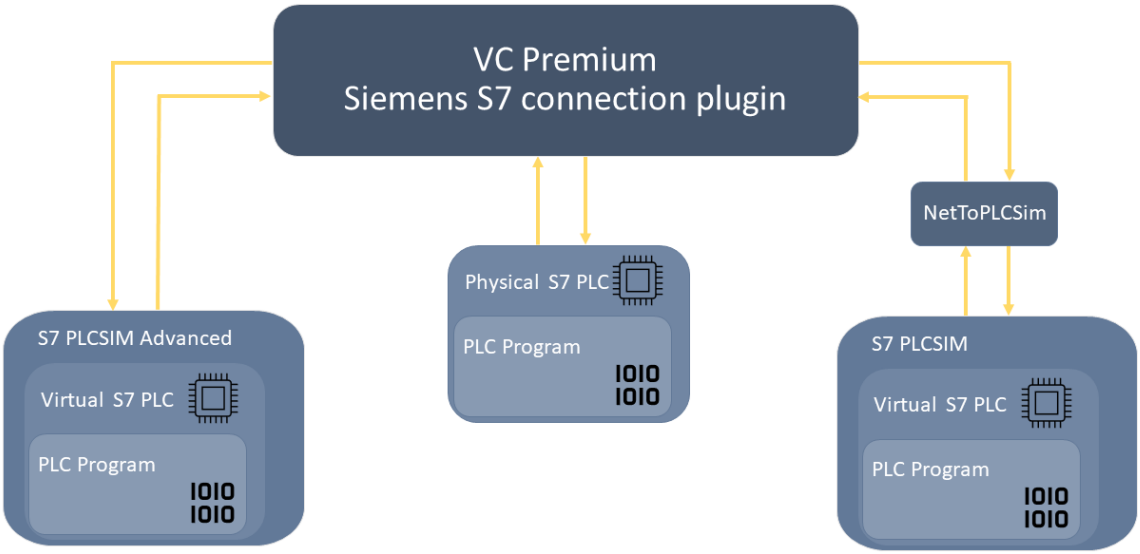
# Siemens S7 connection plugin tutorial

Visual Components 4.2.2 Premium | Version: May 26, 2020




The Siemens S7 connection plugin is a new feature introduced to Visual Components in version 4.2.2. The plugin is available in the Premium version of the software, and it can be used to connect Visual Components simulation to Siemens S7-series programmable logic controllers (PLC).

This tutorial requires a basic level understanding of Visual Components and TIA Portal. During this tutorial, you will learn how to establish a connection between S7-series PLC and Visual Components. This involves setting up a project and downloading it from TIA Portal to a PLC, and using Visual Components to simulate a production environment to test the implemented PLC logic.

**Support**
support@visualcomponents.com
**Visual Components Forum**
forum.visualcomponents.com

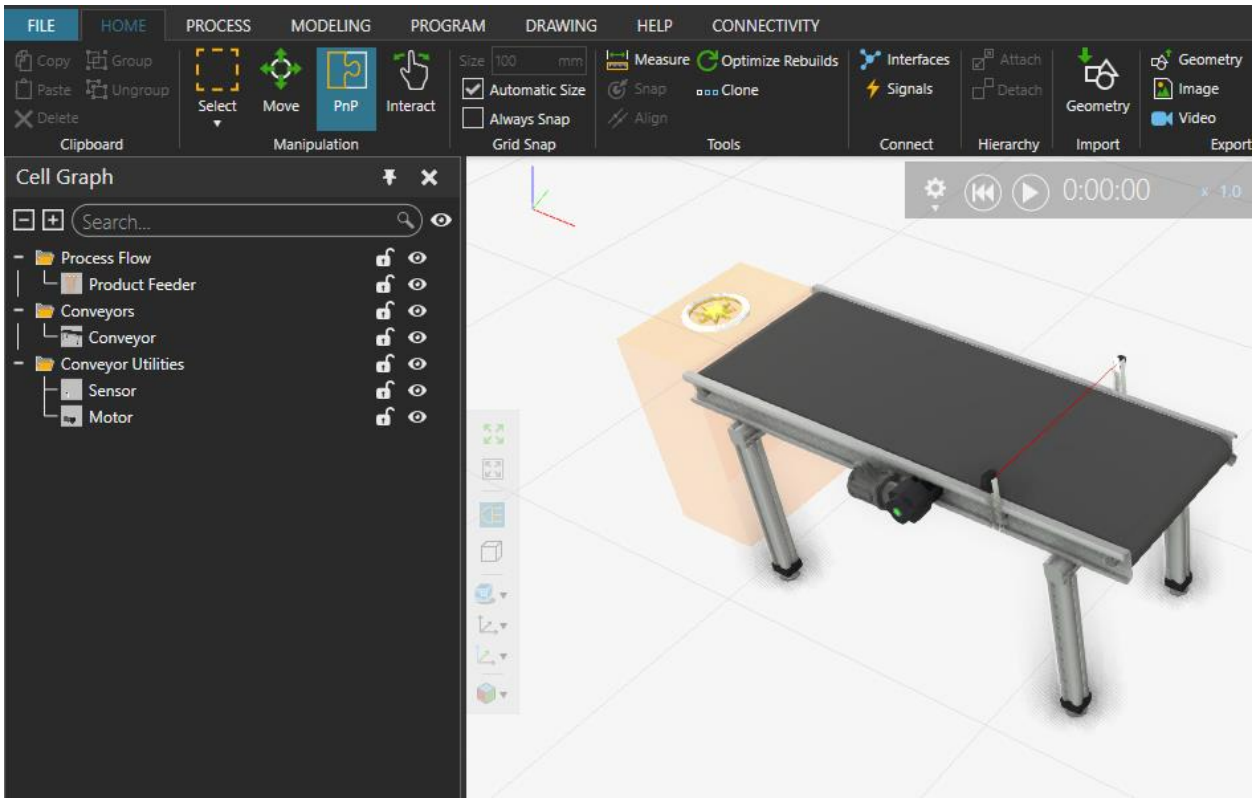
# Contents

Getting started ..... 3
Setting up the properties in TIA Portal ..... 4
Creating your PLC program block ..... 7
Exporting the tag table ..... 8
PLCSIM and NetToPLCSim ..... 9
Connecting Visual Components to the PLC ..... 10
Pairing the variables ..... 12
Review ..... 15
Appendix ..... 16

# Getting started

The Siemens S7 connection plugin can be configured in the connectivity tab of Visual Components Premium 4.2.2 or newer software. If the connectivity tab is not visible in your application, you must first enable it by navigating to File, Options, Add On to enable the Connectivity view. After enabling connectivity, you must save the changed settings by pressing OK at the bottom, and then you must restart the application.

This tutorial uses a simple layout setup with a feeder, conveyor, sensor, and a motor. You can construct a similar layout by looking up the components from the corresponding categories in the eCatalog, and then connecting the components with the Plug and Play (PnP) tool. Alternatively, you can adapt the training and try the same approach with your custom logic and layout.



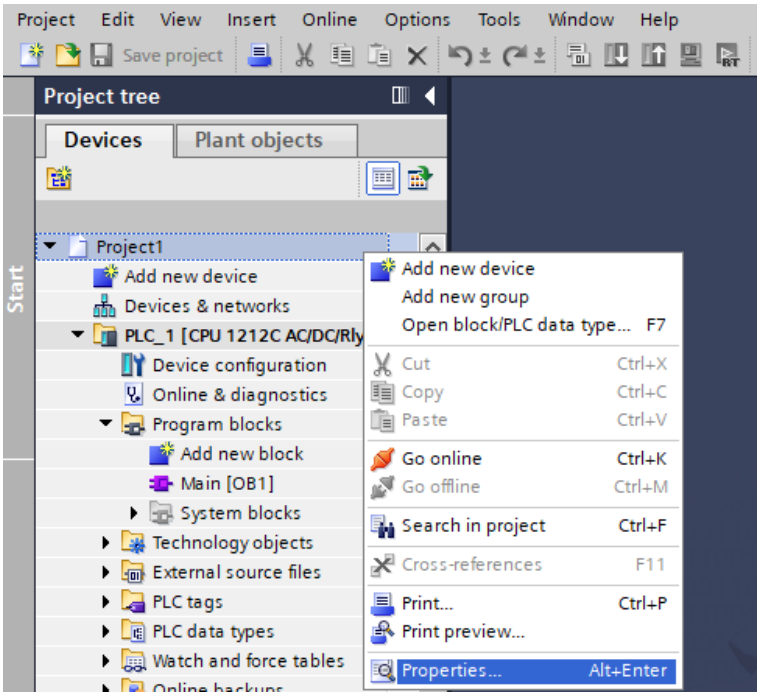
# Setting up the properties in TIA Portal

This tutorial is done using TIA Portal V16 and a virtual S7-1200 series PLC that is simulated with S7-PLCSIM V16. A similar procedure will also work with other S7-series PLCs, such as 300, 400, and 1500, that are either physical or simulated with PLCSIM or PLCSIM Advanced. Note that no OPC functionalities are required for the PLC to connect to Visual Components when using the S7 plugin.

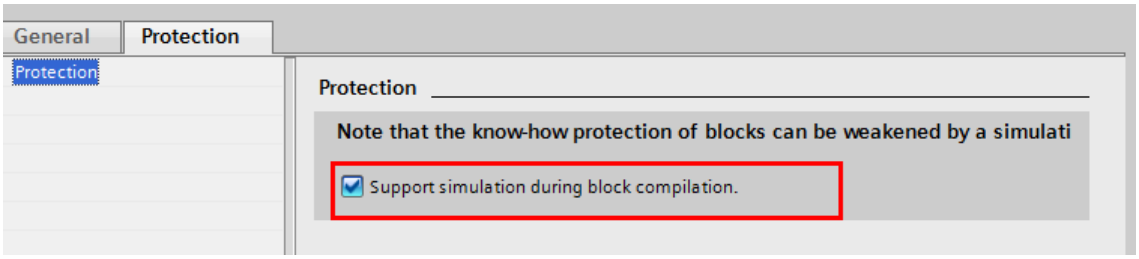
The basic steps for configuring the PLC project for simulation are:

- Support simulation during block compilation (Optional)
- Configure the PLC Ethernet interface
- Disable physical input updating (Optional)
- Enable full access and PUT/GET communication
- Configure NetToPLCSim (Only for PLCSIM)

To get started in TIA Portal, you can either create a new project or use an existing project. If you are using simulated PLC, you must go to the project properties to enable the simulation support. You can enter the project properties by right-clicking on the project.

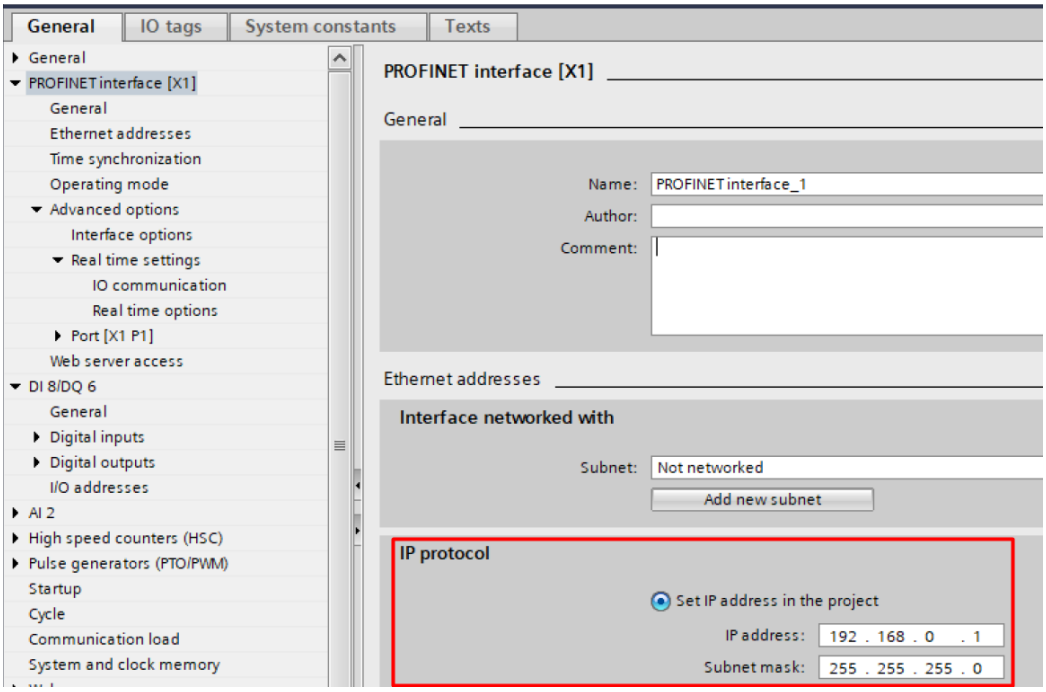


Select the Protection tab in the project properties. You must make sure that the setting 'Support simulation during block compilation' is enabled.

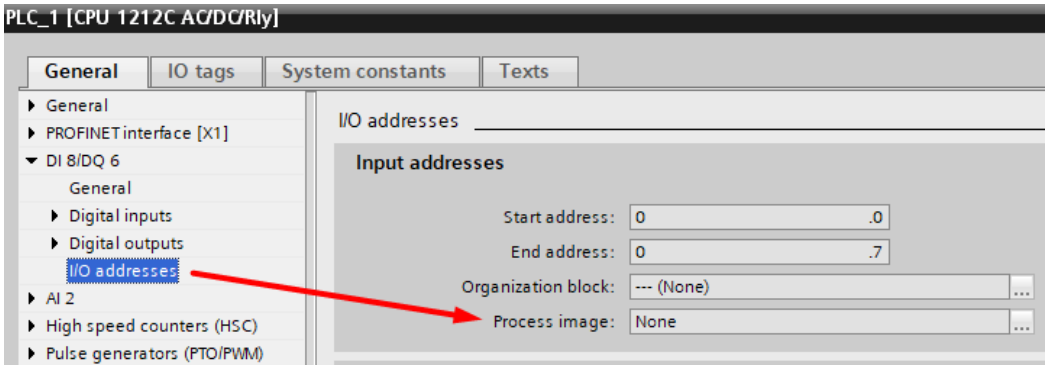



You must add one or more devices to your project by pressing the Add new device button. This tutorial uses a 1212C AC/DC/Rly CPU as the only device. After adding the device, you can enter the device properties by right-clicking on the device and selecting Properties. After you have finished the virtual commissioning, you should update the properties back to the project-specific values.

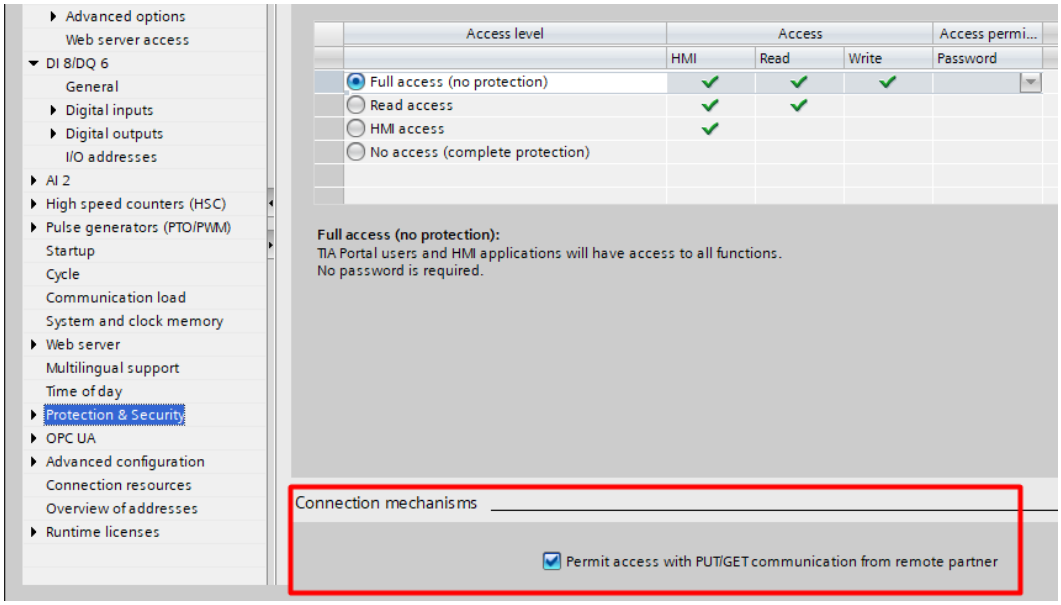
The first property to set is usually the device **IP address** found in the PROFINET Interface settings. When using a simulated PLC, it is often a good practice to set the IP address to a private IP address, such as 192.168.0.1.



The next property to check is Process Image, which is found in the I/O addresses settings. The **Process image** property can be set to **None** so that the physical IOs of the PLC do not overwrite the values written by the simulation.

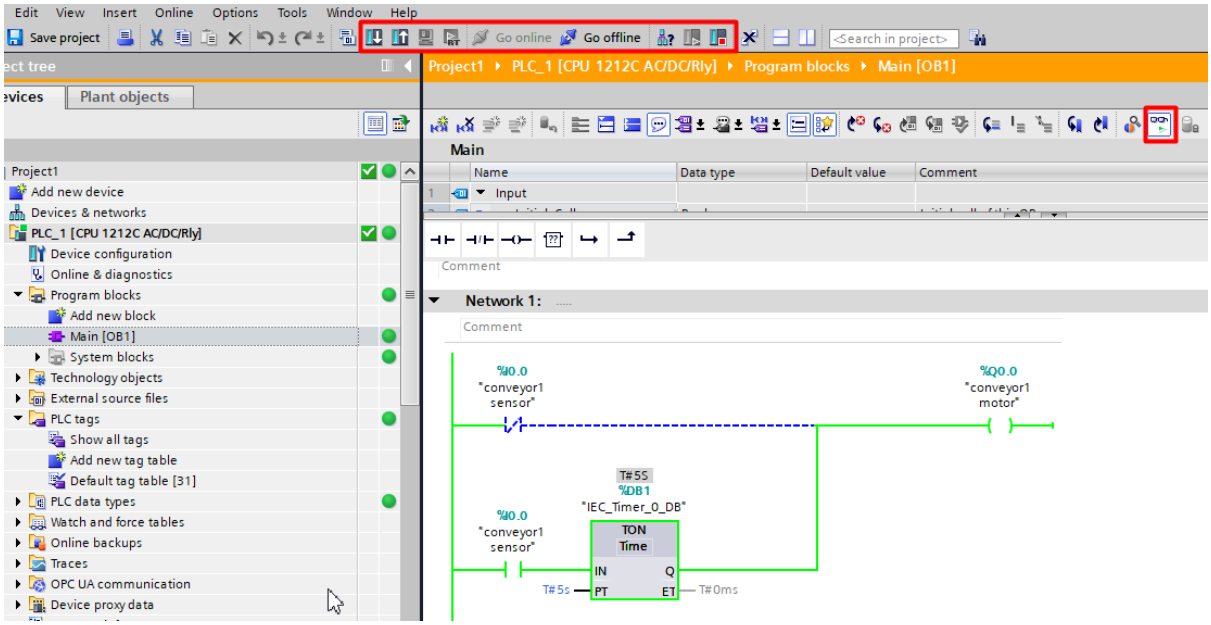



The final properties that are relevant for connecting the device to the simulation are found from the Protection & Security settings. To enable the simulation to read and write to the PLC, you must first enable **Full access**, and secondly, enable **Permit access with PUT/GET communication from remote partner**. Note that this will leave your PLC vulnerable, and should only be done for testing in a secure network.



# Creating your PLC program block

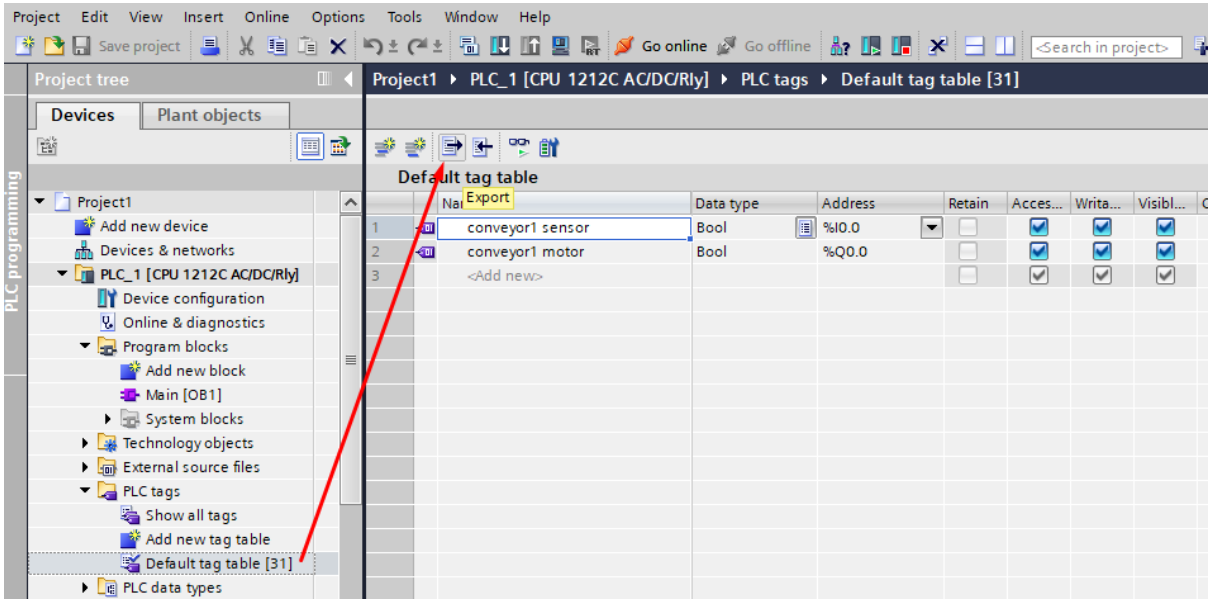
After setting the device properties, you can start creating your PLC program logic. The control logic for this tutorial has been made with ladder logic, but you can use any of the IEC 61131 programming languages.



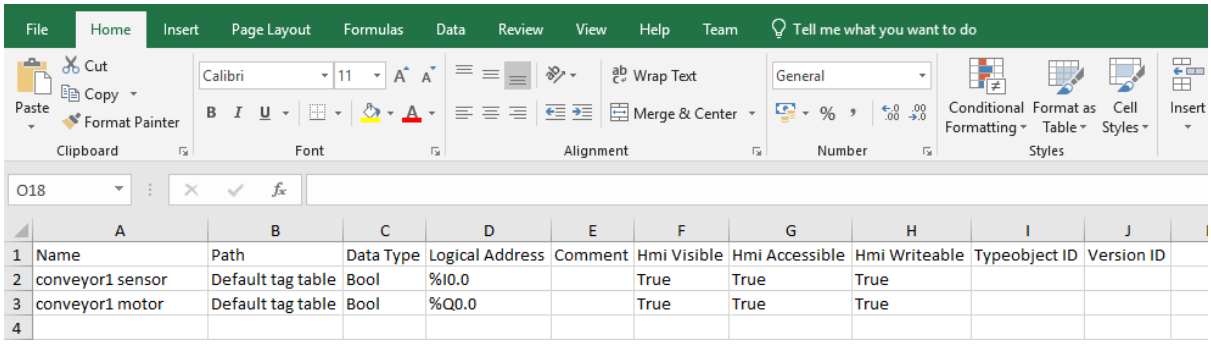

The example program logic will power the conveyor motor until the sensor detects a part. Once the sensor is activated, the conveyor will be stopped and started again after a delay. The program consists of one network with two switches from the sensor input, one timer, and an output coil for the conveyor motor. For the inputs, one of the switches is normally closed (NC), and the second switch is normally open (NO). The timer type is TON, which counts a specified time PT, and after that activates the output Q. In this case, the processing time PT is set to T#5S, so the conveyor will stop for five seconds each time the sensor is triggered.

# Exporting the tag table

When using the S7 connection plugin, the variables that are paired with the simulation must be exported from the PLC program to Visual Components using a tag table. When creating the variables for the sensor and the motor, you must use the global memory areas so that the variables become visible in the tag table and can be exported to the simulation. In the example program, the sensor memory address is in the global input %I0.0, and the motor memory address is global output %Q0.0. The full list of supported memory areas is shown in appendix 1.can be found from the help file.




The tag table can be exported from the tag table view by pressing the Export button. The supported file types are .sdf and .xlsx files. As you can only pair variables that are exported in the tag table, it is good to double-check the tag table file content after exporting. With .xlsx files this is easy, as all that you need to do is open the file in Excel to view and modify the content.




| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Name | Path | Data Type | Logical Address | Comment | Hmi Visible | Hmi Accessible | Hmi Writeable | Typeobject ID | Version ID |
| 2 | conveyor1 sensor | Default tag table | Bool | %I0.0 | | True | True | True | | |
| 3 | conveyor1 motor | Default tag table | Bool | %Q0.0 | | True | True | True | | |
| 4 | | | | | | | | | | |

The data types that are currently supported by the connection plugin are shown in the help file. When creating project-specific user-defined data types, only the main variable with the custom data type is exported with the tag table. See the appendix 13 for how to export user-defined data-types fully to your tag table.

If you are using a physical PLC, S7-PLCSIM Advanced, or S7-PLCSIM with NetToPLCSim already running, you can compile the PLC program, download it to the device, log in and start monitoring the variables. If you are using S7-PLCSIM, you must save your project, exit TIA portal, and configure NetToPLCSim to enable the virtual PLC to communicate with the simulation.

# PLCSIM and NetToPLCSim

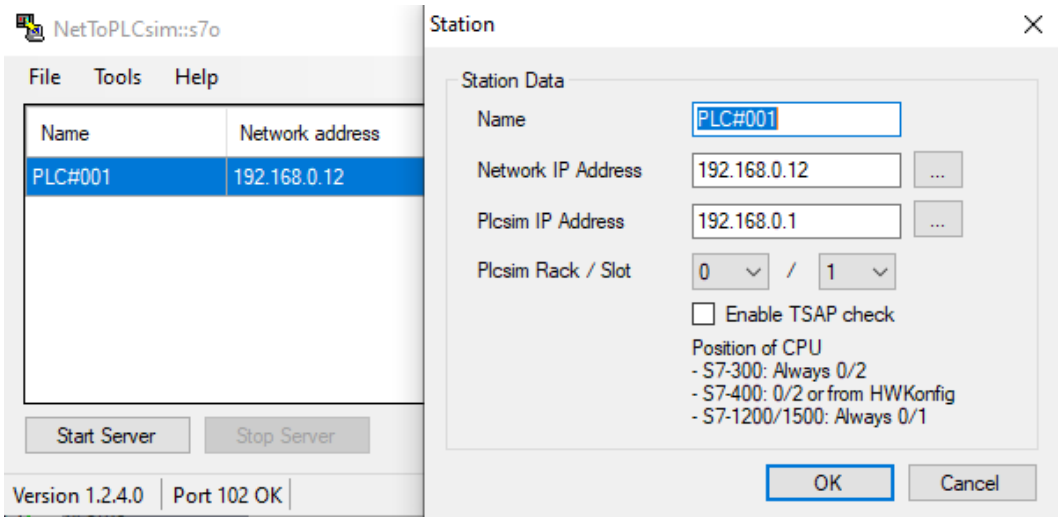
This section is only necessary if you are using PLCSIM to simulate your PLC. PLCSIM doesn't support TCP/IP communication (but PLCSIM Advanced does), so creating a direct connection with the S7 connection plugin doesn't work directly with PLCSIM. A free tool NetToPLCSim can be used as a bridge between the S7 communication and Siemens softbus interface the PLCSIM has.

NetToPLCSim is free software that can be downloaded from here:

http://NetToPLCSim.sourceforge.net/

The download comes with a full user manual for the software that you can refer to if you encounter any questions while using the software.

TIA portal and PLCSIM **must be closed** before launching NetToPLCSim to ensure that the connection works. You need to start the NetToPLCSim as an **administrator** and allow it to shut down the service that is blocking TCP port 102. Only after the service has been stopped can you open TIA Portal to start PLCSIM and load the project to it. Doing these in a wrong order is likely to make the connection fail, and a Windows **reboot** is needed to get it working again.

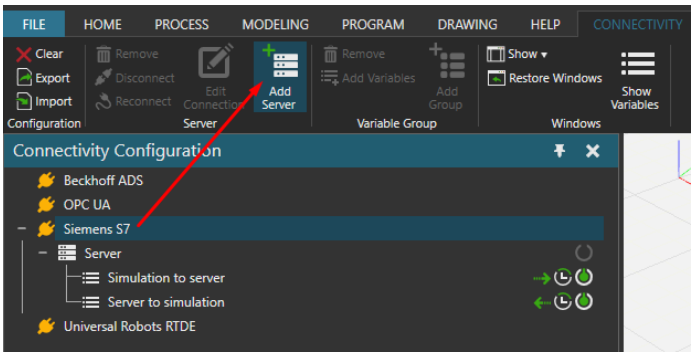


After you have NetToPLCSim, TIA portal, and PLCSIM running, you can add a new station by pressing the Add button in NetToPLCSim. A new window will open where you can set the server properties. The network IP Address is the IP address of your computer, which can be changed by going to the network adapter settings in control panel. The network IP address must be in the same address space as the PLC address. Pressing the … -buttons will open yet another window that will automatically propose valid addresses for both fields. The rack and slot indexes must be set to the correct value that is determined by your PLC type. The next section will give more information about the rack and slot indexes. After configuring the properties, you must start the server by pressing the Start Server -button.

Note for the next chapter: When using NetToPLCSim, you need to connect the S7 connection plugin to the Network IP address instead of the address of the PLC.

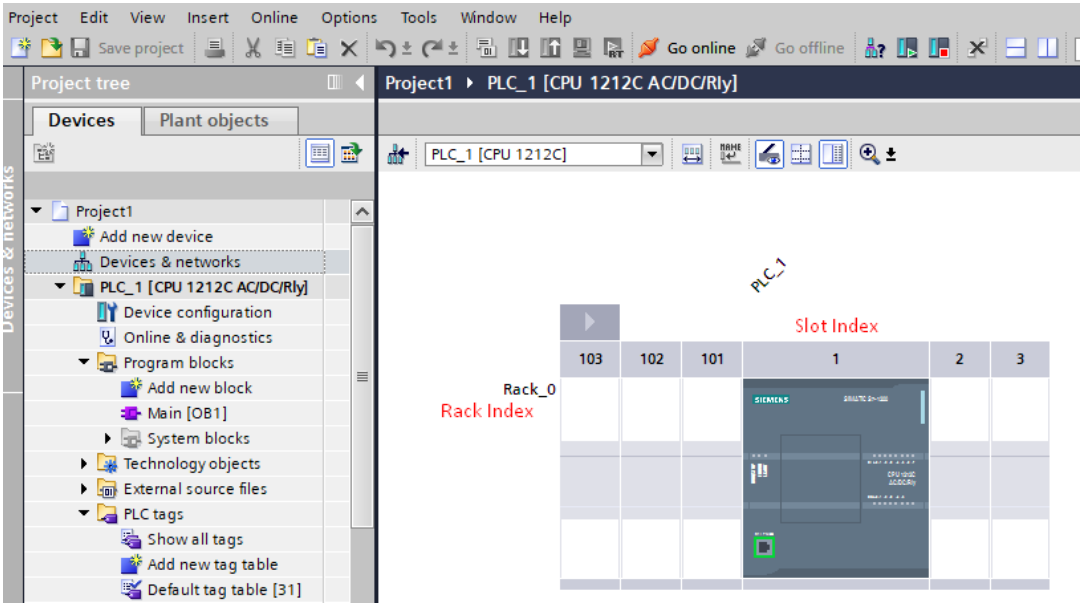
# Connecting Visual Components to the PLC

Once you have downloaded your program to the PLC and exported the tag table that contains the variables to be paired, you are ready to switch back to Visual Components. To start configuring the S7 connection plugin, open the connectivity view, select the Siemens S7 plugin, and press Add Server.




Once you have the server selected, the edit connection panel will open by default on the right side of your application. To establish a connection to the PLC, you must enter the IP address and the rack & slot indexes. If you are using PLCSIM with NetToPLCSim, the IP address will be your computer's IP address, so the Network IP address that you entered in NetToPLCSim settings. If you are not using PLCSIM, you can enter the IP address of your PLC that was configured in the TIA portal's interface properties.

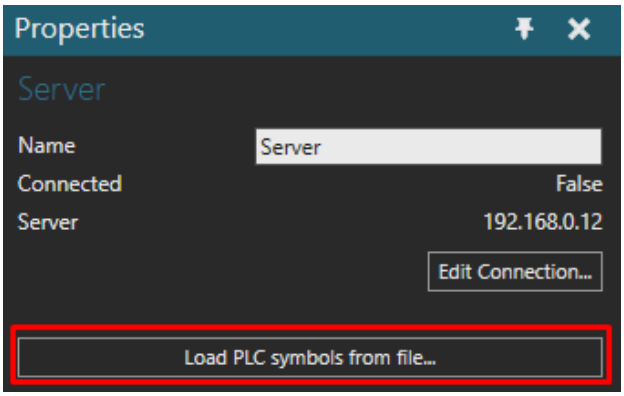
The rack and slot indexes are determined by the type of PLC that you are using. If you are not sure of the correct indexes, you can inspect the correct values from TIA portal by opening the Devices & networks view.  For S7-1200 and 1500 series PLC, the default rack index is 0, and the slot index is 1.



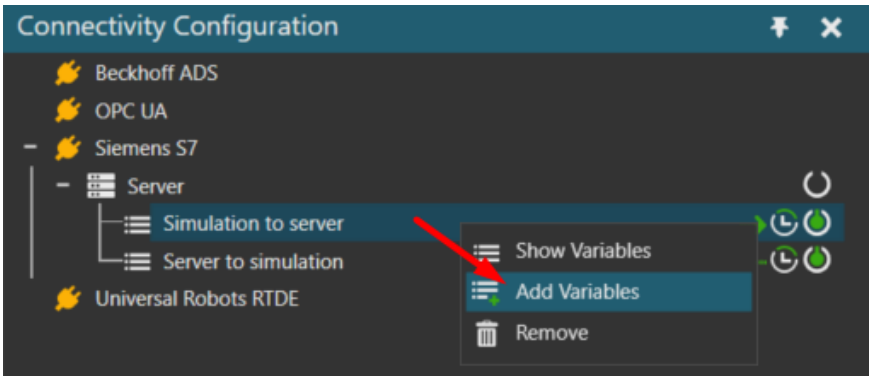

Once you have entered the correct IP address along with rack and slot indexes, you can verify that the connection can be established by pressing the Test Connection -button. Once a valid connection has been verified, you can save the server settings by pressing Apply.

# Pairing the variables

Unlike other connection plugins, the Siemens S7 connection plugin does not have a browsing functionality to pair the variables between the simulation and the PLC directly. Due to the limitation in the communication protocol, the plugin can only access the PLC variables that are in the exported tag table. To start pairing the variables, you must press the Load PLC symbols from file -button in the server properties. The file explorer will open from which you can select the tag table that you exported from TIA portal. The plugin can import tag tables in .xlsx and .sdf file formats.



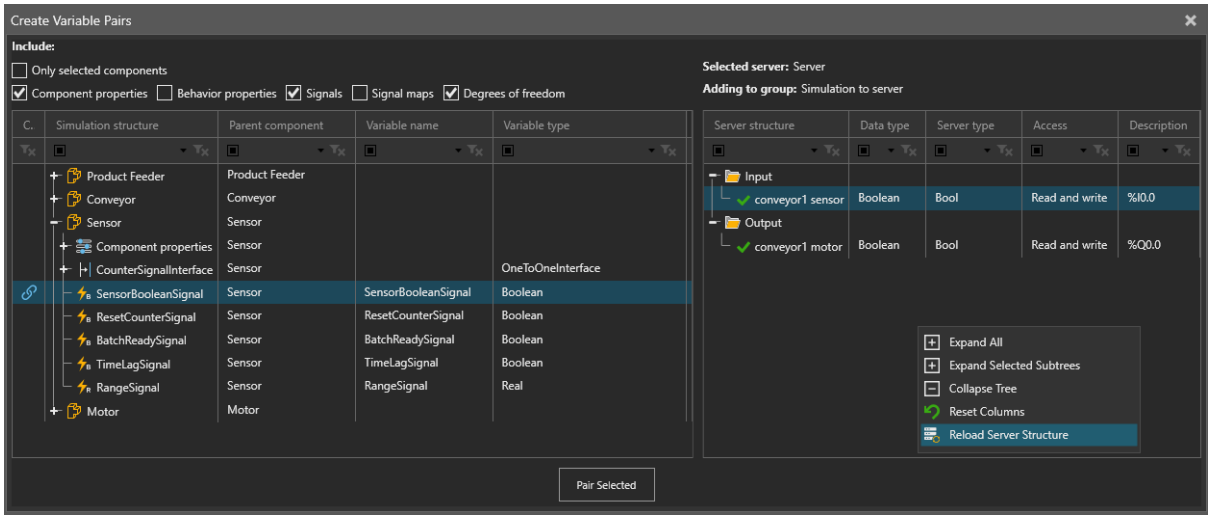

You will get a confirmation that the tags have been imported successfully, along with the number of available tags. The next step is to pair the variables that you have in the simulation with the tags that are in the PLC. The pairing must be done separately for communication directions, so simulation to server, and server to simulation. Pairing the variables is done by selecting the transmit direction and pressing the Add variables -button.



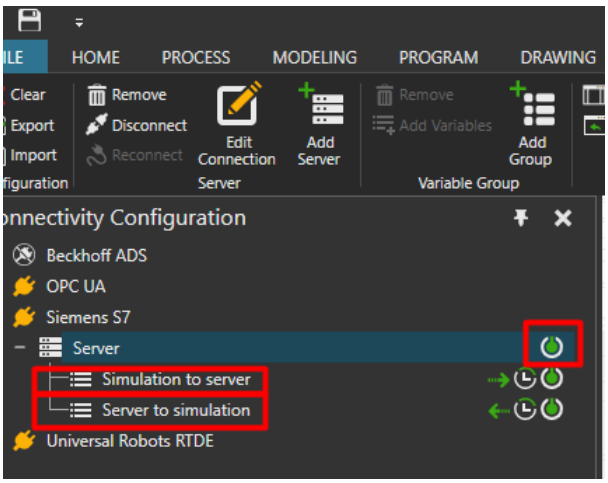

For this exercise, the sensor is used as an input, so the sensor is sending data from simulation to the server (PLC), while the motor signal is an output, so the motor control signal is sent from server to simulation. If either side of the Create variable pairs -window is empty or

showing old variables, the view should be refreshed by right-clicking on the panel and selecting Reload Structure.

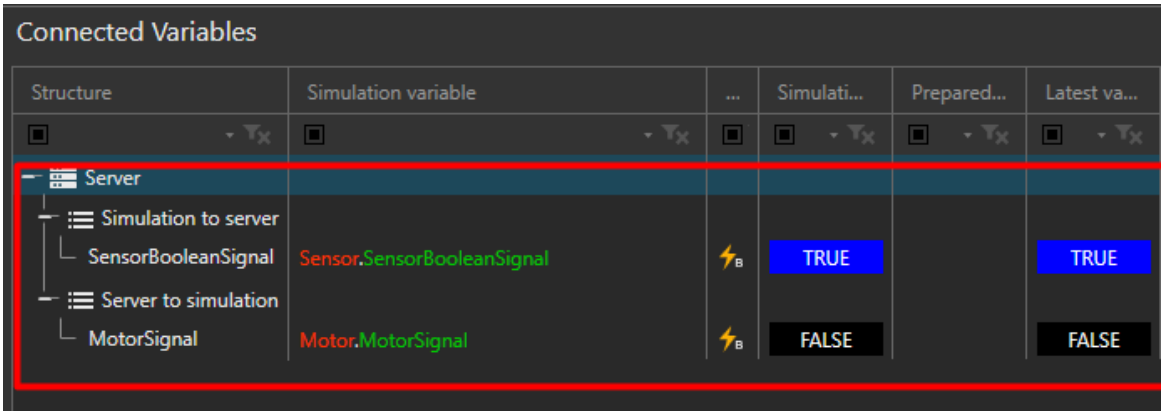
Only pair the variables for the correct transmit direction, so first in the Simulation to Server direction in the left side expand the sensor and select the ‘SensorBooleanSignal’. After the signal is selected, you can select the corresponding PLC variable ‘conveyor1 sensor’ from the right side. Once the desired variables are both selected and highlighted, press the Pair Selected -button to pair the variables together. The link icon to the left of the simulation variable indicates a successful pairing.



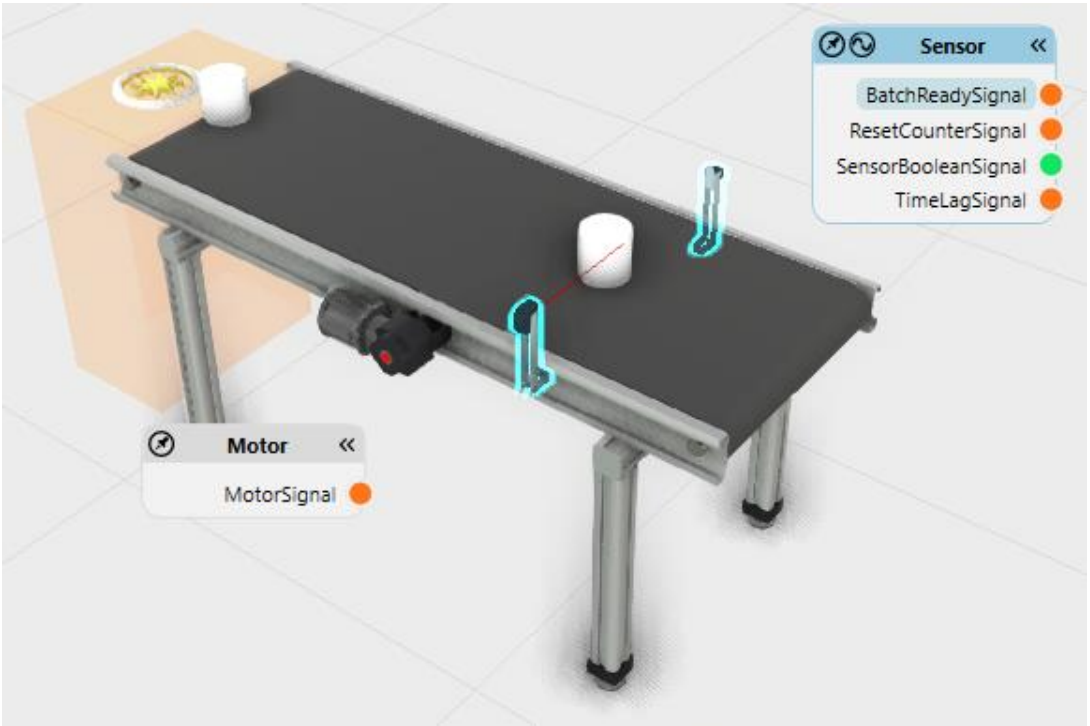

After you have paired the sensor signal you can select the server to simulation group to pair the variables in this transmit direction. For simulation side the correct variable is the MotorSignal, which is found under the motor signal interface. From server side select the conveyor1 motor variable, and press Pair Selected. Now that you have paired the variables in both directions you can connect to the server by either pressing on the icon next to the server, or by selecting the server and pressing the Reconnect button.



Once the connection is established you can view the Connected Variables window, which should show your server along with the variable groups. Make sure that the correct variables are in the correct groups as shown below.




Finally, as the connection has been enabled, and the correct variables have been paired, you are ready to start the simulation. Make sure that your PLC is in RUN-mode and hit the Play -button. The feeder will start producing parts that move on the conveyor. Once a part reaches the sensor, the SensorBooleanSignal will switch from False to True. The paired conveyor1 sensor variable should reflect this change, and change the input value for your PLC program, thus setting the MotorSignal to False for the process duration. You can also inspect the variables in TIA portal by logging in to the PLC and monitoring the program block to see the program logic operating.



# Review

In this tutorial, you learned how to use the Siemens S7 connection plugin to connect Visual Components Premium to a Siemens PLC. The tutorial result is a simple conveyor system with a sensor and a motor. As the sensor detects a part, the PLC program will shut down the conveyor motor. After five seconds, the motor will be restarted, and so the flow continues.

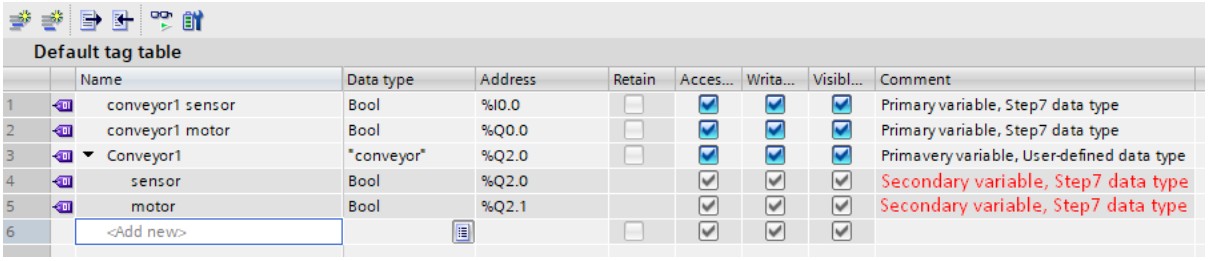
You can now adapt what you have learned to more complex projects with hundreds of IOs. Just remember that the plugin can only pair tags from the tag table, and the tags data types must be among the supported data types.

The help documentation is a great resource for learning more about connectivity. You can open the Help documentation in Visual Components by pressing F1. Connectivity related topics are found in contents view by navigating to Tasks - Connectivity.

# Appendix

## Using user-defined data types

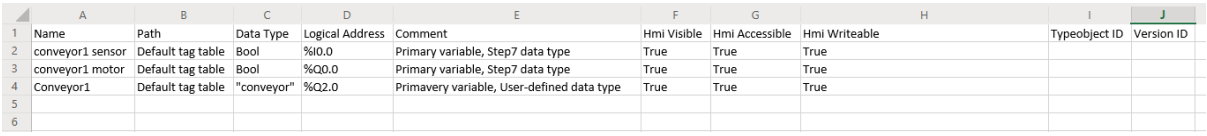
Your project specific PLC program might contain user-defined data types that consist of multiple Step7 data types. When you export the tag table from TIA portal, only the heading variable with the custom data type are exported, while the contained Step7 data types are not exported.



Default tag table

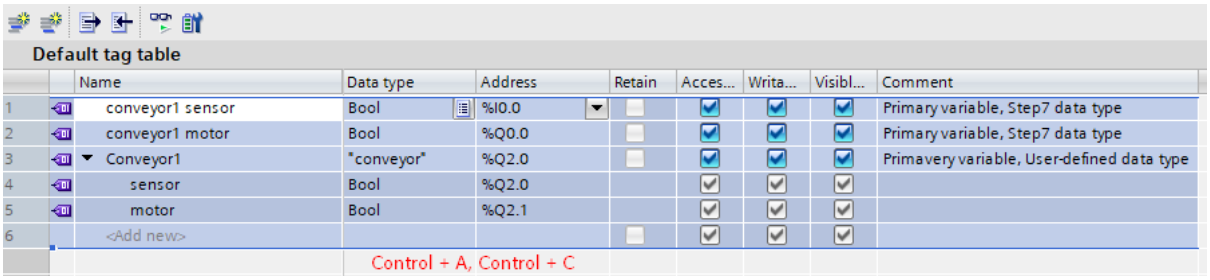
| | Name | Data type | Address | Retain | Acces... | Writa... | Visibl... | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | conveyor1 sensor | Bool | %I0.0 | | ☑ | ☑ | ☑ | Primary variable, Step7 data type |
| 2 | conveyor1 motor | Bool | %Q0.0 | | ☑ | ☑ | ☑ | Primary variable, Step7 data type |
| 3 | Conveyor1 | "conveyor" | %Q2.0 | | ☑ | ☑ | ☑ | Primavery variable, User-defined data type |
| 4 | sensor | Bool | %Q2.0 | | ☑ | ☑ | ☑ | Secondary variable, Step7 data type |
| 5 | motor | Bool | %Q2.1 | | ☑ | ☑ | ☑ | Secondary variable, Step7 data type |
| 6 | <Add new> | | | | ☑ | ☑ | ☑ | |

Since the S7 connection plugin only supports the system variable data types that are shown in the help file, user-defined data types are not visible in Visual Components application.



| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Name | Path | Data Type | Logical Address | Comment | Hmi Visible | Hmi Accessible | Hmi Writeable | Typeobject ID | Version ID |
| 2 | conveyor1 sensor | Default tag table | Bool | %I0.0 | Primary variable, Step7 data type | True | True | True | | |
| 3 | conveyor1 motor | Default tag table | Bool | %Q0.0 | Primary variable, Step7 data type | True | True | True | | |
| 4 | Conveyor1 | Default tag table | "conveyor" | %Q2.0 | Primavery variable, User-defined data type | True | True | True | | |
| 5 | | | | | | | | | | |
| 6 | | | | | | | | | | |

To get around this issue, the user may add the Step7 data types to the exported .xlsx or .sdf file directly. Select the desired variables from the tag table in TIA portal and copy the tags with Control + C. Control + A can be used to select all of the tags in the tag table.



Default tag table

| | Name | Data type | Address | Retain | Acces... | Writa... | Visibl... | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | conveyor1 sensor | Bool | %I0.0 | | ☑ | ☑ | ☑ | Primary variable, Step7 data type |
| 2 | conveyor1 motor | Bool | %Q0.0 | | ☑ | ☑ | ☑ | Primary variable, Step7 data type |
| 3 | Conveyor1 | "conveyor" | %Q2.0 | | ☑ | ☑ | ☑ | Primavery variable, User-defined data type |
| 4 | sensor | Bool | %Q2.0 | | ☑ | ☑ | ☑ | |
| 5 | motor | Bool | %Q2.1 | | ☑ | ☑ | ☑ | |
| 6 | <Add new> | | | | ☑ | ☑ | ☑ | |

Control + A, Control + C

Paste the tags to your .xlsx or .sdf file with Control + V.

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Name | Path | Data Type | Logical Address | Comment | Hmi Visible | Hmi Accessible | Hmi Writeable | Typeobject ID | Version ID |
| 2 | conveyor1 sensor | Default tag table | Bool | %I0.0 | Primary variable, Step7 data type | True | True | True | | |
| 3 | conveyor1 motor | Default tag table | Bool | %Q0.0 | Primary variable, Step7 data type | True | True | True | | |
| 4 | Conveyor1 | Default tag table | "conveyor" | %Q2.0 | Primavery variable, User-defined data type | True | True | True | | |
| 5 | | | | | | | | | | |
| 6 | | | | | Control + V | | | | | |
| 7 | conveyor1 sensor | Bool | %I0.0 | FALSE | TRUE | TRUE | TRUE | Primary variable, Step7 data type | | |
| 8 | conveyor1 motor | Bool | %Q0.0 | FALSE | TRUE | TRUE | TRUE | Primary variable, Step7 data type | | |
| 9 | Conveyor1 | conveyor | %Q2.0 | FALSE | TRUE | TRUE | TRUE | Primavery variable, User-defined data type | | |
| 10 | sensor | Bool | %Q2.0 | | Accessible | Accessible | Accessible | | | |
| 11 | motor | Bool | %Q2.1 | | Accessible | Accessible | Accessible | | | |
| 12 | | | | | | | | | | |

The tag data will initially not match the header format, so you will have to reformat the data to a suitable format. This can be done manually by cutting and pasting the data to correct cells, or for larger projects, the user should implement a script to format the data automatically.

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Name | Path | Data Type | Logical Address | Comment | Hmi Visible | Hmi Accessible | Hmi Writeable | Typeobject ID | Version ID |
| 2 | conveyor1 sensor | Default tag table | Bool | %I0.0 | Primary variable, Step7 data type | True | True | True | | |
| 3 | conveyor1 motor | Default tag table | Bool | %Q0.0 | Primary variable, Step7 data type | True | True | True | | |
| 4 | Conveyor1 | Default tag table | "conveyor" | %Q2.0 | Primavery variable, User-defined data type | True | True | True | | |
| 5 | sensor | Default tag table | Bool | %Q2.0 | Secondary variable, Step7 data type | True | True | True | | |
| 6 | motor | Default tag table | Bool | %Q2.1 | Secondary variable, Step7 data type | True | True | True | | |
| 7 | | | | | | | | | | |

After reformatting the exported tag table so that the desired variables are available with the data in the correct fields, it is possible to import the file to the connection plugin and pair the PLC variables with the simulation.