SOFTSTARTER TYPE PSTX

Fieldbus Plug

Modbus RTU

PREPARED	STATUS	SECURITY LEVEL		
2021-03-12	Approved	Public		
RELEASED	DOCUMENT KIND			
2021-04-27	Manual			
OWNING ORGANIZATION	DOCUMENT ID.	REV.	LANG.	PAGE
Motor Starting & Safety Softstarters	1SFC132092M0201	F	en	1/32

Contents

1. Modbus RTU

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. 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 FbPlug.

– 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).

– Parameter 12.4 Fieldbus address set to an available Modbus slave id. In the examples (section 4) the fieldbus address is set to 47, but this parameter can be set to any value between 1-247.

The parity and number of stop bits are automatically detected by the fieldbus plug.

The baud rate depends on the configured fieldbus address according to the table below:

Fieldbus address	Baud rate bits/s
1 to 32	9600
33 to 65	19200
66 to 98	57600
99 to 247	19200

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 “FbPlug”.

If there is no message passed between the PSTX softstarter and the Fieldbus plug for more than 100ms, the PSTX softstarter will trip on fieldbus communication failure protection (P1E00) and with the default configuration, the motor will be stopped. If the fieldbus 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

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. The input data is updated every 20 ms.

Use Modbus function code 01, Read Coil Status.

Protocol Address	Modicon Address	Bit	Data	Description
0000h	00001	0	Auto Mode status[1]	0 = softstarter control from fieldbus not allowed
0001h	00002	1	Event status	0 = No active fault/warning/protection
0002h	00003	2	Ready to start	0 = A start will probably cause a fault, 1 = A start will probably not cause a fault
0003h	00004	3	FBT Response 0	See Fieldbus Tasks
0004h	00005	4	FBT Response 1	See Fieldbus Tasks
0005h	00006	5	FBT Toggle Bit	See Fieldbus Tasks
0006h	00007	6	Programmable Digital Input 1	Function of programmable digital input, see section 3.2
0007h	00008	7	Programmable Digital Input 2	
0008h	00009	8	Programmable Digital Input 3	
0009h	00010	9	Programmable Digital Input 4	
000Ah	00011	10	Programmable Digital Input 5	
000Bh	00012	11	Programmable Digital Input 6	
000Ch	00013	12	Programmable Digital Input 7	
000Dh	00014	13	Programmable Digital Input 8	
000Eh	00015	14	Programmable Digital Input 9	
000Fh	00016	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 Fieldbus Plug Accessory.
- The state of the Local/Remote switch on the HMI.
- The parameter “Fieldbus control”.
- The digital input “Fieldbus disable”.

3.2. 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.

Function	Data
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.
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.

3.3. 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. 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
0200h	30513	FBT Return Value	See Fieldbus Tasks
0201h	30514	Programmable Analog Input 1	Function of programmable analog input, see section 3.4
0202h	30515	Programmable Analog Input 2	
0203h	30516	Programmable Analog Input 3	
0204h	30517	Programmable Analog Input 4	
0205h	30518	Programmable Analog Input 5	
0206h	30519	Programmable Analog Input 6	

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
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

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.

3.5. Digital output telegram

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

Protocol Address	Modicon Address	Bit	Data	Description
0100h	257	0	Start	Commence a start when signal is set.
0101h	258	1	Stop	Commence a stop when signal is negated.
0102h	259	2	Fault reset	Reset signal for possible events.
0103h	260	3	Auto mode	This must be set for controlling the motor.
0104h	261	4	Slow speed reverse	Perform slow speed reverse when signal is set.
0105h	262	5	Slow speed forward	Perform slow speed when signal is set.
0106h	263	6	Spare	
0107h	264	7	Start1	Start1 if sequence start.
0108h	265	8	Start2	Start2 if sequence start.

Protocol Address	Modicon Address	Bit	Data	Description
0109h	266	9	Start3	Start3 if sequence start
010Ah	267	10	Motor heating	Perform motor heating when signal is set
010Bh	268	11	Stand still brake	Perform stand still brake when signal is set
010Ch	269	12	Start reverse	Commence a reverse start when signal is set
010Dh	270	13	Spare	
010Eh	271	14	Emergency mode	Set to “1” to enable emergency mode
010Fh	272	15	FBT Toggle Bit	See Fieldbus Tasks
0110h	273	16	User defined trip	Set to “1” to trigger user defined protection
0111h	274	17	Switch to remote control	Switch to remote control when signal is set (rising edge triggered)
0112h	275	18	Pump cleaning automatic	Perform automatic pump cleaning when signal is set
0113h	276	19	Pump cleaning forward	Perform forward pump cleaning when signal is set
0114h	277	20	Pump cleaning reverse	Perform reverse pump cleaning when signal is set
0115h	278	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”
0116h	279	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”
0117h	280	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”
0118h	281	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”
0119h	282	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”
011Ah	283	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”
011Bh	284	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”
011Ch	285	28	Refresh parameters	Restart fieldbus interface to refresh communication parameters
011Dh	286	29	Spare	
011Eh	287	30	Spare	

Protocol Address	Modicon Address	Bit	Data	Description
011Fh	288	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), Write Multiple Registers

Protocol Address	Modicon Address	Data	Representation
0300h			Address offset
0300h	40769	FBT Control Word	This register is used to read parameters (see fieldbus tasks)
0301h	40770	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 [%]”
0402h	40771	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 RTU - A set-up example

4.1. Softstarter PST Modbus RTU communication

This document describes an application example between a Modbus RTU master (PLC CPU, PC, etc.) and the ABB softstarter PSTX equipped with an Anybus Modbus-RTU module. In this example the softstarter address is 47.

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

Modbus RTU FBP Fieldbus Plug MRP21. FBP technical description 2CDC194001D0203.

Softstarter PSTX Installation and commissioning manual, document 1SFC132081M0201.

4.2. Settings

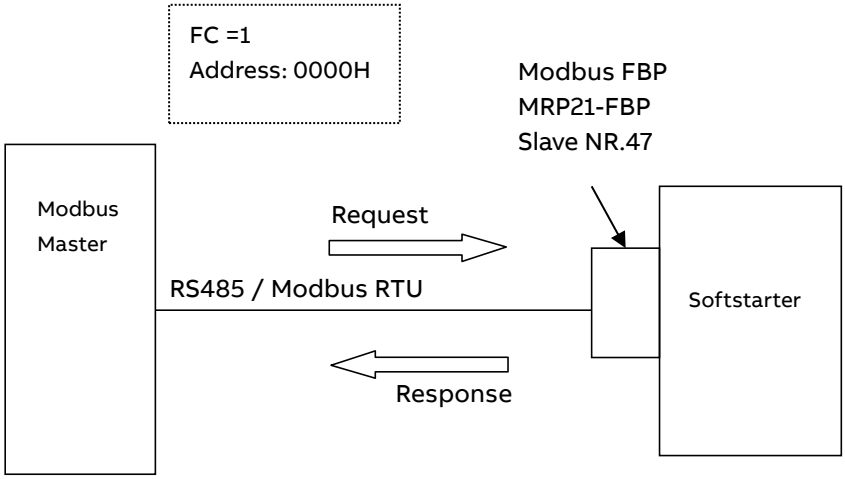
1. **Set the softstarter address and field bus communication.**
   Change the address of the softstarter to 47 (Fieldbus Address) and enable fieldbus control (Fieldbus control = On).

