AG03-1

Siemens S7-300®
PROFIBUS® Interface Module
for SIMATIC® Manager V5.5 + SP3
via HMS Anybus® Communicator™

Software Description

V1.0.0

Table of contents

1 General Notes 3
1.1 Trademarks 3
1.2 Liability 3
1.3 Limitations 3
1.4 Requirements 4
1.5 Versions Overview 4
1.6 List of Abbreviations 4

2 Hardware Configuration 5
2.1 Setup of Anybus® Communicator™ 5
2.1.1 ABC Configuration File 5
2.1.2 Transfer Configuration 5
2.1.3 ABC Cycle Time 6
2.2 Setup of PLC, PROFIBUS® and ABC 7
2.2.1 Add Your PLC to the Project 7
2.2.2 Register the GSD Device Description File for ABC 9
2.2.3 Add New Module to Your Hardware Configuration 10
2.2.4 Configure the Data Input / Output of ABC 11

3 Software Configuration 13
3.1 Import the SIKO-Library 13
3.1.1 Integrate the Symbole from the Library to Project 13
3.2 Add Required Function Blocks 16
3.3 Call SIKO Function Block Cyclically 16
3.3.1 Setup the Input and Output Addresses of the SIKO Function Block 17
3.4 Insert Variable Table for Testing 18
3.5 Complete the Project 18
3.6 Work with the Project 19

4 Communication Settings 20
4.1 Data Exchange 20
4.2 Parameter Access 20
4.2.1 Read Parameters 20
4.2.2 Write Parameters 20
4.2.3 Copy Parameters from Read to Write 20
4.3 S-Commands 20
4.4 Counter Value 21
4.4.1 Error Codes 22

5 Additional Information 23
5.1 Corresponding Node Address 23
5.2 Priority of “bcStartRead”, “bcStartWrite”, “bcStartCopy” or “bcResetCounterActivity” 23
5.3 Difference in SN5 Protocol Between PLC <-> ABC and ABC <-> SN5 device 23

1 General Notes

1.1 Trademarks

All trademarks or brand names including those protected for third parties shall unconditionally be subject to the provisions of the applicable laws governing trademarks and the proprietary rights of the registered owners. All trademarks, brand names or firm names are or may be trademarks or registered trademarks of their respective proprietors and are used only for description and unique identification. All rights not explicitly granted here are reserved.

Failure to explicitly identify trademarks used in this manual does not indicate that a name is free from rights of third parties.

Windows®, Windows® XP are trademarks of Microsoft® Corporation in the USA and other countries.

Anybus® is a registered trademark of HMS Industrial Networks AB, Sweden, USA, Germany and other countries.

SIMATIC® TIA Portal® and S7-300®, S7-400®, S7-1200®, S7-1500® are registered trademarks of Siemens AG.

PROFIBUS® and PROFINET® are registered trademarks of PROFIBUS and PROFINET International (PI).

1.2 Liability

SIKO GmbH assumes no warranty whatsoever regarding topicality, correctness, completeness or quality of the information or software products provided. All liability claims against SIKO GmbH referring to material or immaterial damages caused by using or not using the information or software provided or by using erroneous or incomplete information or software are always excluded.

1.3 Limitations

The library and its function were tested with SIMATIC® S7-300 CPU 6ES7 313-6CE01-0AB0. The interface module was engineered in STL using SIMATIC® Manager V5.5 + SP3.

The configuration file and its function were tested on an Anybus® Communicator™ AB7000. The file was setup using Anybus® Configuration Manager - Communicator RS232/422/485 version v.4.4.1.3 (Win 2000/XP/Vista/7).

1.4 Requirements

- Basic knowledge of handling and programming Siemens systems.
- Familiarity with PROFIBUS®.
- Basic knowledge of setup and handling Anybus® Communicator™
- Familiarity with Anybus® Configuration Manager

1.5 Versions Overview

This manual is related to

- AG03-1 firmware version >= 1.02
- Library “SIKO_SN5-PBvABC_LIB_V501”
- Know-how protected function block FB203 “SIKO_DRV_AG03-1vPB”
- Anybus® configuration file “SIKO_PB_20-SN5_pattern”
- Anybus® Communicator™ file “HMSB1803.gsd”

1.6 List of Abbreviations

FPB	PROFIBUS®	ABC	Anybus® Communicator™
SN5	SIKONETZ-5	ACM	Anybus® Configuration Manager
SW	Status Word	FB	Function Block
CW	Control Word	DB	Data Block

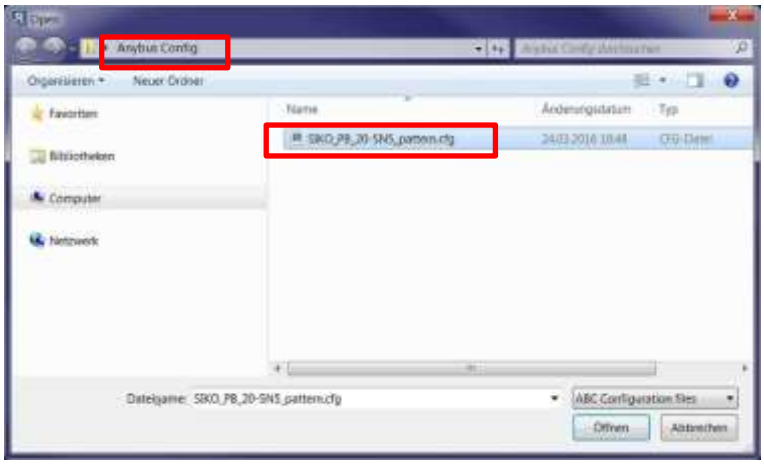
2 Hardware Configuration

2.1 Setup of Anybus® Communicator™

Please note, that the Anybus® configuration file is designed for a flexible SN5 participant numbers of minimum 1 to maximum 20. Later node address has to be modified according to your network requirements. Please consider a FPB address modification when reading further on.

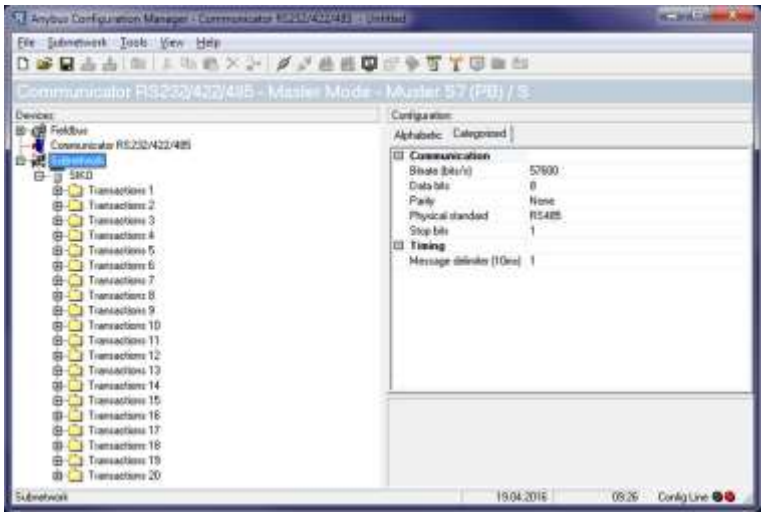
2.1.1 ABC Configuration File

Start ACM and select configuration file “SIKO_PB_20-SN5_pattern” from the folder “Anybus Config”.



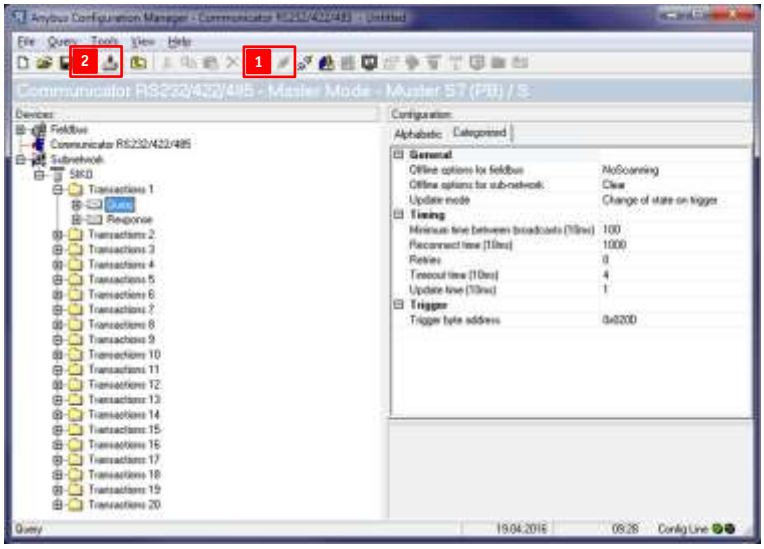
2.1.2 Transfer Configuration

Connect ABC with PC via “RS232 configuration cable” of Anybus accessories.



