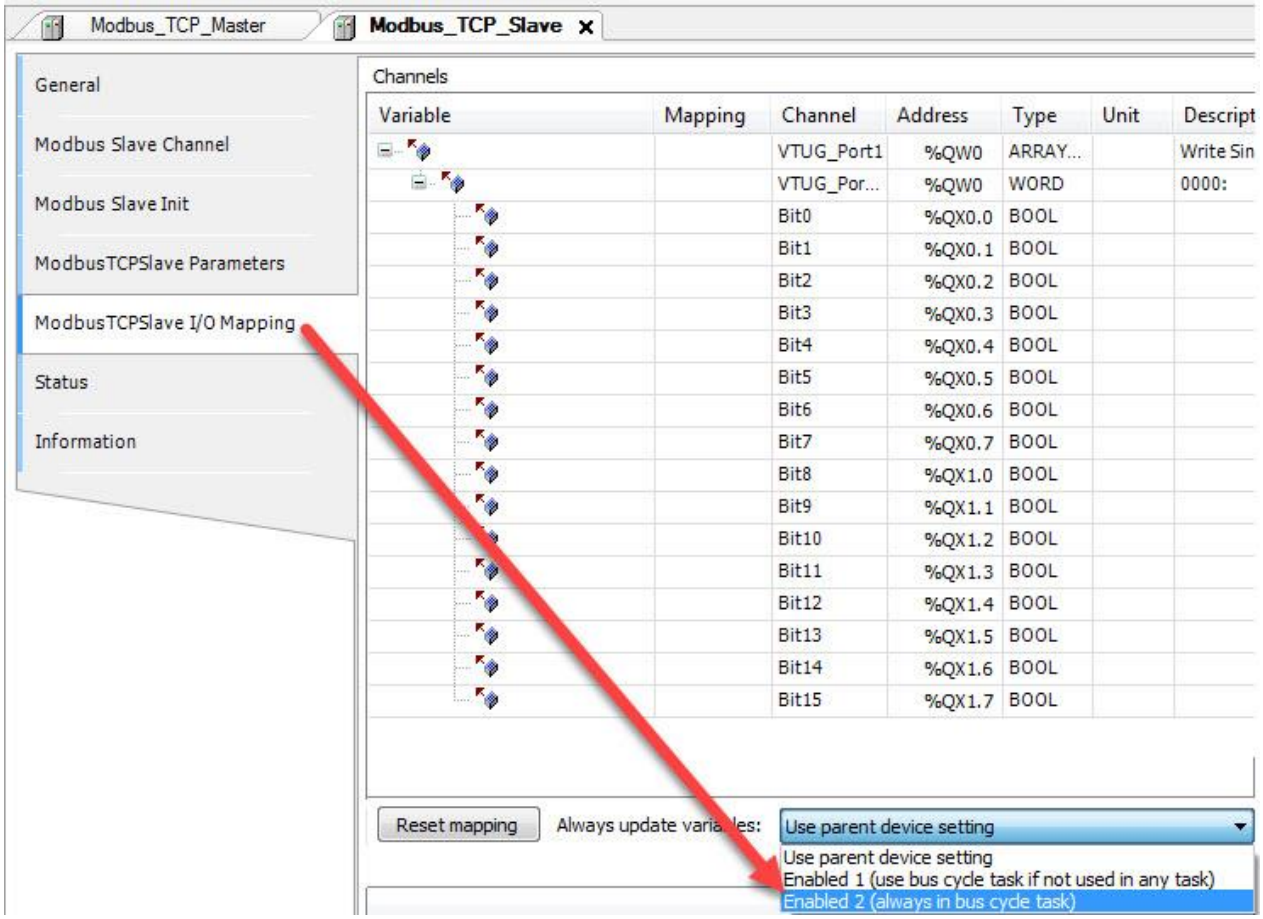
Result:

Modbus_TCP_Slave

Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length
VTUG_Port1	Write Single Register (Function Code 06)	CYCLIC, t#50ms				16#0000	1

Short explanation:
The master/client sends every 50ms the single register write command to the Modbus/TCP slave/server address 0 (Dez)

