

Application Description • 11/2014

# PID Control with PID\_Compact

SIMATIC S7-1200

## Warranty and Liability

### Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use sound practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change in the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

### Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly to regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit <http://www.siemens.com/industrialsecurity>.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit <http://support.automation.siemens.com>.

# Table of Contents

	<b>Warranty and Liability .....</b>	<b>2</b>
<b>1</b>	<b>Task.....</b>	<b>4</b>
1.1	Overview .....	4
<b>2</b>	<b>Solution.....</b>	<b>5</b>
2.1	Overview .....	5
2.2	Description of the core functionality.....	6
2.3	Hardware and software components.....	7
2.3.1	Validity .....	7
2.3.2	Components used .....	7
<b>3</b>	<b>Function Mechanisms .....</b>	<b>10</b>
3.1	General overview.....	10
3.2	Main [OB1] .....	11
3.2.1	HMI [FC1].....	11
3.3	Cyclic interrupt [OB200].....	11
3.3.1	Switch [FC5].....	12
3.3.2	PID_Compact [FB1130].....	13
3.3.3	Simulation [FC2].....	16
<b>4</b>	<b>Installation and Commissioning .....</b>	<b>21</b>
4.1	Hardware adaptation .....	21
4.1.1	Input signal .....	21
4.1.2	Output signal .....	22
4.2	Configuration guide.....	24
4.3	Commissioning the compact controller.....	30
4.4	HMI project part.....	32
4.4.1	Configuring the HMI.....	32
	Loading the HMI project part into the KTP900 Basic.....	33
	Starting PC runtime .....	34
<b>5</b>	<b>Operation of the Application.....</b>	<b>35</b>
5.1	Overview .....	35
5.1.1	Overview (start screen).....	35
5.1.2	Trend view.....	37
5.1.3	Tuning.....	39
5.1.4	Monitoring .....	42
5.1.5	Alarm view.....	43
5.1.6	Configuration .....	44
5.1.7	Simulation .....	47
5.1.8	Settings .....	48
<b>6</b>	<b>Related Literature.....</b>	<b>50</b>
<b>7</b>	<b>History .....</b>	<b>50</b>

# 1 Task

## 1.1 Overview

### Introduction

Influencing technical variables in systems requires controlling these variables. In automation technology, controllers are also used in many different ways, for example for speed control.

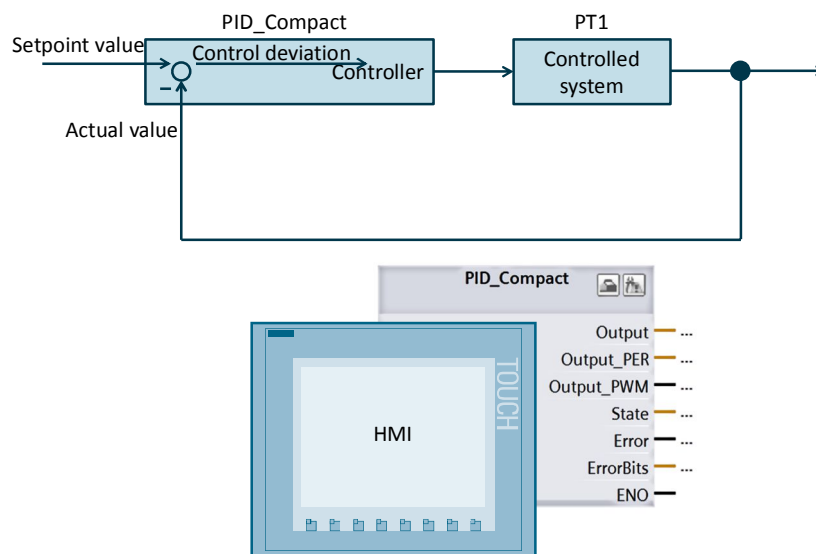
For SIMATIC S7-1200, from CPU firmware V4, the “PID\_Compact” (version 2.2) technology object will be provided for actuators operating proportionally.

### Overview of the automation task

The automation task is to set up a control loop for influencing physical parameters in a technical process, the control loop consisting of the following elements:

- “PID\_Compact” as the controller
- Simulated technical processes as controlled system

Figure 1-1



### Description of the automation task

The application must meet the following requirements:

- Configuration and settings of the software controller (“PID\_Compact” block) should be explained.
- Options for optimizing the “PID\_Compact” should be shown.
- The controlling process should be operated and monitored via HMI.

