SOFTSTARTER TYPE PSTX

Fieldbus communication

Modbus TCP

PREPARED	STATUS	SECURITY LEVEL		
2019-10-10	Approved	Public		
RELEASED	DOCUMENT KIND			
2019-12-18	Manual			
OWNING ORGANIZATION	DOCUMENT ID.	REV.	LANG.	PAGE
Drives softstarters	1SFC132087M0201	E	en	1/28

Contents

Cyber Security Legal Disclaimer
This product is designed to be connected to and to communicate information and data via a network interface. It is your sole responsibility to provide and continuously ensure a secure connection between the product and your network or any other network (as the case may be). You shall establish and maintain any appropriate measures (such as but not limited to the installation of firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc.) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Ltd and its affiliates are not liable for damages and/or losses related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

Risk Mitigation and Secure Deployment
The idea is to create defence-in-depth protection for each network by allocating firewall solutions to the front of internal trusted networks of each network by manage firewalls, their configurations and access rules. The softstarter must be positioned in a trusted network, strictly limited and in a hosted portion of a network or control system. Configure firewalls according to the principle of denying everything that is not needed nor used. For secure remote access, use a VPN connection with an encryption layer to create a secure channel over an insecure network. For more information regarding cyber security and risk mitigation, download the document Cyber security for ABB drives from the ABB library.

Information
The device uses the default Modbus TCP port 502 (0x01F6). This port must be open in the firewall for both incoming and outgoing traffic to allow Modbus-TCP compatible devices to communicate.

1. Modbus TCP

Modbus is a master-slave protocol and only one device can transmit on the line at any time. The master (which in most cases is a PLC) manages the exchanges and only it can take the initiative. It interrogates each of the slaves in succession and no slave can send a message unless it is invited to do so. The master repeats the question when there is an incorrect exchange, and declares the interrogated slave absent if no response is received within a given time period. If a slave does not understand a message, it sends an exception response to the master who may or may not repeat the request.

The Modbus protocol is a fieldbus protocol that provides full control and status information of the softstarter, reading as well as writing of parameters. Through the fieldbus it is possible to start and stop the motor, read out currents and frequency, get information about protections, warnings, faults and much more.

See chapter 8 in the Installation and commissioning manual, document 1SFC132081M0201, for fieldbus related settings.

Before the Modbus RTU can be taken in operation following parameters must be set in the softstarter:

– Parameter 12.2 FB interface connector set to Anybus

– Parameter 12.3 Fieldbus control set to On (if using fieldbus only to monitor this parameter can be set to Off)

– The parameters controlling the Anybus CompactCom module IP settings needs to be set to match existing network configuration:

  - Parameter 12.5 Fieldbus IP address
  - Fieldbus 12.6 IP gateway
  - Fieldbus 12.7 IP net mask
  - Fieldbus 12.8 IP DHCP client

Information
After changing any of the communication parameters it is needed to perform a power cycle of the device for the parameter values to be taken into effect. Or another way for a commu-nication parameter value change to be taken into effect is to set parameter 12.2 FB interface connector to “None” and then set it back to “Anybus”.

If there is no message passed between the PSTX softstarter and the Anybus module for more than the configured fieldbus failure timeout time (parameter 19.12), the PSTX softstarter will trip on fieldbus communication failure protection (P1E00) and with the default configuration the motor will be stopped. If the communication system is setup in such a way that commands/requests are not continuously passed between the PLC and softstarter, this protection function should be disabled. The parameter 19.4 (Fieldbus failure op) can then be set to “Off”.

Caution!
The motor may start unexpectedly if there is a start signal present when doing any of the actions listed below.

- Switching from one type of control to another (fieldbus control/hardwire control)
- Reset all Settings

Information
When fastening the module into the com1 port, make sure that the module is properly aligned in the socket prior to applying any force. Rough handling and/or excessive force in combination with misalignment may cause mechanical damage to the module and/or the com1 and socket.

2. Modbus Addressing

When talking about Modbus addressing, there is often a misunderstanding about what an address really is. This section will try to clarify the conventions in this document.

2.1. Protocol Address

The Modbus standard specification uses one kind of address, a two byte unsigned integer (0-65535).

This is the address that is actually transmitted to the device.

2.2. Modicon Address

Modbus was originally developed by Modicon and the notation used then is still often used today, though considered obsolete by present standards.

The Modicon notation combines two pieces of information in a single number:

1. The register type
2. The register number

A register number offset defines the type and makes it possible to translate between the two types of addresses.

Table 1 Register types and ranges

Prefix	Register Type	Range
0x	Coil	00001-00001
1x	Discrete Input	10001-19999
3x	Input Register	30001-39999
4x	Holding Register	40001-49999

2.3. Translating Modicon address to protocol address

An example:
Modicon address 40002 selects the holding register at protocol address 0001 (40002 – 40001 = 1). The protocol address 0001 will be transmitted in the message packet.

3. PSTX Data

3.1. Digital input telegram

To PLC from the softstarter.
Use Modbus function code 02, Read Input Status.

Protocol Address	Modicon Address	Bit	Data	Description
0000h	10001	0	Auto Mode status[1]	0 = softstarter control from fieldbus not allowed.
0001h	10002	1	Event status	0 = No active fault/warning/protection.
0002h	10003	2	Ready To Start	0 = A start will probably cause a fault, 1 = A start will probably not cause a fault.
0003h	10004	3	FBT Response 0	See Fieldbus Tasks.
0004h	10005	4	FBT Response 1	See Fieldbus Tasks.
0005h	10006	5	FBT Toggle Bit	See Fieldbus Tasks.
0006h	10007	6	Programmable Digital Input 1	Function of programmable digital input, see section 3.2
0007h	10008	7	Programmable Digital Input 2	
0008h	10009	8	Programmable Digital Input 3	
0009h	10010	9	Programmable Digital Input 4	
000Ah	10011	10	Programmable Digital Input 5	
000Bh	10012	11	Programmable Digital Input 6	
000Ch	10013	12	Programmable Digital Input 7	
000Dh	10014	13	Programmable Digital Input 8	
000Eh	10015	14	Programmable Digital Input 9	
000Fh	10016	15	Programmable Digital Input 10	

[1)] Auto mode reflects the control state of the Softstarter. This is affected by a combination of:

- The Auto mode input signal from the PLC (Digital output telegram).
- The state of the Local/Remote switch on the HMI.
- The parameter “Fieldbus control”.

3.2. The digital input “Fieldbus disable”.Programmable Digital Inputs

The functions of the programmable Digital inputs are controlled by the parameters Fieldbus DI 1 through Fieldbus DI 10. The following functions are available for selection:

Function	Data
None	Value is set to 0.
Start feedback	Status of Start signal.
Stop feedback	Status of Stop signal.
Fault reset feedback	Status of Reset signal.
Slow speed reverse feedback	Status of Slow speed reverse signal.
Slow speed forward feedback	Status of Slow speed forward signal.
Start 1 feedback	Status of Start 1 signal.
Start 2 feedback	Status of Start 2 signal.
Start 3 feedback	Status of Start 3 signal.
Motor heating feedback	Status Motor heating signal.
User defined feedback	Status of User defined protection signal.
Stand still brake feedback	Status of Stand still brake signal.
Emergency mode feedback	Status of Emergency mode signal.
Start reverse feedback	Status of Start reverse signal.
Run status	1 = Indicates when the softstarter gives voltage to the motor.
TOR status	Top of Ramp. 1 = Indicates that motor runs on full voltage.
Line	Line or Inside Delta Connection; 0 = Line, 1 = Delta.
Phase sequence	0 = L1, L2, L3; 1 = L1, L3, L2.
Event group 0 status	0 = No active events present in group 0.
Event group 1 status	0 = No active events present in group 1.
Event group 2 status	0 = No active events present in group 2.
Event group 3 status	0 = No active events present in group 3.
Event group 4 status	0 = No active events present in group 4.
Event group 5 status	0 = No active events present in group 5.
Event group 6 status	0 = No active events present in group 6.
Sequence 1 Run status	Run status of sequence connected motor 1.
Sequence 2 Run status	Run status of sequence connected motor 2.
Sequence 3 Run status	Run status of sequence connected motor 3.
Sequence 1 TOR status	Top of Ramp status of sequence connected motor 1.
Sequence 2 TOR status	Top of Ramp status of sequence connected motor 2.
Sequence 3 TOR status	Top of Ramp status of sequence connected motor 3.
Run reverse status	1 = Indicates when the softstarter gives voltage to the motor after a reverse start.
Enable status	Status of Enable signal.