2. **Set the communication parameters.**
   The communication parameters are:
   1 Start bit, 8 Data bits, the Parity will be adapted to the master.
   The baud rate is mapped to the slave address; e.g. the baud rate to the slave 47 is 19200.

   Therefore the master and the slave will have the communication parameters:
   1 start bit, 1 stop bit, 8 data bit, even parity, 19200 baud.

3. Select the FBP interface.
   The previous changes are taken in effect when the fieldbus interface is changed.
   Change parameter FB interface connector to FbPlug.

4.3. Read binary input telegram



E.g.: Read 16 bit starting at the address: 0000H

Request:

2F 01 00 00 00 10 3B 88

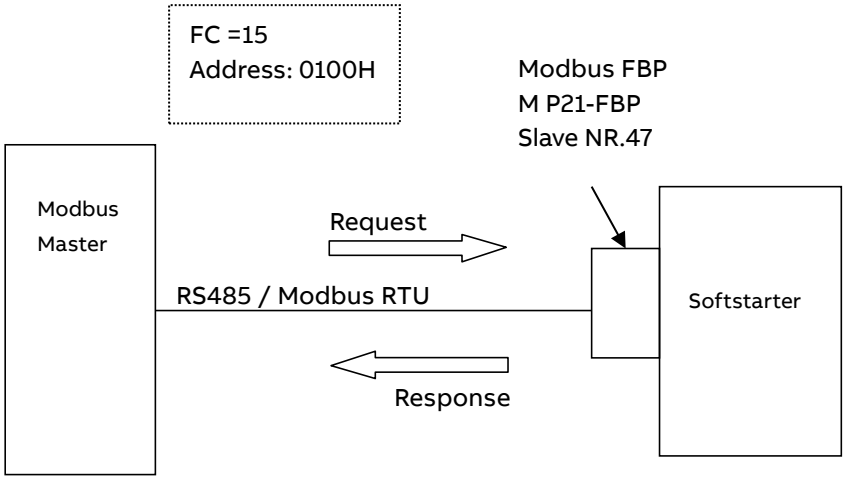
Slave 47 | FC = 01 | Address | 16 bit | CRC

Response:

2F 01 02 00 00 51 FA

Slave 47 FC = 01 2 Byte CRC

4.4. Write binary output telegram



E.g.: Write 16 bit starting at the address: 0100H

Request:

2F 0F 01 00 00 10 02 00 00 27 41

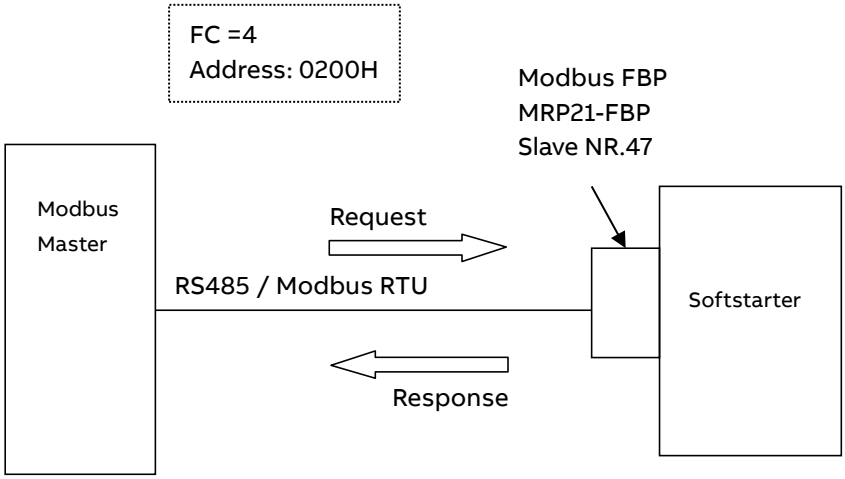
Slave 47 FC = 15 Address 16 bit 2 Byte Output CRC

Response:

2F 0F 01 00 00 10 53 B5

Slave 47 FC = 15 Address 16 bit CRC

4.5. Read analog input telegram



E.g.: Read analog output words 2 & 3. Phase L1 current and Phase L2current with default settings.

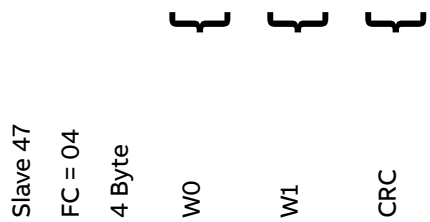
Request:

2F 04 02 01 00 02 27 FD

Slave 47 FC = 04 Address 2 words Output

Response:

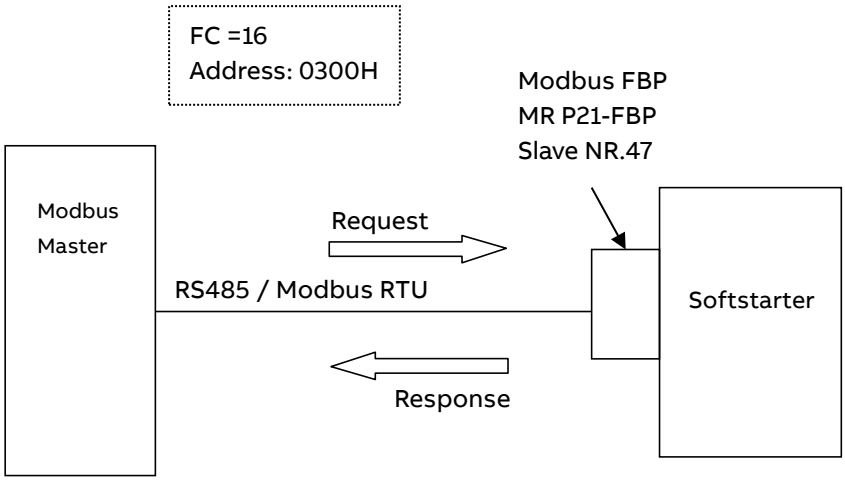
2F 04 04 00 00 00 00 35 86



W0: Phase L1 current

W1: Phase L2 current

4.6. Write analog input telegram



E.g.: Write analog output words 1 & 23. FBT Control Word and FBT Argument 2.

Request:

2F 10 03 00 00 02 04 00 00 00 00 78 77

Slave 47	FC = 16	Address	2 words	Address	W0	W1	CRC

W0: FBT Control Word

W1: FBT Argument 2

Response:

2F 10 03 00 00 02 47 C2

Slave 47	FC = 16	Address	2 words	CRC

5. 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.

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's value. 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's value. 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 7.25 Complete parameter list.

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 7.25 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.

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.

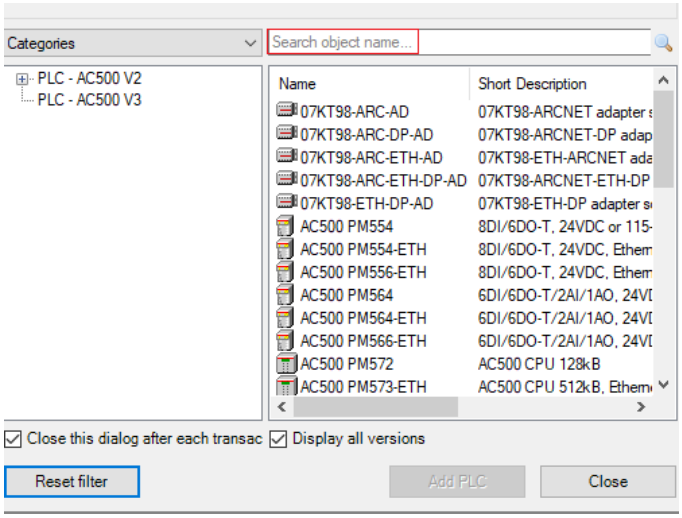
6.1. Create a new project in Automation Builder

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

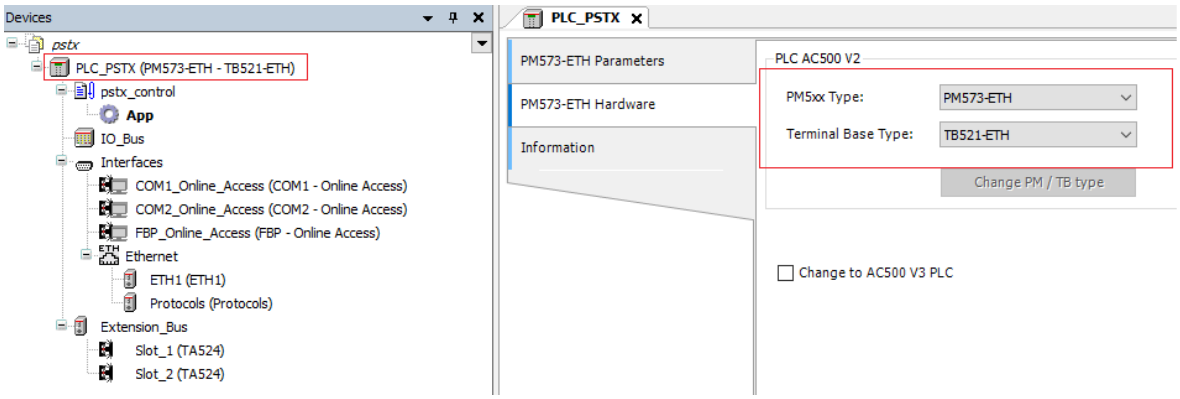
1. Open Automation Builder

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

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

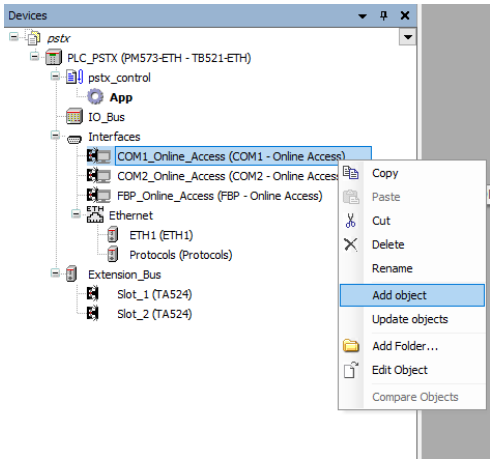


4. Check that the correct device type is selected by double clicking the device name in Devices field. Check that the correct Terminal Base Type is also selected for the tag for Hardware.

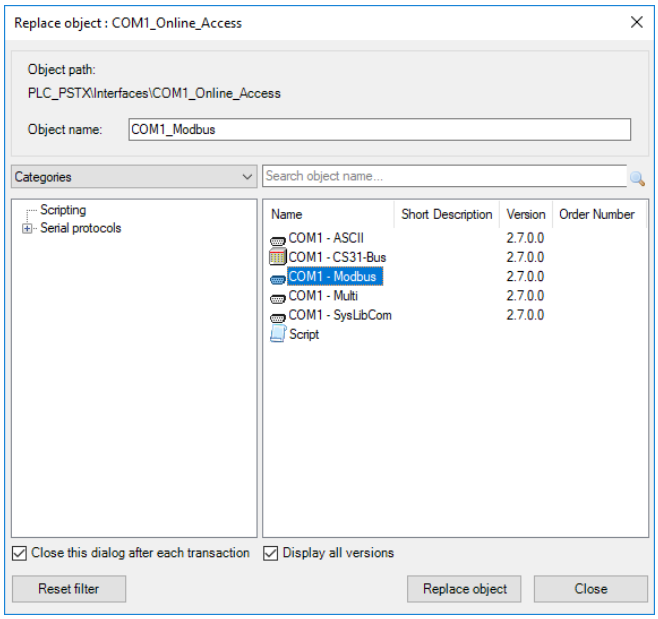


6.2. Add Modbus RTU master to project

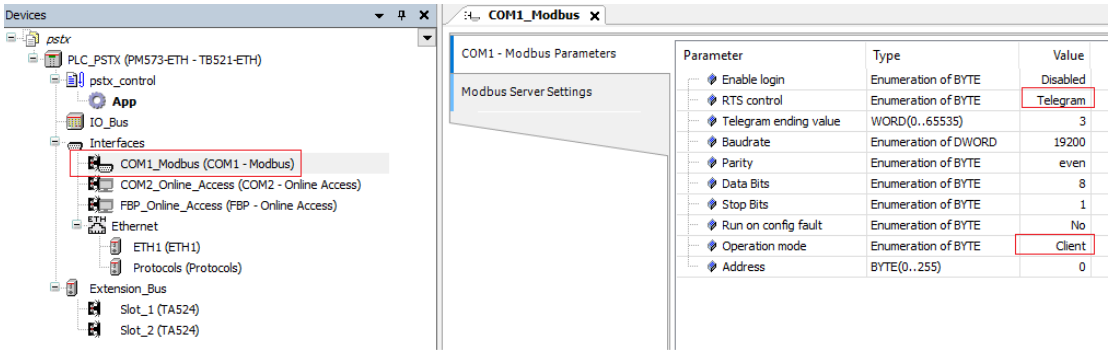
1. Right click on one of the comports and select Add object



2. Select COM1 - Modbus and click Replace object



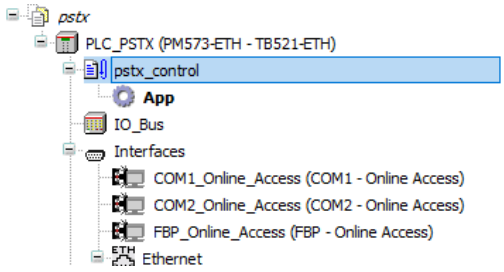
3. Double click interfaces, COM1_Modbus, from the device tree. Set RTS control to Telegram and Operation mode to Client for COM1 – Modbus Parameters in COM1_Modbus



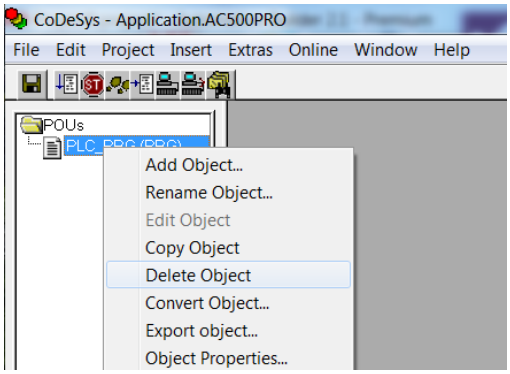
6.3. Build a START-STOP program

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

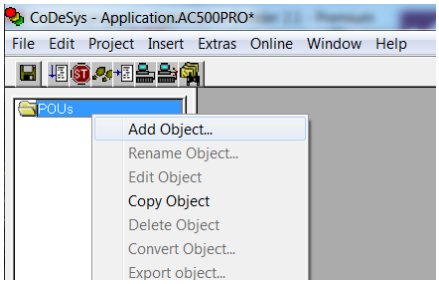
1. Open CoDeSys by double clicking your application in Devices file in Automation Builder, if it is not opened yet.

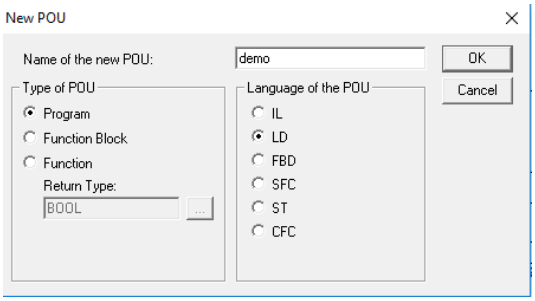


2. Delete the default POU by right click on it and select.

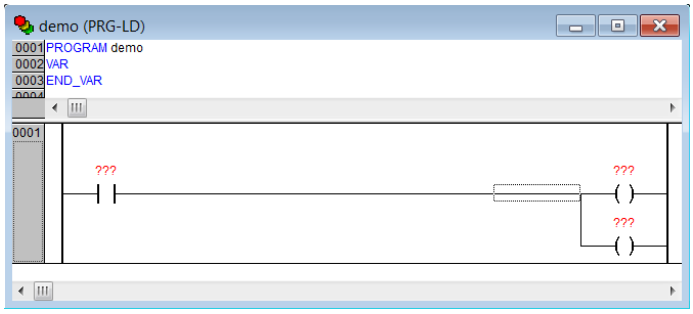


3. We choose to use LD as the language of the POU here by right click POUs -> Add Object… ->Insert Name of the new POU -> Choose “LD” for “Language of the POU” -> OK.

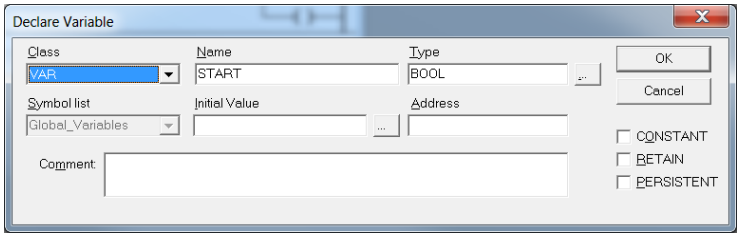




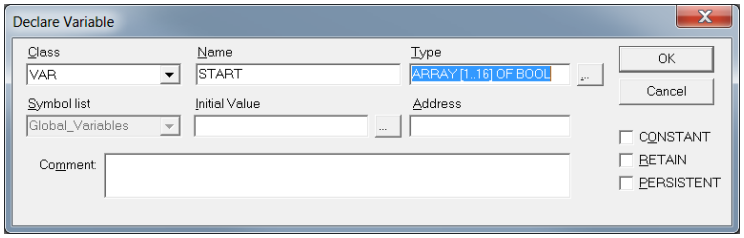
4. Open the newly created POU by double click it and select the first network, create a contact (by CTRL+K) and two coils (by CTRL+L).



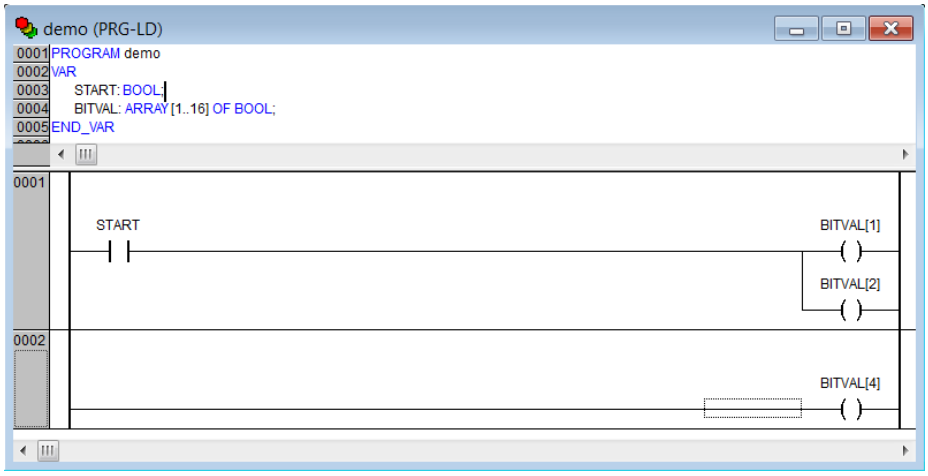
5. Name the contact START by changing the ??? to START, select type BOOL.



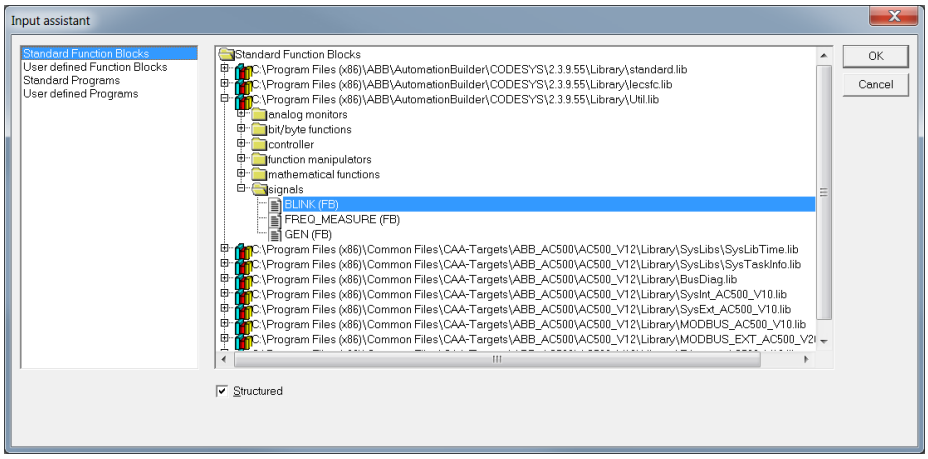
6. Name the coils “BITVAL[1]” and “BITVAL[2]”, set the type to “ARRAY [1..16] OF BOOL”.



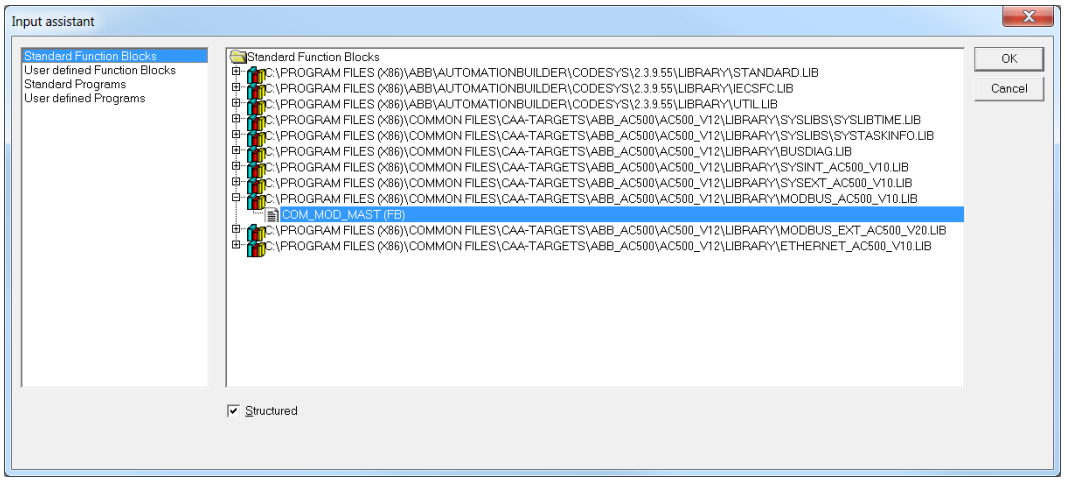
7. Add a second network by CTRL+T and add a single coil (by CTRL+L) named “BITVAL[4]”



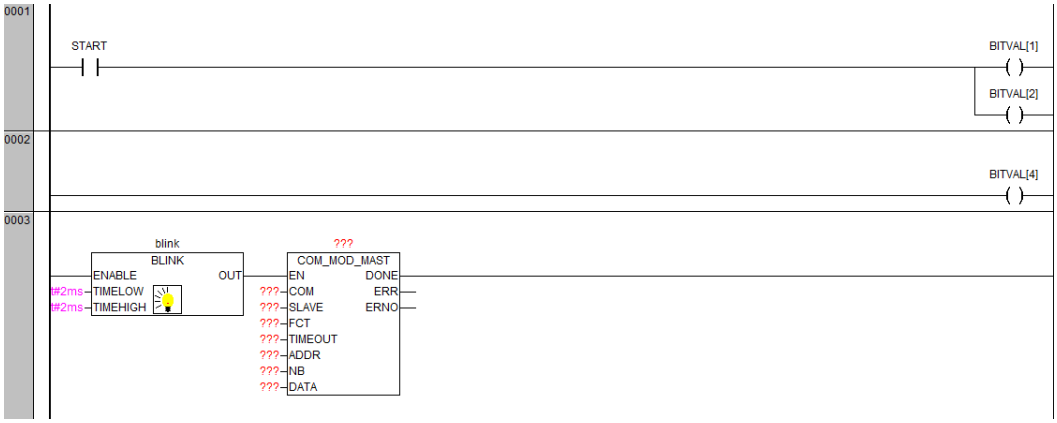
8. Add another network by CTRL+T. Create a function block “BLINK” by CTRL+B and select Standard Function Blocks -> Util.lib -> signals-> BLINK(FB)->Ok.



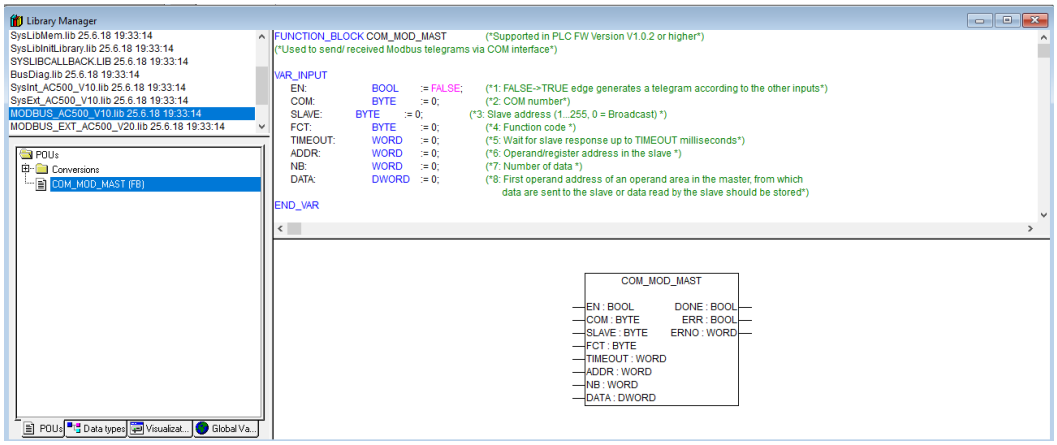
9. We name the BLINK function block as blink. We set t#2ms for TIMELOW and TIMEHIGH.
10. Continue to create a function block “COM_MOD_MAST” by CTRL+B and select Standard Function Blocks -> MODBUS_AC500_V10.LIB -> COM_MOD_MAST(FB) -> OK.



Now, we should have two function blocks in network 0003.



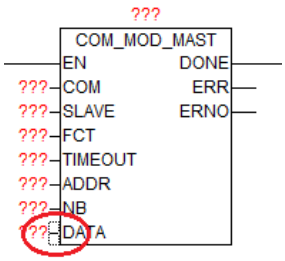
The COM_MOD_MAST is a function block for sending/receiving OpenModbus. Their definition is available from CoDeSys -> Resources -> Library Manager.



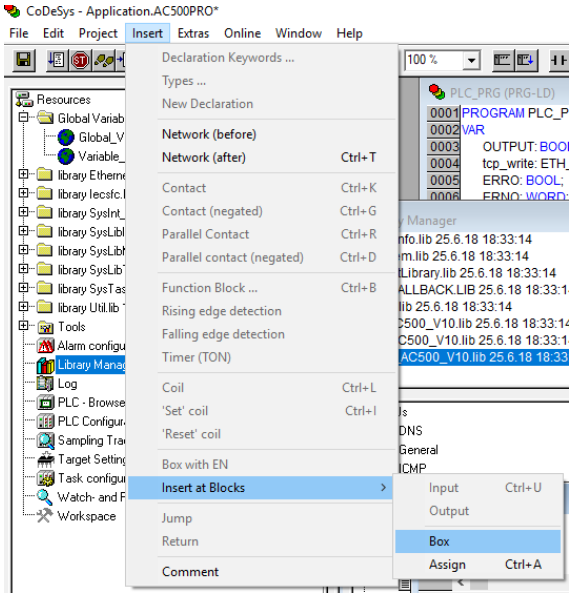
To enable this function block, it is required to send a FALSE->TRUE edge at input EN and therefore we introduce BLINK, which is for creating a flip-flop signal.

11. DATA (the data to send) require DWORD inputs. We can convert data with a box, “ADR”.

    a) Select the bar in front of DATA.



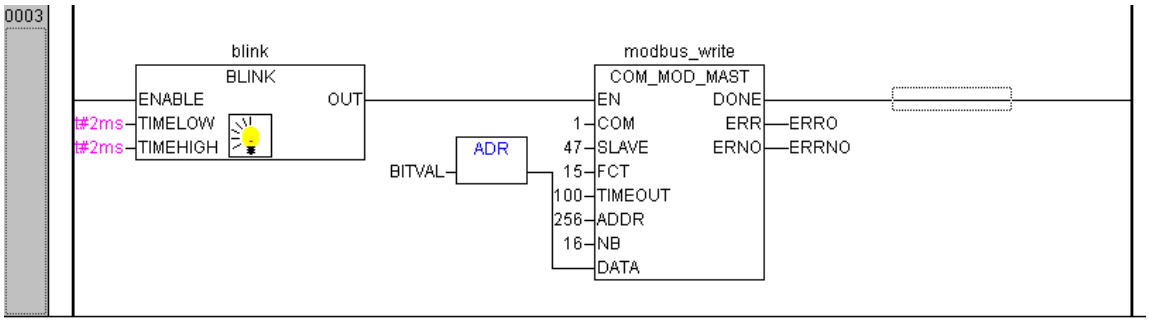
    b) Choose Insert-> Insert at Blocks-> Box.



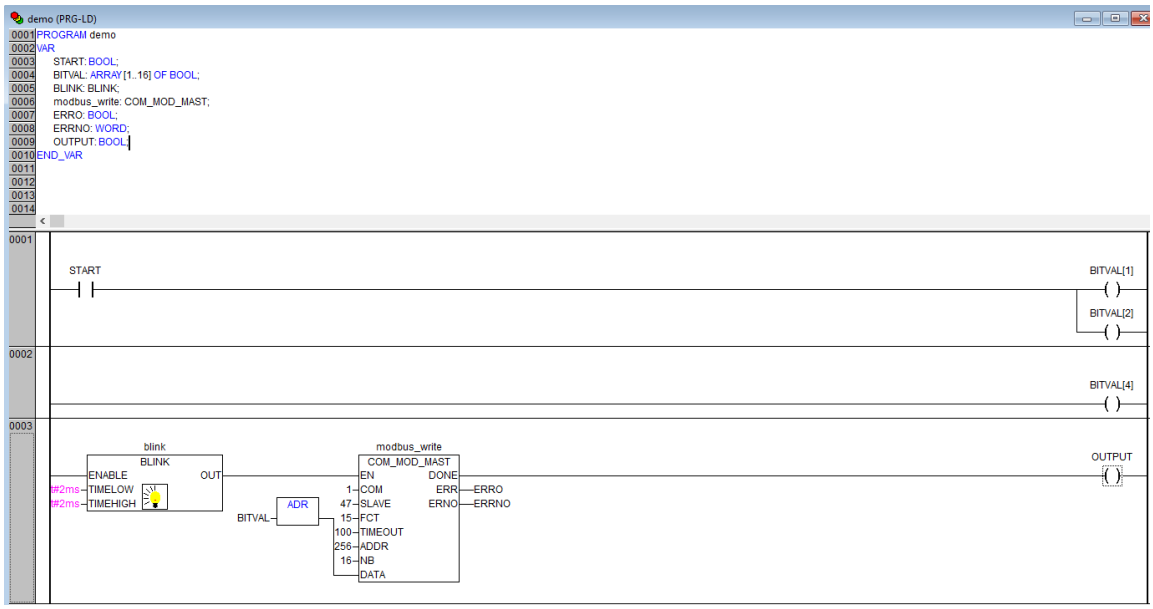
The default box is “AND”, change the name to ADR and it will be a ADR box.

12. Set:

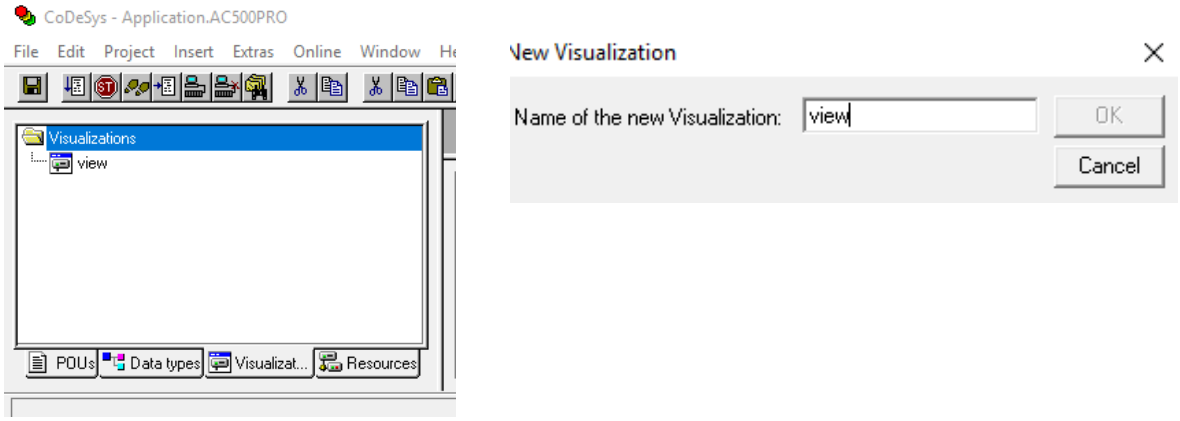
- The name of the block to modbus_write.
- COM to 1 or 2 deepening on comport used.
- Slave to the value of “Fieldbus address” (Parameter 12.4 in PSTX), note that this affects the baud rate.
- FCT to 15.
- TIMEOUT to 100.
- ADDR to 256, according to Section 3.5, the first Protocol Address is 0100h.
- NB to 16.
- ERR to ERRO (new BOOL variable).
- ERRNO (new WORD variable).



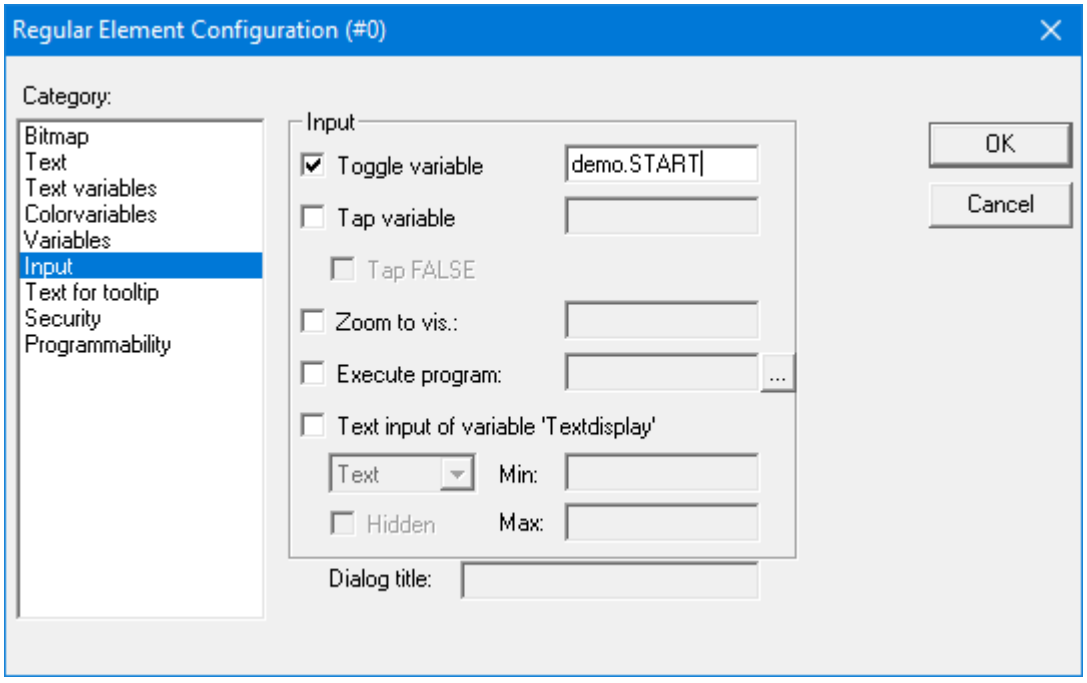
13. Insert a coil named ”OUTPUT” in the last network and the LD-program is done.



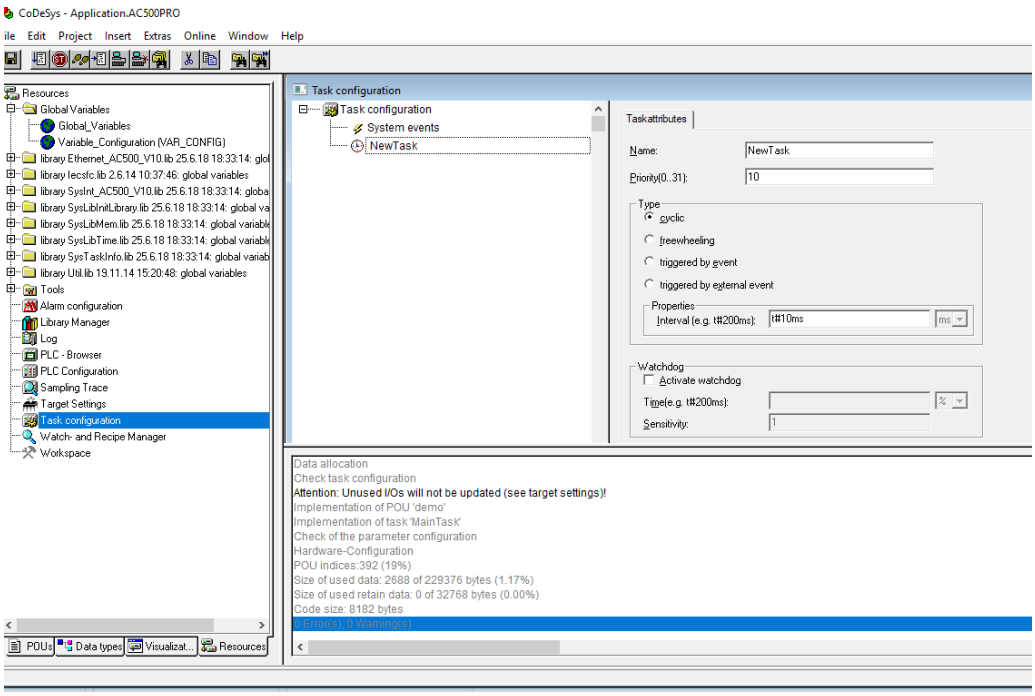
14. Now we want to create one control button for signing the value of “START” from the first network into TRUE. We do this by Visualization -> right click -> Add object -> Write name of the new Visualization as “view” -> OK.



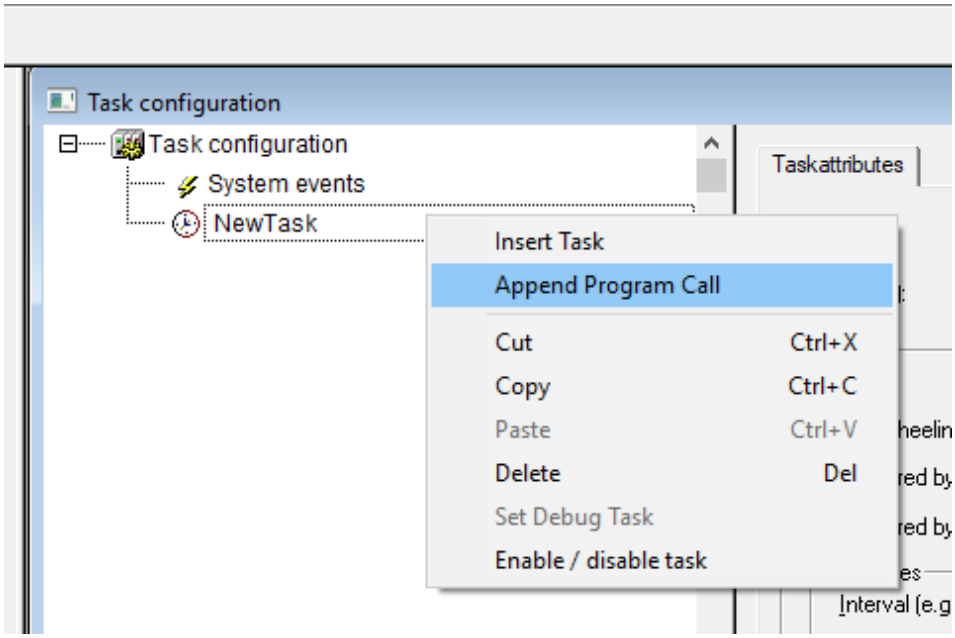
15. We draw a shape as the button  -> double click the shape -> Regular Element Configuration -> Input -> check Toggle variable -> insert “demo.START” ->OK.



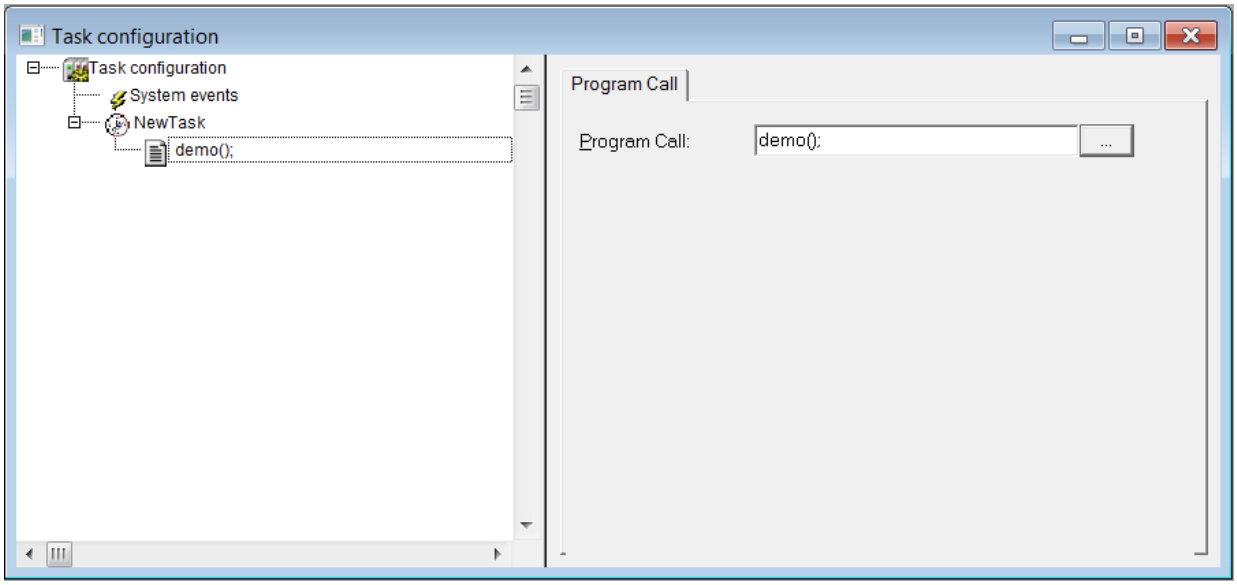
16. We configure this program into task configuration by Resource -> Task configuration -> Right click Task configuration -> Append Task -> Insert t#10ms in Properties in Taskattributes. Then we need to sign our program to this task by right click NewTask-> Append Program Call-> Choose demo(PRG) by clicking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