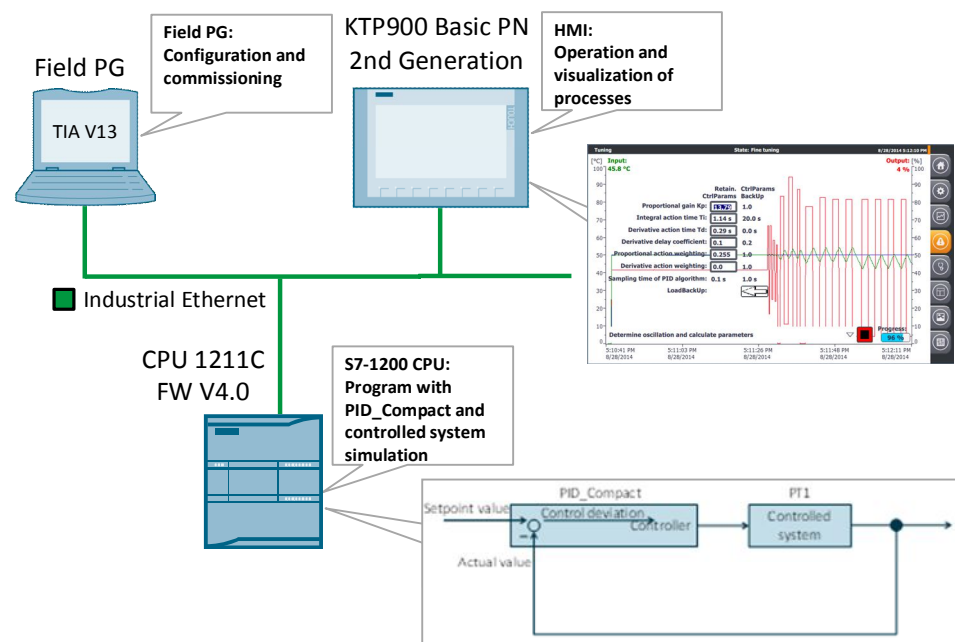
## 2 Solution

### 2.1 Overview

#### Display

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

Figure 2-1



Within a control loop (here the controlled system PT1), the “PID\_Compact” technology object continuously detects the measured actual value and compares this value to the setpoint value (specified via the HMI). From the resulting control deviation, the controller calculates an output value which enables the actual value to reach the setpoint value as quickly and stably as possible.

The output value of the PID controller consists of three values:

- **P-term**  
The P-term of the output value rises proportionally to the control deviation.
- **I-term**  
The I-term of the output value rises until the control deviation is compensated.
- **D-term**  
The D-term rises with increasing modification speed of the control deviation. The actual value is adapted to the setpoint value as quickly as possible. If the modification speed of the control deviation decreases again, the D-term will also decrease.

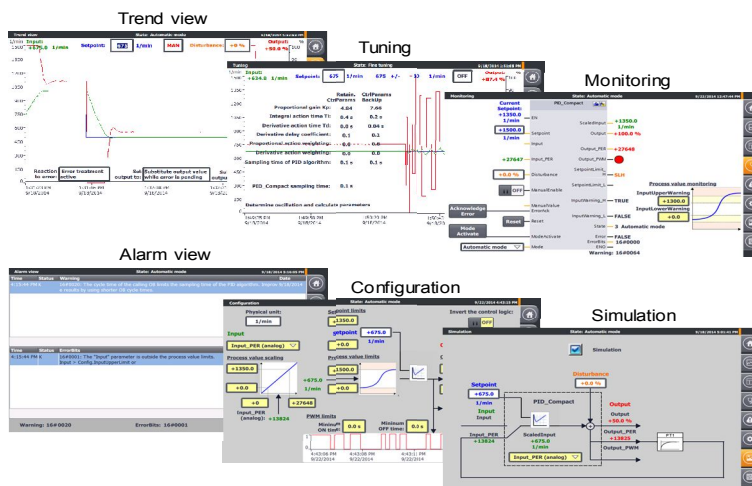
The “PID\_Compact” instruction automatically calculates the P, I, and D parameters for your controlled system during pretuning. The parameters can be further optimized through fine tuning. There is no need to manually determine the parameters.

## 2.2 Description of the core functionality

The core functionality of the application is the operation of the “PID\_Compact” technology object via the HMI.

### Overview and description of the user interface

Figure 2-2



The application is operated through the following 6 screens:

- Trend view
- Tuning
- Monitoring
- Alarm view
- Configuration
- Simulation

The operation of the user interfaces will be described in detail in chapter [Operation of the Application](#).

### Advantages of this solution

The application enables you to use any configuration options and commissioning features via a KTP 900 Basic PN 2nd Generation operator panel or via the HMI simulation integrated in WinCC V13.