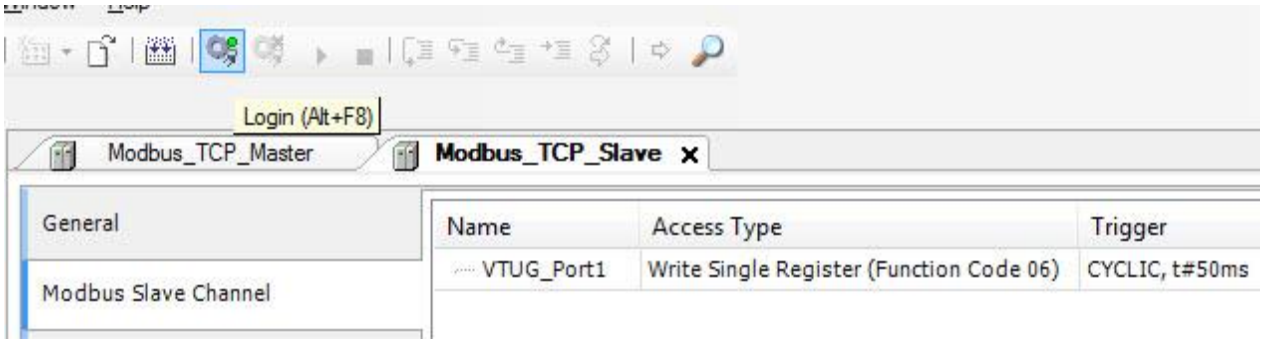
Step3: For testing purpose activate “Always update variables”

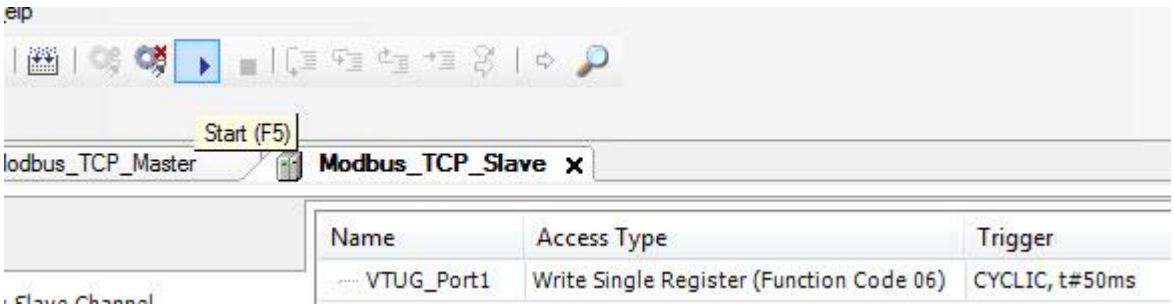


Note

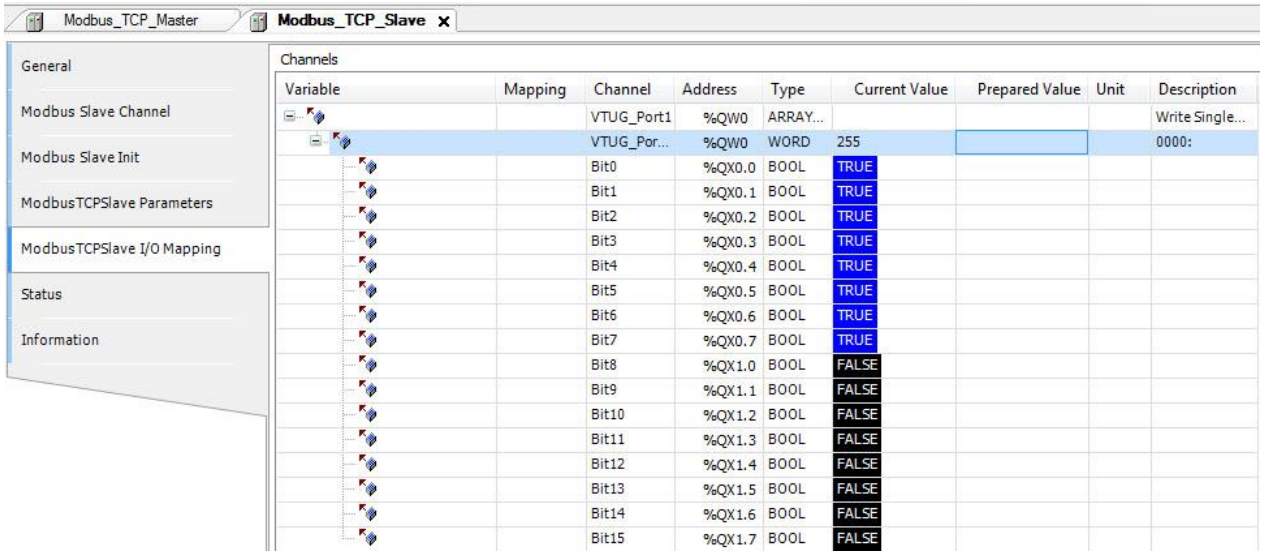
This setting makes sense to test the function in online mode. As soon as the variable is used in a project it should be changed, because it creates an additional load in the PLC.

