Single and Multi Loop Controller Structures (Cascade Control) with PID_Temp

SIMATIC S7-1200/S7-1500

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 at: https://www.siemens.com/industrialsecurity.
Table of Contents

Legal information ............................................................................................................. 2

1 Task.......................................................................................................................... 4

2 Solution...................................................................................................................... 5
   2.1 Overview .............................................................................................................. 5
   2.2 Hardware and software components ................................................................... 7
   2.2.1 Validity ........................................................................................................... 7
   2.2.2 Components used .......................................................................................... 7

3 Basics on Control Engineering ............................................................................... 8

4 Function Principle .................................................................................................... 10
   4.1 General overview .............................................................................................. 10
   4.2 Startup OB – OB100 ........................................................................................ 12
   4.3 Simulated process value “chocolate water bath” .............................................. 12
   4.4 Single-loop control .......................................................................................... 14
   4.5 Triple-loop control ........................................................................................... 14

5 Configuration and Commissioning of the PID_Temp Controller .................. 17
   5.1 PID_Temp configuration, single-loop ............................................................. 17
   5.2 PID_Temp configuration, triple-loop ............................................................ 21
   5.3 Commissioning ................................................................................................. 29

6 Installation and Commissioning of the Example Project .................................. 33
   6.1 Installing the hardware .................................................................................... 33
   6.2 Installing the software ..................................................................................... 34
   6.3 Commissioning ................................................................................................. 34

7 Operation of the WinCC User Interface ............................................................... 36
   7.1 Overview ........................................................................................................... 36
   7.2 Scenario: setpoint step ................................................................................... 38
   7.3 Scenario: disturbance variable impact ............................................................ 39

8 Related Literature .................................................................................................... 40

9 History ....................................................................................................................... 40
1 Task

Introduction

Controlling temperature systems is a great challenge for a controller, even when dealing with inert controlled systems.

Apart from integrated universal controllers, SIMATIC S7-1200 and S7-1500 also offer a special temperature controller for this. It is suitable for heating or heating/cooling applications and can be easily cascaded.

Using the example of a “chocolate water bath” scenario, this application shows the realization of a controller-based control for tempering the chocolate temperature. In order to obtain good product properties (e.g. gloss, consistency, and taste), the process needs to be warmed up and cooled down again in several steps.

The “chocolate water bath” consists of a container with water into which warm and cold water can be fed via valves. The temperature of the water is then transferred to the steel container, i.e. the chocolate mass to be melted.

Overview of the automation task

The figure below provides an overview of the controlled system and its actuators and measuring elements.

Figure 1-1

Description of the automation task

For the control task, a water bath with three process tags shall be simulated and controlled as ideally as possible.

This application also enables a comparison between a control system with three controllers (cascaded control/triple-loop control) and a control system with only one controller regarding

- implementation workload,
- commissioning and
- control behavior and disturbance behavior of the control loop.
2 Solution

2.1 Overview

Display

The figure below shows a schematic overview of the most important components of the solution:

The application realizes:

- a simulated process for the “chocolate water bath” scenario. The process is described in Chapter 4.3.
- a triple-loop control in master-slave interconnection with three PID_Temp blocks. The function mechanism (see Chapter 4.5) and the realization (see Chapter 5.2) are described.
- a single-loop control with the PID_Temp block. The function mechanism (see Chapter 4.4) and the realization (see Chapter 5.1) are described.
- a WinCC user interface for a quick overview over the course of the setpoints and actual values.

1 The PID_Temp block is part of STEP 7 and can be used with the appropriate PLC-firmware (available as free of charge system upgrade).
Visualization

The visualization realized in WinCC enables gaining an overview over both realized controlled systems:

Figure 2-2

Advantages

The example application offers the following advantages:
1. Illustrative example project for the adjustment to own requirements for multi-loop control systems.
2. Step-by-step instructions for programming and commissioning a triple-loop control system in master-slave interconnection.
3. Direct comparison of control behavior of a single-loop and a triple-loop control system via the WinCC user interface.

Delimitation

This application does not include a description of:
- STEP 7
- SCL/LAD/FBD programming languages.

Basic knowledge of these topics is assumed.
Furthermore, this application does not provide a basic overview of the theories of control technology.
2.2 Hardware and software components

2.2.1 Validity

This application is valid for
- STEP 7, as of V15.1
- WinCC, as of V15.1
- S7-1500, as od FW 2.6
- S7-1200, as of FW V4.1

2.2.2 Components used

The application was created with the following components:

Hardware components

<table>
<thead>
<tr>
<th>Component</th>
<th>Qty.</th>
<th>Article number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU 1516-3 PN/DP</td>
<td>1</td>
<td>6ES7516-3AN01-0AB0</td>
<td>You can also select another CPU of the S7-1500 family.</td>
</tr>
<tr>
<td>PM 70W 120/230VAC</td>
<td>1</td>
<td>6EP1332-4BA00</td>
<td>You can also use a different module for supplying the S7-1500 CPU.</td>
</tr>
<tr>
<td>PG M4</td>
<td>1</td>
<td>6ES7716-......0...</td>
<td>You can also use a different PC with respective hardware and software.</td>
</tr>
</tbody>
</table>

Software components

<table>
<thead>
<tr>
<th>Component</th>
<th>Qty.</th>
<th>Article number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMATIC STEP 7 Professional V15.1</td>
<td>1</td>
<td>6ES7822-1AA05-0YA5</td>
<td></td>
</tr>
<tr>
<td>WinCC Runtime Advanced 128 V15.1</td>
<td>1</td>
<td>6AV2104-0BA05-0AA0</td>
<td></td>
</tr>
</tbody>
</table>

Sample files and projects

The following list includes all files and projects that are used in this example.