This application offers the following advantages:

- Switching between automatic and manual mode
- Trend curves of setpoint, actual value and manipulated variable
- Switching between the real controlled system and the simulation
- Disturbance value control in simulation mode
- Specification of the behavior in case of errors and their simulation
- Manual specification of control parameters and automatic tuning
- Online monitoring of the “PID\_Compact” controller block
- Modifying the configuration during runtime

#### Delimitation

This application gives you an overview of the “PID\_Compact” technology object for commissioning with SIMATIC S7-1200.

You can apply the application example to operate your controller conveniently via a KTP 900 Basic PN 2nd Generation and adapt it to your automation task.

The application was tested by simulating the controlled system.

For real operation you have to adapt the application example to the actuator and the actual value sensor you are using:

- Analog control or control via a digital output by means of the pulse-width-modulated signal?
- Required voltage and power for the control?
- Signal properties of the temperature sensor used.

The application is no substitute for the configuration screen of the PID\_Compact wizard, since through the wizard the default values in the instance data block are defined, which are decisive for the restart after a power failure.

Apart from the “PID\_Compact” control block, STEP 7 V1x also provides the “PID\_3Step” for the SIMATIC S7-1200 - a three-point stepper controller with automatic tuning for valves or actuators with integrating behavior ([\10\](#)).

#### Note

Further information on technology object “PID\_3Step” is available in the

- S7-1200 Manual ([\3\](#)) → [Chapter 9.2.4](#) and
- in the STEP 7 Basic V13 Manual ([\6\](#)) → [Chapter 11.1.4](#).

#### Required knowledge

Basic knowledge of control engineering is assumed.

## 2.3 Hardware and software components

### 2.3.1 Validity

This application is valid for

- STEP 7 V13 or higher
- S7-1200 CPU Firmware V4.0 or higher
- “PID\_Compact” V2.2 technology object

### 2.3.2 Components used

The application was created with the following components:

## 2 Solution

### 2.3 Hardware and software components

#### Hardware components

Table 2-1

Component	Qty.	Article number	Note
SIMATIC HMI KTP900 BASIC	1	6AV2123-2JB03-0AX0	Optional (can also be simulated via WinCC V13 Basic Runtime)
COMPACT SWITCH MODULE CSM 1277	1	6GK7277-1AA10-0AA0	
POWER SUPPLY S7-1200 PM1207	1	6EP1332-1SH71	
CPU 1211C, DC/DC/DC, 6DI/4DO/2AI	1	6ES7211-1AE40-0XB0	Firmware V4.0
Fan/motor with analog speed control (0 to 10V / 0 to 20 mA)	1	Manufacturer of fan/motor	- Without integrated speed control electronics - Optionally with integrated actual speed feedback
INCREMENTAL ENCODER WITH HTL 1000 I/U, OPERATING VOLT. 10-30V CLAMPING FLANGE, SHAFT 10 MM FLANGE SOCKET RADIAL	1	e.g.: 6FX2001-4QB00	Optional, if fan/motor does not provide an integrated actual speed feedback
SIGNAL BOARD SB 1232, 1 AQ, (12 bit resolution)	1	6ES7232-4HA30-0XB0	Optional (if the fan/motor is controlled with a current output of 0 to 20 mA)
Programming unit	1		With Ethernet connector
Ethernet line TP CORD RJ45/RJ45 2M	3	6XV1870-3QH20	
Miniature circuit breaker	1	5SX2116-6	1-pin B, 16A
Standard mounting rail	1	6ES5 710-8MA11	35 mm

#### Software components

Table 2-2

Component	Qty.	Article number	Note
SIMATIC STEP 7 Basic V13	1	6ES7822-0AA03-0YA5	- includes WINCC Basic V13; - with Update 5 ( <a href="#">18\</a> )
SIMATIC STEP 7 Basic Upgrade V13	1	6ES7822-0AA03-0YE5	(Optional) Only for upgrade from STEP 7 Basic V11 or V12



