

SOFTSTARTER TYPE PSTX

Anybus CompactCom PROFINET



PREPARED 2021-03-12	STATUS Approved	SECURITY LEVEL Public		
RELEASED 2021-07-27	DOCUMENT KIND Manual			
OWNING ORGANIZATION Motor Starting & Safety Softstarters	DOCUMENT ID. 1SFC132094M0201	REV. F	LANG. en	PAGE 1/29

© Copyright 2021 ABB. All rights reserved.

Contents

1. PROFINET	4
1.1. Software compatibility	5
1.2. Digital input telegram	5
1.3. Programmable Digital Inputs	7
1.4. Analog input telegram	8
1.5. Programmable Analog Inputs	9
1.6. Digital output telegram	10
1.7. Analog output telegram	13
2. Fieldbus Tasks	13
2.1. FBT Control Word	13
2.2. Task ID	14
2.3. Response ID	14
2.4. Error codes	14
2.5. Request parameter value, lower word	14
2.5.1. Arguments	15
2.5.2. Return Value	15
2.6. Change parameter value	15
2.6.1. Arguments	15
2.6.2. Return Value	15
2.7. Set date and time	15
2.7.1. Arguments	15
2.7.2. Return Value	16
2.8. Request parameter value, upper word	16
2.8.1. Arguments	16
2.8.2. Return Value	16
2.9. Parameter numbers and values	16
2.9.1. Negative values	17
3. Configure SNMP community strings	17
4. Configure ABB Automation Builder	18
4.1. Create a new project	18
4.2. Install the Profinet GSDML-file	19
4.3. Add the CM579-PNIO PROFINET IO controller module	21
4.4. Setup IP-Addresses and station name	21
4.5. Add the PSTX slave	23
4.6. Write a simple PLC program to control the softstarter	24
4.7. Build and run the PLC demo program	28
5. Contact us	29

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	2/29

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. It is recommended to implement a rate limiter either at device or network level. 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 following UDP ports, 34962 (PROFINET RT Unicast), 34963 (PROFINET RT Multicast), 34969 (PROFINET RPC Context Manager), 53247 (PROFINET RPC Client/Server). These ports must be open in the firewall to enable communication between compatible devices.

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	3/29

1. PROFINET

Profinet (Process Field Net) is an industry technical standard for data communication over Industrial Ethernet, designed for collecting data from, and controlling, equipment in industrial systems.

The Profinet 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 SFC132081M0201 for fieldbus related settings.

Before the Profinet communication 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** (This parameter can be set to **Off** if the fieldbus interface is only used to monitor the softstarter).



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 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
GSDML-V2.31-ABB Control Products-PSTX-20150225.xml	Device profile for the Anybus M30 module
GSDML-V2.35-ABB-Control-Products-PSTX-20210326.xml	Device profile for the Anybus M40 module

Device Access Point (DAP) selection

Use the table below to select the correct Device Access Point (DAP) version. DAP v2.1 added support for S2 Redundancy and require Softstarter v1.40 together with Anybus Profinet v1.56 or later.

Softstarter firmware version		
Profinet module firmware version	v1.34 - v1.39	v1.40
v1.41	2.0	2.0
v1.56	2.0	2.1

**Information**

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 PSTX softstarter is conformant to PROFINET Class B specification and this implies that the softstarter supports the Simple Network Management Protocol (SNMP). By sending SNMP requests to the softstarter, it is possible to retrieve information about the network setup. To prevent outsiders to get information about the internal network, it is a strong recommendation to employ a firewall with the aim to block any undesired SNMP traffic to ports 161 and 162. With the Anybus Profinet M40 module, it is also strongly recommended to change the SNMP public community strings. With the Anybus M30 module, it is not possible to change the default values of the SNMP community strings.

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/hardware 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. Software compatibility

PROFINET module AB6500-B is supported from PSTX_CB v1.34.6.

PROFINET module AB6968-B and AB6968-C is supported from PSTX_CB v1.36.2.

1.2. Digital input telegram

To PLC from softstarter. The input data is updated every 20 ms.

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	5/29

Word in input data area	Digital input byte	Bit	Data	Description
0	0	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 1.2
		7	Programmable Digital Input 2	
	1	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

Function	Data
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
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 input telegram

To PLC from the softstarter.

All analog data is represented as 16-bit values. The input data is updated every 20 ms.

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

Word in input data area	Analog input word	Data	Representation
1	0	FBT Return Value	See section 2 Fieldbus Tasks
2	1	Programmable Analog Input 1	Function of programmable analog input, see section 1.4
3	2	Programmable Analog Input 2	
4	3	Programmable Analog Input 3	
5	4	Programmable Analog Input 4	
6	5	Programmable Analog Input 5	
7	6	Programmable Analog Input 6	
8	7	Programmable Analog Input 7	
9	8	Programmable Analog Input 8	
10	9	Programmable Analog Input 9	
11	10	Programmable Analog Input 10	

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	8/29

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°C
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°C Example: 750 ⇒ 25°C
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°C
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

Function	Representation
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.

1.6. Digital output telegram

From PLC to the softstarter.

Word in output data area	Digital output byte	Bit	Data	Description
0	0	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	

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	10/29

Word in output data area	Digital output byte	Bit	Data	Description
		7	Start1	Start1 if sequence start
1		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
1	2	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"
	3	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"

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	11/29

Word in output data area	Digital output byte	Bit	Data	Description
		27 (3)	2DO3 relay com- mand	Set "1" to activate the external 2DO3 output relay. Note that parameter 11.12 2DO3 function has to be set as "Fieldbus"
		28 (4)	Refresh parameters	Restart fieldbus interface to refresh communication parameters
		29 (5)	Spare	
		30 (6)	Spare	
		31 (7)	Spare	

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	12/29

1.7. Analog output telegram

From PLC to the softstarter.

All analog data is represented as 16-bit values.

Word in output data area	Analog output word	Data	Representation
2	0	FBT Control Word	This register is used to read parameters (see fieldbus tasks)
3	1	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 [%]"
4	2	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.

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

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.

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	14/29

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 $50/10^2 = 0.5\text{s}$.

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	16/29

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 $1 \cdot 10^2 = 100$.
4. Set FBT Argument 3 to 0 as $100 \leq 65535$ 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 FFFFFFFE7_{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 SNMP community strings

For the M30 version of the Anybus Profinet module, only the read-only access is supported and only the default community string "public" can be used.

For the M40 version, both read-only and read-write access is supported, and the default community strings ("public" and "private") can be changed.

The community strings are only changed if parameter 12.39 Change SNMP community string is configured to "Yes". The community string that is set has eight characters and depends on the value of parameter 12.40 SNMP community string part 1 (which consist of the first 4 characters) and parameter 12.41 SNMP community string part 2 (which consist of the remaining 4 characters). These two parameters are of the data type uint32.

Below is a description of how a community string is configured:

In this example "ABB-PSTX" is the community string that shall be configured.

First separate the string into two parts

"ABB-"
"PSTX".

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	17/29

Then use an ASCII table to convert the strings to hexadecimal format

“ABB-“ is equal to 0x4142422D

“PSTX” is equal to 0x50535458

Next step is to convert the hexadecimal numbers to decimal format

0x4142422D is equal to 1094861357

0x50535458 is equal to 1347638360

Set parameter 12.40 SNMP community string part one to 1094861357 and set parameter 12.41 SNMP community string part two to 1347638360.

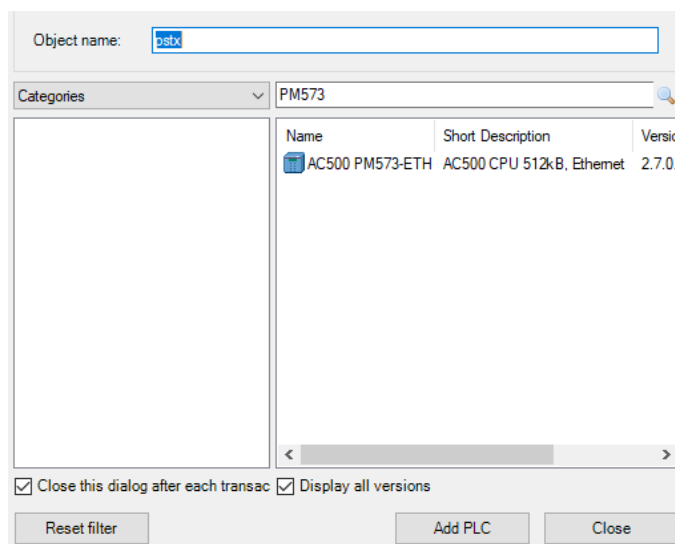
“ABB-PSTX” or whichever community string that is configured will be used for both read-only and read-write access.

4. Configure ABB Automation Builder

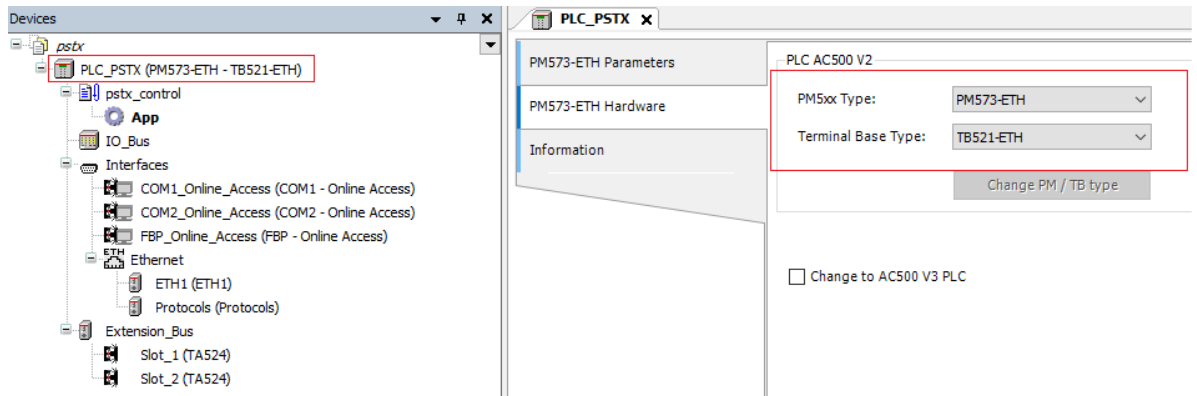
This part of the document describes how to configure an ABB AC500 PLC using the CM579-PNIO module to control a PSTX Softstarter with an Anybus Profinet module.

4.1. Create a new project

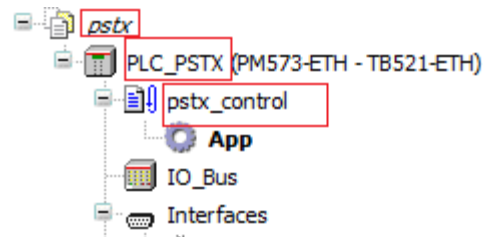
1. Open Automation Builder
2. Select File->New Project->AC500 project->OK.
3. Select the correct PLC CPU in the list and click “Add PLC”.



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.

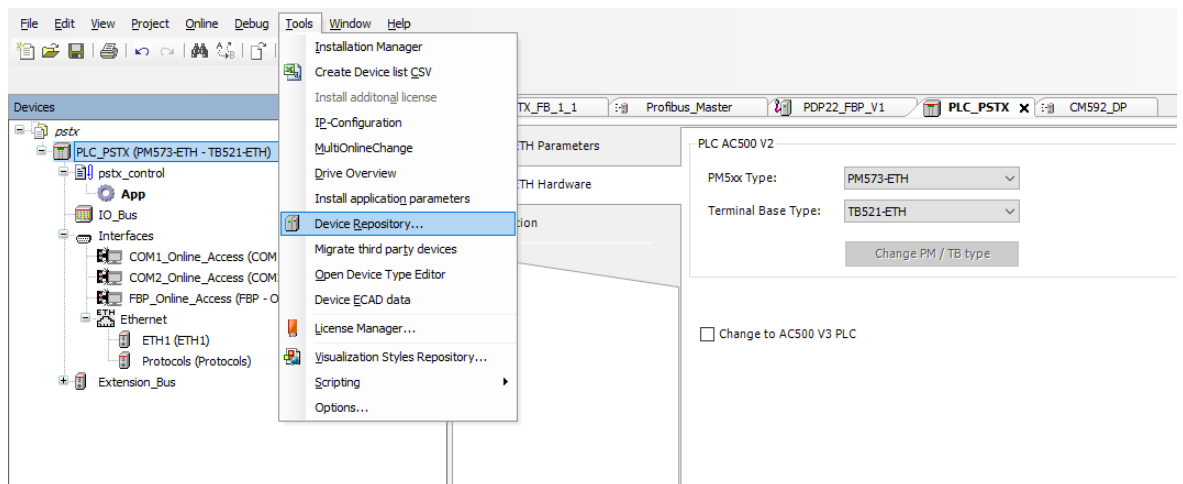


5. Optional: Rename the project to for example "pstx" with File->Save Project As... and the Application to for example "pstx_control" by just clicking on it and rename.



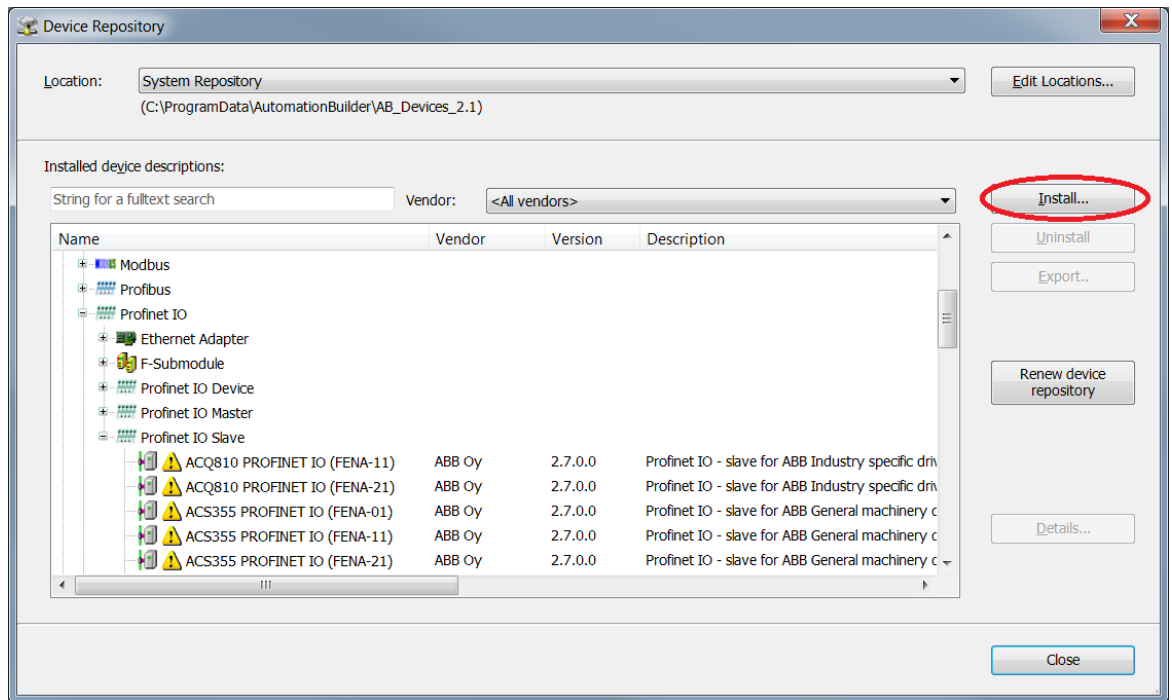
4.2. Install the Profinet GSDML-file

1. Select Device Repository in the Tools menu.



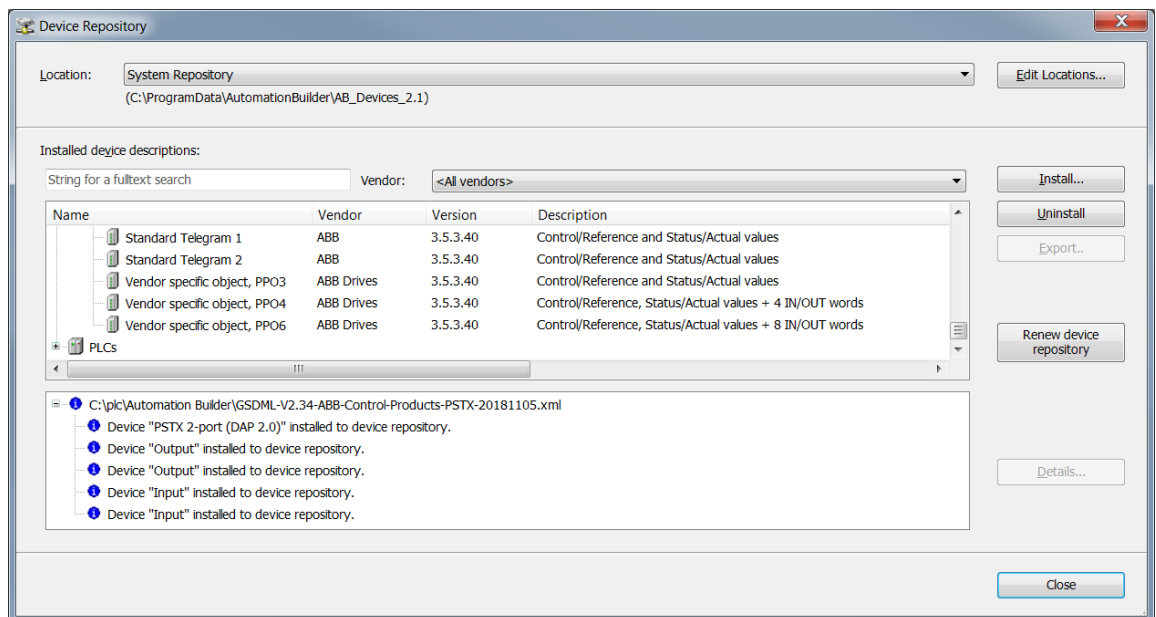
STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	19/29

2. Select System Repository as Location and Click Install.



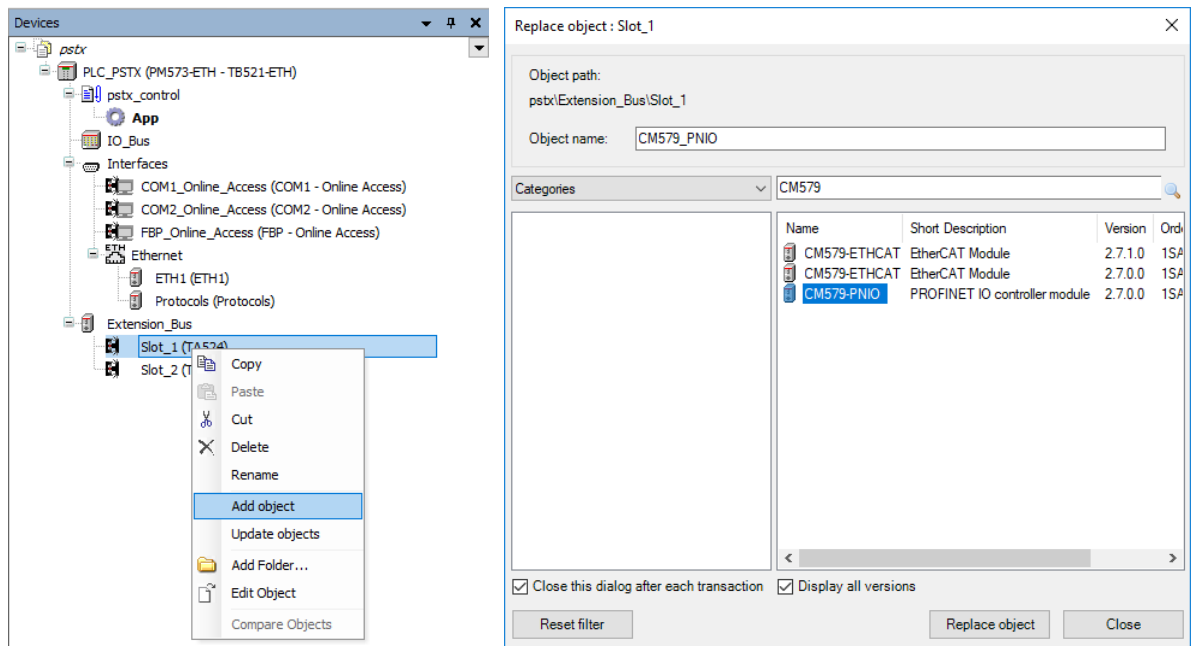
3. Select the GSDML-file that corresponds to the Anybus Profinet module:
 GSDML-V2.31-ABB Control Products-PSTX-20150225.xml - for the Anybus M30 module
 GSDML-V2.34-ABB-Control-Products-PSTX-20181105.xml - for the Anybus M40 module.

It is important to use the correct GSDML file (otherwise the communication will not work). Check the downside of the Anybus module. There it will be shown if the module is of type M30 or M40.



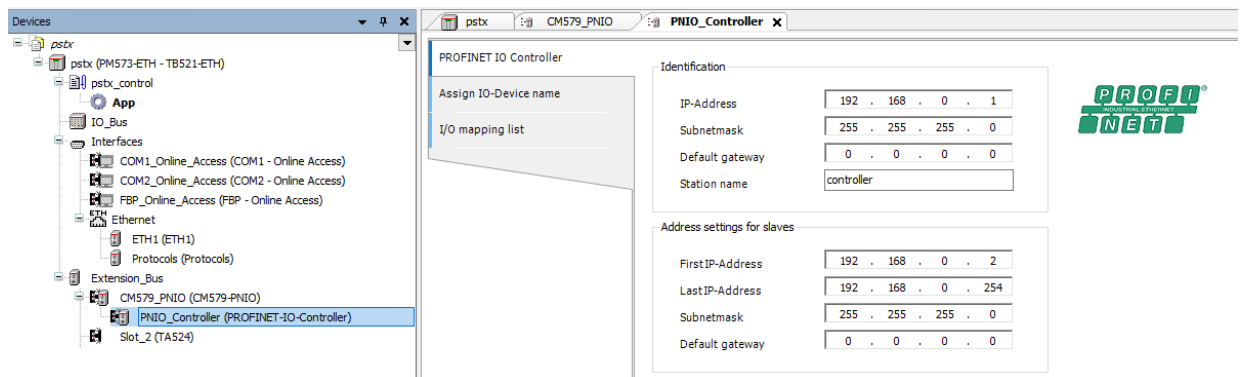
4.3. Add the CM579-PNIO PROFINET IO controller module

1. Right click on the empty slot where the CM579-PNIO module is installed and select Add object. Then select CM579-PNIO in the Replace object window.



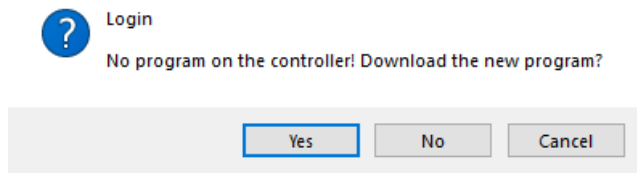
4.4. Setup IP-Addresses and station name

1. Double click the controller and setup IP-addresses for the master controller and a range for slaves in the same network.

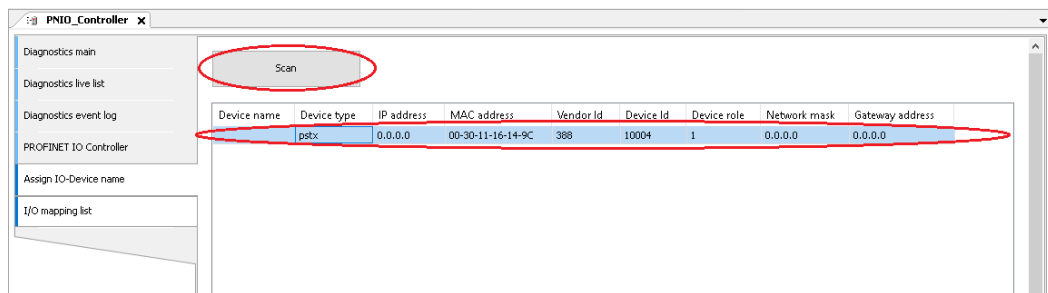


2. Make sure that the PLC Profinet master module is connected to the Anybus Profinet module in PSTX slave.

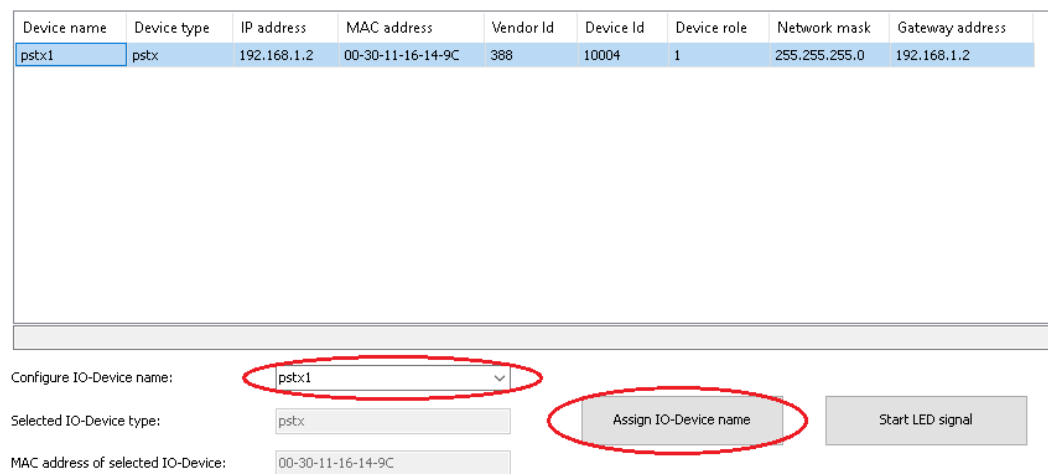
- Go online with the PLC by clicking Alt+F8, and upload the PLC-program by selecting yes in prompt up window and run.



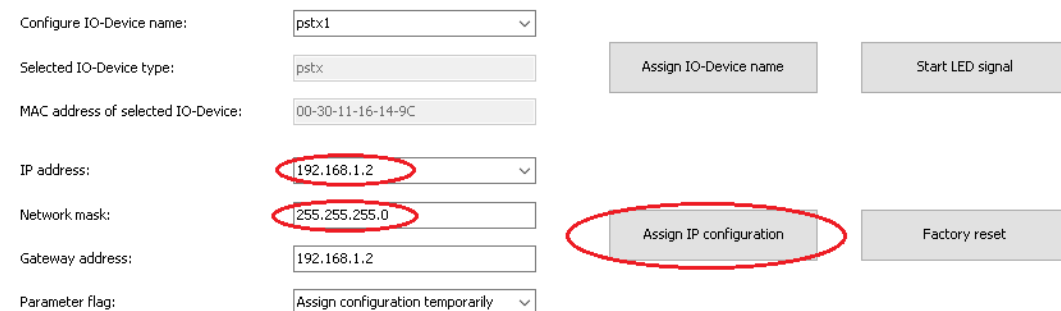
- Scan for slaves to find the PSTX slave and select it.



- Set IO-Device name and click Assign IO-Device name.

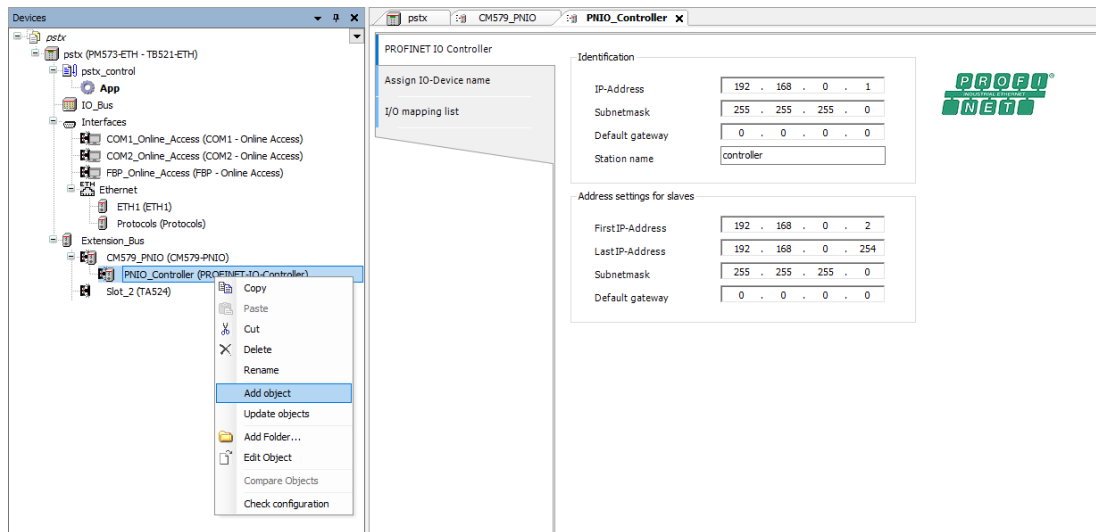


- Set IP address and network mask and click Assign IP configuration.

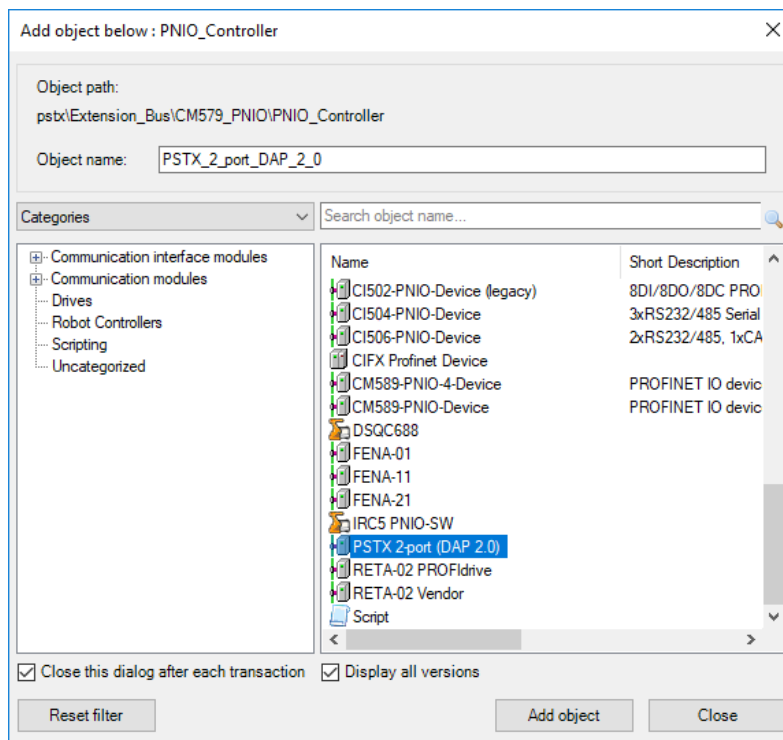


4.5. Add the PSTX slave

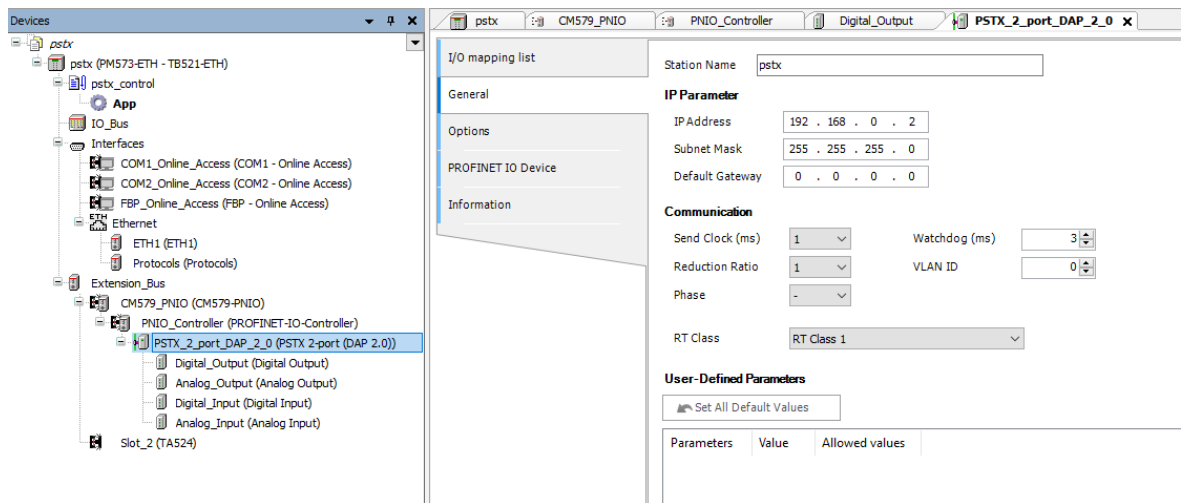
1. Right click on the newly installed Profibus_Master and select Add object.



2. Select PSTX 2-port (DAP 2.0) found under Uncategorized and click Add object.

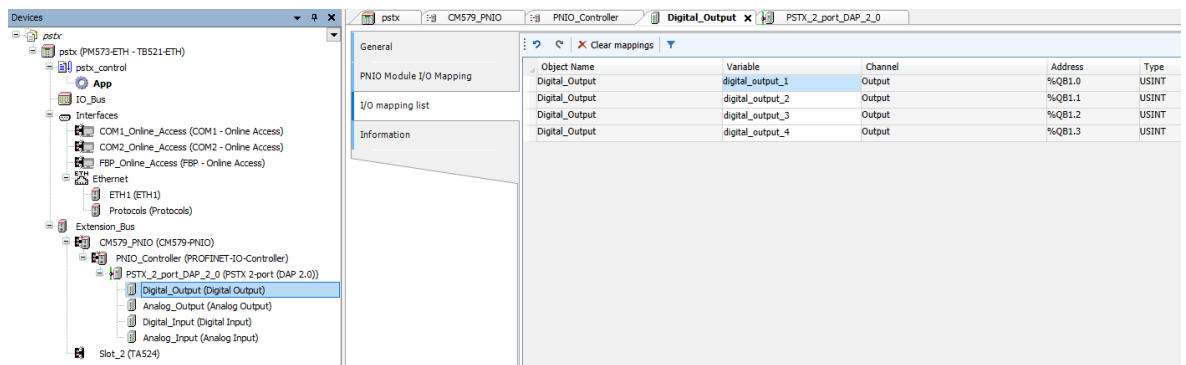


3. Double click PSTX_2_port_DAP_2_0 in device tree. Click on General and change Station Name and IP Parameter.



The station name and IP parameters should match what you entered in section 3.4 “Setup IP-Addresses and slave station name”.

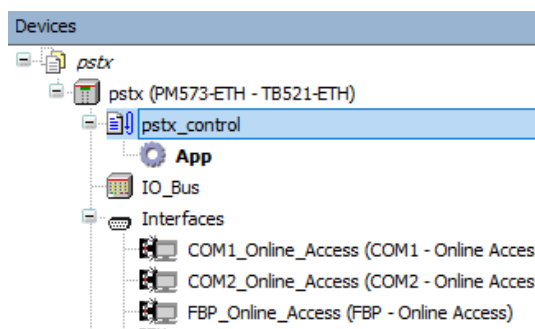
4. Map the signals to variable names in I/O Mapping list for the object PSTX_2_port_DAP_2_0. For example:



4.6. Write a simple PLC program to control the softstarter

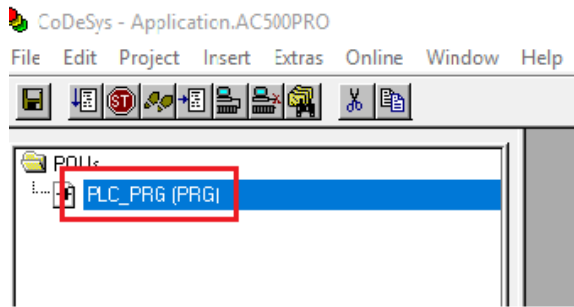
We perform the following steps for building our start-stop demo program in CoDeSys.

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

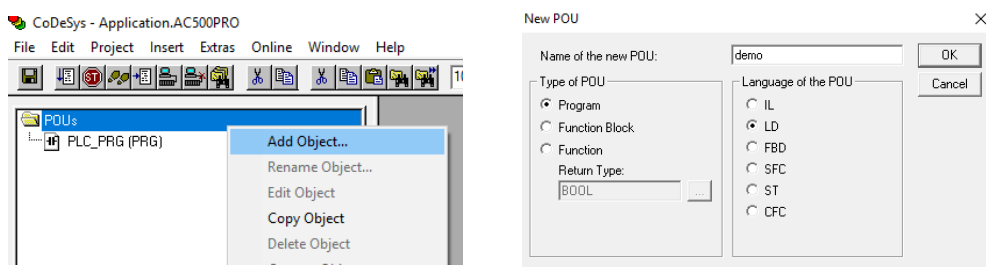


STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	24/29

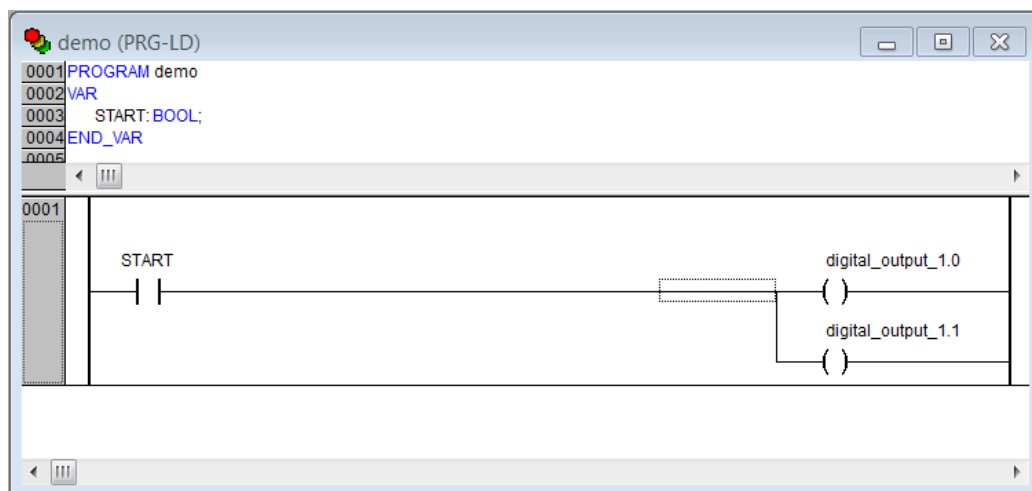
- Open program window by double clicking the default program in POU's in CoDeSys.



- We choose to use LD as the language of the POU here by right click POU's -> Add Object... -> Insert Name of the new POU with "demo"-> Choose "LD" for "Language of the POU" -> OK.



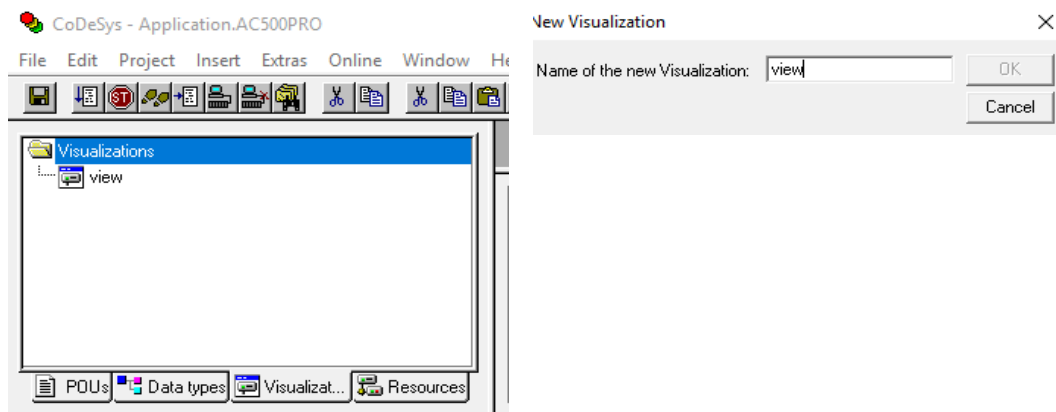
- Select the first network, create a contact "START" (by CTRL+K and putting name at "??") and two coils "digital_output_1.0" and "digital_output_1.1" (by CTRL+L) in first network. We let data types as default by clicking OK directly in Declare Variable window. We set digital_output_1 bit 0 and 1 because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




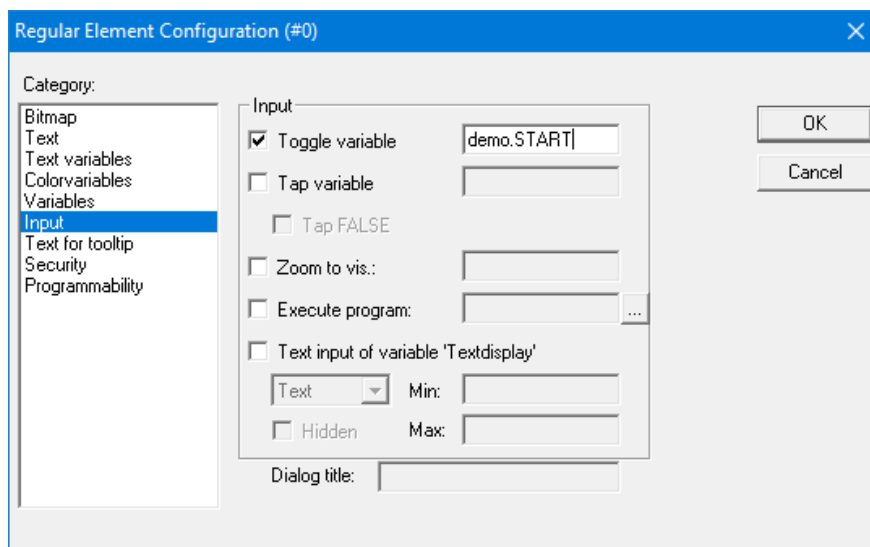
5. Create a second network by CTRL+T.
6. Select the second network, add a coil for automode “digital_output_1.3” (by CTRL+L).



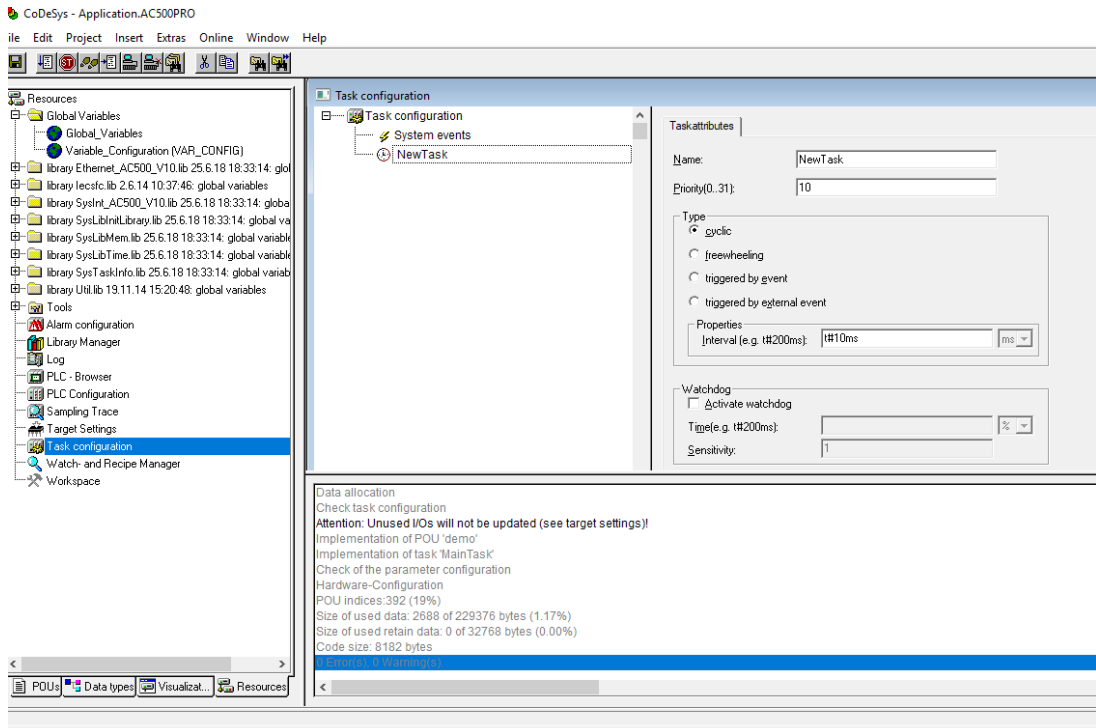
7. Now we want to create one control button for signing the value of “START” from the first network into TRUE. We do this by Visualization -> right click -> Add object -> Write name of the new Visualization as “view” -> OK.



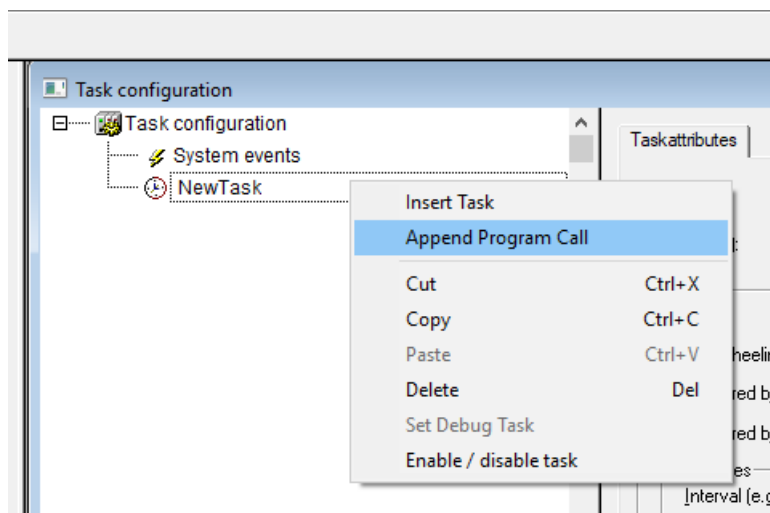
8. We draw a shape as the button  -> double click the shape -> Regular Element Configuration -> Input -> check Toggle variable -> insert “demo.START” ->OK.



- 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.

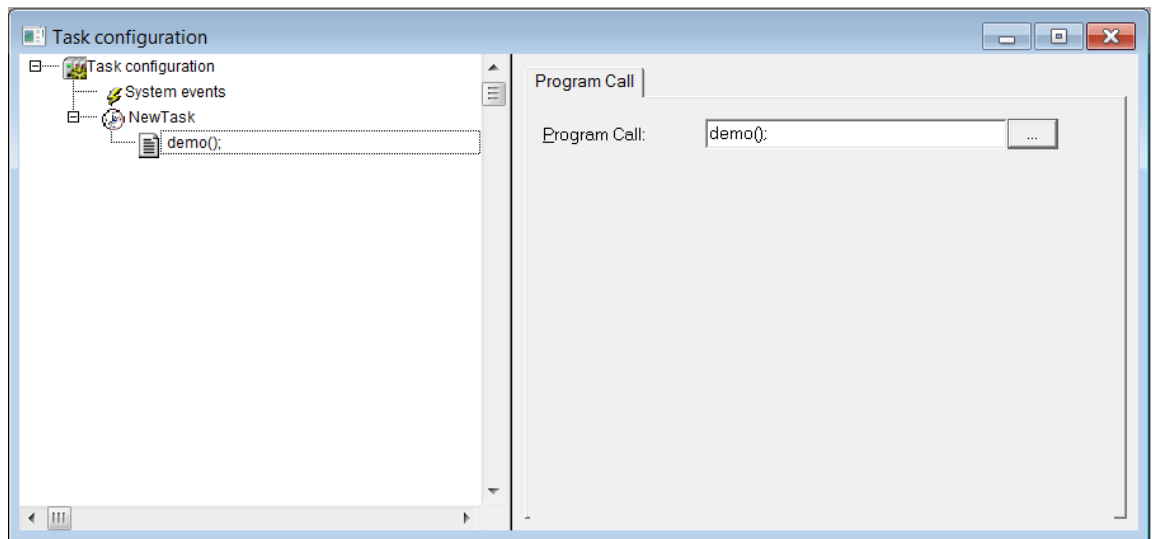


- Right click on the NewTask and select Append Program Call.



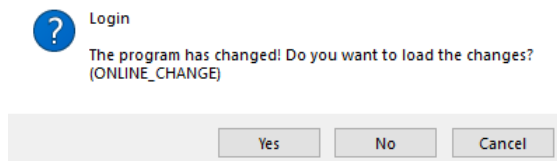
STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	27/29

11. Select the demo program call.

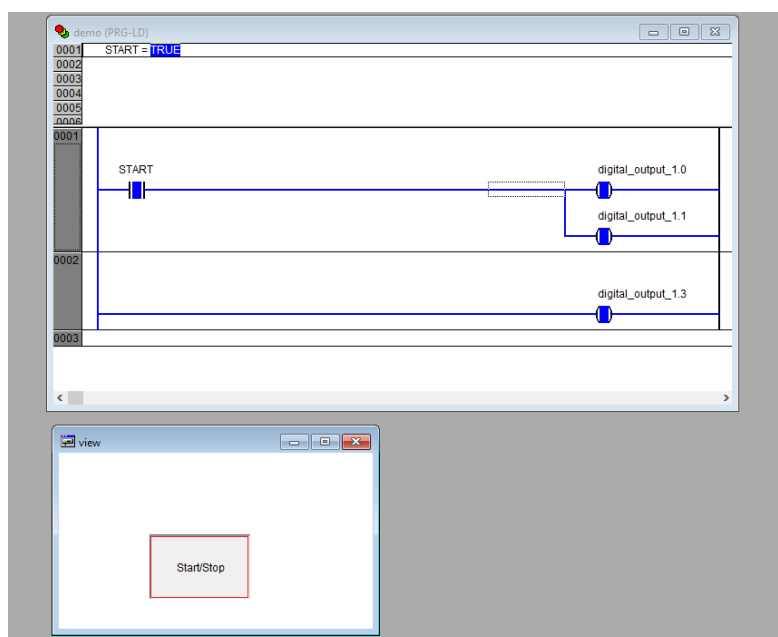


4.7. Build and run the PLC demo program

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.



5. Contact us

For more information, please contact
your local ABB representative or visit
<https://solutions.abb/softstarters>

STATUS	SECURITY LEVEL	DOCUMENT ID.	REV.	LANG.	PAGE
Approved	Public	1SFC132094M0201	F	en	29/29