Function	Data
Digital In0 status	Status of internal digital input In0.
Digital In1 status	Status of internal digital input In1.
Digital In2 status	Status of internal digital input In2.
Local control status	0 = Remote control, 1 = Local control (HMI).
Cancel brake feedback	Status of Cancel brake signal.
Pump cleaning auto status	Status of automatic pump cleaning.
Pump cleaning forward status	Status of forward pump cleaning.
Pump cleaning backward status	Status of reverse pump cleaning.
External digital 1DI0 status	Status of external digital input 1DI0 (not supported when using internal Modbus RTU).
External digital 1DI1 status	Status of external digital input 1DI1 (not supported when using internal Modbus RTU).
External digital 1DI2 status	Status of external digital input 1DI2 (not supported when using internal Modbus RTU).
External digital 1DI3 status	Status of external digital input 1DI3 (not supported when using internal Modbus RTU).
External digital 1DI4 status	Status of external digital input 1DI4 (not supported when using internal Modbus RTU).
External digital 2DI5 status	Status of external digital input 2DI5 (not supported when using internal Modbus RTU).
External digital 2DI6 status	Status of external digital input 2DI6 (not supported when using internal Modbus RTU).
External digital 2DI7 status	Status of external digital input 2DI7 (not supported when using internal Modbus RTU).
HW DI Start status	Status of the hard wire internal digital input Start.
HW DI Stop status	Status of the hard wire internal digital input Stop.
Ready to start (line contactor)	Same conditions as the Ready To Start bit except that the incoming three phase voltage condition is excluded. The bit can be used when a line contactor is connected.

3.3. Analog input telegram

To PLC from the softstarter.

All analog data is represented as 16-bit values.
Use Modbus function code 04, Read Input Registers.

A protocol for Fieldbus tasks is used to read and write parameters. It is applicable for all fieldbuses.

Protocol Address	Modicon Address	Data	Representation
0001h	30002	FBT Return Value	See Fieldbus Tasks
0002h	30003	Programmable Analog Input 1	Function of programmable analog input, see section 3.4
0003h	30004	Programmable Analog Input 2	
0004h	30005	Programmable Analog Input 3	

Protocol Address	Modicon Address	Data	Representation
0005h	30006	Programmable Analog Input 4	
0006h	30007	Programmable Analog Input 5	
0007h	30008	Programmable Analog Input 6	
0008h	30009	Programmable Analog Input 7	
0009h	30010	Programmable Analog Input 8	
000Ah	30011	Programmable Analog Input 9	
000Bh	30012	Programmable Analog Input 10	

3.4. Programmable Analog Inputs

The functions of the programmable analog inputs are controlled by the parameters Fieldbus AI 1 through Fieldbus AI 10. The following functions are available for selection:

Function	Representation
None	Value is set to 0
Phase L1 current[1]	Value = 1000 ⇒ 100A
Phase L2 current[1]	Value = 1000 ⇒ 100A
Phase L3 current[1]	Value = 1000 ⇒ 100A
Active power (hp)	Value = 1000 ⇒ 10hp
Active power	Value = 1000 ⇒ 10kW
Apparent power	Value = 1000 ⇒ 10kVA
Mains voltage	Value = 1000 ⇒ 100V
Power factor	Value = 100 ⇒ 1 Example: 87 ⇒ 0.87
Motor voltage	Value = 100 ⇒ 100%
Active energy (resettable)	Value = 1000 ⇒ 10kWh
EOL time to trip	Value = 100 ⇒ 100s Value = 65535 ⇒ No overload Value = 0 ⇒ Trip already occurred
Mains frequency	Value = 1000 ⇒ 100Hz
Max phase current1	Value = 1000 ⇒ 100A
Motor current	Value = 1000 ⇒ 100A
Motor run time (resettable)	Value = 100 ⇒ 1000h
Motor temperature	Value = 100 ⇒ 100℃
Motor temperature percent	Value = 100 ⇒ 100%
Number of starts (resettable)	Value = 1 ⇒ 100
Phase sequence	Value = 0 ⇒ L1->L2->L3 Value = 1 ⇒ L1->L3->L2 Value = 2 ⇒ No sequence detected
PT100 temperature	Value = n ⇒ n/10 – 50℃ Example: 750 ⇒ 25℃
PTC resistance	Value = 100 ⇒ 100Ω

Function	Representation
Reactive energy (resettable)	Value = 1000 ⇒ 10kVArh
Reactive power	Value = 1000 ⇒ 100VAr
Remaining time to start	Value = 100 ⇒ 100s
Thyristor temperature	Value = 100 ⇒ 100℃
Thyristor temperature percent	Value = 100 ⇒ 100%
EOL time to cool	Value = 100 ⇒ 100s
Top event code	Value = 1000 ⇒ 1000
Motor current in percent of IE.	Value = 100 ⇒ 100%
Thyristor run time (resettable)	Value = 1 ⇒ 10h
Motor connection	Value = 0 ⇒ auto Value = 1 ⇒ In-line Value = 2 ⇒ Inside delta – UI Value = 3 ⇒ Inside delta – IU Value = 4 ⇒ 2-phase L1 shorted Value = 5 ⇒ 2-phase L2 shorted Value = 6 ⇒ 2-phase L3 shorted
Phase L1 current high range[2]	Value = 100 ⇒ 100A
Phase L2 current high range[2]	Value = 100 ⇒ 100A
Phase L3 current high range[2]	Value = 100 ⇒ 100A
Active power (hp) high range[2]	Value = 100 ⇒ 100hp
Active power high range[2]	Value = 100 ⇒ 100kW
Apparent power high range[2]	Value = 100 ⇒ 100kVA
Reactive power high range[2]	Value = 100 ⇒ 100kVAr
Max phase current high range[2]	Value = 100 ⇒ 100A
Max motor current high range[2]	Value = 100 ⇒ 100A
Active energy high range[2]	Value = 1 ⇒ 10000kWh
Reactive energy high range[2]	Value = 1 ⇒ 10000kVArh
Number of starts (high precision)	Value = 1 ⇒ 1

[1)] Phase current L1, L2 and L3 indicate the current through the softstarter, while the Max phase current is always the line current.

[2)] High Range alternatives are available for a few signals where there is a possibility for the values to wrap. The values are 16-bit so the maximum value for each signal is 65535. The High Range alternatives have different scaling and will never wrap around but instead have lower precision.

3.5. Digital output telegram

From PLC to the softstarter.
Use Modbus function code 15 (0Fh), Force Multiple Coils.

Protocol Address	Modicon Address	Bit	Data	Description
0000h	1	0	Start	Commence a start when signal is set.
0001h	2	1	Stop	Commence a stop when signal is negated.
0002h	3	2	Fault reset	Reset signal for possible events.
0003h	4	3	Auto mode	This must be set for controlling the motor.
0004h	5	4	Slow speed reverse	Perform slow speed reverse when signal is set.
0005h	6	5	Slow speed forward	Perform slow speed when signal is set.
0006h	7	6	Spare	
0007h	8	7	Start1	Start1 if sequence start.
0008h	9	8	Start2	Start2 if sequence start.
0009h	10	9	Start3	Start3 if sequence start.
000Ah	11	10	Motor heating	Perform motor heating when signal is set.
000Bh	12	11	Stand still brake	Perform stand still brake when signal is set.
000Ch	13	12	Start reverse	Commence a reverse start when signal is set.
000Dh	14	13	Spare	
000Eh	15	14	Emergency mode	Set to “1” to enable emergency mode.
000Fh	16	15	FBT toggle bit	See Fieldbus Tasks.
0010h	17	16	User defined trip	Set to “1” to trigger user defined protection.
0011h	18	17	Switch to remote control	Switch to remote control when signal is set (rising edge triggered).
0012h	19	18	Pump cleaning automatic	Perform automatic pump cleaning when signal is set.
0013h	20	19	Pump cleaning forward	Perform forward pump cleaning when signal is set.
0014h	21	20	Pump cleaning reverse	Perform reverse pump cleaning when signal is set.
0015h	22	21	K4 relay command	Set “1” to activate the internal K4 output relay. Note that parameter 10.4 K4 function has to be set as “Fieldbus”
0016h	23	22	K5 relay command	Set “1” to activate the internal K5 output relay. Note that parameter 10.5 K5 function has to be set as “Fieldbus”

Protocol Address	Modicon Address	Bit	Data	Description
0017h	24	23	K6 relay command	Set “1” to activate the internal K6 output relay. Note that parameter 10.6 K6 function has to be set as “Fieldbus”
0018h	25	24	1DO0 relay command	Set “1” to activate the external 1DO0 output relay. Note that parameter 11.9 1DO0 function has to be set as “Fieldbus”
0019h	26	25	1DO1 relay command	Set “1” to activate the external 1DO1 output relay. Note that parameter 11.10 1DO1 function has to be set as “Fieldbus”
001Ah	27	26	2DO2 relay command	Set “1” to activate the external 2DO2 output relay. Note that parameter 11.11 2DO2 function has to be set as “Fieldbus”
001Bh	28	27	2DO3 relay command	Set “1” to activate the external 2DO3 output relay. Note that parameter 11.12 2DO3 function has to be set as “Fieldbus”
001Ch	29	28	Spare	
001Dh	30	29	Spare	
001Eh	31	30	Spare	
001Fh	32	31	Spare	

3.6. Analog output telegram

From PLC to the softstarter.