Step4: Download project and test the valves





Test result in online mode:



2.5 Example how to change the parameters of a connected I-Port device

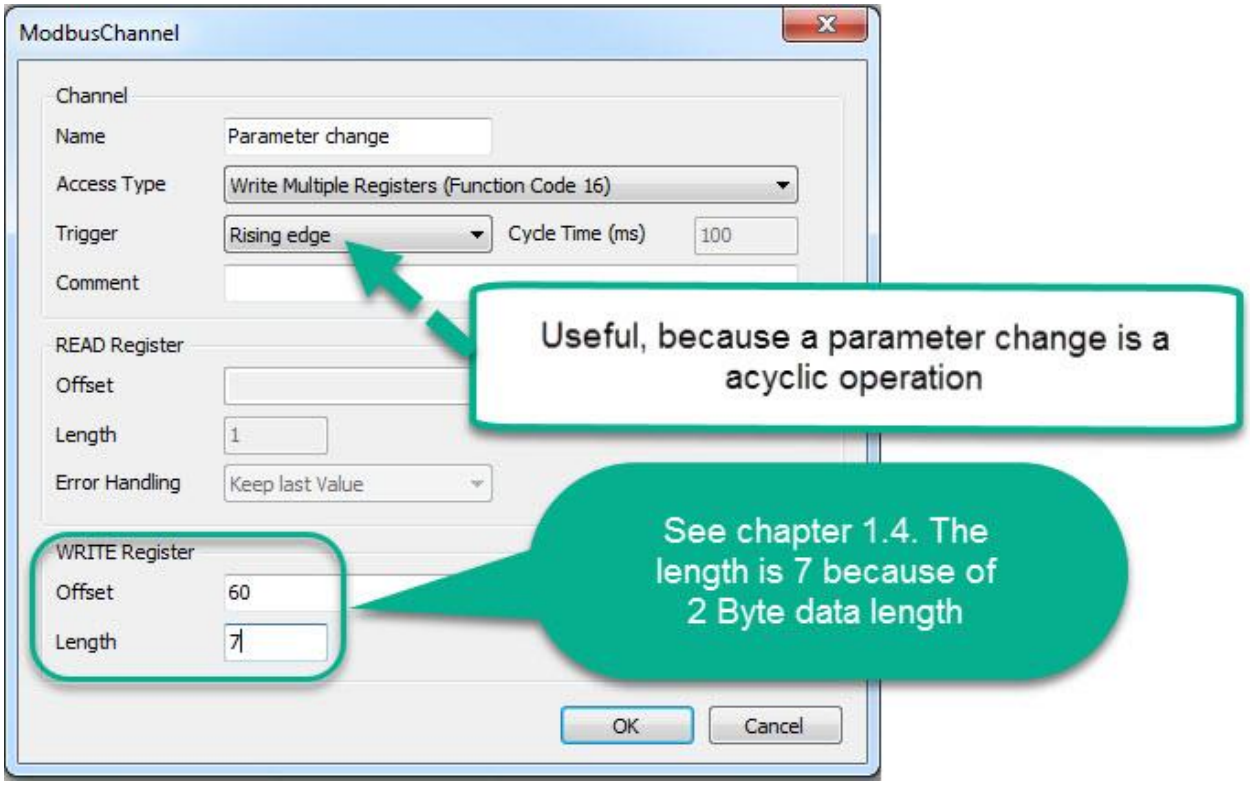
In the IO-Link application node of the SPAU-..-LK.. sensor you find e.g. following parameter

https://www.festo.com/net/SupportPortal/Files/376692/SPAU%20IO-Link%20parameter%20description.pdf

