
SIEMENS

# Simulation with SIMIT Simulation Platform and PCS 7 in a practical example

SIMIT Simulation Platform V11, SIMATIC PCS 7 V9.1 SP2

https://support.industry.siemens.com/cs/ww/en/view/77362399

Siemens Industry Online Support

# Legal information

**Use of application examples**

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

**Disclaimer of liability**

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

**Other information**

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

**Security information**

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/cert.

# Preface

### Aim of this Application Example

The aim of this application example is to introduce additional functions and libraries in a practical example, in addition to the application example "SIMIT Simulation V11.0 SP1 Getting Started". The associated example project includes a complete simulation from the signals to the process.

### Key Content

The following main topics are covered in this application example:

- The structure of a simulation project from the signal level to the process level (partly with reference to existing documentation)
- Customizing templates
- The structure of the process level with the FLOWNET library
- The simulation of a conveyor system with components of the CONTEC library
- Automatic generation of the device level using the simulated conveyor system
- Pre-prepared scripts which can be used for operator training, for example.

### Validity

- SIMIT Simulation Platform V11
- SIMATIC PCS 7 V9.1 SP2

# Table of contents

# 1 Task Description and Solution

## 1.1 Task

PCS 7 projects are becoming more and more complex due to the increasingly significant demand for availability and individuality of systems. In this regard, the automation programs must also undergo extensive testing.
To make this possible, certain system states and feedback are required from actuators and sensors to test whether the automation program is functioning correctly.
The provision of feedback or the system state is very laborious or not possible without a suitable tool. For this reason, nowadays one can find tools such as SIMIT Simulation Platform (hereafter referred to as SIMIT), which simplify the simulation of signals, devices and process states in a significant way.

## 1.2 Solution

This application example given here describes how to use the SIMIT simulation software to easily and quickly create the required simulation for a unit for the manufacture and packaging of soft drinks. The plant sections raw material tanks, reactors and a bottling plant are used for the simulation.

The basis for the simulation project described here is the PCS 7 project "bottling plant", which you can find on the same article page.

The application example provides a template which includes the simulation of important physical processes, devices, and signals of a raw material tank, stirred tank reactor and the filling unit. The installation is modular and is based on physical principles.

Its utilization offers the following advantages:

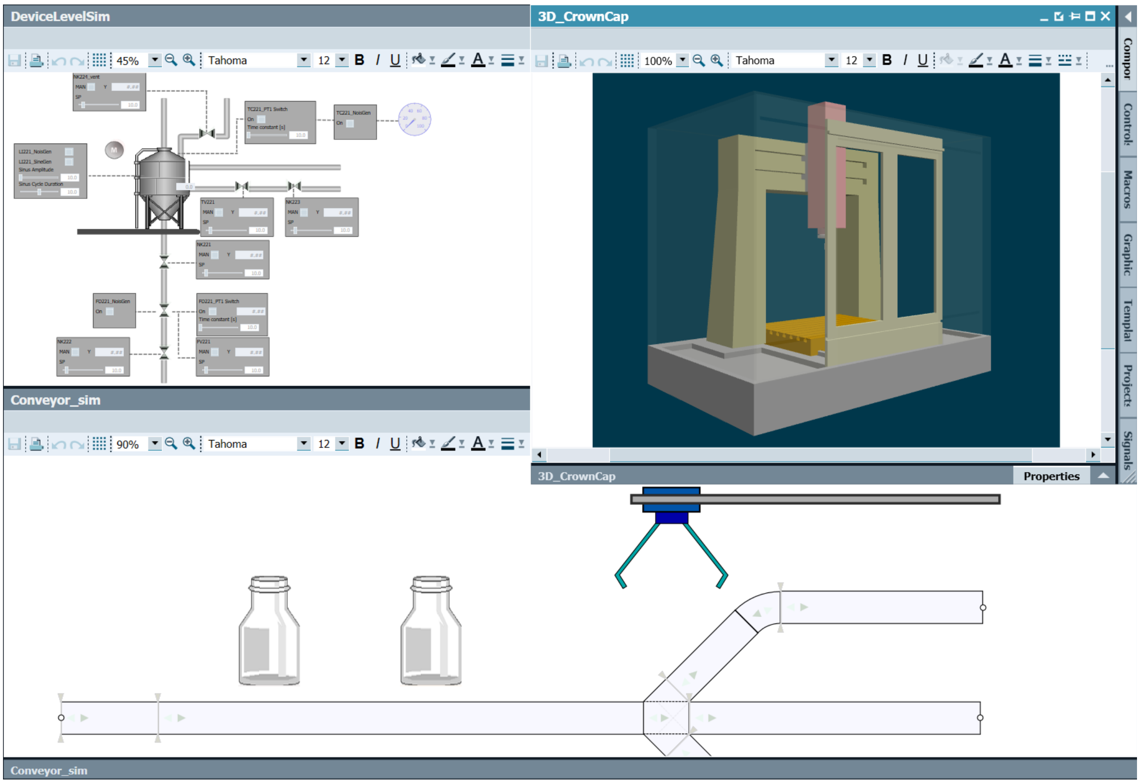
- A reduction of the knowledge necessary to develop simulations
- A decrease in the configuration effort
- Flexible installation and adjustment
- Standardized structures

### 1.2.1 Overview of the complete solution

#### Diagram

The following figure shows parts of a possible style depth of a simulation solution of a filling unit.

Figure 1-1




#### Description

The application example "virtual commissioning with SIMIT Simulation Platform for typical process and production automation" includes a PCS 7 project from a beverage blending and bottling plant and the associated simulation model. The PCS 7 project does not form part of the description given here and serves solely as a basis for the description of the simulation model.

The simulation model is divided into three levels:

- Signal level
- Device level
- Process level

You can find a description of the levels and of SIMIT in the application example "SIMIT Simulation V11.0 SP1 Getting Started" in chapter 1 "SIMIT at a glance".

**Aim**

The aim of the application example given here is to introduce functions and libraries which are not described in Getting Started. In this example, the following topics will be highlighted in particular:

- FLOWNET library
- CONTEC library
- The function "Generating the device level"
- Script function
- Creating/optimizing templates

**Delimitation**

Physically speaking, the technical process is illustrated in a simplified way by assuming ideal conditions.

**Required Knowledge**

Fundamental knowledge of the following specialist fields is a prerequisite:

- Basic knowledge of process technology
- Basic knowledge of physical modeling
- Engineering with SIMATIC PCS 7 and Advanced Process Library (APL)
- Knowledge of control technology

### 1.2.2 Core Functionality

The individual components of the PCS 7 project "Bottling Plant" simulation are described in the following section. The simulation consists of three main components:

- Raw material tanks
- Reactors
- Filling

The main components with the associated technical functions are derived from the technological hierarchy of the PCS 7 project.

You can find information on the generation procedure and the individual components such as signals, devices and processes in the application example "SIMIT simulation of a stirred tank reactor with PCS 7".

### 1.2.3 Hardware and Software Components

The application example has been created with the following components:

#### Software Components

Table 1–1

| Component | Readme documents |
|---|---|
| SIMATIC PCS 7 V9.1 SP2 | https://support.industry.siemens.com/cs/ww/en/view/109806027 |
| S7-PLCSIM V5.4 SP8 | https://support.industry.siemens.com/cs/ww/en/view/109750064 |
| SIMIT SP V11 | https://support.industry.siemens.com/cs/ww/en/view/109810223 |

**Note**

SIMIT V8.1 is offered with different license models. An overview of the contained models is available in chapter 9.1 "License model" of the manual "SIMATIC SIMIT Simulation Platform (V11)". Since this application example has about 500 simulation tags, a license model that can handle this number should be used. In the application example, the modules "PLCSIM coupling" and "CMT import (CMT - control module type)" as well as the libraries FLOWNET and CONTEC are used. These do not form part of SIMIT SP V11.

#### Hardware Components

**Note**

Please take heed of the suggested hardware configuration for installing the software components.

The suggested hardware configuration can be found in the PCS 7 Readme V9.1 SP2 (Online).



#### Example Files and Projects

The following table contains all the files and projects used in this application example.

Table 1–2

| File/project | Note |
|---|---|
| 77362399_BottlingPlant_SIMIT_PROJ_V912.zip | SIMIT SP V11 example project |
| 77362399_BottlingPlant_PCS_7_PROJ_V912.zip | PCS 7 V9.1 SP2 example project |
| 77362399_DOCU_Bottling_Plant_en.pdf | This document |

# 2 Initial Work on the Project

The following will demonstrate how to create the simulation project. The functions and approaches which are already described in other application examples will simply be named. The corresponding description will be referenced.

## 2.1 Presentation of the Project

### Description of the Plant

The basic liquid materials are dispensed from three raw material tanks into two reactors. The flow rates are regulated by valves. The liquids are then heated or cooled in the reactors. Then the liquids are bottled. After the filling process, the bottles are taken away on conveyor belts.

### Task Description for the PCS 7 Project "Bottling Plant"

The bottling plant is depicted in SIMIT:

- Generating the signal level and coupling with PLCSIM
- Generating the device level and the simulation of the conveyor belt
  While doing this you will get to know the function "Generating the device level".
- Generating the physical models
  When generating the physical models you will get to know the FLOWNET libraries.
- Generating the scripts which can be used for operator training sessions, for example.

## 2.2 Configuring the PLCSIM Interface in the SIMATIC Manager

Before you start configuring SIMIT, you must first retrieve the "BottlingPlant_MP" project in the SIMATIC Manager, change the settings and load PLCSIM.

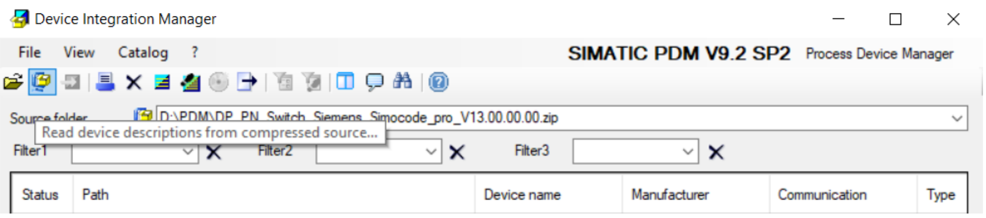
In order to load the CPU AS1, device data of the SIMOCODE pro V must be added in this project via the SIMATIC Process Device Manager.

The corresponding device data can be found in the SIMOCODE pro EDD V13.0, which can be obtained via the following link:

SIMOCODE pro EDD V13.0

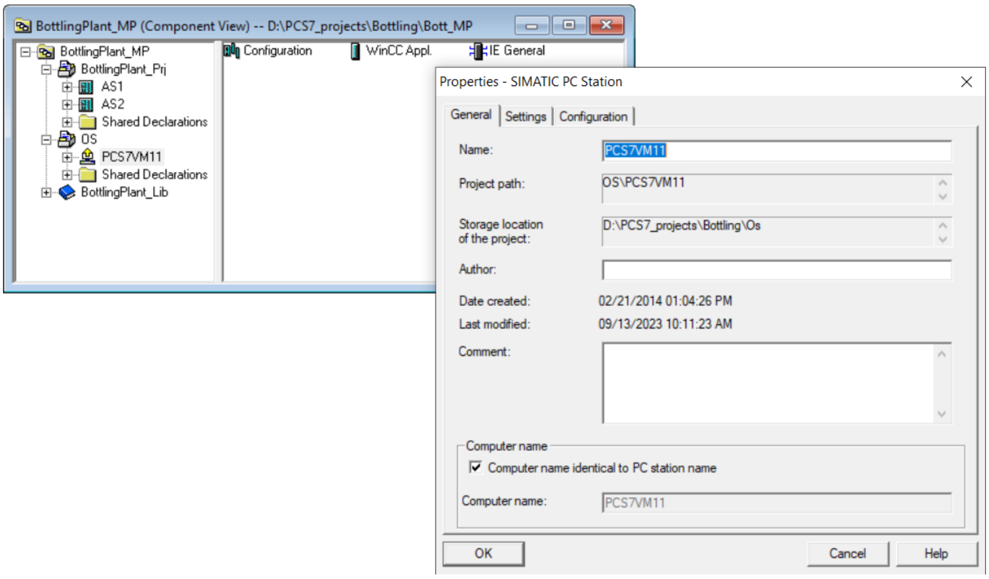
### Preparations in PCS 7 / Process Device Manager

1. Start the SIMATIC Manager. Restore and open the PCS 7 repository.
2. Start the Device Integration Manager and import the device description from the archived source "SIMOCODE pro EDD V13.0".

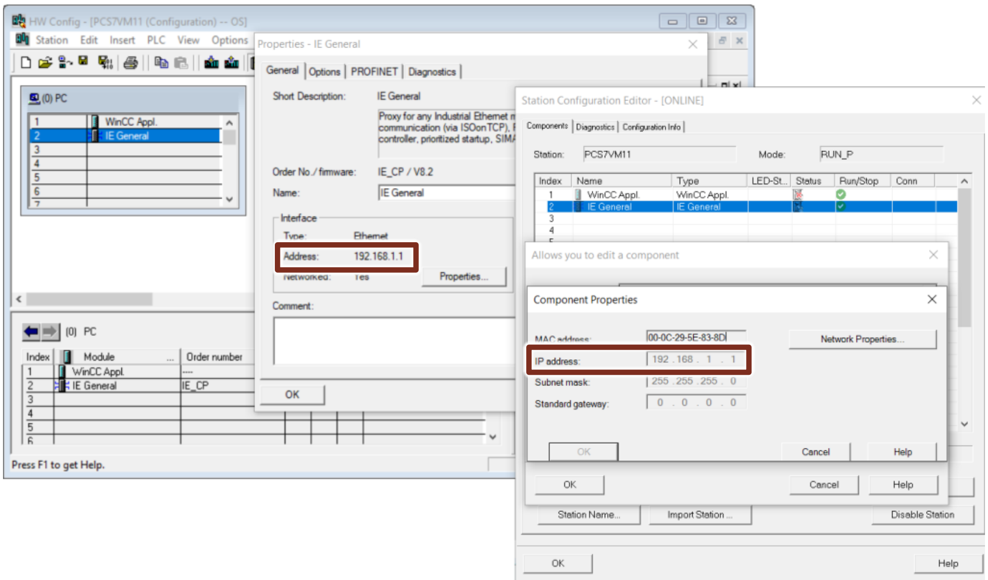



3. Integrate the imported device data in the PDM to PCS 7.

4. Restart PCS 7 and open the properties of the OS station in the Component View.
5. Enter the name of your PC in the "Computer name" field (in this example "PCS7VM11").

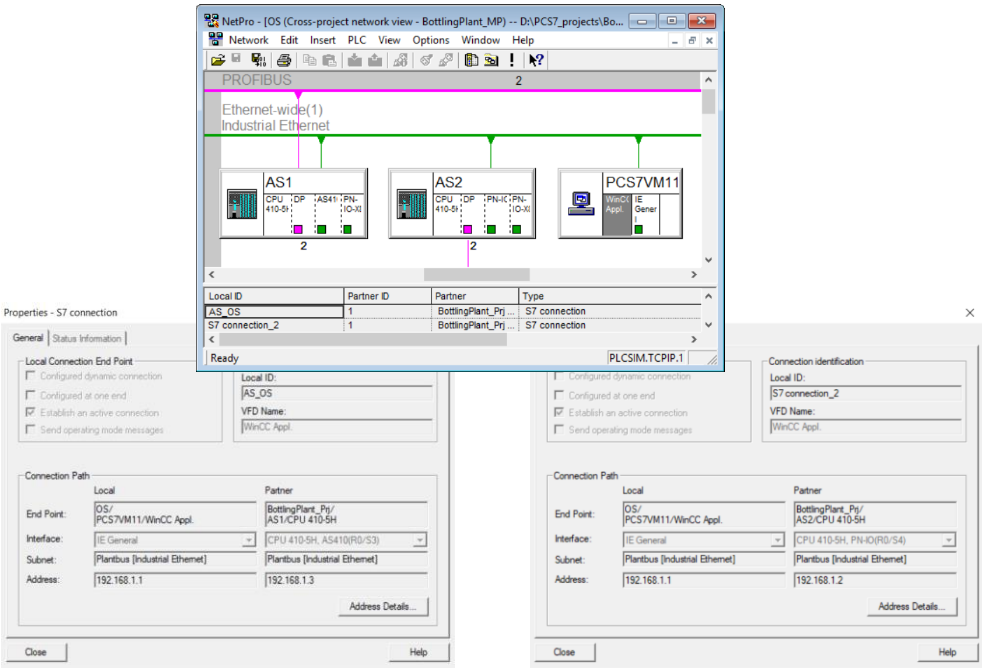



6. Open the Station Configuration Editor and the hardware configuration of the OS. Align the configuration in PCS 7 with that of the Station Configuration Editor and assign the correct IP address to the OS.

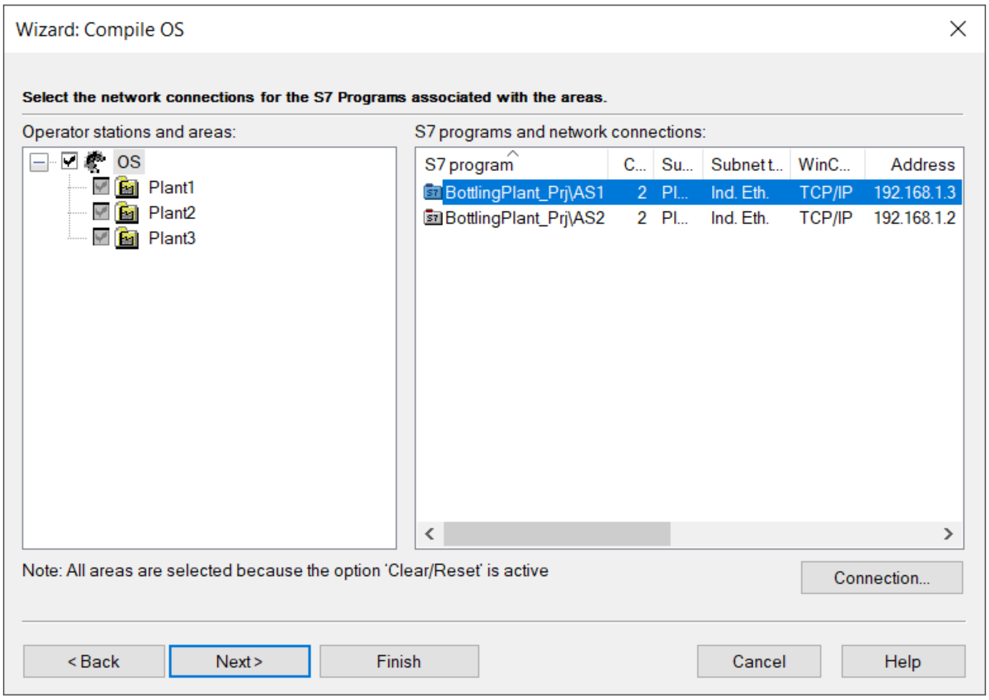



7. Assign IP addresses to AS1 and AS2 that are in the same subnet.

8. Configure the connections between AS1, AS2 and OS in NetPro.




9. Start the function "PLC > Configure..." for the OS.
10. Load the configuration of the OS into the Station Configuration Editor using the "Compile" function. Make sure that Industrial Ethernet with TCP/IP is selected as the protocol as the connection from both stations to the OS.