## 2 Solution

---

### 2.3 Hardware and software components

#### Sample files and projects

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

Table 2-3

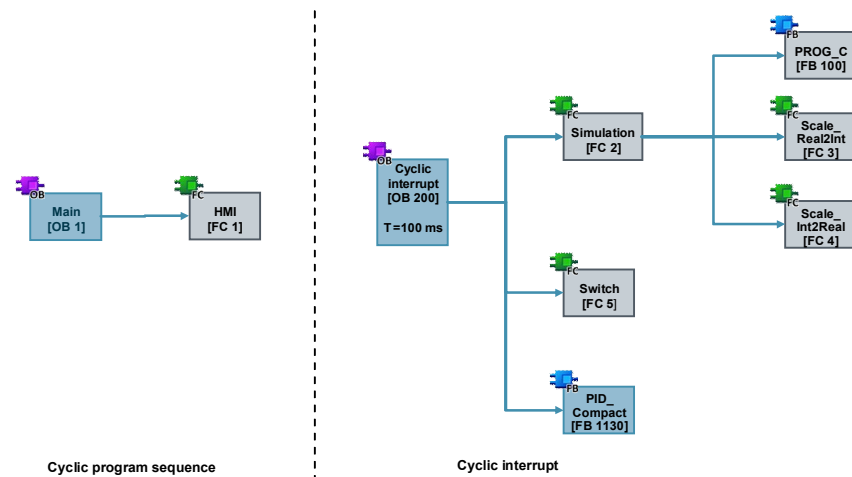
Component	Note
100746401_S7-1200_PID_Compact_CODE_v1d0.zip	<This zip file contains the STEP 7 project.>
100746401_S7-1200_PID_Compact_DOKU_v1d0_de.pdf	This document.

## 3 Function Mechanisms

### 3.1 General overview

Figure 3-1 shows the sequence of block calls in the control unit of the application project.

Figure 3-1



The control project part consists of the organization blocks:

- Main [OB1], from where the function for the HMI transfer is called
- Cyclic interrupt [OB200], which calls the compact controller cyclically every 100 milliseconds through the simulation blocks.

The parameters are transferred between the functions with instance data blocks:

- PID\_Compact\_1 [DB1130]
- PROG\_C\_DB [DB100]

as well as the data blocks:

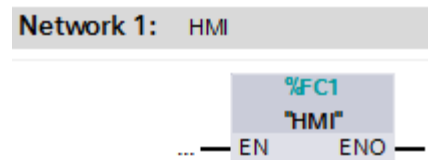
- tags [DB1] (contains all tags not required for the simulation of the controlled system)
- Simulation\_tags [DB2] (contains all tags required for the simulation of the controlled system).

## 3.2 Main [OB1]

The function for the HMI transfer is called from the “Main” organization block.

### 3.2.1 HMI [FC1]

Figure 3-2



The “HMI” function defines tags which the operator panel requires for the display animation of objects and elements.

Further descriptions are available in the network headers.

## 3.3 Cyclic interrupt [OB200]

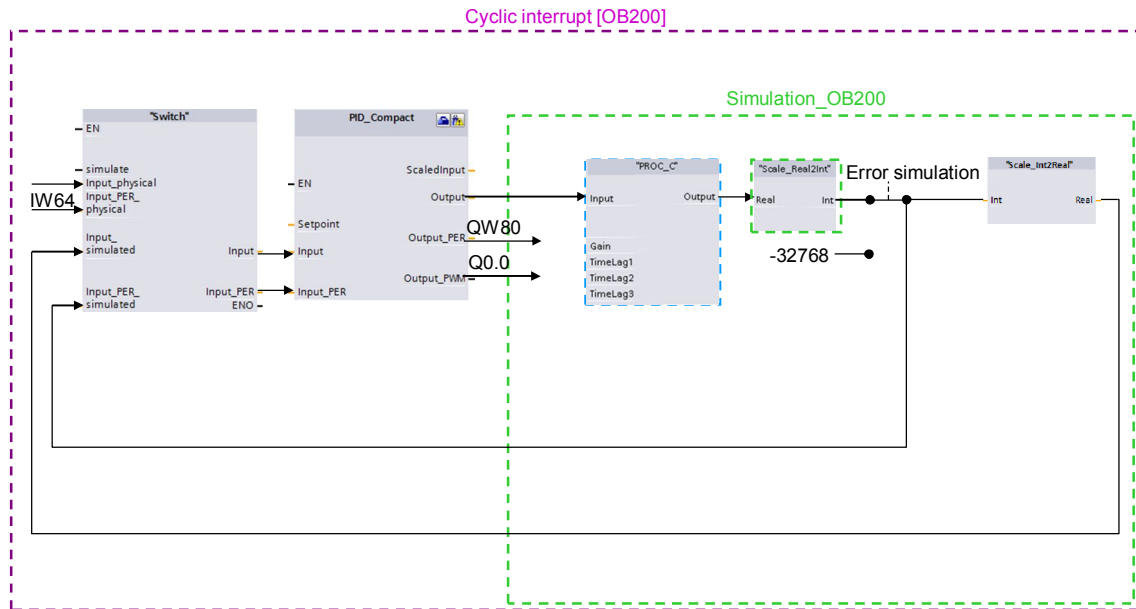
The actual program (the call of the “PID\_Compact” compact controller) takes place in the cyclic interrupt OB, since discrete software controls must be called in a defined time interval for optimizing the controller quality.