All analog data is represented as 16-bit values.
Use Modbus function code 16 (10h), Preset Multiple Registers.

Protocol Address	Modicon Address	Data	Representation
0002h	40003	FBT Control Word	This register is used to read parameters (see fieldbus tasks).
0003h	40004	Fieldbus AO 1 (FBT Argument 2 or Internal analog output)	Parameter 12.37 Fieldbus AO1 decides the use of this register. If set as "FBT Argument 2”, it is used to write parameters and set time (see fieldbus tasks). If set as “Internal analog output” this value of this register controls the internal analog output. Note that parameter 10.8 AO type needs to be set as “Fieldbus [%]”.
0004h	40005	Fieldbus AO 2 (FBT Argument 3 or External analog output)	Parameter 12.38 Fieldbus AO2 decides the use of this register. If set as "FBT Argument 3”, it is used to write parameters and set time (see fieldbus tasks). If set as “External analog output” this value of this register controls the external analog output. Note that parameter 11.14 1AO0 type needs to be set as “Fieldbus [%]”.

4. Modbus TCP - A set-up example

4.1. Softstarter PST Modbus TCP communication

This document describes an application example between a Modbus TCP master (PLC CPU, PC, etc.) and the ABB softstarter PSTX using the built-in Modbus TCP interface.

Please always use the actual softstarter manuals. In this particular example following documents has been used:

– Softstarter PSTX Installation and commissioning manual, document 1SFC132081M0201

4.2. Settings

1. **Set the softstarter network settings and field bus communication.**
   Either set Fieldbus IP address, Fieldbus IP gateway and Fieldbus IP net mask to desired network configuration or enable Fieldbus IP DHCP client parameter to receive the IP configuration from your DHCP server.
   Enable fieldbus control (Fieldbus control = On) to allow softstarter control from the fieldbus.
2. **Select the Anybus CompactCom interface.**
   The previous changes are taken in effect when the fieldbus interface is changed. Change parameter FB interface connector to Anybus.

5. Fieldbus Tasks

By using Fieldbus Tasks it is possible to read/write parameters and set the real-time clock.

Which task to execute is selected by filling in the FBT Control Word. There are three signals for arguments to the task:

– FBT Argument 1 is packed together with the Task ID in the FBT Control Word.

– There are two additional 16-bit arguments in separate analog output signals, FBT Argument 2 and FBT Argument 3.

To control when the task is executed, the digital output signal FBT Toggle Bit shall be changed. The softstarter will detect the change, execute the task, fill in the return values, and toggle the digital input signal FBT Toggle Bit as acknowledgement. Thus, the return values must be disregarded if the two toggle bits have different value.

5.1. FBT Control Word

The control word is a 16-bit analog output value sent from the PLC to the softstarter. It consists of a Task ID and an 11-bit argument packed together.

15	**14, 13, 12,**	**11**	**10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0**
-	Task ID	-	Argument 1

5.2. Task ID

The task identifier controls which function should be performed.

Task ID	Task	Response ID	
		Positive	Negative
0	No task	0	-
1	Request parameter value, lower word	1	2
2	Change parameter value	1	2
3	Set date and time	1	2
4	Request parameter value, upper word	1	2

5.3. Response ID

The response ID is the softstarter response to a task. It tells whether a task was executed successfully. If there was an error, an additional error code is returned in the FBT Return Value analog input. The Response ID is transmitted as two digital input signals, FBT Response 0 and FBT Response 1.

Response ID	FBT Response 1	FBT Response 0	Explanation
0	0	0	No response
1	0	1	Task executed
2	1	0	Task cannot be executed (with error number)
3	1	1	Reserved.

5.4. Error codes

The following error codes are sent when a task cannot be executed.

Error code	Explanation
0	Illegal parameter number
1	Parameter value cannot be changed
3	Lower or upper limit violated
4	Invalid argument
5	No error
6	Invalid task number

5.5. Request parameter value, lower word

This task reads the lower 16 bits of the specified parameter. See chapter 5.9 for parameter number and value scaling information.

5.5.1. Arguments

– FBT Argument 1: parameter number.

5.5.2. Return Value

– Response ID 1 and parameter value in FBT Return Value on success.

– Response ID 2 and error number in FBT Return Value on failure.

5.6. Change parameter value

This task writes a specified value to a parameter. See chapter 5.9 for parameter number and value scaling information.

5.6.1. Arguments

– FBT Argument 1: parameter number.

– FBT Argument 2: parameter value (lower word).

– FBT Argument 3: parameter value (upper word).

5.6.2. Return Value

– Response ID 1 on success.

– Response ID 2 and error number in FBT Return Value on failure.

5.7. Set date and time

This task updates the real-time clock on the softstarter. The date and time fields have the following limits:

Year: 0-63 (2000-2063)

Month: 1-12

Day: 1-31

Hour:0-23

Minute:0-59

Second:0-59

5.7.1. Arguments

FBT Argument 2: year, month, day and least significant bit of seconds.

15	**14, 13, 12, 11, 10, 9**	**8, 7, 6, 5**	**4, 3, 2, 1, 0**
s0	year	month	day

FBT Argument 3: hour, minute, seconds, bit 1-5.

15, 14, 13, 12, 11	10, 9, 8, 7, 6, 5	4, 3, 2, 1, 0
hour	minute	seconds, bit 1-5

5.7.2. Return Value

Response ID 1 on success.

Response ID 2 and error number in FBT Return Value on failure. In case the supplied time didn't differ from the set time, error code 5 (no error) is used.

5.8. Request parameter value, upper word

This task reads the upper 16 bits of the specified parameter. See chapter 5.9 for parameter number and value scaling information.

5.8.1. Arguments

– FBT Argument 1: parameter number.

5.8.2. Return Value

– Response ID 1 and parameter value in FBT Return Value on success.

– Response ID 2 and error number in FBT Return Value on failure.

5.9. Parameter numbers and values

To access parameters from the fieldbus, a unique parameter number is needed, this can be found in document 1SFC132081M0201, Chapter 7.19 Complete parameter list, Table 5.

Since the parameter values need to be represented as integers on the fieldbus while, the parameter values with greater precision need to be scaled. In document 1SFC132081M0201, Chapter 7.19 Complete parameter list,Table 5, there is a column specifying the number of decimals for each parameter.

– Parameter values that are read from the fieldbus needs to be divided by $10^{\text{numbers of decimals}}$.

– Parameters values that are written from the fieldbus needs to be multiplied by $10^{\text{numbers of decimals}}$.

For example:
The parameter Kick start time has parameter number 24 and 2 decimals. To read this parameter:

1. Set FBT Task ID to 1.
2. Set FBT Argument 1 to 24.
3. Toggle FBT Toggle Bit output and wait for the FBT Toggle Bit input to update.
4. Response ID 1 should now contain value 1, indicating success.
5. FBT Return Value contains the value 50 (this is an example and depends on the actual set value).
6. The return value should be interpreted as $50/10^2 = 0.5$s.

To change the Kick start time **parameter to 1s:**

1. Set FBT Task ID to 2.
2. Set FBT Argument 1 to 24.
3. Set FBT Argument 2 to $1*10^2 = 100$.
4. Set FBT Argument 3 to 0 as 100 <= 65535 which means it doesn't require more than 16 bits.
5. Toggle FBT Toggle Bit output and wait for the FBT Toggle Bit input to update.
6. Response ID 1 should now contain value 1, indicating success

5.9.1. Negative values

Negative values are represented internally using 32-bit two's complement numbers.

Example:
Setting parameter 17.5 PT100 reset temp (parameter number 249) to a value of -25°C:

The two's complement of -25 is $FFFFFFE7_{hex}$. The upper word is $FFFF_{hex}$ and the lower $FFE7_{hex}$, in decimal notation 65535 and 65511.