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



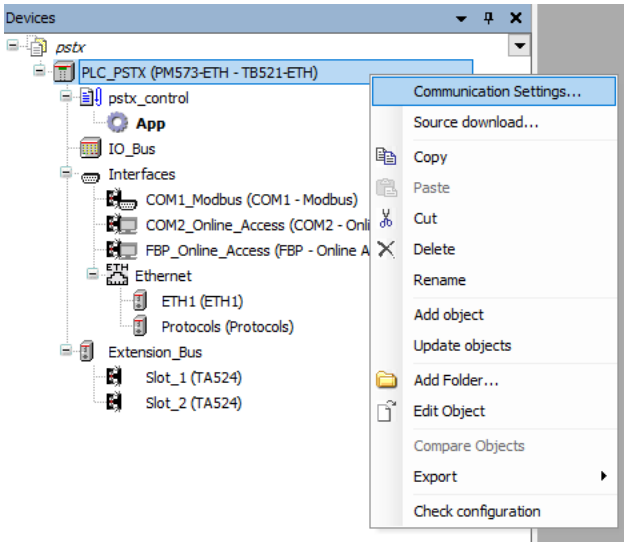
18. Select the demo program call.



19. Now we can build the project by Project -> Build. Check again if fieldbus is connected correctly. We can then run the program by pressing ALT+F8 and then F5.

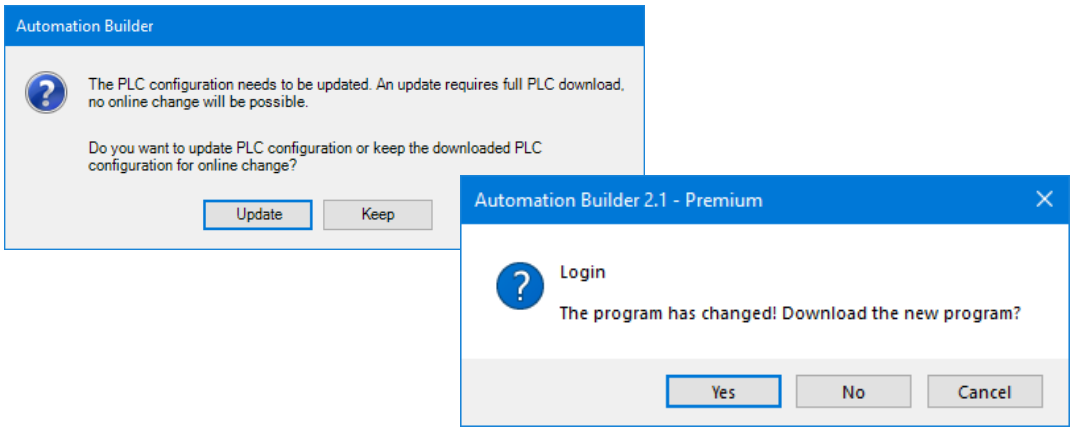
6.4. Connect to PLC using TCP/IP

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.



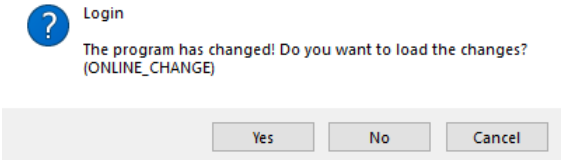
2. Control that the IP address for PC and the PLC is in the same network but not the same IP address. This can be checked by using through Ethernet Properties.
3. Click the icon “Login”, for building the configuration and checking if configuration is correct.

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.

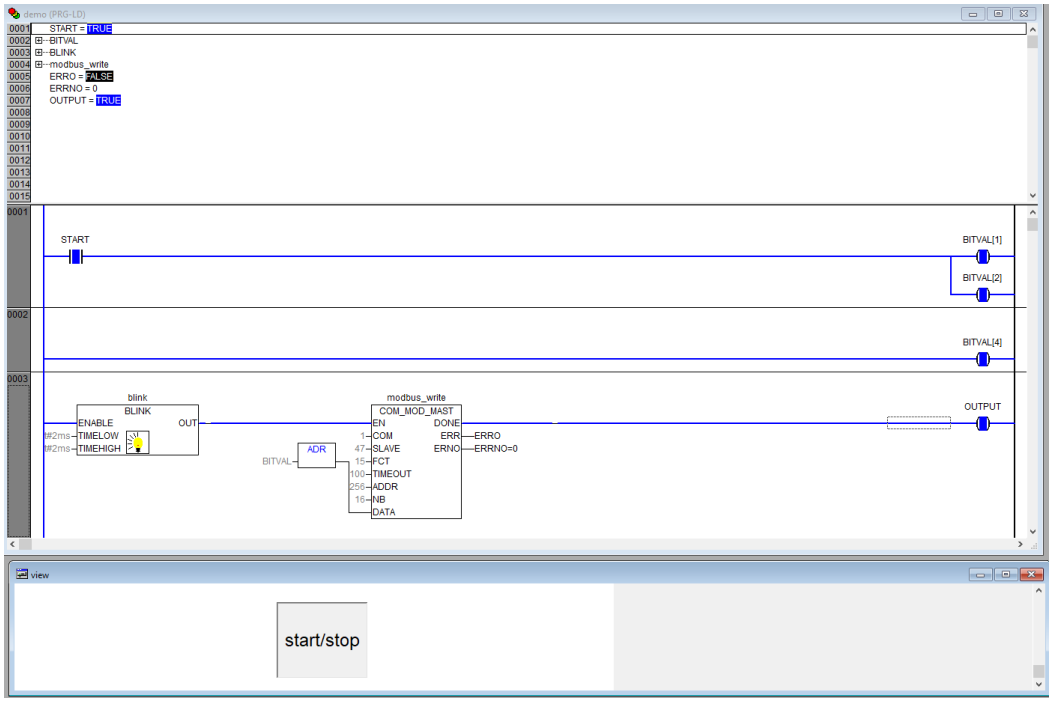


6.5. Build and run the PLC demo program

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



7. Contact us

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