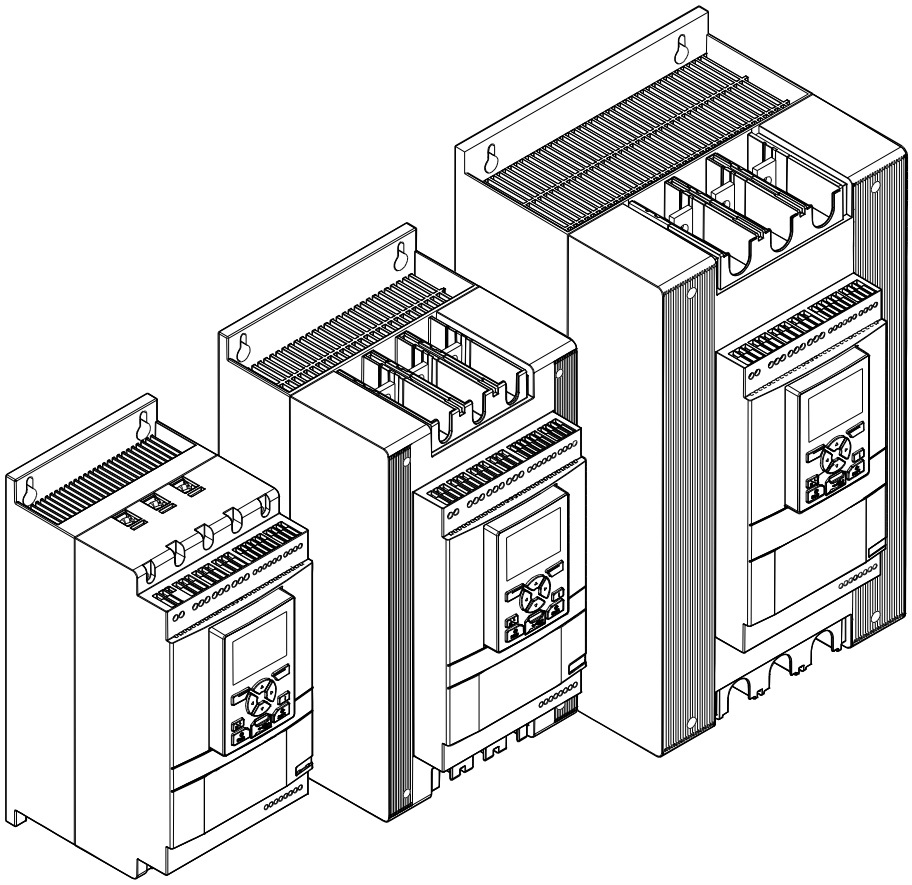
FIELDBUS COMMUNICATION MANUAL

Softstarters Type PSTX

Anybus CompactCom EtherCAT



PREPARED		STATUS	SECURITY LEVEL		
2019-07-09	Theodor Lindberg	Approved	Public		
APPROVED		DOCUMENT KIND			
2019-07-09	Aravindh Vishnu	Manual			
OWNING ORGANIZATION		DOCUMENT ID.	REV.	LANG.	PAGE
ABB AB		1SFC132382M0201	A	en	1/27

Contents

1. EtherCAT

EtherCAT is a Real Time Ethernet technology which aims to maximize the use of the full duplex Ethernet bandwidth. It overcomes the overhead normally associated with Ethernet by employing “on the fly” processing hardware. An EtherCAT bus consists of a master system and up to 65535 slave devices, connected with standard Ethernet cabling. The slave devices process the incoming Ethernet frames directly, extract or insert relevant data and transfer the frame to the next EtherCAT slave device. The last slave device in the bus segment sends the fully processed frame back to the master.

The EtherCAT protocol provides full control and status information of the softstarter, reading as well as writing of parameters. 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 SFC132081M0201 for fieldbus related settings.

Before the EtherCAT communication can be taken in operation following parameters must be set in the softstarter:

- Parameter 12.02 *FB interface connector* set to **Anybus**
- Parameter 12.03 *Fieldbus control* set to **On** (if using fieldbus only to monitor, this parameter can be set to **Off**)

Optionally, the following parameter may be set to configure a Station Alias for the device:

- Parameter 12.04 *Fieldbus address* set to the station alias for this slave device

Note: Fieldbus address (parameter 12.04) should be kept at its default value of 0 if a station alias should not be set by the slave device.

Note: 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 communication 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".

To do the programming of the PLC, the following files are available:

GSDML file	Type of file
EtherCAT Anybus M40 v2.1 PSTX Softstarter v1.0.xml	EtherCAT slave information file

Note: 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.

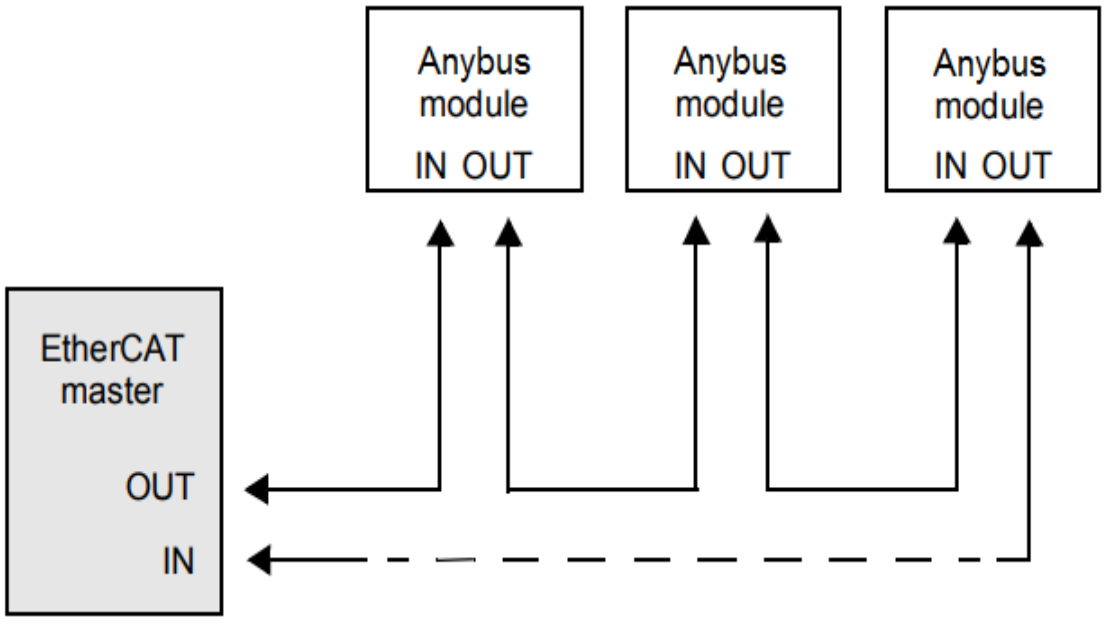
- ***Switching from one type of control to another (fieldbus control/hardwire control)***
- ***Reset all Settings***

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.

1.1. Example topology of the EtherCAT link

A topology example is shown below.



1.2. Digital Inputs

To PLC from softstarter.