100ms were set as a constant time interval for the scan time of OB200.

### Program overview

The entire simulated control circuit is calculated in the cyclic interrupt OB.

Figure 3-3



### Configuration explanations

The “Switch” function enables you to switch between a real controlled system (signal evaluation via the control periphery) or a simulation of the controlled system.

The selected signals are then transferred to the “PID\_Compact” compact controller as input parameters. From the control deviation = setpoint - actual value, depending on the PID parameters, the compact controller calculates the manipulated variable which is transferred as a pulse-width-modulated signal to the I/O control outputs on an analog or digital basis.

The manipulated variable is transferred to the “PROC\_C” block as a floating-point number.

The “PROC\_C” block simulates a PT1 system behavior and outputs the actual value as a floating-point number, which is converted into an analog value via “Scale\_Real2Int”.

In the error simulation, the actual value is overwritten with the incorrect value (-32768) and transferred to the simulated analog input “Input\_PER\_simulated” of the “Switch” block.

In addition, the simulated analog value is converted into the corresponding floating-point number for the “Input\_simulated” input via “Scale\_Int2Real”.

#### 3.3.1 Switch [FC5]

The “Switch” function is used for switching between the signal evaluation via the control I/O and the calculated simulated input signals for transfer to the “PID\_Compact” compact controller.

Figure 3-4

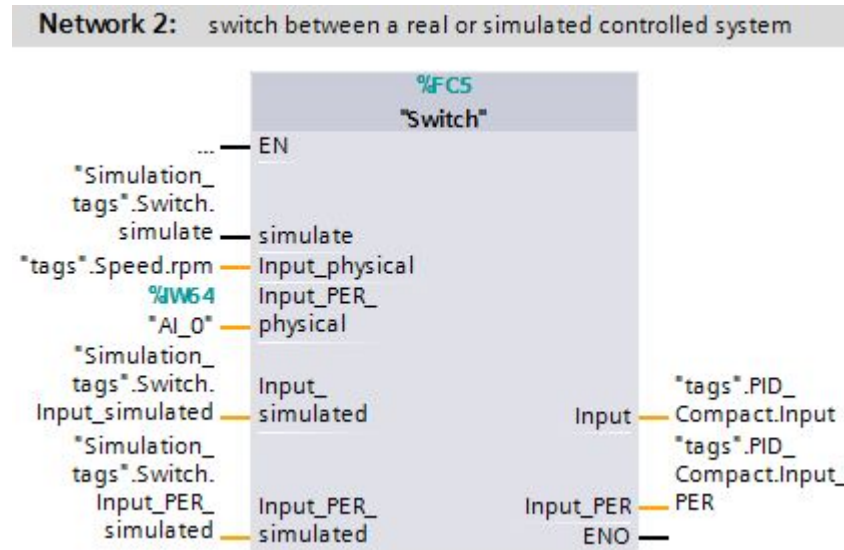


Table 3-1

	Name	Data type	Description
Input	simulate	Bool	FALSE = physical input signals TRUE = simulated input signals are transferred to the outputs
	Input_physical	Real	Actual value I/O input signal as floating-point number
	Input_PER_physical	Int	Analog actual value I/O input signal
	Input_simulated	Real	Simulated actual value input signal as floating-point number
	Input_PER_simulated	Int	Simulated analog actual value input signal
Output	Input	Real	Process value transfer to PID_Compact
	Input_PER	Int	Analog process value transfer to PID_Compact

**Note**

All inputs must be assigned (even if some are not required due to the controller configuration).

### 3.3.2 PID\_Compact [FB1130]

STEP 7 V13 provides the "PID\_Compact" version 2.2 technology object with the installation.

This function block was especially developed for controlling actuators acting proportionally.

