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

CARL-BENZ-SCHULE GAGGENAU
Mercantec
Co-funded by the European Union
2024-1-DE02-KA210-VET-000246677
www.digitwin4teach.eu

Module 4

Documentation
Key Concept Product development

DIGITAL TWIN FOR EDUCATION

Table of Contents

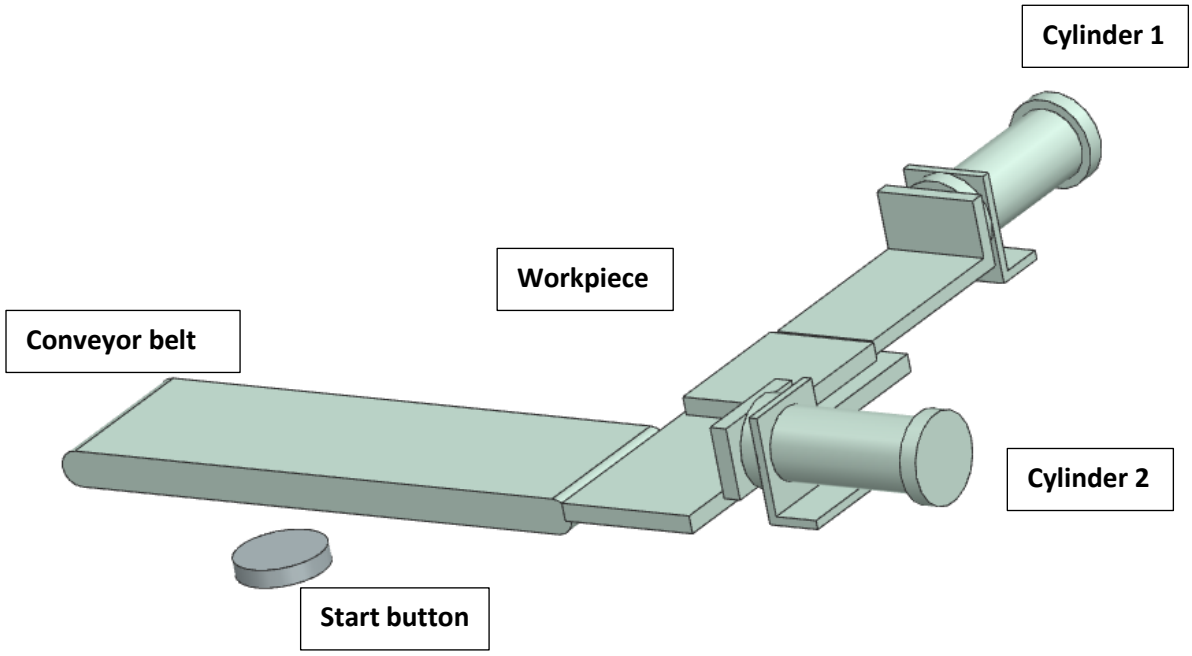
Note: The numbering of the captions starts with 13 because of the continuing of the process chain. In Module 3 you already made the steps 1 to 12.

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.

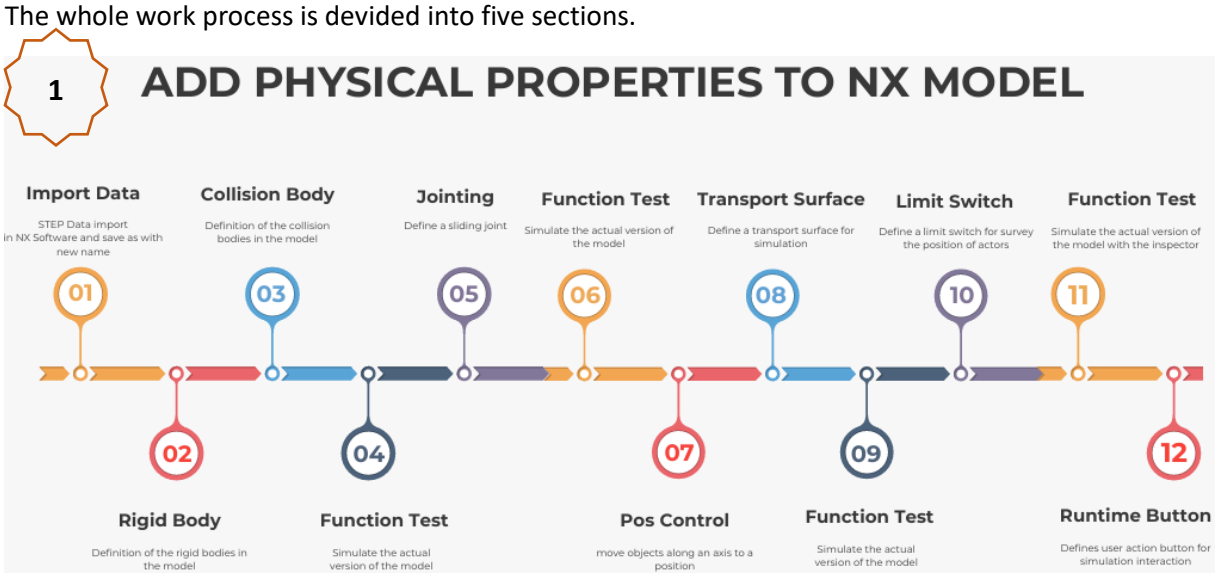
An additional start button can be used to initiate a sequence of steps for the sequence control. The double-acting pneumatic cylinders are controlled via solenoid valves and their position is monitored by limit switches.

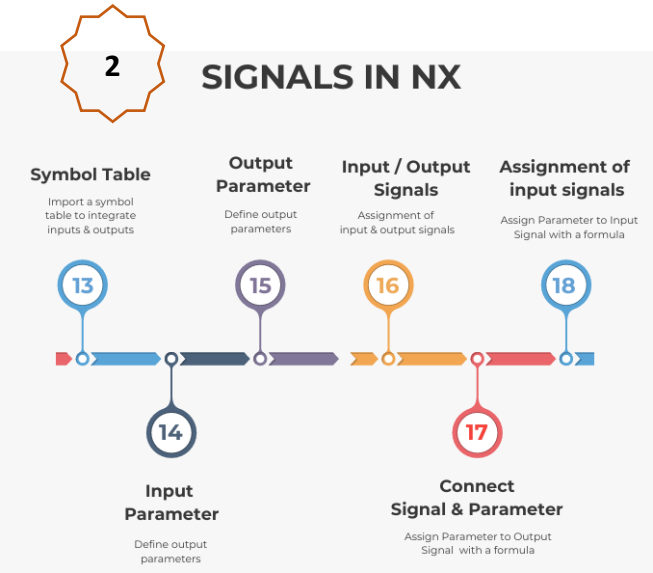


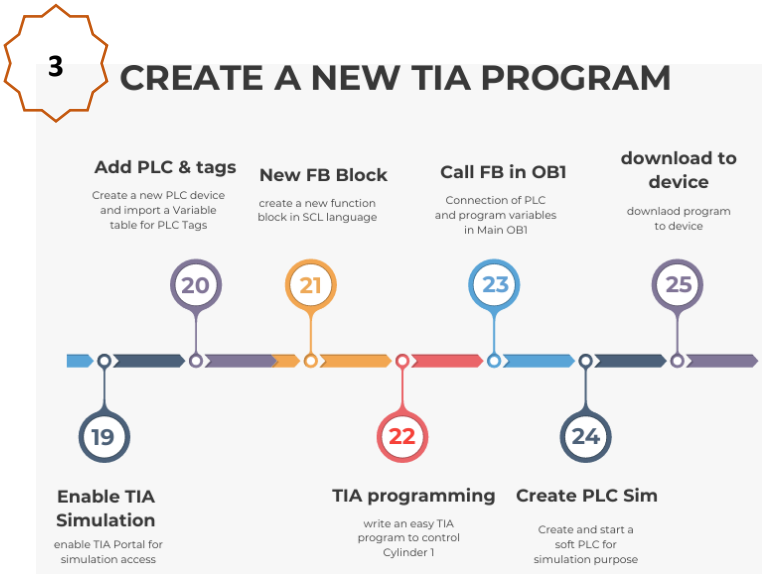
In the following work process, the already existing CAD model (assembly in STEP file format) is imported to Siemens NX MCD .

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 devided into five sections.

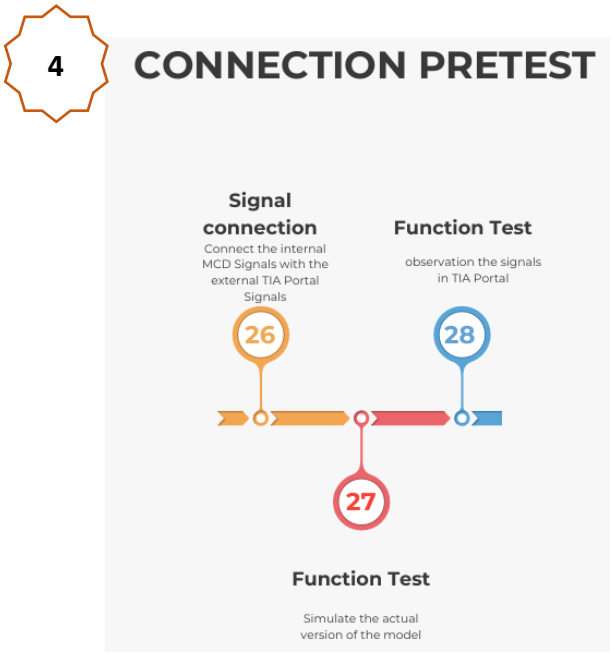


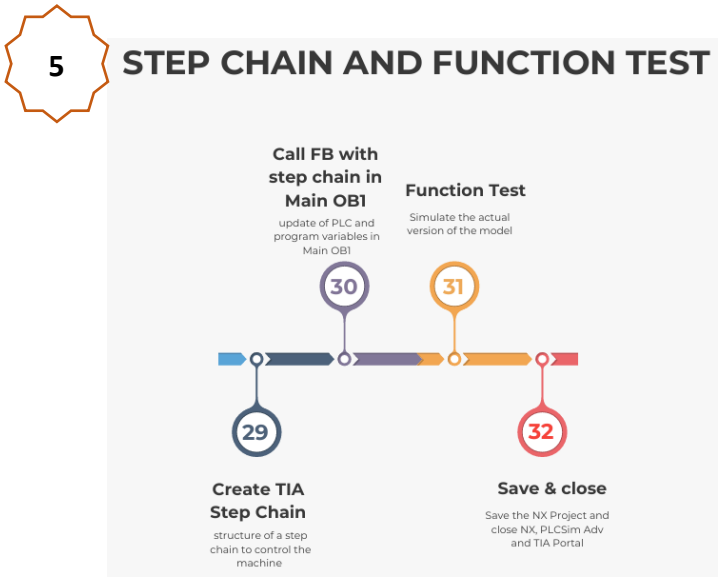




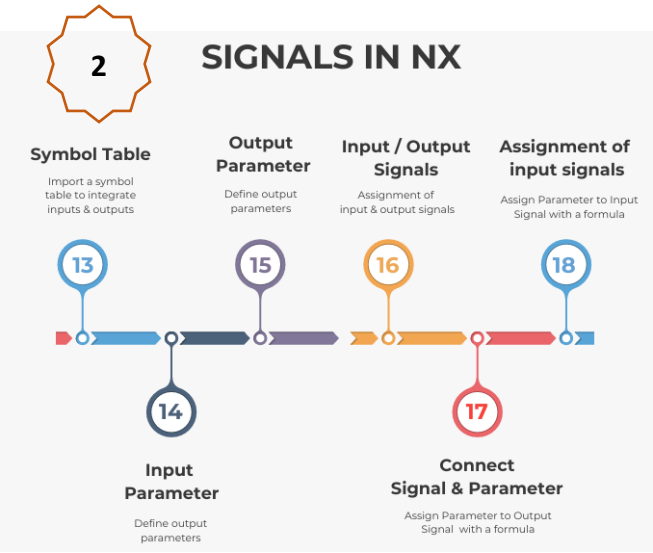
Module 4

Documentation
Key Concept Product development





Part 2



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.

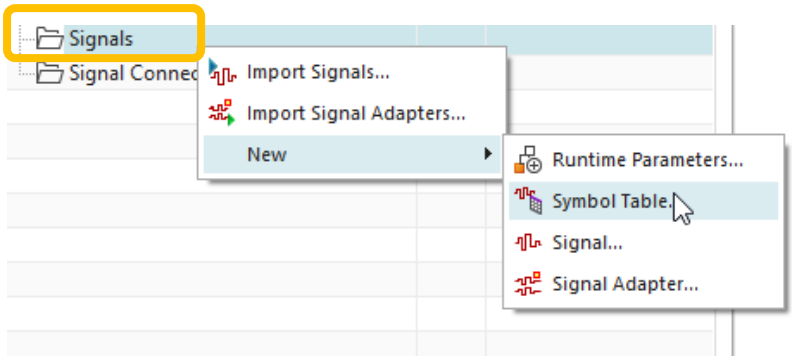
This mapping takes place in the **Signal Adapter**, which forms the core of communication between the digital model and the PLC Software.

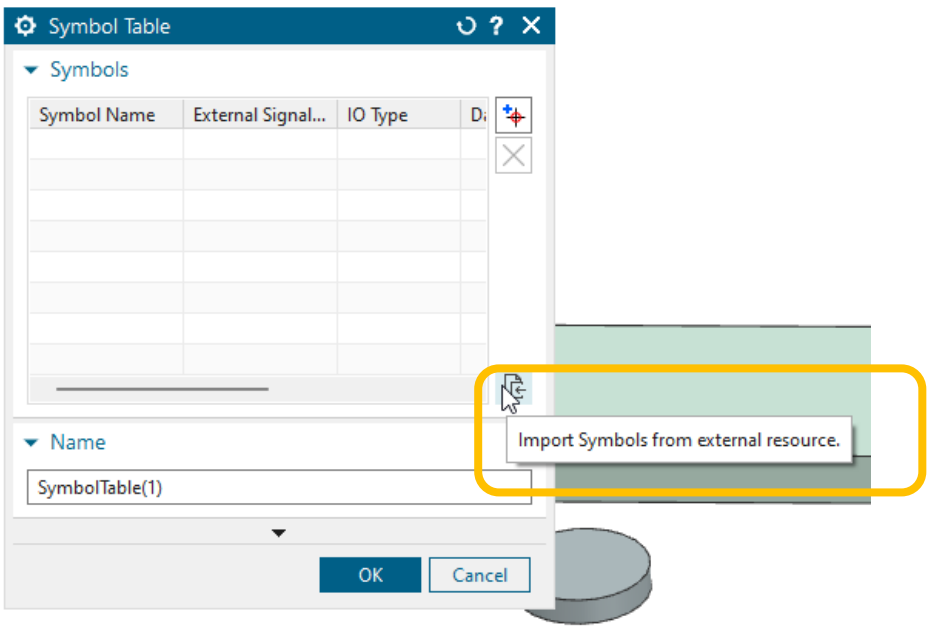
13 Importing a symbol table for variable names

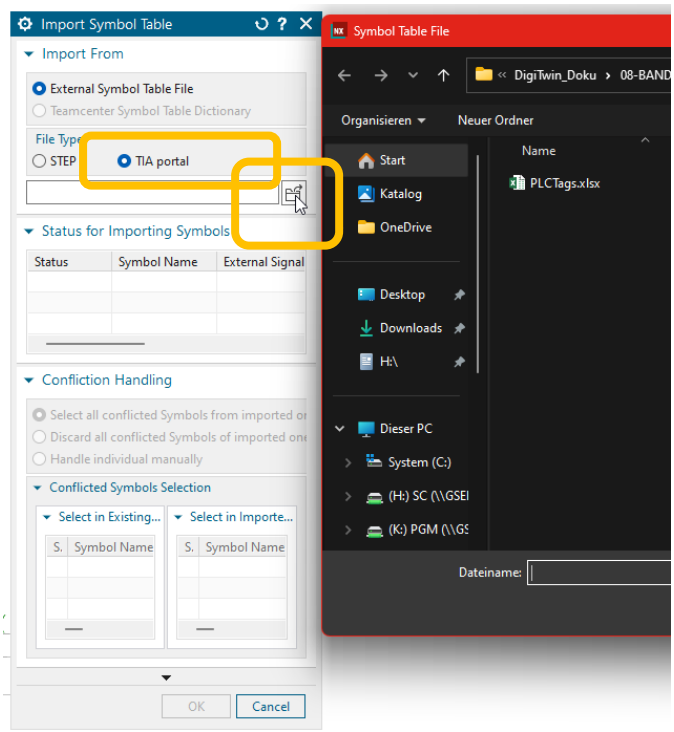
Video: 13Import Symbols

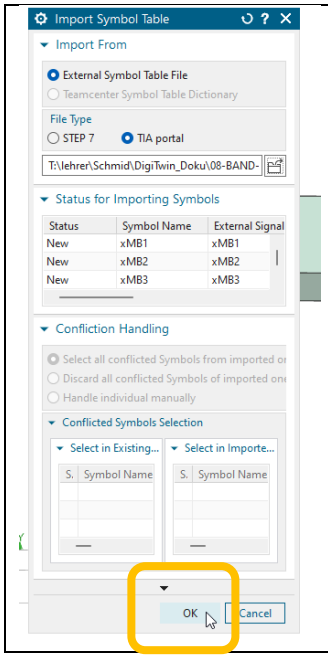
The next step is to import a symbol table containing the variable names of the PLC variables from the TIA Portal project in order to facilitate the signal mapping with the PLC program later on.

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

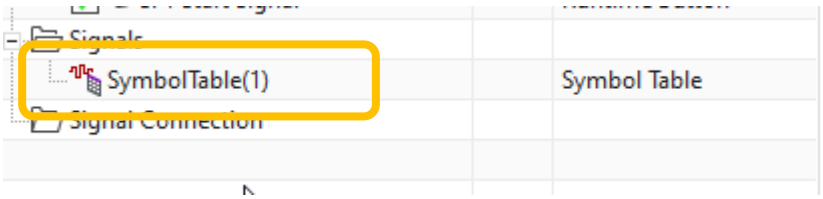






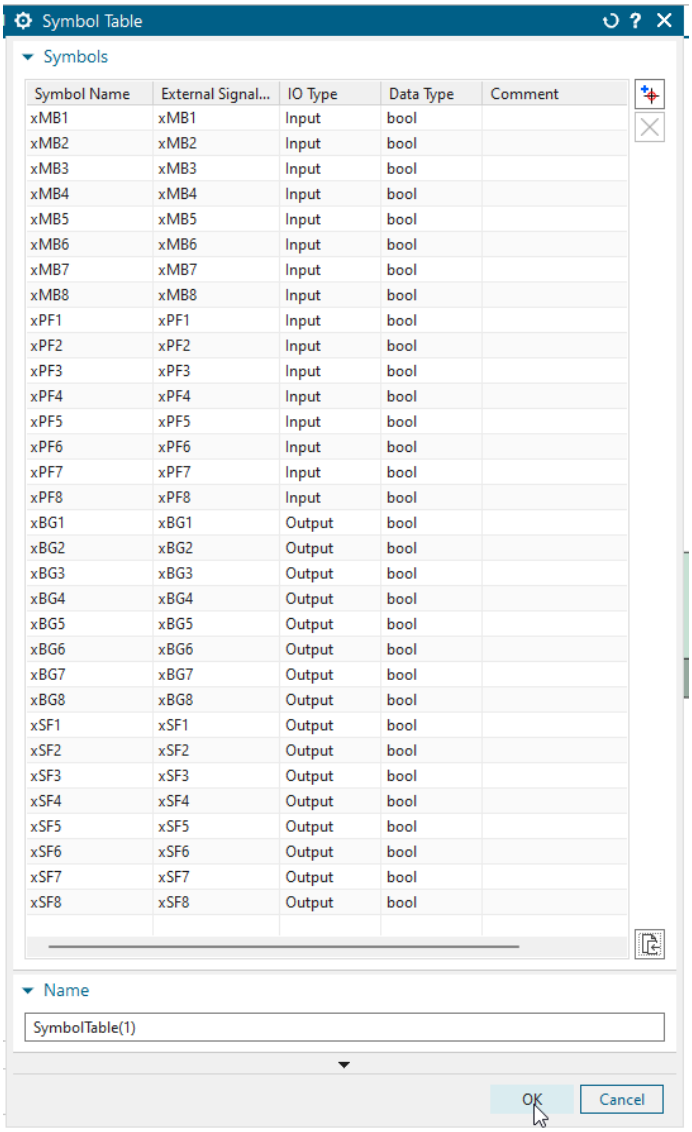


The symbol table is then generated automatically.



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.

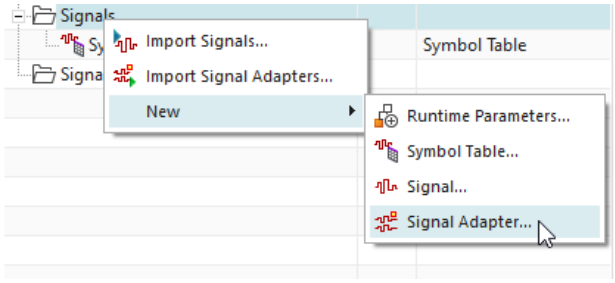
The reason for this is that the variables are viewed from the perspective of the NX model, and in this case, a sensor is an output because it delivers a signal to the controller, and the solenoid valve is an input because the signal comes into the NX model as an input signal from the PLC.

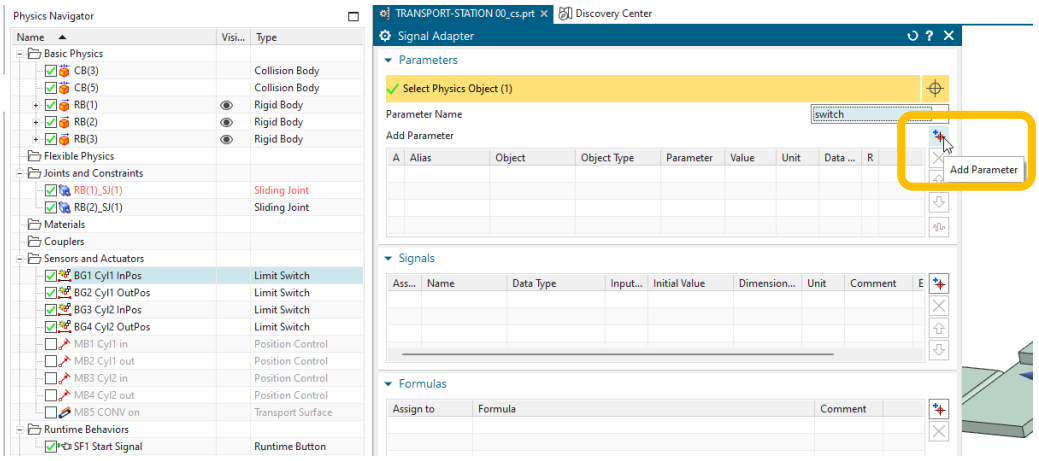


14 Selection and definition of parameters for the inputs in NX

Video: 14Signals-Ad-Inputs

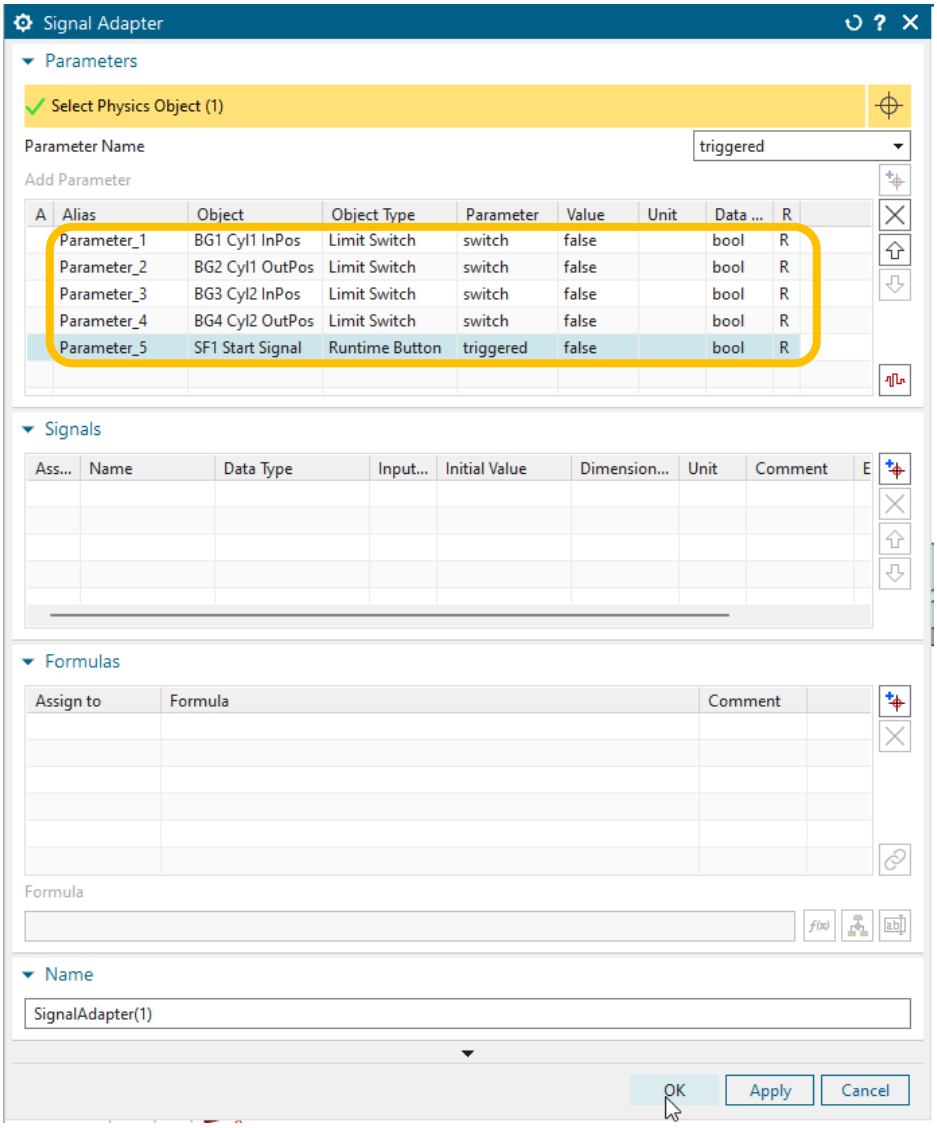
In this step, parameters for individual sensors are defined specifically according to the respective application.





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

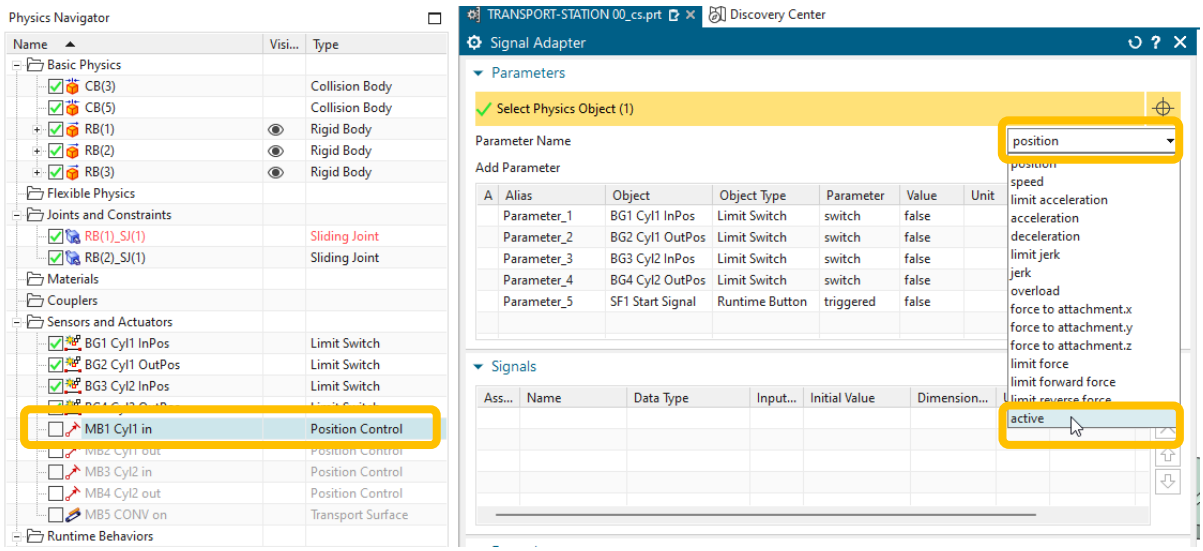
Please check the overall view about the sensor situation.



15 Selection and definition of parameters for outputs in NX

Video: 15Signals-Ad-Outputs

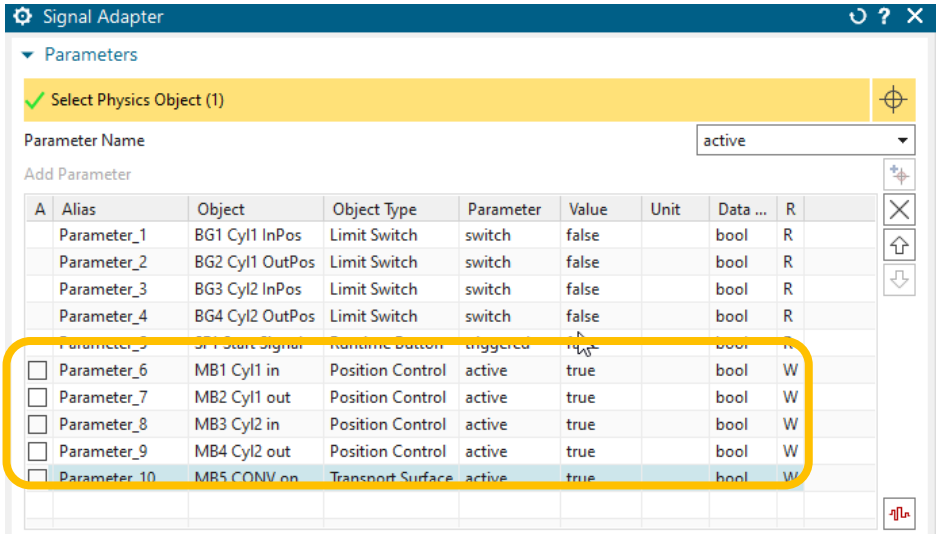
In this step, parameters for individual actuators are defined specifically for the respective application.



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

Module 4

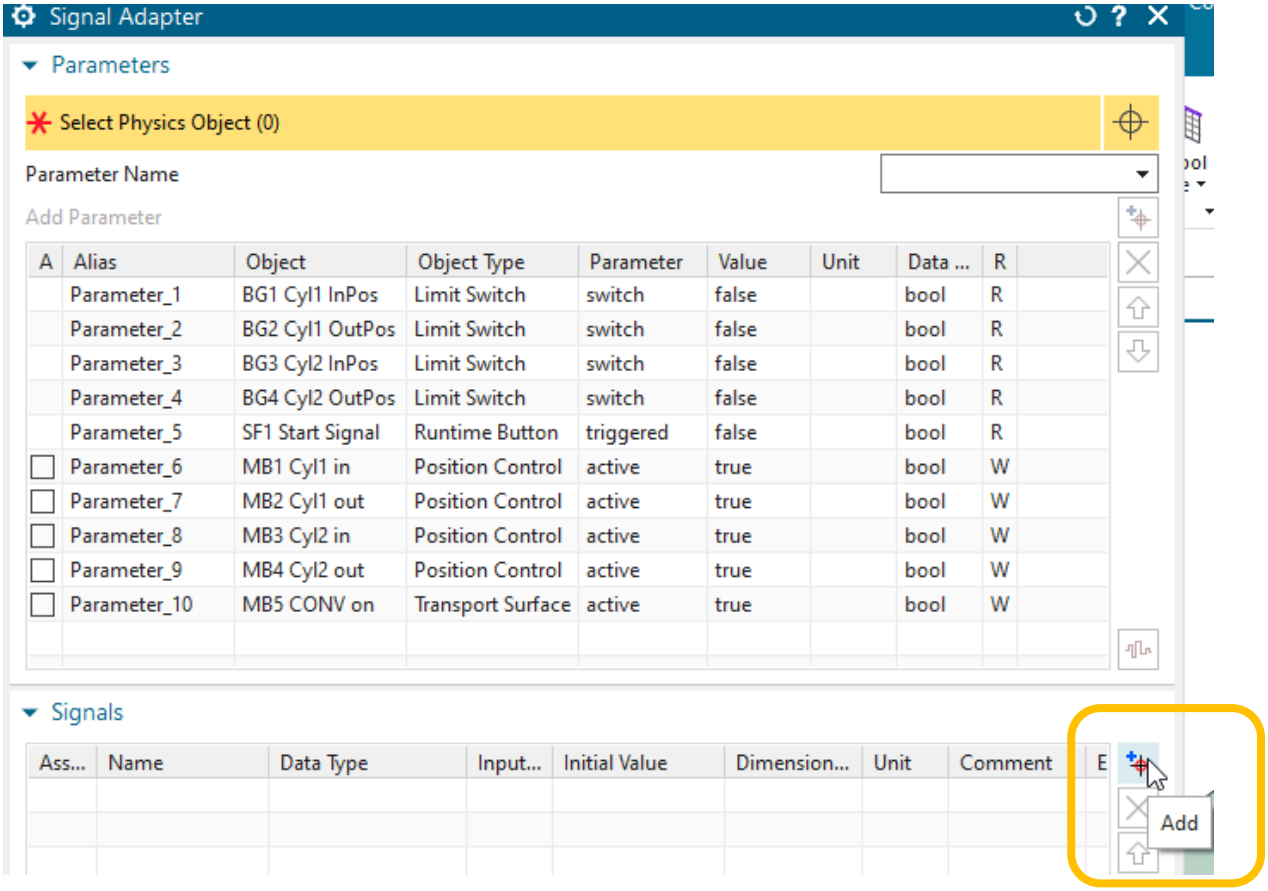
Documentation
Key Concept Product development



16 Assignment of signals in the signal adapter

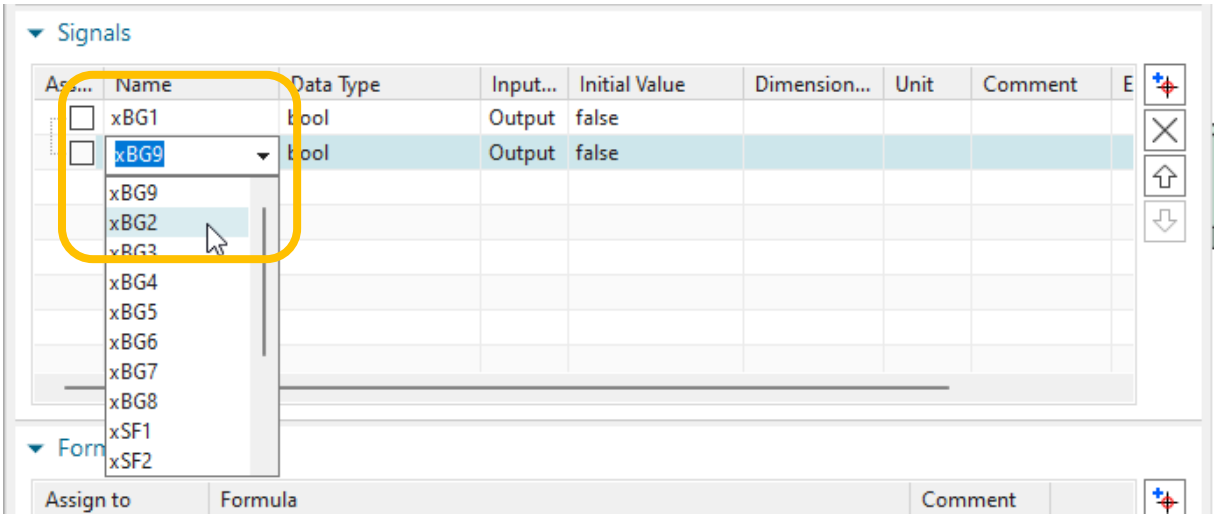
Video: 16Add_Signals

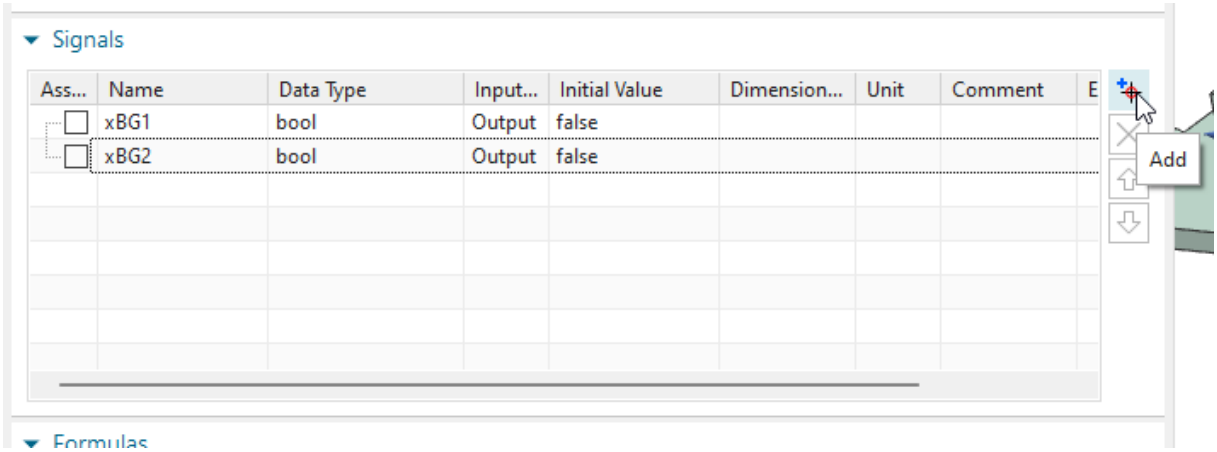
All input and output signals are then created in the signal adapter.

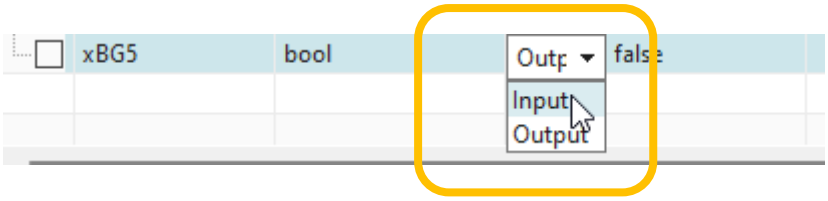


The assignment of whether it is an INPUT or OUTPUT is made from the perspective of the NX model. This means that an actuator is defined as an INPUT and a sensor as an OUTPUT.

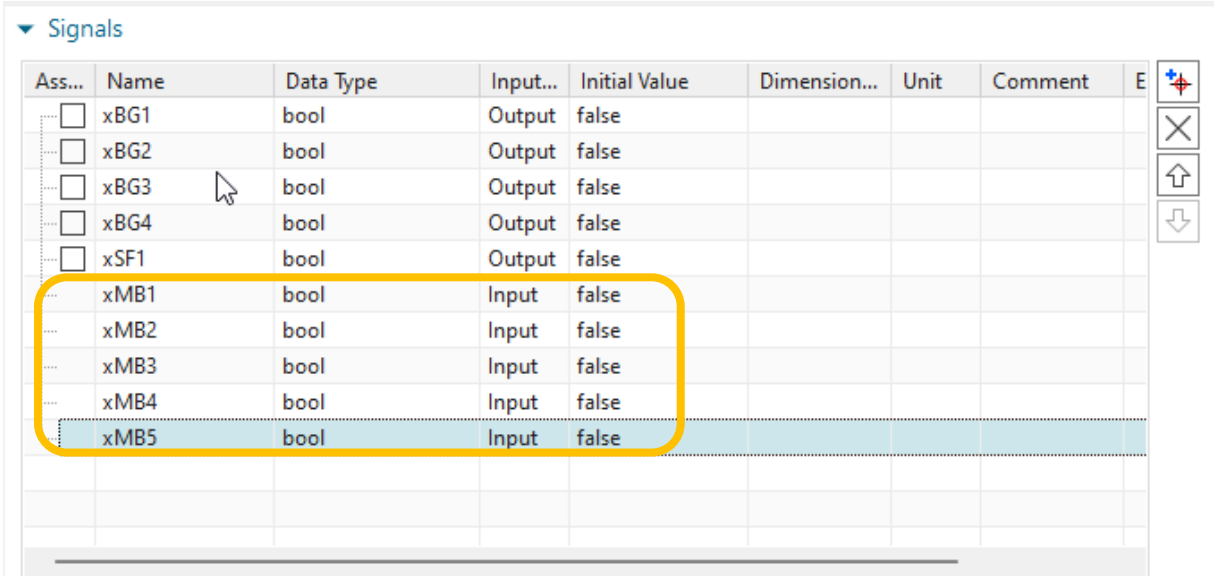
After the import of the sensors (Outputs for the simulation) the topic must be change for the solenoid valves to an Input.







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.

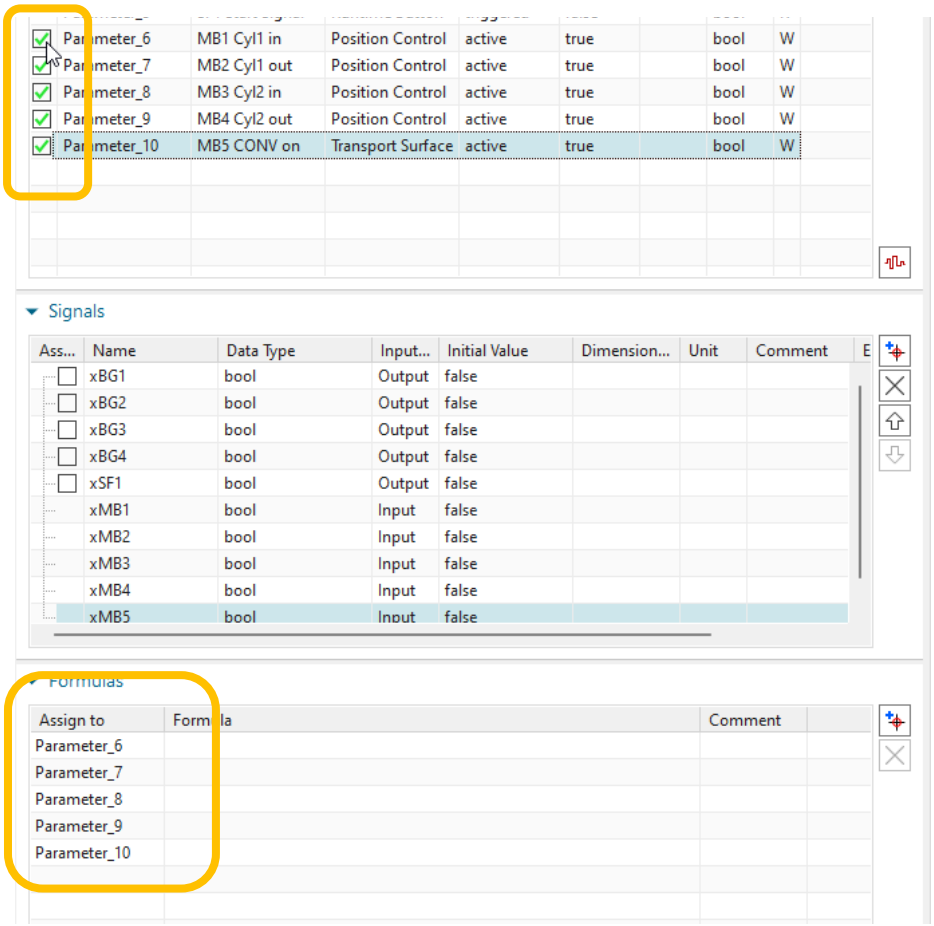


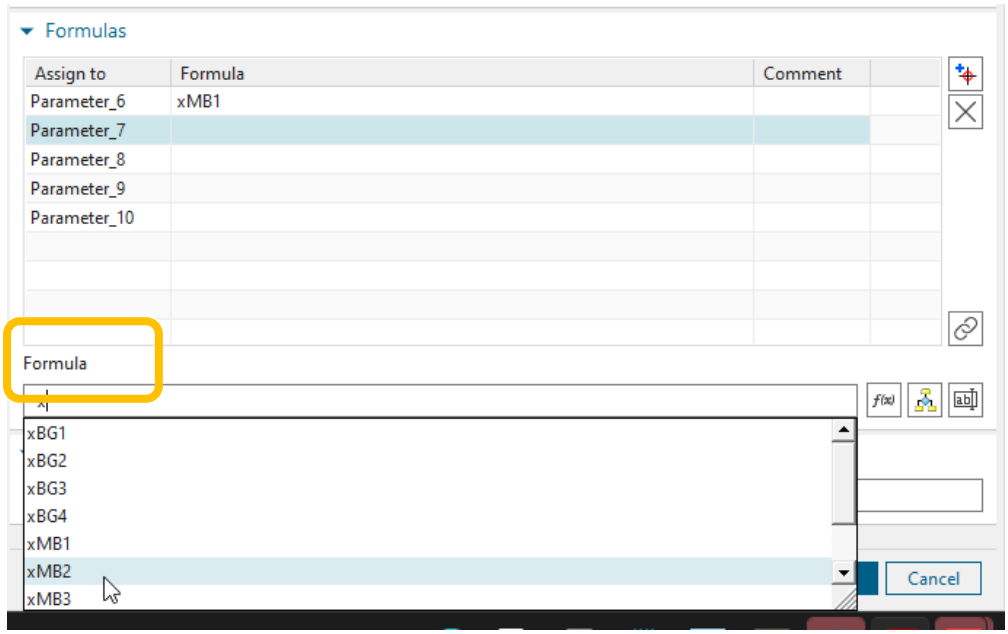
17 Connecting parameters and signals in the signal adapter

Video: 17Connect_MB

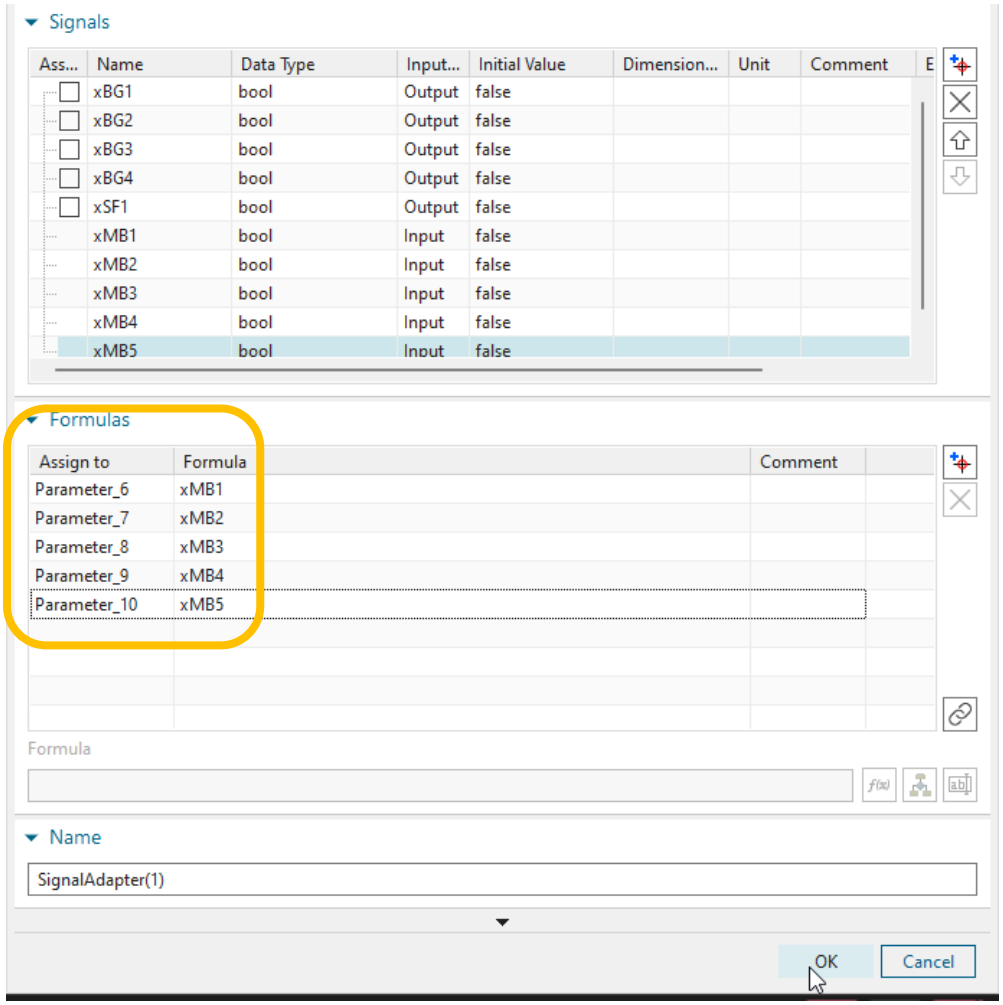
In this step, the individual signals are manually merged with the respective matching parameter.

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





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


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
xSF1 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

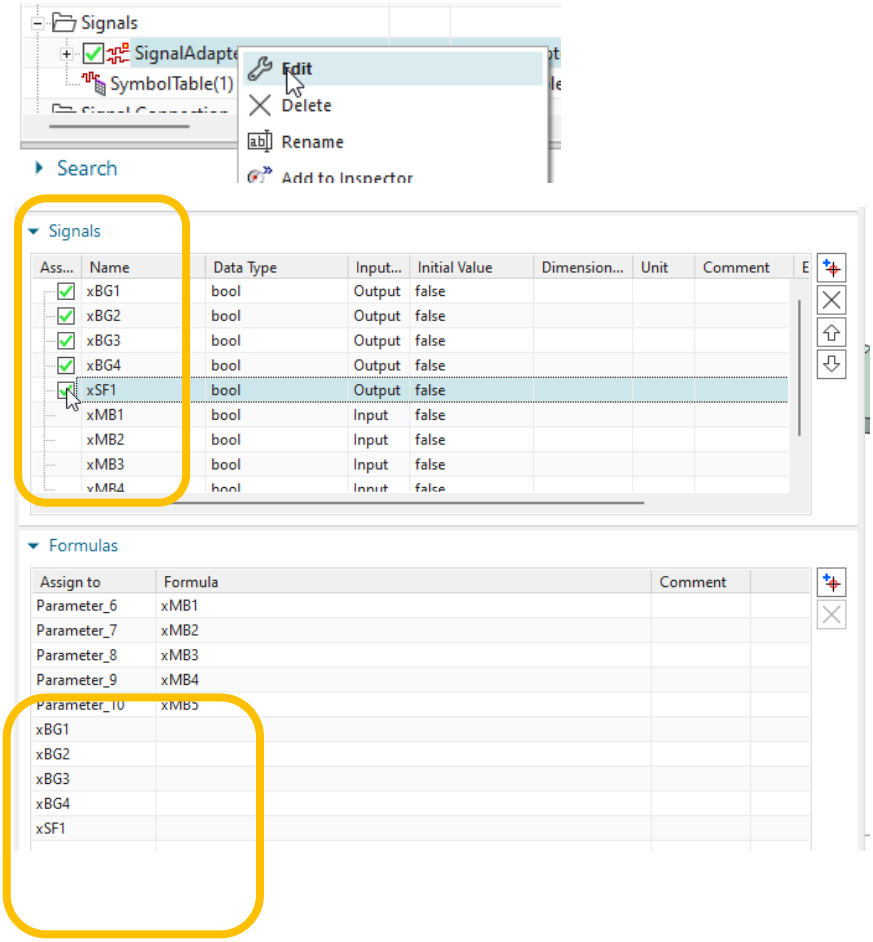
18 Connecting input parameters and signals in the signal adapter

Video: 17Connect_BG

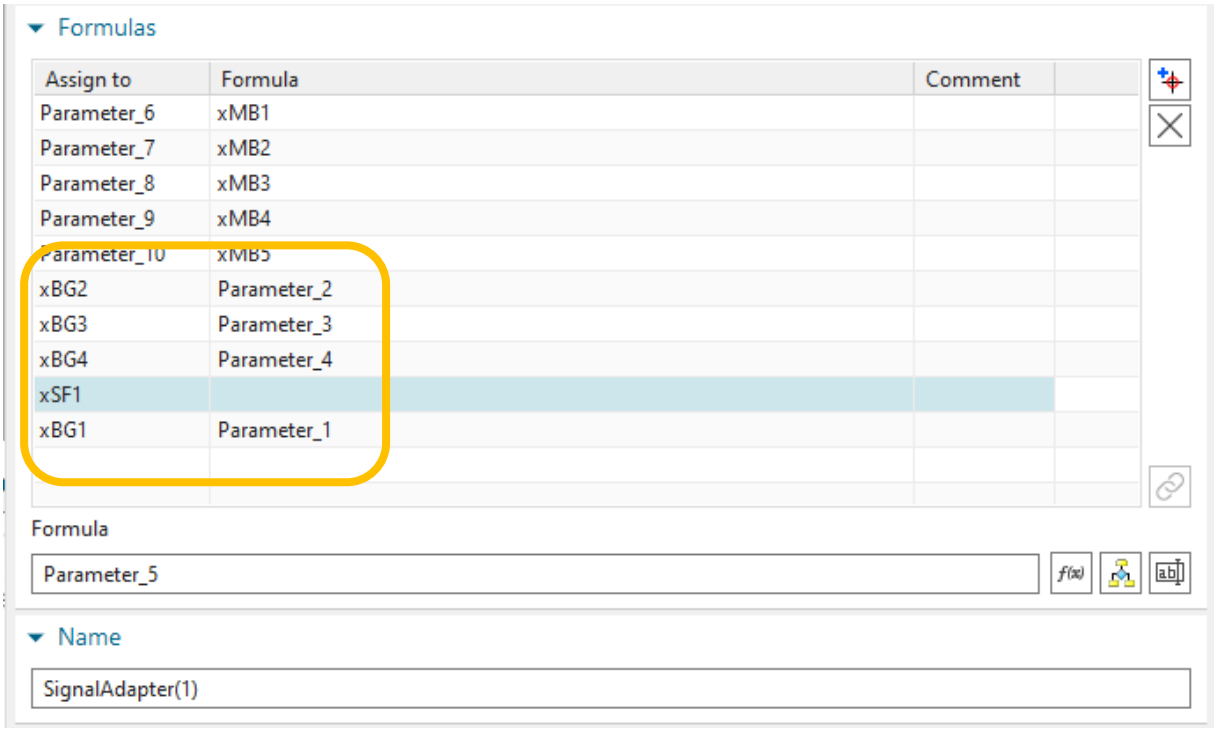
In this step, the individual signals are manually merged with the respective matching parameters.

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

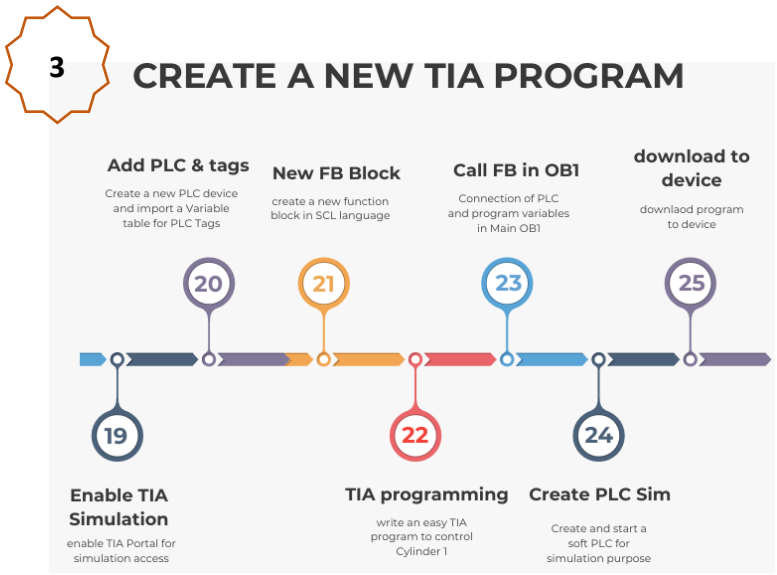
This corresponds to a reverse assignment as in the previous point.



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



Part 3



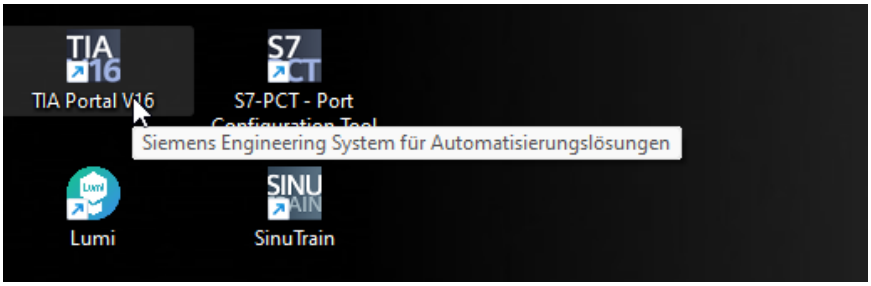
In this section we create the TIA Portal Project for the programming part of the digital twin.

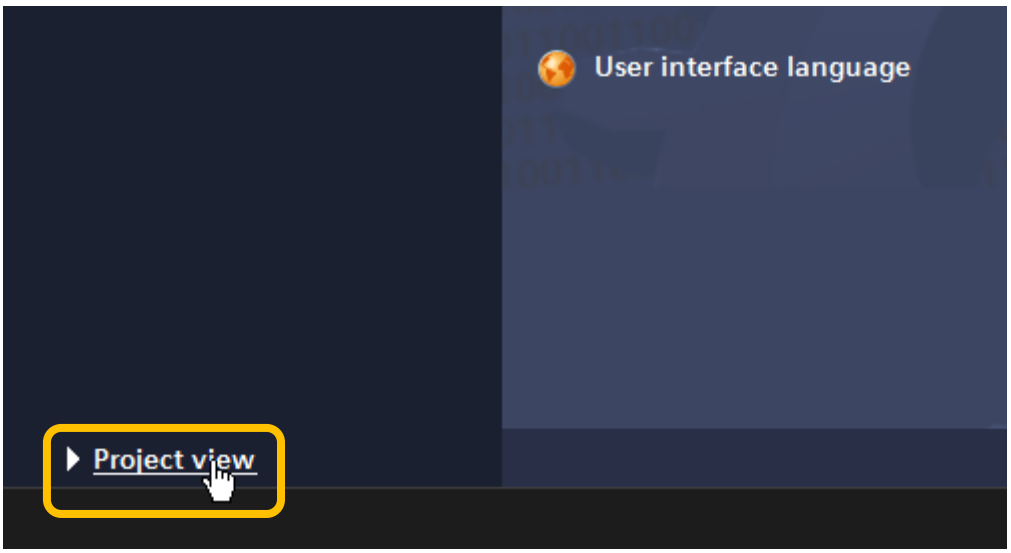
We start with a new function block (FB) and add the code to control one cylinder. PLC Sim advanced takes the role of a soft PLC and we download our program to an instance of it.

19 Simulation release in the TIA Portal

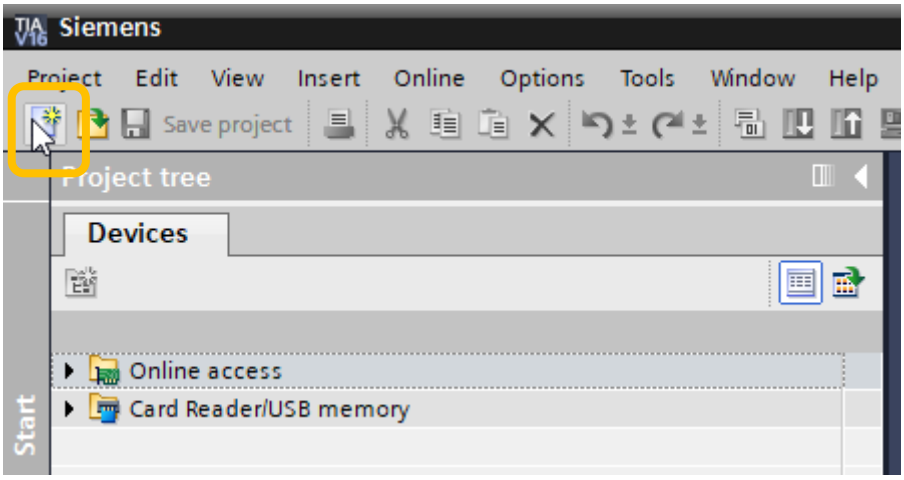
Video: 19TIA_new_enable_Simulation

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

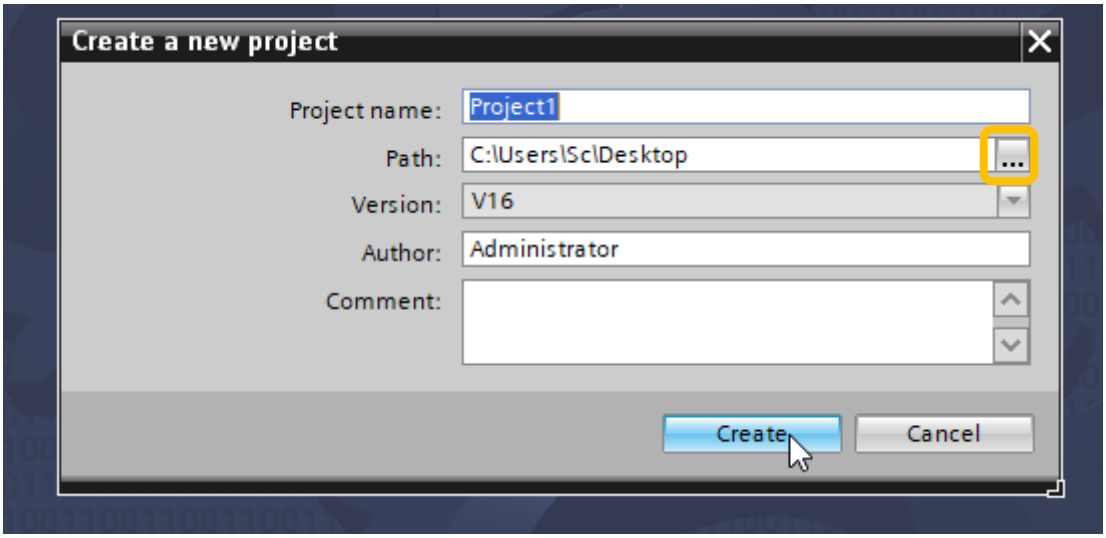


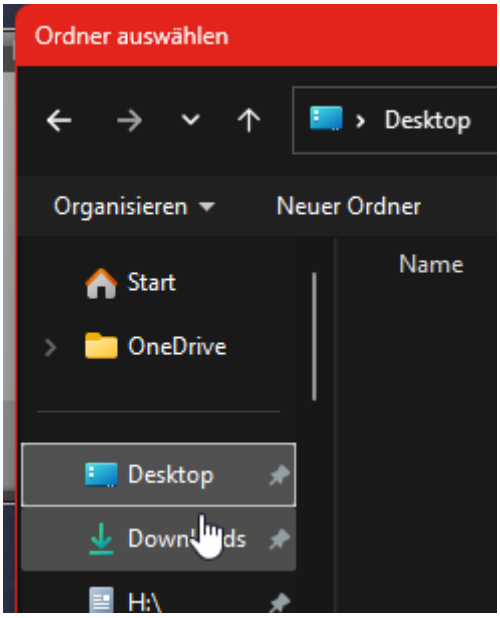


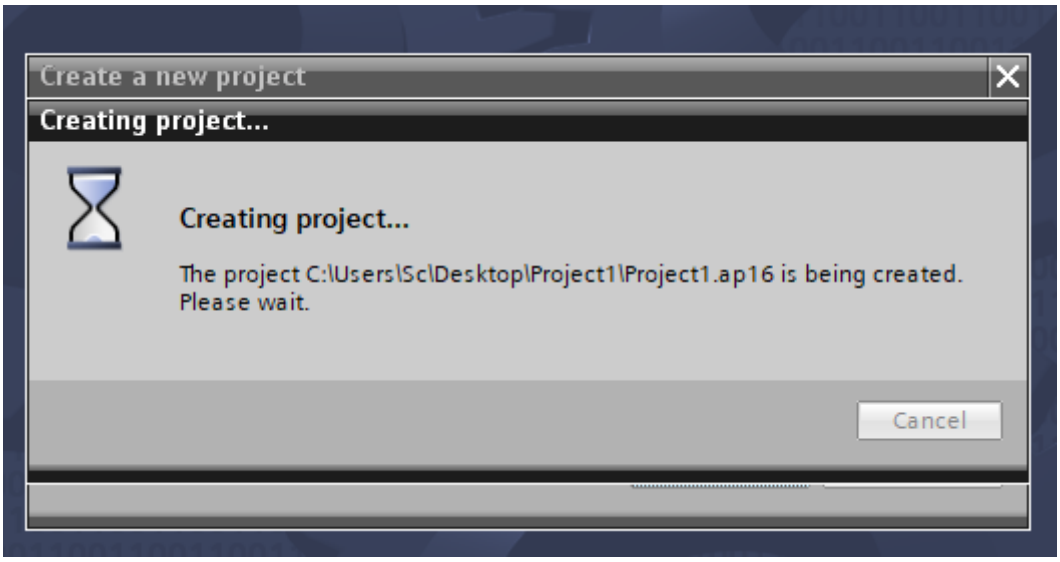
Create empty/new project



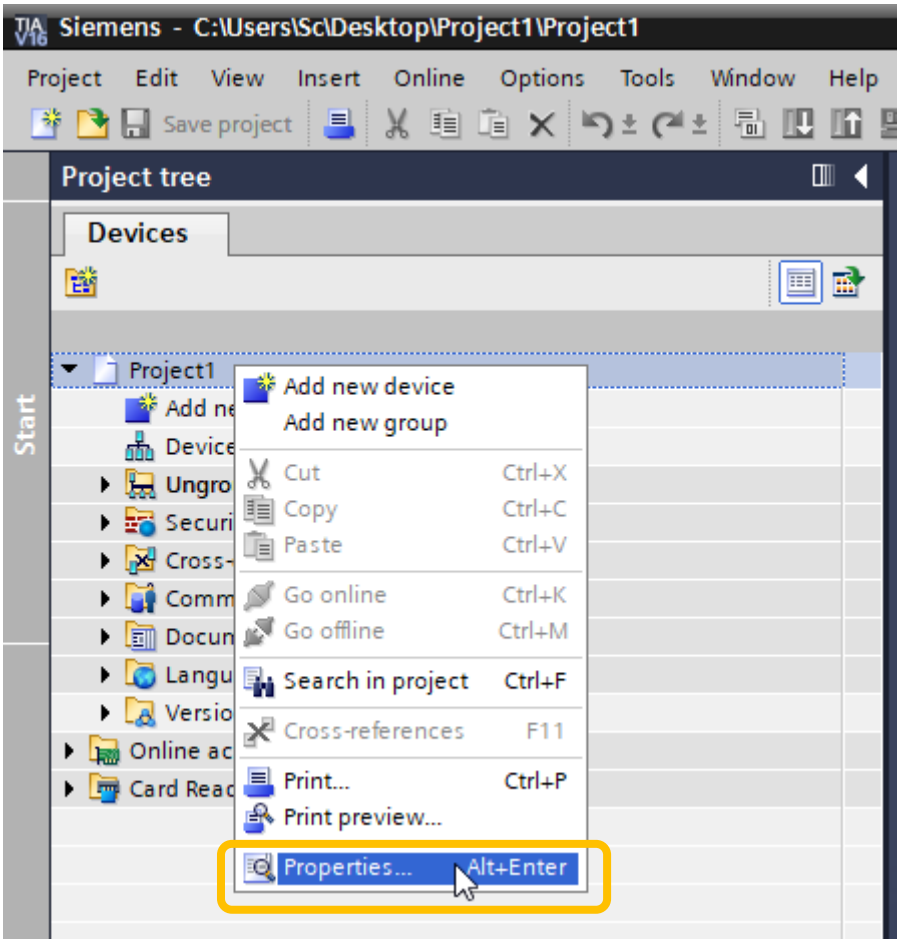
Set the path to the C:// drive (this is different in the video, please use your desktop)



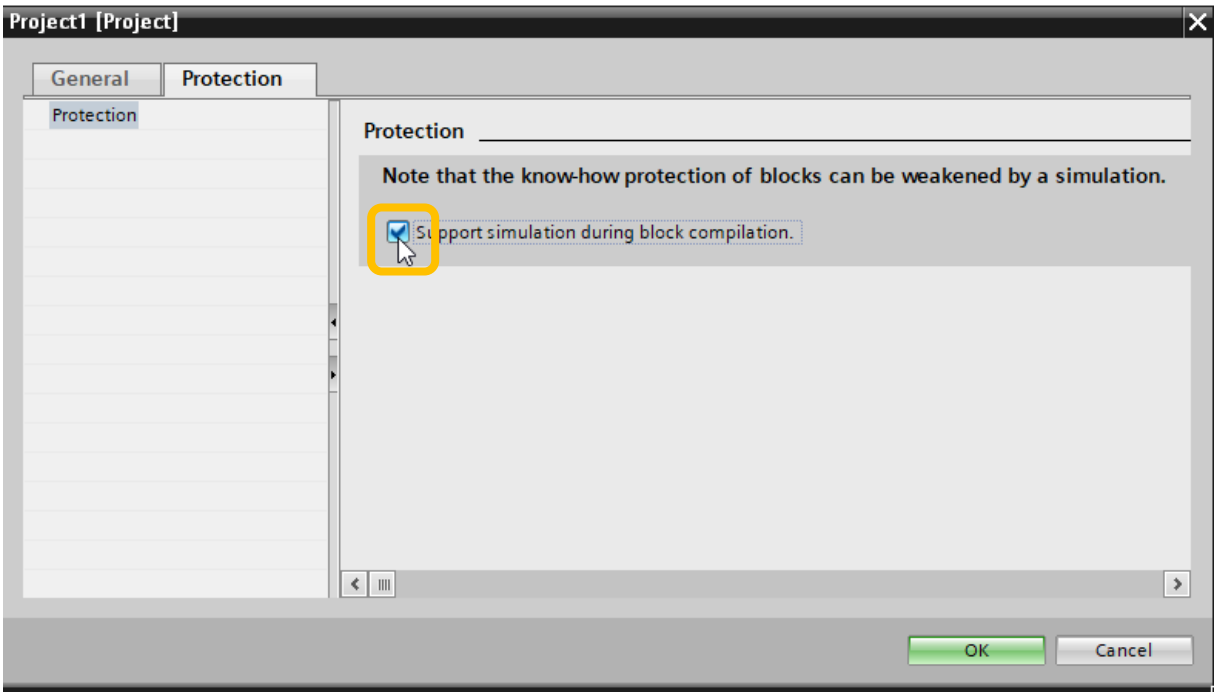

Ordner auswählen
Desktop
Organisieren
Neuer Ordner
Name
Start
OneDrive
Desktop
H:\


Create a new project
Creating project...
Creating project...
The project C:\Users\Scl\Desktop\Project1\Project1.ap16 is being created.
Please wait.
Cancel

Right-click on the project and then select Properties



Simulation must be activated

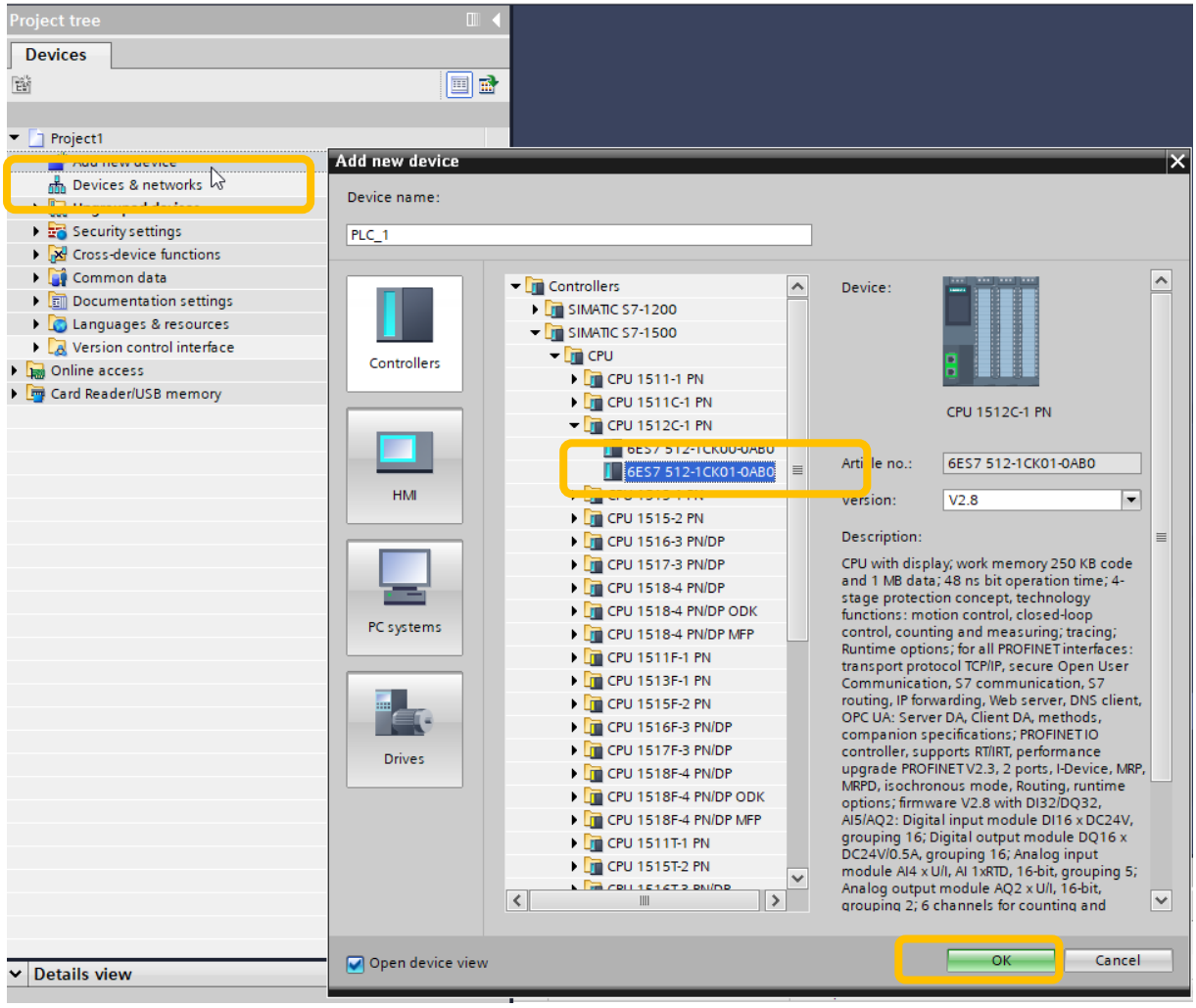


20 Create PLC hardware and integrate PLC variables

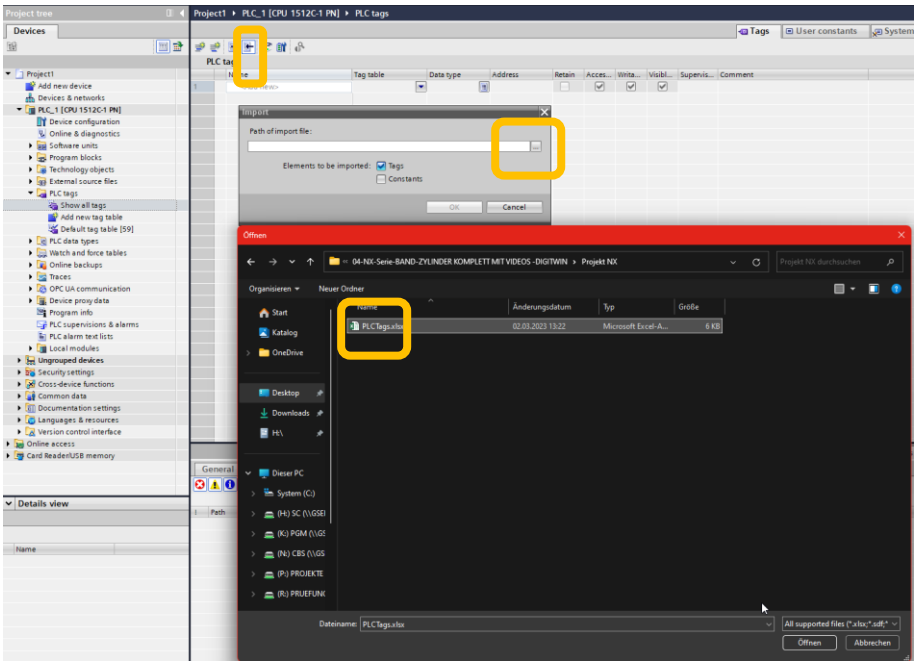
Video: 20TIA_new_PLC_Import_Variables

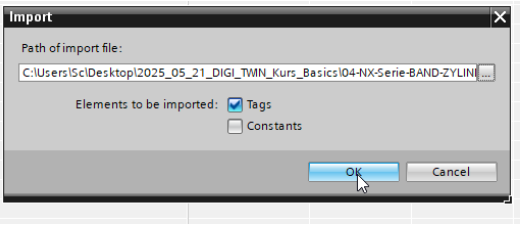
A hardware configuration for a PLC must be defined.

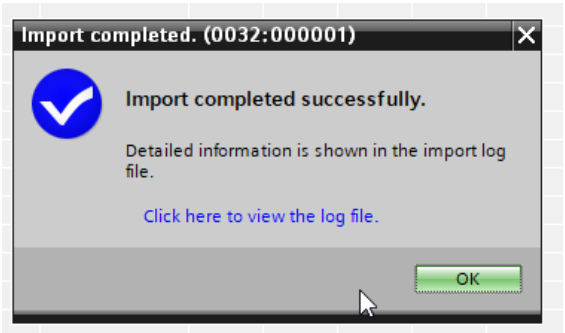
In this case, a 1500 PLC is used, as this corresponds to the reality on site.



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.



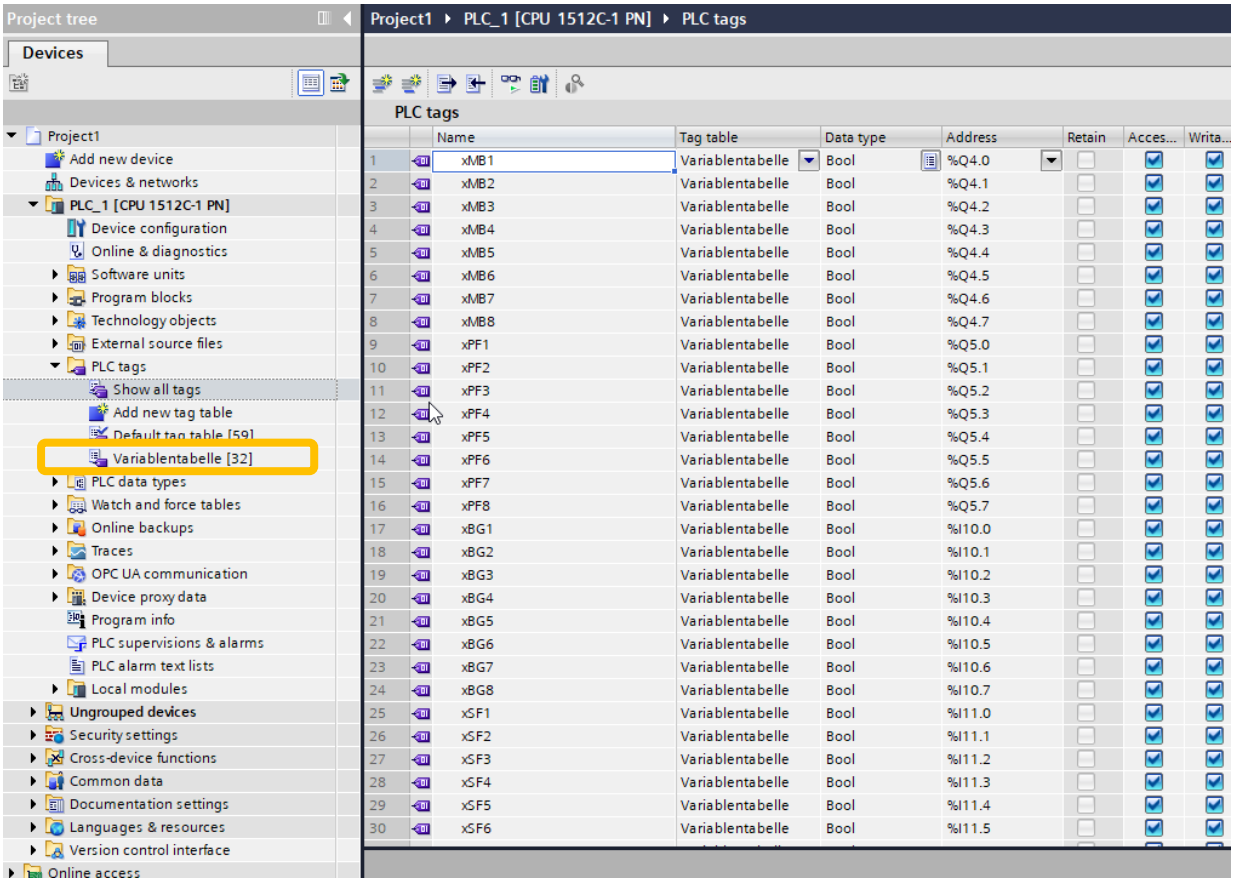




Overview of the imported PLC variables.

Module 4

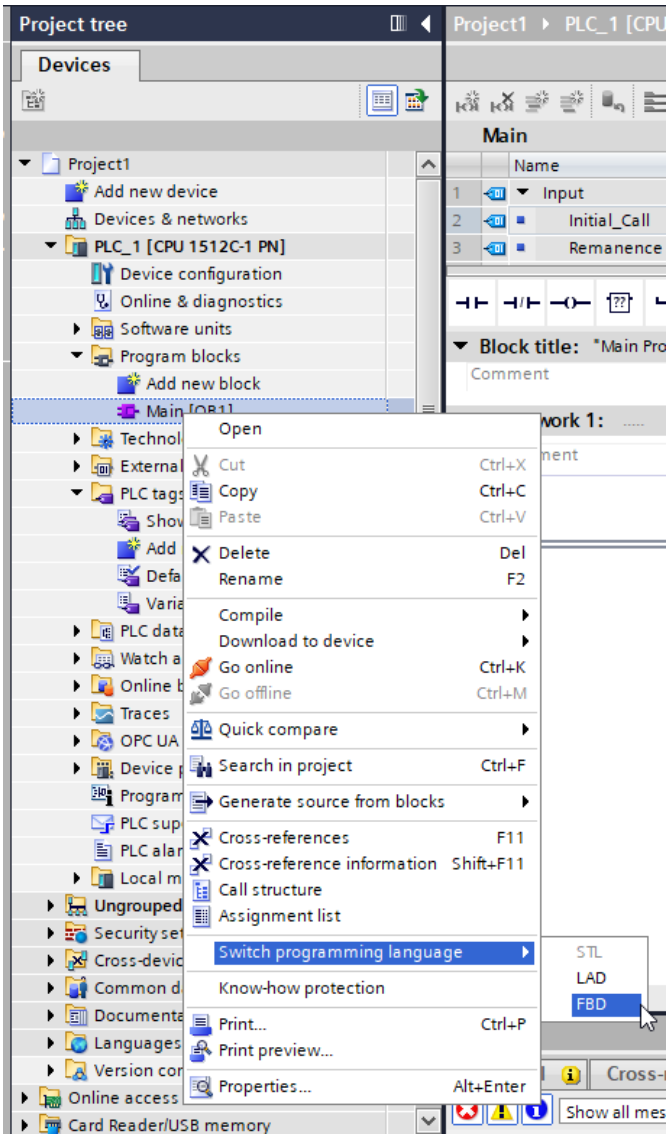
Documentation
Key Concept Product development



21 Creating a function block in the TIA Portal

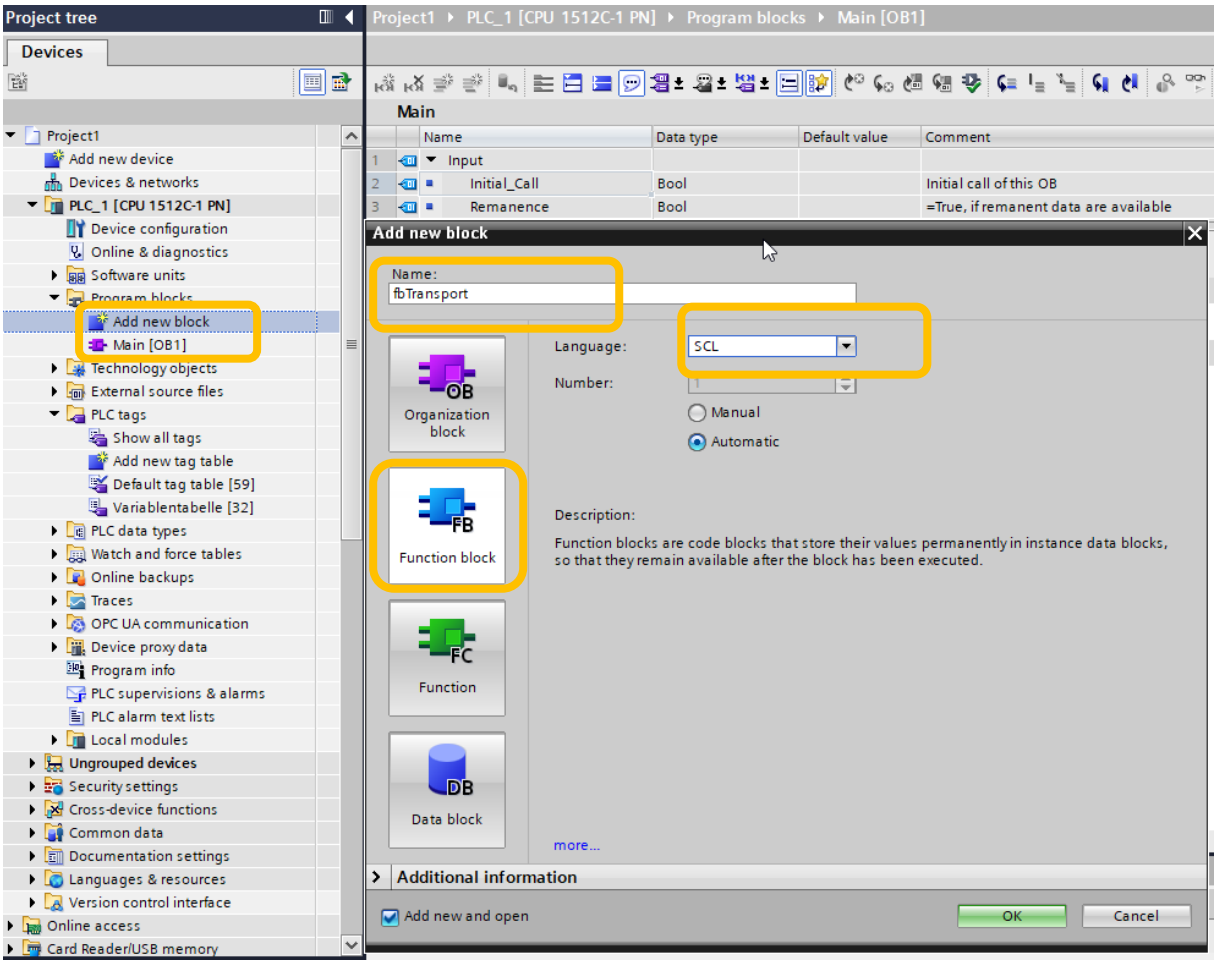
Video: 21TIA_New_FB

Changing the programming language in the MAIN OB1 block



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

Please select function block!

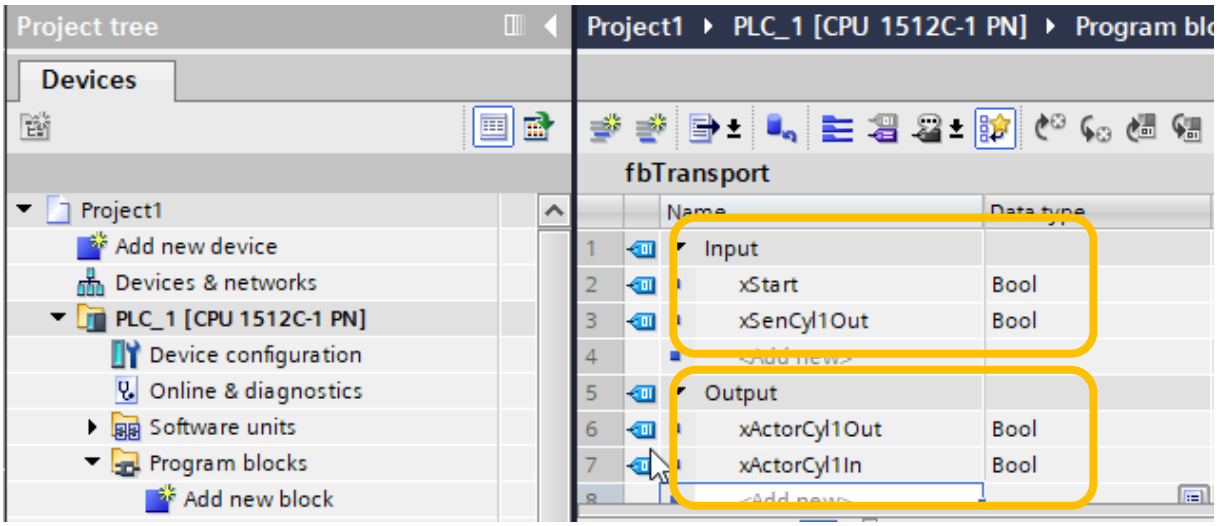


Please select Function block!

22 Program creation in the function block

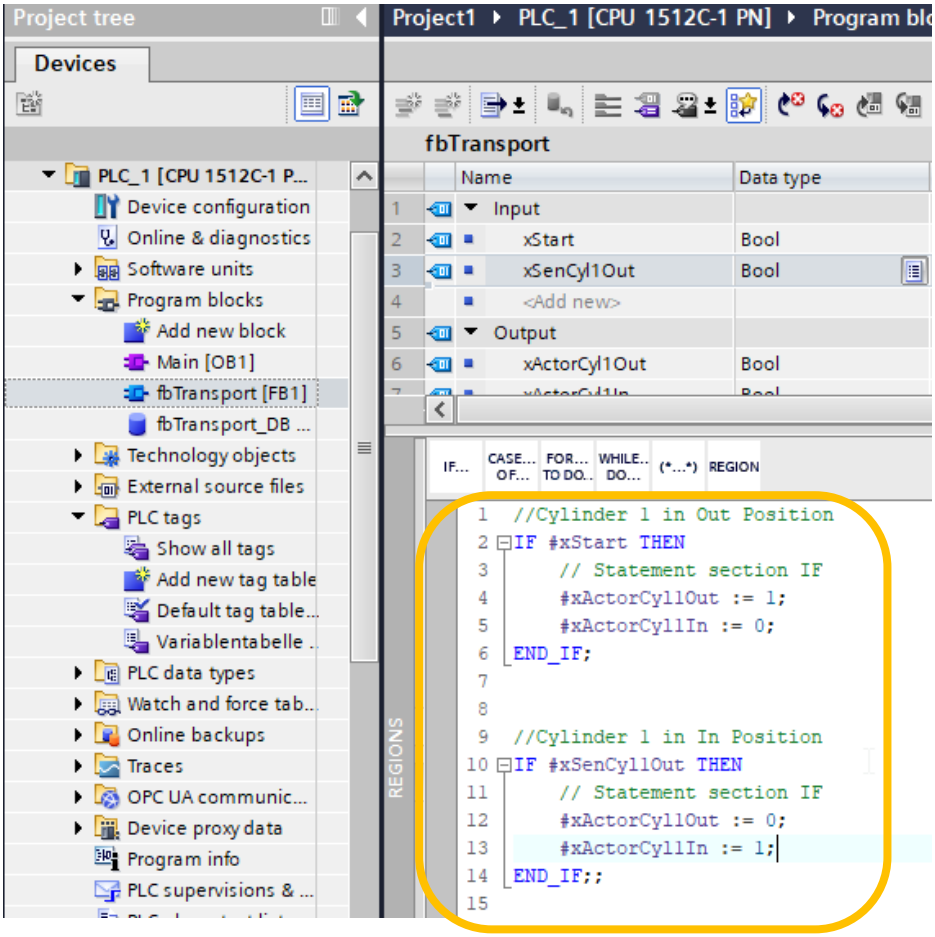
Video: 22TIA_FB_Prog

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

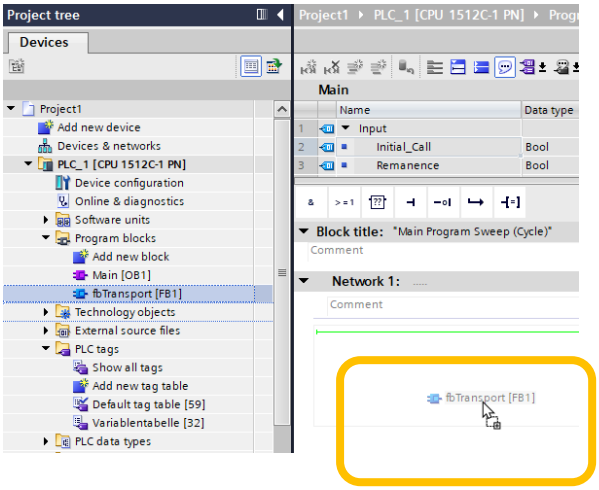
Documentation
Key Concept Product development

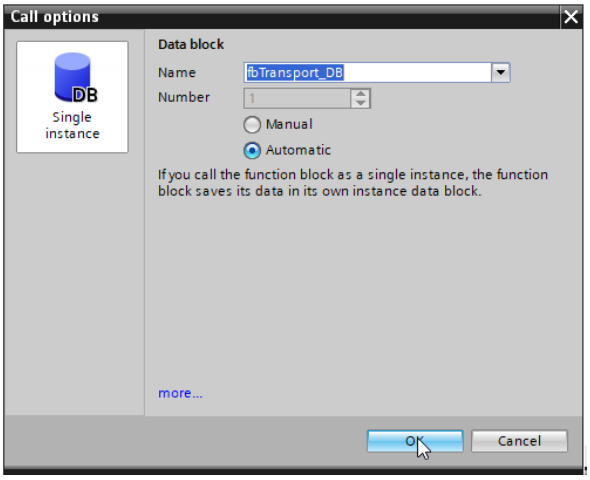


23 Call the FB in Main OB1 and save the program

Video: 23TIA_Prog_Save

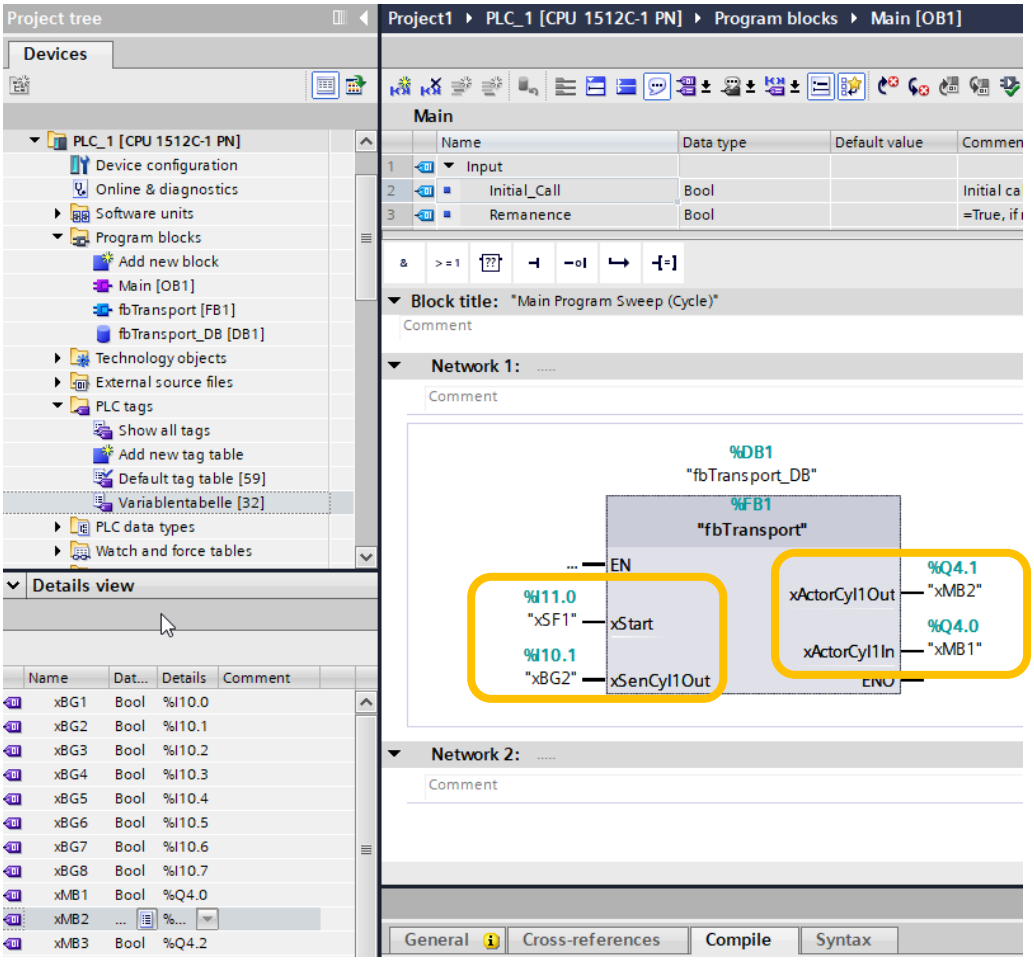
In MAIN OB1, the variables of the function block are linked to the PLC variables (PLC tags). First, the function block must be dragged and dropped into MAIN OB1.





Module 4

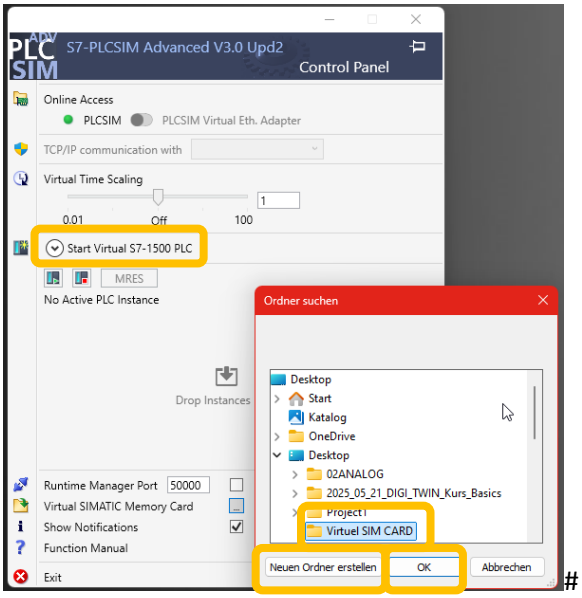
Documentation
Key Concept Product development



24 Creating a new simulation instance in PLCSim Advanced

Video: 24_Sim_New_Inst

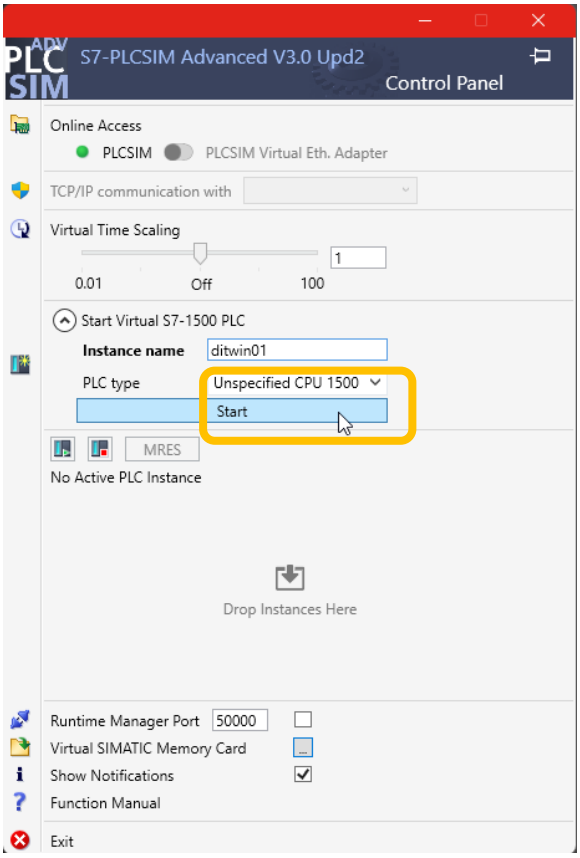
A virtual soft PLC can be created by starting the PLC Sim Advanced software. This controller, named **ditwin01**, is used to load the current PLC.

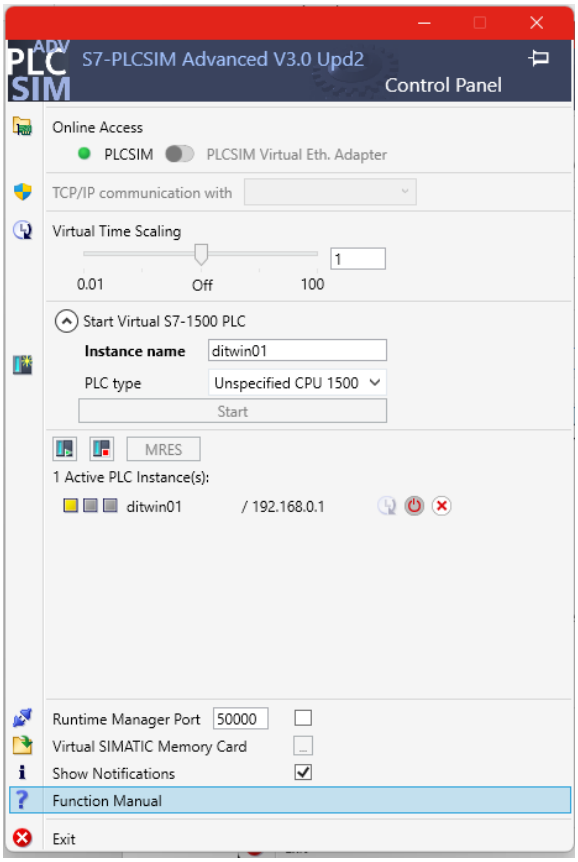


PLEASE NAME THE SOFT PLC – ditwin01!

Module 4

Documentation
Key Concept Product development



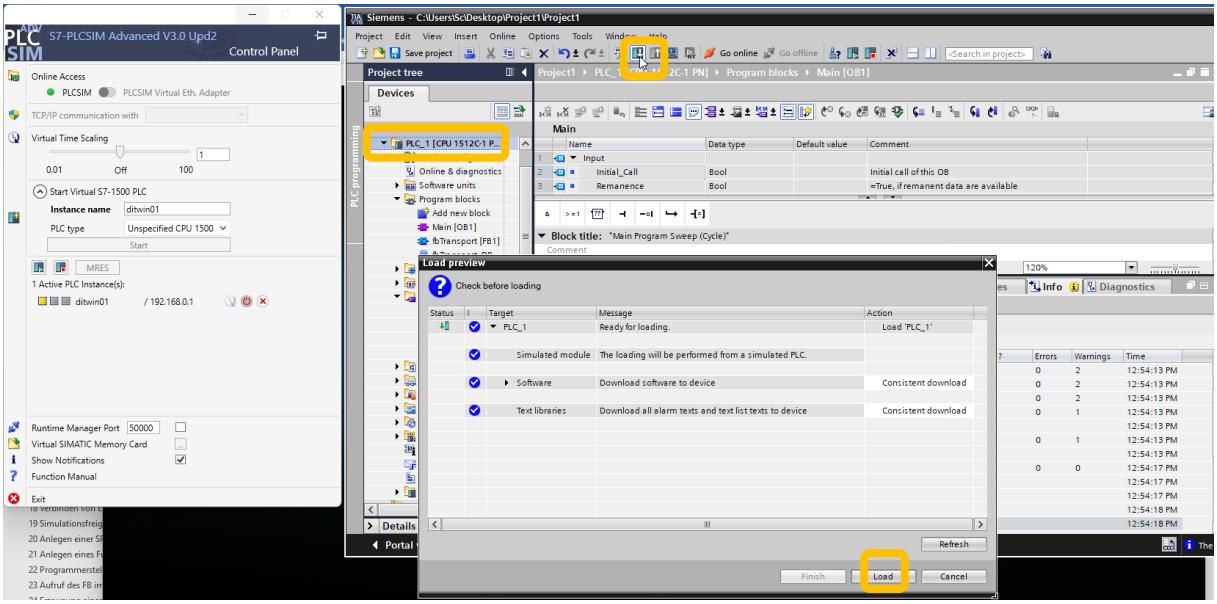


CARL-BENZ-SCHULE GAGGENAU
Mercantec
Co-funded by the European Union
2024-1-DE02-KA210-VET-000246677
www.digitwin4teach.eu

25 Load TIA program in PLC Sim

Video: 25_TIA_Sim_Transfer

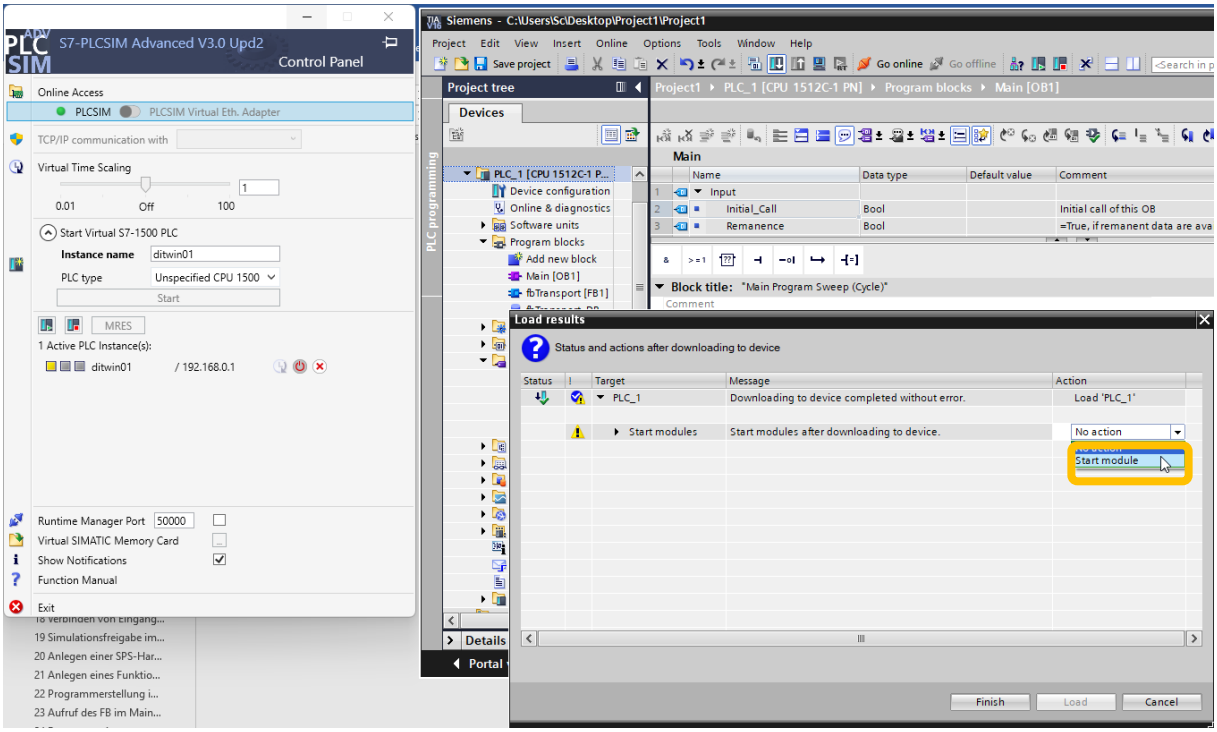
The current PLC program can now be loaded into PLC SIM Advanced.

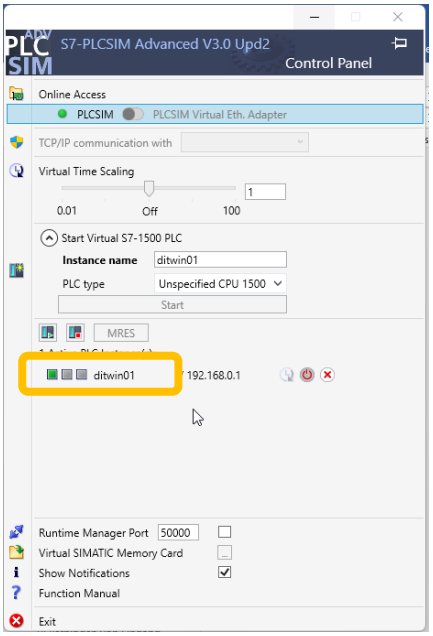


When the module is started, the PLC changes its status from Stop to Start and the LED turns green.

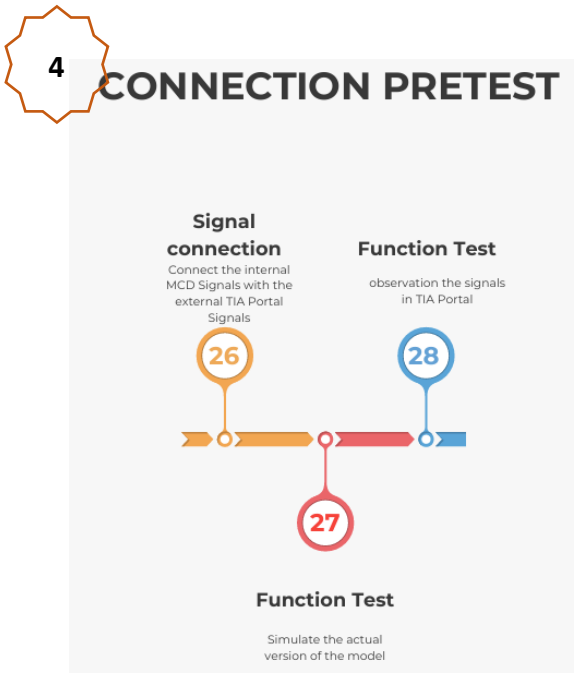
Module 4

Documentation
Key Concept Product development





Part 4

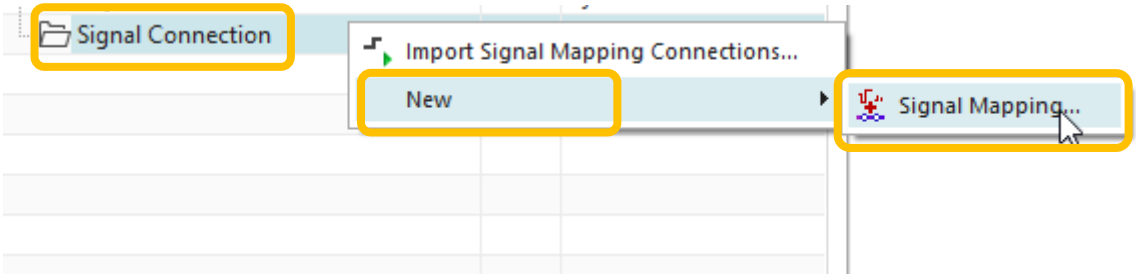


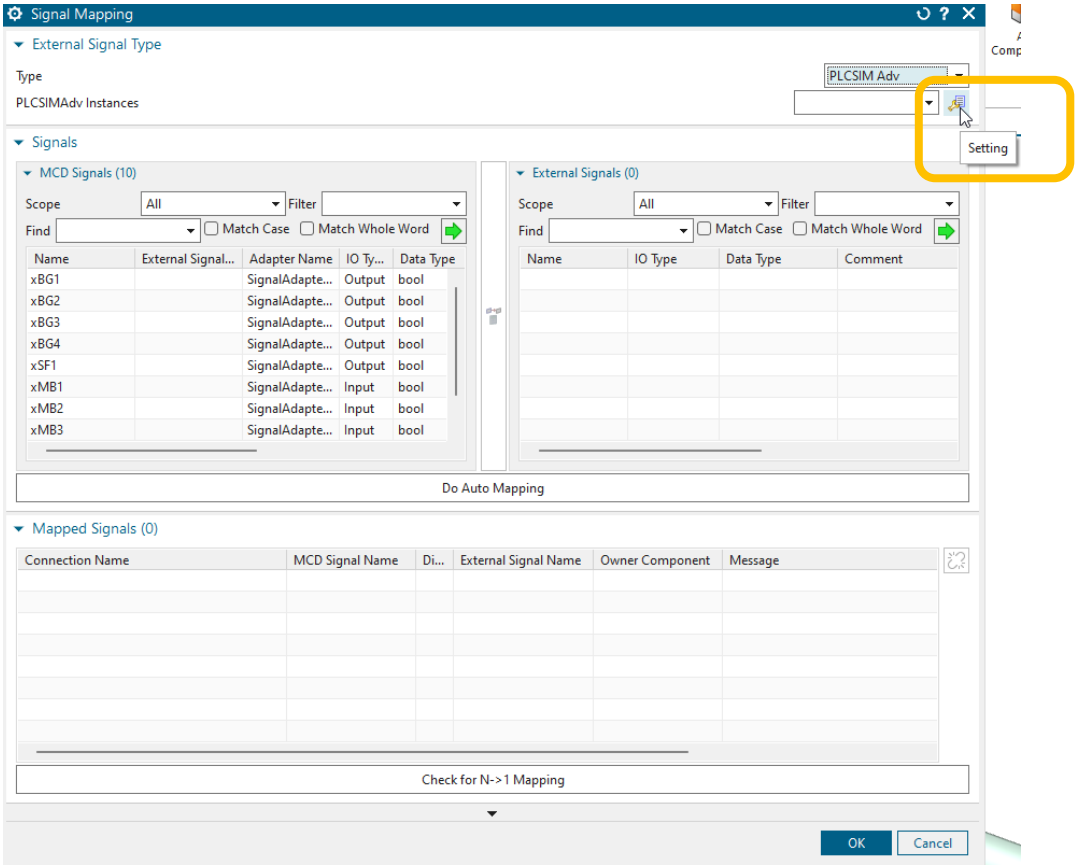
In this small section we are making some adjustments in NX and we run the test program.

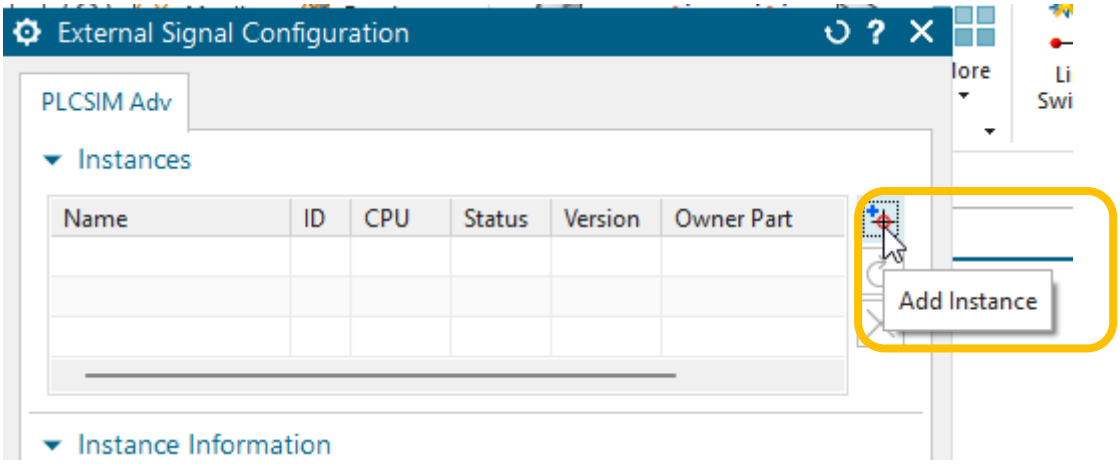
26 Creating signal connections in MCD

Video: 26_NX_New_Signal_Conn

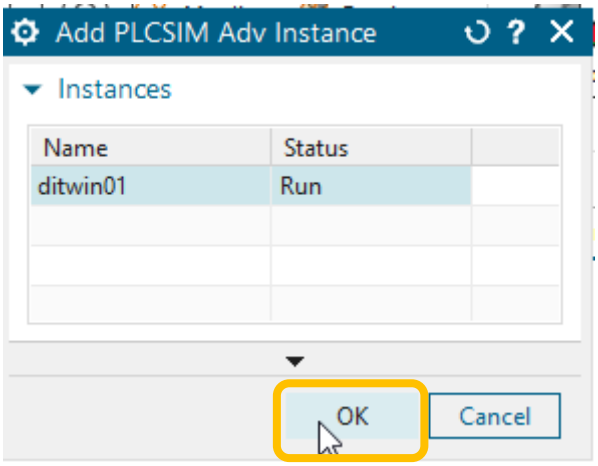
The simulation level is then ready for operation and the signal connections in Siemens NX MCD must be created.

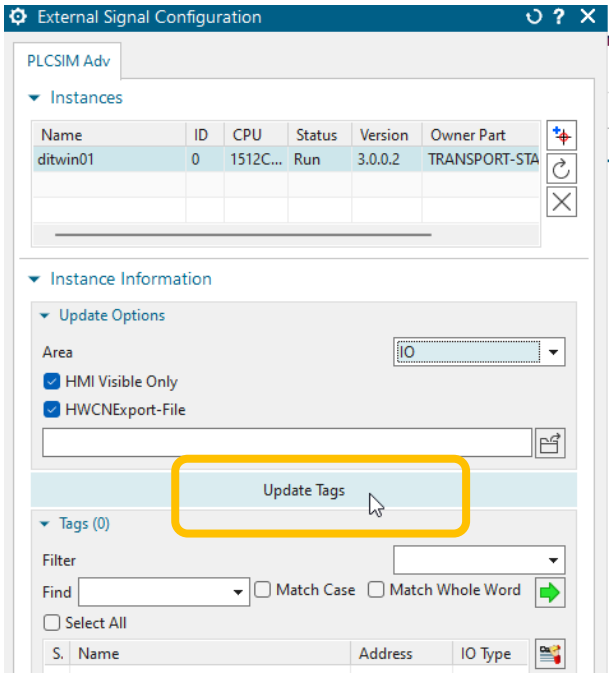


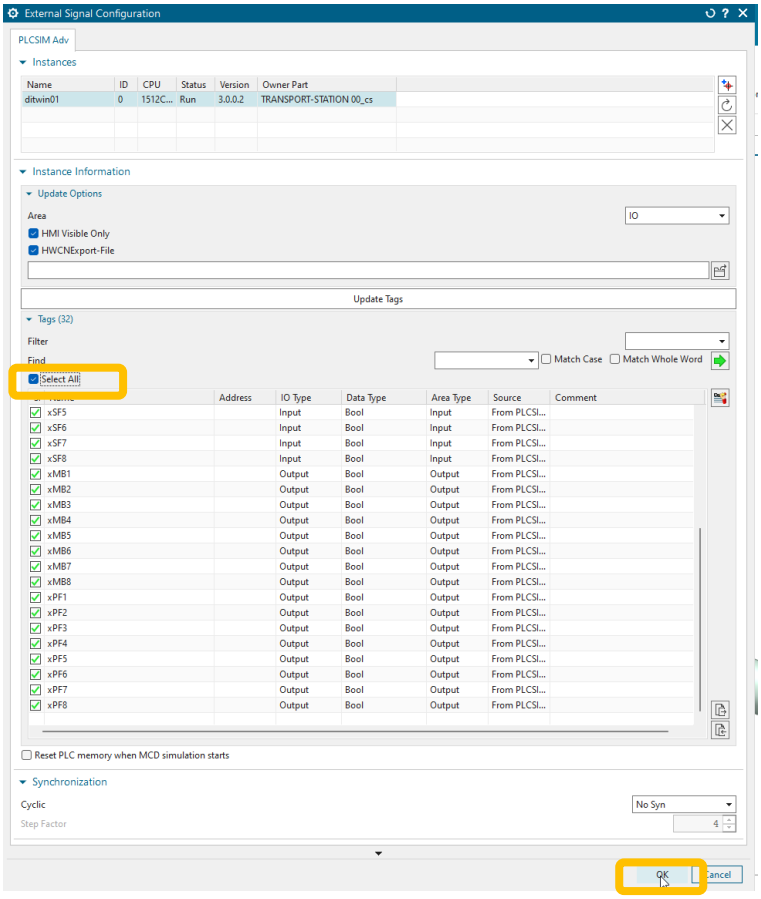

Signal Mapping
External Signal Type
Type
PLCSIM Adv
PLCSIMAdv Instances
Setting
Signals
MCD Signals (10)
External Signals (0)
Scope
All
Filter
Find
Match Case
Match Whole Word
Name
External Signal...
Adapter Name
IO Ty...
Data Type
IO Type
Comment
xBG1
xBG2
xBG3
xBG4
xSF1
xMB1
xMB2
xMB3
SignalAdapte...
Output
Input
bool
Do Auto Mapping
Mapped Signals (0)
Connection Name
MCD Signal Name
Di...
External Signal Name
Owner Component
Message
Check for N->1 Mapping
OK
Cancel


External Signal Configuration
PLCSIM Adv
Instances
Name
ID
CPU
Status
Version
Owner Part
Add Instance
Instance Information

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



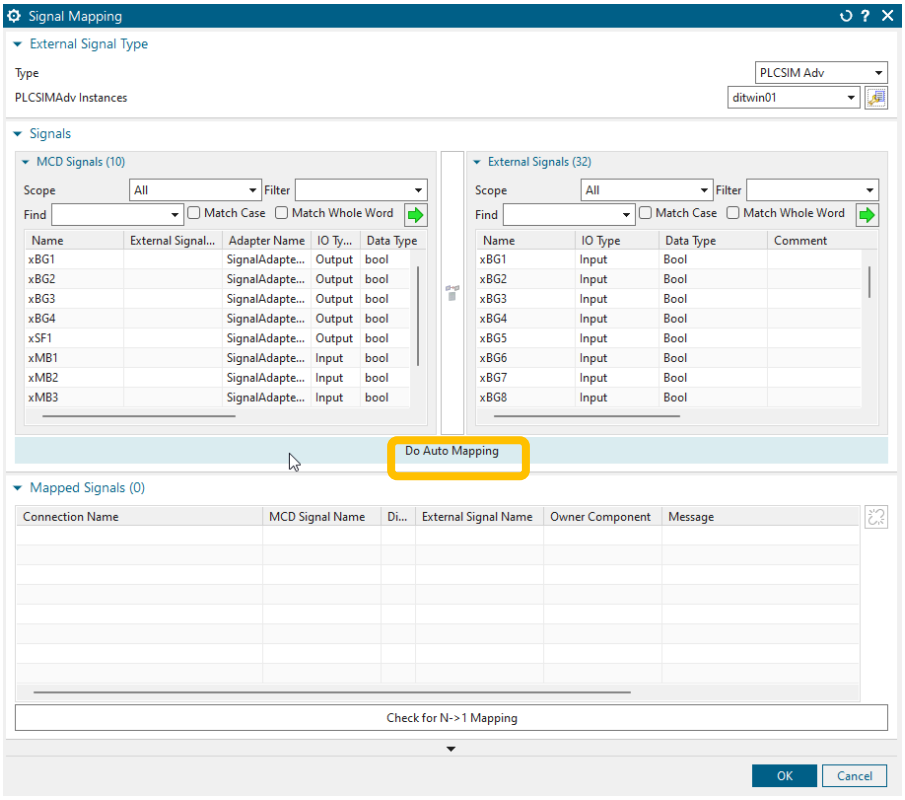




Select all variables and then select automatic assignment. The automatic assignment can be performed successfully thanks to the identical naming.

Module 4

Documentation
Key Concept Product development



27 Temporary storage and testing of the software status in NX

Video: 27_NX_Save_Test

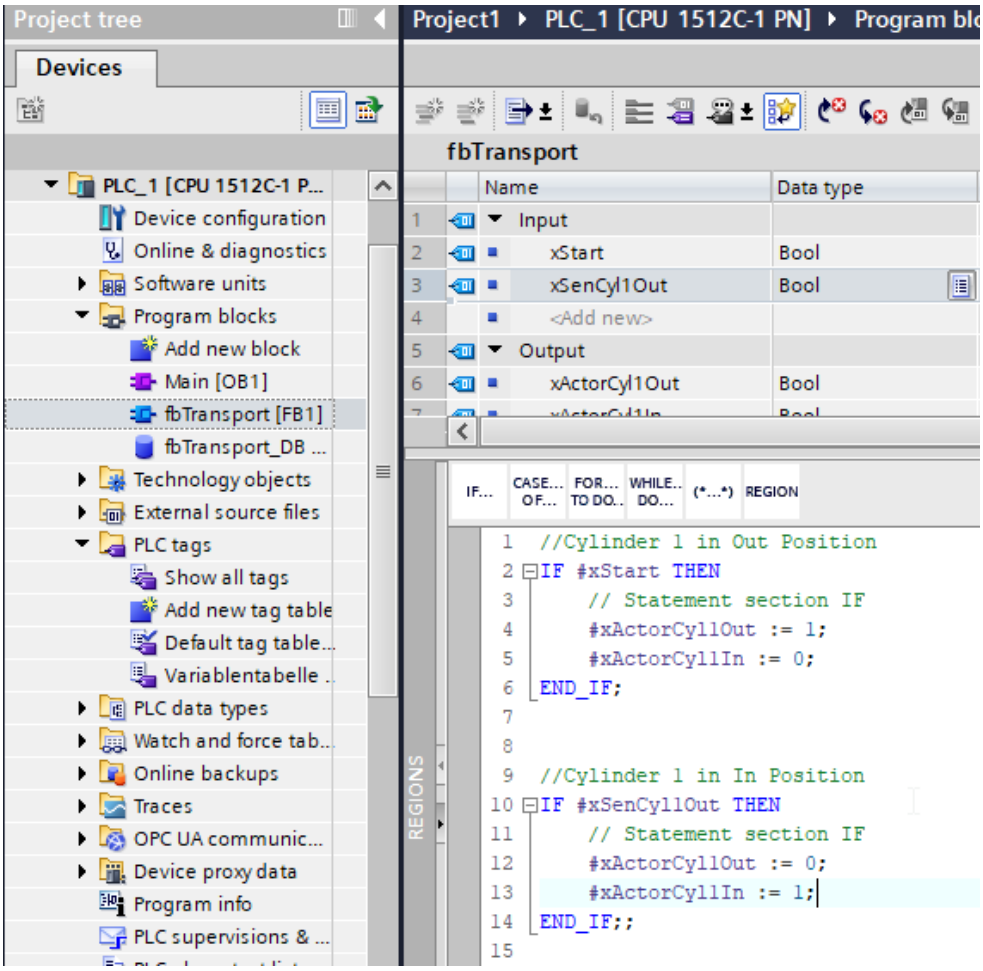
Video: 27_TIA_Observe

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

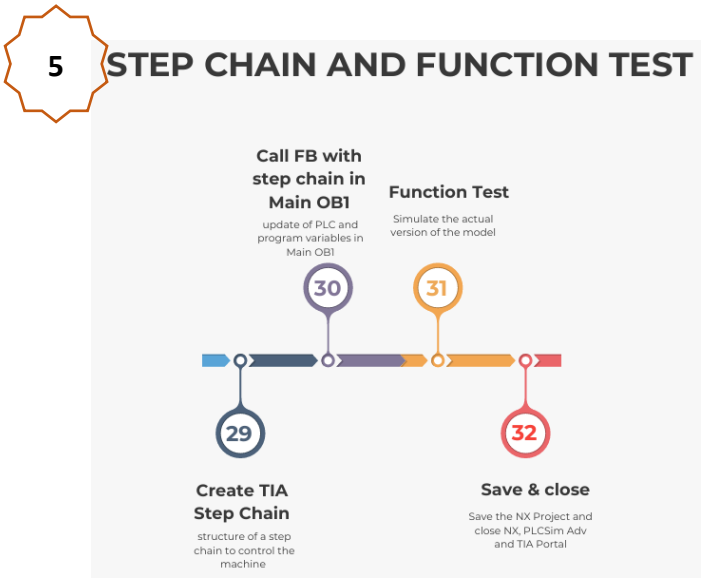
28 Testing the software status in the TIA Portal

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.



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.

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...	Writa...	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.

```
CASE #iStep OF
    0:
        IF #xStart THEN // Statement section case 1
            #iStep := #iStep + 1;
        END_IF      ;
    1:
        IF #xSenCyl1Out THEN // Statement section case 1
            #iStep := #iStep + 1;
        END_IF;
    2:
        IF #xSenCyl1In THEN // Statement section case 1
            #iStep := #iStep + 1;
        END_IF;
    3:
        IF #xSenCyl2Out THEN // Statement section case 1
            #iStep := #iStep + 1;
        END_IF;
    4:
        IF #xSenCyl2In THEN // Statement section case 1
            #iStep := 0;
        END_IF;

END_CASE;
```

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.

fbTransport

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

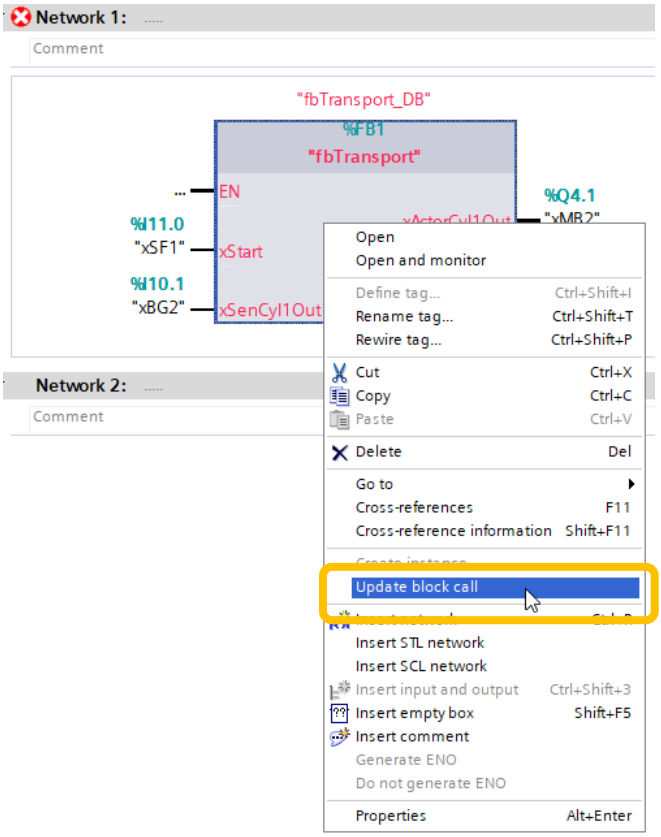
IF... CASE... OF... FOR... TO DO.. WHILE.. DO... (*...*) REGION

```
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
```

REGIONS

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

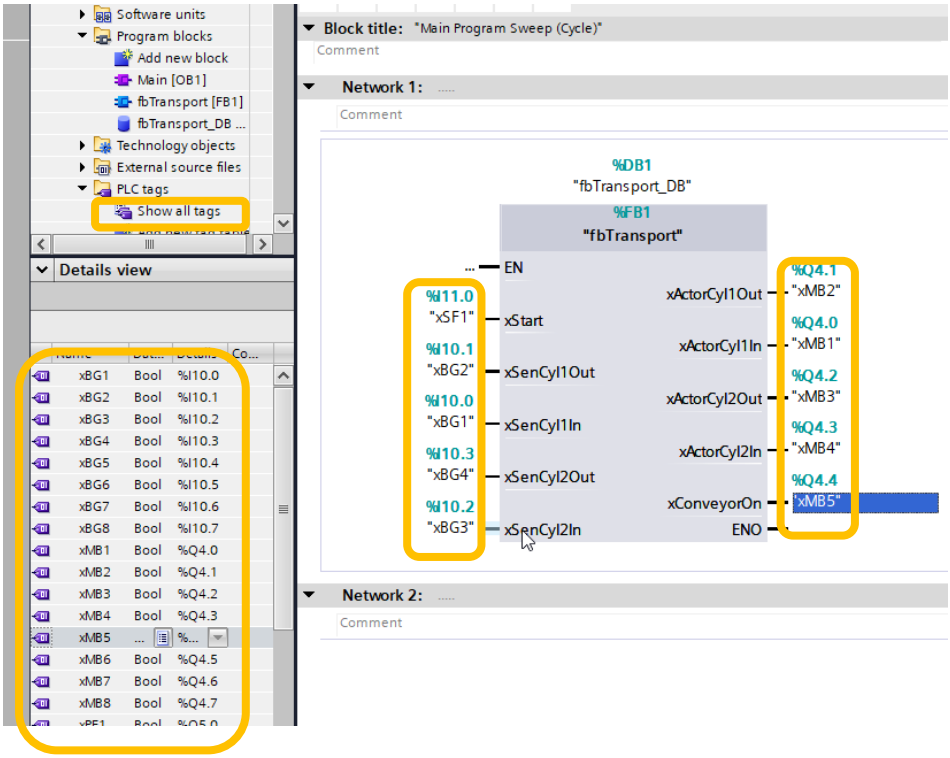
To do this, the content in MAIN OB1 must first be updated so that the PLC variables can then be reassigned.



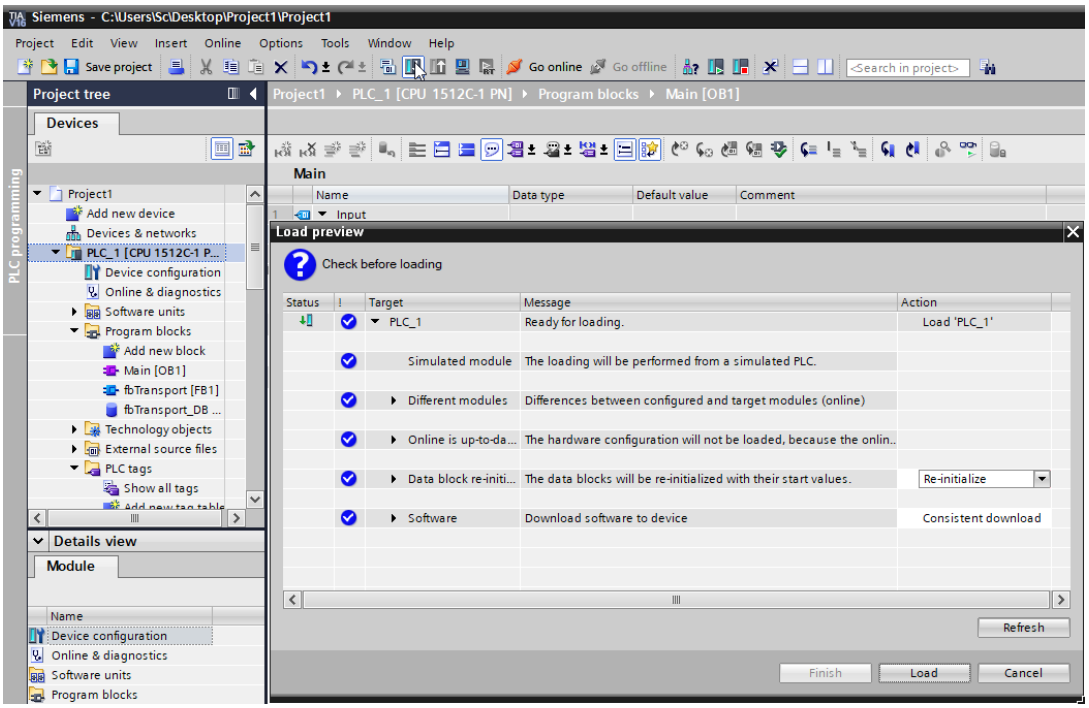
Simply click on Show all tags once and assign them to MAIN OB1 using drag & drop.

Module 4

Documentation
Key Concept Product development



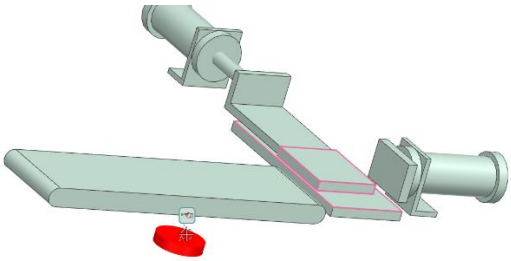
The current software version can then be reloaded from the TIA Portal into the soft PLC in PLCSIM Advanced.

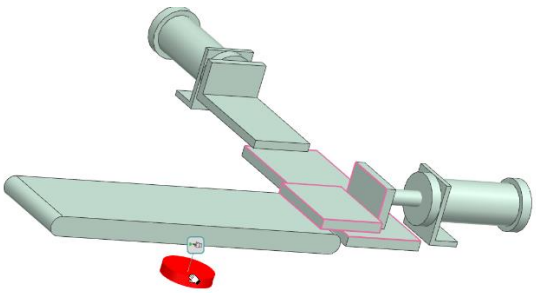


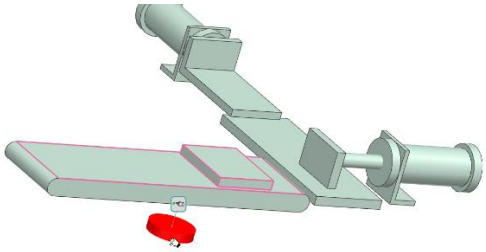
31 Testing the software version with NX

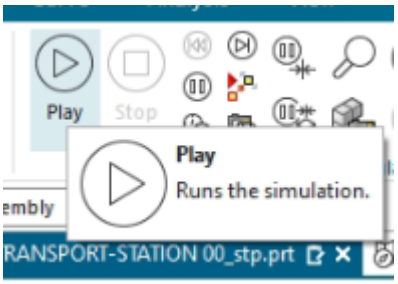
Video: 31_NX_Prog_Chain_Test

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.





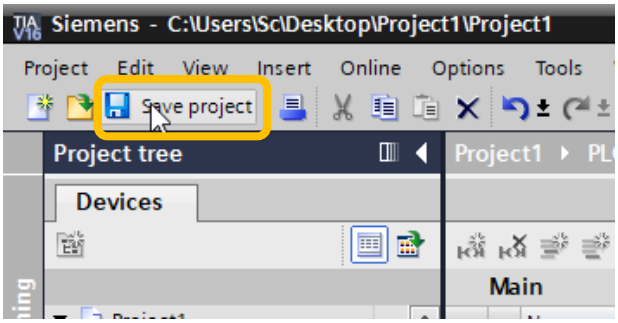




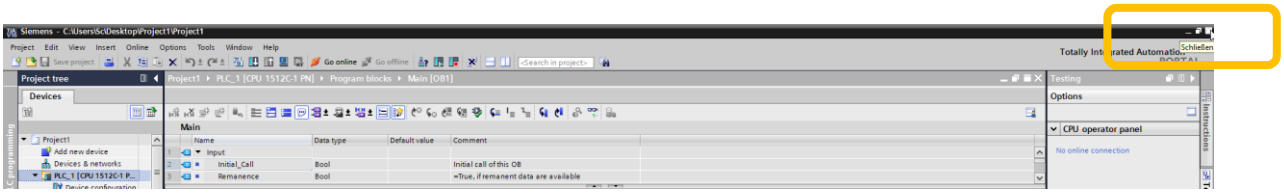
32 Save status and close entire project

Video: 32_Save_Close_All

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



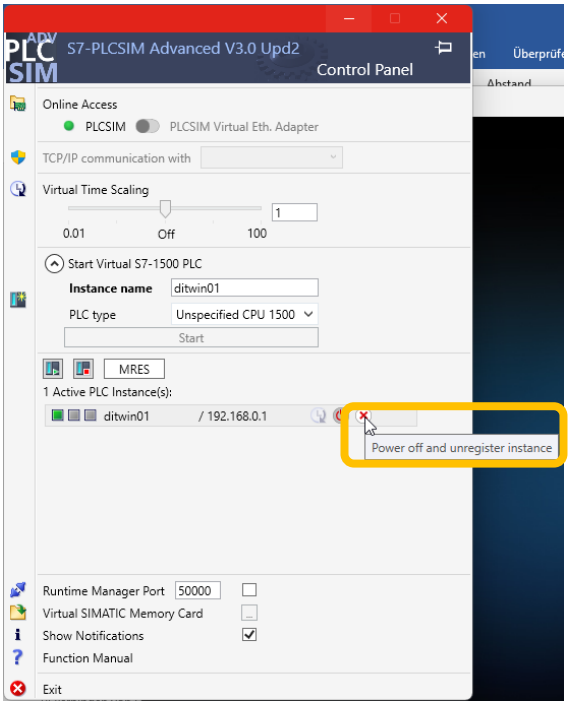
The TIA Portal project can then be closed.

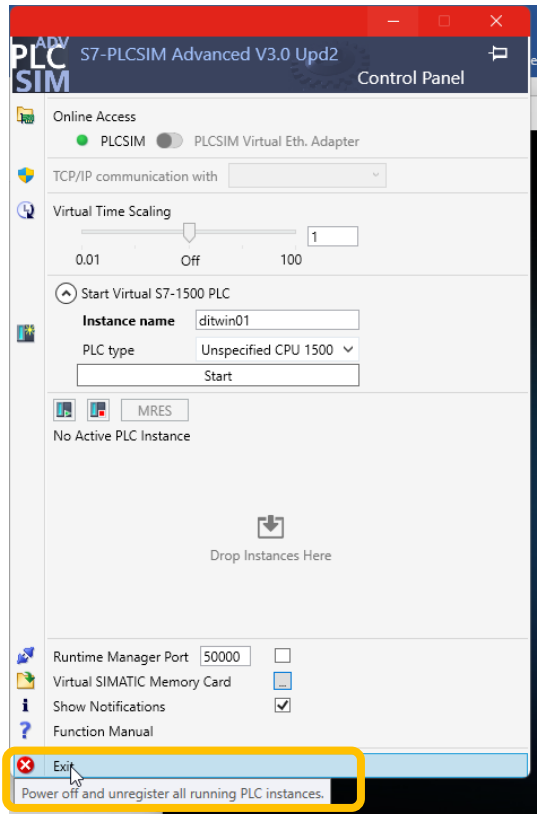


To close PLC SIM Advanced properly, it is best to proceed as shown.

Module 4

Documentation
Key Concept Product development





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.

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.

39 Adjustment for the stop signal in the signal connection

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

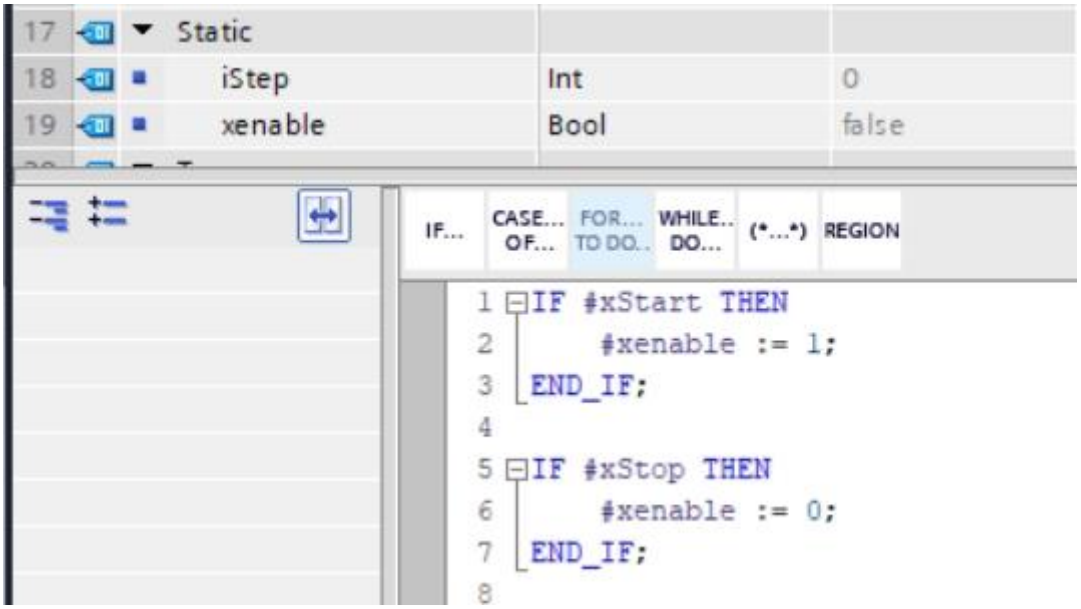
For communication, the new stop button labeled SF2 must also be assigned to the TIA program in the Signal Connection. This assignment is carried out within the Signal Connection.

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.



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.

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.

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.

Disclaimer