Index	Subindex	Name	Bit	Data	Description
2003 h	**1**	**Digital Input Byte1**	**0**	Auto Mode status[1]	0 = Softstarter control through fieldbus communication not allowed 1 = Softstarter control through fieldbus communication allowed
			1	Event status	0 = No active fault/warning/protection 1 = Active fault/warning/protection
			2	Ready To Start	0 = A start will probably cause a fault 1 = A start will not cause a fault
			3	FBT Response 0	See section 2 Fieldbus Tasks
			4	FBT Response 1	See section 2 Fieldbus Tasks
			5	FBT Toggle Bit	See section 2 Fieldbus Tasks
			6	Programmable Digital Input 1	Function of programmable digital input, see section 1.3
			7	Programmable Digital Input 2	
	2	**Digital Input Byte2**	**8 (0)**	Programmable Digital Input 3	
			9 (1)	Programmable Digital Input 4	
			10 (2)	Programmable Digital Input 5	
			11 (3)	Programmable Digital Input 6	
			12 (4)	Programmable Digital Input 7	
			13 (5)	Programmable Digital Input 8	
			14 (6)	Programmable Digital Input 9	
			15 (7)	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’.
- The digital input ‘Fieldbus disable’.

1.3. 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.
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.

Function	Data
Pump cleaning backward status	Status of reverse pump cleaning.
External digital 1DI0 status	Status of external digital input 1DI0
External digital 1DI1 status	Status of external digital input 1DI1
External digital 1DI2 status	Status of external digital input 1DI2
External digital 1DI3 status	Status of external digital input 1DI3
External digital 1DI4 status	Status of external digital input 1DI4
External digital 2DI5 status	Status of external digital input 2DI5
External digital 2DI6 status	Status of external digital input 2DI6
External digital 2DI7 status	Status of external digital input 2DI7
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.

1.4. Analog Inputs

To PLC from the softstarter.

All analog data is represented as 16-bit values.

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

Index	Subindex	Name	Data	Representation
2004 h	**1**	**AnalogInputWord1**	FBT Return Value	See section 2 Fieldbus Tasks
	2	**AnalogInputWord2**	Programmable Analog Input 1	Function of programmable analog input, see section 1.5
	3	**AnalogInputWord3**	Programmable Analog Input 2	
	4	**AnalogInputWord4**	Programmable Analog Input 3	
	5	**AnalogInputWord5**	Programmable Analog Input 4	
	6	**AnalogInputWord6**	Programmable Analog Input 5	
	7	**AnalogInputWord7**	Programmable Analog Input 6	
	8	**AnalogInputWord8**	Programmable Analog Input 7	
	9	**AnalogInputWord9**	Programmable Analog Input 8	
	10	**AnalogInputWord10**	Programmable Analog Input 9	
	11	**AnalogInputWord11**	Programmable Analog Input 10	

1.5. 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 current[1]	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
Reactive energy (resettable)	Value = 1000 ⇒ 10kVArh
Reactive power	Value = 1000 ⇒ 100kVAr
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

Function	Representation
	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.

1.6. Digital Outputs

From PLC to the softstarter.

Index	Subindex	Name	Bit	Data	Description
2001h	**1**	**DigitalOutputByte1**	**0**	Start	Commence a start when signal is set.
			1	Stop	Commence a stop when signal is negated.
			2	Fault reset	Reset signal for possible events.
			3	Auto mode	This must be set for controlling the motor.
			4	Slow speed reverse	Perform slow speed reverse when signal is set.
			5	Slow speed forward	Perform slow speed when signal is set.
			6	Spare	
			7	Start1	Start1 if sequence start.
	2	**DigitalOutputByte2**	**8 (0)**	Start2	Start2 if sequence start.
			9 (1)	Start3	Start3 if sequence start.
			10 (2)	Motor heating	Perform motor heating when signal is set.

			11 (3)	Stand still brake	Perform stand still brake when signal is set.
			12 (4)	Start reverse	Commence a reverse start when signal is set.
			13 (5)	Spare	
			14 (6)	Emergency mode	Set to “1” to enable emergency mode.
			15 (7)	FBT Toggle Bit	See Fieldbus Tasks.
	3	**DigitalOutputByte3**	**16 (0)**	User defined trip	Set to “1” to trigger user defined protection.
			17 (1)	Switch to remote control	Switch to remote control when signal is set (rising edge triggered).
			18 (2)	Pump cleaning automatic	Perform automatic pump cleaning when signal is set.
			19 (3)	Pump cleaning forward	Perform forward pump cleaning when signal is set.
			20 (4)	Pump cleaning reverse	Perform reverse pump cleaning when signal is set.
			21 (5)	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”
			22 (6)	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”
			23 (7)	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”
	4	**DigitalOutputByte4**	**24 (0)**	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”

			25 (1)	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”
			26 (2)	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”
			27 (3)	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”
			28 (4)	Spare	
			29 (5)	Spare	
			30 (6)	Spare	
			31 (7)	Spare	

1.7. Analog Outputs

From PLC to the softstarter.

All analog data is represented as 16-bit values.

Index	Subindex	Name	Data	Representation
2002h	**1**	**AnalogOutputWord1**	FBT Control Word	This register is used to read parameters (see fieldbus tasks).
	2	**AnalogOutputWord2**	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 [%]”.
	3	**AnalogOutputWord3**	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 [%]”.

2. Fieldbus Tasks

By using Fieldbus Tasks it is possible to read/write parameters and to 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.

2.1. 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										

2.2. Task ID

The task identifier controls which function should be performed.

		Response ID	
Task ID	**Task**	**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

2.3. 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.

2.4. 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

2.5. Request parameter value, lower word

This task reads the lower 16 bits of the specified parameter's value. See chapter 2.9 for parameter number and value scaling information.

2.5.1. Arguments

- *FBT Argument 1*: parameter number.

2.5.2. 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.

2.6. Change parameter value

This task writes a specified value to a parameter. See chapter 2.9 for parameter number and value scaling information.

2.6.1. Arguments

- *FBT Argument 1*: parameter number.
- *FBT Argument 2*: parameter value (lower word)
- *FBT Argument 3*: parameter value (upper word)

2.6.2. Return Value

- Response ID 1 on success.
- Response ID 2 and error number in FBT Return Value on failure.

2.7. 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

2.7.1. 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				

2.7.2. 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.

2.8. Request parameter value, upper word

This task reads the upper 16 bits of the specified parameter's value. See chapter 2.9 for parameter number and value scaling information.

2.8.1. Arguments

- *FBT Argument 1*: parameter number.

2.8.2. 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.

2.9. Parameter numbers and values

To access parameters from the fieldbus a unique parameter number is needed, this can be found in document 1SFC132081M0201, Chapter 7.25 Complete parameter list. Since the parameter values need to be represented as integers on the fieldbus, the parameter values with greater precision need to be scaled. In document 1SFC132081M0201, Chapter 7.25 Complete parameter list, 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 $10^{\text{numbers of decimals}}$.
- Parameter 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 to specify the parameter.
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 value set).
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 for *Change parameter value*.
2. Set *FBT Argument 1* to 24 to specify the parameter.
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.

2.9.1. 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.