<table>
<thead>
<tr>
<th>Component</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>103526819_PID_Temp_CODE_v11.zip</td>
<td>This zip file contains the STEP 7 project.</td>
</tr>
<tr>
<td>103526819_PID_Temp_DOKU_v11_d.pdf</td>
<td>This document.</td>
</tr>
</tbody>
</table>
3 Basics on Control Engineering

Overview

Control engineering is an engineering science researching how to influence specific given parameters in technological systems, with the aim of reaching and maintaining the desired value of this parameter under certain conditions.

This chapter contains a very short extract on the topic of “control technology”.

The system manual of STEP 7 Professional discusses PID control with basics on control technology (3).

Controlled system

A controlled system contains the parameter to be controlled, such as the temperature of a room. In order to identify the type of a system and then dynamically controlling it in an optimal way requires a precise analysis of the system to be controlled.

One possibility of identification is to look at the step response of a controlled system. The picture below depicts the example of a PTn system (temperature in a room, for example).

The time behavior can be defined approximately by the parameters

- Delay time $T_u$
- Compensation time $T_g$
- Maximum value $X_{\text{max}}$

Figure 3-1 Step response PTn system
Controller

The controller controls an actuator to bring the controlled system to a desirable state. The simplest controllers are two-point controllers, which only know the states “ON” and “OFF” and control the controlled system via the actuator.

The frequently used PID controller consists of three parts:

- The P-fraction creates an output signal proportional to the control deviation.
- The I-fraction integrates the control deviation over time and affects the controlled system due to this integration.
- The D-fraction on the other hand, reacts to the changed control deviation (temporal derivation of the control deviation).

These three parts of the ideal PID controller are weighted by the following coefficients: proportional gain, integration time, and differentiation time.

With the “PID_Compact”, “PID_3Step”, and “PID_Temp” blocks, SIMATIC S7-1500 already offers a possibility of software control integrated into the firmware.

Note

“PID_Temp” is used in this application. Further information on “PID_Compact” and “PID_3Step” is available in the manual \[3\] and in the online help of the TIA Portal.

Control loop

In a control loop, the control deviation between setpoint and actual value is determined by the controller and a manipulated variable derived from it. The manipulated variable acts on the controlled system via an actuator (see Figure 3-2).

Figure 3-2 control loop, single

A simple example for a control loop is controlling the room temperature through a heater. The room temperature is measured with a sensor and fed to a controller, which compares the current room temperature with a setpoint value and calculates an output value (control value) for controlling the heater.

If several sensors acquire different values of one process, it often makes sense to use a multi-loop controller system. This application realizes a triple-loop control system with the “PID_Temp” controller.
4 **Function Principle**  
This chapter describes the basic function mechanisms of the application example.

4.1 **General overview**  
The user program of the S7-1500 CPU is divided into two parts.

**Scenario – Cascaded control loop**  
The cyclically (every 10 ms) called OB30 calls the function blocks PID_Temp for control, as well as the simulation blocks for the simulated “chocolate water bath” process (see Figure 4-2).

The block diagram shows the feedback of one respective measured variable of the simulated process to an input of the PID_Temp block. The controllers are interconnected in the form of a cascade.

Figure 4-1 Block diagram: “triple-loop” interconnection

Figure 4-2 PLC program triple-loop control
4 Function Principle

4.1 General overview

Scenario – Single-loop control

The cyclically (every 10ms) called OB31 calls the PID_Temp function block for controlling, as well as the simulation blocks for the simulated “chocolate water bath” process (see Figure 4-4). The block diagram shows the feedback of the measured value “chocolate” to the input of the PID_Temp controller.

Figure 4-3 Block diagram: “single-loop” interconnection

Figure 4-4 PLC program single-loop control

Differences of the scenarios

Both scenarios mainly differ in using the existing “process tags” (corresponds to the output parameters of the simulated components of the process).

In triple-loop cascade control, the temperature values of the subsystems water, steel container, and chocolate are fed back to one controller each.

In single-loop control only the chocolate temperature is controlled by one controller. The improved behavior of the controlled system with triple-loop control is clearly shown illustrated 4.5.
4.2 **Startup OB – OB100**

The startup OB (OB100) realizes the following functions when restarting the CPU:
- initializing the simulation blocks
- initializing the PID_Temp controller

4.3 **Simulated process value “chocolate water bath”**

**Process**

The following figure shows the process to be simulated:

Figure 4-5

![Diagram of the chocolate water bath process](image)

**Library used**

The simulation blocks used are largely taken from the “Sim_controlprocess” library which can be downloaded in [18].

**“Chocolate water bath” process**

The “chocolate water bath” is realized by serial interconnection of three simulation FBs.

The individual controlled systems have different time constants, but identical reinforcements.