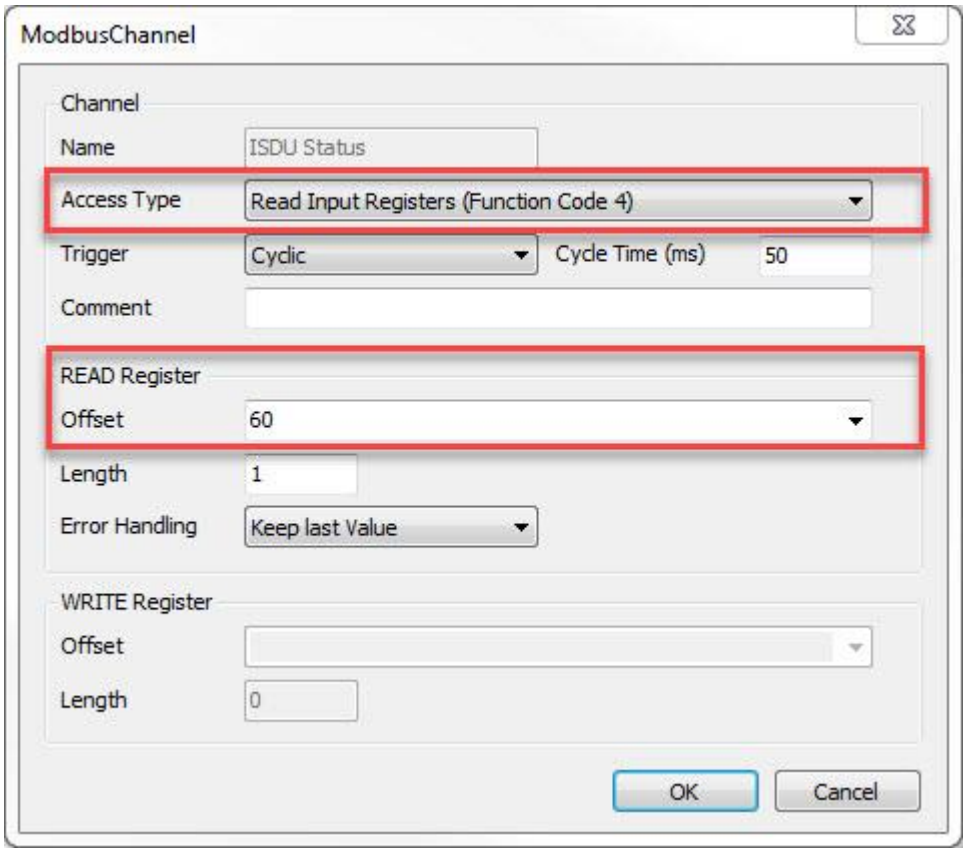
3.3.5 Device specific parameters

Index	Sub-Index	Name	Value	Access[1]			Length	Format
				U	M	S		
0x0112	0	OutA, Auto difference monitoring, max. signal delta (s.obS) boundary value for constant signal observation	16 ... 328, default 33	R	R/W	R/W	2 Bytes	UInteger16
0x0113	0	OutA, Auto difference monitoring, time delta (t.obS) time period for constant signal observation (msec)	5 ... 9999, default 200	R	R/W	R/W		
0x0114	0	OutA , Auto difference monitoring, switchpoint delta (d.SP) threshhold for pressure difference	82 ... 8192, default 328	R	R/W	R/W		
0x0118	0	OutA, backlight color (COLR)	0, allways blue (with display) allways green (without display) 1, red if Out = 0 2, red if Out = 1 default 0	R	R/W	R/W		