1. Press “Connect” to go online with the ABC.
2. Press “download to the ABC”.

You should have the following view:



2.1.3 ABC Cycle Time

Each transaction consists of “Query” and “Response”. The minimum time for a used transaction is about >= 10 ms. Each Subnetwork cycle has an overhand of about >= 100 ms.

Since the plc cycle time is asynchronous with ABC cycle time the FB integrates a trigger instead of checksum to the SN5 structure. When a master telegram is completed by the FB the trigger will be incremented. The corresponding transaction will be updated by change of state on trigger and exchanges the trigger with checksum byte before the SN5 telegram is send into subnetwork.
The subnetwork reply is checked and checksum byte is exchanged with incremented trigger byte before the telegram is send to plc by ABC. The FB will wait until a valid slave telegram is responded before sending a new telegram. Due to it the minimum time between update cycle amounts to >= 110 ms minimum. Please consider this for time critical applications!

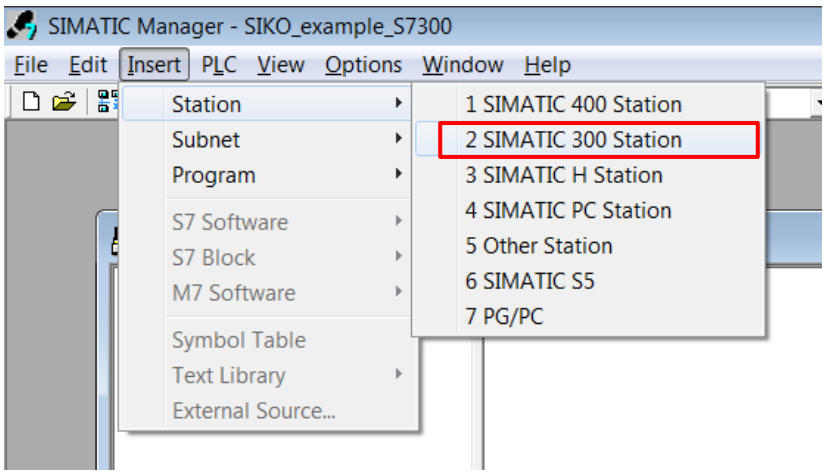
2.2 Setup of PLC, PROFIBUS® and ABC

Create New SIMATIC® Manager Project

1. Start the SIMATIC® Manger and create a new project: "File" > "New".
2. Enter the project name “SIKO_example_S7300”.
3. Choose a project path.
4. Execute the command “OK”.

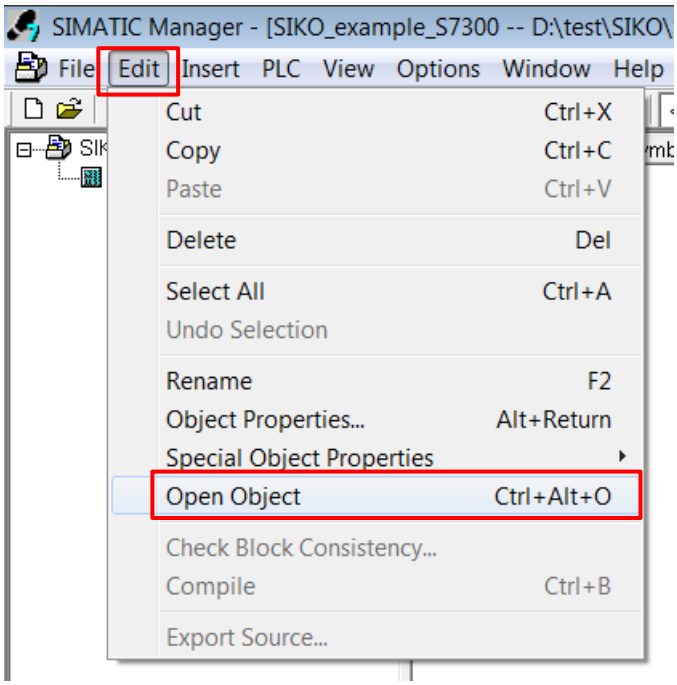
2.2.1 Add Your PLC to the Project

1. Insert a new S7-300 station by using the command
   "Insert" > "Station" > "SIMATIC 300 Station"



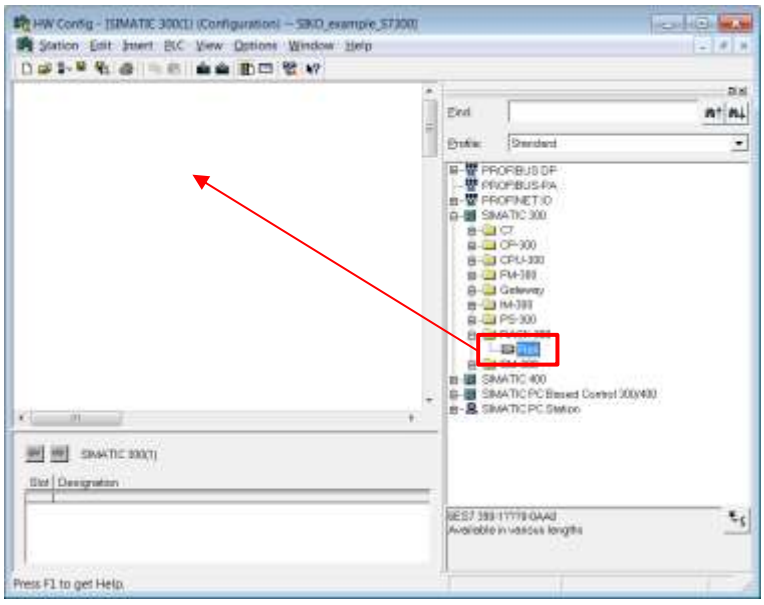
The new station appears below the project name.

2. Open the hardware configuration tool with command "Edit" > "Open Object"

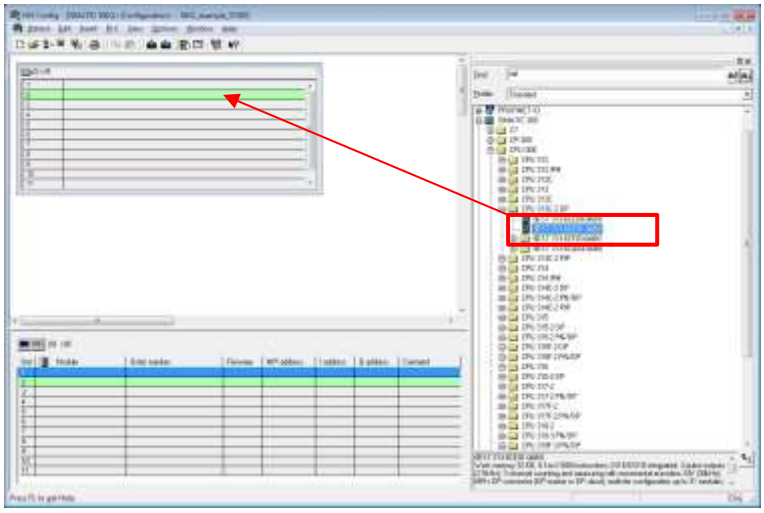


Now an empty hardware configuration window is opened.

3. Choose "Rail" from the folder "RACK-300" of the hardware catalog on the right side. Use drag and drop to move the "Rail" to the hardware configuration.



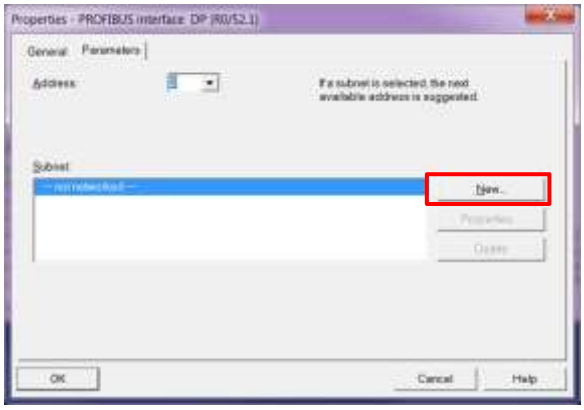
4. Choose the CPU from the hardware catalog and plug it into slot 2 of the rail.