Figure 4-6 Chocolate water bath

![Diagram of the chocolate water bath process](image)

The individual elements have the following characteristic:

- Process Water: different PT1 behavior for heating and cooling
- Process Steel: aperiodic PT2 system (no overshoot for step response)
- Process Chocolate: single PT1 system
4 Function Principle

4.3 Simulated process value “chocolate water bath”

The following time constants were selected:

Table 4-1

<table>
<thead>
<tr>
<th></th>
<th>GAIN</th>
<th>TM_LAG1 or TM_LAG_Heat</th>
<th>TM_LAG_Cool</th>
<th>TM_LAG2</th>
</tr>
</thead>
<tbody>
<tr>
<td>asym. PT1 element</td>
<td>1</td>
<td>1.0</td>
<td>0.75</td>
<td>-</td>
</tr>
<tr>
<td>(water)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aper. PT2 element</td>
<td>1</td>
<td>15.0</td>
<td>-</td>
<td>5.0</td>
</tr>
<tr>
<td>(steel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT1 element</td>
<td>1</td>
<td>125.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(chocolate)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note

The example does not claim to deliver a detailed depiction of reality. This application is focused on programming and commissioning the PID_Temp controller blocks.

Cold and warm water feed

The controlled system has the two inputs “INV_COOL” and “INV_HEAT”. One cooling or heating power value, which affects the controlled system, can each be mapped via both inputs (with positive values).

In the application, the real outputs OutputHeat and OutputCool of the PID_Temp controller are interconnected with the inputs of the controlled system.

Both inputs simulate the variably adjustable valves of a real process.

With the settings made in the example application, the process can be controlled between the values 1 °C and +130 °C.

Identical simulation

Independent of each other, however, the process is simulated with identical parameters in OB “30_three_loops” (OB30) and in OB “31_one_loop” (OB31).
4.4 Single-loop control

Function principle

For single-loop control, a PID_Temp controller is interconnected with the simulated process (see Figure 4-3).

Table 4-2

<table>
<thead>
<tr>
<th>Function block</th>
<th>Measured variable (feedback from the simulation)</th>
<th>Slave of</th>
<th>Master for</th>
<th>Control value interconnected with</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID_Temp_4</td>
<td>Chocolate temperature</td>
<td>-</td>
<td>-</td>
<td>warm and cold water ‘valve’</td>
</tr>
</tbody>
</table>

Advantage

In comparison with a triple-loop controller, a single-loop controller does yield worse results; however, this type of interconnection also offers some advantages:

- reduced hardware demand (only one sensor for feeding back the controlled variable from the process).
- only one controller needs to be commissioned (thanks to the support by graphic technology objects, commissioning several controllers isn’t a complex procedure either).

4.5 Triple-loop control

Function principle

For triple-loop control, three PID_Temp controllers are interconnected (see Figure 4-1).

Each controller receives a different measured variable for monitoring from the simulated process.

The PID_Temp controllers are master-slave-connected, which yields advantages during commissioning (see Chapter 5.3).

For commissioning the controllers, only the PI controller set is used instead of the PID controller set. When cascading controllers, a D-fraction in the controllers strongly affects the outputs. Even the noise of the actual value can, due to the cascade connection of the controllers, cause a heavily overshooting control (see Figure 4-7, the red line is the heating value). This behavior is not desired in a real plant, since it strongly wears the existing hardware (valves for example).

Figure 4-7 Heating and cooling value for PID control
4 Function Principle

4.5 Triple-loop control

The following table gives you an overview of the interconnection of the controllers:

Table 4-3

<table>
<thead>
<tr>
<th>Function block</th>
<th>Measured variable (feedback from the simulation)</th>
<th>Slave of</th>
<th>Master for</th>
<th>Control value interconnected with</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID_Temp_1 (&quot;outside&quot;)</td>
<td>Chocolate temperature</td>
<td>-</td>
<td>PID_Temp_2</td>
<td>PID_Temp_2</td>
</tr>
<tr>
<td>PID_Temp_2 (&quot;center&quot;)</td>
<td>Melting tank temperature</td>
<td>PID_Temp_1</td>
<td>PID_Temp_3</td>
<td>PID_Temp_3</td>
</tr>
<tr>
<td>PID_Temp_3 (&quot;inside&quot;)</td>
<td>Water temperature</td>
<td>PID_Temp_2</td>
<td>-</td>
<td>warm and cold water ‘valve’</td>
</tr>
</tbody>
</table>

**Advantage**

Feeding back and monitoring more than one measured value yields a lower susceptibility for the controlled system as compared with single-loop control (see Figure 4-10).

**Comparison of control behavior**

Each figure displays the setpoint, actual value, as well as the heating and cooling power of the output.

The control behavior varies depending on the system used.

Figure 4-8 Control behavior after “fine tuning”

triple-loop:

![Triple-loop control behavior](image)

single-loop:

![Single-loop control behavior](image)
4 Function Principle

4.5 Triple-loop control

**Comparison of disturbance behavior**

*Figure 4-10* shows the disturbance behavior after a disturbance variable affects the simulated "water bath" process (FB “1_1ProcessWater” and FB “2_1ProcessWater”). Each impact on the controlled variable is marked red.

*Figure 4-9 Disturbance on simulated system*

![Disturbance on simulated system](image)

The different reaction time speeds and times until reaching the setpoint again can be explained as follows:

For single-loop control the effect of the disturbance variable in the ‘outer’ process (water) must first affect the ‘inner’ process (chocolate). Only then does the single-loop controller detect the disturbance and controls accordingly.

For triple-loop control, the inner controller directly detects the impact on the process and starts controlling accordingly. As can be seen in the figure, the impact of the disturbance variable does not affect the inner process (the input for the outer controller).
5 Configuration and Commissioning of the PID.Temp Controller

5.1 PID.Temp configuration, single-loop

Overview

The following table gives you an overview of the interconnection of the controller.