1. Set FBT Task ID to 2 for Change parameter value.
2. Set FBT Argument 1 to 249 to specify the parameter.
3. Set FBT Argument 2 to 65511 to specify the lower word.
4. Set FBT Argument 3 to 65535 to specify the upper word.
5. Toggle FBT Toggle Bit output and wait for the FBT Toggle Bit input to update.
6. Response ID 1 should now contain value 1, indicating success.

6. Example application with Automation Builder

This section shows a demo about how to start and stop motor by sending commands from fieldbus that is controlled by Programmable logic controller (PLC). We use Automation Builder as an example platform and show the demo about building such communication setting.

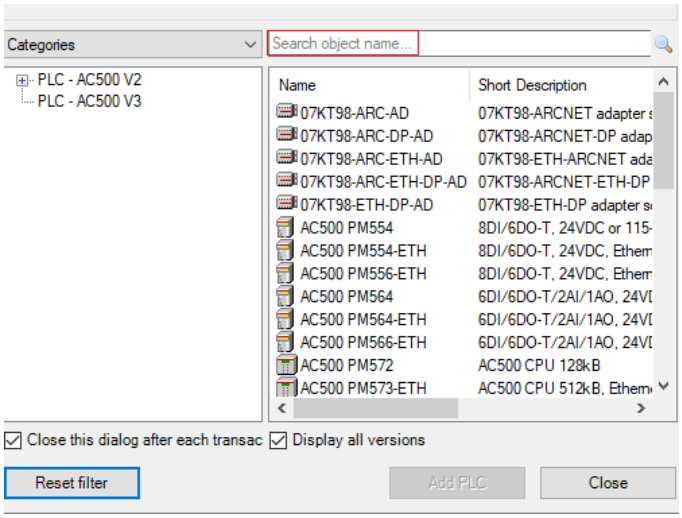
6.1. Create a new project in Automation Builder

We perform the following steps in Automation Builder 2.1 for PLC AC500 PM573.

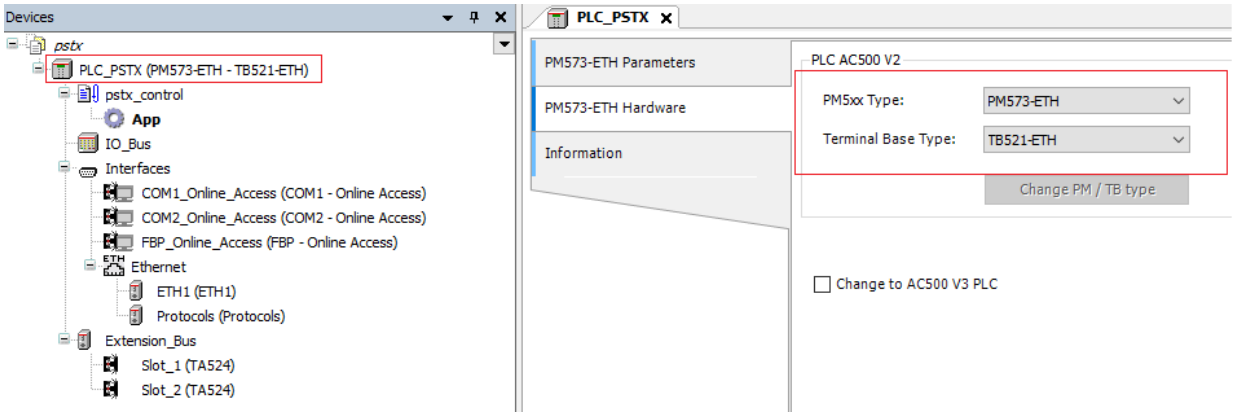
1. Open Automation Builder

2. Select File->New Project->AC500 project->OK

3. Select the correct PLC CPU in Search object name ...-> Add PLC.

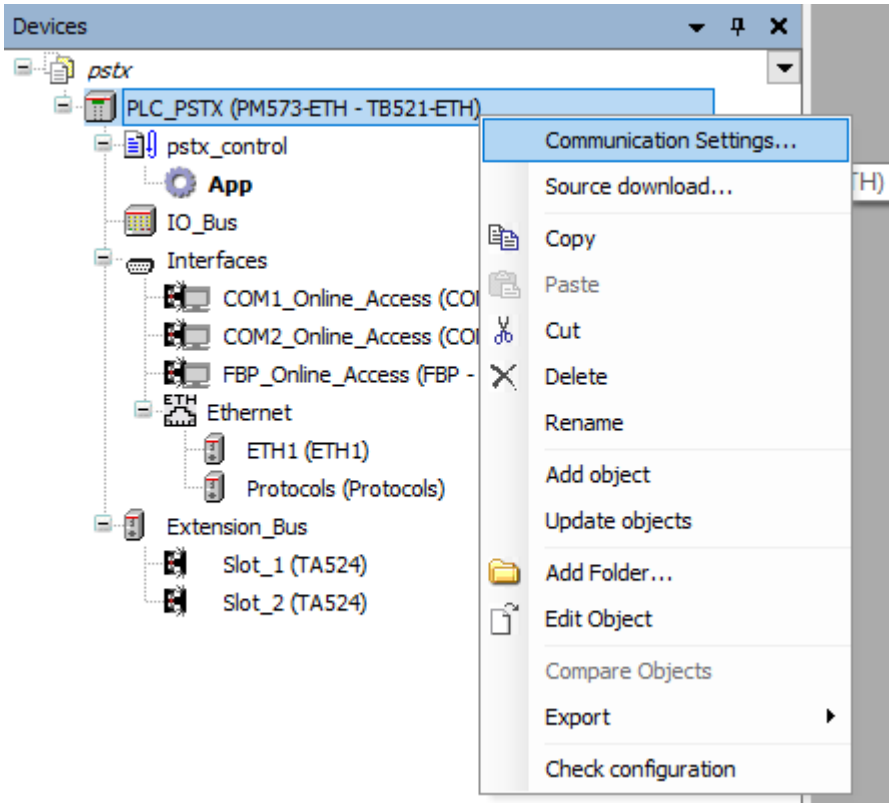


4. Check that the correct device type is selected by double clicking the device name in Devices field. Check that the correct Terminal Base Type is also selected for the tag for Hardware.



6.2. Connect to PLC using TCP/IP

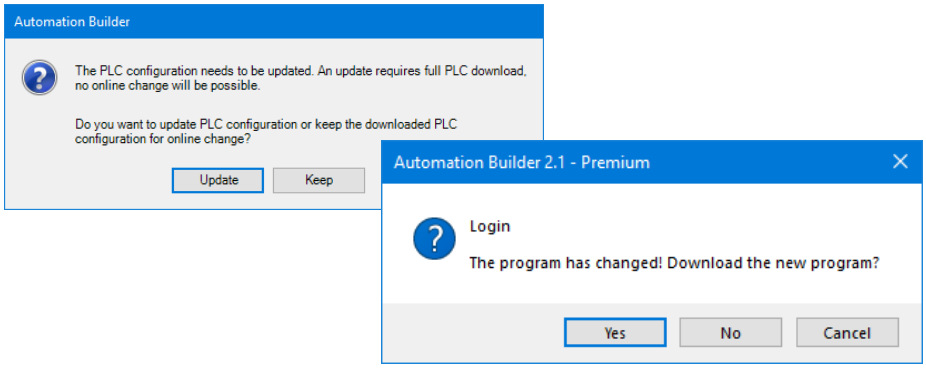
1. Control the IP address for the device is also correct by right click the device name and then chose communication setting. The IP address should be the address of PLC CPU device.



2. Control that the IP address for PC and the PLC is in the same network but not the same IP address. This can be checked by using through Ethernet Properties.

3. Click the icon “Login”, for building the configuration and checking if configuration is correct.

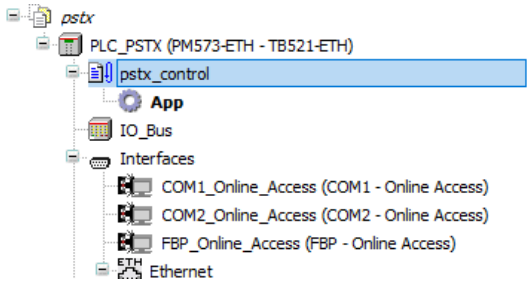
4. If the configuration is correct, a program for building PLC should be opened in the PLC environment, CoDeSys. Automation Builder will ask for downloading PLC configuration. Choose “Update”. Automation Builder will confirm that the program has changed.