11. Open the OS project with WinCC Explorer.
12. Open the properties of the configured computer and enter the name of your PC by clicking the "Use Local Computer Name" button.

13. Open the Tag Management. Select "TCP/IP > System Parameters" ("TCP/IP > System Parameters") and change the "Logical Device Name" to "PLCSIM.TCPIP.1".
14. Open an instance of PLCSIM. In the drop-down list, select "PLCSIM(TCP/IP)" as the interface. Start the "PLC >> Compile and Configure Objects" function for AS1.
15. Open another instance of PLCSIM. In the drop-down list, select "PLCSIM(TCP/IP)" as the interface. Start the "PLC >> Compile and Configure Objects" function for AS2.
16. Put both PLCSIM instances in RUN-P

**Export of symbol tables**

SIMIT provides an import wizard for PLCSIM coupling, with which you can easily import the symbols contained in the AS program.

This requires a one-time export of the symbol table from the AS program. To do this, follow these steps:

1. Open the symbol table of the AS program.
2. Sort the table by addresses.
3. Select all incoming and outgoing addresses. These will be needed later in SIMIT.
4. Export the symbols using the menu function "Symbol Table > Export...".
5. Save the symbols in ASCII format with a unique name.

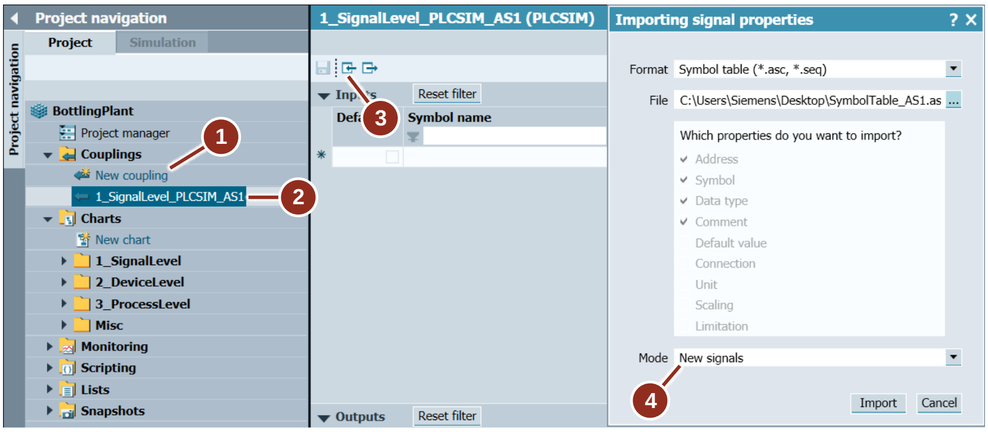
**NOTE**

The "BottlingPlant_MP" project contains two automation systems. Therefore, steps 1 through 5 must be performed for AS1 and AS2.

## 2.3 Configuring the PLCSIM Interface in SIMIT

To enable communication between the process level and the automation system, the coupling must be completed in SIMIT.

1. Switch to the project view of SIMIT.
2. In the "Coupling" folder, double-click on the "New coupling" entry (1).
3. Select the "PLCSIM" option in the dialog.
4. Change the name of the pairing to "1_SignalLevel_PLCSIM_AS1".
5. Open the coupling in the workspace by double-clicking (2).
6. Import (3) the input and output signals from the export file of the symbol table for AS1.
7. Select the "New signals" mode (4) and start the import.

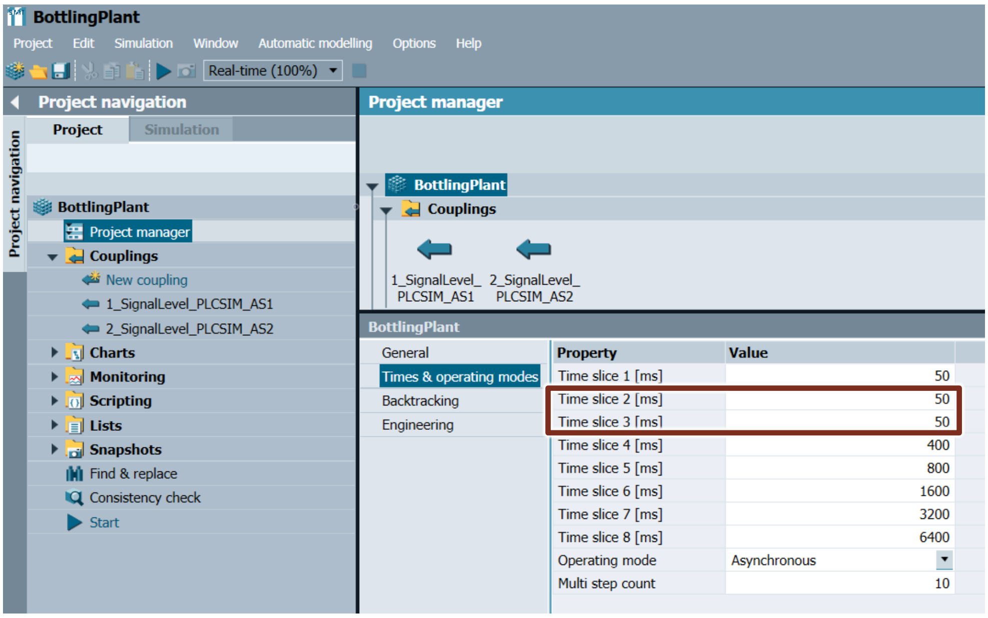



8. Repeat steps 2 through 7 for AS2. Name this pairing "2_SignalLevel_PLCSIM_AS2".

### Assigning the Time Slice

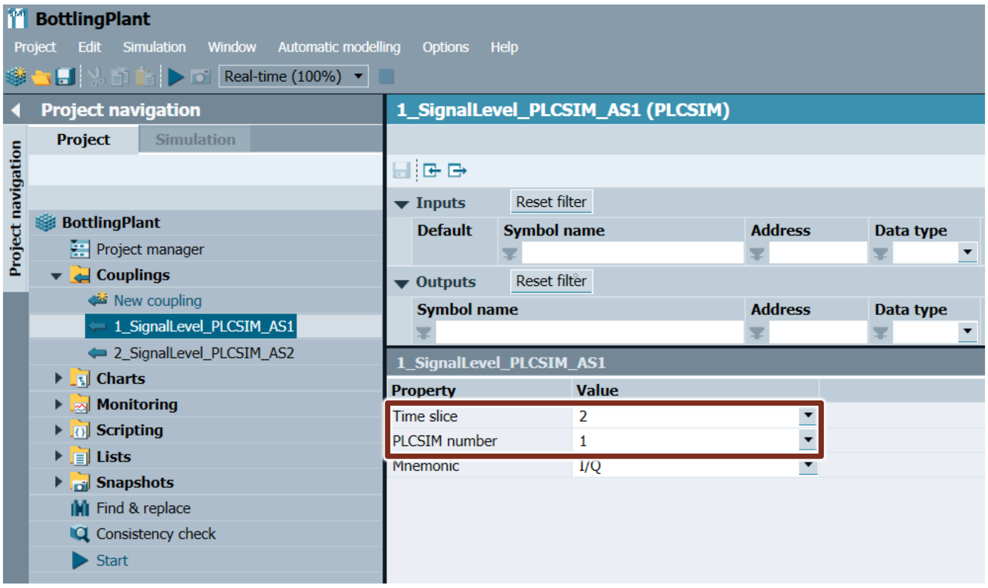
You can set the cycle times of the time slices in the properties window of the project manager. Time slices 2 and 3 are used in the application example. A cycle time of 50 ms is set for each time slice.

Figure 2-1




You can assign the relevant time slice in the properties window of the coupling. You can enter the number of the PLCSIM instance in the field "PLCSIM number". You will find the number in the header of the PLCSIM window for each AS. The PLCSIM number 1 is also assigned to the coupling for AS1 in the following image.

Figure 2-2



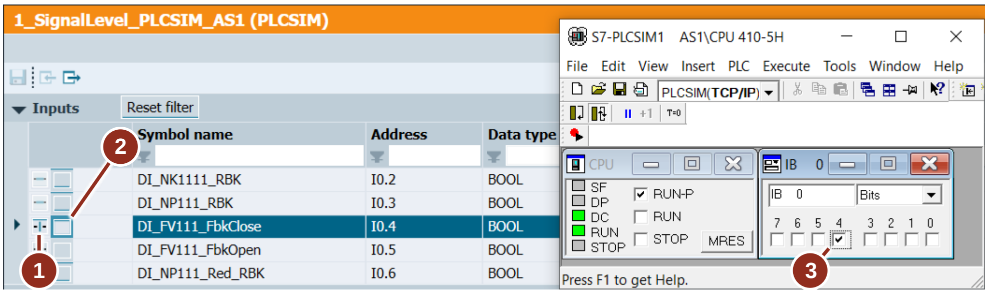

Set the time slice 3 and the PLCSIM number 2 for AS2.

## 2.4 Verifying Communication between PLCSIM and SIMIT

In order to prevent communication errors between PLCSIM and SIMIT, you can test the connection in a simple way.

Proceed as follows:

1. Start the simulation runtime ( ▶) in SIMIT. PLCSIM must have been started; otherwise, the system issues an error message.
2. Activate the signal isolator (1) so that you can operate the signal manually.
3. Activate a binary signal using the associated switch (2), e.g. "DI_NK111_FbkClose" with the address "I0.4".
4. If the connection is working, the system displays the signal in PLCSIM in input peripheral "IB 0 Signal 4" (3).




After this procedure, check the communication between SIMIT and AS2 with the signal "DI_Conv1_Maint" with the address "I1.2".

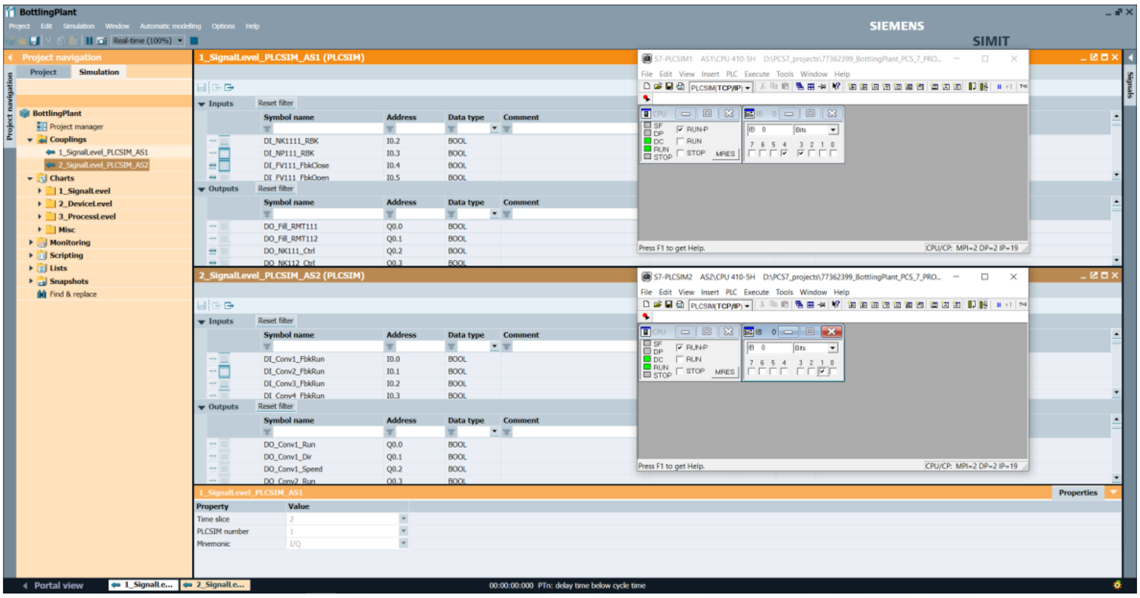
## 2.5 The Current Status of your Project

In preparation for the actual creation of the process simulation for the "Bottling Plant" project, you have completed the following tasks:

- You have retrieved the PCS 7 project "Bottling Plant" and downloaded the configured hardware in the virtual AS (PLCSIM).
- You have created two PLCSIM couplings in SIMIT and imported the symbol tables you had previously exported from PCS 7 into SIMIT.
- You have set the cycle times for the time slices and assigned a time slice to each coupling.
- You have assigned the PLCSIM number to each coupling.
- You have tested the communication between PLCSIM and SIMIT.

At the current state of your project you are now able to perform signal tests. At this stage of the simulation, you can see whether the signals are communicated from the control to the signal level (see Figure 2-3).

Figure 2-3




Tests such as the opening and closing of valves can be validated at this stage. You are still unable to simulate the time response or the transient response of processes. This requires you to describe the process in detail and simulate it in SIMIT. The following chapters describe a possible process simulation for the "Bottling Plant" project.

# 3 Creating the Device Level

In the previous chapter, the PLCSIM couplings were configured. The PLCSIM data can be accessed for writing and reading.
If the AS sends control signals, these are sent in real application to actuators. These then perform an action that changes the process in some way. This could be, for example, a level increase or the sinking of a mass flow. This change must be detected by sensors and sent back to the AS.

In this chapter, the actuators and sensors are replicated. While doing this, you will get to know and use the functions "CMT import" and "Generation of a device level".

## 3.1 Creating a Template

To be able to use the "CMT import" and "Generation of a device level" functions effectively, you must create corresponding templates for the actuators and sensors used. The templates must match the CMTs in PCS 7. That means that the template name is identical to the CMT name.

The following list comprises the required templates.

- AMON_Std
- BottlingAnalogVlv
- Bottling_2WayValve
- Bottling_Dose
- Bottling_MotorLean
- Bottling_MotorRev
- Bottling_MotorSimoCode
- Bottling_MotorSinamics
- Bottling_PID
- Bottling_ValveLean
- Conv

There are two options for creating templates. The first option is to adapt the existing templates (base templates). The second option is to create a new one.
In order to adapt the base templates, they have to be copied. For example, they could be copied into the folder project templates. After that it is possible to open and edit them.

**Procedure**

The following section will provide you with some examples of how to create or adapt templates.

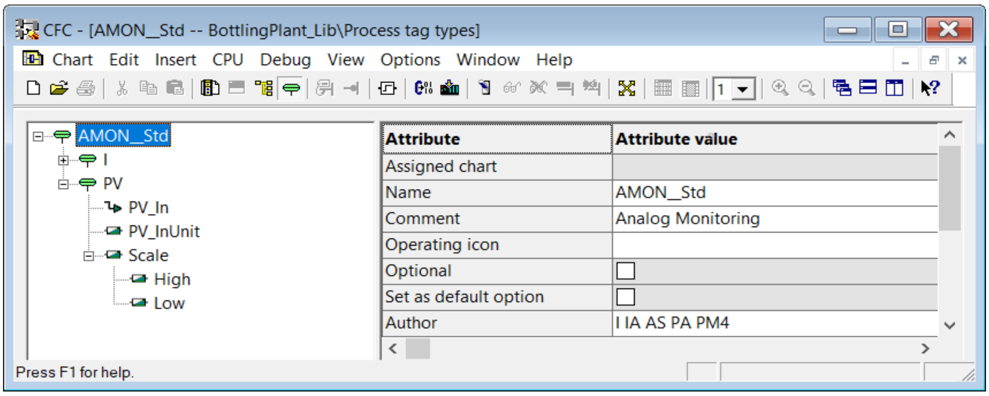
AMON_Std:

First of all, the interfaces to the CMT must be defined before creating a template. To do this, the following tasks must be performed in the SIMATIC Manager:

1. Open the CMT "AMON_Std" in the master data library of the "BottlingPlant" project.
2. Open the technical connections of the CMT "View > Technical connections".
3. Take note of the names of the assigned blocks and the names of the associated signals and parameters which you need in the SIMIT template.

The following figure outlines these for the "AMON_Std".

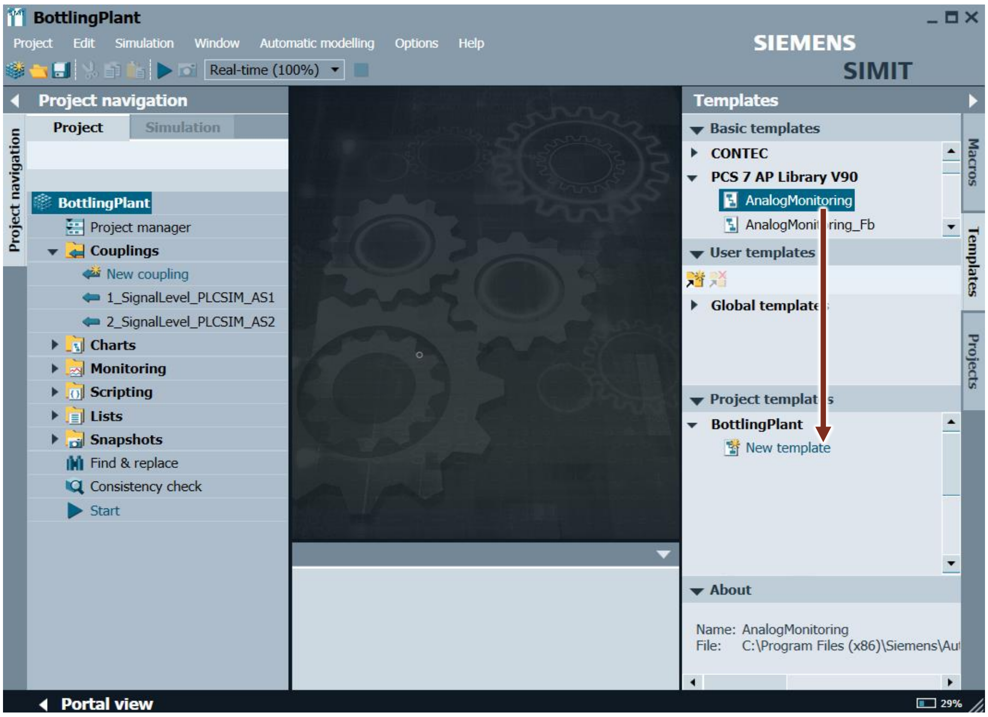
Figure 3-1




The template "AMON_Std" is created from the base template "AnalogMonitoring" from the folder "PCS 7 AP Library V90". The following tasks need to be performed:

4. Open the base template folder "PCS 7 AP Library V90" in the Task-Card "templates" in the Tools window.
5. Select the template "AnalogMonitoring".
6. Drag the template and drop it into the project templates.