<table>
<thead>
<tr>
<th>Measured variable (feedback from the simulation)</th>
<th>Slave of</th>
<th>Master for</th>
<th>Control value interconnected with</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID.Temp_4 Chocolate temperature</td>
<td>-</td>
<td>-</td>
<td>warm and cold water ‘valve’</td>
</tr>
</tbody>
</table>

Controller configuration

The following table provides a step-by-step instruction for configuring a single-loop PID.Temp controller.

The screenshots were created in the “PID.Temp” example project. The programming language used is SCL. The interconnections apply in the same way for other programming languages such as LAD, FBD or STL.

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Prepare the simulated controlled system.</td>
</tr>
<tr>
<td>2.</td>
<td>Add a “cyclic interrupt” OB into your project Select a call interval (10 000μs for example)</td>
</tr>
</tbody>
</table>

Note: The “PID.Temp” instruction must be called cyclically.
5 Configuration and Commissioning of the PID_Temp Controller

5.1 PID_Temp configuration, single-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>Add the “PID_Temp” instruction into your block in “Technology &gt; PID Control &gt; Compact PID”.</td>
</tr>
<tr>
<td></td>
<td><img src="image1.png" alt="Technology tree" /></td>
</tr>
<tr>
<td></td>
<td>Note: If you wish to use the advantages of a technology object, do not add the “PID_Temp” instruction into your project as a multi-instance block.</td>
</tr>
<tr>
<td>4.</td>
<td>Interconnect the following parameters of the instruction:</td>
</tr>
<tr>
<td></td>
<td>• Setpoint: with a setpoint tag of your project.</td>
</tr>
<tr>
<td></td>
<td>• Input: with the sensor/output of the controlled system</td>
</tr>
<tr>
<td></td>
<td>The Input parameter is used for interconnection with a REAL value. The Input_PER parameter can be used for direct interconnection with an analog value.</td>
</tr>
<tr>
<td></td>
<td>```javascript</td>
</tr>
<tr>
<td></td>
<td>//call of the PID_Temp instruction</td>
</tr>
<tr>
<td></td>
<td>```PID_Temp_4``(Setpoint:=&quot;Control_DB&quot;.SETPOINT,</td>
</tr>
<tr>
<td></td>
<td>Input:=&quot;2_3ProcessChocolate&quot;.Output);</td>
</tr>
<tr>
<td>5.</td>
<td>Click on the “PID_Temp_4” writing and open the properties page in the inspector window.</td>
</tr>
<tr>
<td></td>
<td>Go to the “Basic settings &gt; Input/output parameters” tab.</td>
</tr>
<tr>
<td></td>
<td>Select the following for the input: from the drop-down menu the “Input” value in order to activate the interconnected process value.</td>
</tr>
<tr>
<td></td>
<td><img src="image2.png" alt="PID Temp properties" /></td>
</tr>
<tr>
<td>6.</td>
<td>In the project tree of your project you go to the “Technology objects” folder.</td>
</tr>
<tr>
<td></td>
<td>There, you double-click to open the configuration under the third inserted instruction.</td>
</tr>
<tr>
<td></td>
<td><img src="image3.png" alt="Technology objects" /></td>
</tr>
</tbody>
</table>
5 Configuration and Commissioning of the PID_Temp Controller

5.1 PID_Temp configuration, single-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
</table>
| 7.  | Go the “Basic settings” tab. Enter the following settings:  
- Controller mode: Temperature  
  A temperature process shall be controlled.  
- Mode after CPU restart: Auto  
  The controlling is to start automatically after a cold start of the PLC (after a full loading or an MRES).  
- Input: Input  
  Already assigned in step 5:  
- OutputHeat: OutputHeat  
  The Real output of the controller shall be used.  
- Activate the “Activate cooling” checkbox  
  The block shall initiate heating as well as cooling.  
- OutputCool: OutputCool  
  The Real output of the controller shall be activated. |

In “Cascade”, no settings need to be made since only a single-loop control is used.

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
</table>
| 8.  | Go to the “Process value settings” tab.  
Adjust the limits for the process value to your application (in the example project 140 °C were selected for the upper limit and -10 °C for the lower limit). |
## 5 Configuration and Commissioning of the PID_Temp Controller

### 5.1 PID_Temp configuration, single-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
</table>
| 9.  | Go to the “Output settings” tab.  
The following settings were made in the example project:  
Basic settings of output  
Heating/Cooling  
Method for heating/cooling: Switch PID parameters for heating/cooling  
Cooling Factor: 1.0  
Reaction to error  
Set PidOutputSum to: Substitute output value while error is pending  
Substitute output value: 0.0 %  
The real outputs (“OutputHeat” and “OutputCool”) for controlling the simulated controlled system are used.  
The following setting in the example project enables controlling between “+1 °C” and “+130 °C” (the ambient temperature of the process is specified as 23 °C).  
Using the technology object of the controller enables scaling the percentage value of the controller directly to other values (e.g. according to your valve). |
| 10. | In the example project, the extended settings  
• process value monitoring  
• minimal on/off times for pulse width modulation  
• manual setting of PID parameters  
are not used.  
In order to influence the commissioning process, a controller structure for tuning can be set.  
If this value is set to PID (temperature), it is attempted during the commissioning process to find PID parameters which cause as little an overshoot of the process above the setpoint as possible.  
This setting can be made separately for heating and cooling. |
5.2 PID_Temp configuration, triple-loop

Overview

The following table gives you an overview of the interconnection of the controllers.