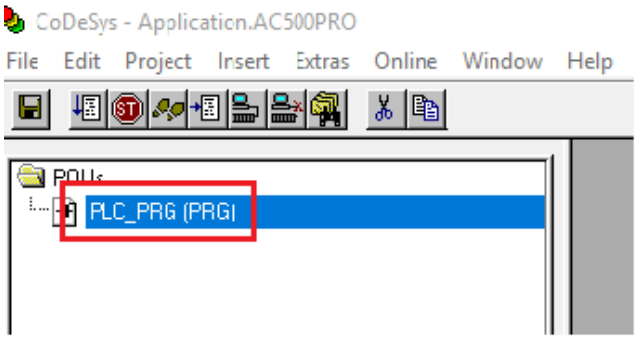
6.3. Build a START-STOP program

We perform the following steps for building our start-stop demo program in CoDeSys.

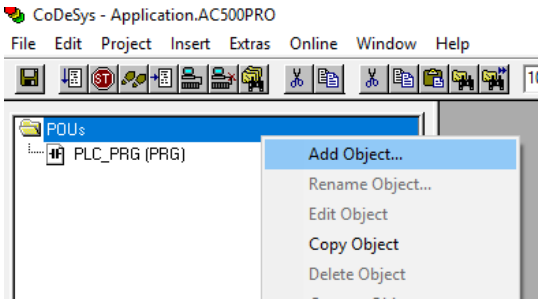
1. Open CoDeSys by double clicking your application in Devices file in Automation Builder, if it is not opened yet.

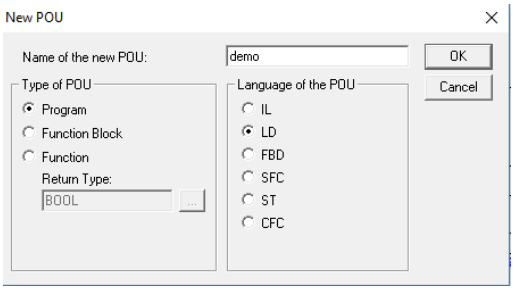


2. Open program window by double clicking the default program in POUs in CoDeSys.

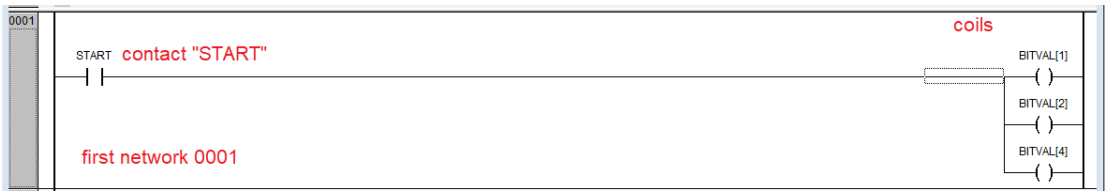


3. We choose to use LD as the language of the POU here by right click POUs -> Add Object…->Insert Name of the new POU -> Choose “LD” for “Language of the POU” -> OK.

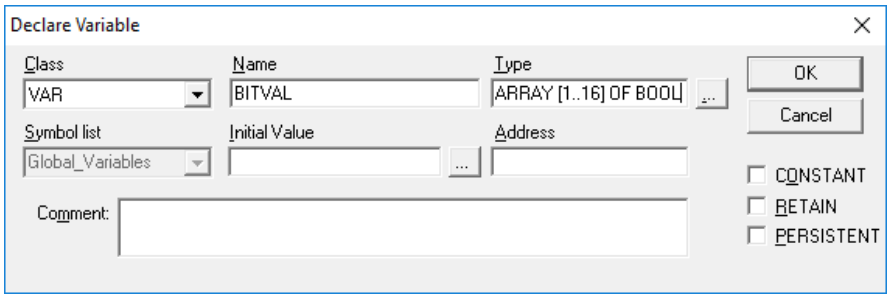




4. Select the first network, create a contact “START” (by CTRL+K and putting name at “???”) and three coils “BITVAL[1]”, “BITVAL[2]” and “BITVAL[4]” (by CTRL+L) in first network. We set BITVAL 1, 2 and 4 because we want to set TRUE for “Start”, “Stop” and “Auto mode” according Section 3.5 for data transmission.

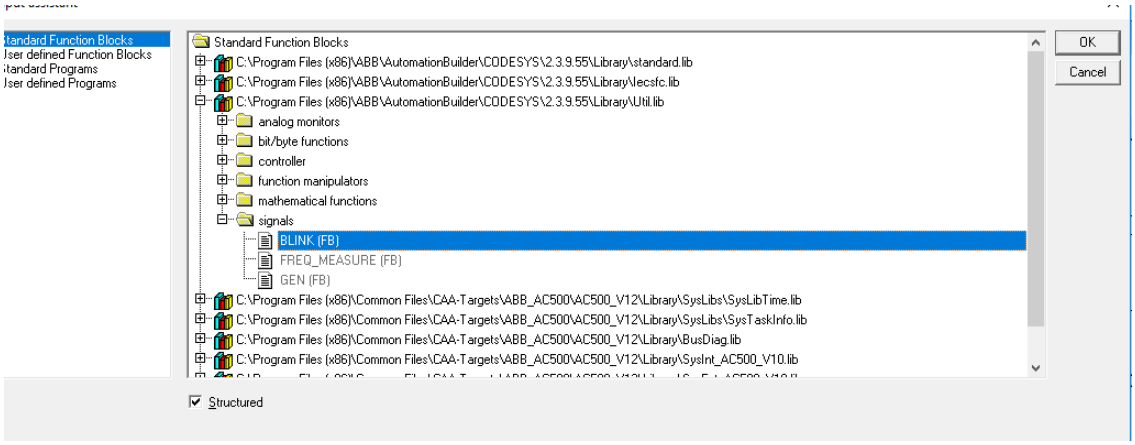


5. Select the data type for BITVAL as “ARRAY [1..16] OF BOOL” in “Declare Variable” window. We choose BITVAL as an array of 16 because fieldbus task has 16 bits according to Section 5.

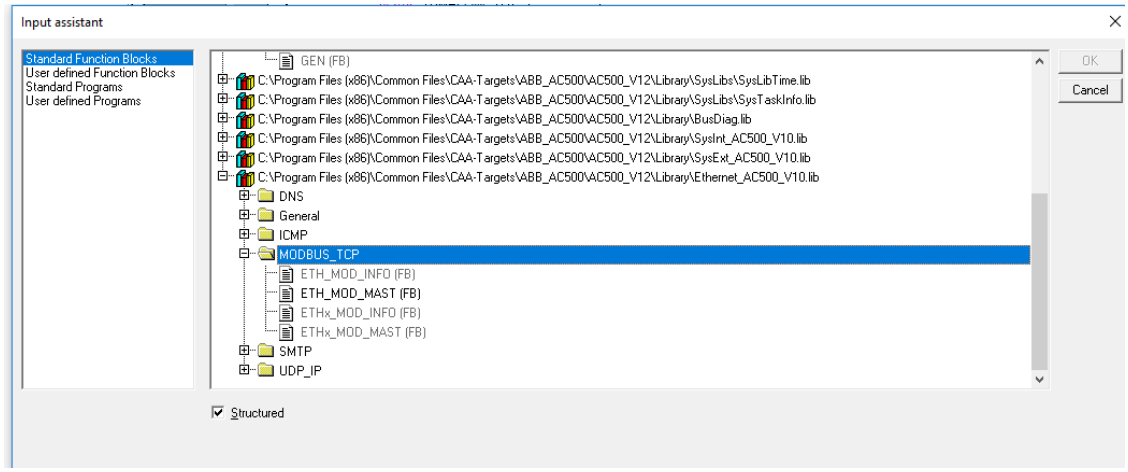


6. Create a second network by CTRL+T

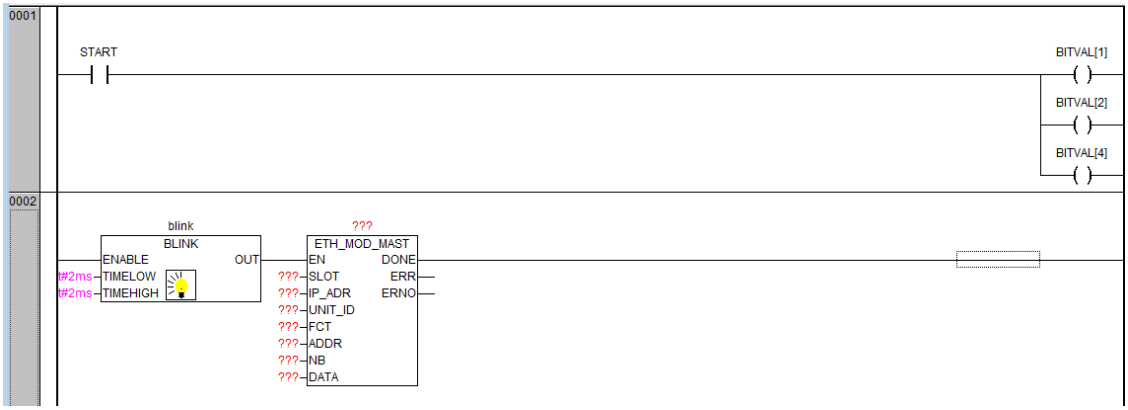
7. We continue create a function block “BLINK” by CTRL+B and select Standard Function Blocks -> Util.lib -> signals-> BLINK(FB)->Ok. We name the BLINK function block as blink. We set t#2ms for TIMELOW and TIMEHIGH.