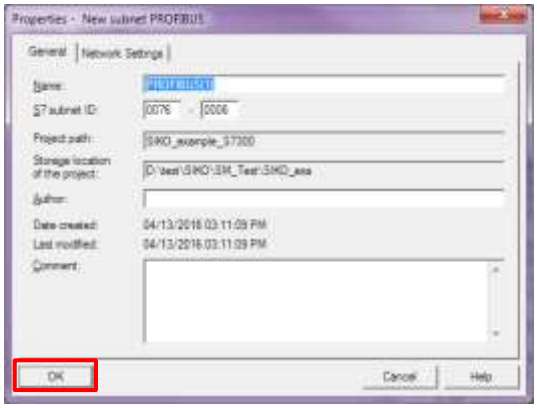
The Properties – PROFIBUS interface DP will open.

NOTICE	The Profibus Address in this example can cause problems under certain circumstances.

5. Create a new Subnet "New..."



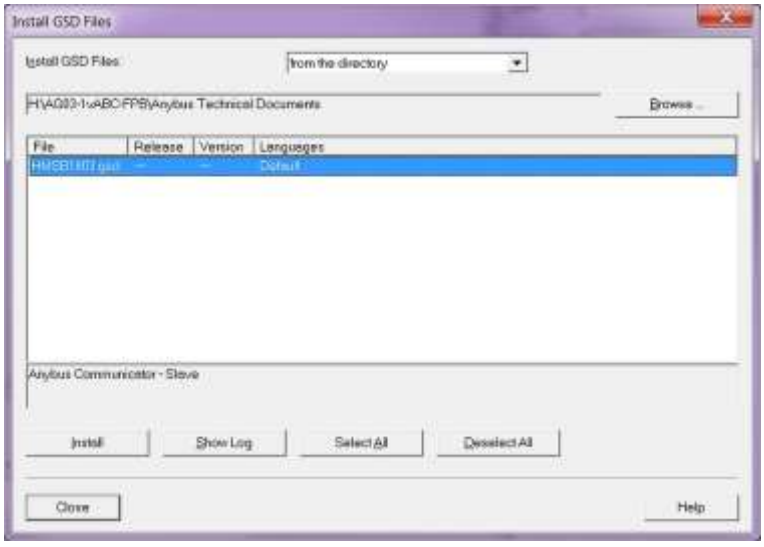
6. Confirm the settings with "OK".



7. Close the "Properties – PROFIBUS interface DP" window.

2.2.2 Register the GSD Device Description File for ABC

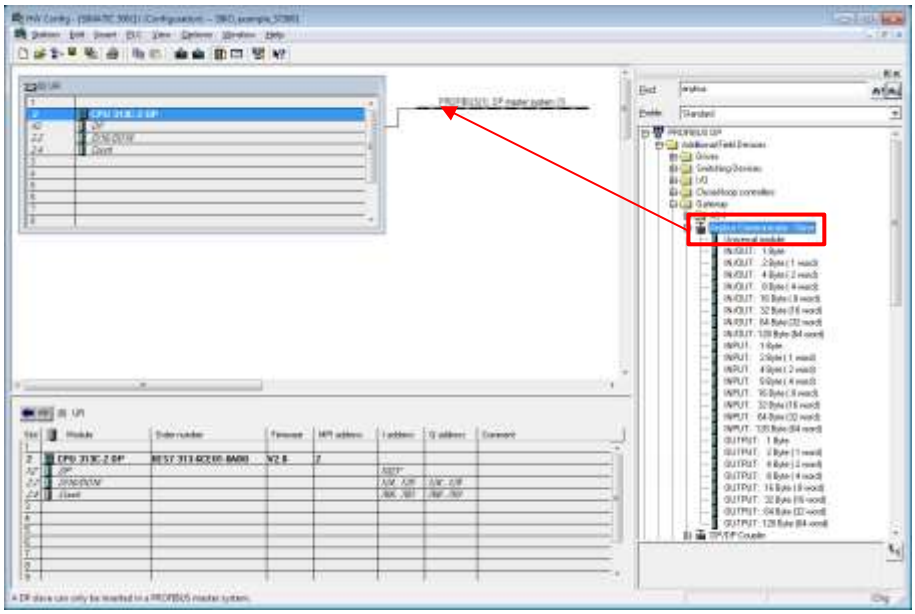
1. Install GSD for ABC via command "Options" > "Install GSD File..."
2. Browse to the storage location of the GSD file and choose the actual version.
3. Install the selected file.



After installation the ABC is available in the hardware catalog.

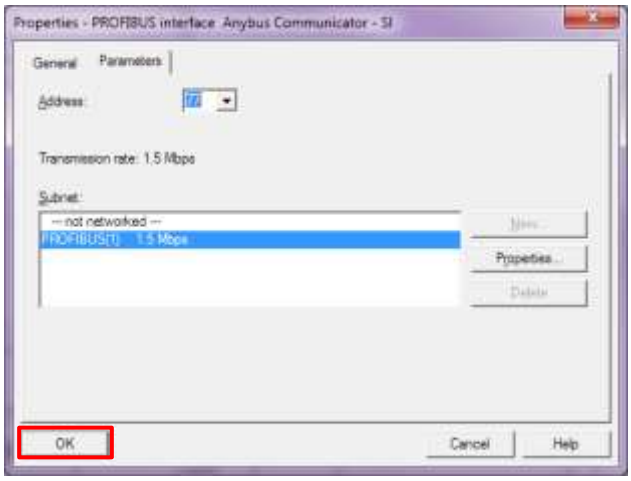
2.2.3 Add New Module to Your Hardware Configuration

1. Choose "Anybus Communicator Slave" within the folder "Gateway" from the hardware catalog and attach it to the PROFIBUS subnet using drag and drop.



The Properties – PROFIBUS interface Anybus Communicator SI will open.

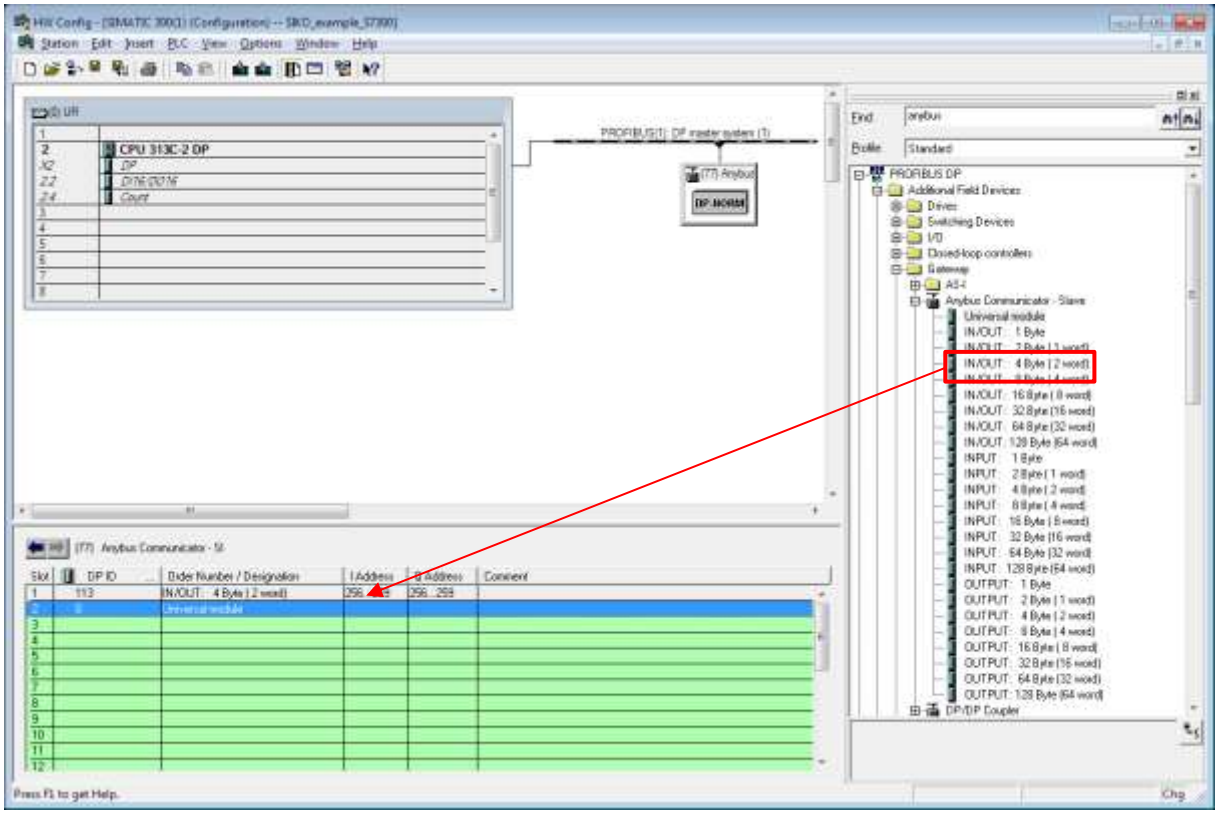
NOTICE	The Profibus Address in this example can cause problems under certain circumstances.



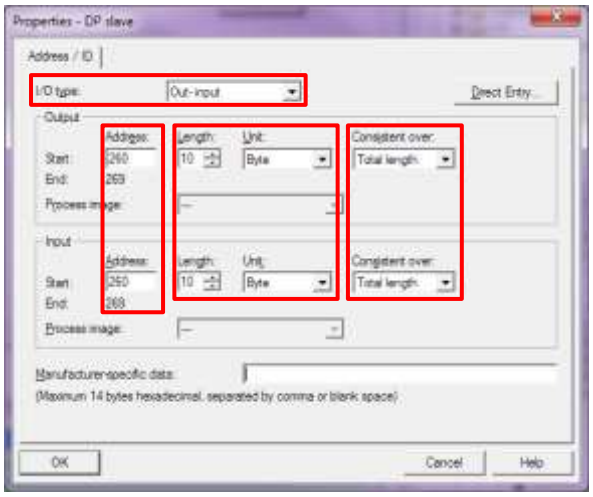
2. Enter a valid PROFIBUS® Address, “77” for example.
3. Confirm the settings with "OK".

2.2.4 Configure the Data Input / Output of ABC

1. Add "IN/OUT 4 Bytes (2 word)" from device “Anybus Communicator Slave” of the hardware catalog to the “Devices overview”.
2. Add "Universal module" from device “Anybus Communicator Slave” of the hardware catalog to the “Devices overview”.



3. Double click on “Universal module” to open “Properties DP slave”.



Setup the “I/O type” to “Out – input”.

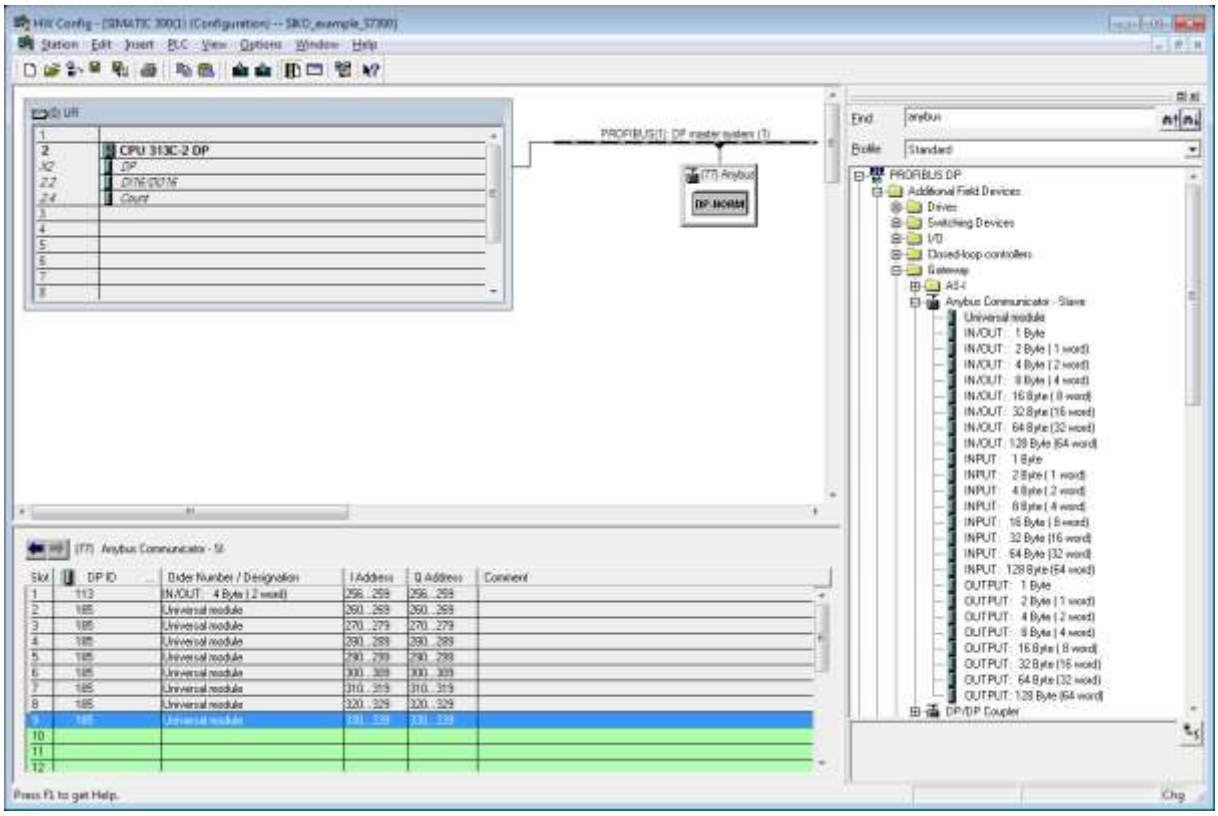
Choose the next “Address” after "IN/OUT 4 Bytes (2 word)".

Modify the “Length” to “10” of “Unit” “Byte”.

Select for “Consistent over” “Total length”.

4. Confirm the settings with "OK".

5. For each SN5 device a “Universal module” is required.
   Select the before modified “Universal Module”. Copy “Ctrl + C” and Paste “Ctrl + V” it to the required amount.



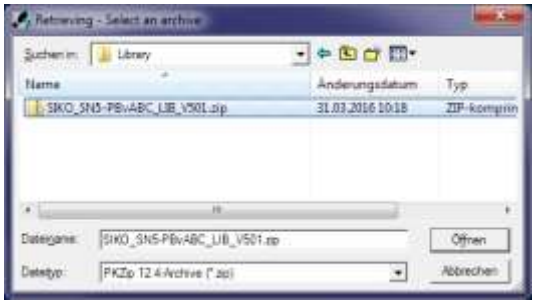
6. Save and compile the hardware configuration.
7. Close "HW Config".

The hardware configuration of the project is now complete.

3 Software Configuration

3.1 Import the SIKO-Library

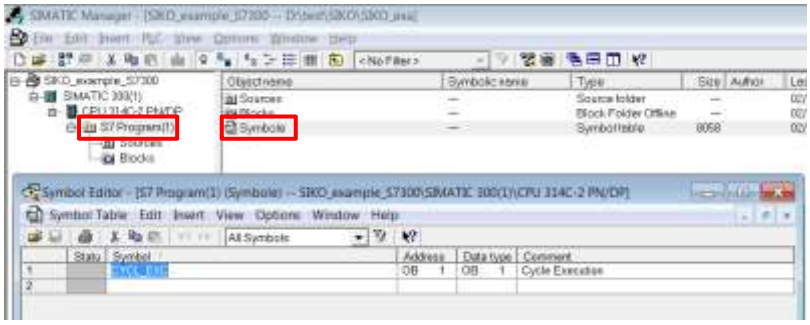
1. Execute command "File" > "Retrieve".
2. Browse to the storage location of the SIKO-library.
3. Select the archive SIKO_SN5-PBvABC_LIB_V501.zip.
4. Execute the command “Open”.



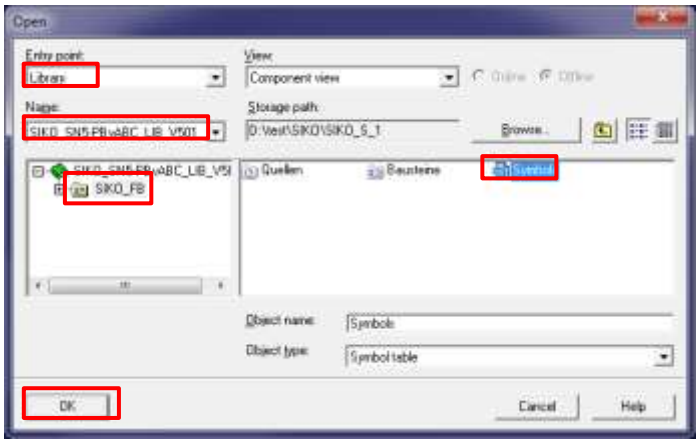
5. Select a destination directory accordingly and open it.

3.1.1 Integrate the Symbole from the Library to Project

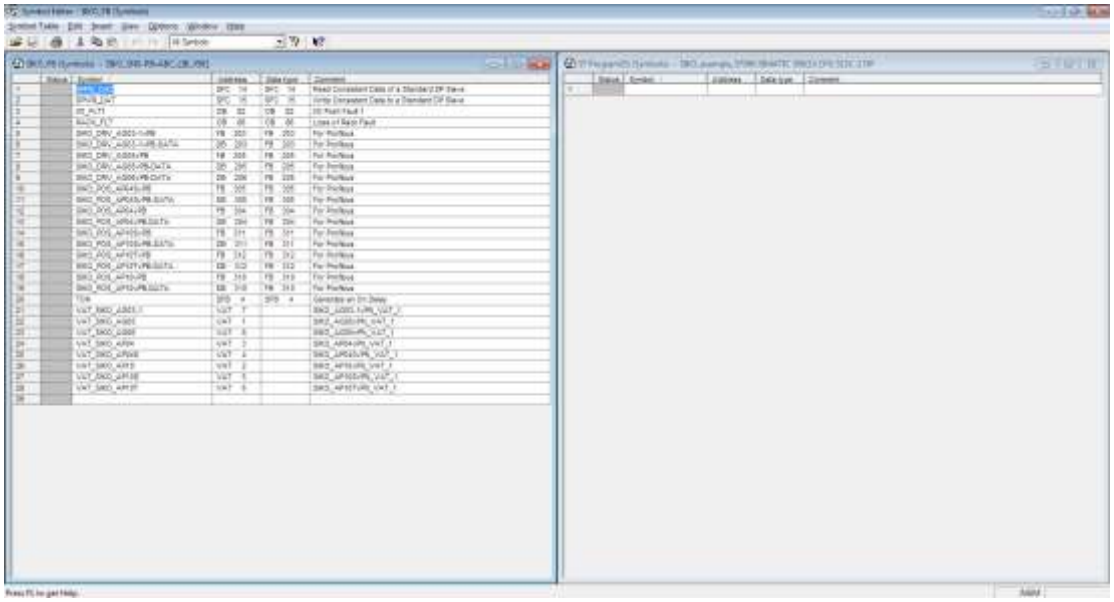
1. Navigate to the tab “S7 Program(1)” in the project tree
2. Double-click on the object “Symbole” in the right window



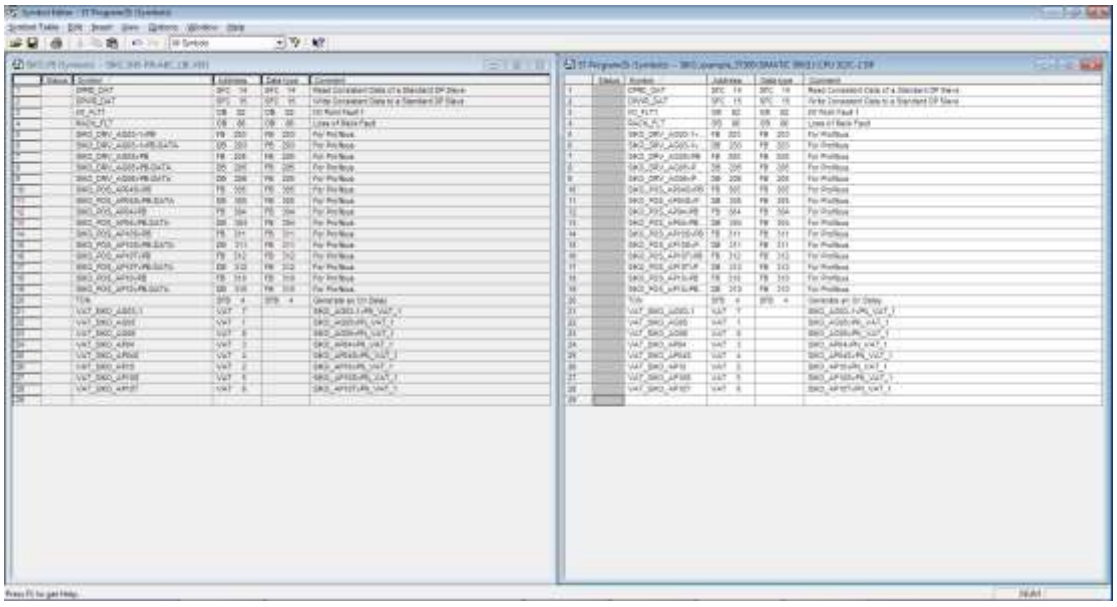
3. In the Symbole Editor execute command "Symbole" > "open".
4. Select the library SIKO_SN5-PBvABC_LIB_V501 and “SIKO_FB”.
5. Choose “Symbole” and confirm your selection with “OK”.



6. Execute command "Window" > "Arange" > “Vertically”.



7. Select ALL by the keystroke combination [**Ctrl**] + [**A**].
8. Copying it into the clipboard with the keystroke combination [**Ctrl**] + [**C**].
9. Click into line 2 (emty one) of your project symbole table and enter with the keystroke combination [**Ctrl**] + [**V**].
10. Delete the the red marked line.

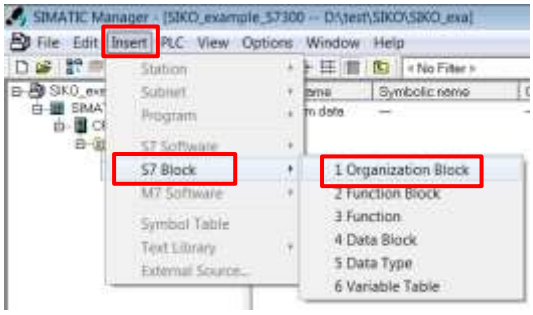


11. Close symbole table “SIKO_FB”.
12. Save and close your project symbole table.

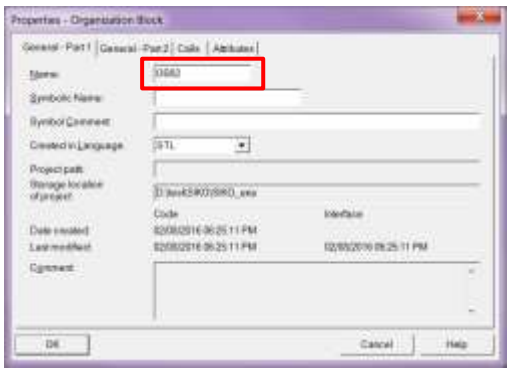
3.2 Add Required Function Blocks

Add OB82, OB86 and OB122 to the folder "Blocks".

1. Select folder "Blocks".
2. Execute command "Insert" > "S7 Block" > "1 Organization Block".



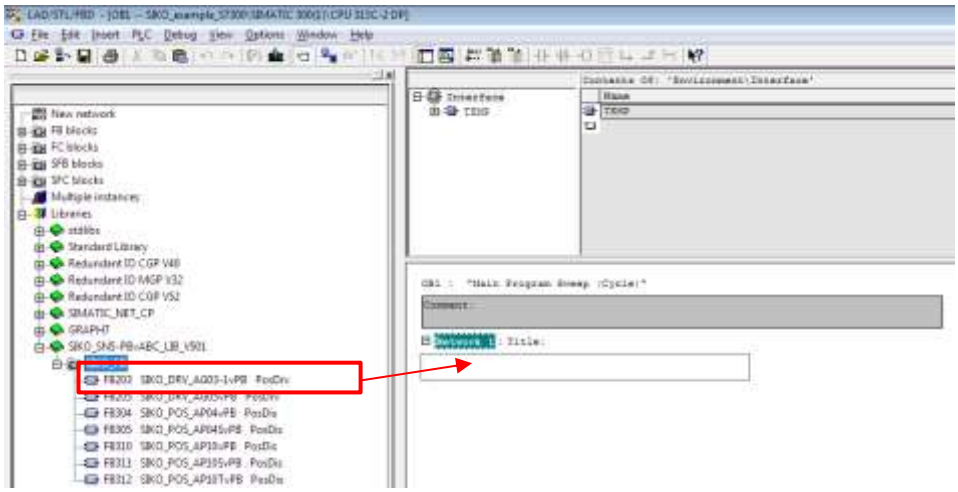
3. Enter organization block name "OB82".



4. Repeat steps 2 and 3 with block name "OB86" and "OB122".

3.3 Call SIKO Function Block Cyclically

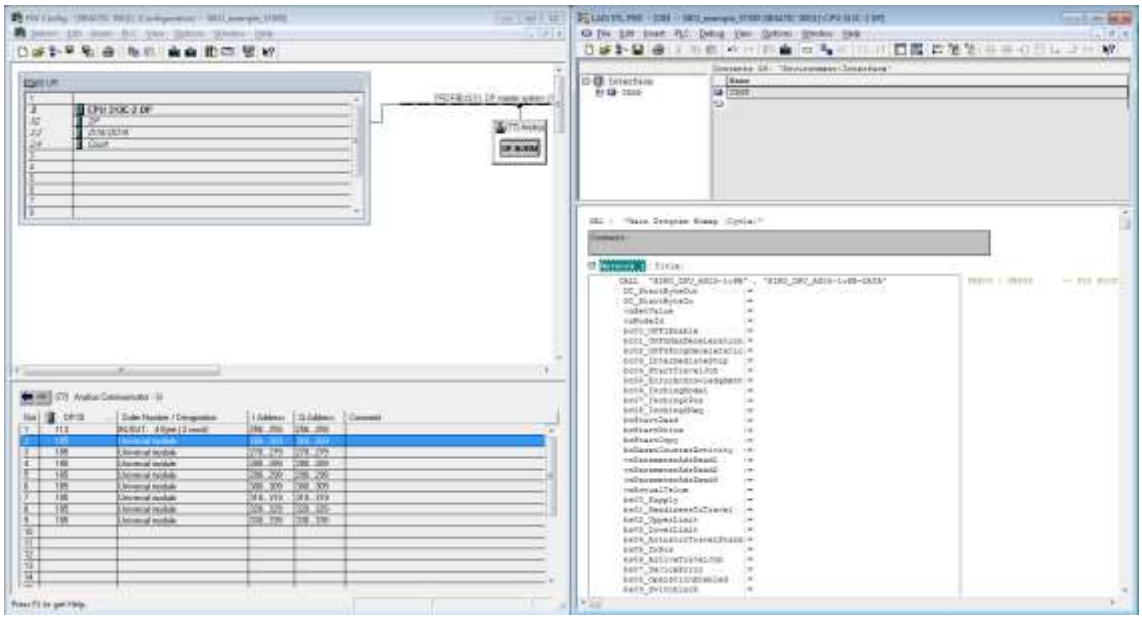
1. Double click on "OB1".
2. Add function block FB203 from library to main program.



3. Enter the name "DB203" for instance data block.
4. Confirm the message window / selection with "Yes", so the instance DB203 is generated.

3.3.1 Setup the Input and Output Addresses of the SIKO Function Block

To setup the input and output addresses of the function block the decimal address values from the hardware configuration have to be converted to hexadecimal values.



In this example the conversion table looks like this:

Slot 2 – Q address	260...269dec	DC_StartByteOut	W#16#104
Slot 2 – I address	260...269dec	DC_StartByteIn	W#16#104

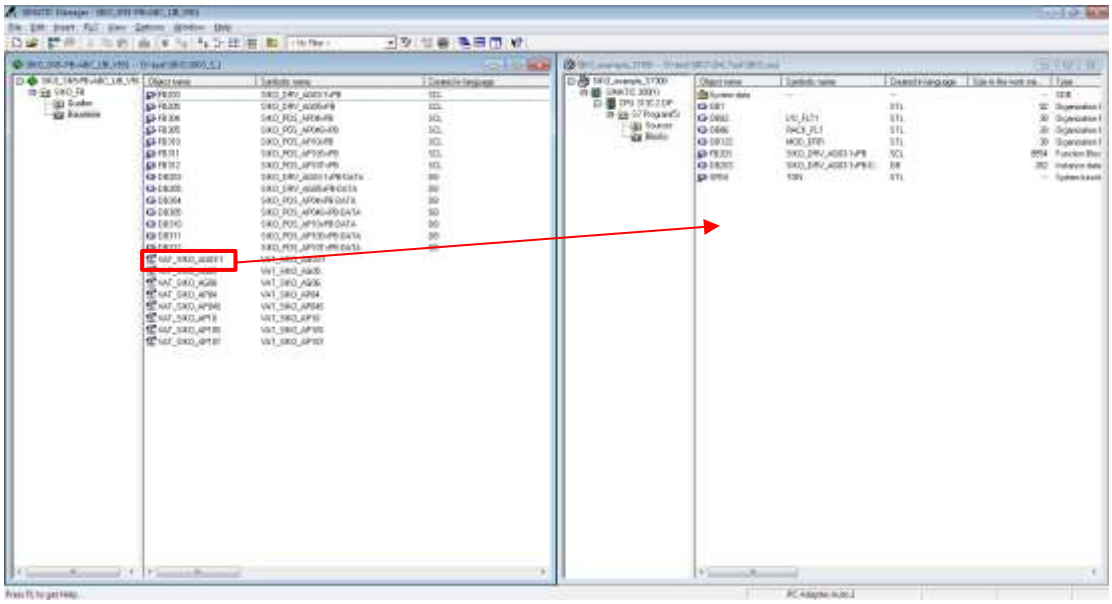
The input "nNodeId" of the function block must be connected with the set note address of the SN5 device.

In this example the address B#16#3 (3dec) is used.

After set up of the addresses save the settings and close the editior window and confirm "OK".

3.4 Insert Variable Table for Testing

1. While your project is still open execute command “File” > "open"
2. Select the library SIKO_SN5-PBvABC_LIB_V501
3. Execute command "Window" > "Arange" > “Vertically”.
4. Choose “VAT_SIKO_AG03-1” and copy it into your project using drag and drop.



5. Close library “SIKO_SN5-PBvABC_LIB_V501”.

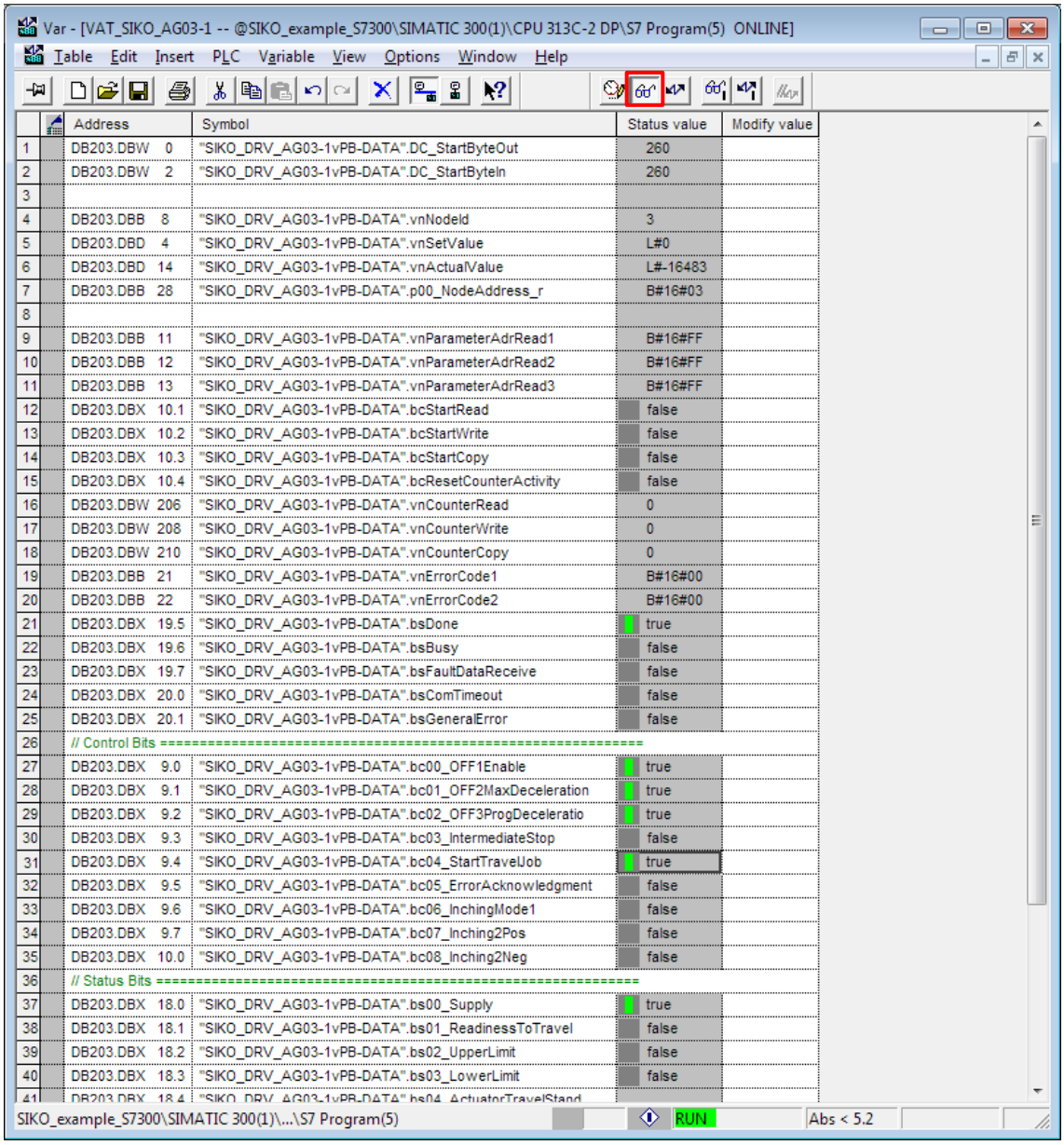
3.5 Complete the Project

1. Select folder "Blocks".
2. Click into the right window and select ALL by the keystroke combination [**Ctrl**] + [**A**].
3. Execute the command “Download”.

4. Confirm the following message boxes with “Yes” or “OK” respectively.

3.6 Work with the Project

1. Double-click on “VAT_SIKO_AG03-1”.
2. Enable the “Monitor” option in the VAT window.
3. Now you can control the SIKO-AG03-1 by setting the control bits.



	Address	Symbol	Status value	Modify value
1	DB203.DBW 0	"SIKO_DRV_AG03-1vPB-DATA".DC_StartByteOut	260	
2	DB203.DBW 2	"SIKO_DRV_AG03-1vPB-DATA".DC_StartByteIn	260	
3				
4	DB203.DBB 8	"SIKO_DRV_AG03-1vPB-DATA".vnNodeId	3	
5	DB203.DBD 4	"SIKO_DRV_AG03-1vPB-DATA".vnSetValue	L#0	
6	DB203.DBD 14	"SIKO_DRV_AG03-1vPB-DATA".vnActualValue	L#-16483	
7	DB203.DBB 28	"SIKO_DRV_AG03-1vPB-DATA".p00_NodeAddress_r	B#16#03	
8				
9	DB203.DBB 11	"SIKO_DRV_AG03-1vPB-DATA".vnParameterAdrRead1	B#16#FF	
10	DB203.DBB 12	"SIKO_DRV_AG03-1vPB-DATA".vnParameterAdrRead2	B#16#FF	
11	DB203.DBB 13	"SIKO_DRV_AG03-1vPB-DATA".vnParameterAdrRead3	B#16#FF	
12	DB203.DBX 10.1	"SIKO_DRV_AG03-1vPB-DATA".bcStartRead	false	
13	DB203.DBX 10.2	"SIKO_DRV_AG03-1vPB-DATA".bcStartWrite	false	
14	DB203.DBX 10.3	"SIKO_DRV_AG03-1vPB-DATA".bcStartCopy	false	
15	DB203.DBX 10.4	"SIKO_DRV_AG03-1vPB-DATA".bcResetCounterActivity	false	
16	DB203.DBW 206	"SIKO_DRV_AG03-1vPB-DATA".vnCounterRead	0	
17	DB203.DBW 208	"SIKO_DRV_AG03-1vPB-DATA".vnCounterWrite	0	
18	DB203.DBW 210	"SIKO_DRV_AG03-1vPB-DATA".vnCounterCopy	0	
19	DB203.DBB 21	"SIKO_DRV_AG03-1vPB-DATA".vnErrorCode1	B#16#00	
20	DB203.DBB 22	"SIKO_DRV_AG03-1vPB-DATA".vnErrorCode2	B#16#00	
21	DB203.DBX 19.5	"SIKO_DRV_AG03-1vPB-DATA".bsDone	true	
22	DB203.DBX 19.6	"SIKO_DRV_AG03-1vPB-DATA".bsBusy	false	
23	DB203.DBX 19.7	"SIKO_DRV_AG03-1vPB-DATA".bsFaultDataReceive	false	
24	DB203.DBX 20.0	"SIKO_DRV_AG03-1vPB-DATA".bsComTimeout	false	
25	DB203.DBX 20.1	"SIKO_DRV_AG03-1vPB-DATA".bsGeneralError	false	
26	// Control Bits ==========			
27	DB203.DBX 9.0	"SIKO_DRV_AG03-1vPB-DATA".bc00_OFF1Enable	true	
28	DB203.DBX 9.1	"SIKO_DRV_AG03-1vPB-DATA".bc01_OFF2MaxDeceleration	true	
29	DB203.DBX 9.2	"SIKO_DRV_AG03-1vPB-DATA".bc02_OFF3ProgDeceleratio	true	
30	DB203.DBX 9.3	"SIKO_DRV_AG03-1vPB-DATA".bc03_IntermediateStop	false	
31	DB203.DBX 9.4	"SIKO_DRV_AG03-1vPB-DATA".bc04_StartTravelJob	true	
32	DB203.DBX 9.5	"SIKO_DRV_AG03-1vPB-DATA".bc05_ErrorAcknowledgment	false	
33	DB203.DBX 9.6	"SIKO_DRV_AG03-1vPB-DATA".bc06_InchingMode1	false	
34	DB203.DBX 9.7	"SIKO_DRV_AG03-1vPB-DATA".bc07_Inching2Pos	false	
35	DB203.DBX 10.0	"SIKO_DRV_AG03-1vPB-DATA".bc08_Inching2Neg	false	
36	// Status Bits ==========			
37	DB203.DBX 18.0	"SIKO_DRV_AG03-1vPB-DATA".bs00_Supply	true	
38	DB203.DBX 18.1	"SIKO_DRV_AG03-1vPB-DATA".bs01_ReadinessToTravel	false	
39	DB203.DBX 18.2	"SIKO_DRV_AG03-1vPB-DATA".bs02_UpperLimit	false	
40	DB203.DBX 18.3	"SIKO_DRV_AG03-1vPB-DATA".bs03_LowerLimit	false	
41	DB203.DBX 18.4	"SIKO_DRV_AG03-1vPB-DATA".bs04_ActuatorTravelStand		

4 Communication Settings

4.1 Data Exchange

The FB is designed to send or receive in alternation the “nSetValue” (Write, Parameter: 0xFF “Set Point ”) or the “nActualValue” (Read, Parameter: 0xFE “Actual Position”) respectively, while no specific parameter access is active.

With the “nParameterAdrRead1”, “nParameterAdrRead2” and “nParameterAdrRead3” further parameter can be included in the data read cycle. With default value 0xFE the inclution is disabled.

NOTE: If “bsFaultDataReceive” is indicated the complete data exchange is stopped, while Control and Status Word are still updated! A missing or not responsing subnetwork participant is indicated by “bsComTimeout” (0,5sec. + time set in parameter 0x02 Bus Timeout).

4.2 Parameter Access

The present module contains the parameter data in addition to the process data (CW/SW). Parameters that can be changed (read/write) exist in programming as actual value (_r) and as target value (_w) as well. Furthermore, it is differentiated between pure read parameters (only indicated as actual value) and pure write parameters (only indicated as target value).

A rising edge must be applied either to the “bcStartRead” or to the “bcStartWrite” input on the module described here in order to enable a read or write process of one of the variables.

4.2.1 Read Parameters

If a rising edge is applied to the ”bcStartRead” input, then all parameters will be read and can be used for further programming. If counter read value is not reset to “0” the read cycle was interrupted by read failure. This indicates to a communication failure.

4.2.2 Write Parameters

If a rising edge is applied to the “bcStartWrite” input of the module, then all parameters will be transferred to the module. If counter write value is not reset to “0” the write cycle was interrupted by a write failure. This indicates to a communication failure or parameter value is beyond range of value accepted by AG03-1.

4.2.3 Copy Parameters from Read to Write

If a rising edge is applied to the “bcStartCopy” input of the module, then all actual values (_r) are copied to their corresponding target values (_w).

4.3 S-Commands

After executing a S-Command a read cycle is been triggered to refresh all actual values (_r).