Abbildung 5-1

Outer controller (#1)

For the configuration of the outer controller, please follow the instructions in the table below.

Table 5-3

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Repeat step 1 and step 3 in Table 5-2.</td>
</tr>
<tr>
<td>2.</td>
<td>Interconnect the controller.</td>
</tr>
<tr>
<td></td>
<td>• Setpoint: with a tag which specifies the setpoint.</td>
</tr>
<tr>
<td></td>
<td>• Input: with the sensor/output of the last part of the controlled system.</td>
</tr>
<tr>
<td></td>
<td>&quot;PID_Temp_1&quot; (Setpoint := &quot;Control_DB&quot;.SETPOINT, Input := &quot;1 3ProcessChocolate&quot;.Output)</td>
</tr>
<tr>
<td>3.</td>
<td>Click on the &quot;PID_Temp_2&quot; writing and open the properties page in the inspector window.</td>
</tr>
<tr>
<td></td>
<td>Go to the &quot;Basic settings &gt; Input/output parameters&quot; tab.</td>
</tr>
<tr>
<td></td>
<td>Select the following for the input: from the drop-down menu the &quot;Input&quot; value in order to activate the interconnected process value.</td>
</tr>
<tr>
<td>4.</td>
<td>In the project tree of your project you go to the “Technology objects” folder. There, you double-click to open the configuration under the instruction &quot;PID_Temp_1&quot;.</td>
</tr>
</tbody>
</table>
5 Configuration and Commissioning of the PID_Temp Controller

5.2 PID_Temp configuration, triple-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

5. Go the "Basic settings" tab.

Enter the following settings:
- **Controller mode: Temperature**
  A temperature process shall be controlled.
- **Mode after CPU restart: Auto**
  The controlling is to start automatically after a cold start of the PLC (after a full loading or an MRES).
- **Input: Input**
  Was already supplied in step 2 and 3.
- **OutputHeat: OutputHeat**
  The real output OutputHeat of the controller is used.
- **Deactivate the “Activate cooling” checkbox**
  The block works as master for the downstream PID_Temp.
- **Activate the “Controller is master” checkbox**
  The controller is the master for the downstream PID_Temp (#2).
5 Configuration and Commissioning of the PID_Temp Controller

5.2 PID_Temp configuration, triple-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Basic settings</strong>&lt;br&gt;Controller type&lt;br&gt;Temperature&lt;br&gt;Activate Mode after CPU restart&lt;br&gt;Set Mode to: Automatic mode</td>
</tr>
<tr>
<td></td>
<td><strong>Input / output parameters</strong>&lt;br&gt;Setpoint:&lt;br&gt;Input&lt;br&gt;OutputHeat&lt;br&gt;Activate cooling&lt;br&gt;OutputCool:&lt;br&gt;OutputCool_PER (analog)&lt;br&gt;Cascade&lt;br&gt;Controller is master&lt;br&gt;Number of slaves: 1&lt;br&gt;Controller is slave&lt;br&gt;Master:</td>
</tr>
</tbody>
</table>

6. Go to the “Process value settings” tab. Adjust the limits for the process value to your application (in the example project 140 °C were selected for the upper limit and -10 °C for the lower limit).
5 Configuration and Commissioning of the PID_Temp Controller

5.2 PID_Temp configuration, triple-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
</table>
| 7.  | Go to the “Output settings” tab.  
The following settings were made in the example project:  
![Image of Output settings tab](image)  
The output value is set to the lower limit 1 and the upper limit +130. |

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
</table>
| 8.  | In the example project, the extended settings  
- process value monitoring  
- minimal on/off times for pulse width modulation  
- manual setting of PID parameters  
are not used.  
In order to influence the commissioning process, a controller structure for tuning can be set.  
When interconnecting the PID_Temp controller as a cascade, it is recommended using a pure PI structure as controller (see chapter 4.5).  
![Image of tuning rule](image) |
5 Configuration and Commissioning of the PID_Temp Controller

5.2 PID_Temp configuration, triple-loop

Center controller (#2)

For the configuration of the center controller, please follow the instructions in the table below.

Table 5-4

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>From the instructions in “Technology &gt; PID Control &gt; Compact PID” you add the “PID_Temp” instruction after the call of the outer controller.</td>
</tr>
</tbody>
</table>
| 2.  | Interconnect the controller.  
     | • Setpoint: with the output of the “outer controller”.  
     | • Input: with the sensor/output of the center part of the controlled system.  
     | • Master: with the “Slave” parameter of the “outer controller”.  
     | The interconnection of the parameters “Setpoint” and “Master” can be performed conveniently via the inspector window. |
| 3.  | Click on the “PID_Temp_2” writing and open the properties page in the inspector window.  
     | Go to the “Basic settings > Input/output parameters” tab.  
     | Select the following for the input: from the drop-down menu the “Input” value in order to activate the interconnected process value.  
     | Select the following for the output: from the drop-down menu the “OutputHeat” value. |
| 4.  | Go to the “Cascade” tab and activate the “Controller is master” checkbox. Also activate the “Controller is slave” checkbox and specify the master controller (here: PID_Temp_1). |
| 5.  | In the project tree of your project you go to the “Technology objects” folder. There, you double-click to open the configuration of the instruction “PID_Temp_2”. |
5 Configuration and Commissioning of the PID_Temp Controller
5.2 PID_Temp configuration, triple-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
</table>
| 6.  | Go the “Basic settings” tab. Enter the following settings:  
• Controller mode: Temperature  
  A temperature process shall be controlled.  
• Mode after CPU restart: Auto  
  The controlling is to start automatically after a cold start of the PLC (after a full loading or an MRES).  
• Input: Input  
  Already supplied in step 2 and 3:  
• OutputHeat: OutputHeat  
  The real output OutputHeat of the controller is used.  
• Deactivate the “Activate cooling” checkbox  
  The block works as master for the downstream PID_Temp.  
• Activate the “Controller is master” checkbox  
  The controller is the master for the downstream PID_Temp (#3).  
• Activate the “Controller is slave” checkbox  
  The controller is the slave for the downstream PID_Temp (#1). |