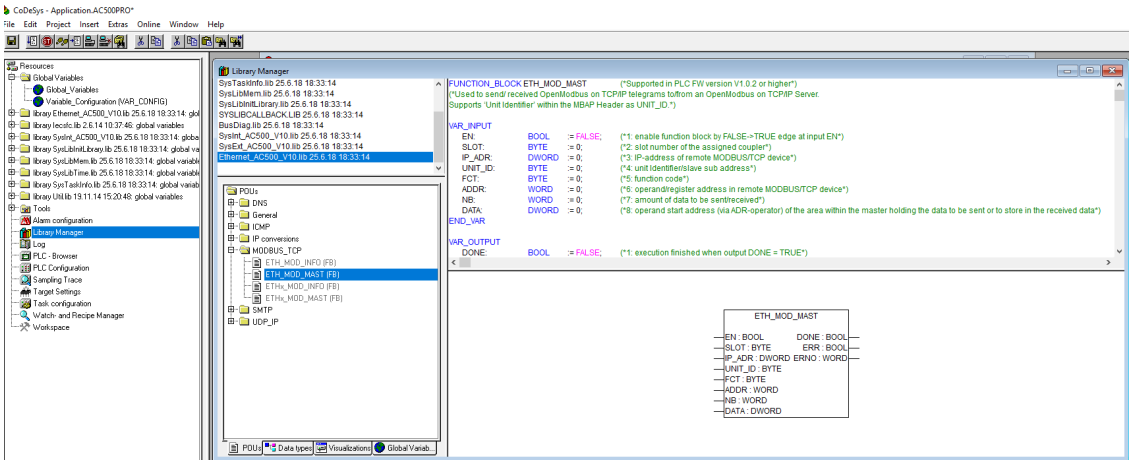
8. We continue create a function block “ETH_MOD_MAST” by CTRL+B and select Standard Function Blocks -> Ethernet_AC500_V10.lib -> MODBUS_TCP-> ETH_MOD_MAST(FB) -> OK.



Now, we should have two function blocks in network 0002.



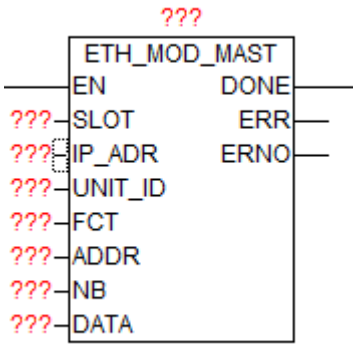
The ETH_MOD_MAST is a function block for sending/receiving OpenModbus on TCP/IP telegrams to/from an OpenModbus on TCP/IP Server. The function has inputs: EN, SLOT, IP_ADR, UNIT_ID, FCT, ADDR, NB and DATA. Their definition is available from CoDeSys -> Resources -> Library Manager.



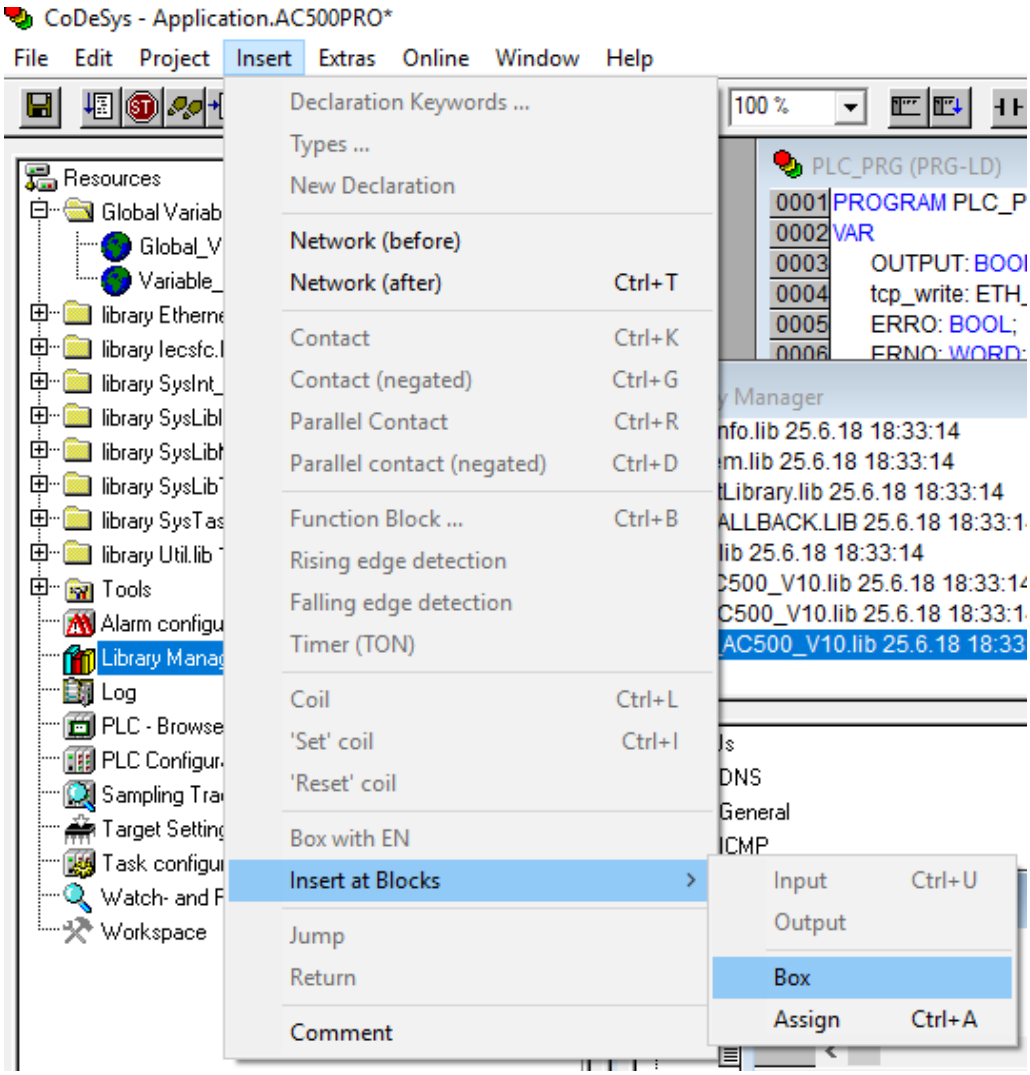
To enable this function block, it is required to send a FALSE->TRUE edge at input EN and therefore we introduce BLINK, which is for creating a flip-flop signal.

9. Both IP_ADR (IP-address of MODBUS TCP/IP device) and DATA (the data to send) require DWORD inputs. We can convert our IP-address with a box, “IP_ADR_STRING_TO_DWORD”, and convert data with a box, “ADR”.

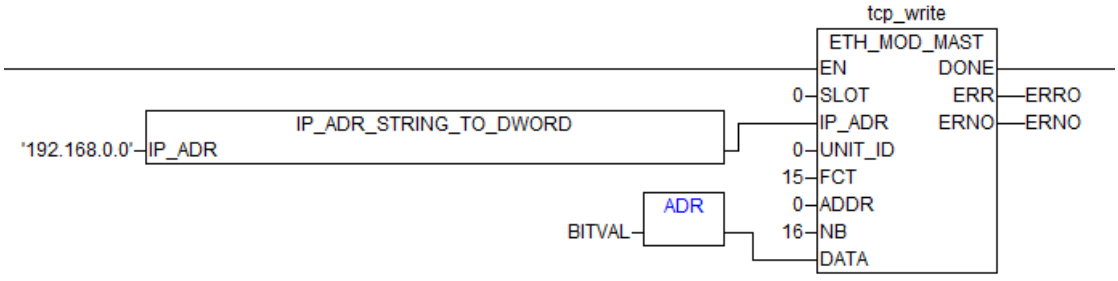
   a) Select the bar in front of IP_ADR



b) Choose Insert-> Insert at Blocks-> Box



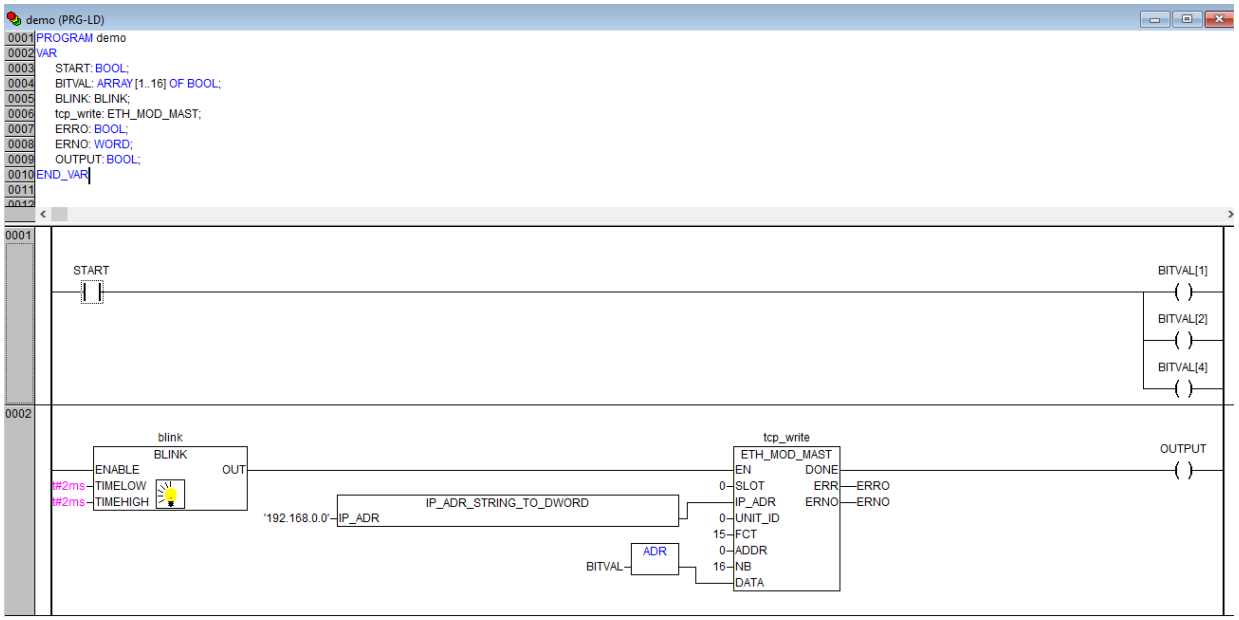
The default box is “AND”, and we insert “IP_ADR_STRING_TO_DWORD” at the blue marked area, then press ENTER. We can also find this box by pressing F2 when the marked area is selected. We insert IP-address in front of IP_ADR of “IP_ADR_STRING_TO_DWORD”.



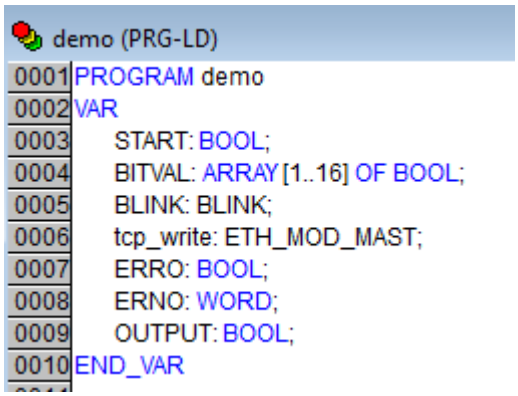
10. Set:

- The name of the block to tcp_write
- SLOT to 0
- IP_ADR to IP address of MODBUS/TCP device
- UNIT_ID to 0
- FCT to 15
- ADDR to 0, according to Section 3.5, the first Protocol Address is 0000h.
- NB to 16
- ERR to ERRO (new BOOL variable)
- ERRNO (new WORD variable)

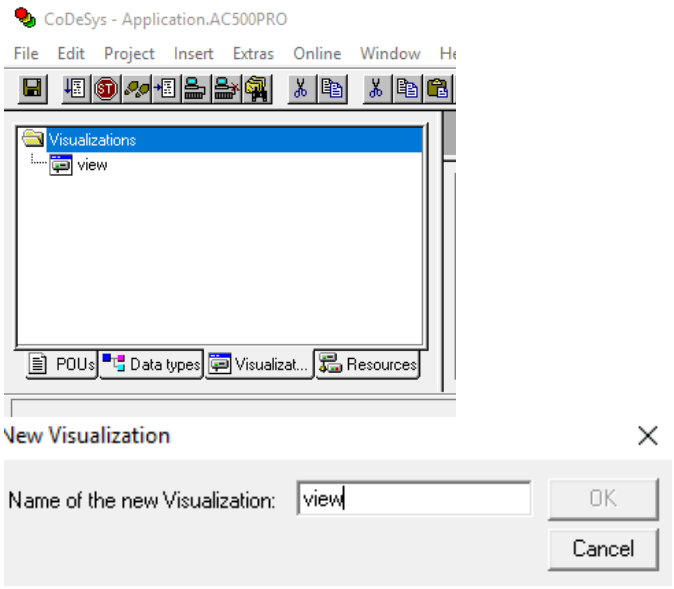
11. Insert a coil named ”OUTPUT” in the second network. Thus we have complete two networks here.



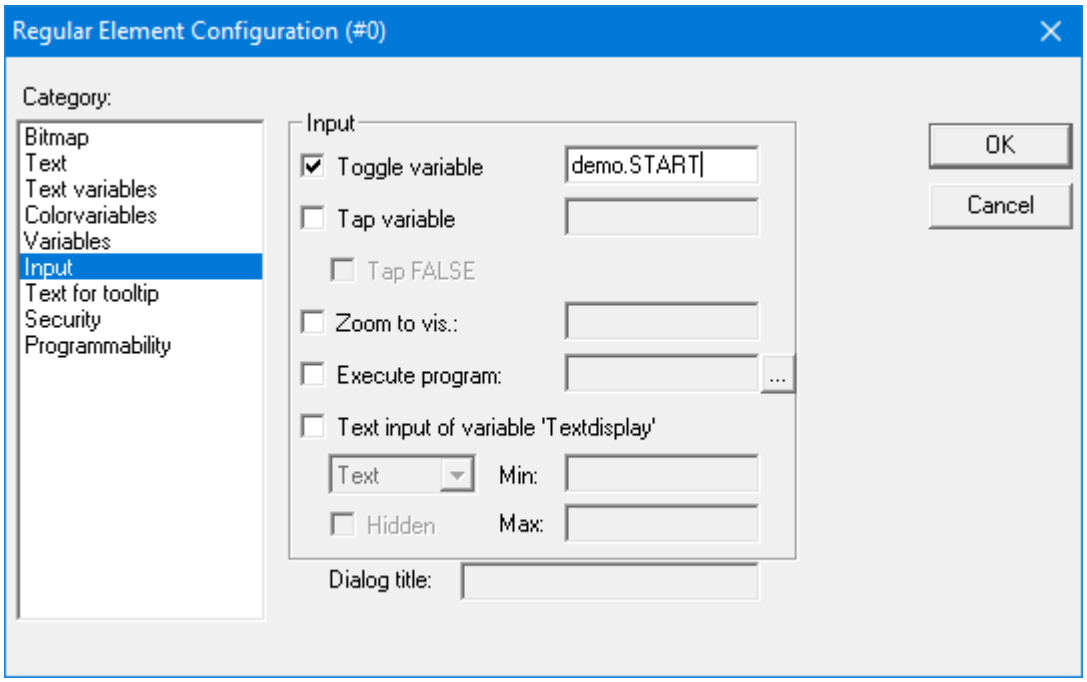
12. Control that we have correct data type for all the variables we created inside networks.



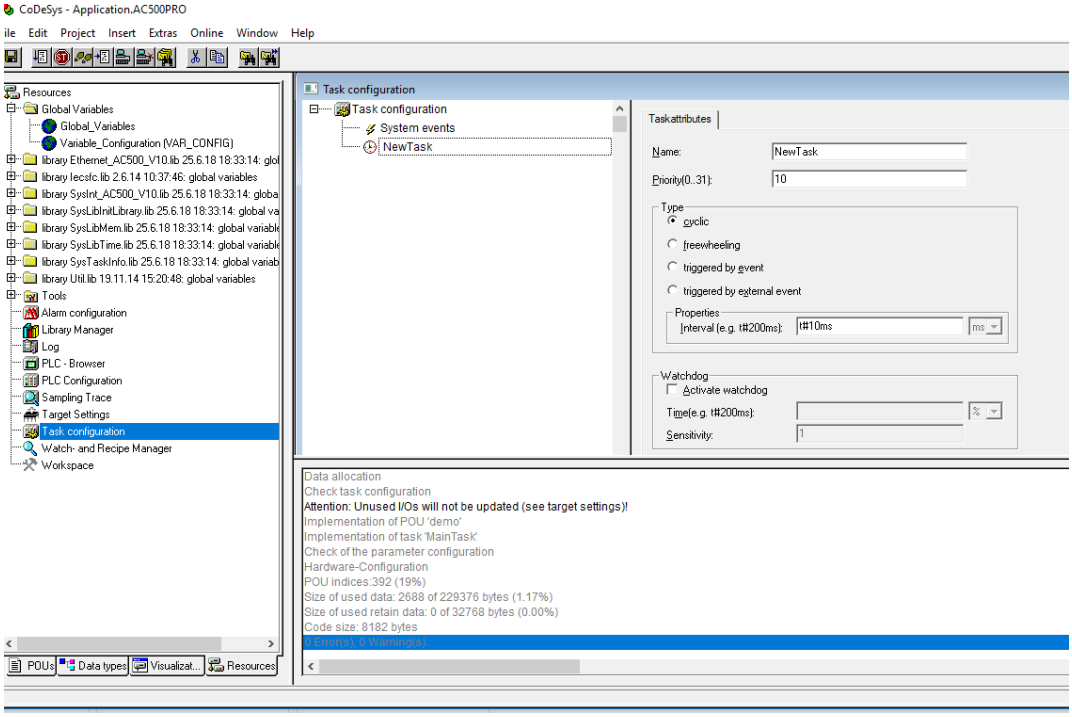
13. Now we want to create one control button for signing the value of “START” from the first network into TRUE. We do this by Visualization -> right click -> Add object -> Write name of the new Visualization as “view” -> OK



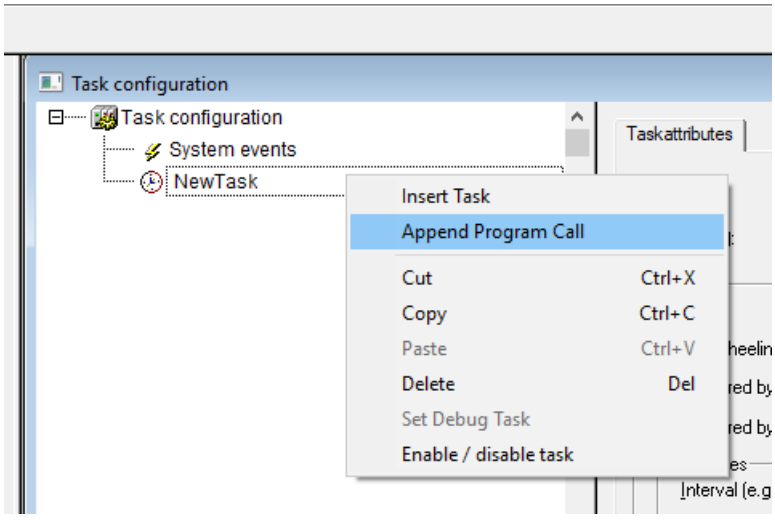
14. We draw a shape as the button  -> double click the shape -> Regular Element Configuration -> Input -> check Toggle variable -> insert “demo.START” ->OK



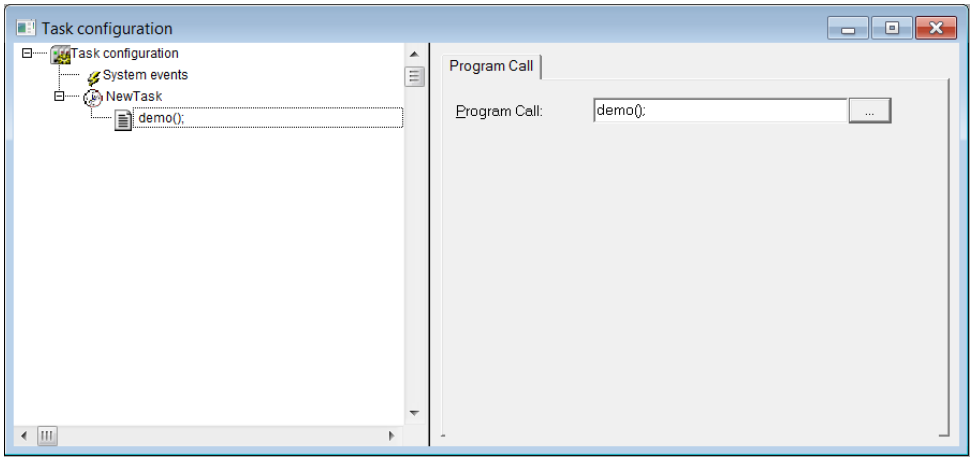
15. We configure this program into task configuration by Resource -> Task configuration -> Right click Task configuration -> Append Task -> Insert t#10ms in Properties in Taskattributes. Then we need to sign our program to this task by right click NewTask-> Append Program Call-> Choose demo(PRG) by clicking the select button in Program Call ->OK.



16. Right click on the NewTask and select Append Program Call.

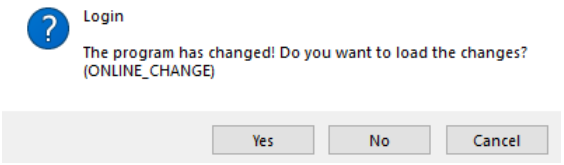


17. Select the demo program call

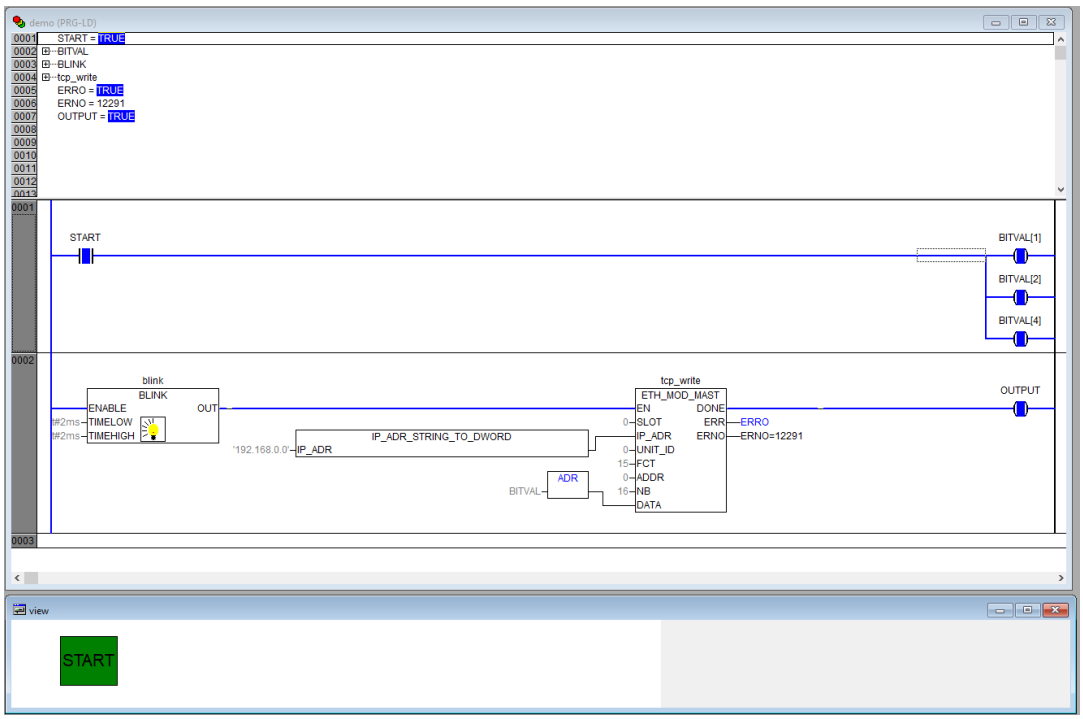


6.4. Build and run the PLC demo program

Use the key, F11, to build the program once. Login and start project from Automation Builder by clicking Alt+F8 to login the CodeSys. Click yes to login



Click F5 to start. Switch to CoDeSys and click Alt+F8 to login demo. The program can be controlled with the view from CodeSys



7. Contact us

For more information, please contact
your local ABB representative or visit
https://new.abb.com/drives/softstarters