Figure 3-2




7. Change the name to "AMON_Std".
8. Double-click to open the template.

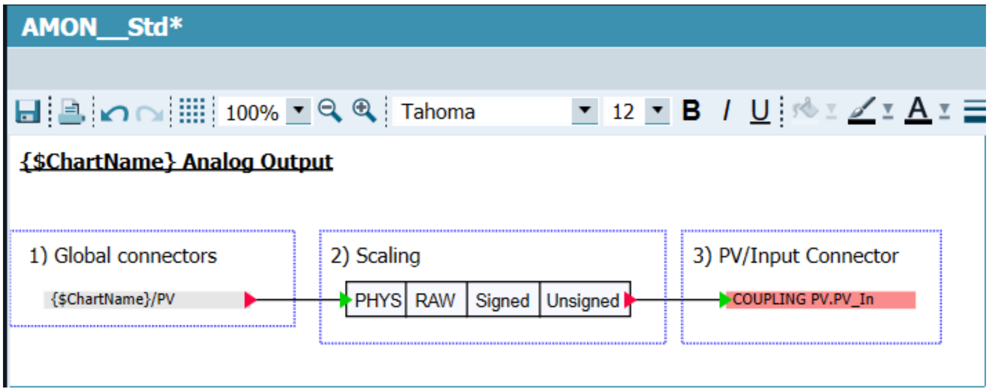
9. Make the changes according to the following table.

Table 3–1

| Component | Property | Old value | New value |
|---|---|---|---|
| Connector | Name | {$ChartName}/X | {$ChartName}/PV |
| Phys2Raw | Name | Phys2Raw#1 | {$ChartName}_Phys2Raw |
| | Phys_Lower_ Limit | PV_LR_Value | PV\Scale\Low |
| | Phys_Upper_ Limit | PV_HR_Value | PV\Scale\High |
| Signed2Unsigned | Name | Signed2Unsigned #1 | {$ChartName}_Signed2Unsig ned |
| Input | Signal | COUPLING PV_SymbolName | COUPLING PV.PV_In |

10. Insert text boxes to improve clarity according to the following figure.

Figure 3-3




11. Save and close the template.

Bottling_2WayValve:

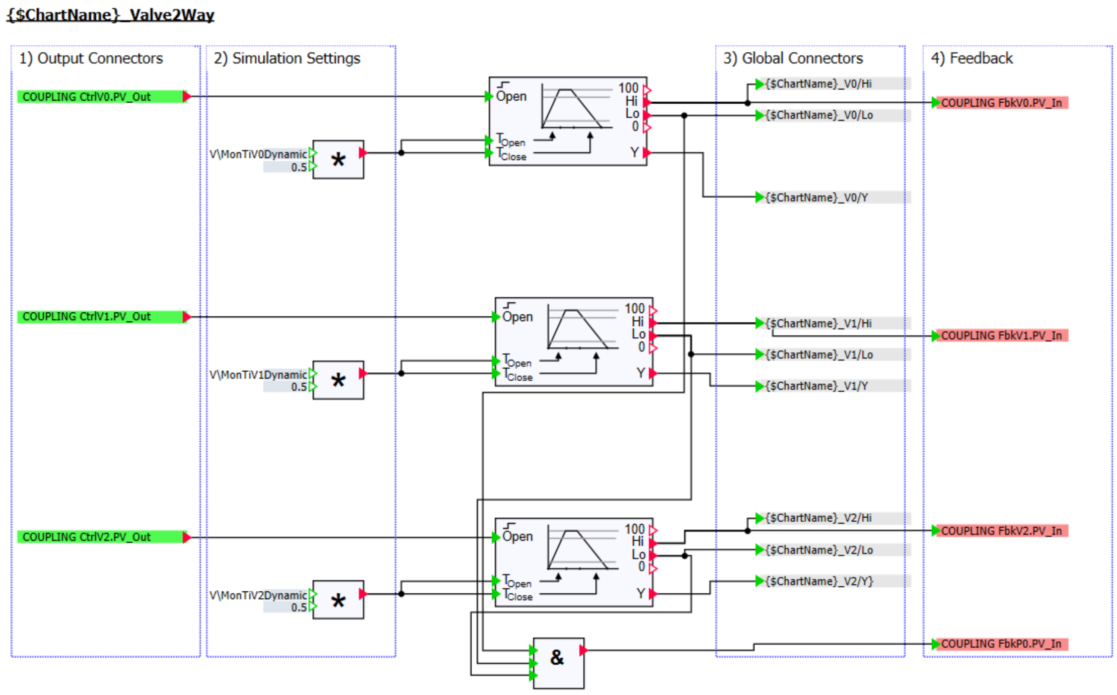
Before creating the template, the names of the assigned blocks and the associated signals and parameters which you want to address in the SIMIT template need to be compiled (see steps 1 to 4 AMON_Std). The following table compiles these:

Table 3–2

| Block | Signal / parameter |
|---|---|
| CtrlV0 | PV_Out |
| CtrlV1 | PV_Out |
| CtrlV2 | PV_Out |
| FbkP0 | PV_In |
| FbkV0 | PV_In |
| FbkV1 | PV_In |
| FbkV2 | PV_In |
| V | MonTiV0Dynamic |
| V | MonTiV1Dynamic |
| V | MonTiV2Dynamic |

The template can be created with this information. The following figure shows the arrangement and the interconnection of the necessary components.

Figure 3-4



The following list comprises the properties of the components.

Table 3–3

| Component | Property | Value |
|---|---|---|
| Output | Signal | COUPLING Ctrl0.PV_Out |
| Output | Signal | COUPLING Ctrl1.PV_Out |
| Output | Signal | COUPLING Ctrl2.PV_Out |
| MUL | Name | {$ChartName}_MUL_V0 |
| | X1 | V\MonTiV0Dynamic |
| MUL | Name | {$ChartName}_MUL_V1 |
| | X1 | V\MonTiV1Dynamic |
| MUL | Name | {$ChartName}_MUL_V2 |
| | X1 | V\MonTiV2Dynamic |
| DriveV1 | Name | {$ChartName}_V0 |
| DriveV1 | Name | {$ChartName}_V1 |
| DriveV1 | Name | {$ChartName}_V2 |
| AND | Name | {$ChartName}_AND |
| Connector | Name | {$ChartName}_V0/Hi |
| Connector | Name | {$ChartName}_V0/Lo |
| Connector | Name | {$ChartName}_V0/Y |
| Connector | Name | {$ChartName}_V1/Hi |
| Connector | Name | {$ChartName}_V1/Lo |
| Connector | Name | {$ChartName}_V1/Y |
| Connector | Name | {$ChartName}_V2/Hi |
| Connector | Name | {$ChartName}_V2/Lo |
| Connector | Name | {$ChartName}_V2/Y |
| Input | Signal | COUPLING FbkV0.PV_In |
| Input | Signal | COUPLING FbkV1.PV_In |
| Input | Signal | COUPLING FbkV2.PV_In |
| Input | Signal | COUPLING FbkP0.PV_In |

Create the remaining templates according to the procedure described above.

**Note**

It is a huge advantage if you give every component which you place on a chart a unique name. By using {$ChartName} this can be easy and convenient when using templates. When generating charts, {$ChartName) is replaced by the plan name. If you place additional components on charts, give each of them a unique name. This makes it easier to interconnect the components beyond the limits of the chart at a later date.

## 3.2 CMT Import

With the aid of templates, you can create the device level with the function "CMT Import”.

### Requirements

To use the function "CMT Import" for automatic model creation, a corresponding XML file is required, exported from the PCS 7 project. The PCS 7 project must also be created with the help of CMTs (control module types).

### Preparation

Create the following subfolders in the folder "Chart".

- 1_SignalLevel
- 2_DeviceLevel
- 3_ProcessLevel

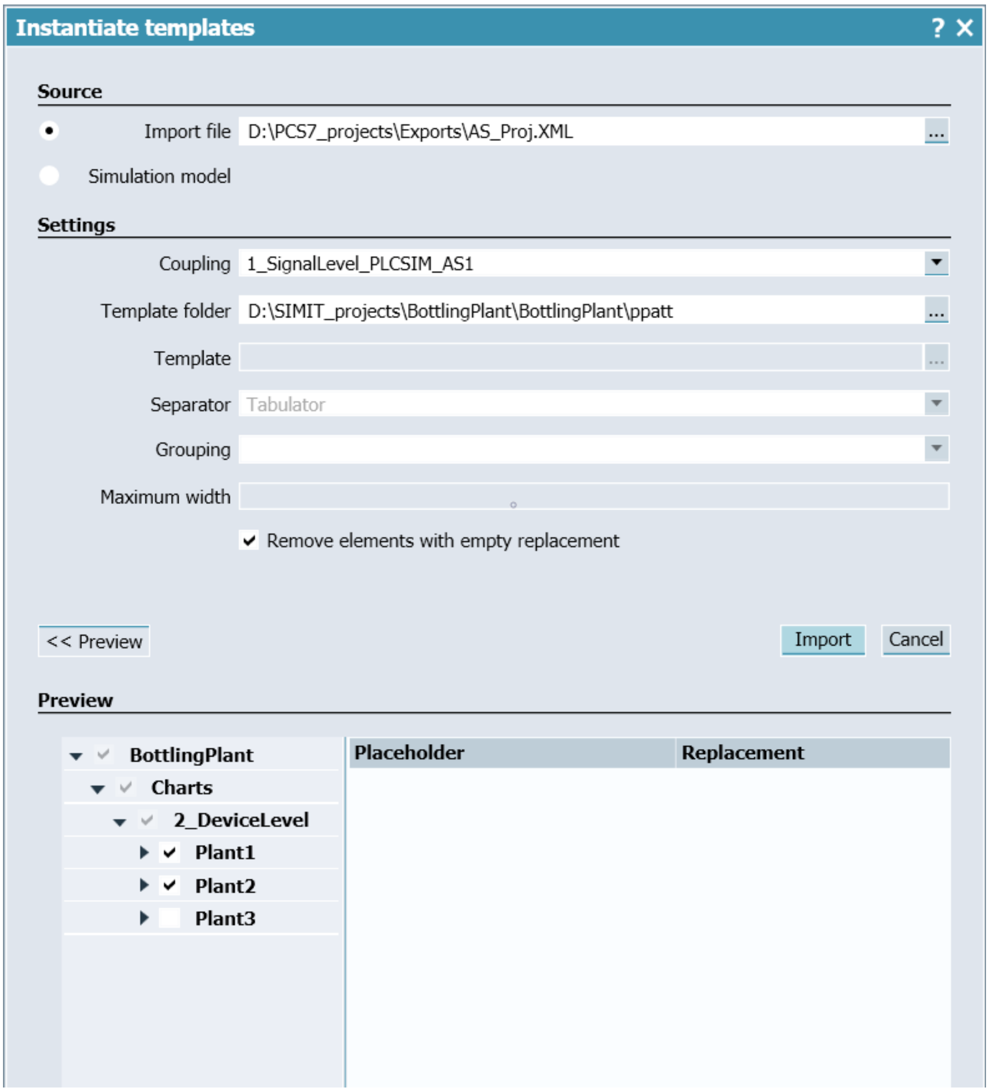
### Procedure

The application example "SIMIT Simulation V11.0 SP1 Getting Started" in chapter 6.3.1 "Exporting an XML file from PCS 7” describe in detail how to export the required XML files and carry out the CMT import.

The CMT import for the application example given here is described step by step in the following section.

1. Right-click the "DeviceLevel" folder.
2. Select "Automatic model generation > Instantiate templates” in the context menu.
3. In the "Coupling” box, select the coupling "1_SignalLevel_PLCSIM_AS1”.
4. Untick the box "Plant3” in the preview.

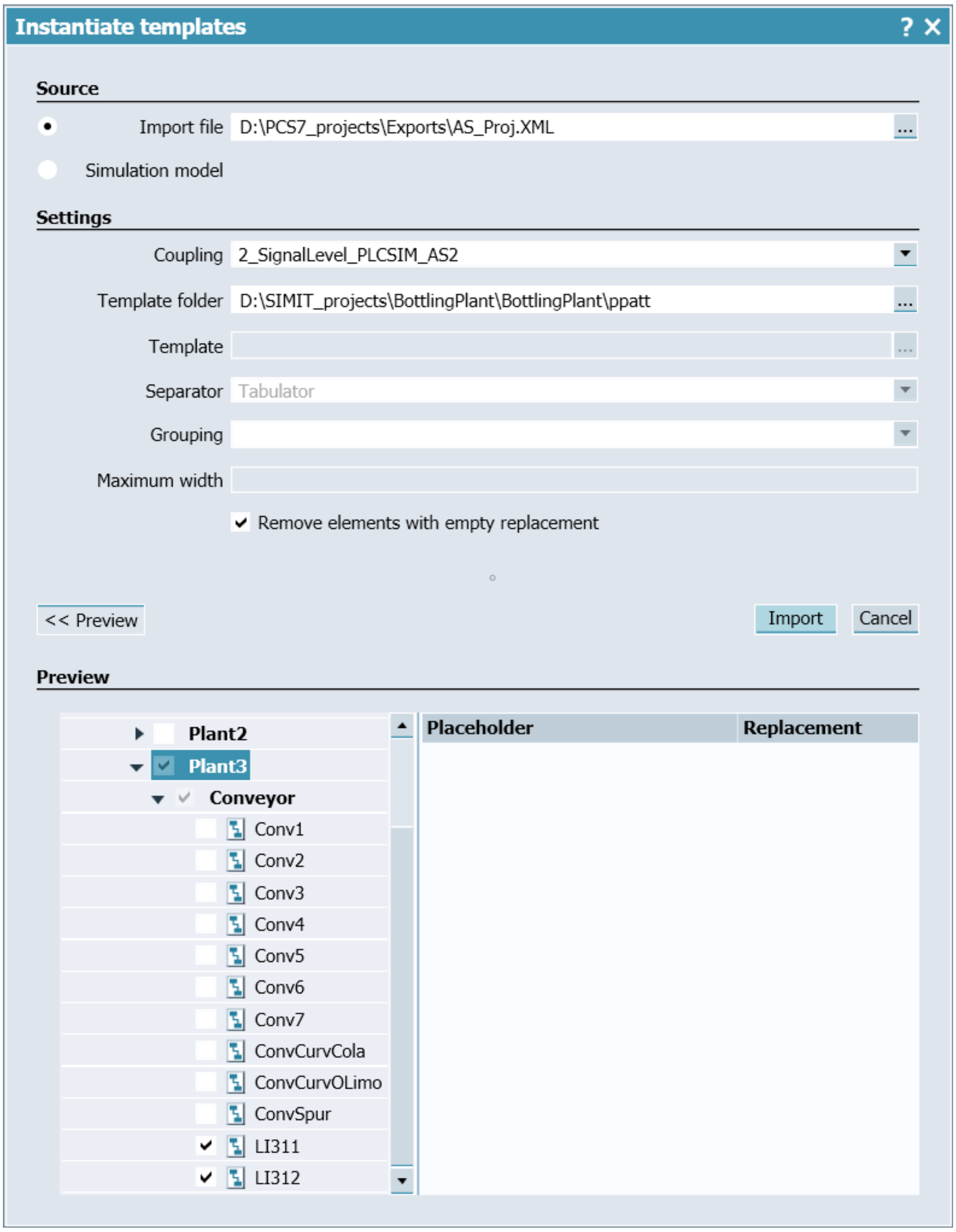
Figure 3-5




5. Click "Import" to start the import.
6. Click "OK" in the window "Import complete".
7. Execute steps 1 to 6 for the coupling "2_SignalLevel_PLCSIM_AS2". Apply your settings as shown in the following figure.

**Note** Only the charts for the fill level measuring points are created. The charts for the conveyor line drives are created with the aid of the function "Generation of device level".

Figure 3-6




### Result

The device level is created in its entirety for the plant sections "Plant1" and "Plant2" For the plant section "Plant3", only the fill level measuring points "LI311" and "LI312" were created.

## 3.3 Generating the Device Level

An additional option for creating a model automatically is the function "Generating the Device Level". When using this function, the device level is created using information from the conveyor system model which is constructed using CONTEC library components.

Templates are required for this, as with the CMT import. The templates for the "Generation of the device level" are created in the same way as the templates for the CMT import.

In the application example given here, the CMT import template "Bottling_MotorRev" is adapted and saved under the name "Conv". Then the conveyor system is replicated using the "Conveyor" components in the CONTEC library.

### 3.3.1 Adapting the Template

The following procedure describes how to adapt the template "Bottling_MotorRev".

1. Create a new template.
2. Name it "Conv".
3. Open the template "Bottling_MotorRev", select all the components contained in the template and copy them.
4. Close the template "Bottling_MotorRev" and switch to the template "Conv."
5. Insert the copied components into the template "Conv".
6. Adapt the available components as described in the table below:

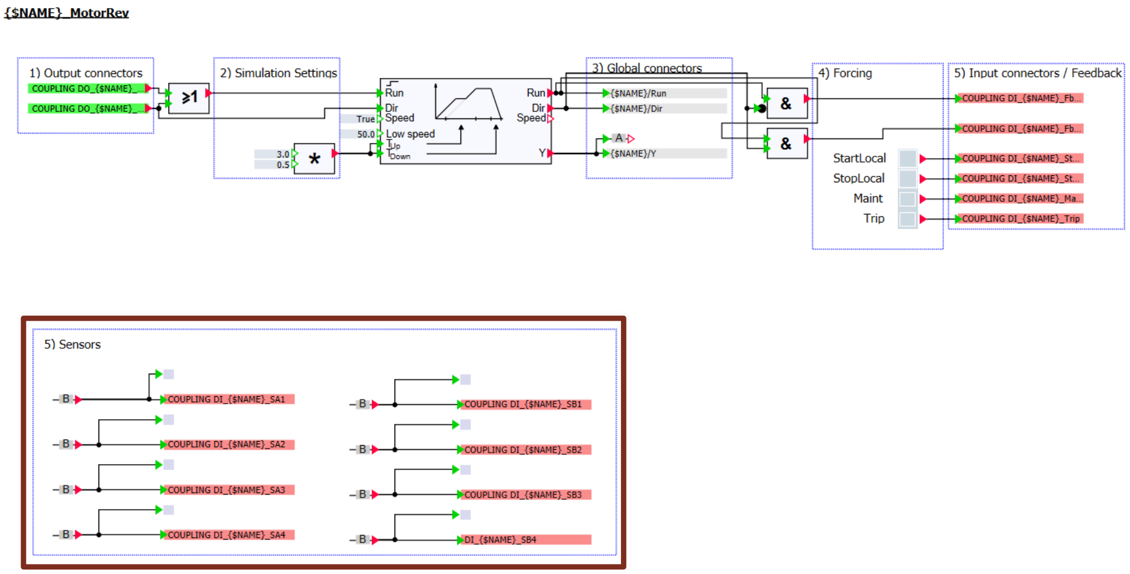
Table 3-4

| Component | Property | Old value | New value |
|---|---|---|---|
| Text | Text | {$ChartName}_MotorRev | {$NAME}_MotorRev |
| Output | Signal | COUPLING Fwd\PV_Out | COUPLING DO_{$NAME}_Fwd |
| Output | Signal | COUPLING REv\PV_Out | COUPLING DO_{$NAME}_Rev |
| OR | Name | OR#1 | OR_{$NAME} |
| MUL | Name | MUL#1 | MUL_{$Name} |
| | Input X1 | U\MonTiDynamic | 3 |
| | Input X2 | 1.0 | 0.5 |
| DriveP1 | Name | DriveP1#1 | {$NAME}_U |
| Connector | Name | Connector#1 | {$NAME}/Run |
| Connector | Name | Connector#2 | {$NAME}/Dir |
| Connector | Name | Connector#3 | {$NAME}/Y |
| NOTc | Name | NOTc#1 | NOTc_{$NAME} |
| AND | Name | AND#1 | AND_Fwd_{$NAME} |
| AND | Name | AND#2 | AND_Rev_{$NAME} |
| Pushbutton | Name | Pushbutton#1 | Pushbutton_{$NAME}_Startlocal |
| Pushbutton | Name | Pushbutton#2 | Pushbutton_{$NAME}_Stoplocal |
| Switch | Name | Switch#1 | Switch_{$NAME}_Maint |
| Switch | Name | Switch#2 | Switch_{$NAME}_Trip |

| Component | Property | Old value | New value |
|---|---|---|---|
| Input | Signal | COUPLING FbkFwd\PV_In | COUPLING DI_{$NAME}_FbkFwd |
| Input | Signal | COUPLING FbkRev\PV_In | COUPLING DI_{$NAME}_FbkRev |
| Input | Signal | COUPLING StartLocal\PV_In | COUPLING DI_{$NAME}_StartLocal |
| Input | Signal | COUPLING StopLocal\PV_In | COUPLING DI_{$NAME}_StopLocal |
| Input | Signal | COUPLING Maint\PV_In | COUPLING DI_{$NAME}_Maint |
| Input | Signal | COUPLING Trip\PV_In | COUPLING DI_{$NAME}_Trip |

7. Insert the components which are additionally required into the template, as shown in the following figure. These are essential for the sensors of the components in the conveyor system.

Figure 3-7




8. Adjust the properties of the components as summarized in the table below.

Table 3–5

| Component | Property | Value |
|---|---|---|
| BConnector | Name | BConnector_{$NAME}-SensorA1 |
| | Input | {$NAME} SensorA1 |
| BConnector | Name | BConnector_{$NAME}-SensorA2 |
| | Input | {$NAME} SensorA2 |
| BConnector | Name | BConnector_{$NAME}-SensorA3 |
| | Input | {$NAME} SensorA3 |

| Component | Property | Value |
|---|---|---|
| BConnector | Name | BConnector_{$NAME}-SensorA4 |
| | Input | {$NAME} SensorA4 |
| Binary Display | Name | Binary display_{$NAME}_SensorA1 |
| Binary Display | Name | Binary display_{$NAME}_SensorA2 |
| Binary Display | Name | Binary display_{$NAME}_SensorA3 |
| Binary Display | Name | Binary display_{$NAME}_SensorA4 |
| Input | Signal | COUPLING DI_{$NAME}_SA1 |
| Input | Signal | COUPLING DI_{$NAME}_SA2 |
| Input | Signal | COUPLING DI_{$NAME}_SA3 |
| Input | Signal | COUPLING DI_{$NAME}_SA4 |
| BConnector | Name | BConnector_{$NAME}-SensorB1 |
| | Input | {$NAME} SensorB1 |
| BConnector | Name | BConnector_{$NAME}-SensorB2 |
| | Input | {$NAME} SensorB2 |
| BConnector | Name | BConnector_{$NAME}-SensorB3 |
| | Input | {$NAME} SensorB3 |
| BConnector | Name | BConnector_{$NAME}-SensorB4 |
| | Input | {$NAME} SensorB4 |
| Binary Display | Name | Binary display_{$NAME}_SensorB1 |
| Binary Display | Name | Binary display_{$NAME}_SensorB2 |
| Binary Display | Name | Binary display_{$NAME}_SensorB3 |
| Binary Display | Name | Binary display_{$NAME}_SensorB4 |
| Input | Signal | COUPLING DI_{$NAME}_SB1 |
| Input | Signal | COUPLING DI_{$NAME}_SB2 |

| Component | Property | Value |
|---|---|---|
| Input | Signal | COUPLING DI_{$NAME}_SB3 |
| Input | Signal | COUPLING DI_{$NAME}_SB4 |
| AConnector | Name | {$NAME}-Speed |

9. Save and close the template "Conv".

### 3.3.2 Conveyor System Simulation Chart

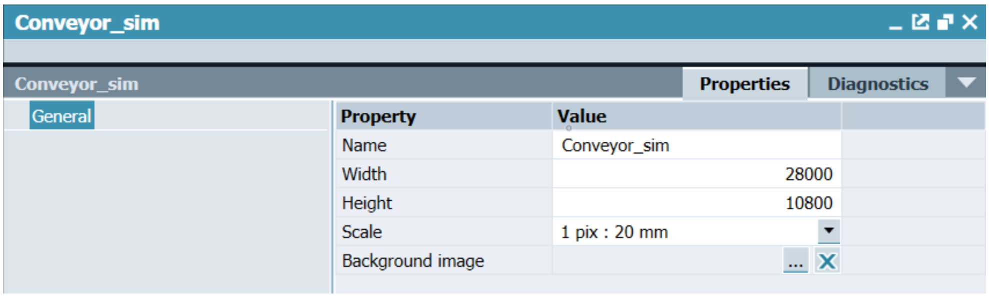
In the application example given here, there is a separate chart for the simulation of the conveyor system. The following will describe how to create this chart.

#### Chart Properties

1. Create the folder "Conveyor" in the folder "3_ProcessLevel".
2. Create a new chart "Conveyor_sim" in this folder.
3. Adapt the size and scale in line with the following figure.

**Note** The dimensions of the components from the CONTEC library play a decisive role in the simulation as the lengths of the conveyor lines and the sensor positions are derived from these. You can find more information in chapter 7.5.2.5 "Scalability" in the manual "SIMATIC SIMIT Simulation Platform (V11)".

Figure 3-8




#### Conveyor System

The conveyor system is created using components from the CONTEC library. The following section will describe how to create the simulation of the conveyor system.

1. Place the components "Conveyor-S4", "ConveyorCurve45-R60" and "SpurConveyor-2" on the chart "Conveyor_sim", as shown in the following figure.

2. Adjust the properties of the components using the table below.

Figure 3-9

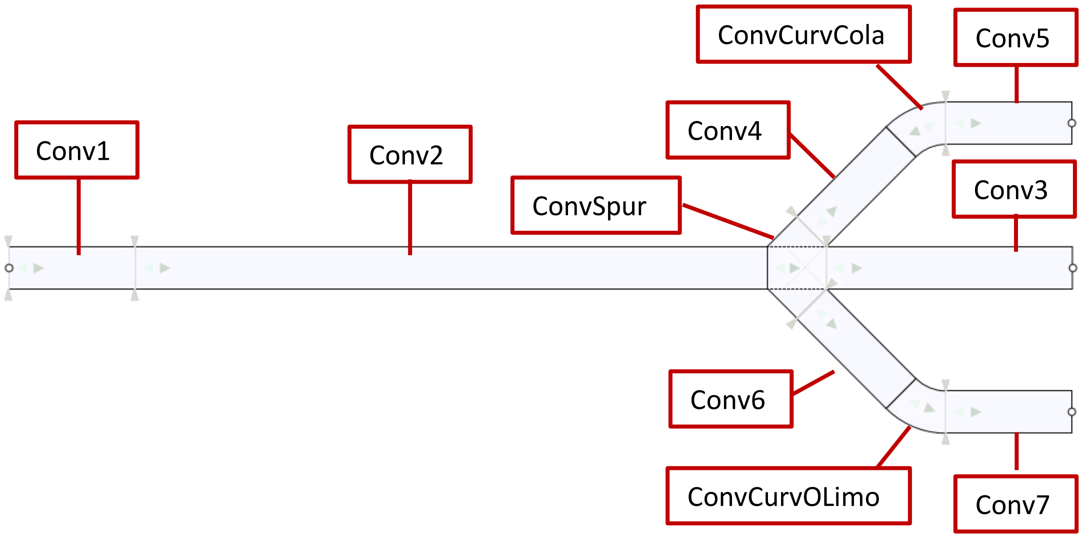



Table 3–6

| Component | Property | Value |
|---|---|---|
| Conv1 | Name | Conv1 |
| | SensorPositionA1 | 1200 |
| | MatarialType | CBoxDS256 |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |
| Conv2 | Name | Conv2 |
| | Width | 12000 |
| | NominalSpeed | 0.5 |
| | NbrOfSensorsA | 3 |
| | SensorPositionA1 | 2000 |
| | SensorPositionA2 | 6000 |
| | SensorPositionA3 | 10000 |
| ConvSpur | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |
| Conv3 | Name | Conv3 |
| | Width | 7200 |
| | RemoveA3 | Conv3-SensorA3 OUT |
| | NbrOfSensorsA | 3 |
| | SensorPositionA1 | 1000 |
| | SensorPositionA2 | 6000 |
| | SensorPositionA3 | 7000 |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |

| Component | Property | Value |
|---|---|---|
| Conv4 | Name | Conv4 |
| | SensorPositionA1 | 1000 |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |
| Conv6 | Name | Conv6 |
| | SensorPositionA1 | 1000 |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |
| ConvCurvCola | Name | ConvCurvCola |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |
| ConvCurvOLimo | Name | ConvCurvOLimo |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |
| Conv5 | Name | Conv5 |
| | Width | 5000 |
| | RemoveA2 | Conv5-SensorA2 OUT |
| | NbrOfSensorsA | 3 |
| | SensorPositionA1 | 4000 |
| | SensorPositionA2 | 4900 |
| | SensorPositionA3 | 10 |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |
| Conv7 | Name | Conv7 |
| | Width | 5000 |
| | RemoveA2 | Conv7-SensorA2 OUT |
| | NbrOfSensorsA | 2 |
| | SensorPositionA1 | 4000 |
| | SensorPositionA2 | 4800 |
| | TEMPLATE | Conv |
| | HIERARCHY | Conveyor |

### 3.3.3 Procedure for "Generating the Device Level”

The following section will describe the procedure for the function "Generating the Device Level”.

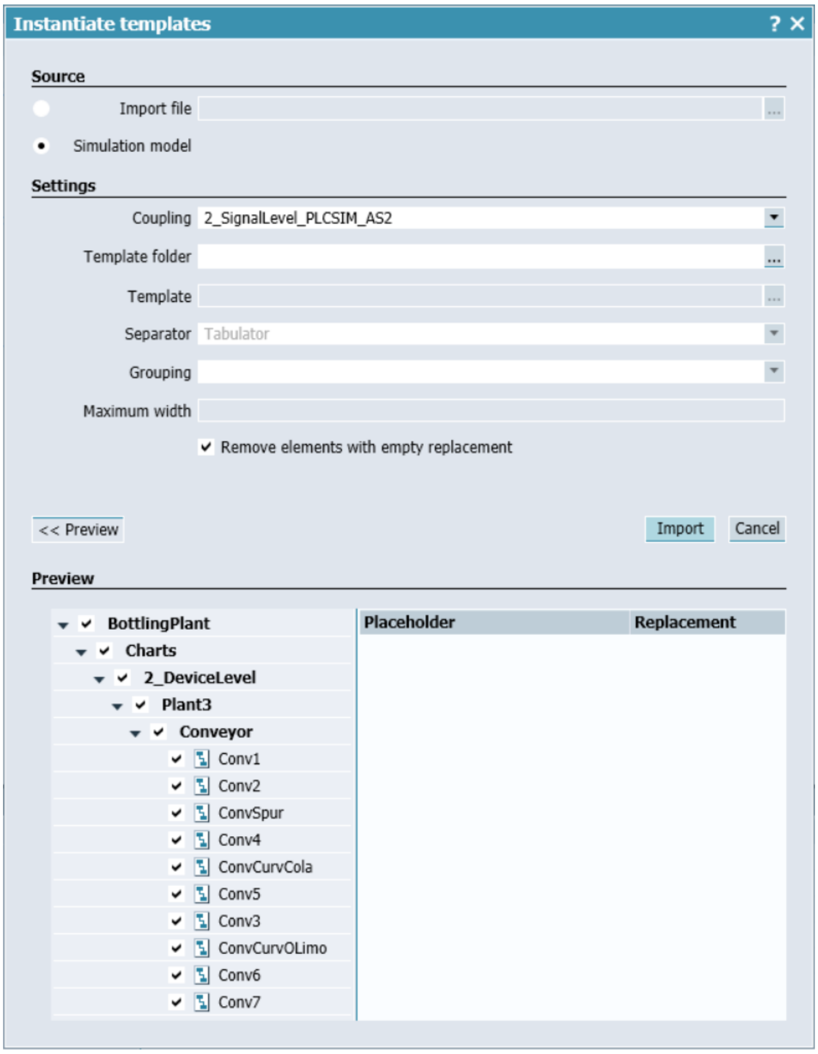
#### Requirement

The requirement for using the function "Generating the Device Level” is that you have created the relevant template (if you are not using the base templates) and you have created the model of the conveyor system with the components from the CONTEC library.

## Procedure

1. Right-click "Plant3" in the "DeviceLevel" folder.
2. Select "Automatic model generation > Instantiate template" in the context menu.
3. Select "Simulation model" as source
4. Set the coupling "2_SignalLevel_PLCSIM_AS2" in the "Coupling" box.

Figure 3-10




5. Click "Import".
6. Click "OK" in the window "Import complete".

### 3.3.4 Final Adjustments/Additions

Adjustments and/or additions have to be made to the charts "ConvSpur" and "ConvCurvCola" because the template used does not meet all requirements.

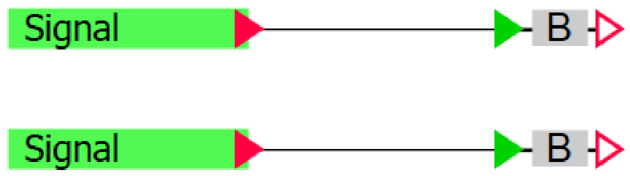
**Note**

> If you have several charts of the same type which need to be adapted after generating the device level, it is sensible to have a separate template for these charts.

**ConvSpur**

1. Open the chart ConvSpur.
2. Insert two components from each of the types "Output" and "BConnector".
3. Interconnect these as shown in the following figure.

Figure 3-11




Adjust the properties of the components as summarized in the table below.

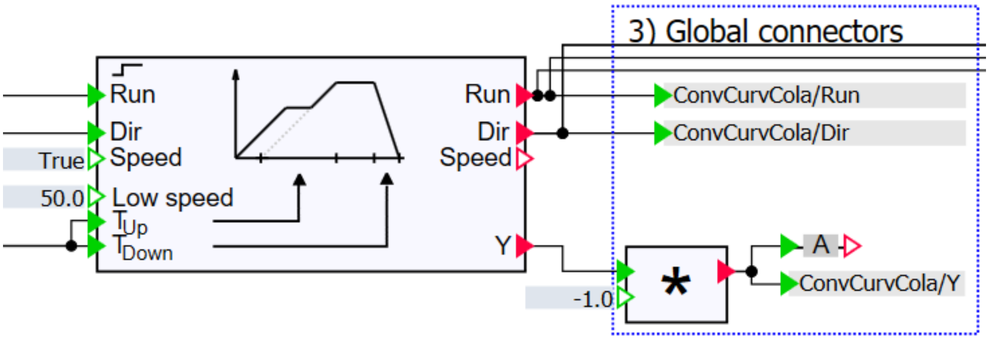
Table 3–7

| Component | Property | Value |
|---|---|---|
| Output | Signal | 2_SignalLevel_PLCSIM_AS2DO_ConvSpure_S_AB |
| Output | Signal | 2_SignalLevel_PLCSIM_AS2DO_ConvSpure_S_AD |
| BConnector | Name | ConvSpur-Switch1 |
| BConnector | Name | ConvSpur-Switch2 |

**ConvCurvCola**

1. Open the chart "ConvCurvCola".
2. Insert the component "MUL".
3. Interconnect these as shown in the following figure.

Figure 3-12




4. Adjust the properties of the components as summarized in the table below.

Table 3–8

| Component | Property | Value |
|---|---|---|
| MUL | Name | MUL_Speed_ConvCurvCola |
| | X2 | -1.0 |

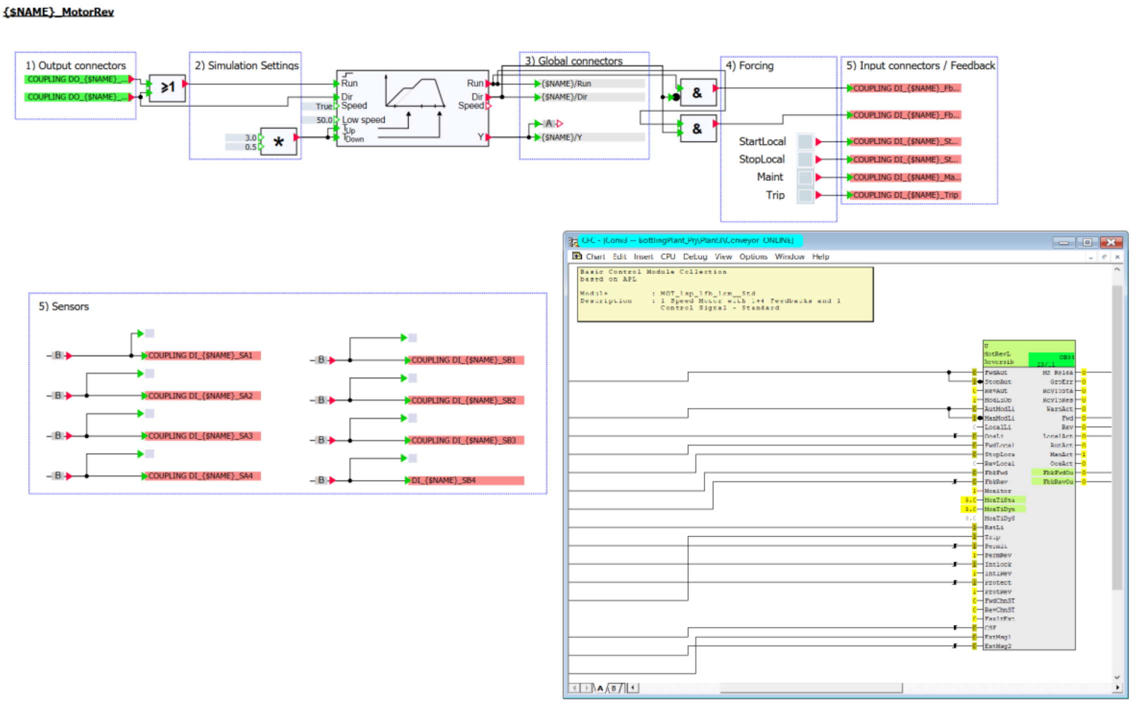
**Result**

The device level is created in its entirety for the plant section "Plant3”.

## 3.4 The Current Status of your Project

You have reproduced the device level of the PCS 7 project in SIMIT in full and can now execute the measuring point test.

Figure 3-13



# 4 Modeling Physical Correlations

The following section will describe how to reproduce the models of physical correlations in the PCS 7 project "Bottling Plant" in SIMIT, using components from the standard library, the FLOWNET library and a component created with the CTE tool.

## 4.1 Establishing the Physical Process Model

The basic procedure for establishing a physical process model is described in the application example "SIMIT simulation of a stirred tank reactor with PCS 7" chapter 4 "Fundamentals - Process Technology".

## 4.2 Model Plant Section "Plant1"

The following section will describe the development of the process model for the plant section "Plant1". The plant section "Plant1" describes raw material tanks and injection into the reactors of plant section "Plant2". The plant section consists of three raw material tanks. The modeling is carried out for raw material tanks 1 and 3 with components from the standard library. Raw material tank 2 is created using components from the FLOWNET library. Macros are created for calculations which occur repeatedly.

### 4.2.1 Creating Macros

The following section will describe the procedure for creating the required macros.

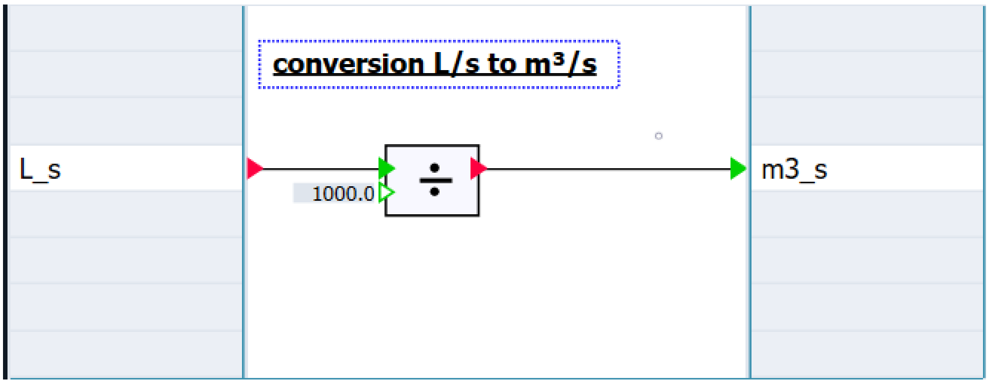
**Macro "LiterSTom3s"**

The macro "LiterSTom3s" converts the volume flow from liters per second to m³ per second. The procedure for creating this is described below step-by-step.

1. Switch to the Task-Card "Macros" in the Tools window.
2. Create a new macro.
3. Rename this macro "LiterSTom3s".
4. Open the macro.
5. Open the properties of the macro.
6. Assign the property "Code" the value "LiterSTom3s".
7. Insert the component "DIV" from the standard library into the macro.
8. Assign it a suitable name (e.g. DIV_convert_Liter_s_to_m3_s).
9. Assign the input X2 the value 1000,0.
10. Connect the input "X1" of the "DIV" component to the sidebar for inputs.
11. Rename the input "L_s".
12. Connect the input "Y" of the "DIV" component to the sidebar for outputs.

13. Rename the output "m3_s".

Figure 4-1

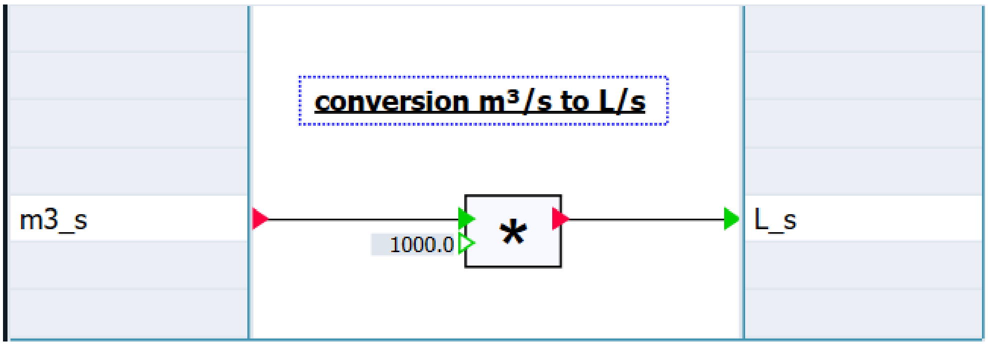



## Macro "m3sToLiterS"

The macro "m3sToLiterS" converts the volume flow from m³ per second to liters per second. The procedure for creating the macro is the same as the procedure for creating the macro "LiterSTom3s".

The following figure shows the components and their connection to the macro.

Figure 4-2




Adjust the properties of the macro and components as summarized in the table below.

Table 4–1

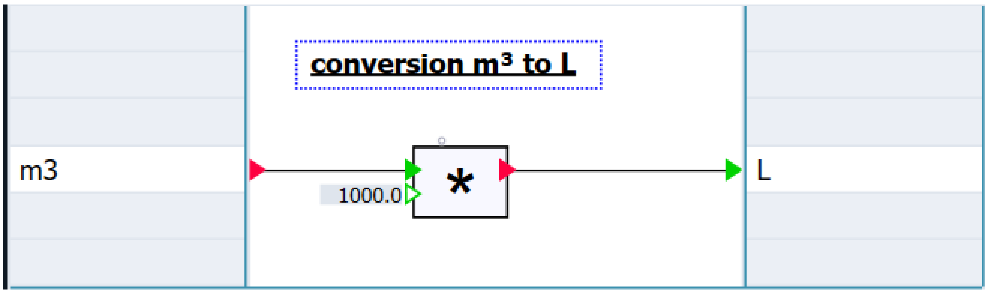
| Component | Property | Value |
|---|---|---|
| Macro | Code | m3sToLiterS |
| MUL | Name | MUL_convert_m3_s_to_Liter_s |
| | X1 | Connection to m3_s |
| | X2 | 1000 |
| | Y | Connection to L_s |

**Macro "m3ToLiter"**

The macro "m3ToLiter" converts the volume from m³ to liters. The procedure for creating the macro is the same as the procedure for creating the macro "LiterSTom3s".

The following figure shows the components and their connection to the macro.

Figure 4-3




Adjust the properties of the macro and components as summarized in the table below.

Table 4–2

| Component | Property | Value |
|---|---|---|
| Macro | Code | m3ToLiter |
| MUL | Name | MUL_convert_m3_to_Liter |
| | X1 | Connection to m3 |
| | X2 | 1000 |
| | Y | Connection to L |

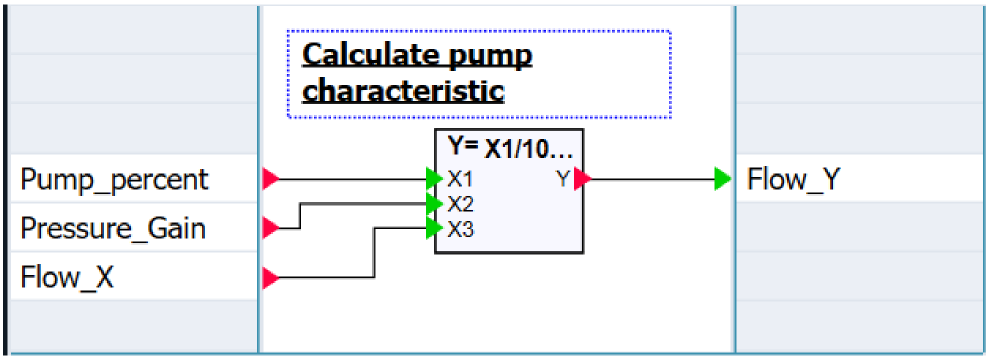


**Macro "PumpPress"**

The macro "PumpPress" simulates the characteristic of a pump. The procedure for creating the macro is the same as the procedure for creating the macro "LiterSTom3s".

The following figure shows the components and their connection to the macro.

Figure 4-4



Adjust the properties of the macro and components as summarized in the table below.

Table 4–3

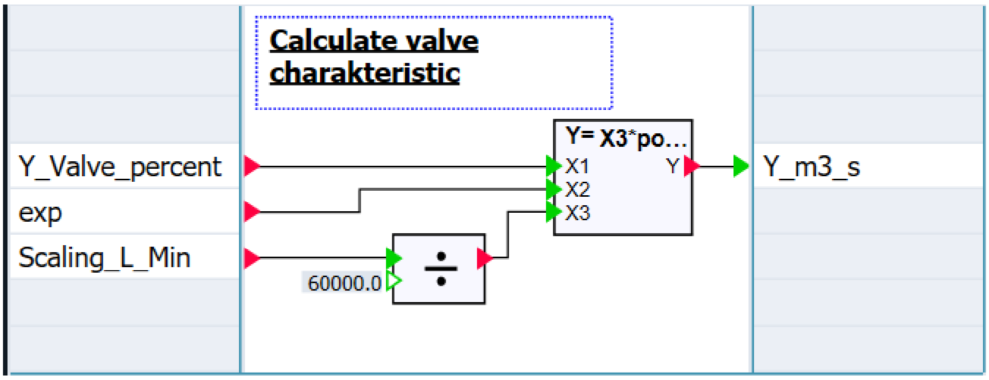
| Component | Property | Value |
|---|---|---|
| Macro | Code | PumpPress |
| AFormula | Name | AFormula_PumpPress |
| | X1 | Connection to Pump_percent |
| | X2 | Connection to Pressure_Gain |
| | X3 | Connection to Flow_X |
| | Y | Connection to Flow_X |
| | Formula | X1/100*X2*X3 |

**Macro "ValveCurve”**

The macro "ValveCurve" simulates the equal percentage characteristic of a valve. The procedure for creating the macro is the same as the procedure for creating the macro "LiterSTom3s”.

The following figure shows the components and their connection to the macro.

Figure 4-5



Adjust the properties of the macro and components as summarized in the table below.

Table 4–4

| Component | Property | Value |
|---|---|---|
| Macro | Code | ValveCurve |
| AFormula | Name | AFormula_ ValveCurve |
| | X1 | Connection to Y_Valve_percent |
| | X2 | Connection to exp |
| | Y | Connection to Y_m3_s |
| | Formula | X3*pow(X1/100,X2) |
| DIV | DIV_ValveCurve | Connection to Scaling_L_Min |
| | X2 | 60000,0 |

### 4.2.2 Additional Preparations

Besides the actual models, components are also essential for the simulation of the physical correlations as they can be used to set specific states. Separate charts are created for these.

1. Create the folder "Misc” (from "miscellaneous”) in the folder "Charts”.
2. Create the chart "Connections” in the folder "Misc”.

Components and their connections which are used in this chart are described in the following section.

### 4.2.3 Creating the Chart "RMT1”

The purpose of the chart is to simulate the physical model of the raw material tank "RMT1”.

In the application example given here, the capacity of the raw material tank is simulated according to the following formula:

$$V = \int \dot{V}_{in} + \dot{V}_{out} \, dt$$

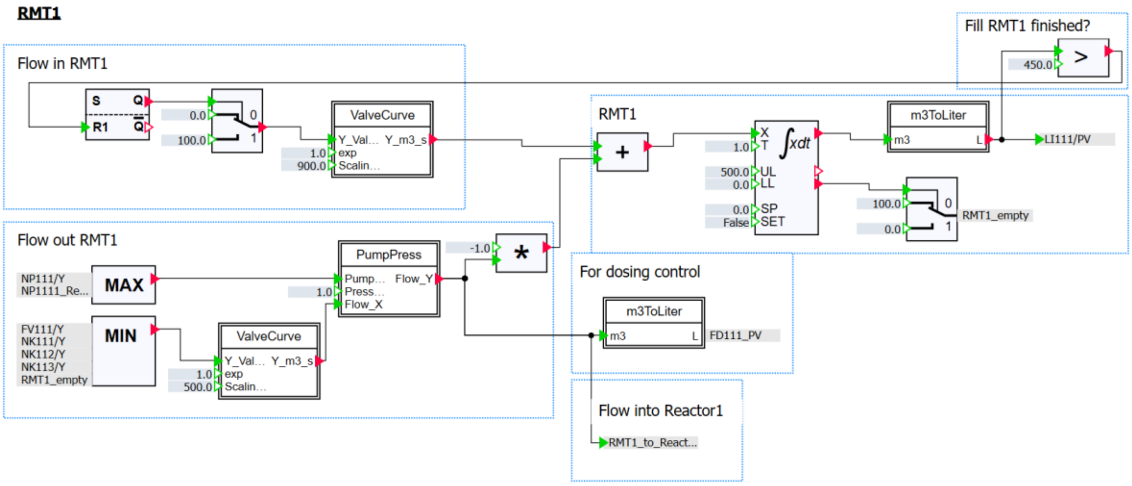
**Procedure**

The following section will describe the creation procedure.

1. Create a new folder in the folder "3_ProcessLevel”.
2. Rename it "Plant1”.
3. Create a new chart in the folder "Plant1”.
4. Rename it "RMT1”.
5. Insert the relevant components into the chart (see the figure below).

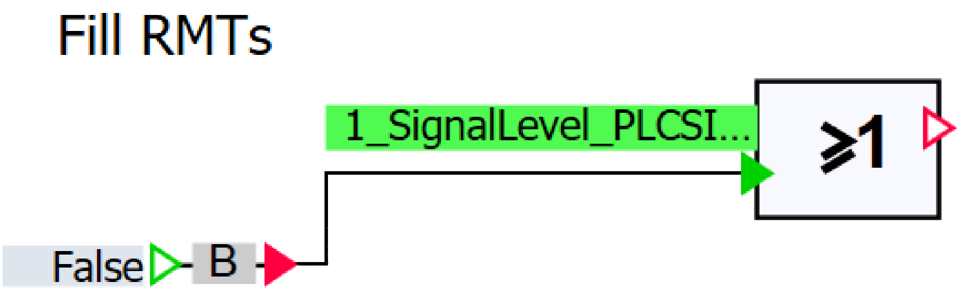
6. Interconnect the components as shown in the following figure.

Figure 4-6




7. Open the chart "Connections".
8. Insert the components shown in the following figure into the chart.
9. Interconnect the components as shown in the following figure.

Figure 4-7




10. Adjust the properties of the components as summarized in the table below.

Table 4–5

| Component | Property | Value |
|---|---|---|
| BConnector | Name | BConnector_Fill_RMTs |
| Output | Signal | 1_SignalLevel_PLCSIM_AS1 DO_RMT1_Filling |
| OR | Name | OR_Fill_RMT1 |

11. Save and close the chart "Connections".
12. Open the chart "RMT1".
13. Adjust the properties of the components as summarized in the table below.

Table 4–6 Flow in RMT1

| Component | Property | Value |
|---|---|---|
| RS_FF | Name | RS_FF_Fill_RMT1 |
| | S | OR_Fill_RMT1 OUT |
| Selection | Name | Selection_Fill_RMT1 |
| | X1 | 100 |
| ValveCurve | Name | ValveCurve_Fill_RMT1 |
| | exp | 1 |
| | Scaling_L_Min | 900 |

Table 4–7 Flow out RMT1

| Component | Property | Value |
|---|---|---|
| Connector | Name | NP111/Y |
| Connector | Name | NP1111_Red/Y |
| Connector | Name | FV111/Y |
| Connector | Name | NK111/Y |
| Connector | Name | NK112/Y |
| Connector | Name | NK113/Y |
| Connector | Name | RMT1_empty |
| MinMax | Name | MinMax_Pumps_RMT1 |
| | MinMax | MAX |
| MinMax | Name | MinMax_Valves_RMT1 |
| | MinMax | MIN |
| ValveCurve | Name | ValveCurve_RMT1 |
| | exp | 1.0 |
| | Scaling_L_Min | 500 |
| PumpPress | Name | PumpPress_RTM1 |
| | Press_Gain | 1.0 |
| MUL | Name | MUL_RMT1_Out |
| | X1 | -1.0 |

Table 4–8 RMT1

| Component | Property | Value |
|---|---|---|
| ADD | Name | ADD_sum_flow_RMT1 |
| INT | Name | INT_RMT1 |
| | UL | 500.0 |
| m3ToLiter | Name | m3ToLiter_RMT1 |
| Selection | Name | Selection_RMT1_empty |
| | X1 | 100 |
| Connector | Name | LI111/PV |
| Connector | Name | RMT1_empty |

Table 4–9 Fill RMT1 finished?, For Dosing Control, Flow into Reactor1

| Component | Property | Value |
|---|---|---|
| m3ToLiter | Name | m3ToLiter_FD111/PV |
| Connector | Name | FD111_PV |
| Connector | Name | RMT1_to_Reactor_1 |
| Compare | Name | Compare_Fill_RMT1_finished |
| | X2 | 450.0 |
| | Comparison | > |

14. Save and close the chart "RMT1”.

### 4.2.4 Creating the Chart "RMT2”

The purpose of the chart is to simulate the physical model of the raw material tank "RMT2”.