7.  | Go to the “Process value settings” tab. Adjust the limits for the process value to your application (in the example project 140 °C were selected for the upper limit and -10 °C for the lower limit). |
5 Configuration and Commissioning of the PID_Temp Controller

5.2 PID_Temp configuration, triple-loop

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td>Go to the “Output settings” tab. The following settings were made in the example project:</td>
</tr>
<tr>
<td></td>
<td><img src="Image1.png" alt="Image" /> The output value is set to the lower limit 1°C and the upper limit +130°C.</td>
</tr>
</tbody>
</table>

9. In the example project, the extended settings
   - process value monitoring
   - minimal on/off times for pulse width modulation
   - manual setting of PID parameters
   are not used. In order to influence the commissioning process, a controller structure for tuning can be set. When interconnecting the PID_Temp controller as a cascade, it is recommended using a pure PI structure as controller.

### Inner controller (#3)

For configuring the inner controller, follow the instructions in Table 5-2 with the modifications described below.

Modify step 4 by interconnecting the parameters as follows:

- **Setpoint**: with the output of the “center controller”.
- **Input**: with the sensor/output of the first part of the controlled system
- **Master**: with the “Slave” parameter of the “center controller”.

© Siemens AG 2019 All rights reserved
5 Configuration and Commissioning of the PID_Temp Controller

5.2 PID_Temp configuration, triple-loop

The interconnection at “Setpoint” and “Master” can be performed automatically via the inspector window.

Figure 5-2

Modify step no. 5 by selecting the “Slave” setting for the controller in “Cascade” (see Figure 5-3). Select the “center” controller (here: PID_Temp) as master after it has been configured.

Figure 5-3

Modify step no. 10 by selecting “PI” as the parameter for tuning. This prevents an excessive reaction of the control loop to changes of setpoint or actual value (see Chapter 4.5).

Figure 5-4
5.3 Commissioning

This chapter describes the commissioning process for a triple-loop PID_Temp control. Commissioning the single-loop PID_Temp controller is performed in the same way.

Prerequisite

In order to perform the commissioning process as described, the following requirements must be met:

- Calling the PID_Temp controller as single-instance (in order to be able to use the functions of the technology object).
- Correct configuration and interconnection analog to Chapter 5.2

Steps

The following procedure is generally suitable for commissioning:

1. Optimizing the controller “from inside out”.
2. Using the substitute setpoints for commissioning.
3. Tuning in two steps:
   a. Pretuning
   b. Fine tuning

Detailed procedure

The following table shows the detailed procedure for commissioning a triple-loop PID_Temp control by means of the example application.

Note

It is recommended to always perform the tuning of the control loop for the operating point of the controlled system

Tuning the controlled system in the example project refers to an operating point of 80 °C.

Table 5-5

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Download your project to the CPU.</td>
</tr>
<tr>
<td>2.</td>
<td>Open the commissioning page of the inner controller and go to the CPU online.</td>
</tr>
<tr>
<td>3.</td>
<td>Start measuring to be able to conveniently follow the tuning process.</td>
</tr>
</tbody>
</table>
5 Configuration and Commissioning of the PID.Temp Controller

5.3 Commissioning

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>Specify the value “23.0” as the ambient temperature via manual input.</td>
</tr>
</tbody>
</table>
| 5.  | Create the requirements in order to perform a first commissioning (see also online help of STEP 7). Activate the “Subst. Setpoint” checkbox for this and transfer a setpoint to the block which has a sufficient distance to the input value. In this example, the fictitious operating point “80.0”.

![Online status of controller diagram](image)

6. In the mask you select the tuning mode “Pretuning heating” and start tuning.

![Tuning mode](image)

7. Observe the progress bar to view the progress of the tuning process.

![Tuning status](image)

8. After “Pretuning heating” has been performed, select “Pretuning cooling” as the next step. A cooling pulse is sent to the system.
### 5.3 Commissioning

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td>Afterwards, select tuning mode “Fine tuning heating” and start fine tuning. “Fine tuning heating” attempts to oscillate the controlled system.</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Fine tuning heating" /></td>
</tr>
<tr>
<td>10.</td>
<td>After “Fine tuning heating” has been performed, select the “Fine tuning cooling” option from the drop-down menu. Starting “Fine tuning cooling” requires the “Tuning offset” in the fictitious operating point of the example application (80.0°C). For the tuning offset you select a value larger than the heating value in the operating point. Wait for the stabilization of the system and then start “Fine tuning cooling”.</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Tuning offset" /></td>
</tr>
<tr>
<td>11.</td>
<td>Deactivate the “Subst. Setpoint” checkbox</td>
</tr>
<tr>
<td>12.</td>
<td>Terminate the commissioning process for the inner controller by uploading the automatically detected PID parameters into your project.</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Upload PID parameters" /></td>
</tr>
<tr>
<td>13.</td>
<td>Optimize the center controller analog to steps 2-7, as well as 11+12. Perform “Pretuning heating” only.</td>
</tr>
<tr>
<td>14.</td>
<td>Optimize the outer controller analog to steps 2-7, as well as 11+12. Perform “Pretuning heating” only.</td>
</tr>
</tbody>
</table>
5.3 Commissioning