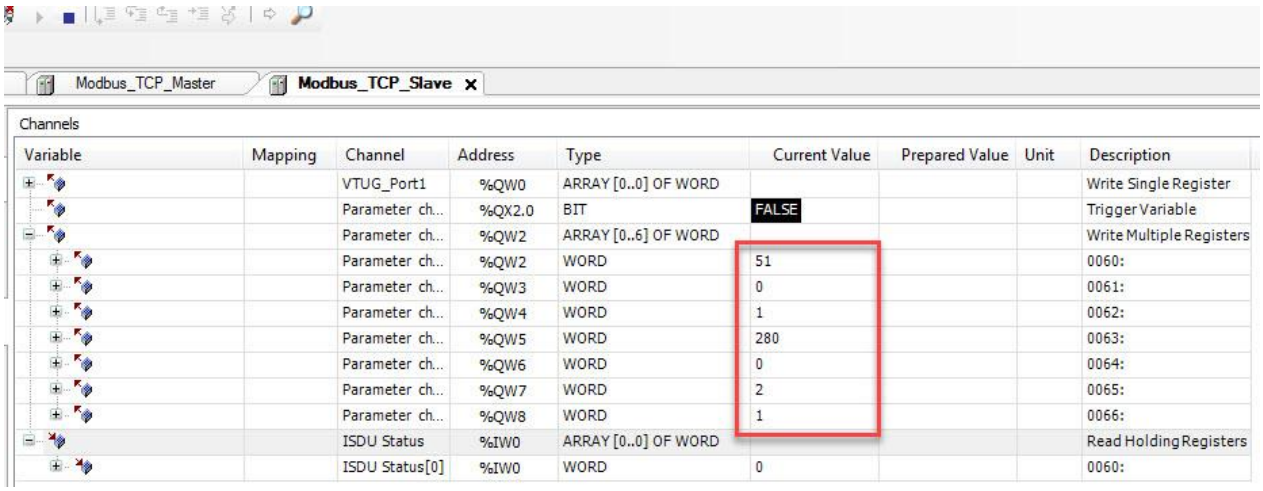
To change this Index you can use e.g. following Modbus command



Add a command for the ISDU status to check if everything works correctly



Download the project and define the values for the write command in online mode



Short value explanation:

QW2: 51 = ISDU Write operation

QW3: 0 = CTEU-EP module

QW4: 1 = I-Port X2, because on this port the SPAU-..-LK-.. is connected

QW5: 118 (Hex) = 280 (Dec) = Parameter Index

QW6: 0 (Hex) = Sub index

QW7: 2 = Byte length

QW8: 1 = Change display colour to red

Via the Rising edge the master/client send the command to the slave/server one time

The result is an error:

Channels

Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
		VTUG_Port1	%QW0	ARRAY [0..0] OF WORD				Write Single Register
		Parameter_ch...	%QX2.0	BIT	TRUE			Trigger Variable
		Parameter_ch...	%QW2	ARRAY [0..6] OF WORD				Write Multiple Registers
		Parameter_ch...	%QW2	WORD	51			0060:
		Parameter_ch...	%QW3	WORD	0			0061:
		Parameter_ch...	%QW4	WORD	1			0062:
		Parameter_ch...	%QW5	WORD	280			0063:
		Parameter_ch...	%QW6	WORD	0			0064:
		Parameter_ch...	%QW7	WORD	2			0065:
		Parameter_ch...	%QW8	WORD	1			0066:
		ISDU Status	%IW0	ARRAY [0..0] OF WORD				Read Holding Registers
		ISDU Status[0]	%IW0	WORD	254			0060:

The reason for the error is the data byte swap!

--› Change the data from 0000 0000 0000 0001 (Dec 1) to 0000 0001 0000 0000 (Dec 256)

The new result is:

Channels

Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
		VTUG_Port1	%QW0	ARRAY [0..0] OF WORD				Write Single Register
		Parameter_ch...	%QX2.0	BIT	TRUE			Trigger Variable
		Parameter_ch...	%QW2	ARRAY [0..6] OF WORD				Write Multiple Registers
		Parameter_ch...	%QW2	WORD	51			0060:
		Parameter_ch...	%QW3	WORD	0			0061:
		Parameter_ch...	%QW4	WORD	1			0062:
		Parameter_ch...	%QW5	WORD	280			0063:
		Parameter_ch...	%QW6	WORD	0			0064:
		Parameter_ch...	%QW7	WORD	2			0065:
		Parameter_ch...	%QW8	WORD	256			0066:
		ISDU Status	%IW0	ARRAY [0..0] OF WORD				Read Holding Registers
		ISDU Status[0]	%IW0	WORD	0			0060: