

Module 4

Documentation
Key Concept Product development



Course materials

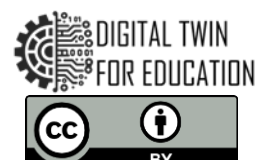
Digital twin documentation
Key concept for product development

Status: Jun 2026

Author: Dr. K. Krettenauer
C. Schmid
B. Geibel



2024-1-DE02-KA210-VET-000246677
www.digitwin4teach.eu



Module 4

Documentation Key Concept Product development



Table of Contents

0	Model analysis with task	4
13	Importing a symbol table for variable names.....	7
14	Selection and definition of parameters for the inputs in NX.....	10
15	Selection and definition of parameters for outputs in NX	12
16	Assignment of signals in the signal adapter	14
17	Connecting parameters and signals in the signal adapter	17
18	Connecting input parameters and signals in the signal adapter	20
19	Simulation release in the TIA Portal	23
20	Create PLC hardware and integrate PLC variables	28
21	Creating a function block in the TIA Portal.....	31
22	Program creation in the function block.....	33
23	Call the FB in Main OB1 and save the program	35
24	Creating a new simulation instance in PLCSim Advanced	37
25	Load TIA program in PLC Sim	39
26	Creating signal connections in MCD	42
27	Temporary storage and testing of the software status in NX	47
28	Testing the software status in the TIA Portal	48
29	Creation of a step chain program	50
30	Control of the outputs of the step chain program	51
31	Testing the software version with NX.....	56
32	Save status and close entire project.....	57
33	Reopen the SimCaer virtual PLC	59
34	Reopen the NX simulation project.....	59
35	Integration of a stop button in the NX simulation	59
36	Integration of a text description in NX	60

Module 4

Documentation Key Concept Product development



37	Embed the function runtime button for the stop signal	60
38	Adjustment for the stop signal in the signal adapter	60
39	Adjustment for the stop signal in the signal connection	61
40	Adjustments in the TIA Program for the stop signal	61
41	Environment test of the stop button in the NX simulation	62
42	Integration of two new collision sensors in the NX simulation	63
43	Adjustment for the collision sensors in the signal adapter	63
44	Integration of a source and sink for the production parts	63
45	Adjustment for workpiece source in the signal adapter	64
46	Adjustment for a rising edge in the signal adapter	64

Note: The numbering of the captions starts with 13 because of the continuing of the process chain. In Module 3 you already made the steps 1 to 12.

Module 4

Documentation Key Concept Product development

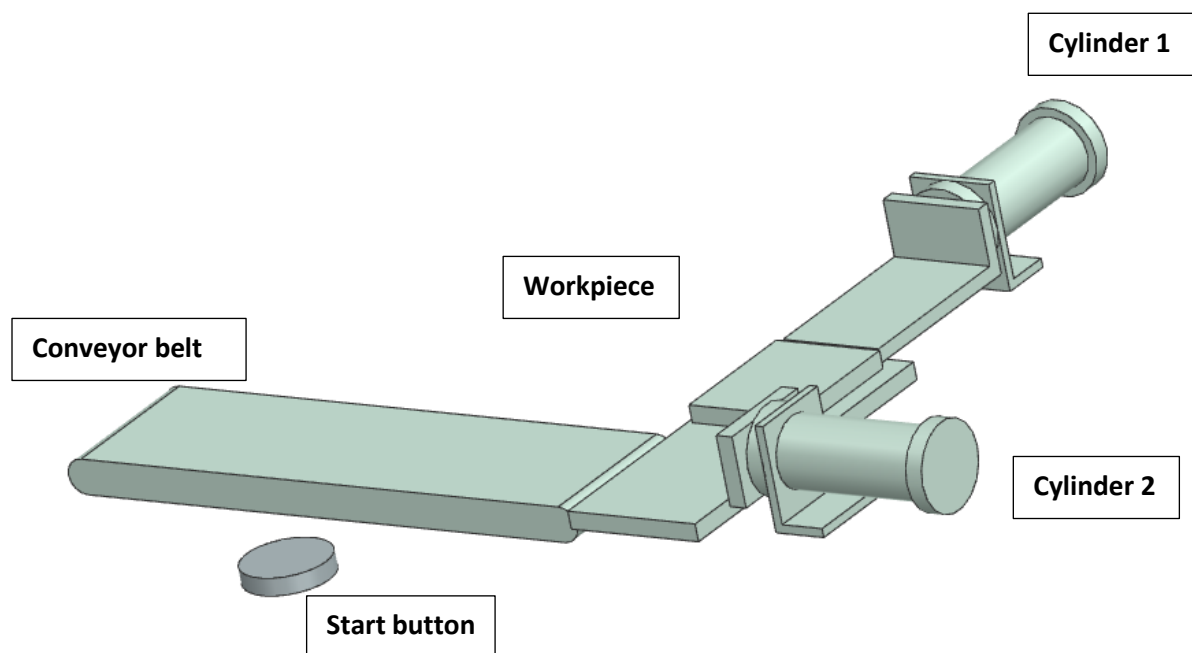


0 Model analysis with task

The NX model used for this exercise contains two pneumatic cylinders and a conveyor belt.

The task of the system is to first push a workpiece (rectangular body) laying on a plate with two pneumatic cylinders and then transport the workpiece away using the conveyor belt.

An additional start button can be used to initiate a sequence of steps for the sequence control. The double-acting pneumatic cylinders are controlled via solenoid valves and their position is monitored by limit switches.



In the following work process, the already existing CAD model (assembly in STEP file format) is imported to Siemens NX MCD .

Module 4

Documentation Key Concept Product development

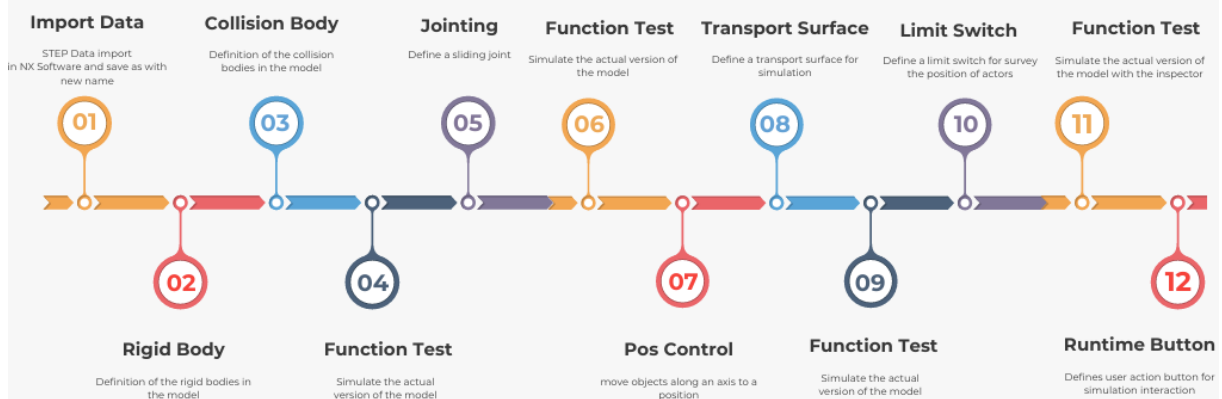


Parts of the model will be made „physically“ movable so that the model can be linked to a control system (in our case PLC SIM advanced from SIEMENS) and the associated programming environment (in our case TIA-Portal from SIEMENS).

The whole work process is divided into five sections.

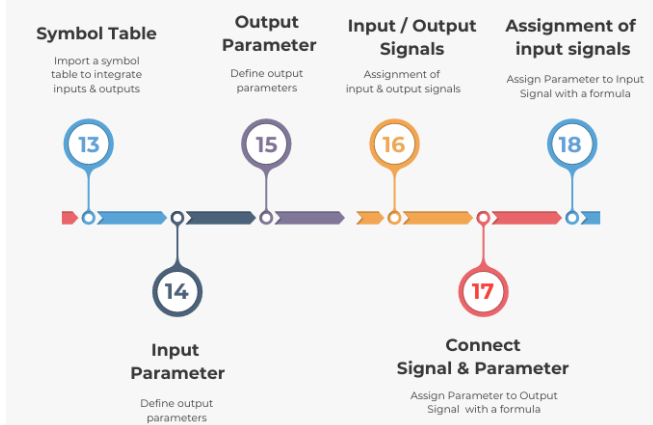
1

ADD PHYSICAL PROPERTIES TO NX MODEL



2

SIGNALS IN NX



3

CREATE A NEW TIA PROGRAM



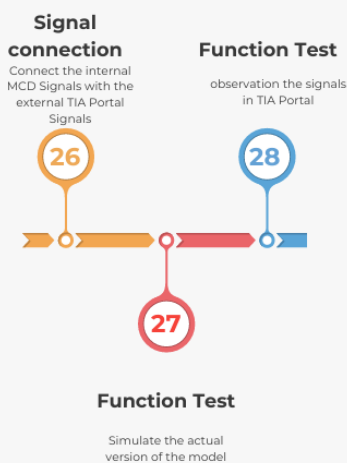
Module 4

Documentation
Key Concept Product development



4

CONNECTION PRETEST



5

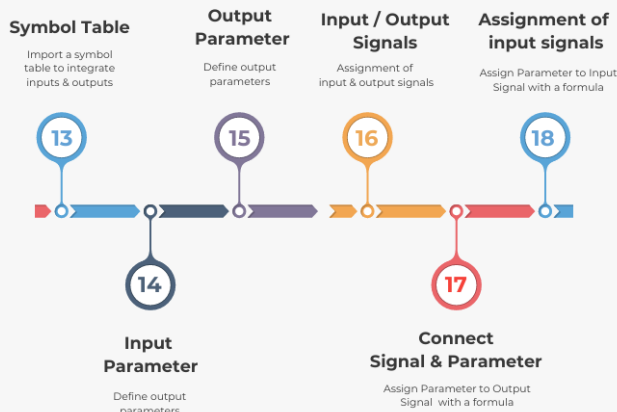
STEP CHAIN AND FUNCTION TEST



Part 2

2

SIGNALS IN NX



A crucial part of working with NX MCD is the creation of signals and their assignment.

Movements of actuators and sensor signals are assigned to parameters in order to be connected to PLC signals later on.

This mapping takes place in the **Signal Adapter**, which forms the core of communication between the digital model and the PLC Software.

Module 4

Documentation Key Concept Product development

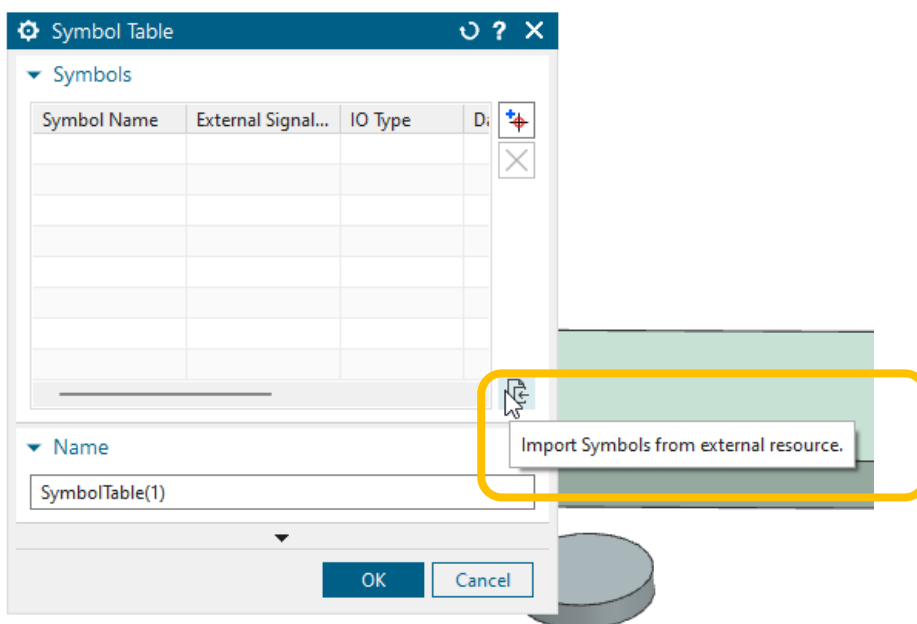
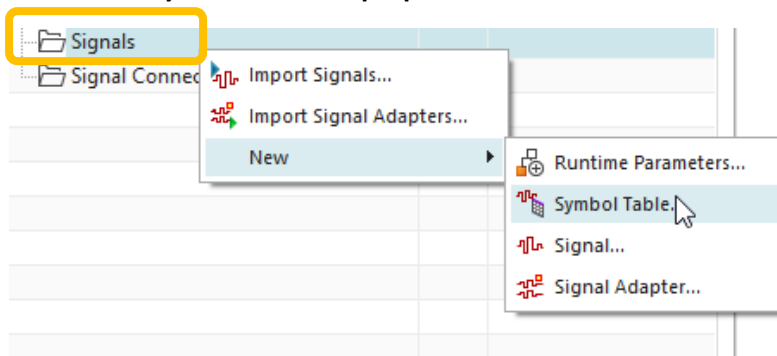


13 Importing a symbol table for variable names

Video: 13Import Symbols

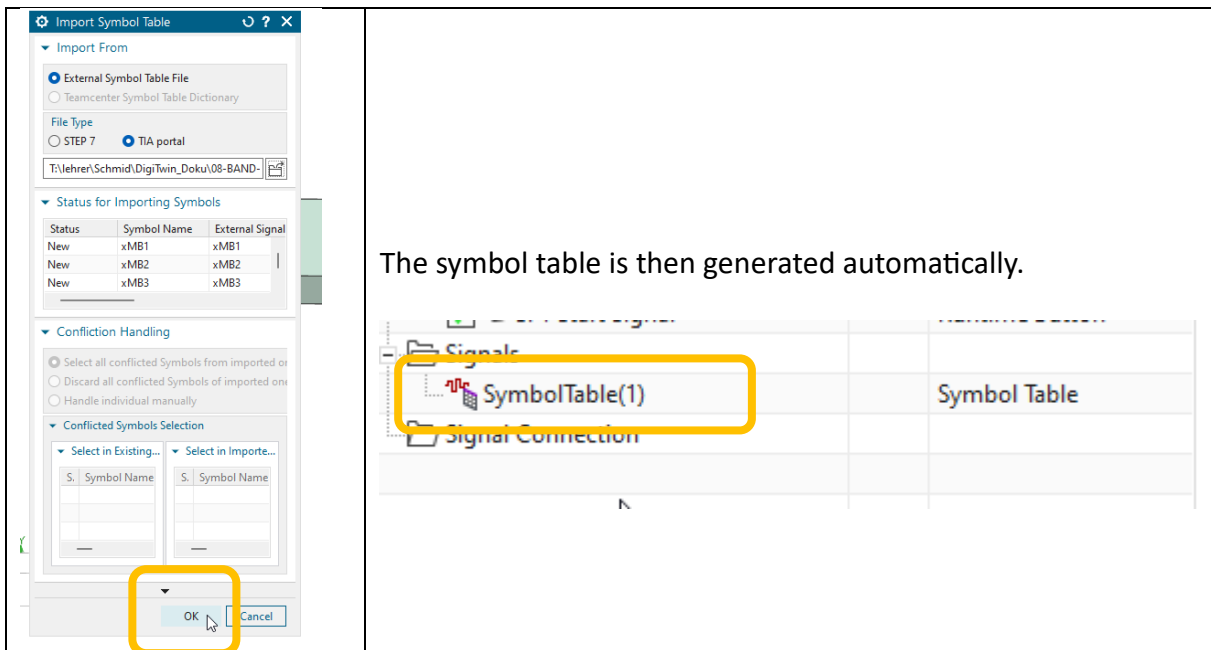
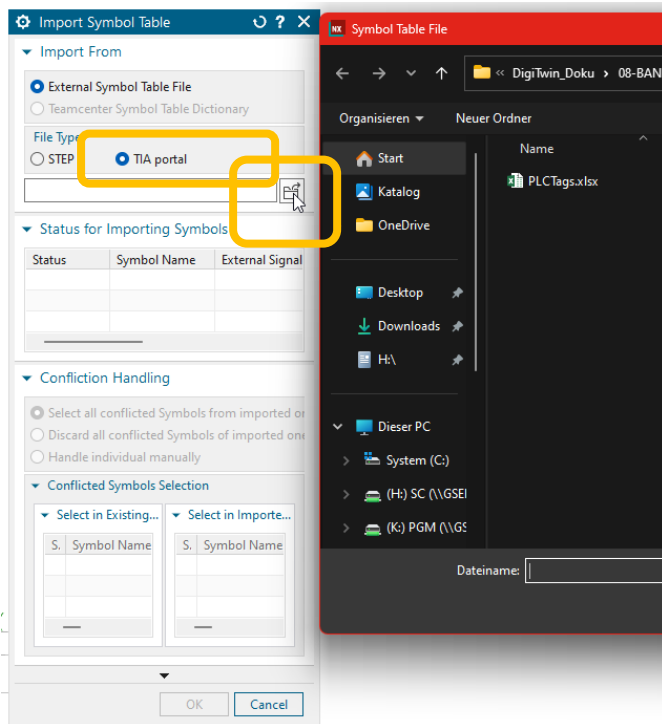
The next step is to import a symbol table containing the variable names of the PLC variables from the TIA Portal project in order to facilitate the signal mapping with the PLC program later on.

If, as in our example, the naming structure is identical, the variable assignment can be done automatically for simulation purposes.



Module 4

Documentation Key Concept Product development



Module 4

Documentation Key Concept Product development



In the symbol table in the NX model, you can see that the assignment for MB1 (solenoid valve), for example, is declared as an input, and for sensors such as BG1 (end position monitoring), it is declared as an output.

The reason for this is that the variables are viewed from the perspective of the NX model, and in this case, a sensor is an output because it delivers a signal to the controller, and the solenoid valve is an input because the signal comes into the NX model as an input signal from the PLC.

The screenshot shows a 'Symbol Table' dialog box with a table of symbols. The table has five columns: Symbol Name, External Signal..., IO Type, Data Type, and Comment. The symbols are listed in two groups: MB (Motor/Valve) and BG/SF (Sensors/Status Functions). MB symbols are marked as 'Input' and BG/SF symbols are marked as 'Output'. All data types are 'bool'. At the bottom of the dialog, there is a 'Name' field containing 'SymbolTable(1)' and 'OK' and 'Cancel' buttons.

Symbol Name	External Signal...	IO Type	Data Type	Comment
xMB1	xMB1	Input	bool	
xMB2	xMB2	Input	bool	
xMB3	xMB3	Input	bool	
xMB4	xMB4	Input	bool	
xMB5	xMB5	Input	bool	
xMB6	xMB6	Input	bool	
xMB7	xMB7	Input	bool	
xMB8	xMB8	Input	bool	
xPF1	xPF1	Input	bool	
xPF2	xPF2	Input	bool	
xPF3	xPF3	Input	bool	
xPF4	xPF4	Input	bool	
xPF5	xPF5	Input	bool	
xPF6	xPF6	Input	bool	
xPF7	xPF7	Input	bool	
xPF8	xPF8	Input	bool	
xBG1	xBG1	Output	bool	
xBG2	xBG2	Output	bool	
xBG3	xBG3	Output	bool	
xBG4	xBG4	Output	bool	
xBG5	xBG5	Output	bool	
xBG6	xBG6	Output	bool	
xBG7	xBG7	Output	bool	
xBG8	xBG8	Output	bool	
xFF1	xFF1	Output	bool	
xFF2	xFF2	Output	bool	
xFF3	xFF3	Output	bool	
xFF4	xFF4	Output	bool	
xFF5	xFF5	Output	bool	
xFF6	xFF6	Output	bool	
xFF7	xFF7	Output	bool	
xFF8	xFF8	Output	bool	

Module 4

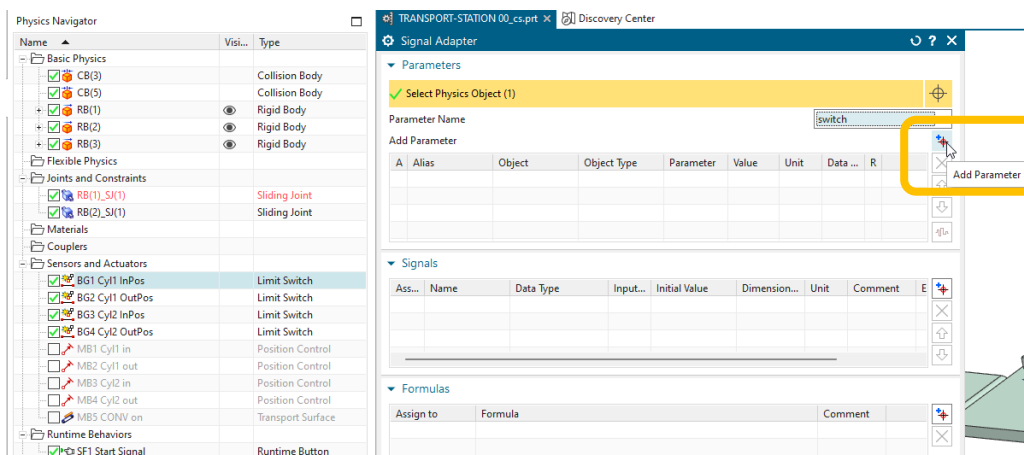
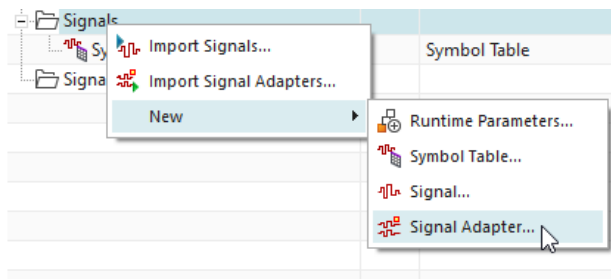
Documentation Key Concept Product development



14 Selection and definition of parameters for the inputs in NX

Video: 14Signals-Ad-Inputs

In this step, parameters for individual sensors are defined specifically according to the respective application.



Module 4

Documentation Key Concept Product development



The limit position sensors are defined with the parameter type "switch" and the start button with the parameter type "triggered."

Please check the overall view about the sensor situation.

The 'Signal Adapter' dialog box is shown with the 'Parameters' tab selected. A yellow box highlights the parameter table. The table lists five parameters: four limit switches and one triggered start button. The 'Signals' and 'Formulas' sections are empty. The 'Name' field at the bottom contains 'SignalAdapter(1)'. The 'OK' button is highlighted with a mouse cursor.

A	Alias	Object	Object Type	Parameter	Value	Unit	Data ...	R
Parameter_1	BG1 Cyl1 InPos	Limit Switch	switch	false		bool	R	
Parameter_2	BG2 Cyl1 OutPos	Limit Switch	switch	false		bool	R	
Parameter_3	BG3 Cyl2 InPos	Limit Switch	switch	false		bool	R	
Parameter_4	BG4 Cyl2 OutPos	Limit Switch	switch	false		bool	R	
Parameter_5	SF1 Start Signal	Runtime Button	triggered	false		bool	R	

Module 4

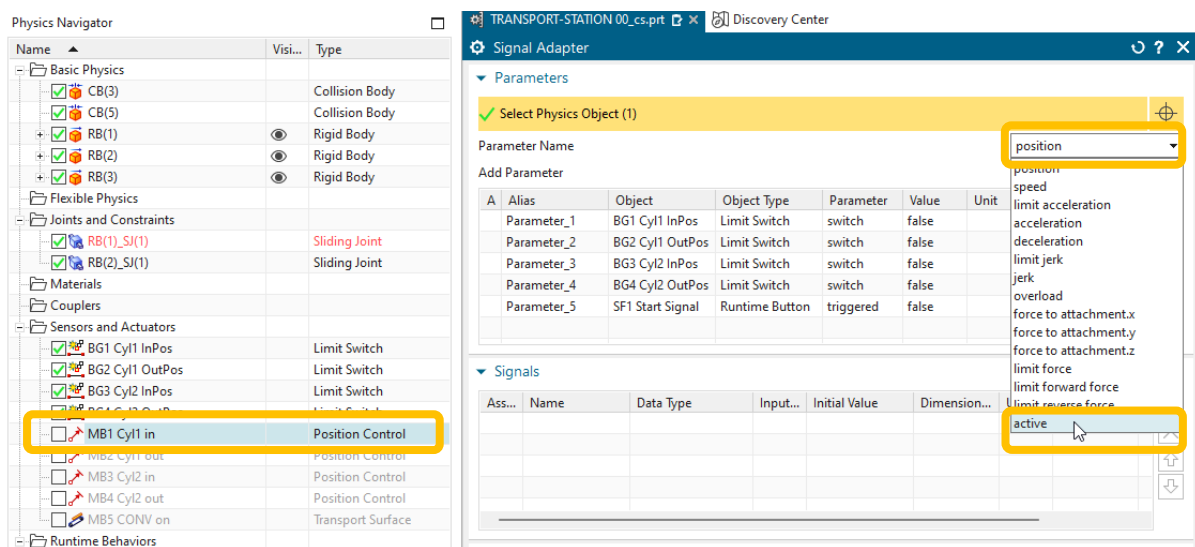
Documentation Key Concept Product development



15 Selection and definition of parameters for outputs in NX

Video: 15Signals-Ad-Outputs

In this step, parameters for individual actuators are defined specifically for the respective application.



When defining the parameters, a definition with the parameter type "active" is used.

Module 4

Documentation Key Concept Product development



Signal Adapter

Parameters

✓ Select Physics Object (1)

Parameter Name: active

Add Parameter

A	Alias	Object	Object Type	Parameter	Value	Unit	Data ...	R
	Parameter_1	BG1 Cyl1 InPos	Limit Switch	switch	false		bool	R
	Parameter_2	BG2 Cyl1 OutPos	Limit Switch	switch	false		bool	R
	Parameter_3	BG3 Cyl2 InPos	Limit Switch	switch	false		bool	R
	Parameter_4	BG4 Cyl2 OutPos	Limit Switch	switch	false		bool	R
	Parameter_5	SP1 Start Signal	Runtime Button	triggered	false		bool	R
☐	Parameter_6	MB1 Cyl1 in	Position Control	active	true		bool	W
☐	Parameter_7	MB2 Cyl1 out	Position Control	active	true		bool	W
☐	Parameter_8	MB3 Cyl2 in	Position Control	active	true		bool	W
☐	Parameter_9	MB4 Cyl2 out	Position Control	active	true		bool	W
☐	Parameter_10	MB5 CONV on	Transport Surface	active	true		bool	W

Module 4

Documentation Key Concept Product development



16 Assignment of signals in the signal adapter

Video: 16Add_Signals

All input and output signals are then created in the signal adapter.

The screenshot shows the 'Signal Adapter' window with two main sections: 'Parameters' and 'Signals'. The 'Parameters' section has a yellow header with a red asterisk and the text 'Select Physics Object (0)'. Below it is a table with 10 parameters. The 'Signals' section has a table with 8 columns: 'Ass...', 'Name', 'Data Type', 'Input...', 'Initial Value', 'Dimension...', 'Unit', and 'Comment'. A yellow box highlights the 'Add' button in the bottom right corner of the 'Signals' table.

A	Alias	Object	Object Type	Parameter	Value	Unit	Data ...	R
	Parameter_1	BG1 Cyl1 InPos	Limit Switch	switch	false		bool	R
	Parameter_2	BG2 Cyl1 OutPos	Limit Switch	switch	false		bool	R
	Parameter_3	BG3 Cyl2 InPos	Limit Switch	switch	false		bool	R
	Parameter_4	BG4 Cyl2 OutPos	Limit Switch	switch	false		bool	R
	Parameter_5	SF1 Start Signal	Runtime Button	triggered	false		bool	R
☐	Parameter_6	MB1 Cyl1 in	Position Control	active	true		bool	W
☐	Parameter_7	MB2 Cyl1 out	Position Control	active	true		bool	W
☐	Parameter_8	MB3 Cyl2 in	Position Control	active	true		bool	W
☐	Parameter_9	MB4 Cyl2 out	Position Control	active	true		bool	W
☐	Parameter_10	MB5 CONV on	Transport Surface	active	true		bool	W

Ass...	Name	Data Type	Input...	Initial Value	Dimension...	Unit	Comment

The assignment of whether it is an INPUT or OUTPUT is made from the perspective of the NX model. This means that an actuator is defined as an INPUT and a sensor as an OUTPUT.

Module 4

Documentation Key Concept Product development



After the import of the sensors (Outputs for the simulation) the topic must be change for the solenoid valves to an Input.

▼ Signals

Ass...	Name	Data Type	Input...	Initial Value	Dimension...	Unit	Comment	E
☐	xBG1	bool	Output	false				
☐	xBG9	bool	Output	false				
	xBG9							
	xBG2							
	xBG3							
	xBG4							
	xBG5							
	xBG6							
	xBG7							
	xBG8							
	xF1							
	xF2							

▼ Formulas

Assign to	Formula	Comment
-----------	---------	---------

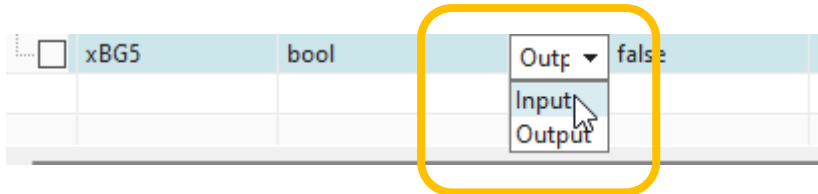
▼ Signals

Ass...	Name	Data Type	Input...	Initial Value	Dimension...	Unit	Comment	E
☐	xBG1	bool	Output	false				
☐	xBG2	bool	Output	false				

▼ Formulas

Module 4

Documentation Key Concept Product development



The assignment of whether it is an INPUT or OUTPUT is made from the perspective of the NX model. This means that an actuator is defined as an INPUT and a sensor as an OUTPUT.

▼ Signals

Ass...	Name	Data Type	Input...	Initial Value	Dimension...	Unit	Comment	E
☐	xBG1	bool	Output	false				
☐	xBG2	bool	Output	false				
☐	xBG3	bool	Output	false				
☐	xBG4	bool	Output	false				
☐	xFF1	bool	Output	false				
☐	xMB1	bool	Input	false				
☐	xMB2	bool	Input	false				
☐	xMB3	bool	Input	false				
☐	xMB4	bool	Input	false				
☐	xMB5	bool	Input	false				

Module 4

Documentation Key Concept Product development



17 Connecting parameters and signals in the signal adapter

Video: 17Connect_MB

In this step, the individual signals are manually merged with the respective matching parameter.

In the first step, the output parameters are checked and then assigned to the corresponding signal via the Formulas area.

The screenshot shows a software interface with three main sections. The top section is a table of parameters, the middle is a 'Signals' table, and the bottom is a 'Formulas' table. A yellow box highlights the first five rows of the parameters table, and another yellow box highlights the first five rows of the formulas table.

Parameter	Signal	Control	Active	True	Unit
Parameter_6	MB1 Cyl1 in	Position Control	active	true	bool W
Parameter_7	MB2 Cyl1 out	Position Control	active	true	bool W
Parameter_8	MB3 Cyl2 in	Position Control	active	true	bool W
Parameter_9	MB4 Cyl2 out	Position Control	active	true	bool W
Parameter_10	MB5 CONV on	Transport Surface	active	true	bool W

Ass...	Name	Data Type	Input...	Initial Value	Dimension...	Unit	Comment
☐	xBG1	bool	Output	false			
☐	xBG2	bool	Output	false			
☐	xBG3	bool	Output	false			
☐	xBG4	bool	Output	false			
☐	xsF1	bool	Output	false			
☐	xMB1	bool	Input	false			
☐	xMB2	bool	Input	false			
☐	xMB3	bool	Input	false			
☐	xMB4	bool	Input	false			
☐	xMB5	bool	Input	false			

Assign to	Formula	Comment
Parameter_6		
Parameter_7		
Parameter_8		
Parameter_9		
Parameter_10		

Module 4

Documentation
Key Concept Product development



▼ Formulas

Assign to	Formula	Comment
Parameter_6	xMB1	
Parameter_7		
Parameter_8		
Parameter_9		
Parameter_10		

Formula

xBG1
xBG2
xBG3
xBG4
xMB1
xMB2
xMB3

Cancel

Please press enter to complete and confirm the formula input!!!

Module 4

Documentation Key Concept Product development



▼ Signals

Ass...	Name	Data Type	Input...	Initial Value	Dimension...	Unit	Comment	E
☐	xBG1	bool	Output	false				
☐	xBG2	bool	Output	false				
☐	xBG3	bool	Output	false				
☐	xBG4	bool	Output	false				
☐	xFF1	bool	Output	false				
☐	xMB1	bool	Input	false				
☐	xMB2	bool	Input	false				
☐	xMB3	bool	Input	false				
☐	xMB4	bool	Input	false				
☐	xMB5	bool	Input	false				

▼ Formulas

Assign to	Formula	Comment
Parameter_6	xMB1	
Parameter_7	xMB2	
Parameter_8	xMB3	
Parameter_9	xMB4	
Parameter_10	xMB5	

Formula

▼ Name

SignalAdapter(1)

OK Cancel

Module 4

Documentation Key Concept Product development



18 Connecting input parameters and signals in the signal adapter

Video: 17Connect_BG

In this step, the individual signals are manually merged with the respective matching parameters.

In this step, the input signals are checked and then assigned to the corresponding parameter via the Formulas area.

This corresponds to a reverse assignment as in the previous point.

The screenshot shows a software interface with two main panels. The top panel, titled 'Signals', contains a table with columns: Ass..., Name, Data Type, Input..., Initial Value, Dimension..., Unit, Comment, and E. The bottom panel, titled 'Formulas', contains a table with columns: Assign to, Formula, and Comment. Both panels have a yellow box highlighting specific rows. In the 'Signals' panel, the row for 'xSF1' is highlighted. In the 'Formulas' panel, the row for 'Parameter_10' is highlighted, and the 'Formula' column contains the value 'xMB5'. Below this, the 'Formula' column is empty for the rows 'xBG1', 'xBG2', 'xBG3', 'xBG4', and 'xSF1'.

Ass...	Name	Data Type	Input...	Initial Value	Dimension...	Unit	Comment	E
☒	xBG1	bool	Output	false				
☒	xBG2	bool	Output	false				
☒	xBG3	bool	Output	false				
☒	xBG4	bool	Output	false				
☒	xSF1	bool	Output	false				
☐	xMB1	bool	Input	false				
☐	xMB2	bool	Input	false				
☐	xMB3	bool	Input	false				
☐	xMB4	bool	Input	false				

Assign to	Formula	Comment
Parameter_6	xMB1	
Parameter_7	xMB2	
Parameter_8	xMB3	
Parameter_9	xMB4	
Parameter_10	xMB5	
xBG1		
xBG2		
xBG3		
xBG4		
xSF1		

Module 4

Documentation Key Concept Product development



Please press enter to complete and confirm the formula output!!!

Formulas

Assign to	Formula	Comment	
Parameter_6	xMB1		
Parameter_7	xMB2		
Parameter_8	xMB3		
Parameter_9	xMB4		
Parameter_10	xMB5		
xBG2	Parameter_2		
xBG3	Parameter_3		
xBG4	Parameter_4		
xSF1			
xBG1	Parameter_1		

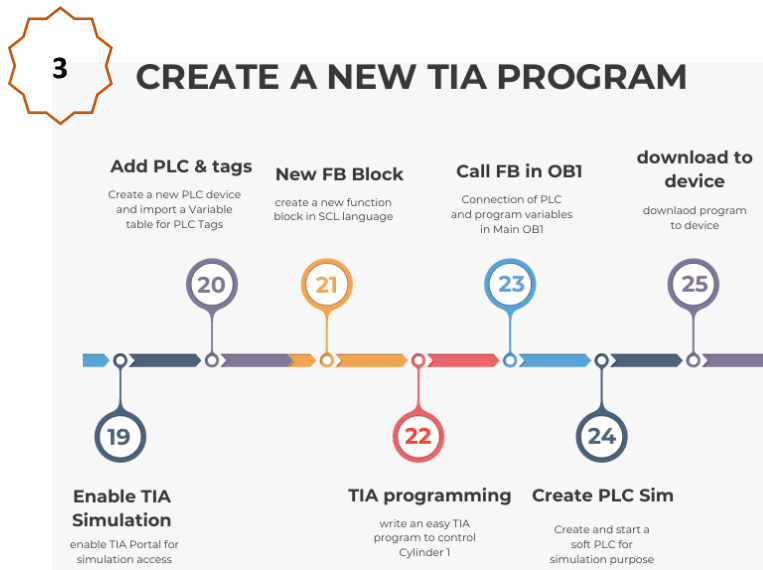
Formula

Parameter_5

Name

SignalAdapter(1)

Part 3



In this section we create the TIA Portal Project for the programming part of the digital twin.

We start with a new function block (FB) and add the code to control one cylinder. PLC Sim advanced takes the role of a soft PLC and we download our program to an instance of it.

Module 4

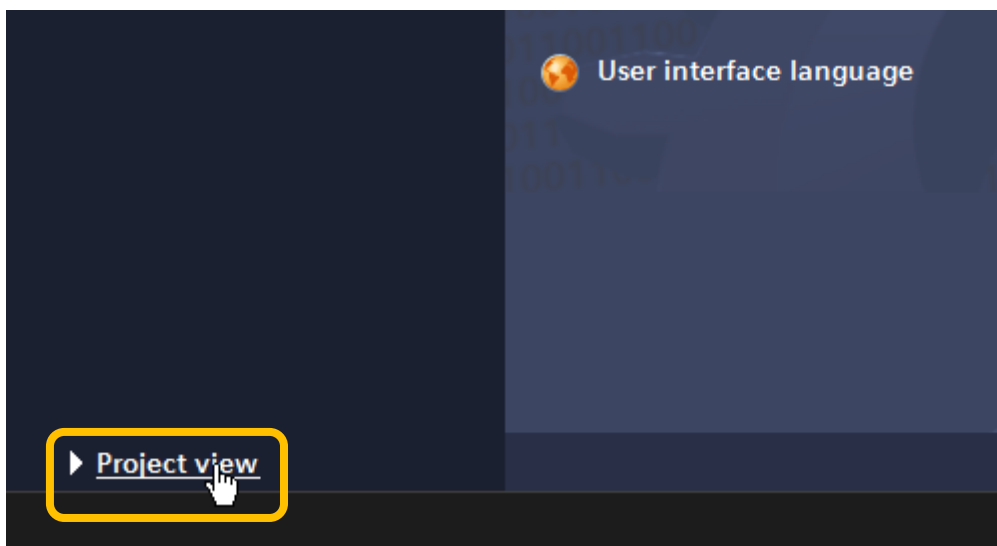
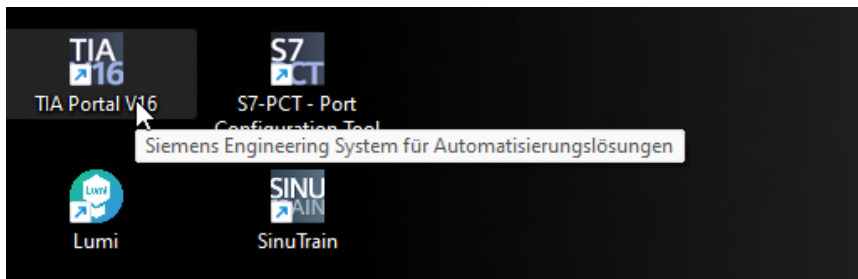
Documentation Key Concept Product development



19 Simulation release in the TIA Portal

Video: 19TIA_new_enable_Simulation

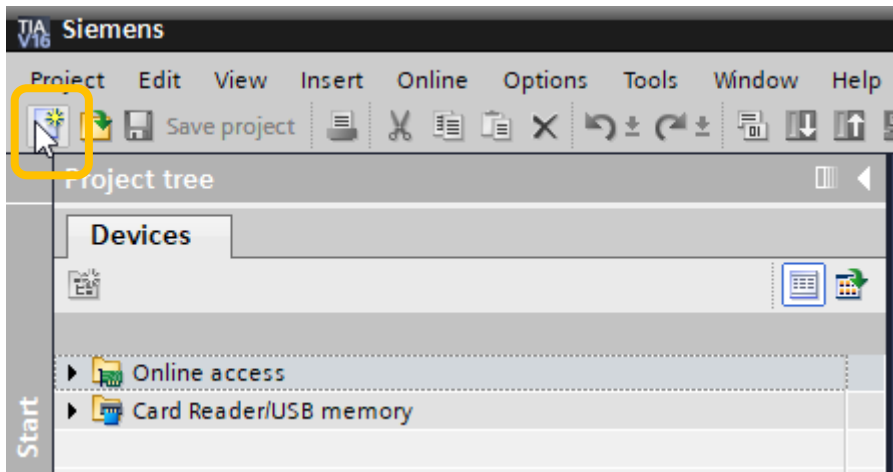
In order to work with a simulation in the TIA Portal, this simulation option must first be activated in an empty project.



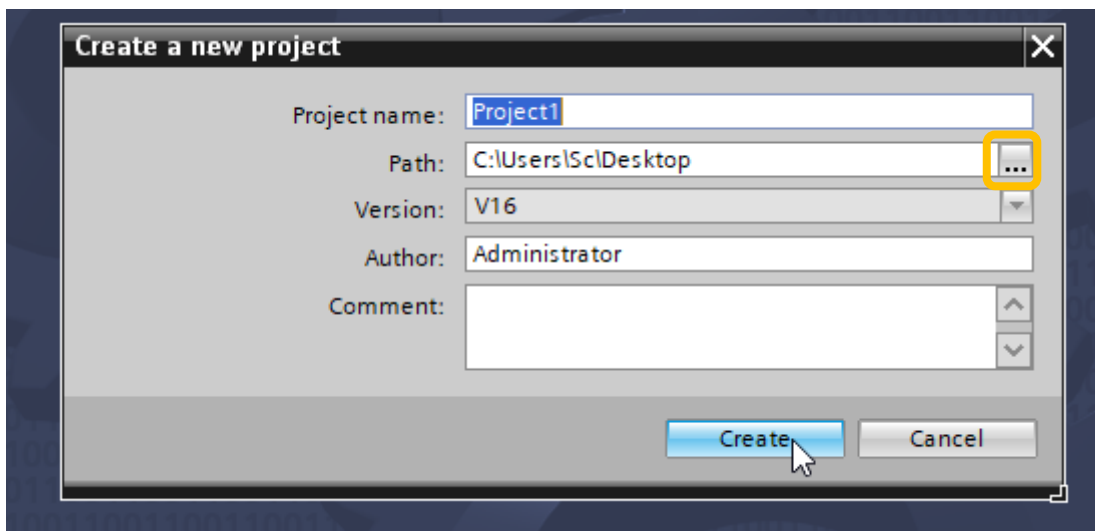
Module 4

Documentation
Key Concept Product development

Create empty/new project

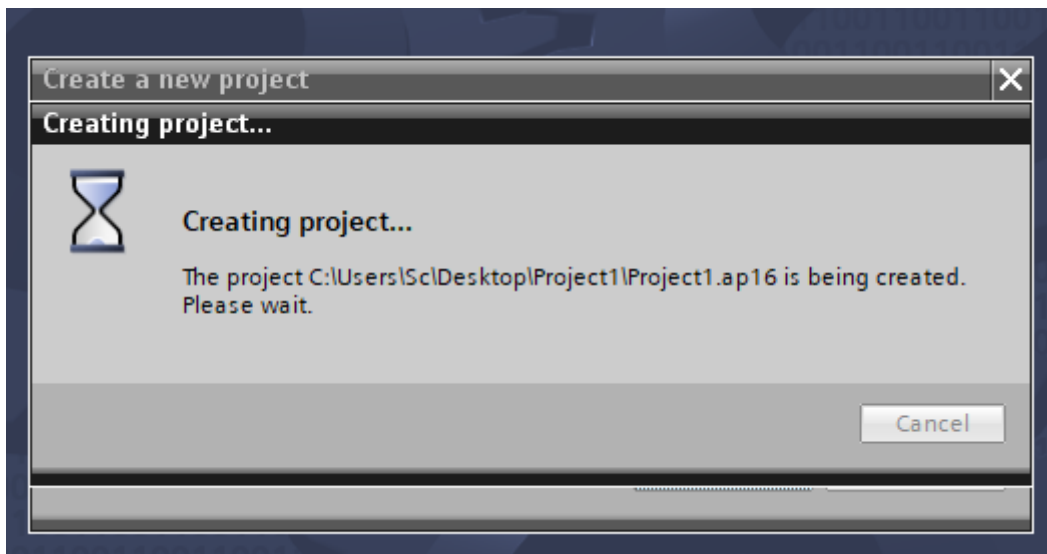
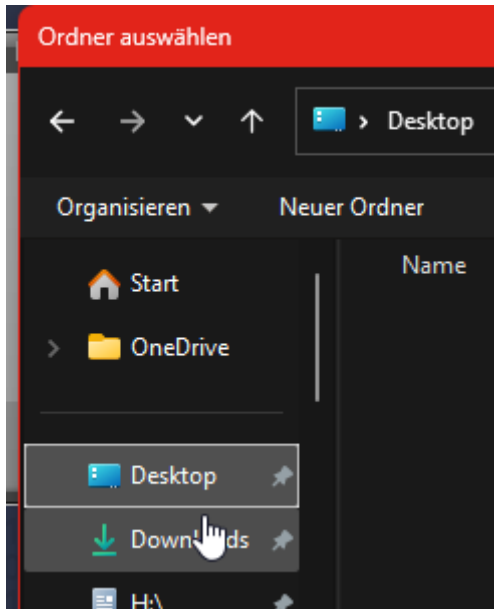


Set the path to the C:// drive (this is different in the video, please use your desktop)



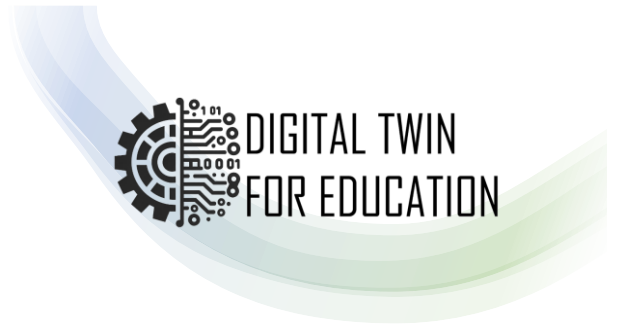
Module 4

Documentation
Key Concept Product development

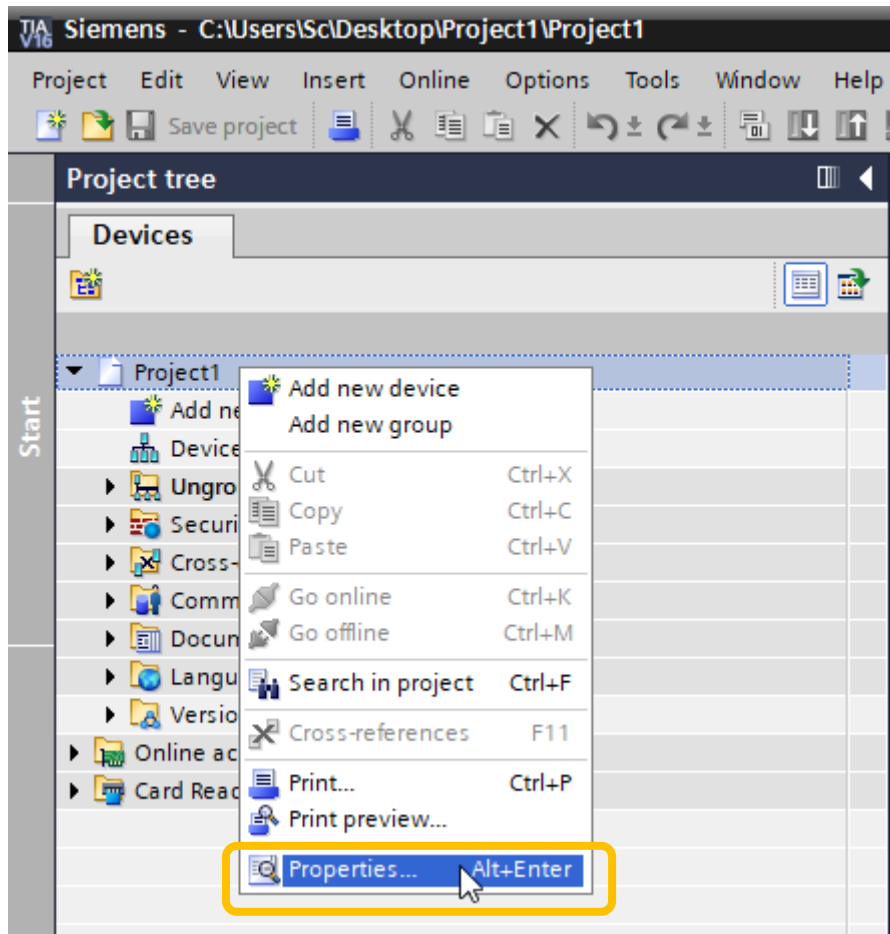


Module 4

Documentation Key Concept Product development



Right-click on the project and then select Properties

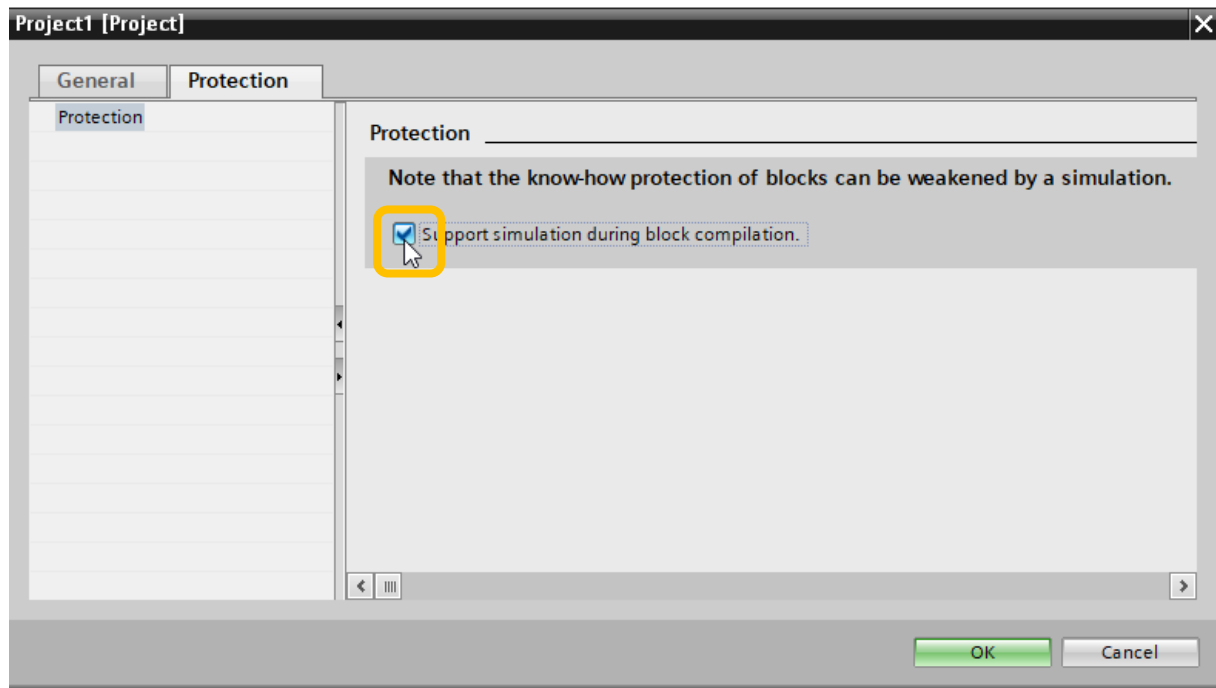


Module 4

Documentation
Key Concept Product development



Simulation must be activated



Module 4

Documentation Key Concept Product development

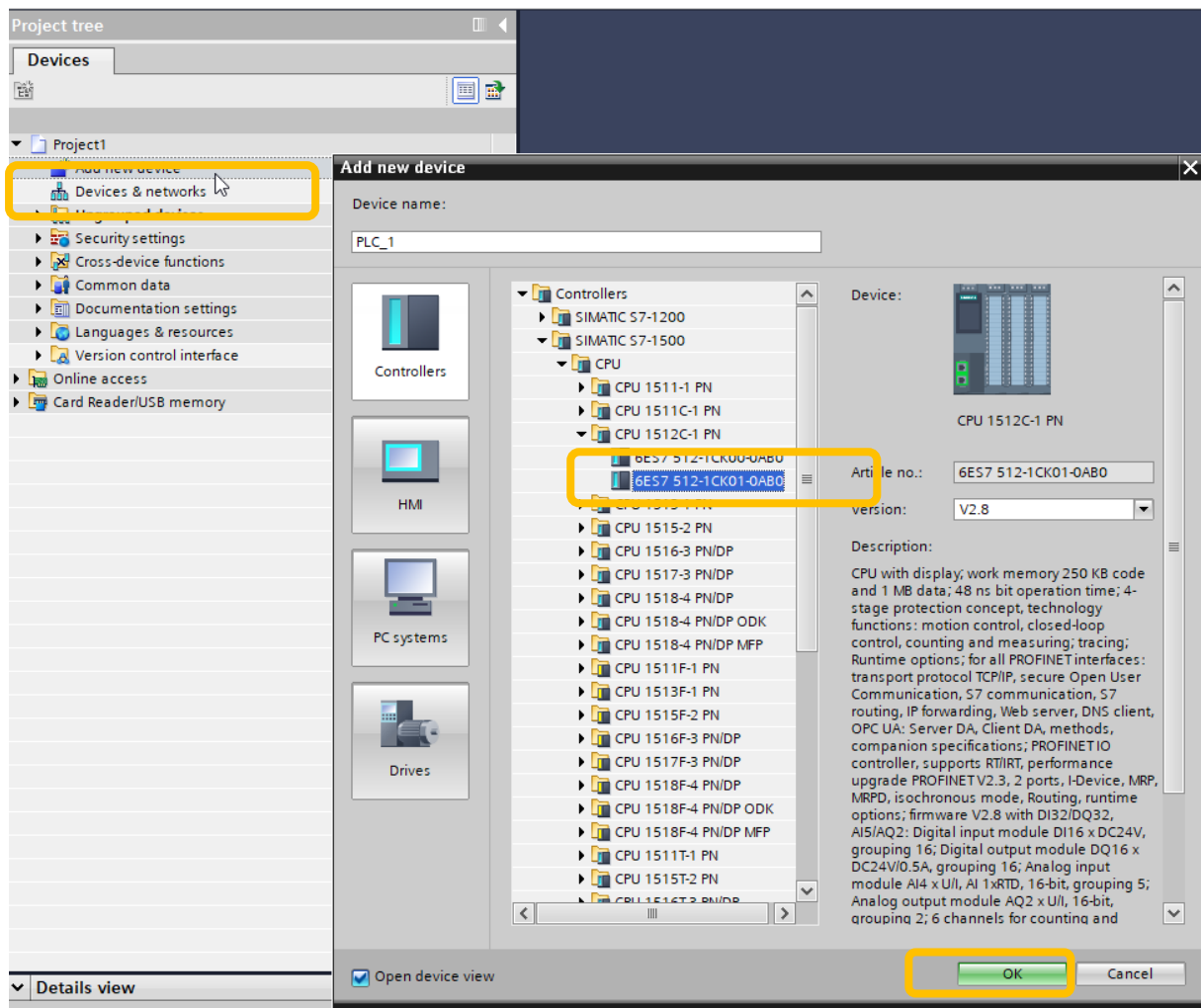


20 Create PLC hardware and integrate PLC variables

Video: 20TIA_new_PLC_Import_Variables

A hardware configuration for a PLC must be defined.

In this case, a 1500 PLC is used, as this corresponds to the reality on site.

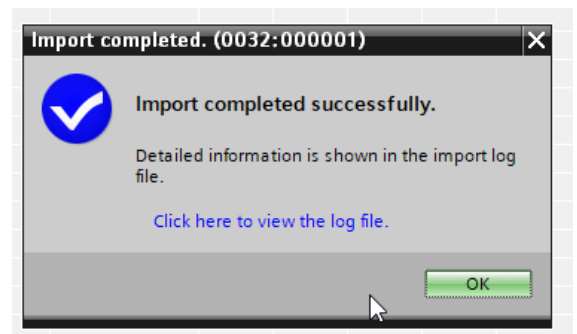
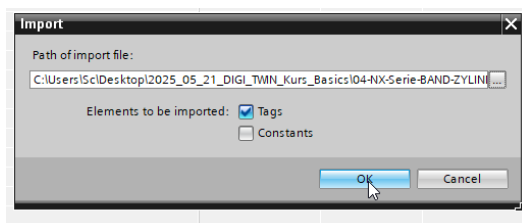
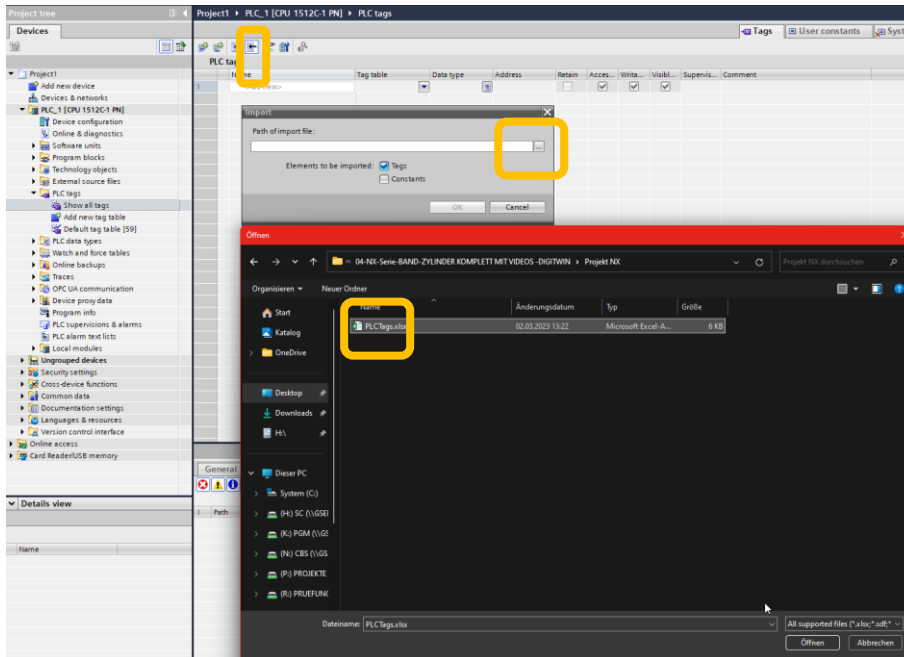


Module 4

Documentation Key Concept Product development



Once the PLC has been configured, the PLC variables (PLC tags) are imported using an Excel file. The file is located in the folder – Appendix → PLC_Tags.xlsx.



Overview of the imported PLC variables.

Module 4

Documentation Key Concept Product development



Project tree

Project1 ▸ PLC_1 [CPU 1512C-1 PN] ▸ PLC tags

Devices

- Project1
 - Add new device
 - Devices & networks
 - PLC_1 [CPU 1512C-1 PN]
 - Device configuration
 - Online & diagnostics
 - Software units
 - Program blocks
 - Technology objects
 - External source files
 - PLC tags
 - Show all tags
 - Add new tag table
 - Default tag table [59]
 - Variablenabelle [32]**
 - PLC data types
 - Watch and force tables
 - Online backups
 - Traces
 - OPC UA communication
 - Device proxy data
 - Program info
 - PLC supervisions & alarms
 - PLC alarm text lists
 - Local modules
 - Ungrouped devices
 - Security settings
 - Cross-device functions
 - Common data
 - Documentation settings
 - Languages & resources
 - Version control interface
 - Online access

PLC tags							
	Name	Tag table	Data type	Address	Retain	Access...	Writa...
1	xMB1	Variablenabelle	Bool	%Q4.0	☐	☒	☒
2	xMB2	Variablenabelle	Bool	%Q4.1	☐	☒	☒
3	xMB3	Variablenabelle	Bool	%Q4.2	☐	☒	☒
4	xMB4	Variablenabelle	Bool	%Q4.3	☐	☒	☒
5	xMB5	Variablenabelle	Bool	%Q4.4	☐	☒	☒
6	xMB6	Variablenabelle	Bool	%Q4.5	☐	☒	☒
7	xMB7	Variablenabelle	Bool	%Q4.6	☐	☒	☒
8	xMB8	Variablenabelle	Bool	%Q4.7	☐	☒	☒
9	xPF1	Variablenabelle	Bool	%Q5.0	☐	☒	☒
10	xPF2	Variablenabelle	Bool	%Q5.1	☐	☒	☒
11	xPF3	Variablenabelle	Bool	%Q5.2	☐	☒	☒
12	xPF4	Variablenabelle	Bool	%Q5.3	☐	☒	☒
13	xPF5	Variablenabelle	Bool	%Q5.4	☐	☒	☒
14	xPF6	Variablenabelle	Bool	%Q5.5	☐	☒	☒
15	xPF7	Variablenabelle	Bool	%Q5.6	☐	☒	☒
16	xPF8	Variablenabelle	Bool	%Q5.7	☐	☒	☒
17	xBG1	Variablenabelle	Bool	%I10.0	☐	☒	☒
18	xBG2	Variablenabelle	Bool	%I10.1	☐	☒	☒
19	xBG3	Variablenabelle	Bool	%I10.2	☐	☒	☒
20	xBG4	Variablenabelle	Bool	%I10.3	☐	☒	☒
21	xBG5	Variablenabelle	Bool	%I10.4	☐	☒	☒
22	xBG6	Variablenabelle	Bool	%I10.5	☐	☒	☒
23	xBG7	Variablenabelle	Bool	%I10.6	☐	☒	☒
24	xBG8	Variablenabelle	Bool	%I10.7	☐	☒	☒
25	xSF1	Variablenabelle	Bool	%I11.0	☐	☒	☒
26	xSF2	Variablenabelle	Bool	%I11.1	☐	☒	☒
27	xSF3	Variablenabelle	Bool	%I11.2	☐	☒	☒
28	xSF4	Variablenabelle	Bool	%I11.3	☐	☒	☒
29	xSF5	Variablenabelle	Bool	%I11.4	☐	☒	☒
30	xSF6	Variablenabelle	Bool	%I11.5	☐	☒	☒

Module 4

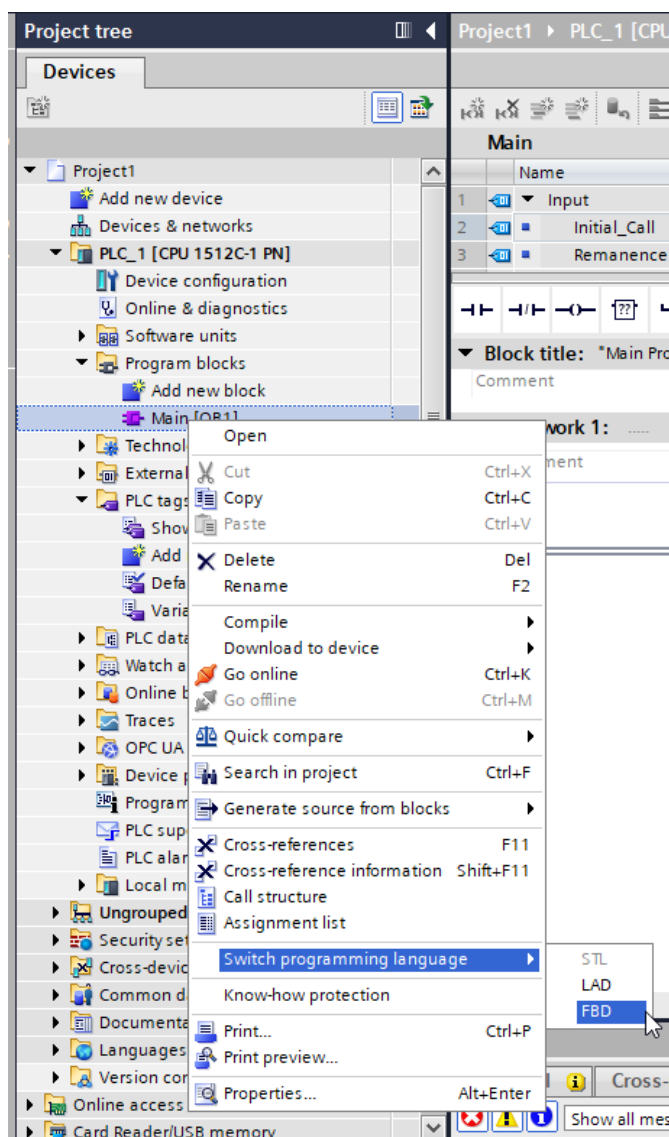
Documentation Key Concept Product development



21 Creating a function block in the TIA Portal

Video: 21TIA_New_FB

Changing the programming language in the MAIN OB1 block



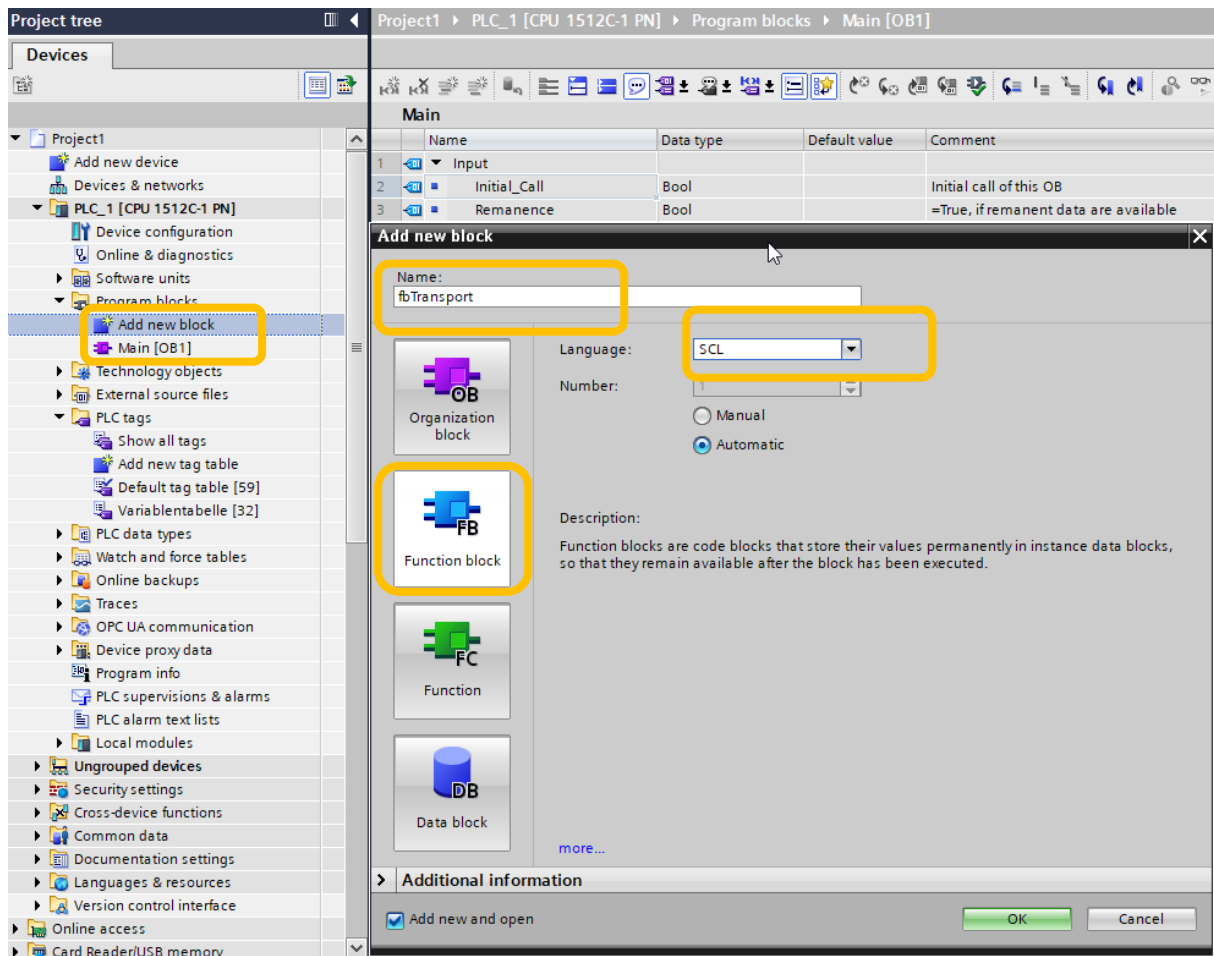
A new function block is then created using the SCL programming language.

Module 4

Documentation Key Concept Product development



Please select function block!



Please select Function block!

Module 4

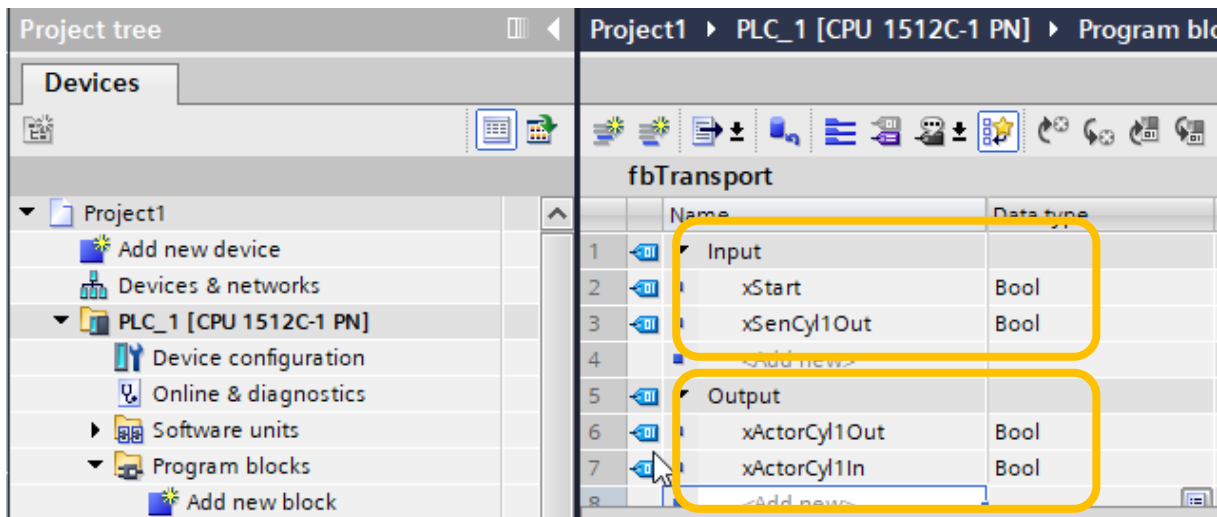
Documentation
Key Concept Product development



22 Program creation in the function block

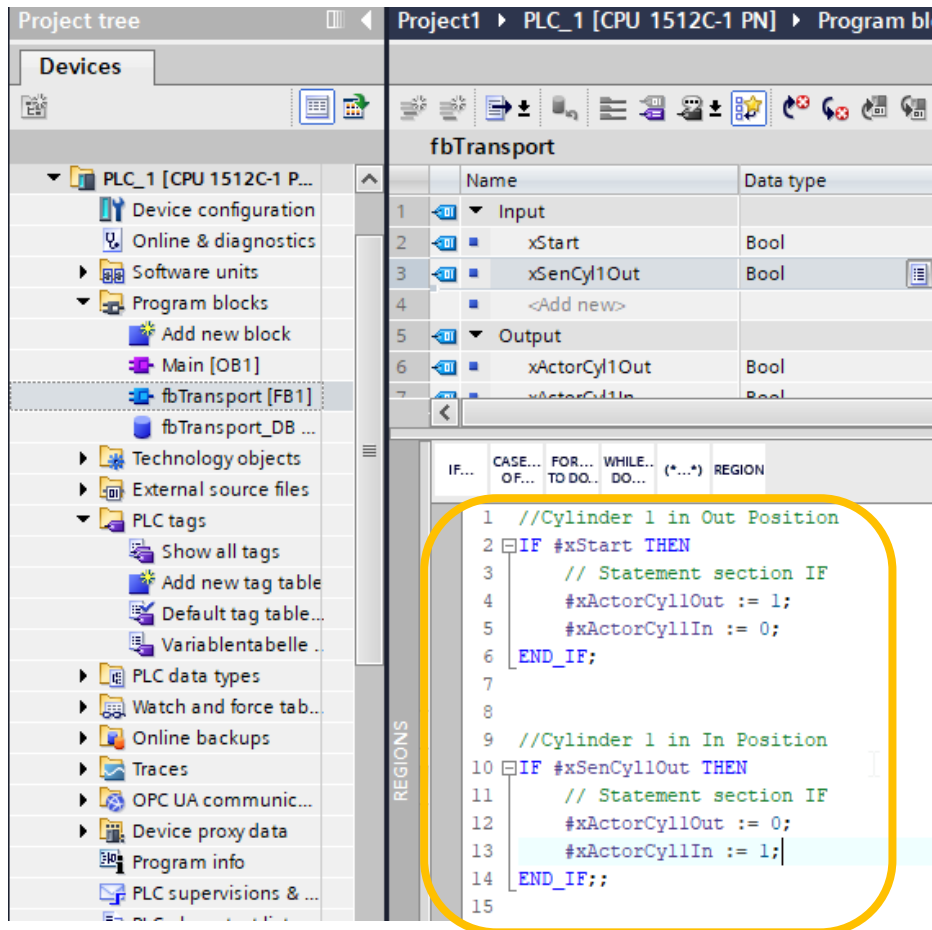
Video: 22TIA_FB_Prog

For the first test, a few input and output variables are created and used to write a small PLC program for cylinder 1.



Module 4

Documentation
Key Concept Product development



Module 4

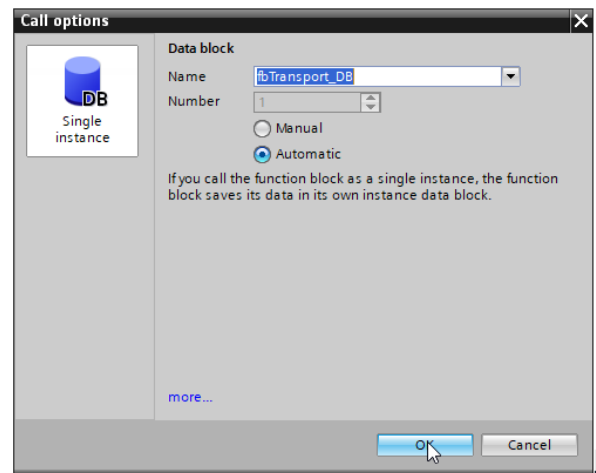
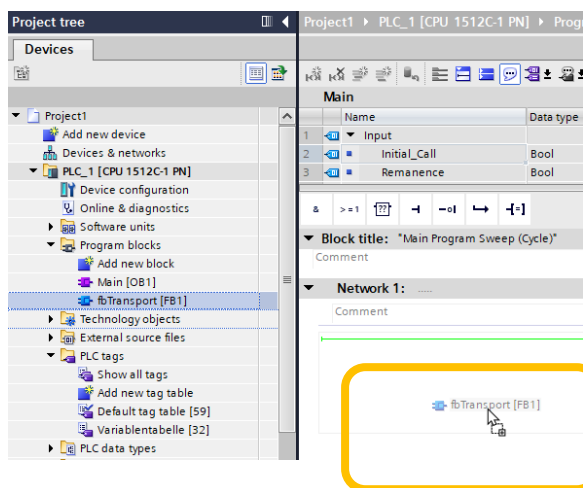
Documentation Key Concept Product development



23 Call the FB in Main OB1 and save the program

Video: 23TIA_Prog_Save

In MAIN OB1, the variables of the function block are linked to the PLC variables (PLC tags). First, the function block must be dragged and dropped into MAIN OB1.



Module 4

Documentation
Key Concept Product development



Project tree

Project1 > PLC_1 [CPU 1512C-1 PN] > Program blocks > Main [OB1]

Devices

- PLC_1 [CPU 1512C-1 PN]
 - Device configuration
 - Online & diagnostics
 - Software units
 - Program blocks
 - Add new block
 - Main [OB1]
 - fbTransport [FB1]
 - fbTransport_DB [DB1]
 - Technology objects
 - External source files
 - PLC tags
 - Show all tags
 - Add new tag table
 - Default tag table [59]
 - Variablen-tabelle [32]
 - PLC data types
 - Watch and force tables

Details view

Name	Dat...	Details	Comment
xBG1	Bool	%I10.0	
xBG2	Bool	%I10.1	
xBG3	Bool	%I10.2	
xBG4	Bool	%I10.3	
xBG5	Bool	%I10.4	
xBG6	Bool	%I10.5	
xBG7	Bool	%I10.6	
xBG8	Bool	%I10.7	
xMB1	Bool	%Q4.0	
xMB2	...	%...	
xMB3	Bool	%Q4.2	

Main

Name	Data type	Default value	Comment
1	Input		
2	Initial_Call	Bool	Initial ca
3	Remanence	Bool	=True, if

Block title: "Main Program Sweep (Cycle)"

Network 1:

Comment

Diagram showing the connection of the "fbTransport" block. The block is connected to the "xStart" input (xSF1) and the "xSenCyl1Out" output (xBG2). The block is also connected to the "xActorCyl1Out" output (xMB2) and the "xActorCyl1In" input (xMB1). The block is connected to the "xSenCyl1Out" output (xBG2) and the "xActorCyl1Out" output (xMB2).

Network 2:

Comment

General | Cross-references | Compile | Syntax

Module 4

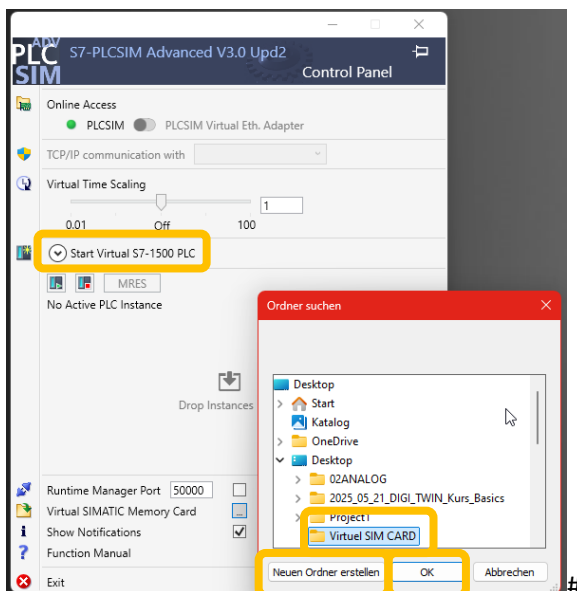
Documentation Key Concept Product development



24 Creating a new simulation instance in PLCSim Advanced

Video: 24_Sim_New_Inst

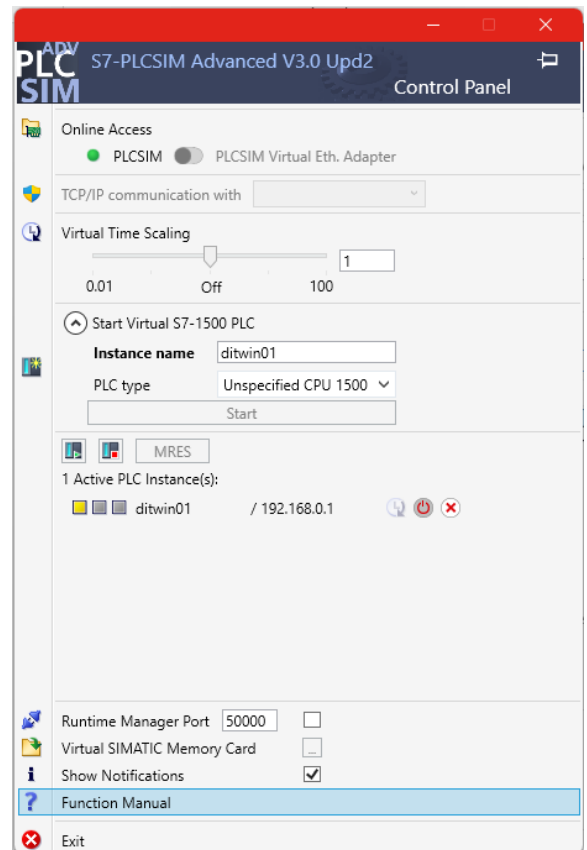
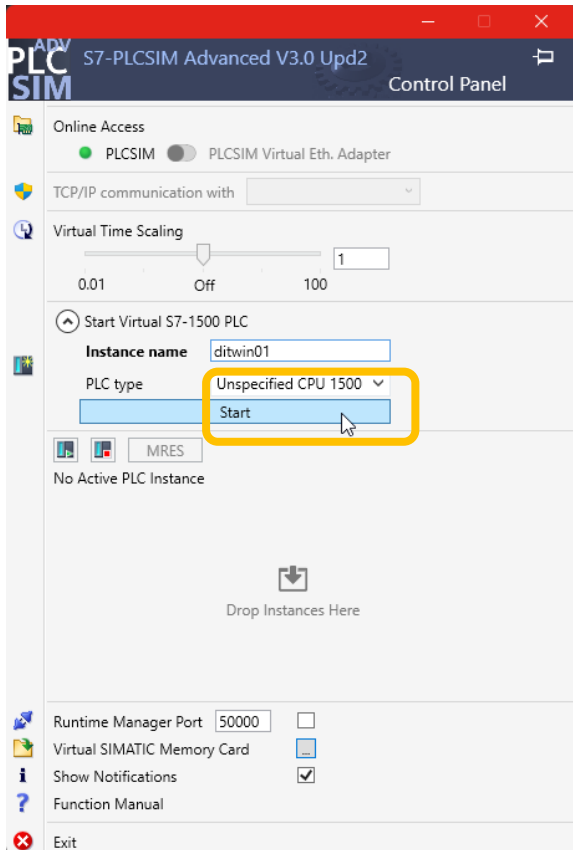
A virtual soft PLC can be created by starting the PLC Sim Advanced software. This controller, named **ditwin01**, is used to load the current PLC.



PLEASE NAME THE SOFT PLC – ditwin01!

Module 4

Documentation Key Concept Product development



Module 4

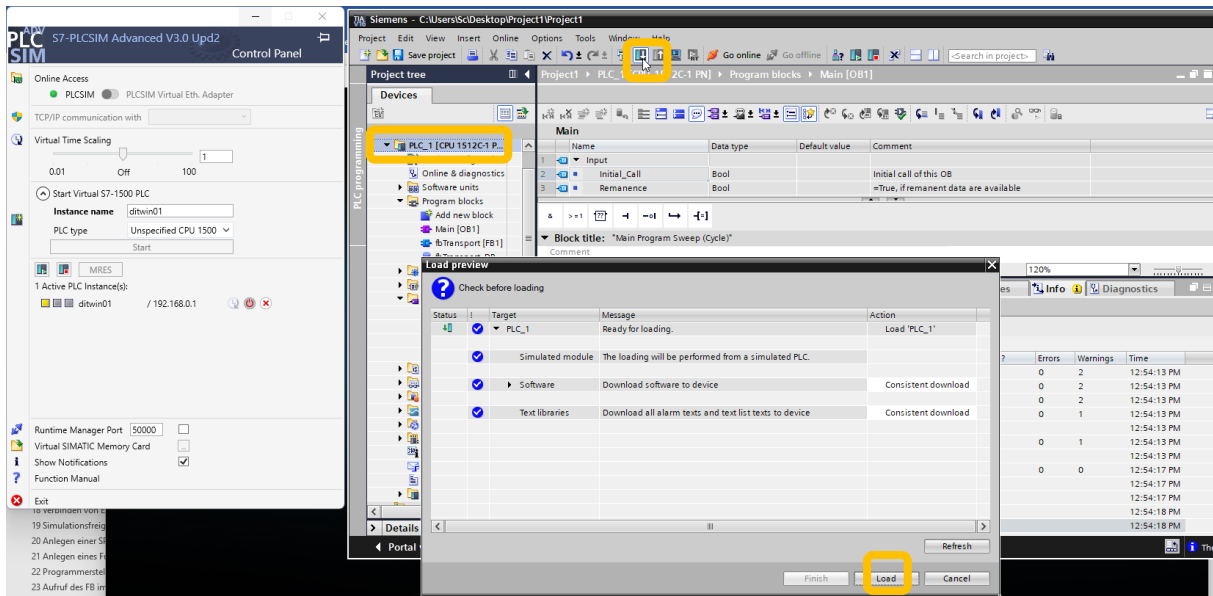
Documentation Key Concept Product development



25 Load TIA program in PLC Sim

Video: 25_TIA_Sim_Transfer

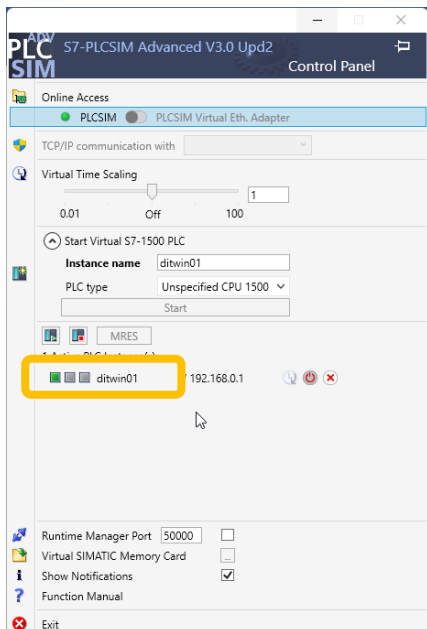
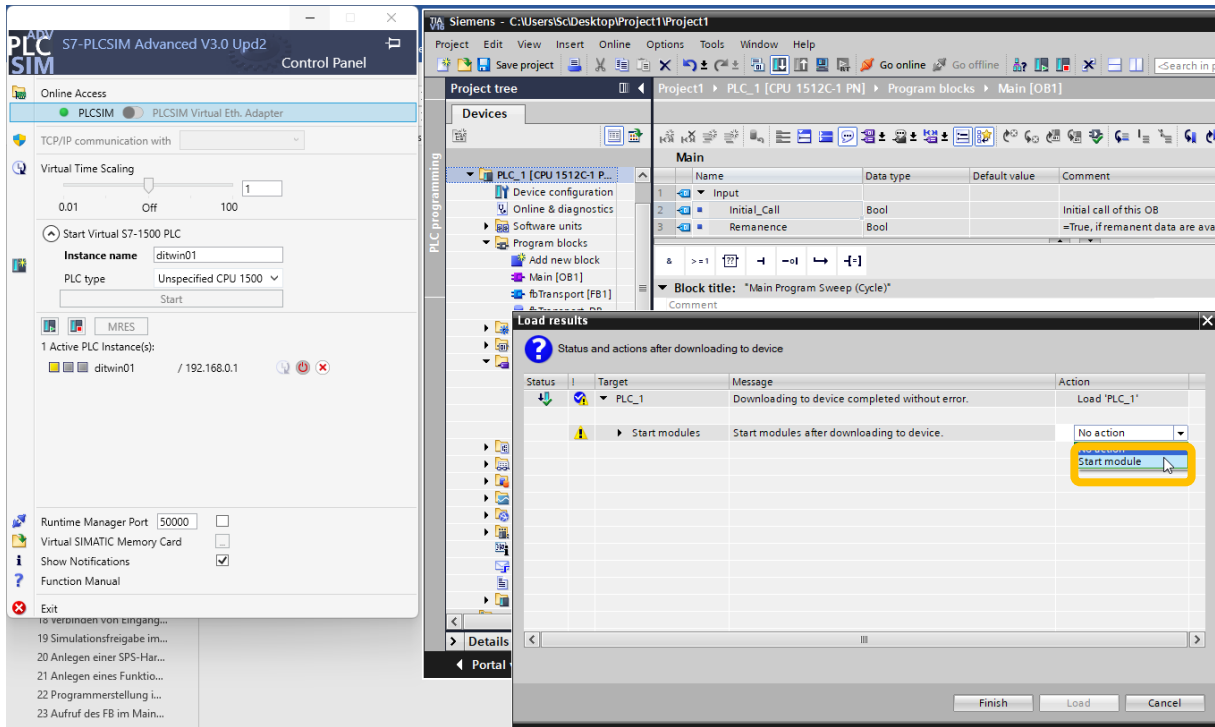
The current PLC program can now be loaded into PLC SIM Advanced.



When the module is started, the PLC changes its status from Stop to Start and the LED turns green.

Module 4

Documentation Key Concept Product development

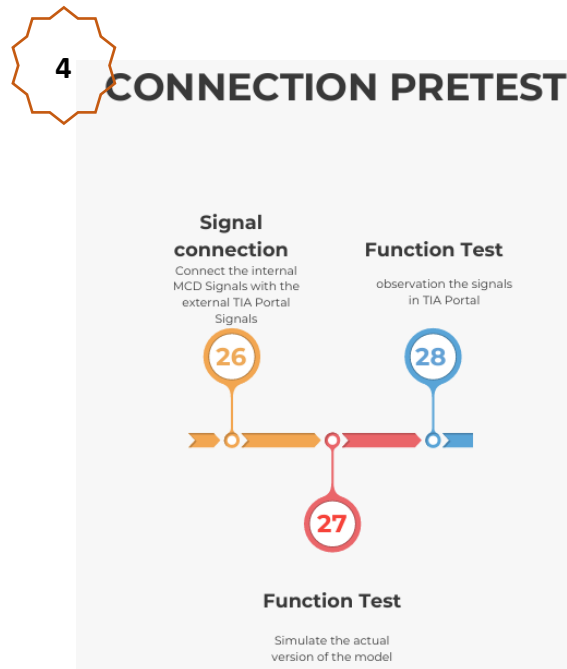


Module 4

Documentation
Key Concept Product development



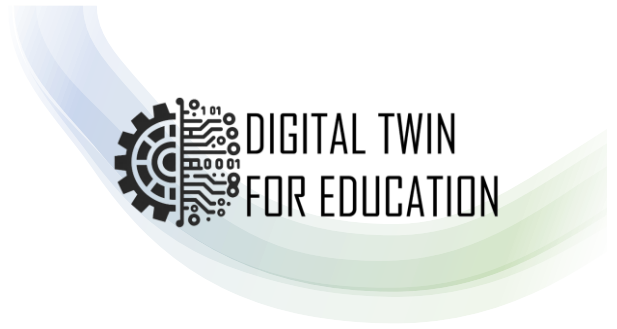
Part 4



In this small section we are making some adjustments in NX and we run the test program.

Module 4

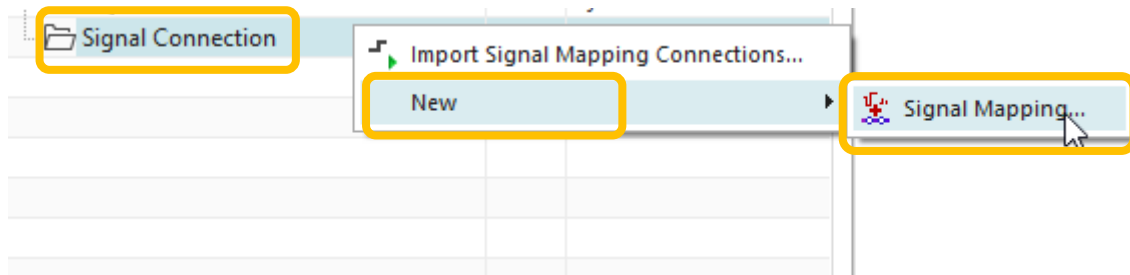
Documentation
Key Concept Product development



26 Creating signal connections in MCD

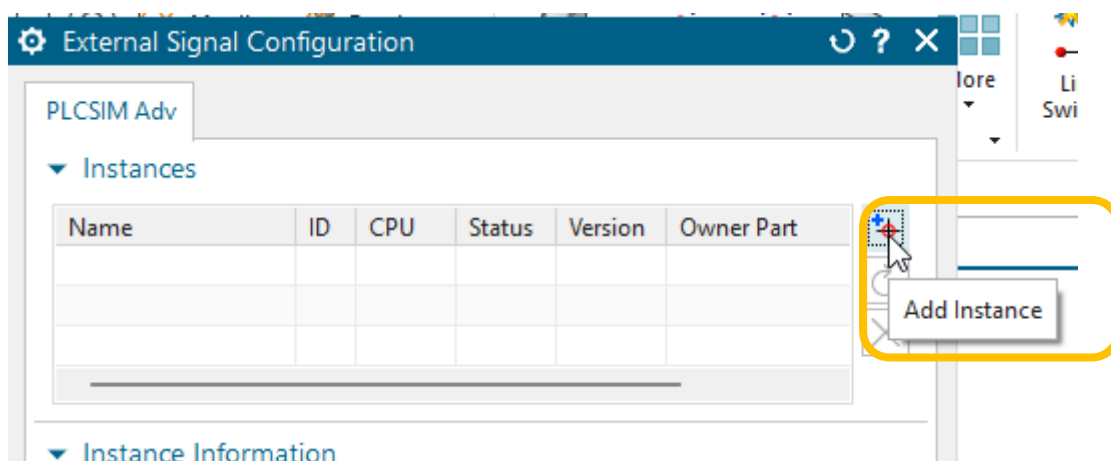
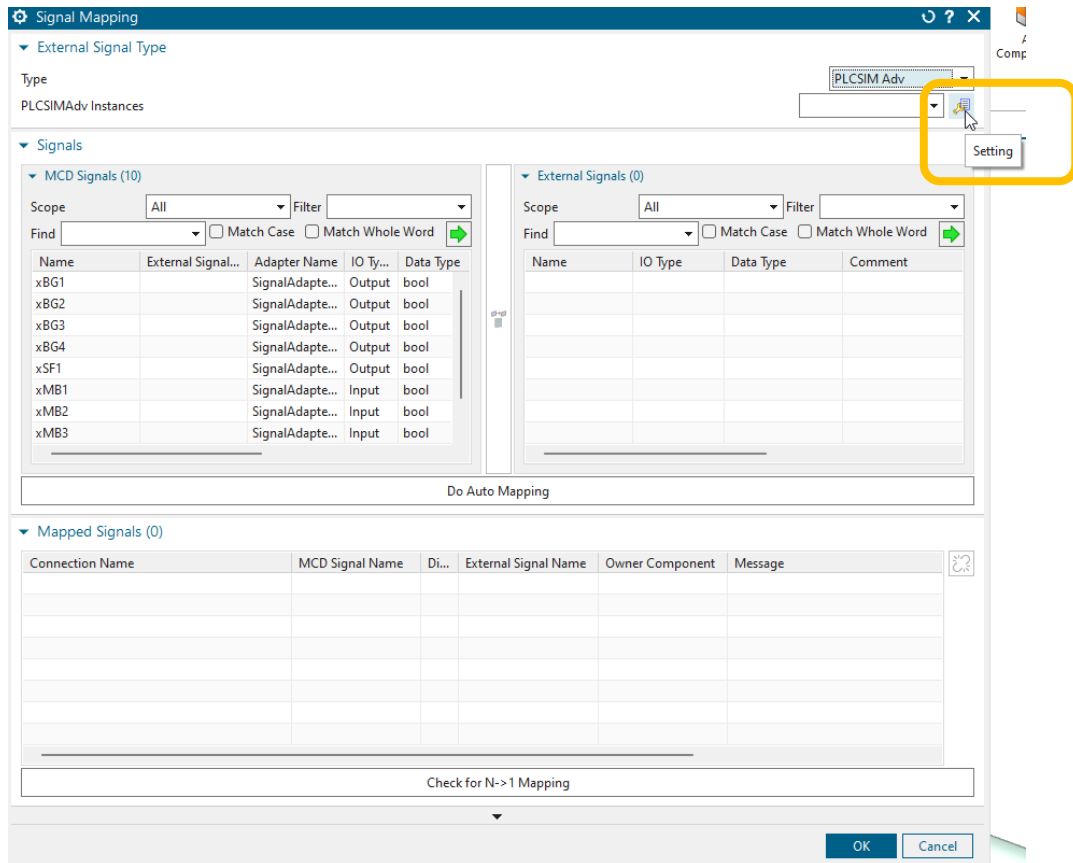
Video: 26_NX_New_Signal_Conn

The simulation level is then ready for operation and the signal connections in Siemens NX MCD must be created.



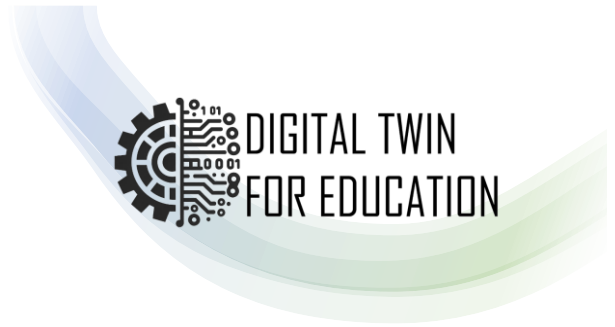
Module 4

Documentation Key Concept Product development

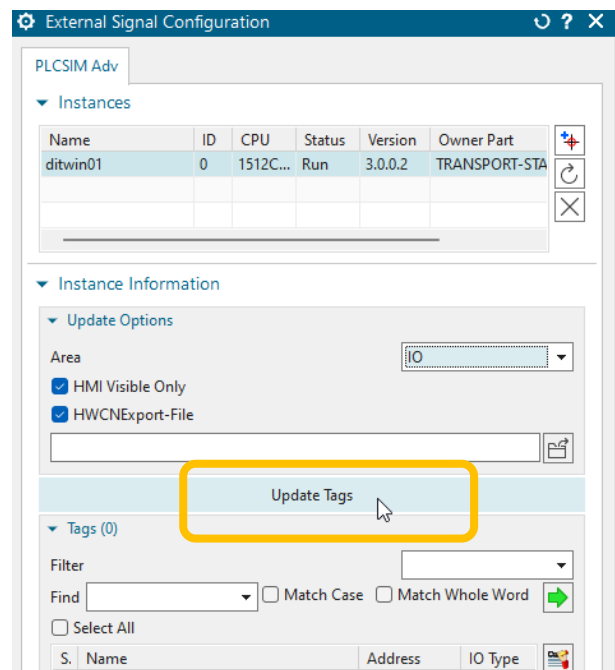
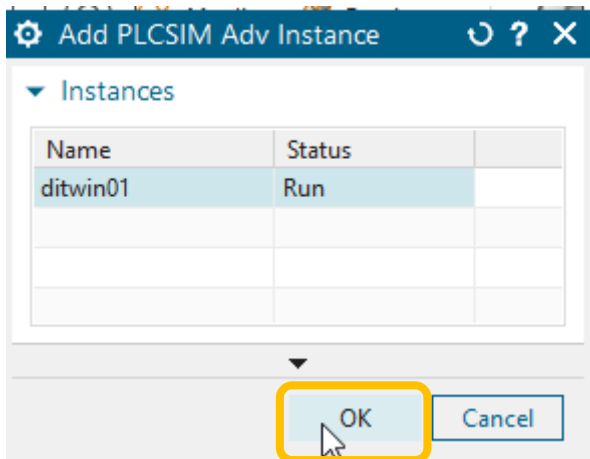


Module 4

Documentation Key Concept Product development

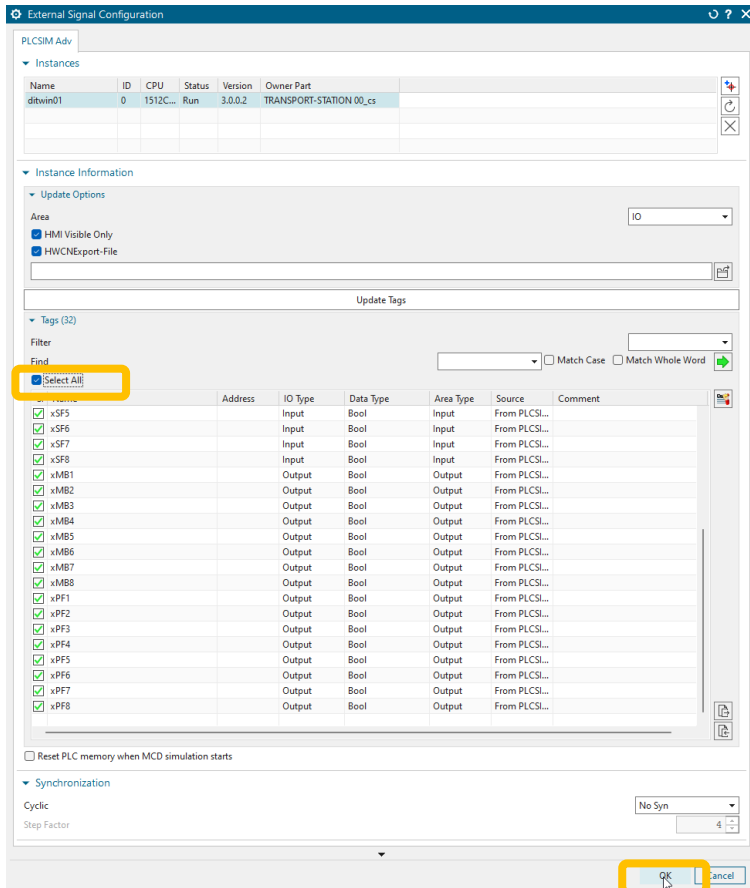


In the signal connections, the soft PLC from PLCSIM Advanced is created as an instance and linked.



Module 4

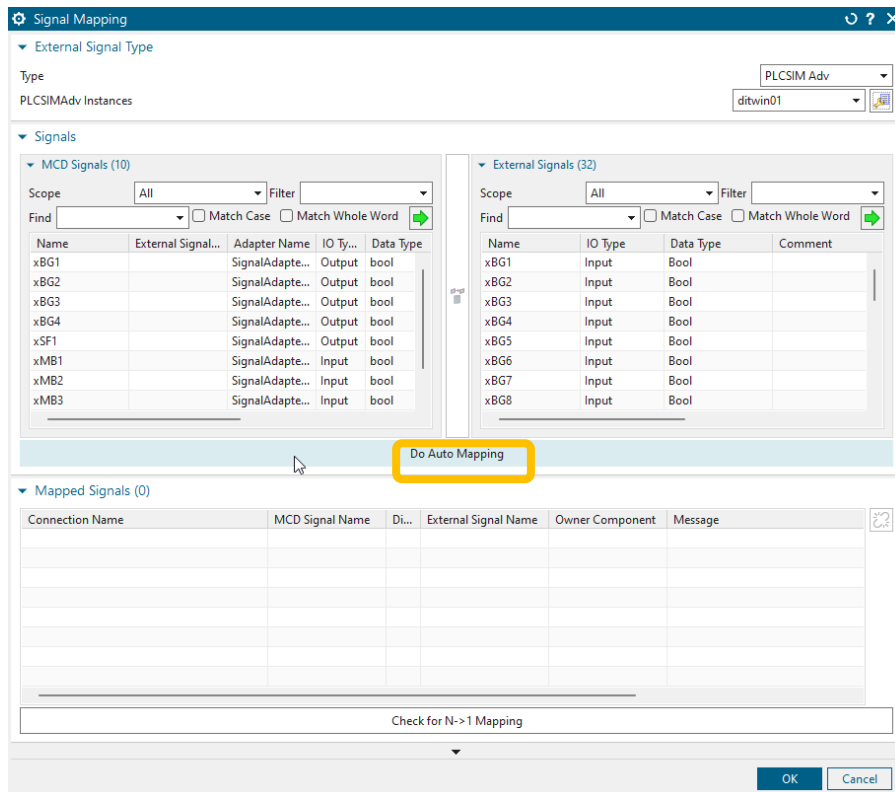
Documentation Key Concept Product development



Select all variables and then select automatic assignment. The automatic assignment can be performed successfully thanks to the identical naming.

Module 4

Documentation Key Concept Product development



Module 4

Documentation
Key Concept Product development



27 Temporary storage and testing of the software status in NX

Video: 27_NX_Save_Test

Video: 27_TIA_Observe

Testing and observation allow the current status to be simulated and checked.

Module 4

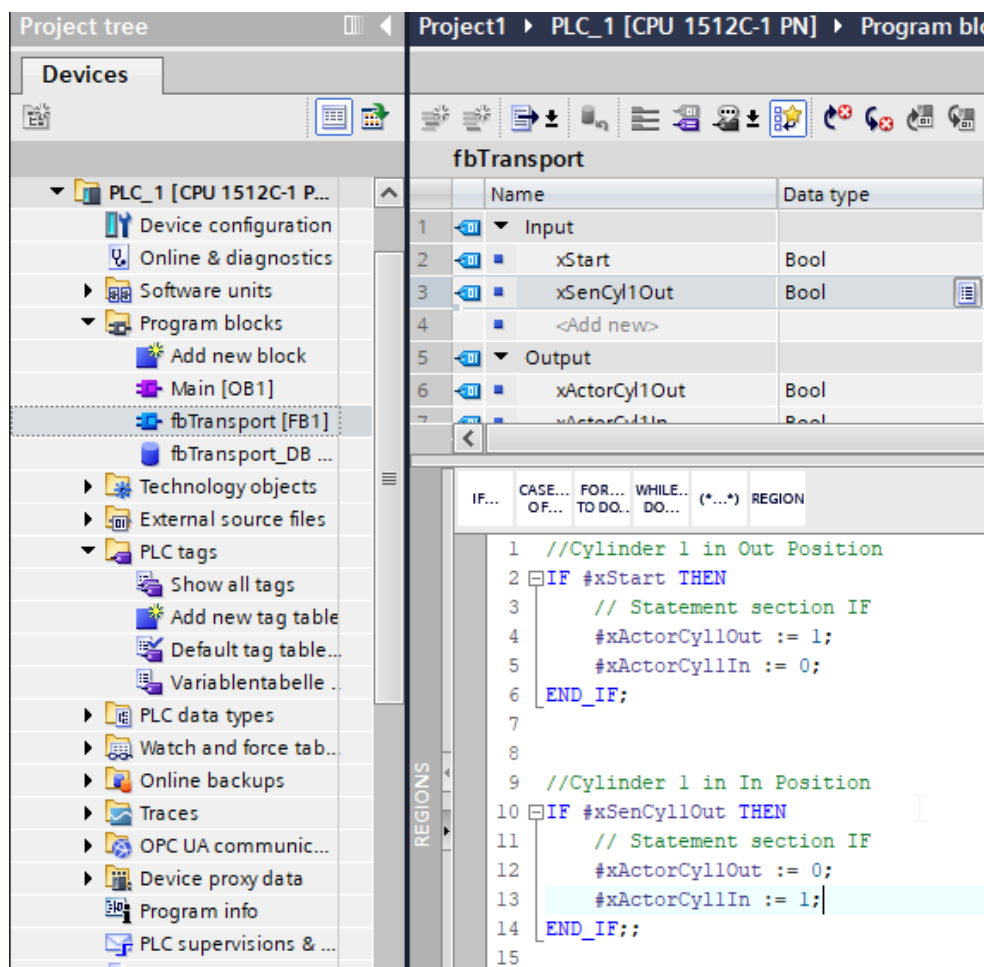
Documentation
Key Concept Product development



28 Testing the software status in the TIA Portal

Video: 28_TIA_Test

The current function block is tested in conjunction with NX MCD and PLCSIM. The variables can also be monitored in observation mode.



Module 4

Documentation
Key Concept Product development



Part 5



Now we are able to program a „step-chain“ code in TIA Portal and test it with our digital twin model in NX MCD.

Module 4

Documentation Key Concept Product development



29 Creation of a step chain program

Video: 29_TIA_Step_Chain

In the next step, the small PLC program for cylinder 1 will be overwritten and replaced by a more comprehensive control program with a step chain.

To do this, new variables must be implemented in the function block in the inputs and outputs area. In addition, the step number with the designation **istep**, which has the default value 0, is added to the static variables area.

fbTransport								
	Name	Data type	Default value	Retain	Accessible f...	Write...	Visible in ...	Setpoint
1	Input							
2	xStart	Bool	false	Non-retain	☒	☒	☒	☐
3	xSenCyl1Out	Bool	false	Non-retain	☒	☒	☒	☐
4	xSenCyl1In	Bool	false	Non-retain	☒	☒	☒	☐
5	xSenCyl2Out	Bool	false	Non-retain	☒	☒	☒	☐
6	xSenCyl2In	Bool	false	Non-retain	☒	☒	☒	☐
7	<Add new>				☐	☐	☐	☐
8	Output							
9	xActorCyl1Out	Bool	false	Non-retain	☒	☒	☒	☐
10	xActorCyl1In	Bool	false	Non-retain	☒	☒	☒	☐
11	xActorCyl2Out	Bool	false	Non-retain	☒	☒	☒	☐
12	xActorCyl2In	Bool	false	Non-retain	☒	☒	☒	☐
13	xConveyorOn	Bool	false	Non-retain	☒	☒	☒	☐
14	<Add new>				☐	☐	☐	☐
15	InOut							
16	<Add new>				☐	☐	☐	☐
17	Static							
18	iStep	Int	0	Non-retain	☒	☒	☒	☐

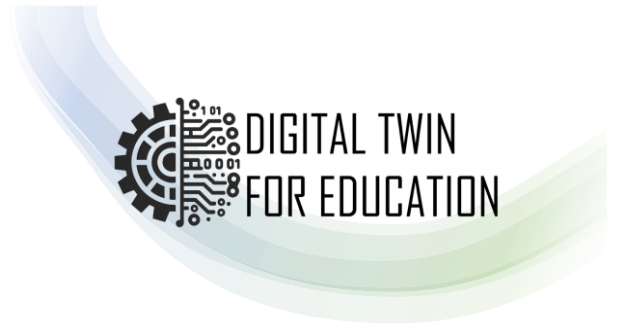
A possible implementation of the step chain in SCL could look like the screenshot below.

```
IF... CASE... FOR... WHILE... ("...") REGION
OF... TO DO... DO...

1 CASE #iStep OF
2 0:
3 IF #xStart THEN // Statement section case 1
4 #iStep := #iStep + 1;
5 END_IF;
6 1:
7 IF #xSenCyl1Out THEN // Statement section case 1
8 #iStep := #iStep + 1;
9 END_IF;
10 2:
11 IF #xSenCyl1In THEN // Statement section case 1
12 #iStep := #iStep + 1;
13 END_IF;
14 3:
15 IF #xSenCyl2Out THEN // Statement section case 1
16 #iStep := #iStep + 1;
17 END_IF;
18 4:
19 IF #xSenCyl2In THEN // Statement section case 1
20 #iStep := 0;
21 END_IF;
22 END_CASE;
23
```

Module 4

Documentation
Key Concept Product development



30 Control of the outputs of the step chain program

Video: 30_TIA_New_Prog_Chain

After the advance conditions were defined in the previous step, the control of the outputs is now defined in the program.

To do this, only the step numbers are assigned to the individual outputs and connected using a basic logical OR (OR) link.

Module 4

Documentation Key Concept Product development



fbTransport

	Name	Data type	Default value	Retain	Accessible f...	Writa...	V
1	Input				☐	☐	
2	xStart	Bool	false	Non-retain	☒	☒	
3	xSenCyl1Out	Bool	false	Non-retain	☒	☒	

IF... CASE... OF... FOR... TO DO... WHILE... DO... (*...*) REGION

```
1 CASE #iStep OF
2   0:
3     IF #xStart THEN // Statement section case 1
4       #iStep := #iStep + 1;
5     END_IF ;
6   1:
7     IF #xSenCyl1Out THEN // Statement section case 1
8       #iStep := #iStep + 1;
9     END_IF;
10  2:
11    IF #xSenCyl1In THEN // Statement section case 1
12      #iStep := #iStep + 1;
13    END_IF;
14  3:
15    IF #xSenCyl2Out THEN // Statement section case 1
16      #iStep := #iStep + 1;
17    END_IF;
18  4:
19    IF #xSenCyl2In THEN // Statement section case 1
20      #iStep := 0;
21    END_IF;
22
23 END_CASE;
24
25 //Controlling the outputs
26 #xActorCyl1Out := (#iStep = 1);
27 #xActorCyl1In := (#iStep = 0) OR (#iStep = 2) OR (#iStep = 3) OR (#iStep = 4);
28 #xActorCyl2Out := (#iStep = 3);
29 #xActorCyl2In := (#iStep = 0) OR (#iStep = 1) OR (#iStep = 2) OR (#iStep = 4);
30 #xConveyorOn := (#iStep = 4);
31
```

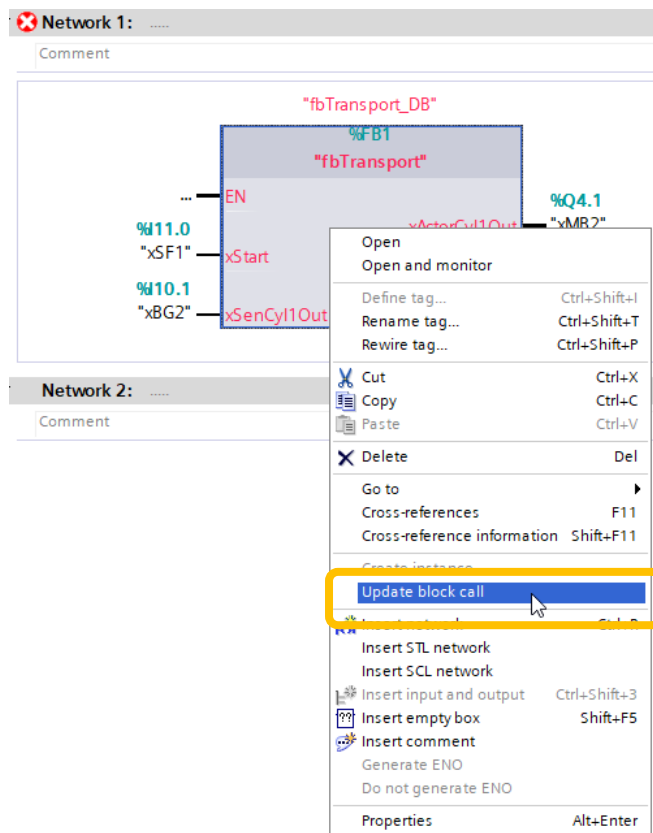
Module 4

Documentation Key Concept Product development



The adjustments in the function block also require adjustments to be made in the MAIN OB1 block with regard to variable assignment.

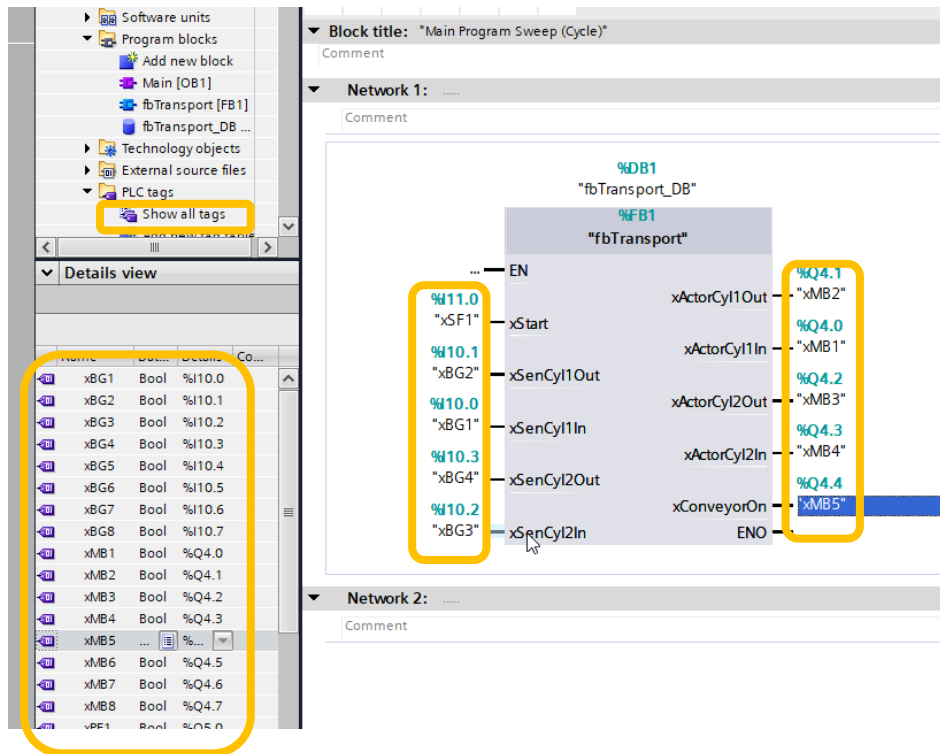
To do this, the content in MAIN OB1 must first be updated so that the PLC variables can then be reassigned.



Simply click on Show all tags once and assign them to MAIN OB1 using drag & drop.

Module 4

Documentation Key Concept Product development

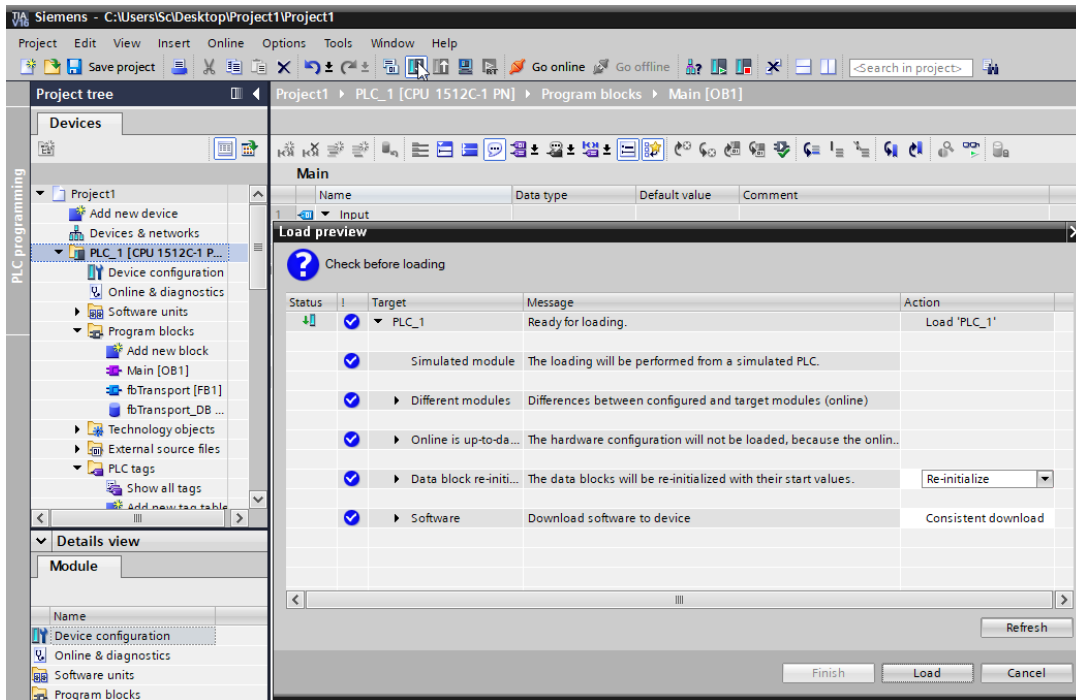


Module 4

Documentation Key Concept Product development



The current software version can then be reloaded from the TIA Portal into the soft PLC in PLCSIM Advanced.



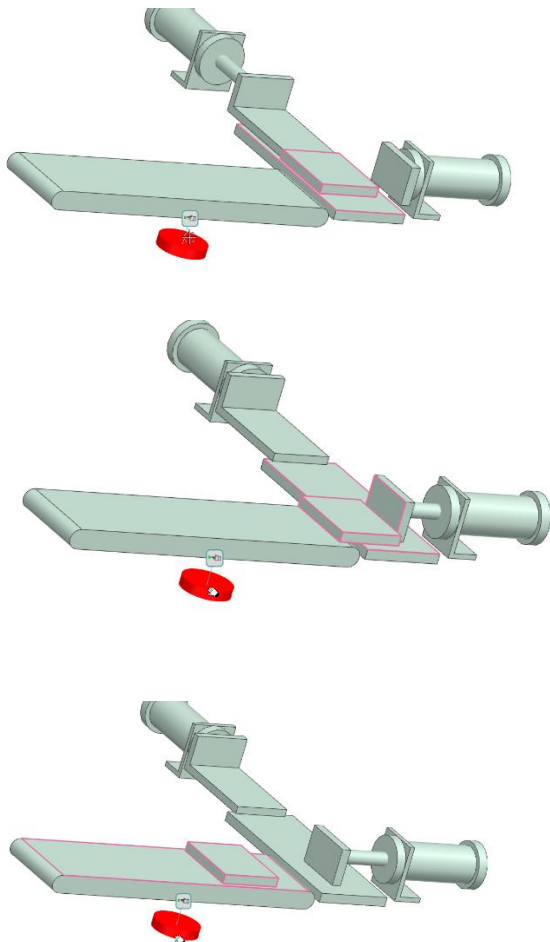
Module 4

Documentation Key Concept Product development

31 Testing the software version with NX

Video: 31_NX_Prog_Chain_Test

Finally, the newly programmed step chain can be tested with the NX model. To do this, first start the simulation (RUN) and press the Start button.



Module 4

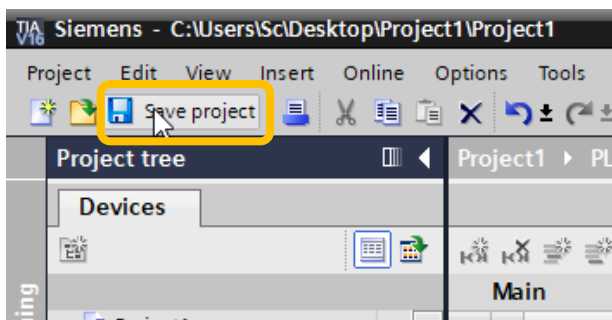
Documentation Key Concept Product development



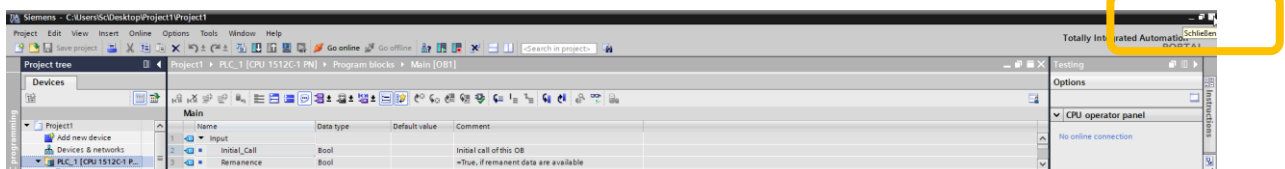
32 Save status and close entire project

Video: 32_Save_Close_All

Finally, all project statuses should be saved so that variations and adjustments can be made afterwards.



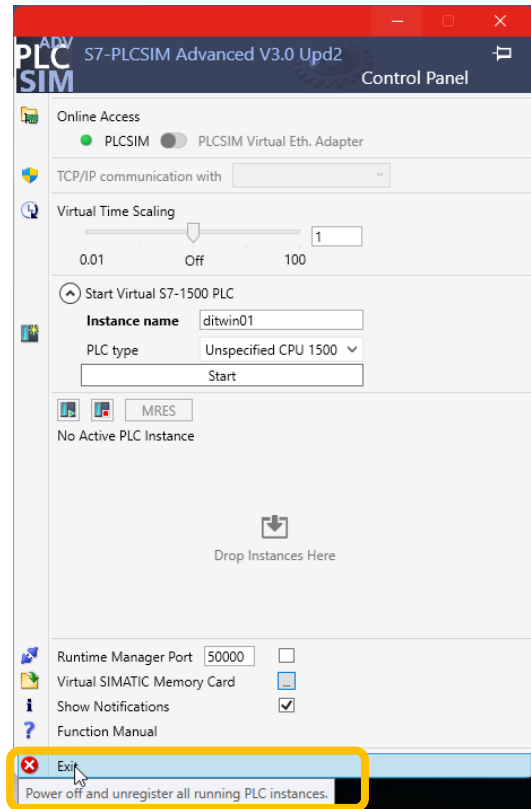
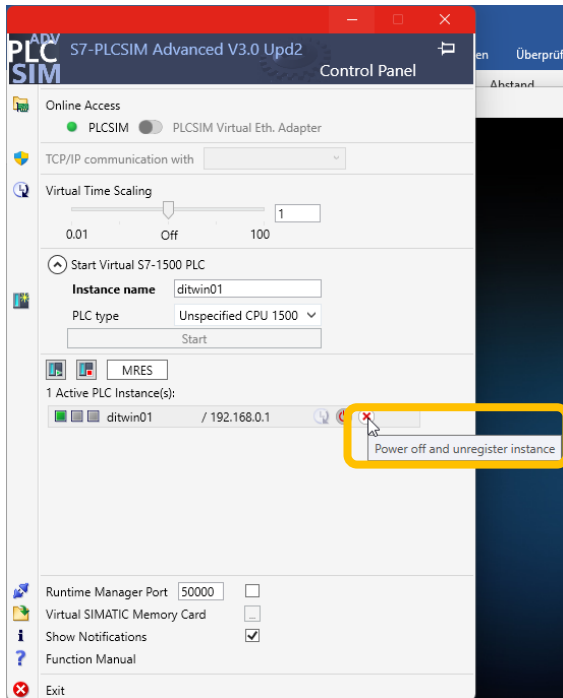
The TIA Portal project can then be closed.



To close PLC SIM Advanced properly, it is best to proceed as shown.

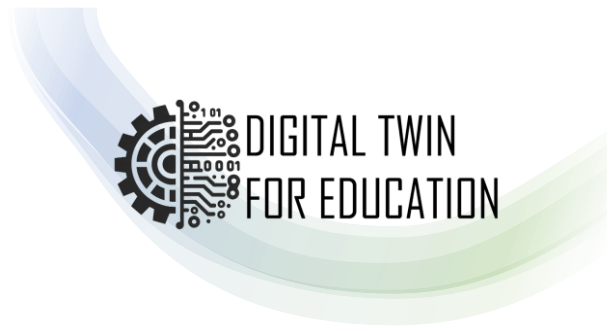
Module 4

Documentation Key Concept Product development



Module 4

Documentation Key Concept Product development



33 Reopen the SimCaer virtual PLC

Video: 33_SIM_reopen-Simcard

In this exercise, the virtual controller, or rather the SIM card within it, is to be restarted. To do this, Siemens PLCSIM Advanced is opened, and the SIM card is dragged back into the designated slot. Afterwards, the PLC can be started again.

34 Reopen the NX simulation project

Video: 34_NX_reopen-Simulate

After the Soft-PLC is operational again, the NX program can be reopened. To do this, the corresponding file is opened—this is a part file for the conveyor belt—which allows the simulation to be started again.

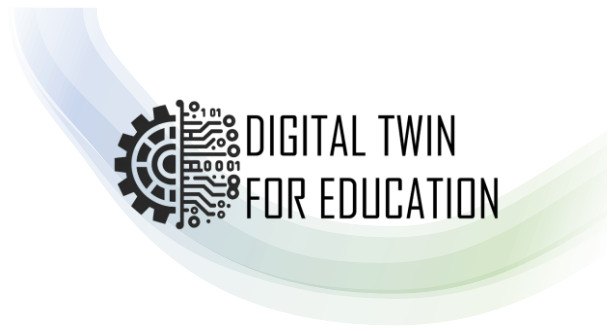
35 Integration of a stop button in the NX simulation

Video: 35_NX_Stop-Button

In the simulation, an additional button is to be integrated. This button should be placed next to the start button and will later function as a stop button.

Module 4

Documentation Key Concept Product development



36 Integration of a text description in NX

Video: **36_NX-Text**

The button should be labeled in the same way as the start button. For this purpose, a corresponding label can be attached to the respective button so that the machine operator can later identify which functions are assigned to each button.

37 Embed the function runtime button for the stop signal

Video: **37_NX-Stop-Signal**

In this step, the function of a runtime button is to be assigned to the stop button as well. This is carried out in the same way as for the start button, following the procedure shown in the video.

38 Adjustment for the stop signal in the signal adapter

Video: **38_NX-Change-Signal-Adapter**

For the stop button, minor adjustments must subsequently be made in the signal adapter. In this case, the stop button must be declared as an output, since a sensor in the simulation assumes the function of an output.

Module 4

Documentation Key Concept Product development



39 Adjustment for the stop signal in the signal connection

Video: **39_NX-Change-Signal-Conn**

For communication, the new stop button labeled SF2 must also be assigned to the TIA program in the Signal Connection. This assignment is carried out within the Signal Connection.

40 Adjustments in the TIA Program for the stop signal

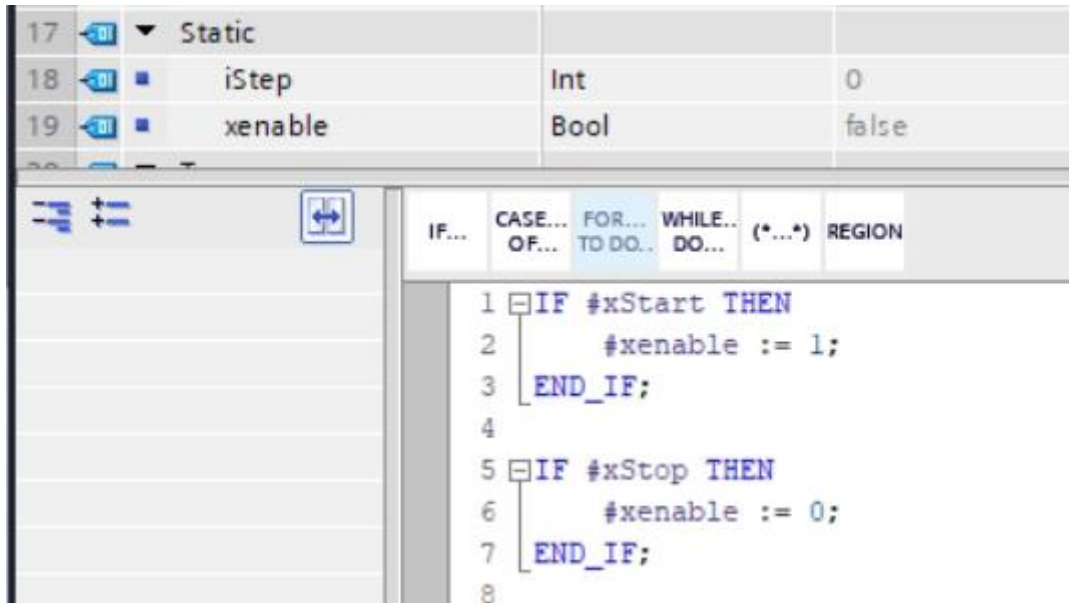
Video: **40_TIA-Change-Prog**

In this step, the hardware-related adjustments for the stop button are to be integrated into the TIA Portal program. For this purpose, the function block is called, and the modifications are implemented using a new variable for the stop button.

To integrate the operating state, an additional static variable is required in the Static section to represent the state of the machine or process. This variable is designated as xENABLE.

Module 4

Documentation
Key Concept Product development



41 Environment test of the stop button in the NX simulation

Video: **41_NX-Test-Stop**

In the next step, the modification and integration of the stop button are tested in the simulation using Siemens NX Mechatronics Concept Designer. To do this, the simulation is started via the Run button, and it is tested whether the process stops when pressing the second button.

Module 4

Documentation Key Concept Product development



42 Integration of two new collision sensors in the NX simulation

Video: **42_NX-New-Coll-Sensors**

In this step, two new collision sensors are integrated into the MX model. One collision sensor is positioned in the middle of the conveyor belt, and another at the end of the belt at workpiece height, to monitor the conveyor system. The collision sensors will later be used in the process sequence to provide feedback on the workpieces located on the conveyor belt.

43 Adjustment for the collision sensors in the signal adapter

Video: **43_NX-Signal-Ad-conn**

In this step, the two new sensors are linked with their respective parameters in the signal adapter to enable communication. Since they are sensors, they are treated as outputs in the simulation and are parameterized accordingly in the signal adapter.

44 Integration of a source and sink for the production parts

Video: **44_NX-Source-Sink**

In the next step, the workpiece is defined as a source. This means that a new workpiece repeatedly appears at the workpiece position. In other words, the source regularly generates a new workpiece when triggered, for example by pressing the start button, to keep the process running continuously.

Subsequently, a sink is defined where the workpieces disappear after reaching it. This ensures a continuous flow, where a new workpiece is introduced while another one automatically disappears at the end of the process. This principle is also referred to as source and sink.

Module 4

Documentation Key Concept Product development



45 Adjustment for workpiece source in the signal adapter

Video: **45_NX-Internal-Signal_Adapter**

In the further course, the object source is integrated into the signal adapter in order to control the workpieces and their appearance.

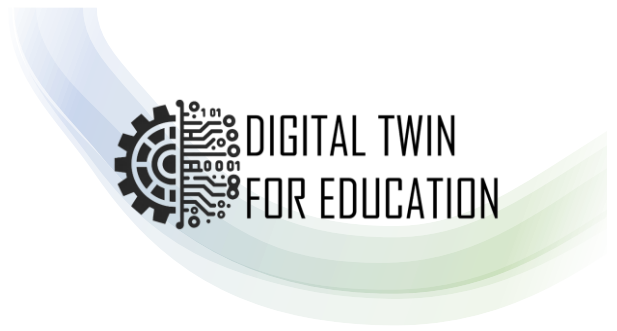
46 Adjustment for a rising edge in the signal adapter

Video: **45_NX-rising-edge**

In order for the components to disappear at the workpiece sink, a setting must be configured in the signal adapter. This setting is based on a rising edge, which must be assigned to the corresponding parameter. The required adjustments can be followed in the video, where the appropriate configuration is carried out in the signal adapter using the “Rising Edge” command.

Module 4

Documentation Key Concept Product development



Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.