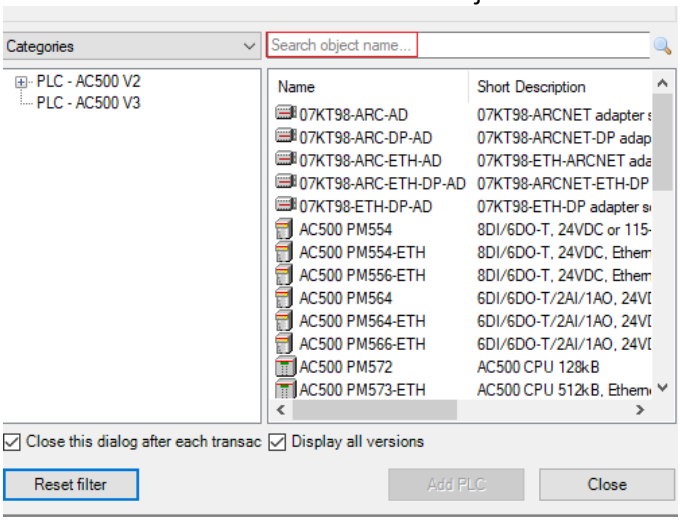
1. Set *FBT Task ID* to 2 for *Change parameter value*.
2. Set *FBT Argument 1* to 249 to specify the parameter.
3. Set *FBT Argument 2* to 65511 to specify the lower word.
4. Set *FBT Argument 3* to 65535 to specify the upper word.
5. Toggle *FBT Toggle Bit* output and wait for the *FBT Toggle Bit*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

3. Configure ABB Automation Builder

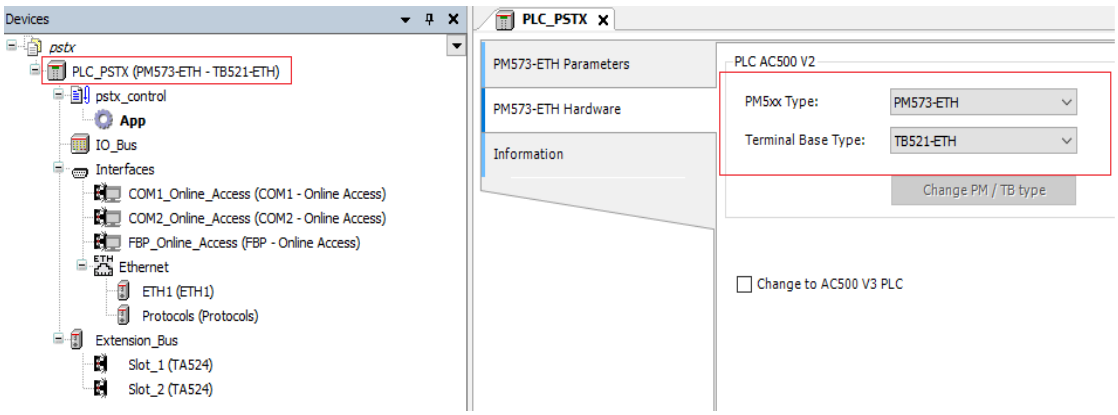
This part of the document describes how to configure an ABB AC500 PLC using the CM579-ETHCAT module to control a PSTX Softstarter with an Anybus EtherCAT module.

3.1. Create a new project

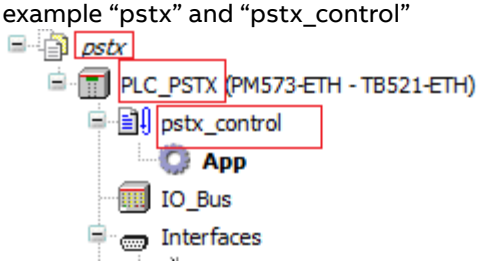
1. Open Automation Builder
2. Select File->New Project->AC500 project->OK
3. Select the correct PLC CPU in Search object name …-> Add PLC



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

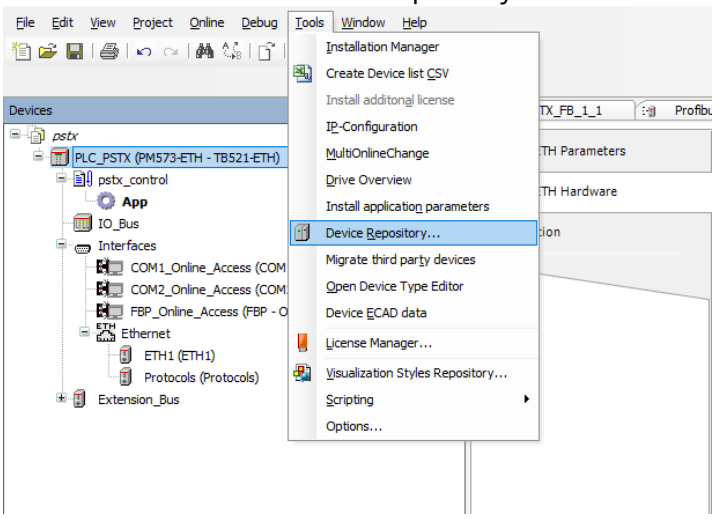


5. Optional: rename the project and the Application to some more suitable names for example “pstx” and “pstx_control”

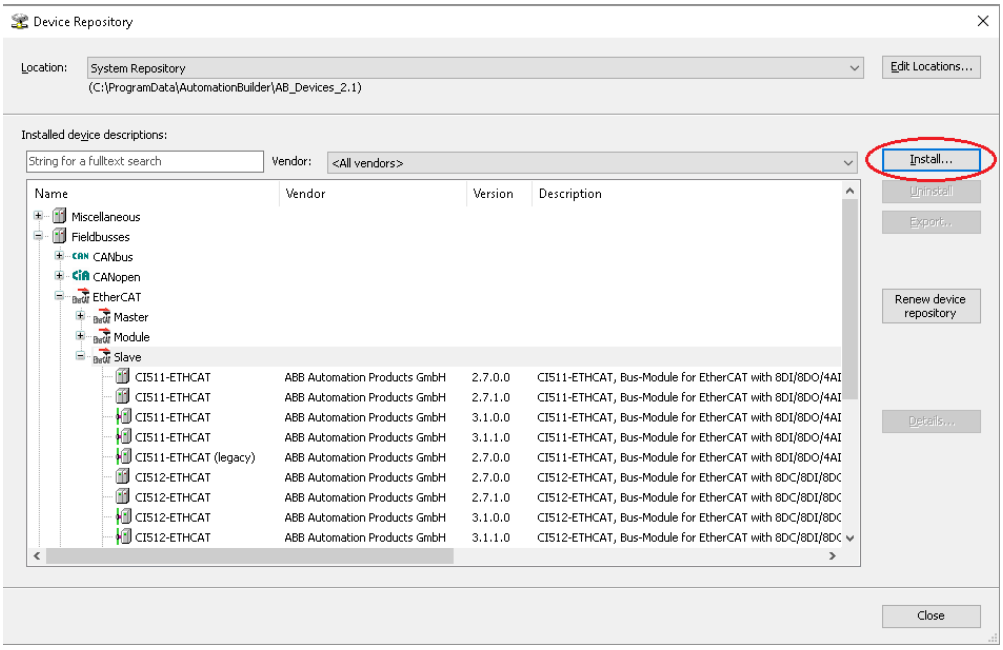


3.2. Install the EtherCAT Slave Information (ESI) file

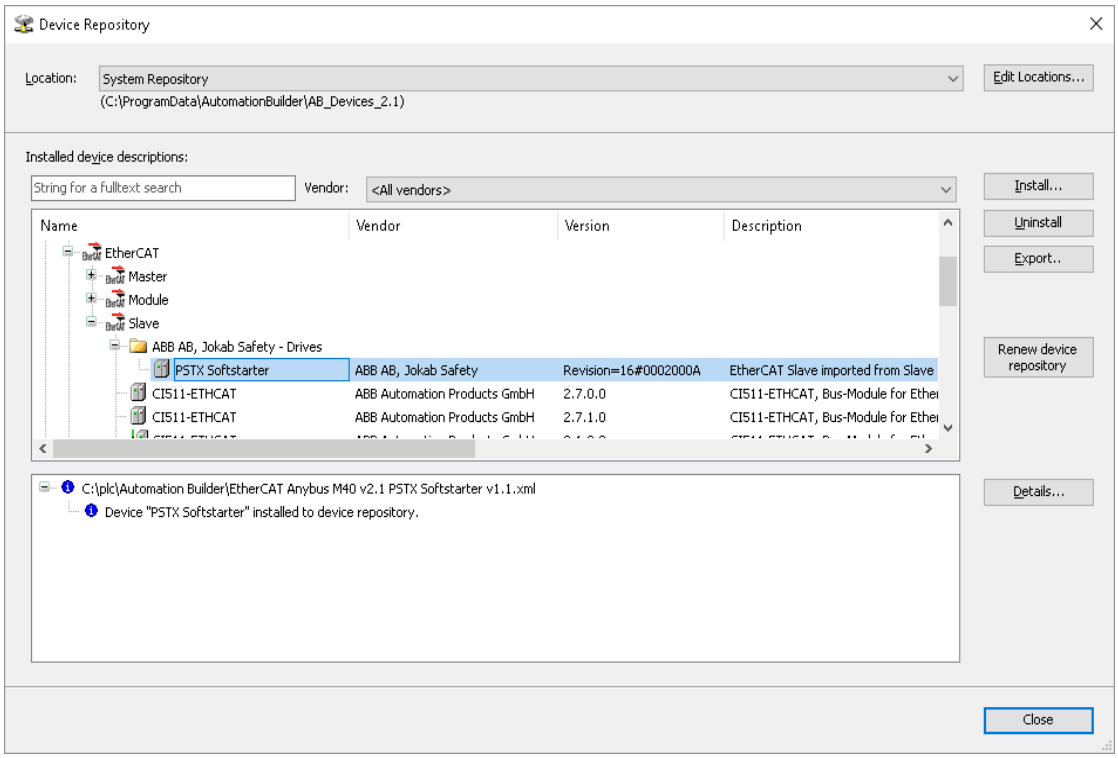
1. In the Tools menu select Device Repository



2. Click Install

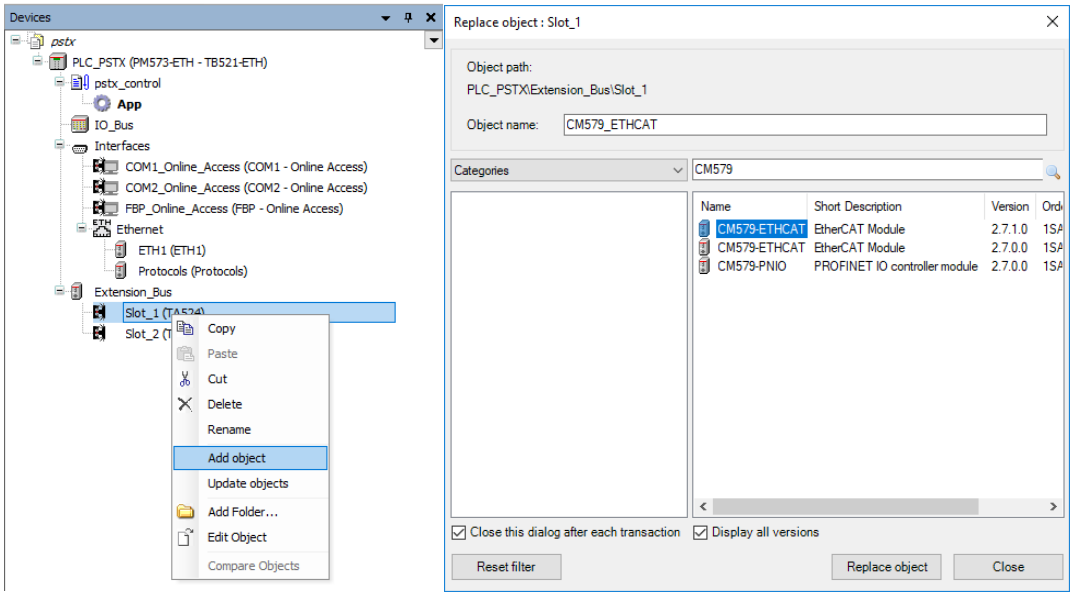


3. Select the EDS-file: EtherCAT Anybus M40 v2.1 PSTX Softstarter v1.1.xml

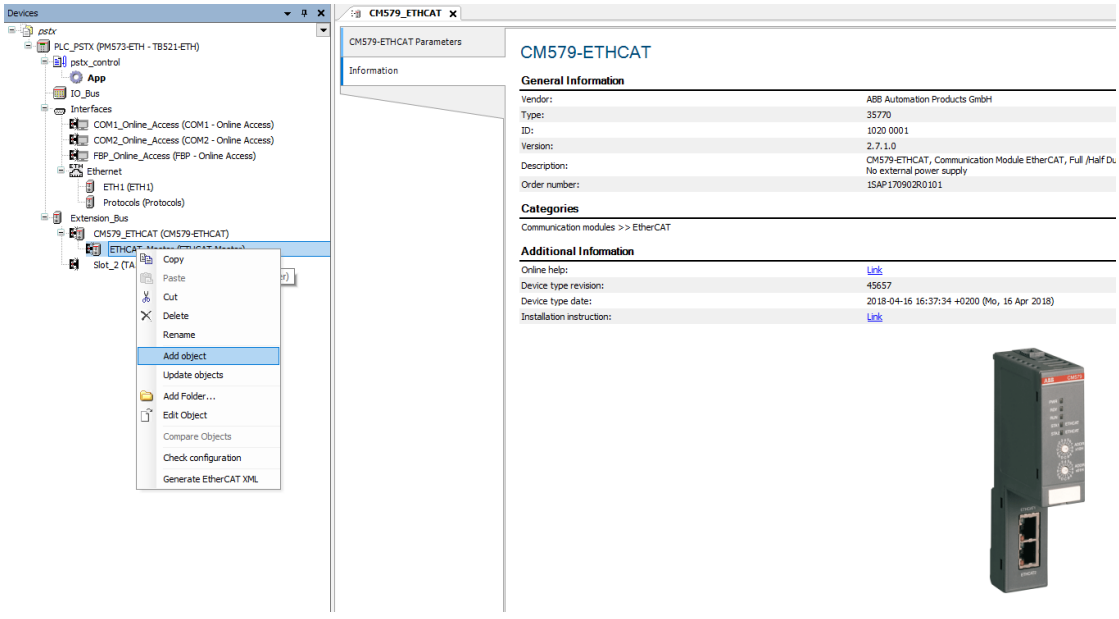


3.3. Add the CM579-ETHCAT EtherCAT master module and the PSTX Slave

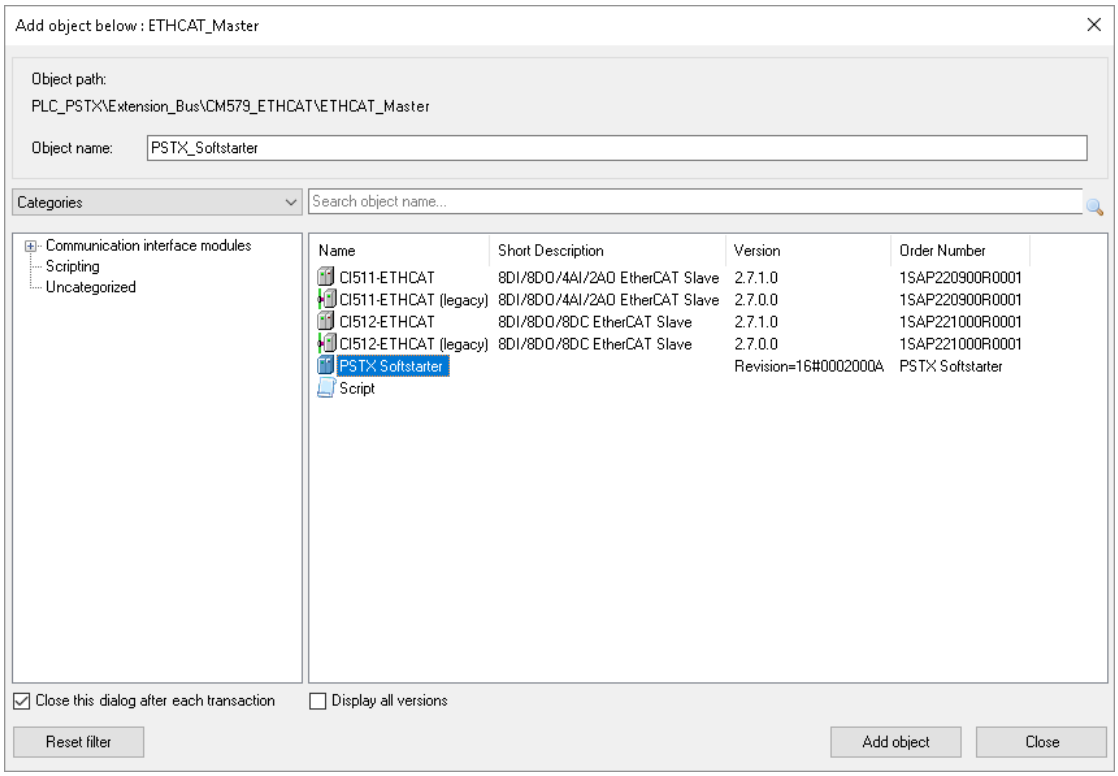
1. Right click on the empty slot where the CM579 module is installed and select Add object. Then select CM579-ETHCAT in the Replace object window. Click “Replace object” to close the Replace object window



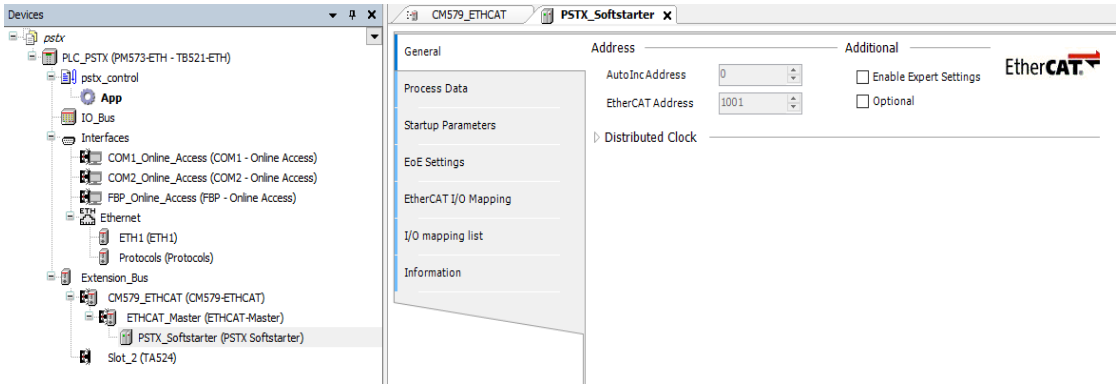
2. Right click on the newly installed ETHCAT_Master and select Add object



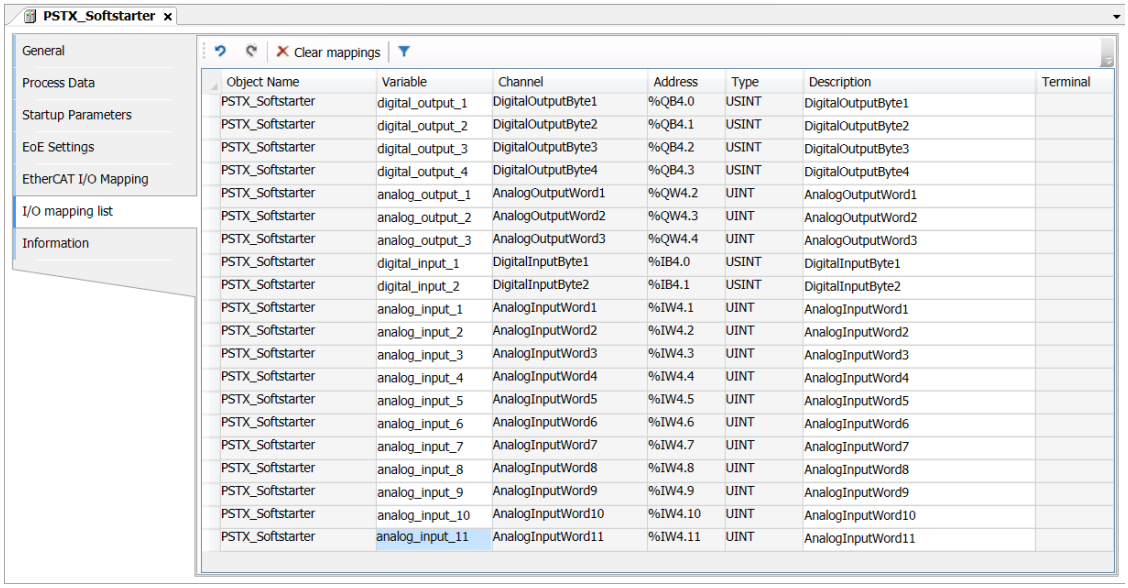
3. Select PSTX Softstarter and click Add object



4. The PSTX Softstarter is now added, the default setting should work



5. Map the signals to variable names

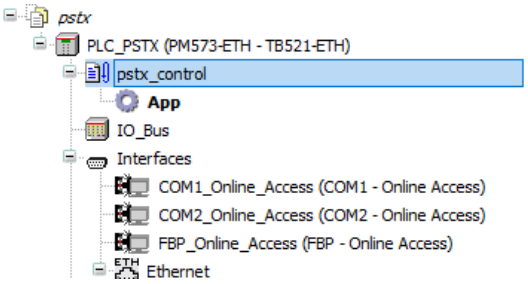


Object Name	Variable	Channel	Address	Type	Description	Terminal
PSTX_Softstarter	digital_output_1	DigitalOutputByte1	%QB4.0	USINT	DigitalOutputByte1	
PSTX_Softstarter	digital_output_2	DigitalOutputByte2	%QB4.1	USINT	DigitalOutputByte2	
PSTX_Softstarter	digital_output_3	DigitalOutputByte3	%QB4.2	USINT	DigitalOutputByte3	
PSTX_Softstarter	digital_output_4	DigitalOutputByte4	%QB4.3	USINT	DigitalOutputByte4	
PSTX_Softstarter	analog_output_1	AnalogOutputWord1	%QW4.2	UINT	AnalogOutputWord1	
PSTX_Softstarter	analog_output_2	AnalogOutputWord2	%QW4.3	UINT	AnalogOutputWord2	
PSTX_Softstarter	analog_output_3	AnalogOutputWord3	%QW4.4	UINT	AnalogOutputWord3	
PSTX_Softstarter	digital_input_1	DigitalInputByte1	%IB4.0	USINT	DigitalInputByte1	
PSTX_Softstarter	digital_input_2	DigitalInputByte2	%IB4.1	USINT	DigitalInputByte2	
PSTX_Softstarter	analog_input_1	AnalogInputWord1	%IW4.1	UINT	AnalogInputWord1	
PSTX_Softstarter	analog_input_2	AnalogInputWord2	%IW4.2	UINT	AnalogInputWord2	
PSTX_Softstarter	analog_input_3	AnalogInputWord3	%IW4.3	UINT	AnalogInputWord3	
PSTX_Softstarter	analog_input_4	AnalogInputWord4	%IW4.4	UINT	AnalogInputWord4	
PSTX_Softstarter	analog_input_5	AnalogInputWord5	%IW4.5	UINT	AnalogInputWord5	
PSTX_Softstarter	analog_input_6	AnalogInputWord6	%IW4.6	UINT	AnalogInputWord6	
PSTX_Softstarter	analog_input_7	AnalogInputWord7	%IW4.7	UINT	AnalogInputWord7	
PSTX_Softstarter	analog_input_8	AnalogInputWord8	%IW4.8	UINT	AnalogInputWord8	
PSTX_Softstarter	analog_input_9	AnalogInputWord9	%IW4.9	UINT	AnalogInputWord9	
PSTX_Softstarter	analog_input_10	AnalogInputWord10	%IW4.10	UINT	AnalogInputWord10	
PSTX_Softstarter	analog_input_11	AnalogInputWord11	%IW4.11	UINT	AnalogInputWord11	

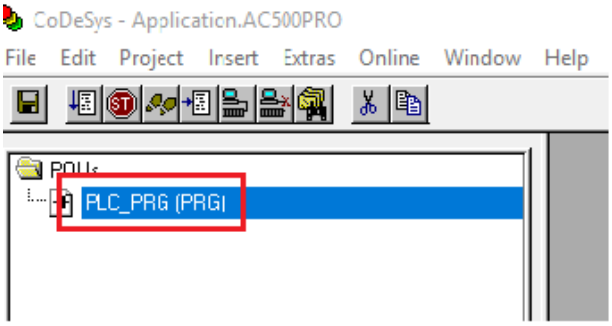
3.4. Write a simple PLC program to control the softstarter

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

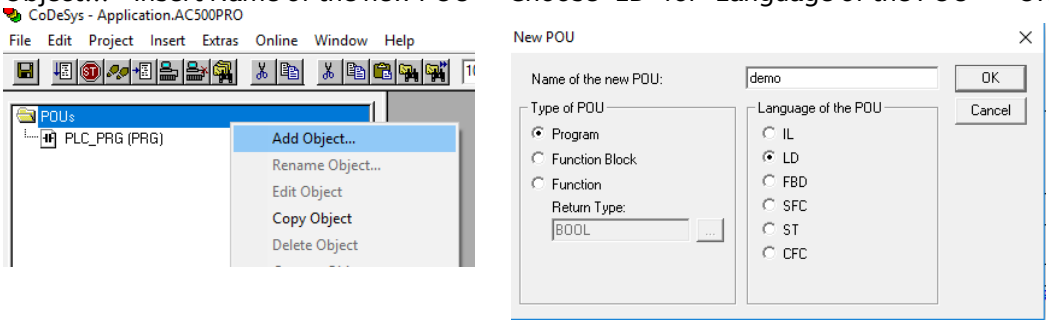
1. Open CoDeSys by double clicking your application in Devices file in Automation Builder, if it is not opened yet.



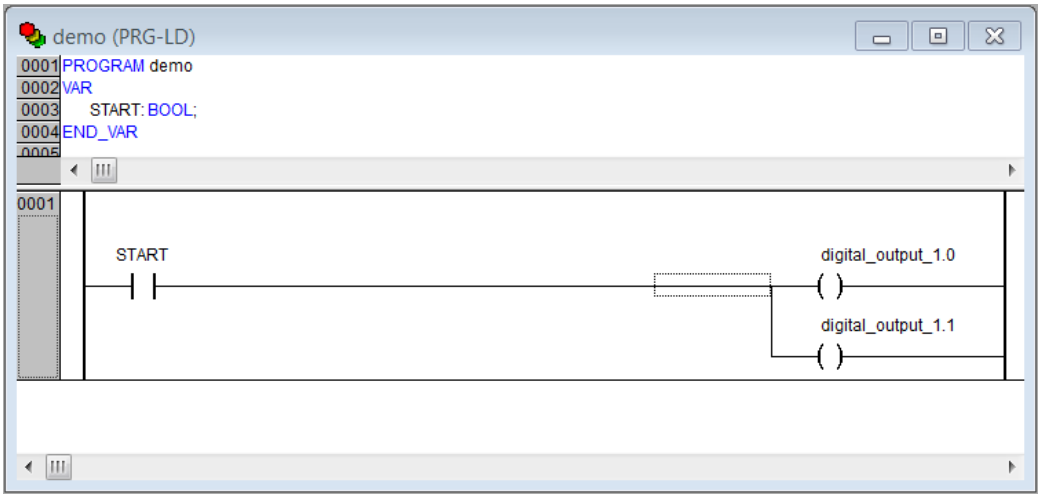
2. Open program window by double clicking the default program in POUs in CoDeSys.



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

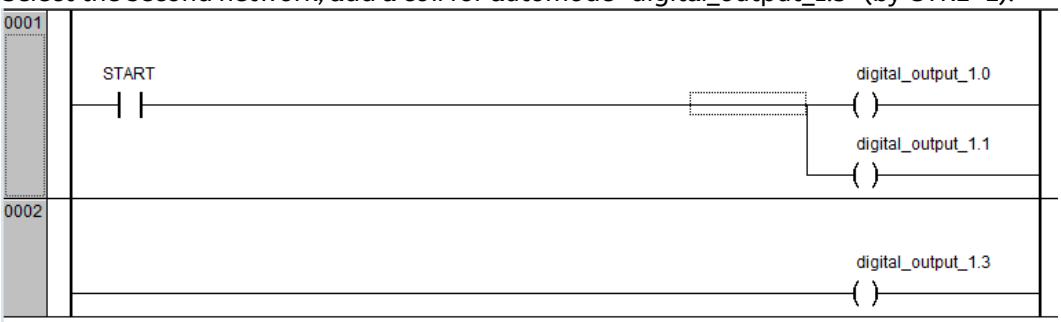


4. Select the first network, create a contact “START” (by CTRL+K and putting name at “???”) and two coils “digital_output_1.0” and “digital_output_1.1” (by CTRL+L) in first network. We let data types as default by clicking OK directly in Declare Variable window. We set digital_output_1 bit 0 and 1 because we want to set TRUE for “Start” and “Stop”, according to Section 1.6. The name digital_output_1 comes from end of Section 3.3 (map signals to variable names).

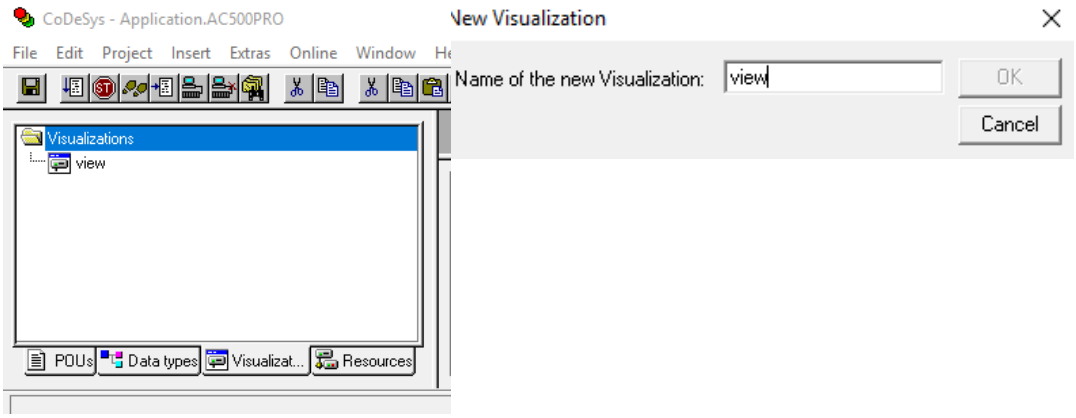


5. Create a second network by CTRL+T

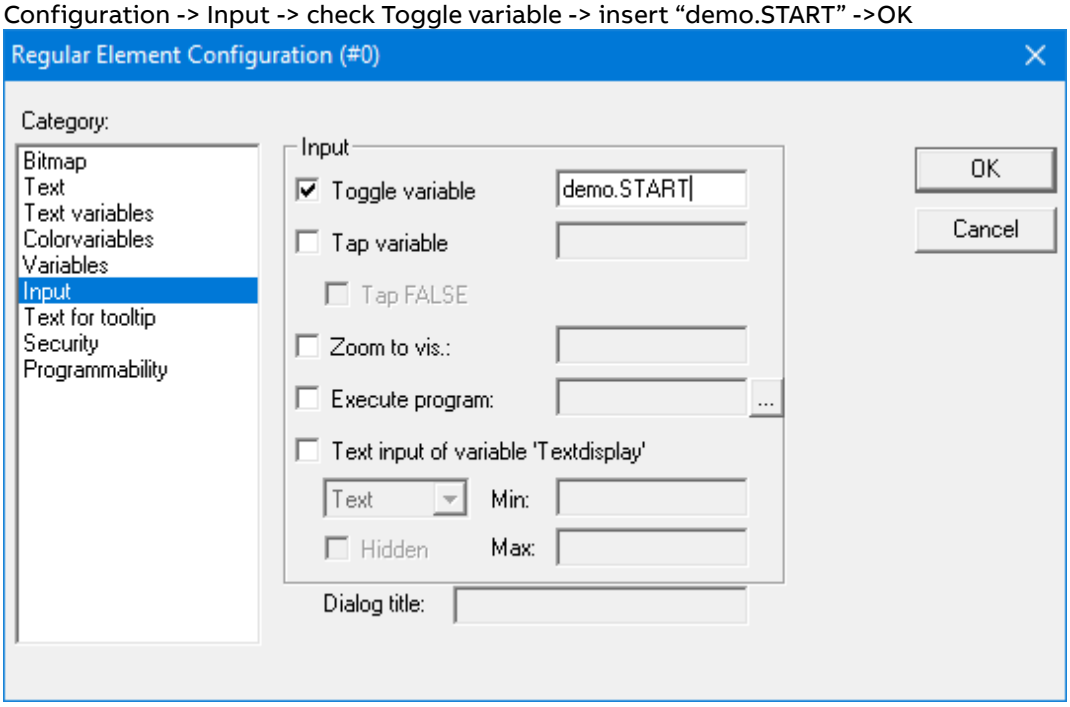
6. Select the second network, add a coil for automode “digital_output_1.3” (by CTRL+L).



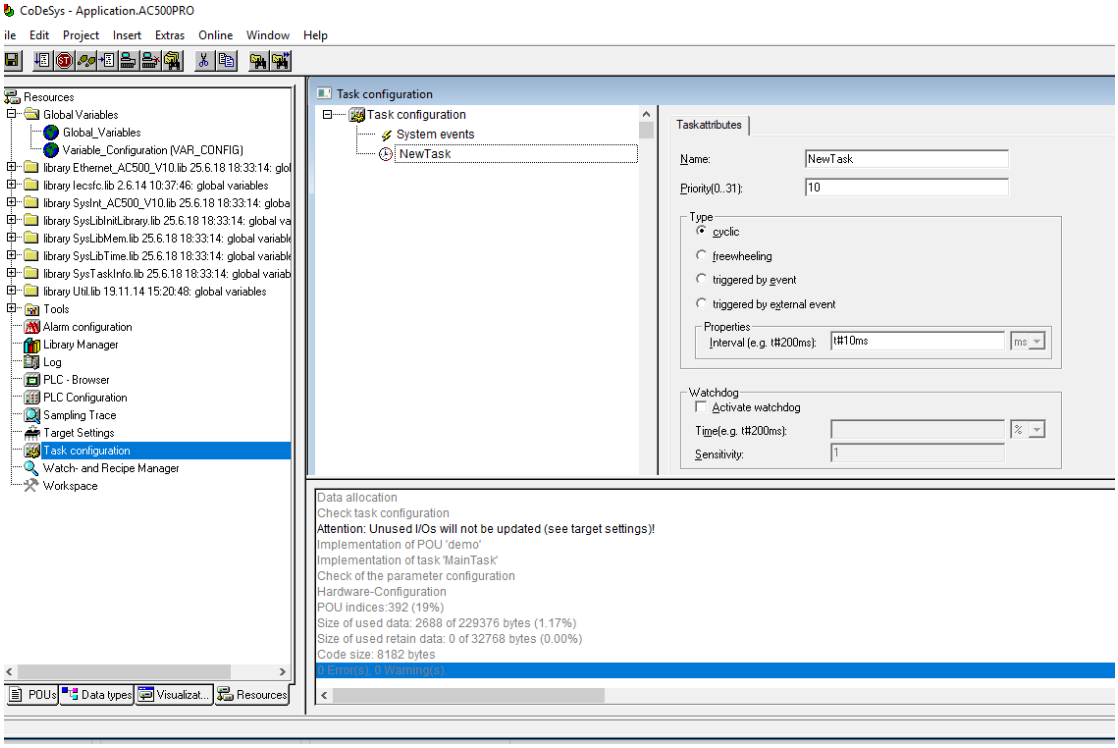
7. Now we want to create one control button for signing the value of “START” from the first network into TRUE. We do this by Visualization -> right click -> Add object -> Write name of the new Visualization as “view” -> OK



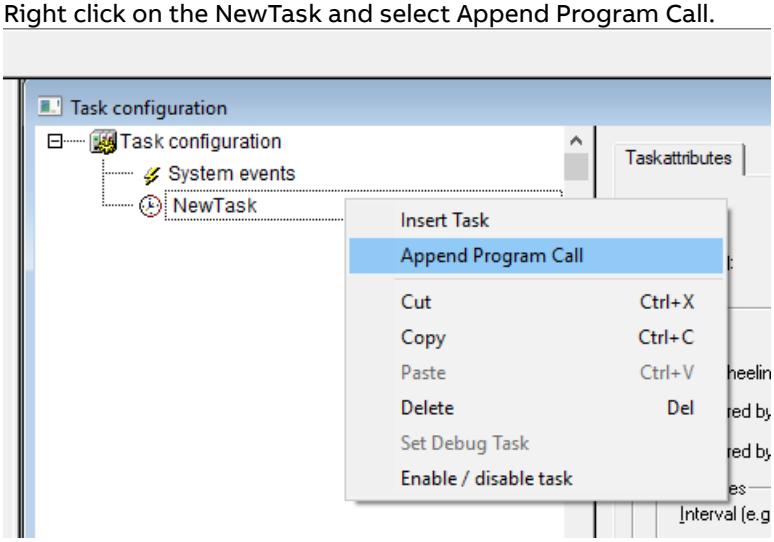
8. We draw a shape as the button  -> double click the shape -> Regular Element Configuration -> Input -> check Toggle variable -> insert “demo.START” ->OK



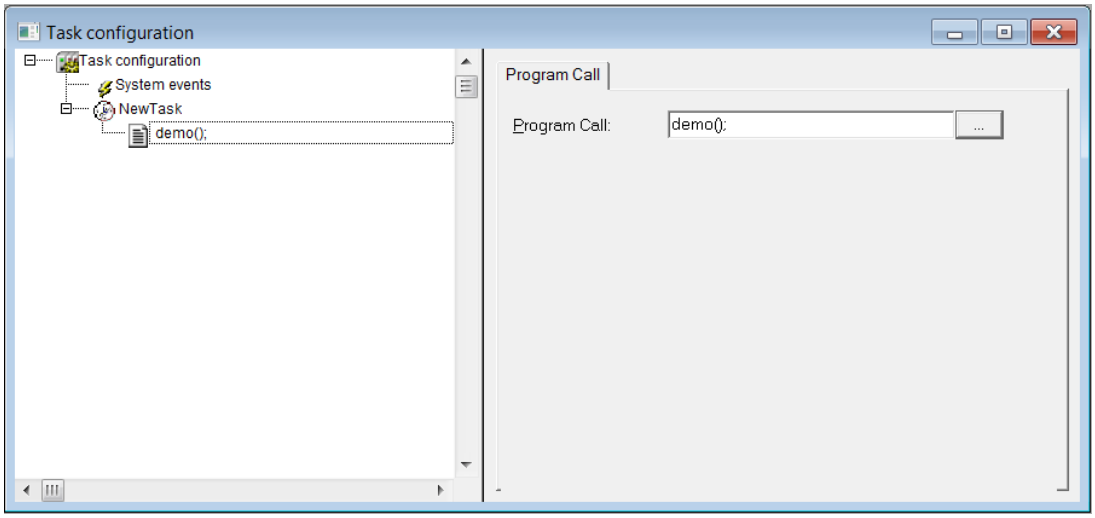
9. We configure this program into task configuration by Resource -> Task configuration -> Right click Task configuration -> Append Task -> Insert t#10ms in Properties in Taskattributes. Then we need to sign our program to this task by right click NewTask-> Append Program Call-> Choose demo(PRG) by clicking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



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

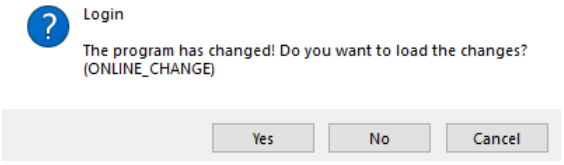


11. Select the demo program call

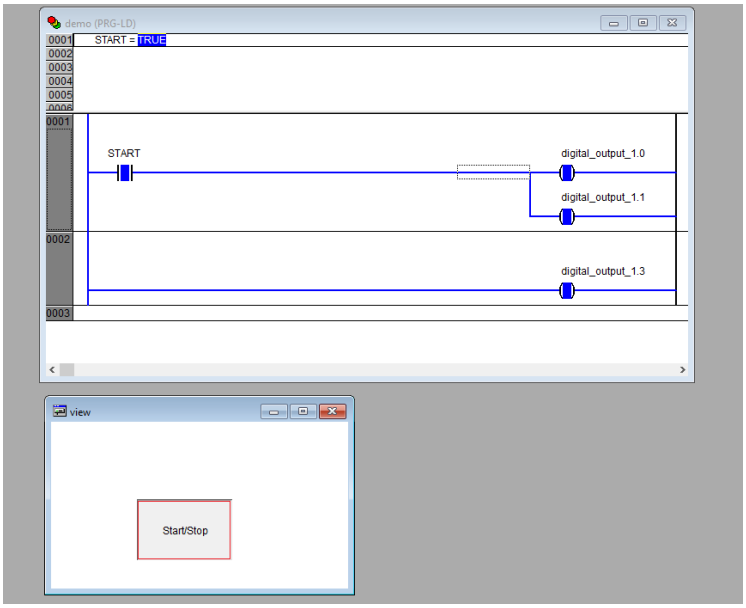


3.5. Build and run the PLC demo program

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



4. Additional Information

4.1. Listing of related documents

Ref #	Document Kind, Title	Document No.
1	PSTX – Installation and commissioning manual	1SFC132081M0201

5. Revisions

Rev.	Page (P) Chapt. (C)	Description	Date Dept./Init.