4.4 Counter Value

Count read value	Count write value	Name	Value range (dec)	Default
	1	0xA8 Programming Mode On/Off	0 ... 1	0
1		0x00 Note address	0 ... 31	1
2		0x01 Baud rate	0 ... 2	1
3	2	0x02 Bus Timeout	0 ... 20	20
4	3	0x03 Response parameter to a setpoint write access	0 ... 2	0
5	4	0x0E Configuration programming mode	0 ... 1	0
6	5	0x10 Controller Parameter P	1 ... 500	100
7	6	0x11 Controller Parameter I	0 ... 500	5
8	7	0x12 Controller Parameter D	0 ... 500	0
9	8	0x13 a-Pos	1 ... 100	50
10	9	0x14 v-Pos	Gear 24:1 => 1 ... 200 48:1 => 1 ... 100	30
11	10	0x15 a-Rot	1 ... 100	50
12	11	0x16 a-Inch	1 ... 100	50
13	12	0x17 v-Inch	Gear 24:1 => 1 ... 200 48:1 => 1 ... 100	30
14	13	0x18 UeNumerator	1 ... 10000	1
15	14	0x19 UeDenominator	1 ... 10000	1
16		0x1A EncoderResolution		1600
17	15	0x1B Counting direction	0 ... 1	0
18	16	0x1C Resolution per revolution	1 ... 59999	720
19	17	0x1E Offset value	-9999 ... 9999	0
20	18	0x1F Calibration value	-9999 ... 9999	0
21	19	0x20 Target window1 (near field)	0 ... 9999	10
22	20	0x21 Positioning type (loop type)	0 ... 2	0
23	21	0x22 Loop length	0 ... 9999	800
24	22	0x23 Inpos Mode	0 ... 2	0
25	23	0x24 Delta Inch	-1000000 ... 1000000	1600
26	24	0x25 AccelerationTypeInchingMode2	0 ... 1	0
27	25	0x26 Inching2 Offset	10 ... 100	100
28	26	0x27 Stop Mode Inching2	0 ... 1	0
29	27	0x28 Operating mode	0 ... 1	0
30	28	0x29 Limit_1	-9999999 ... 9999999	10000000
31	29	0x2A Limit_2	-9999999 ... 9999999	-10000000
32	30	0x2D Contouring Error Limit	1 ... 30000	400
33	31	0x36 d-Pos	1 ... 101	101

Count read value	Count write value	Name	Value range (dec)	Default
34	32	0x37 Torque Disable	20 ... 125	125
35		0x60 Output Stage Temperature		0
36		0x61 Voltage Of Control		0
37		0x62 Voltage Of Output Stage		0
38		0x63 Battery voltage		0
39		0x64 Motor Current		0
40		0x65 DeviceCode		0
41		0x66 SoftwareSecondaryController		0
42		0x67 Software Main Controller		0
43		0x68 Serial Number		0
44		0x69 Production Date		0
45		0x6A Gear Reduction		0
46		0x6B Actual Position		0
47		0x6C Actual Rotational Speed		0
48		0x80 Number Of Errors Recorded		0
49		0x81 Error 01		0
50		0x82 Error 02		0
51		0x83 Error 03		0
52		0x84 Error 04		0
53		0x85 Error 05		0
54		0x86 Error 06		0
55		0x87 Error 07		0
56		0x88 Error 08		0
57		0x89 Error 09		0
58		0x8A Error 10		0
59	33	0xAA FreezeAV	0 ... 1	0
60		0xFA System Status word		0
61		0xFE Position value		0
62		0xFF Setpoint		0
	34	0xA0 System Command	1 ... 3, 5 ... 9	0

4.4.1 Error Codes

If a communication error occurs, there is an error code present at the outputs “nErrorCode1” and “nErrorCode2”. Please refer to the AG03-1 manual (keyword: error codes) for a complete description of these error codes.

5 Additional Information

5.1 Corresponding Node Address

Since the different SN5 devices can be commanded to change the actual used node address, make sure, that before a “bcStartWrite” is executed, “vnNodeId” (temporary address) and “p00_NodeAddress_w” (set address) are adjusted accordingly.
Otherwise a lost of communication after cold start or software reset is possible!

5.2 Priority of “bcStartRead”, “bcStartWrite”, “bcStartCopy” or “bcResetCounterActivity”

If one of “bcStartRead”, “bcStartWrite”, “bcStartCopy” is high/true the others are blocked, except “bcResetCounterActivity”. The priority is: “bcStartRead”, “bcStartWrite”, “bcStartCopy”.

There is no timing or device condition to execute either “bcStartRead” or “bcStartCopy”. For execute “bcStartWrite” the device must be without movement and “vnCounterRead” and “vnCounterCopy” are inactive.
Otherwise “bcStartWrite” is not executed or delayed!

If “bcResetCounterActivity” is high/true ALL counters are set to “0” as well as the internal status of “bcStartRead”, “bcStartWrite”, “bcStartCopy”.

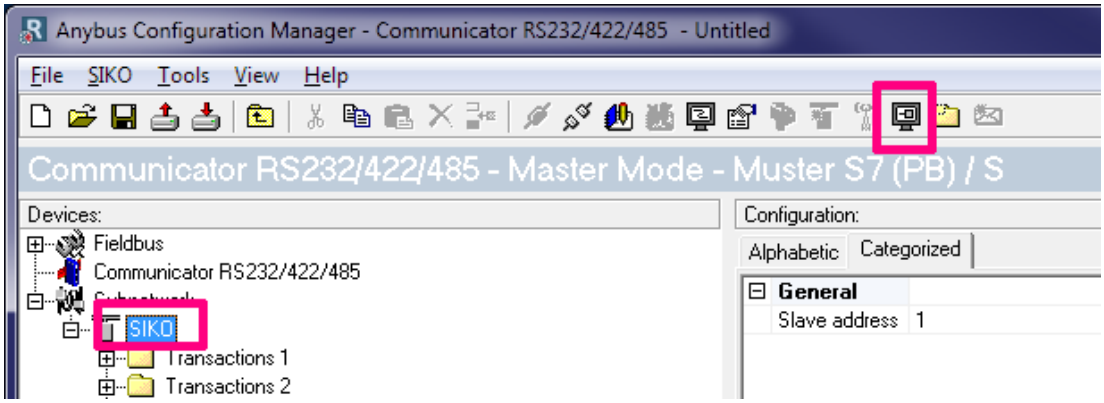
5.3 Difference in SN5 Protocol Between PLC <-> ABC and ABC <-> SN5 device

The FB is creating a protocol string according to the device manual but with one network required modification, which is a trigger byte instead of checksum byte at the protocol end.

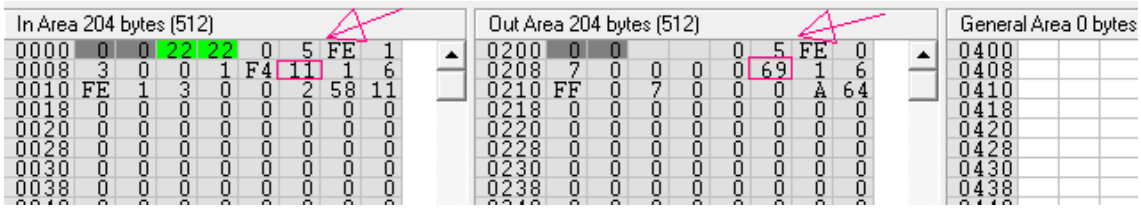
This is required since the plc cycle and update time is much higher than from the ABC. Every time the FB has completed the string, it increases the trigger value by one. The ABC configuration is designed to check this plc trigger. Every time the ABC application update cycle comes together with the plc trigger change, the ABC configuration exchanges the trigger byte with a checksum byte and sends the SN5 protocol into the subnet.

The addressed SN5 slave replies with a respectively protocol and the ABC performs the checksum test. If positively checked, the ABC exchanges the checksum byte with a trigger byte and transfers the protocol string to the master plc. This trigger is monitored (fix 500 ms) and used together with “p02_BusTimeout_r” to create “bsComTimeout”.

The protocol string with trigger byte recived from and transmitted to plc can be viewed in the ABC “Node Monitor”. Go online with ABC while application runs. Click the node underneath “Subnet”. Then start the “Node Monitor” which should be selectable now:



You should have similar few:



In the “Out Area …” you can see what is send from plc. In this example the ID5 is requested for current position.
In the “In Area …” you can see what comes from subnet. In this example the ID5 replies current position (1F4 (h) = 500 (d)).
To the ID 6 the target position (258 (h) = 600 (d)) is send and confirmed.

Please keep in mind, that the 10th Byte is not checksum but trigger value!