Figure 3-5

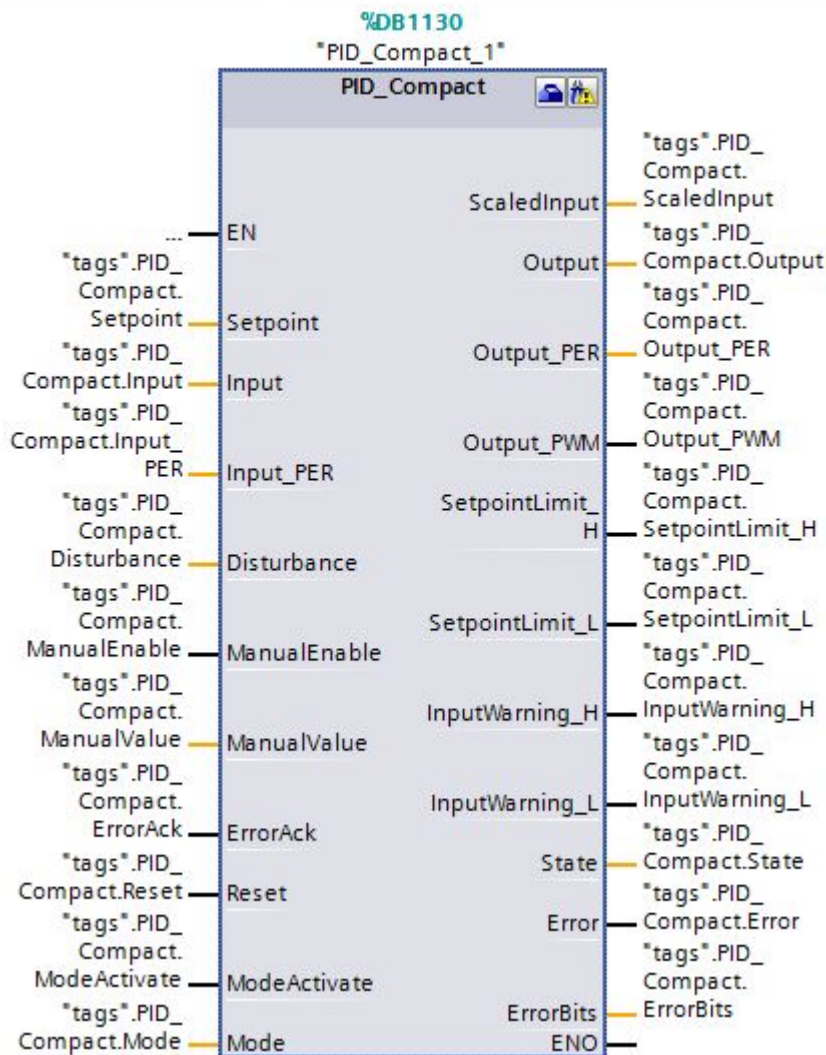
**Network 5:** PID\_Compact

Table 3-2

	Name	Data type	Description
Input	Setpoint	Real	Setpoint input
	Input	Real	Current actual value in REAL
	Input_PER	Int	Current actual value from I/O
	Disturbance	Real	Disturbance connection
	ManualEnable	Bool	Manual value is activated for overwriting the manipulated value output
	ManualValue	Real	Manual value
	ErrorAck	Bool	Clearing the error message
	Reset	Bool	Reset, controller restart
	ModeActivate	Bool	Enable mode
Output	ScaledInput	Real	Scaled I/O actual value
	Output	Real	Output value in REAL
	Output_PER	Int	Output value to I/O
	Output_PWM	Bool	Pulse-width modulated output value

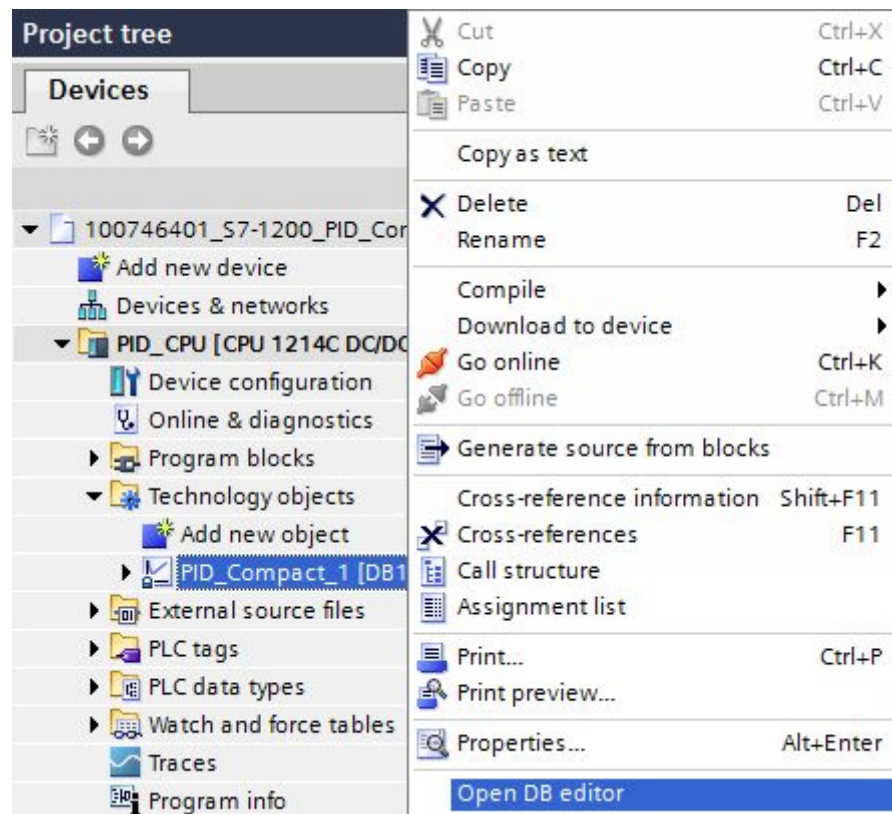
### 3 Function Mechanisms

#### 3.3 Cyclic interrupt [OB200]

	Name	Data type	Description
	SetpointLimit_H	Bool	Setpoint is restricted to the upper limit
	SetpointLimit_L	Bool	Setpoint is restricted to the lower limit
	InputWarning_H	Bool	Actual value exceeded the upper warning limit
	InputWarning_L	Bool	Actual value fell short of the lower warning limit
	State	Int	Display of controller state (0=inactive, 1=SUT, 2=TIR, 3=automatic, 4=manual)
	Error	Bool	Error flag
	ErrorBits	DWord	Error message
InOut	Mode	Int	Mode selection

The "PID\_Compact" controller is called in the "Cyclic interrupt" (OB200). You will find the instance data block DB1130 for "PID\_Compact" in the "Technology objects" folder:  
It can be opened via right-click -> "Open DB editor".  
Apart from the inputs and outputs, the application also accesses the static tags of "PID\_Compact\_1".

Figure 3-6

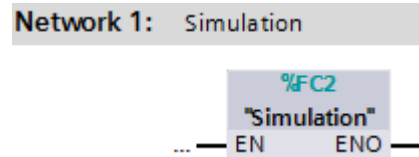


#### Note

A detailed description of the compact controller is available in the STEP 7 V13 online help. Select function block "PID\_Compact" in the program call (see Figure 3-5) and press F1.

#### 3.3.3 Simulation [FC2]

Figure 3-7



From the “Simulation” function, all functions required for simulating the controlled system are called:

- PROC\_C [FB100]
- Scale\_Real2Int [FC3]
- Scale\_Int2Real [FC4]

The “Simulation” is called in the same cyclic interrupt as the “PID\_Compact” compact controller.

Further information is available in the network headers as well as in the description below.



**PROC\_C [FB100]**

Function block "PROC\_C" simulates the continuous behavior of a PT3 controlled system.