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.</td>
<td>Finally, test the optimized system by specifying setpoints or introducing disturbances.</td>
</tr>
</tbody>
</table>

Note

The graphic commissioning functions support you in the commissioning process. The functions should be used differently depending on the application case. The controller results can partly still be optimized by further considerations.

In the example project, the below tuning was performed:

Table 5-6

<table>
<thead>
<tr>
<th>Controller</th>
<th>Tuning</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID_Temp #1</td>
<td>Controller structure: PI&lt;br&gt;Tuning: Pretuning heating</td>
<td>Controller structure: Using the D-fraction of the PID controller results in a higher control activity, i.e. a “more agitated” control value signal. Since for cascaded control the control value signal is transferred to the subsequent controller as setpoint value, using the D-fraction does not always make sense here.</td>
</tr>
<tr>
<td>PID_Temp #2</td>
<td>Controller structure: PI&lt;br&gt;Tuning: Pretuning heating</td>
<td>Tuning: Fine tuning is mainly dimensioned for optimizing the disturbance behavior at the operating point for controllers which directly affect the process. For cascade control, a sufficient disturbance behavior is already achieved with the controller alignment and by executing the pretuning.</td>
</tr>
<tr>
<td>PID_Temp #3</td>
<td>Controller structure: PI&lt;br&gt;Tuning: Pretuning heating&lt;br&gt;Pretuning cooling&lt;br&gt;Fine tuning heating&lt;br&gt;Fine tuning cooling</td>
<td>Controller structure: The PID (temperature) control structure is optimized for single-loop temperature control. Tuning: All tuning modes are used to achieve the best result possible.</td>
</tr>
<tr>
<td>PID_Temp #4</td>
<td>Controller structure: PID (temperature)&lt;br&gt;Tuning: Pretuning heating&lt;br&gt;Pretuning cooling&lt;br&gt;Fine tuning heating&lt;br&gt;Fine tuning cooling</td>
<td></td>
</tr>
</tbody>
</table>
6 Installation and Commissioning of the Example Project

6.1 Installing the hardware

The figure below shows the hardware configuration of the application.

Figure 6-1

Note

The setup guidelines for SIMATIC systems must generally be followed.

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Connect your field PG directly with the S7-1500 using an Ethernet cable.</td>
</tr>
<tr>
<td>2.</td>
<td>Connect the S7-1500 CPU with the 24V connection of the power supply unit.</td>
</tr>
<tr>
<td>3.</td>
<td>Connect the power supply unit with a power pack.</td>
</tr>
</tbody>
</table>
6.2 Installing the software

Table 6-2

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Install STEP 7 on your field PG.</td>
</tr>
<tr>
<td>2.</td>
<td>Install WinCC Advanced on your field PG.</td>
</tr>
</tbody>
</table>

6.3 Commissioning

Table 6-3

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Use the display to set the IP address of the S7-1500 CPU to:&lt;br&gt;IP address: 172.16.46.33&lt;br&gt;Subnet mask: 255.255.0.0</td>
</tr>
<tr>
<td>2.</td>
<td>In &quot;Start &gt; Control Panel&gt; Network and Sharing &gt; Change adapter settings&quot; you set the IP address of the Ethernet adapter of the PG to:&lt;br&gt;IP address: 172.16.46.200&lt;br&gt;Subnet mask: 255.255.0.0</td>
</tr>
<tr>
<td>3.</td>
<td>Unzip the example application from the Online Support portal and open the &quot;103526819_PID_Temp_CODE_v11_d&quot; project.</td>
</tr>
<tr>
<td>4.</td>
<td>Compile the configuration of the S7-1500 CPU by right-clicking on the CPU and selecting the command &quot;Compile &gt; Hardware and software (only changes)&quot;.</td>
</tr>
<tr>
<td>5.</td>
<td>Load the project into the S7-1500 CPU Select the CPU and then &quot;Online &gt; Download and reset PLC program&quot;. Now select your access point to the S7-1500 CPU and then load the project into the CPU.</td>
</tr>
</tbody>
</table>
6. Installation and Commissioning of the Example Project

6.3 Commissioning

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image" alt="Image of PG/PC Interface Configuration" /></td>
</tr>
</tbody>
</table>

6. In “Windows > Control Panel > PG/PC Interface” you check whether the PG/PC interface is set to “TCP/IP” or “Auto”.

7. Start the runtime simulation.
   You can then see the start screen of the application.
7 Operation of the WinCC User Interface

7.1 Overview

Overview and description of the user interface

Figure 7-1

Three pictures exist for the application. To the left of the curves, a legend for understanding the graphs is given.
### 7 Operation of the WinCC User Interface