In the application example given here, the capacity of the raw material tank is simulated according to the following formula:

$$V = \int \dot{V}_{in} + \dot{V}_{out} \, dt$$

The chart "RMT2” is created using components from the FLOWNET library. The FLOWNET library contains components for creating the simulation of piping networks. You can find a detailed description of piping networks and components in the FLOWNET library in chapter 8.2 "CHEM-BASIC and FLOWNET libraries” in the manual "SIMATIC SIMIT Simulation Platform (V11)".



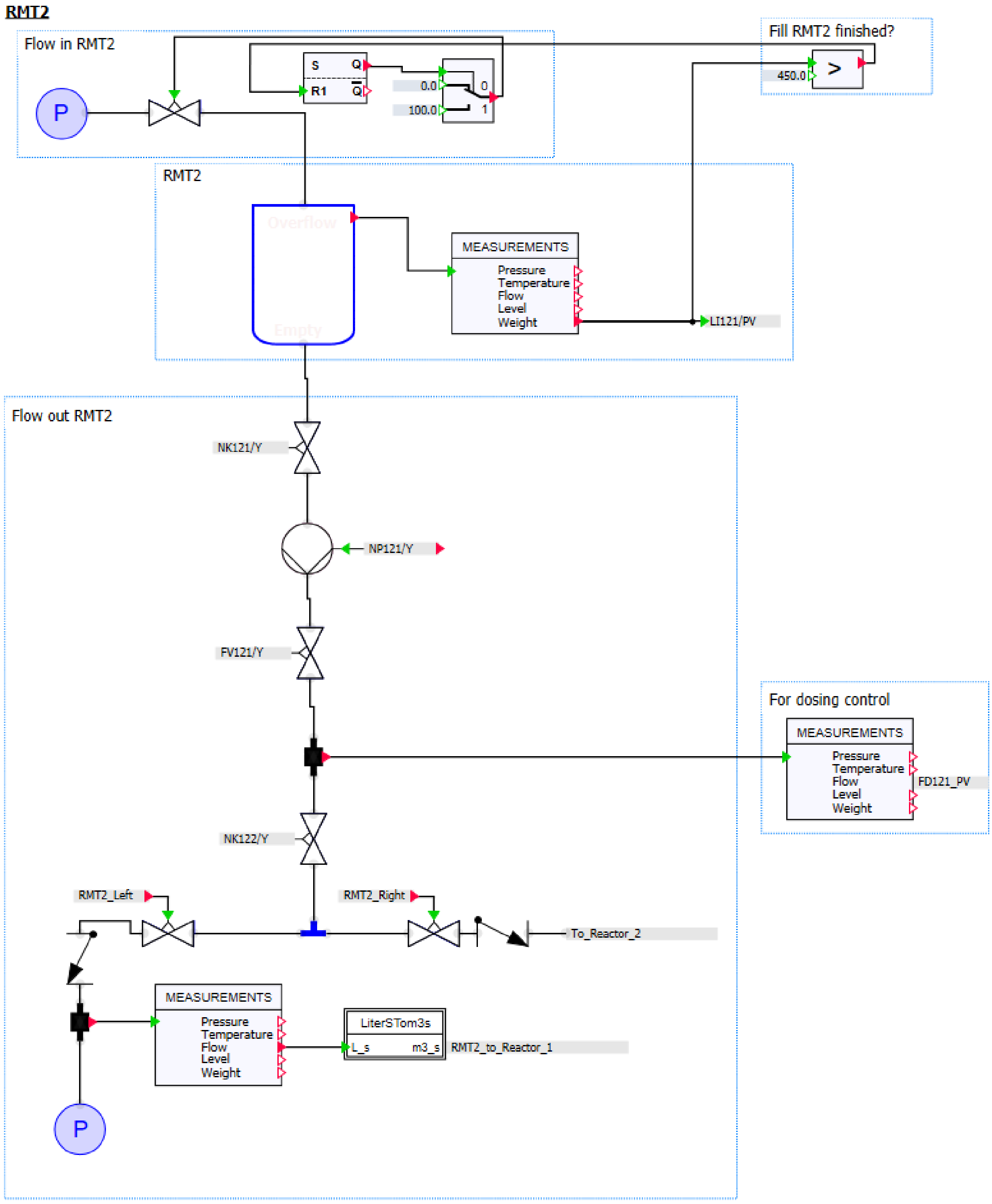
**Procedure**

The following section will describe the creation procedure.

1. Create a new chart in the folder "Plant1”.
2. Rename it "RMT2”.
3. Insert the relevant components into the chart (see the figure below).

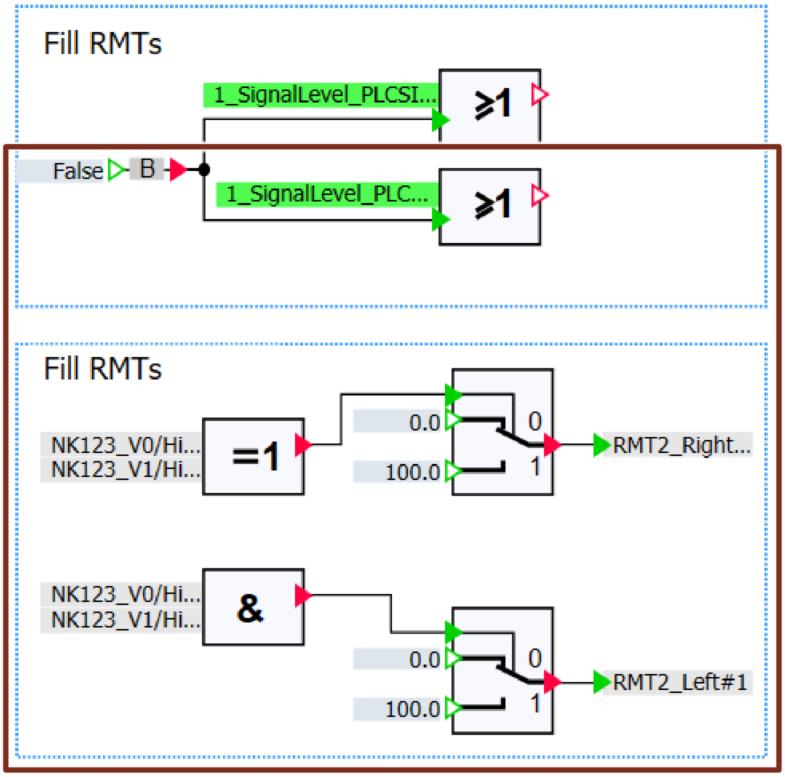
4. Interconnect the components as shown in the following figure.

Figure 4-8




5. Open the chart "Connections".
6. Insert the components with a red frame in the figure into the chart.
7. Interconnect the components as shown in the following figure.

Figure 4-9




8. Adjust the properties of the components as summarized in the table below.

Table 4–10

| Component | Property | Value |
|---|---|---|
| Output | Signal | 1_SignalLevel_PLCSIM_AS1 DO_RMT2_Filling |
| 2x Connector | Name | NK123_V0/Hi |
| 2x Connector | Name | NK123_V1/Hi |
| XOR | Name | XOR_RMT2_Left |
| AND | Name | AND_RMT2_Right |
| Selection | Name | Selection_RMT2_Left |
| | X1 | 100.0 |
| Selection | Name | Selection_RMT2_Right |
| | X1 | 100.0 |
| Connector | Name | RMT2_Left |
| Connector | Name | RMT2_Right |

9. Save and close the chart "Connections".
10. Open the chart "RMT2".
11. Adjust the properties of the components as summarized in the table below.

Table 4–11 Flow in RMT2

| Component | Property | Value |
|---|---|---|
| PnodeLiquid | Name | PnodeLiquid_In_RMT2 |
| | Pressure | 4.0 |
| Valve | Name | Valve_Fill_RMT2 |
| | Cvs | 55.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| RS_FF | Name | RS_FF_Fill_RMT2 |
| | S | OR_Fill_RMT2 OUT |
| Selection | Name | Selection_Fill_RMT2 |
| | X1 | 100.0 |

Table 4–12 RMT2

| Component | Property | Value |
|---|---|---|
| StorageTankLiquid | Name | StorageTankLiquid_RMT2 |
| | Volume | 0.5 |
| | Height | 5.0 |
| | LevelInit | 0.0 |
| Measurements | Name | Measurements_LI121 |
| Connector | Name | LI121/PV |



Table 4–13 Flow out RMT2

| Component | Property | Value |
|---|---|---|
| Valve | Name | Valve_NK121 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Pump | Name | Pump_NP121 |
| | NominalPressure | 4.5 |
| | NominalMassflow | 3 |
| | ShowFlow | True |
| Connector | Name | NP121/Y |
| Valve | Name | Valve_FV121 |
| | Cvs | 30.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | FV121/Y |
| PipeMeasure | Name | PipeMeasure_Dosing_control_FD121 |
| Valve | Name | Valve_NK122 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | NK122/Y |
| JointLiquid | Name | JointLiquid_RMT2_Flow_out |

| Component | Property | Value |
|---|---|---|
| Valve | Name | Valve_RMT2_Right |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | RMT2_Right |
| Topology | Name | To_Reactor_2 |
| Valve | Name | Valve_RMT2_Left |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| PipeMeasure | Name | PipeMeasure_Flow_into_Reacor1 |
| Measurements | Name | Measurements_Flow_into_Reactor |
| LiterSTom3s | Name | LiterSTom3s_Flow_into_Reactor1 |
| Connector | Name | RMT2_to_Reactor_1 |
| PnodeLiquid | Name | PnodeLiquid_Flow_out_RMT2 |

Table 4–14 Fill RMT2 finished?, for dosing control

| Component | Property | Value |
|---|---|---|
| Compare | Name | Compare_Fill_RMT2_finished |
| | X2 | 450.0 |
| | Comparison | > |
| Measurements | Name | Measurements_FD121 |
| Connector | Name | FD121_PV |



12. Save and close the chart "RMT2”.

### 4.2.5 Creating the Chart "RMT3”

The purpose of the chart is to simulate the physical model of the raw material tank "RMT3”.

In the application example given here, the capacity of the raw material tank is simulated according to the following formula:

$$V = \int \dot{V}_{in} + \dot{V}_{out}\, dt$$

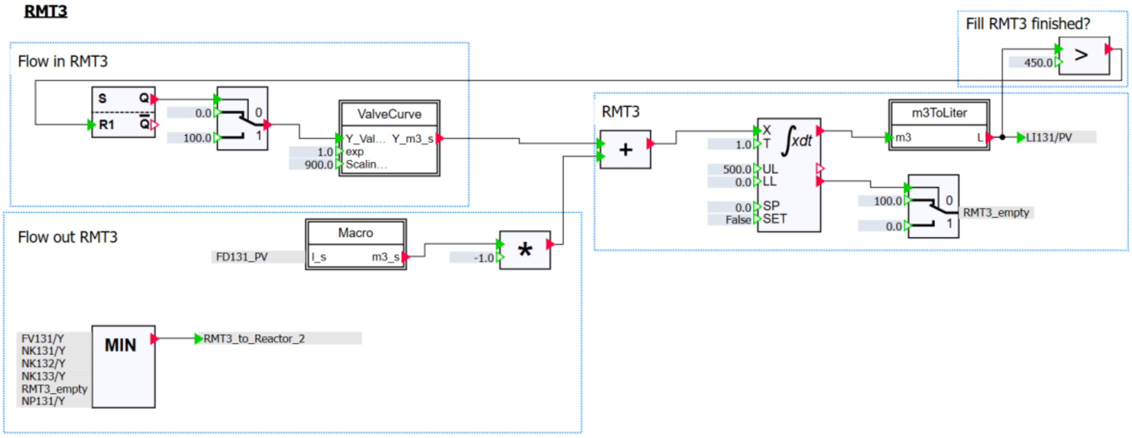
**Procedure**

The following section will describe the creation procedure.

1. Create a new chart in the folder "Plant1”.
2. Rename it "RMT3”.
3. Open the chart "RMT1”.
4. Select all of the components and copy them.
5. Close the chart "RMT1”
6. Insert the copied components into the chart "RMT3”.

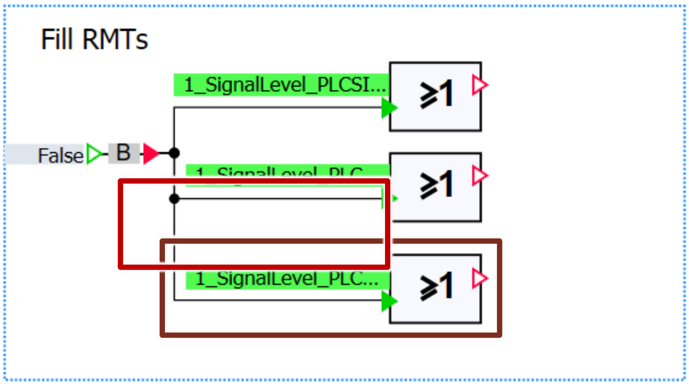
7. Remove the following components:
   - Connector "NP1111_Red/Y"
   - MinMax "MinMax_Pumps_RMT1"
   - PumpPress "PumpPress_RTM1"
   - ValveCurve "ValveCurve_RMT1"
   - m3ToLiter "m3ToLiter_FD111/PV"
8. Connect the connector "NP111/Y" to the component "MinMax_Valves_RMT3", as shown in the following figure.
9. Insert the macro "LiterSTom3s" and connect it as shown in the following figure.

Figure 4-10




10. Open the chart "Connections".
11. Insert the components with a red frame in the figure into the chart.
12. Interconnect the components as shown in the following figure.

Figure 4-11




13. Adjust the properties of the components as summarized in the table below.

Table 4–15

| Component | Property | Value |
|---|---|---|
| Output | Signal | 1_SignalLevel_PLCSIM_AS1DO_RMT3_Filling |
| OR | Name | OR_Fill_RMT3 |

14. Save and close the chart "Connections".
15. Open the chart "RMT3".
16. Adjust the properties of the components as summarized in the table below.

Table 4–16 Flow in RMT3

| Component | Property | Old value | New value |
|---|---|---|---|
| RS_FF | Name | RS_FF_Fill_RMT1 | RS_FF_Fill_RMT3 |
| | S | OR_Fill_RMT1 | OR_Fill_RMT3 |
| Selection | Name | Selection_Fill_RMT1 | Selection_Fill_RMT3 |
| ValveCurve | Name | ValveCurve_Fill_RMT1 | ValveCurve_Fill_RMT3 |

Table 4–17 Flow out RMT3

| Component | Property | Old value | New value |
|---|---|---|---|
| Connector | Name | FV111/Y | FV131/ |
| Connector | Name | NK111/Y | NK131/ |
| Connector | Name | NK111/Y | NK132/ |
| Connector | Name | NK111/Y | NK133/ |
| Connector | Name | RMT1_empty | RMT3_empty |
| MinMax | Name | MinMax_Valves_RMT1 | MinMax_Valves_RMT3 |
| Connector | Name | NP111/Y | NP131/Y |
| MUL | Name | MUL_RMT1_Out | MUL_RMT3_Out |
| Connector | Name | FD111_PV | FD131_PV |
| LiterSTom3s | Name | LiterSTom3#1 | LiterSTom3s_RTM3 |
| Connector | Name | RMT1_to_Reactor_1 | RMT3_to_Reactor_2 |

Table 4–18 RMT3

| Component | Property | Old value | New value |
|---|---|---|---|
| ADD | Name | ADD_sum_flow_RMT1 | ADD_sum_flow_RMT3 |
| INT | Name | INT_RMT1 | INT_RMT3 |
| m3ToLiter | Name | m3ToLiter_RMT1 | m3ToLiter_RMT3 |
| Selection | Name | Selection_RMT1_empty | Selection_RMT3_empty |
| Connector | Name | LI111/PV | LI131/PV |
| Compare | Name | Compare_Fill_RMT1_finished | Compare_Fill_RMT3_finished |

## 4.3 Model Plant Section "Plant2”

The following section will describe the development of the process model for the plant section "Plant2”. The plant section "Plant2” describes the reactors and dosage into the reactors of plant section "Plant3”. The plant section consists of two reactors. Modeling for "Reactor1” takes place with a reactor component created with the CTE tool and with components from the standard library. "Reactor2” is created with components of the FLOWNET library. Macros created previously are used for calculations which occur repeatedly.

### 4.3.1 Preparation

The reactor component "StirredTankReactor” must be created with the CTE tool first of all, to be able create the chart for the "Reactor1” process model in SIMIT.

You can find the model equations for the component "StirredTankReactor” in the application example "SIMIT simulation of a stirred tank reactor with PCS 7" chapter 4.1 "Process simulation”. In the application example given here, these have been adapted and simplified.
You can find information regarding the creation of your own components and the syntax in the CTE tool in the "SIMIT - Component Type Editor" manual.

You can open the component "StirredTankReactor” with the CTE tool, look at the source code and, if required, adapt it.



### 4.3.2 Creating the Chart "Reactor1”

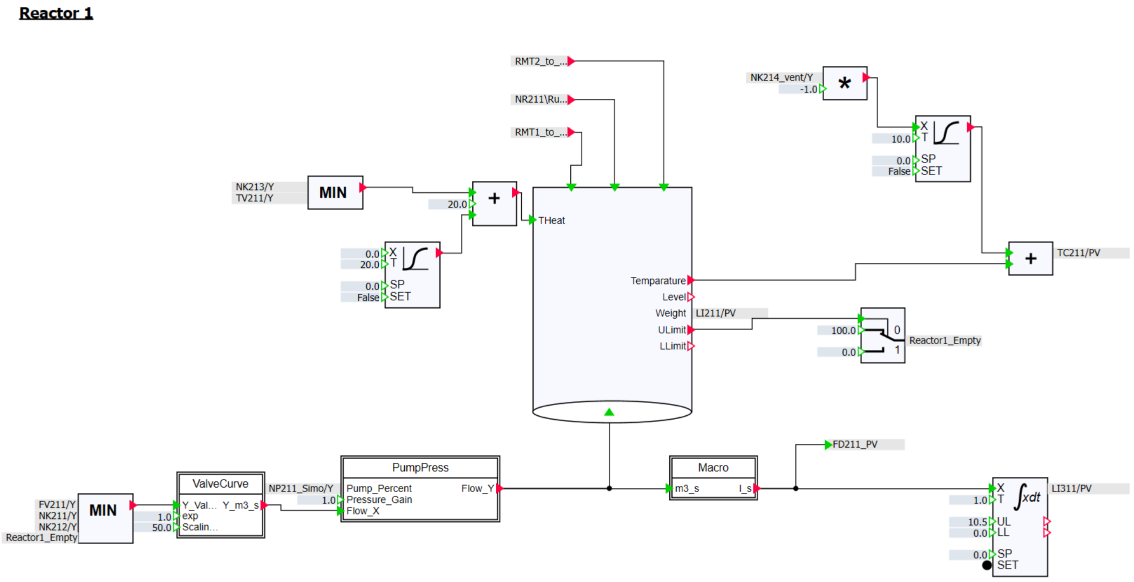
The purpose of the chart is to simulate the physical model of the reactor "Reactor1”.

**Procedure**

1. Create the folder "Plant2” in the folder "3_ProcessLevel”.
2. Create a new chart in the folder "Plant2”.
3. Rename it "Reactor1”.
4. Insert the relevant components into the chart (see the figure below).

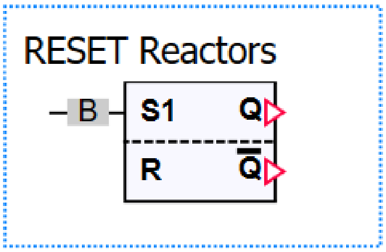
5. Interconnect the components as shown in the following figure.

Figure 4-12




6. Open the chart "Connections”.
7. Insert the components shown in the following figure into the chart.
8. Interconnect the components as shown in the following figure.

Figure 4-13




9. Adjust the properties of the components as summarized in the table below.

Table 4–19

| Component | Property | Value |
|---|---|---|
| BConnector | Name | Bconnector_RESET_Reactors |
| SR_FF | Name | SR_FF_RESET_Reactors |

1. Close the chart "Connections”.
2. Open the chart "Reactor1”.
3. Adjust the properties of the components as summarized in the table below.

Table 4–20

| Component | Property | Value |
|---|---|---|
| Connector | Name | FV211/Y |
| Connector | Name | NK211/Y |
| Connector | Name | NK212/Y |
| Connector | Name | Reactor1_Empty |
| MinMax | Name | MinMax_FlowOut_Reactor1 |
| | Parameter | MIN |
| ValveCurve | Name | ValveCurve_Reactor1 |
| | Exp | 1.0 |
| | Scaling_L_Min | 50 |
| PumpPress | Name | PumpPress_Reactor1 |
| Connector | Name | TV211/Y |
| Connector | Name | NK213/Y |
| MinMax | Name | MinMax_Temperature_Reactor1 |
| | Parameter | MIN |
| PTn | Name | PTn_Disturb_Temp_Reactor1 |
| | T | 20.0 |
| ADD | Name | ADD_Heating_Reactor1 |
| | X2 | 20.0 |
| StirredTankReactor | Name | StirredTankReactor_Reactor1 |
| | SET | Connection to Bconnector_RESET_Reactors OUT |
| | Volume | 1.2 |
| | Height | 5.0 |
| | ULimit | Connection to SR_FF_RESET_Reactors R |
| Connector | Name | LI211/Y |
| Connector | Name | NK214_vent/Y |
| MUL | Name | MUL_Temp_Reactor1 |
| | X2 | -1.0 |
| PTn | Name | PTn_Vent_Reactor1 |
| | T | 10.0 |
| Selection | Name | Selection_Reactor1_Empty |
| | X0 | 100 |

| Component | Property | Value |
|---|---|---|
| Connector | Name | Reactor1_Empty |
| m3sToLiters | Name | m3sToLiters_Reactor1 |
| Connector | Name | FD211_PV |
| ADD | Name | ADD_Temp_Reactor1 |
| NOTc | Name | NOTc_INT_Reactor_LI311 |
| | IN | Connection to Conv2 SensorA2 |
| INT | Name | INT_LI311 |
| Connector | Name | TC221/PV |
| Connector | Name | LI311/PV |

4. Close the chart "Reactor1”.

### 4.3.3 Creating the Chart "Reactor2”

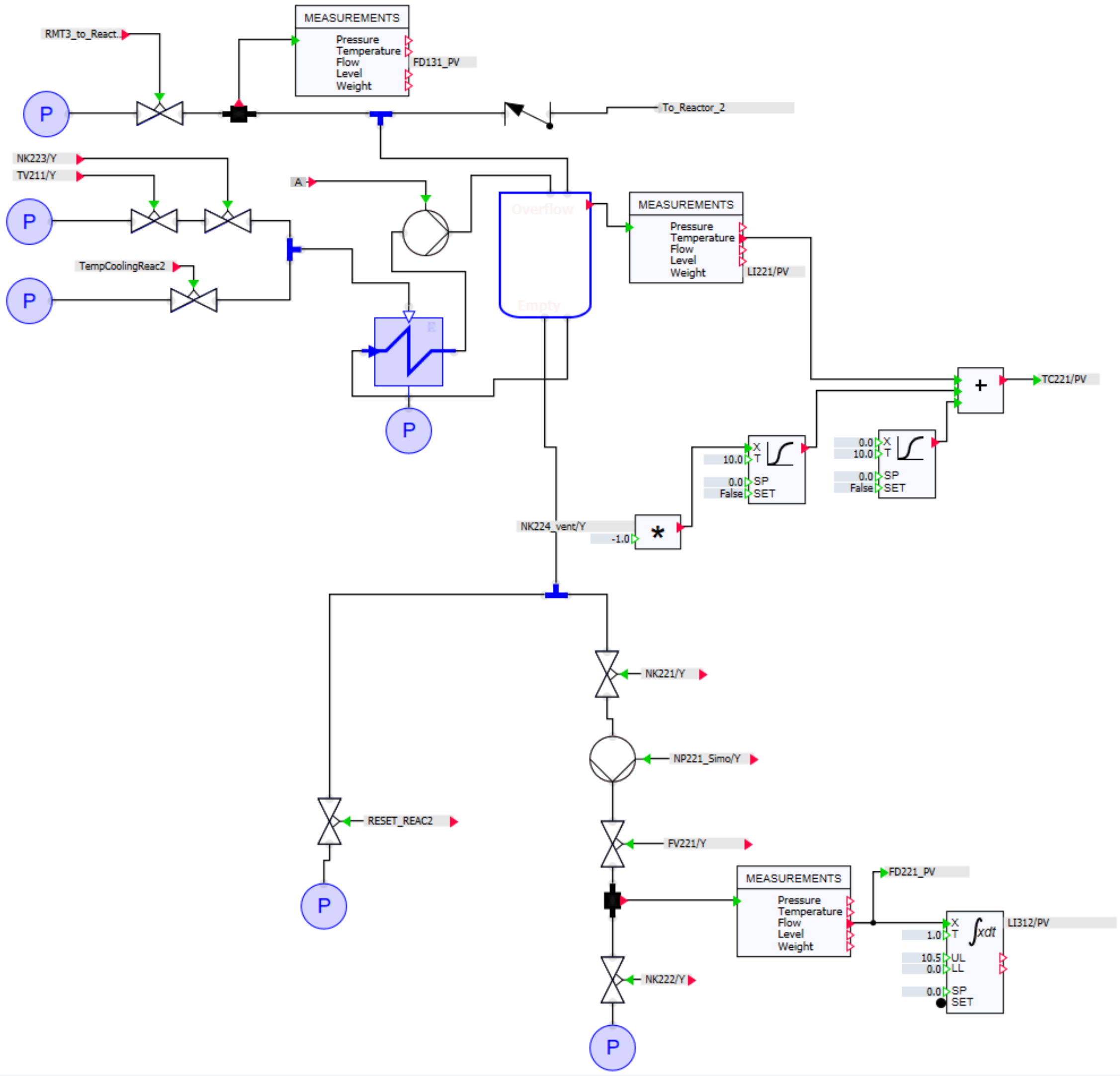
The purpose of the chart is to simulate the physical model of the reactor "Reactor2”.

**Procedure**

1. Create a new chart in the folder "Plant2”.
2. Rename it "Reactor2”.
3. Insert the relevant components into the chart (see the figure below).

4. Interconnect the components as shown in the following figure.

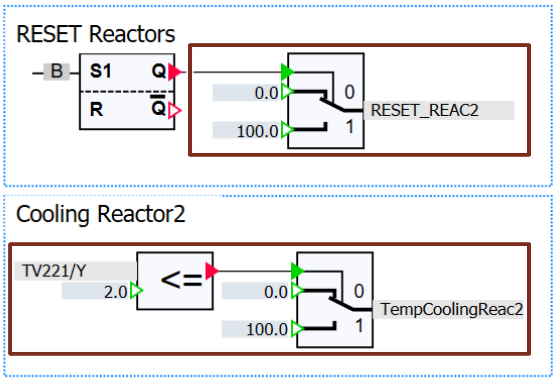
Figure 4-14




5. Open the chart "Connections".
6. Insert the components shown in the following figure into the chart (in a red frame).

7. Interconnect the components as shown in the following figure.

Figure 4-15




8. Adjust the properties of the components as summarized in the table below.

Table 4–21

| Component | Property | Value |
|---|---|---|
| Selection | Name | Selection_RESET_Reactor2 |
| | X1 | 100 |
| Connector | Name | RESET_REAC2 |
| Connector | Name | TV221/Y |
| Compare | Name | Compare_TV221 |
| | X2 | 2.0 |
| | Comparison | <= |
| Selection | Name | Selection_Cooling_Reac2 |
| | X1 | 100 |
| Connector | Name | TempCoolingReac2 |

9. Close the chart "Connections”.
10. Open the chart "Reactor2”.
11. Adjust the properties of the components as summarized in the table below.

Table 4–22

| Component | Property | Value |
|---|---|---|
| PnodeLiquid | Name | PnodeLiquid_Inlet_RMT3_to_Reac2 |
| | Pressure | 40.0 |
| PnodeLiquid | Name | PnodeLiquid_for_Heating_Liquid |
| | Pressure | 50.0 |
| | Temperature | 200.0 |
| PnodeLiquid | Name | PnodeLiquid_for_Heating_Liquid |
| | Pressure | 50 |
| Valve | Name | Valve_RMT3_to_Reactor_2 |
| | CVs | 30.0 |
| | ShowFlow | True |

| Component | Property | Value |
|---|---|---|
| | ShowFlowDirection | True |
| Connector | Name | RMT3_to_Reactor_2 |
| Valve | Name | Valve_TV221/Y |
| | CVs | 360.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | TV221/Y |
| Valve | Name | Valve_NK223/Y |
| | CVs | 360.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | NK223/Y |
| Valve | Name | Valve_TempCoolingReac2 |
| | CVs | 360.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| PipeMeasure | Name | PipeMeasure_RMT3_to_Reactor_2 |
| Measurement | Name | Measurement_FD131 |
| Connector | Name | FD131_PV |
| JointLiquid | Name | JointLiquid_Temp_Reac2 |
| AConst | Name | AConst_Pump_Heating |
| | Constant | 100.0 |
| JointLiquid | Name | JointLiquid_SUM_Inlet_to_Reac2 |
| Pump | Name | Pump_Heating |
| | ShowFlow | True |
| HeatExchangerLiquid | Name | HeatExchangerLiquid_Reac2 |
| | VolumeTS | 1.2 |
| PnodeLiquid | Name | PnodeLiquid_HeatExange_Out |
| | Pressure | 0.1 |
| StopValve | Name | StopValve_RMT2_to_Reactor_2 |
| | CVs | 30.0 |
| Topology | Name | To_Reactor_2 |
| StorageTankLiquid | Name | Reactor2 |
| | Volume | 1.2 |
| | Height | 5.0 |
| | NbrOfStubs | 4 |
| Measurements | Name | Measurements_Reactor2 |
| JointLiquid | Name | LI221/PV |
| MUL | Name | MUL_Reactor2 |
| | X2 | -1.0 |
| Connector | Name | NK224_vent |
| PTn | Name | PT1_Reac2_NK224_vent |
| | T | 10.0 |

| Component | Property | Value |
|---|---|---|
| PTn | Name | PT1_Disturb_Reac2 |
| | T | 10.0 |
| ADD | Name | ADD_Temp_Reactor2 |
| JointLiquid | Name | JointLiquid_Outlet_Reac2 |
| Valve | Name | ResetValve_Reac2 |
| | CVs | 360.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | RESET_REAC2 |
| PnodeLiquid | Name | PnodeLiquid_Reset_Reac2 |
| Valve | Name | Valve_NK221 |
| | CVs | 3.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | NK221/Y |
| Pump | Name | Pump_NP221 |
| | NominalPressure | 4.5 |
| | NominalMassflow | 1.0 |
| | ShuwFlow | True |
| Connector | Name | NP221_Simo/Y |
| Valve | Name | Valve_FV221 |
| | CVs | 0.5 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| Connector | Name | FV221/Y |
| PipeMeasure | Name | PipeMeasure_FD221 |
| Valve | Name | Valve_NK222 |
| | CVs | 3.0 |
| | ShowFlow | True |
| | ShowFlowDirection | True |
| PnodeLiquid | Name | PnodeLiquid_to_Plant3 |
| Measurements | Name | Measurements_FD221 |
| Connector | Name | FD221_PV |
| NOTc | Name | NOTc_LI312 |
| | IN | Connection to Conv2-SensorA2 OUT |
| INT | Name | LI132 |
| | UL | 10.5 |
| Connector | Name | LI312/PV |

12. Close the chart "Reactor2”.

## 4.4 The Current Status of your Project

At the current state of your project you can already test the program of the AS. So far you have created all actuators and sensors in the folder "2_DeviceLevel" and associated their inputs and outputs with the symbolic addresses. Besides this, there are replicas of your plant's physical processes in the "3_ProcessLevel" folder.

# 5 Simulation of the Conveyor System

In chapter 3 you created the conveyor system with the components from the CONTEC library and derived the simulation of the drives with the function "Generating the Device Level”. The following procedure describes how to interconnect the sensors and model the goods conveyed.

## 5.1 Modeling the Goods Conveyed

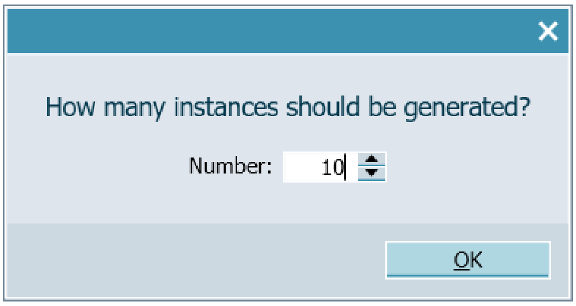
The CONTEC library makes the relevant components available for the simulation of the goods conveyed. These can be used to model the goods conveyed.

### 5.1.1 Creating the List of Goods Conveyed

An inventory of goods to be conveyed is compiled into a materials list for the simulation. Proceed as follows to create this:

1. In the Project window, open the "Material" folder.
2. Double-click "New list”. A new list is created and opened in the workplace.
3. Open the "MATERIAL" folder in the CONTEC library.
4. Drag and drop the component "CBoxDS256” onto the workplace by holding down the "Alt” key.
5. In the window "How many instances should be created?”, set the number to 10 and click "OK”.

Figure 5-1




6. Set the "SizeOfStorage" parameter for the goods conveyed to 8.
7. Save and close the list.

### 5.1.2 Calculating the Color of the Goods Conveyed

In the application example given here, the goods conveyed constitute beverage crates. Depending on the color of the beverage crate, the relevant drink is filled into bottles. The component "BoxProperty” assists in assigning or reading the color of the beverage crate. The component is not part of the CONTEC library and was created using the CTE tool. The application example given here does not describe the creation process.

Besides the component "BoxProperty”, the application example given here contains two other components which are not part of a SIMIT library.

- "CompareBox”
- "CompareMix”

### Component "BoxProperty”

The component "BoxProperty” invokes system functions which enable access to the simulation of the goods conveyed. When the conveyed goods are created by the "Conveyor” component in the CONTEC library, the conveyed goods are assigned a unique ID. This can be read, for example, using the output "SensorIdA1”. The component "BoxProperty” has read or write access to the conveyed goods.

**Note** You can find information regarding the system functions of the CONTEC library in chapter 8.3.4.3 "System functions” in the manual "SIMATIC SIMIT Simulation Platform (V11)".

The "BoxProperty” component has the input "Id” which enables the sensor of the "conveyor” component to read the Id. Furthermore, the component has the inputs "READ” and "WRITE”. If the input "WRITE” is active, then the information present at the other inputs of "BoxProperty” is written to the memory and the properties of the conveyed goods. This includes the values for the inputs "R”, "G”, and "B”. With these values, the color of the conveyed goods in the simulation is defined.

**Note** You can find information regarding the properties of the conveyed goods (e.g. color of conveyed goods) in chapter 8.3.3.4 "Component types for simulating objects” in the manual "SIMATIC SIMIT Simulation Platform (V11)".



In the application example given here, an additional Id is written to memory which represents the color of the goods conveyed.

- Id "10” red
- Id "20” green
- Id "30” blue

If the input "READ” is active, then the memory of the item conveyed is read. Which item conveyed is read, depends on the Id of the item conveyed. This is transferred to the "BoxProperty” component with the aid of the "Conveyor” component sensors.

The information from the memory is written to the outputs of the "BoxProperty” component and can be used there for further processing. The Color-Id is read in the application example given here.

### "CompareBox” Component

The component "CompareBox” imports a number at input X. The outputs are then written using the number. The outputs relate to the color inputs of the item conveyed. Furthermore, the Id of each color is written to the output "Id”.

### "CompareMix” Component

The "CompareMix” component imports the color-Id and reveals at the output whether it is cola, orange-lemonade or a mix of both.

## 5.1.3 "Calc” Chart

The purpose of the "Calc” chart is to model the color of the item conveyed.
It contains the components described in chapter 5.1.2. Besides modeling the color

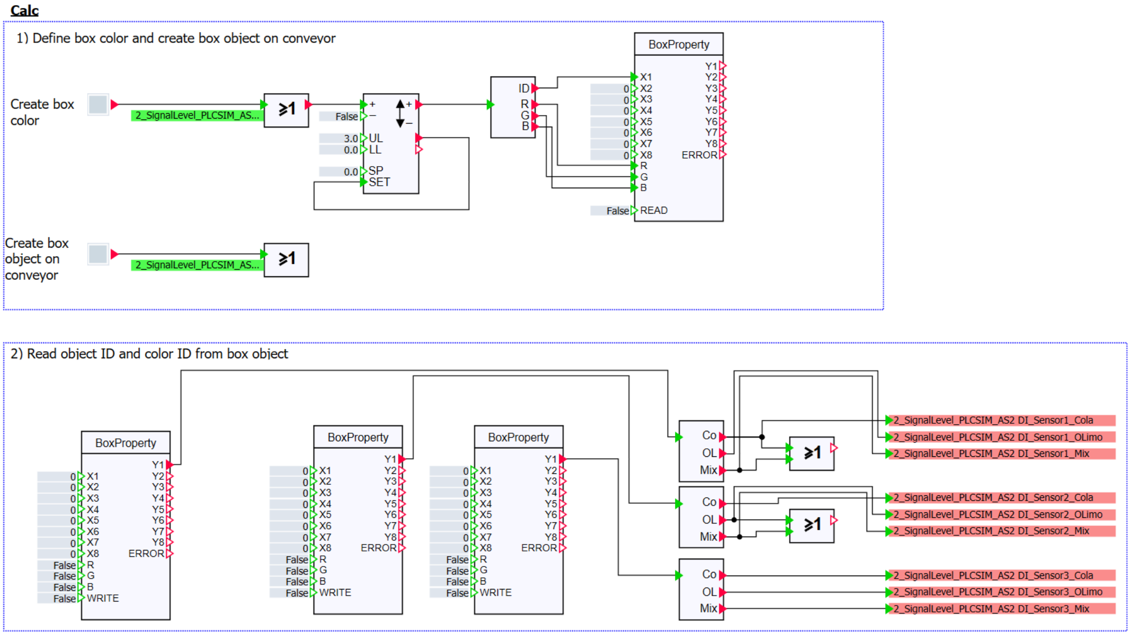
of the item conveyed, the conveyor system’s sensors also analyze the Ids entered. The following section will describe the procedure for creating the chart.

**Procedure**

1. Create a new folder in the folder "ProcessLevel”.
2. Rename it "Conveyor”.
3. Move the chart "Conveyor_sim”, which you have already created (see chapter 3.2.2.) into the folder "Conveyor”.
4. Create another folder in the folder "Conveyor”.
5. Rename it "Misc”.
6. Create a new chart in the folder "Misc”.
7. Rename it "Calc”.
8. Open the chart.
9. Insert the relevant components into the chart (see the figure below).

10. Interconnect the components as shown in the following figure.

Figure 5-2




11. Adjust the properties of the components as summarized in the table below.

Table 5–1 Define box color and create box object on conveyor

| Component | Property | Value |
|---|---|---|
| Pushbutton | Name | Pushbutton_CreateBoxColor |
| Output | Signal | 2_SignalLevel_PLCSIM_AS2 DO_ConvSpure_S_AD |
| OR | Name | OR_CreateBoxColor |
| Counter | Name | Counter_CreateBoxColor |
| | UL | 3.0 |
| CompareBox | Name | CompareBox_CreateBoxColor |
| BoxProperty | Name | BoxProperty |
| | Id | Connection to Conv1 SensorIdA1 |
| | WRITE | Connection to Conv1 SensorA1 |
| Pushbutton | Name | Pushbutton_CreateBoxColor |
| Output | Signal | 2_SignalLevel_PLCSIM_AS2 DO_PickUp |
| OR | Name | OR_CreateBoxObject |
| | OUT | Connection to Conv1 CreateObject |

Table 5–2 Read object ID and color ID from box object

| Component | Property | Value |
|---|---|---|
| BoxProperty | Name | BoxProperty_ReadIDs_Conv2_A1 |
| | Id | Connection to Conv2 SensorIdA1 |
| | READ | Connection to Conv2 SensorA1 |
| BoxProperty | Name | BoxProperty_ReadIDs_Conv2_A2 |
| | Id | Connection to Conv2 SensorIdA2 |
| | READ | Connection to Conv2 SensorA2 |
| BoxProperty | Name | BoxProperty_ReadIDs_Conv2_A3 |
| | Id | Connection to Conv2 SensorIdA3 |
| | READ | Connection to Conv2 SensorA3 |
| CompareMix | Name | CompareMIX_Conv2_A1 |
| | Sin | Connection to Conv2 SensorA1 |
| CompareMix | Name | CompareMIX_Conv2_A2 |
| | Sin | Connection to Conv2 SensorA2 |
| CompareMix | Name | CompareMIX_Conv2_A3 |
| | Sin | Connection to Conv2 SensorA3 |
| OR | Name | OR_Cola_or_Mix |
| OR | Name | OR_OLimo_or_Mix |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor1_Cola |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor1_OLimo |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor1_Mix |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor2_Cola |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor2_OLimo |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor2_Mix |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor3_Cola |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor3_OLimo |
| Input | Signal | 2_SignalLevel_PLCSIM_AS2 DI_Sensor3_Mix |

12. Save and close the chart "Calc”.

## 5.2 The Current Status of your Project

At the current state of your project you can test the automation of the conveyor system.

# 6 Animation of the Crown Cap Machine

Besides the simple two-dimensional animation of graphic elements, SIMIT gives you the option to create three-dimensional animations. The three-dimensional animation of the crown cap machine is created below.

## 6.1 Creating the 3D Viewer Control

The 3D viewer control (Taskcard Control) is contained in the basic library of SIMIT. 3D animations can be created quickly and easily with this control, as it gives you the option to import the geometry model of machines from a CAD system. The import requirement is that the model has been exported in the format VRLM V2.0.

**Note**

The description given here does not cover the generation of the geometry model.

You can find further information regarding the 3D viewer control and the data format requirement in chapter 8.1.6.4 "3D Viewer control" in the manual "SIMATIC SIMIT Simulation Platform (V11)".



**Procedure**

1. Create a new chart in the folder "Conveyor".
2. Rename it "3D_CrownCap".
3. Open the chart.
4. Insert the component "3D viewer" from the task card "Controls".
5. Adjust the properties of the components as summarized in the table below.

Table 6-1

| Component | Property | Value |
|---|---|---|
| 3D-Viewer | Name | CrownCapMaschine |
| | X | 160 |
| | Y | 15 |
| | Width | 520 |
| | Height | 695 |
| | 3D model | MASCHINE.wrl |

**Note**

The file "MASCHINE.wrl" belongs to the ZIP files "77362399_BottlingPlant_SIMIT_PROJ_V912.zip" which is available for download from the article page.

6. Save and close the chart.

## 6.2 Creating the Animation

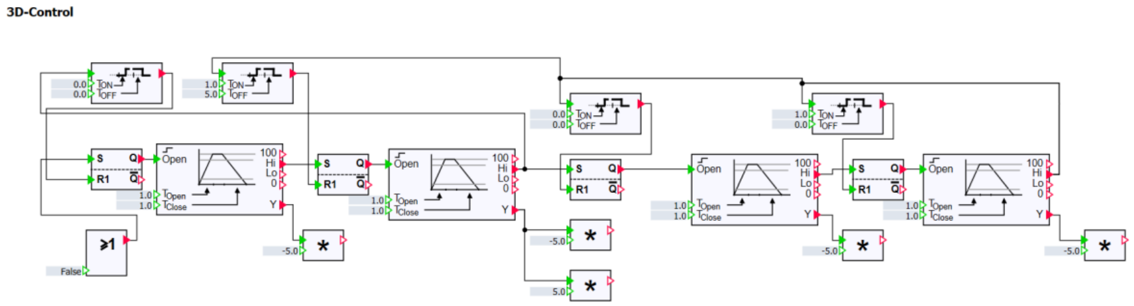
After you have integrated the geometry model of the machine into the 3D viewer control, it will have various connectors (see properties of the 3D viewer control "CrownCapMachine"). These can be used to address and animate the model. For the animation, the movements of the individual machine parts are modeled using standard library components. The animation begins as soon as a item has left the conveyor line.

The following section will describe the procedure for creating the animation.

### Procedure

1. Create a new chart in the "Misc" folder of the "Conveyor" folder.
2. Rename it "AnimationControl".
3. Open the chart.
4. Insert the relevant components into the chart (see the figure below).
5. Interconnect the components as shown in the following figure.

Figure 6-1



6. Adjust the properties of the components as summarized in the table below.

Table 6–2

| Component | Property | Value |
|---|---|---|
| Delay | Name | Delay_sonsor_z_axis_TY_move_back |
| RS_FF | Name | RS_FF_sonsor_z_axis_TY |
| OR | Name | OR_start_animation |
| | IN1 | Connection to Conv5 SensorA2 |
| | IN2 | Connection to Conv3 SensorA3 |
| | IN3 | Connection to Conv7 SensorA2 |
| DriveV1 | Name | DriveV1_sonsor_z_axis_TY |
| MUL | Name | MUL_sonsor_z_axis_TY |
| | X2 | -5.0 |

| Component | Property | Value |
|---|---|---|
| Delay | Name | Delay_doors_close |
| | T_ON | 1.0 |
| | T_OFF | 5.0 |
| RS_FF | Name | RS_FF_door_sensors |
| DriveV1 | Name | DriveV1_door_sensors |
| MUL | Name | MUL_door_sensor_left_TX |
| | X2 | -5.0 |
| MUL | Name | MUL_door_sensor_right_TX |
| | X2 | 5.0 |
| Delay | Name | Delay_sonsor_y_axis_TX_move_back |
| RS_FF | Name | RS_FF_sonsor_y_axis_TX |
| DriveV1 | Name | DriveV1_sonsor_y_axis_TX |
| MUL | Name | MUL_sonsor_y_axis_TX |
| | X2 | -5.0 |
| Delay | Name | Delay_sonsor_x_axis_TX_move_back |
| | T_ON | 1.0 |
| RS_FF | Name | RS_FF_sonsor_x_axis_TX |
| DriveV1 | Name | DriveV1_sonsor_x_axis_TX |
| MUL | Name | MUL_sonsor_x_axis_TX |
| | X2 | -5.0 |

7. Save and close the chart "AnimationControl".
8. Open the chart "3D_CrownCap".
9. Select the 3D viewer control components and adjust the properties according to the table below.

Table 6–3

| Component | Property | Value |
|---|---|---|
| CrownCapMaschine | Sensor_X-axis#TX | Connection to MUL_sonsor_x_axis_TX Y |
| | Sensor_Y-axis#TX | Connection to MUL_sonsor_y_axis_TX Y |
| | Sensor_Z-axis#TX | Connection to MUL_sonsor_z_axis_TY Y |
| | Door_sensor_left#TX | Connection to MUL_door_sensor_left_TX Y |
| | Door_sensor_right#TX | Connection to MUL_door_sensor_right_TX Y |

10. Save and close the chart.

## 6.3 The Current Status of your Project

At the current state of your project you have created a 3D viewer control and integrated a corresponding geometry model. In addition, you have created animation for the 3D viewer control.

Configuration is complete at this point. The following section will introduce you to the scripts and snapshots contained in the project.

In the example project given here, there are charts which contain other animations with graphic elements and controls which are not described in the application example given here. For this, some charts which were created throughout the application example, were adjusted. These adjustments are optional and not part of the description given here.

**Note** You can find information regarding graphic elements and controls in the manual "SIMATIC SIMIT Simulation Platform (V11)" and in the application example "SIMIT Simulation V11.0 SP1 Getting Started", in chapter 8 "Visualization in SIMIT".

# 7 Scripts and Snapshots

The example project given here contains three scripts and a snapshot.
The following section describes the functions of the scripts and snapshots. Both of these functions offer great advantages when developing an Operator Training System.

## 7.1 Scripts

SIMIT gives you the opportunity to influence the simulation by using the Automatic Control Interface with automated scripts. This allows you to create situations such as the overheating of a reactor. Furthermore, scripts can be used to create snapshots. This allows you to create a plant status with a script and then save it as a snapshot. Furthermore, you can create log files with measurement values for the period of time that a script is running. This allows you to determine, for example, the operator reaction time required to prevent, for example, the overheating of a reactor.

### Creating Scripts

Scripts are created in the folder "Scripting" in the project window. These can be created before the simulation starts or during the simulation. Changes can be made to the scripts throughout the simulation. To create a script, open the folder "Scripting" and execute the function "New script". You can enter the commands in the editor window and if the simulation is running, you can start the script using the "Start script" button.



**Note**

> For detailed information about creating and starting scripts and commands, please refer to chapter 6 "Scripts" of the manual "SIMATIC SIMIT Simulation Platform (V11)".

### "LeanPIDTest" Script

A simple control test for temperature regulation of "Reactor1" can be carried out using the "LeanPIDTest" script. After the script has started, a log file is created. The start date and start time are recorded in the log file. A query appears to check whether the target value of 10°C is set on the regulator. If the response is "yes", a plot of the actuating value is created for the valve TV211. After a period of 20,000 cycles, a disturbance of 10°C is added to the temperature value TC211 for another 20,000 cycles. At the end, the disturbance is removed and a snapshot "Lean-OTS" is created.

**Note**

> The paths specified in the script can be adjusted, if required.

**Script "Reactor1OverHeating"**

The overheating of "Reactor1" is simulated using "Reactor1OverHeating". After the script has started, a log file is created. The start date and start time are recorded in the log file. Then a disturbance variable is added to the reactor temperature. The disturbance variable is only removed when the operator opens the valve "NK214_vent" (in the WinCC Runtime). At the end of the script, the operator's reaction time is recorded in the protocol file.

**Note** The paths specified in the script can be adjusted, if required.

**Script "Reactor2OverHeating"**

The script "Reactor2OverHeating" corresponds to the script "Reactor1OverHeating". Only "Reactor2" is heated and reacts to valve "NK224_vent" being opened.

## 7.2 Snapshots

The snapshot "Lean-OTS" is included in the application example given here. This was created automatically by the script "LeanPIDTest". It can be created manually using the function "Snapshots". You can select this in the menu "Simulation > Snapshots" by clicking the button "Snapshots" in the toolbar or in the folder "Snapshots" in the project window.

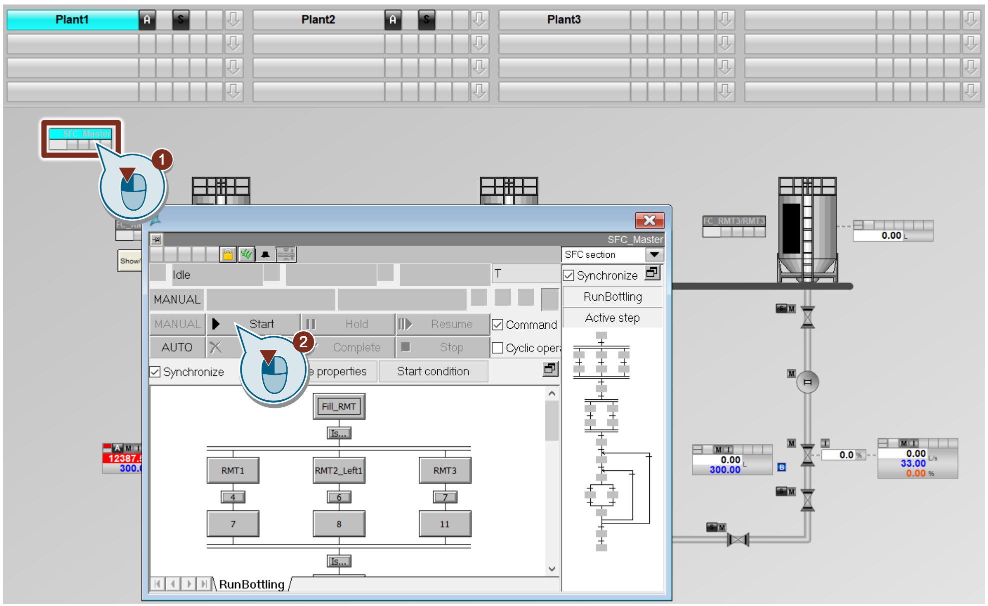
You can retain the plant status in a snapshot and reload it as often as you like.

# 8 Commissioning

The commissioning procedure for the SIMIT and PCS 7 project is described in detail in the application example "SIMIT simulation of a stirred tank reactor with PCS 7".

Start the SFC "SFC_Master" in the WinCC figure "Plant1" after you have started the simulation and the WinCC Runtime (see the following figure).

Figure 8-1



# 9 Appendix

## 9.1 Service and support

### Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

### Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers
– ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:

siemens.com/SupportRequest



### SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

### Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

### Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

## 9.2 Industry Mall


The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:
mall.industry.siemens.com

## 9.3 Links and literature

Table 9-1

| Nr. | Thema |
|---|---|
| \1\ | Siemens Industry Online Support<br>https://support.industry.siemens.com |
| \2\ | Link to this entry page of this application example<br>https://support.industry.siemens.com/cs/ww/en/view/77362399 |
| \3\ | SIMIT Simulation V11.0 SP1 Getting Started<br>https://support.industry.siemens.com/cs/ww/en/view/109746485 |
| \4\ | SIMIT simulation of a stirred tank reactor with PCS 7<br>https://support.industry.siemens.com/cs/ww/en/view/93148023 |
| \5\ | SIMATIC PCS 7 Process Control System PCS 7 Readme V9.1 SP2 (Online)<br>https://support.industry.siemens.com/cs/ww/en/view/109806027 |
| \6\ | SIMATIC SIMIT Simulation Platform (V11) Operating Manual<br>https://support.industry.siemens.com/cs/ww/en/view/109812085 |

## 9.4 Change documentation

Table 9-2

| Version | Date | Modifications |
|---|---|---|
| V1.0 | 10/2015 | First edition |
| V2.0 | 11/2022 | Update to PCS 7 V9.1 SP2 and SIMIT SP V11 |