#### 7.1 Overview

**Table 7-1**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Displayed values</th>
</tr>
</thead>
</table>
| **Comparison** | The picture simultaneously displays the triple-loop and the single-loop control. Using the tags “Setpoint” and “Disturbance” enables changing the setpoint and the disturbance effect, as well as monitoring the behavior of the controlled systems via curves for both simulations at the same time. | • Setpoint value  
Single-loop:  
• Temperature of chocolate  
• Heating value  
• Cooling value  
Triple-loop:  
• Temperature of chocolate  
• Heating value (Controller_3)  
• Cooling value (Controller_3) |
| **One controller** | The picture shows the behavior of the single-loop control.  
You can use “Setpoint” and “Disturbance” for changing setpoint and disturbance effect.                                                                 | Single-loop:  
• Setpoint value  
• Temperature of water  
• Temperature of steel  
• Temperature of chocolate  
• Heating value  
• Cooling value |
| **Cascade control** | The picture shows the behavior of the triple-loop control.  
You can use “Setpoint” and “Disturbance” for changing setpoint and disturbance effect.                                                                 | Triple-loop:  
• Setpoint value  
• Temperature of chocolate  
• Output Controller_1  
• Output Controller_2  
• Temperature of water  
• Temperature of steel  
• Heating value (Controller_3)  
• Cooling value (Controller_3) |
7 Operation of the WinCC User Interface

7.2 Scenario: setpoint step

7.2 Scenario: setpoint step

The scenario shows and compares the behavior of both controlled systems during a setpoint step.

Table 7-2

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Commission the example application as described in Chapter 6.</td>
</tr>
<tr>
<td>2.</td>
<td>Click on “Application example”.</td>
</tr>
<tr>
<td>3.</td>
<td>Another picture opens. The default setpoint value is 23.0. Change the setpoint to 60.0.</td>
</tr>
<tr>
<td>4.</td>
<td>Wait till the setpoint is reached. Generate afterwards an setpoint step by writing the setpoint with the value 80.0</td>
</tr>
<tr>
<td>5.</td>
<td>Now you can use both curves to monitor the behavior of the controlled systems.</td>
</tr>
</tbody>
</table>

Note:
The controlled single-loop system oscillates stronger than the controlled triple-loop system. However, the setpoint is first reached in a shorter time.
7.3 Scenario: disturbance variable impact

The scenario shows and compares the behavior of both controlled systems when a disturbance variable impacts the simulated “water” process.

Table 7-3

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Commission the example application as described in Chapter 6.</td>
</tr>
<tr>
<td>2.</td>
<td>The default setpoint value is 23.0. Change the setpoint to 80.0.</td>
</tr>
<tr>
<td>3.</td>
<td>Wait until the actual value has levelled off at the setpoint.</td>
</tr>
<tr>
<td>4.</td>
<td>Now connect a disturbance variable of +40.0, for example.</td>
</tr>
<tr>
<td>5.</td>
<td>Now you can use both curves to monitor the behavior of the controlled systems.</td>
</tr>
</tbody>
</table>

Note

For cascade control, the disturbance is already corrected by the inner control loop and therefore hardly affects the temperature of the chocolate. For single-loop control, the controller only reacts when the effect of the disturbance causes a deviation in the temperature of the chocolate. This can cause an oscillation of the actual value (of the chocolate temperature).
8 Related Literature

Book directory
Table 8-1

<table>
<thead>
<tr>
<th>Topic</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlling with SIMATIC</td>
<td>Practice book for SIMATIC S7 and SIMATIC PCS7 control systems</td>
</tr>
<tr>
<td></td>
<td>Authors: Müller/ Pfeiffer/ Wieser</td>
</tr>
<tr>
<td></td>
<td>Publicis Publishing, Erlangen</td>
</tr>
<tr>
<td></td>
<td>ISBN: 978-3-89578-340-1</td>
</tr>
</tbody>
</table>

Link directory
Table 8-2

<table>
<thead>
<tr>
<th>Topic</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link to this document</td>
<td><a href="http://support.automation.siemens.com/WW/view/en/103526819">http://support.automation.siemens.com/WW/view/en/103526819</a></td>
</tr>
<tr>
<td>Siemens Industry Online Support</td>
<td><a href="http://support.automation.siemens.com">http://support.automation.siemens.com</a></td>
</tr>
<tr>
<td>SIMATIC STEP 7 Basic/Professional V15.1 and SIMATIC WinCC V15.1</td>
<td><a href="https://support.industry.siemens.com/cs/ww/en/view/109755202">https://support.industry.siemens.com/cs/ww/en/view/109755202</a></td>
</tr>
<tr>
<td>PID Control with PID_Compact (S7-1200)</td>
<td><a href="http://support.automation.siemens.com/WW/view/de/100746401">http://support.automation.siemens.com/WW/view/de/100746401</a></td>
</tr>
<tr>
<td>Controlling simulated controlled systems in S7-1500 with PID_Compact V2</td>
<td><a href="http://support.automation.siemens.com/WW/view/en/79047707">http://support.automation.siemens.com/WW/view/en/79047707</a></td>
</tr>
<tr>
<td>SIMATIC S7-1200, S7-1500 PID control</td>
<td><a href="http://support.automation.siemens.com/WW/view/en/108210036">http://support.automation.siemens.com/WW/view/en/108210036</a></td>
</tr>
</tbody>
</table>

9 History
Table 9-1

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>01/2015</td>
<td>First version</td>
</tr>
<tr>
<td>V1.1</td>
<td>08/2019</td>
<td>Update TIA Portal V15.1</td>
</tr>
</tbody>
</table>