Figure 3-8

**Network 1:** call of the simulation of the pt1 element

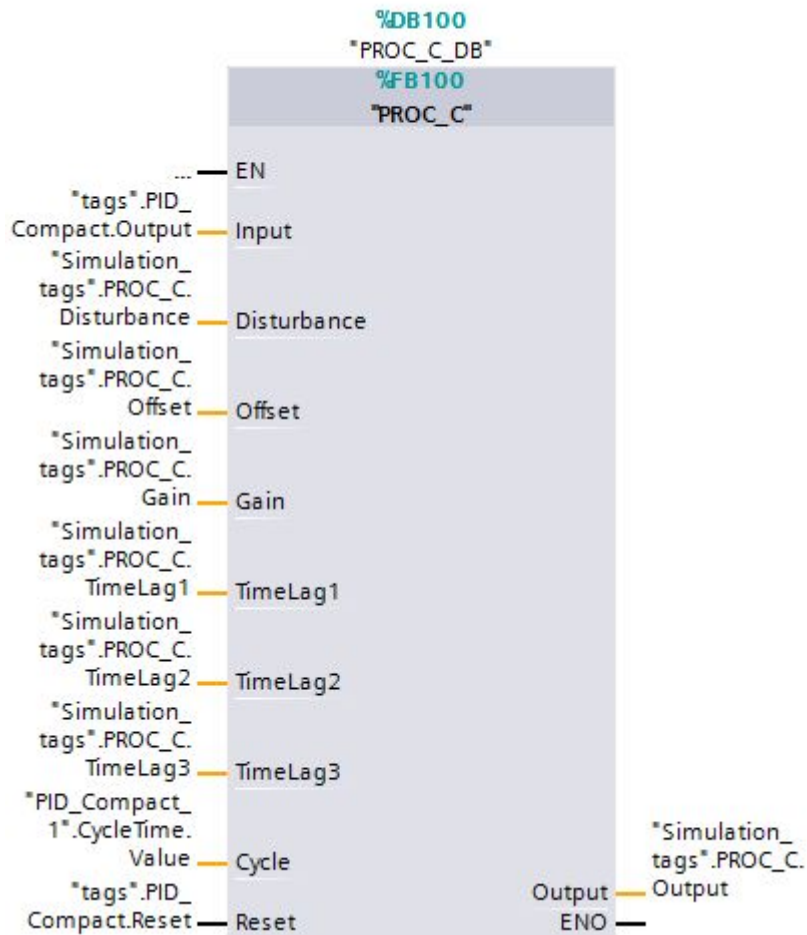


Table 3-3

	Name	Data type	Description
Input	Input	Real	Input value of the simulation of the controlled system
	Disturbance	Real	Disturbance value
	Offset	Real	Output offset
	Gain	Real	Gain factor
	TimeLag1	Real	Time lag 1 (deactivation for TimeLag1=0.0)
	TimeLag2	Real	Time lag 2 (deactivation for TimeLag2=0.0)
	TimeLag3	Real	Time lag 3 (deactivation for TimeLag3=0.0)
	Cycle	Real	Call interval of the cyclic interrupt
	Reset	Bool	Reset input
Output	Output	Real	Calculated output value of the simulation of the controlled system

Calculation of the output value is based on the following formula:

$$Output = \frac{Gain \cdot (Input + Disturbance)}{(TimeLag1 \cdot s + 1) \cdot (TimeLag2 \cdot s + 1) \cdot (TimeLag3 \cdot s + 1)} + Offset$$

s = Laplace operator

In this application, controlled system simulation block "PROC\_C" is designed as controlled system PT1 with a time lag of 3 seconds ("TimeLag2" and "TimeLag3" are disabled).

### Scale\_Real2Int [FC3]

The "Scale\_Real2Int" function is used for linear conversion of a floating-point number (data type: Real) into an analog value (data type: Int) within predefined boundaries.

Figure 3-9

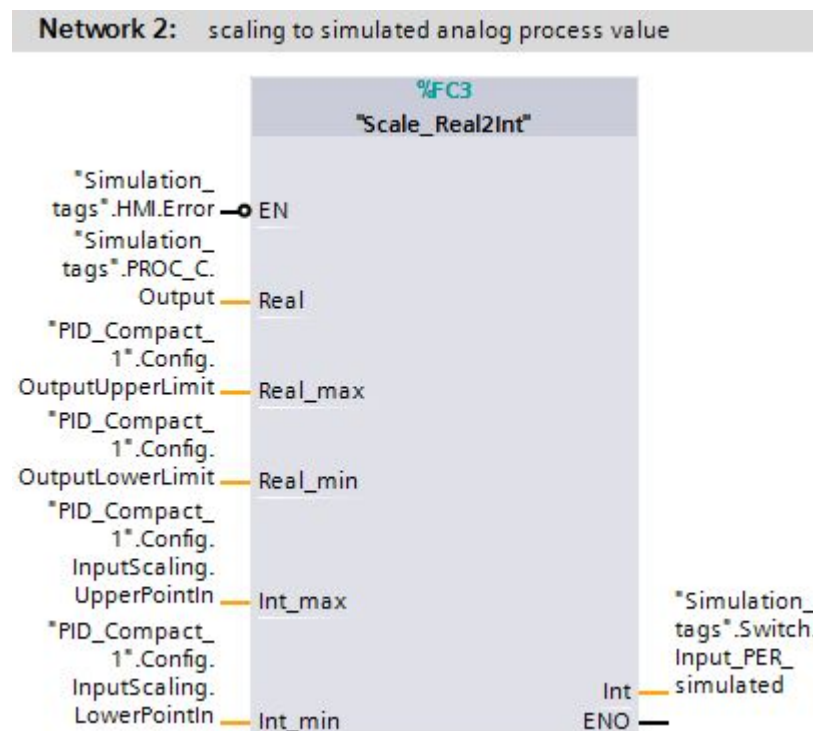


Table 3-4

	Name	Data type	Description
Input	Real	Real	Floating-point input value to be converted
	Real_max	Real	Upper limit of the floating-point input value
	Real_min	Real	Lower limit of the floating-point input value
	Int_max	Real	Upper limit of the analog output value
	Int_min	Real	Lower limit of the analog output value
Output	Int	Int	Analog output value

The specified output limits "Int\_max" and "Int\_min" were intentionally defined as "Real", in order to guarantee the compatibility with the specified limits in the "PID\_Compact" instance data block.

Converting the controlled system output into an analog value is necessary to be able to simulate the behavior in the case of an error.

For a real controlled system, an error occurs if the actual value sensor fails (e.g. due to wire break).

In the simulation this is achieved by overwriting the analog actual value with a value outside the measuring range (-32768) (see Figure 3-3).

#### Scale\_Int2Real [FC3]

The "Scale\_Int2Real" function is used for linear conversion of an analog value (data type: Int) into a floating-point number (data type: Real) within predefined boundaries.

Figure 3-10

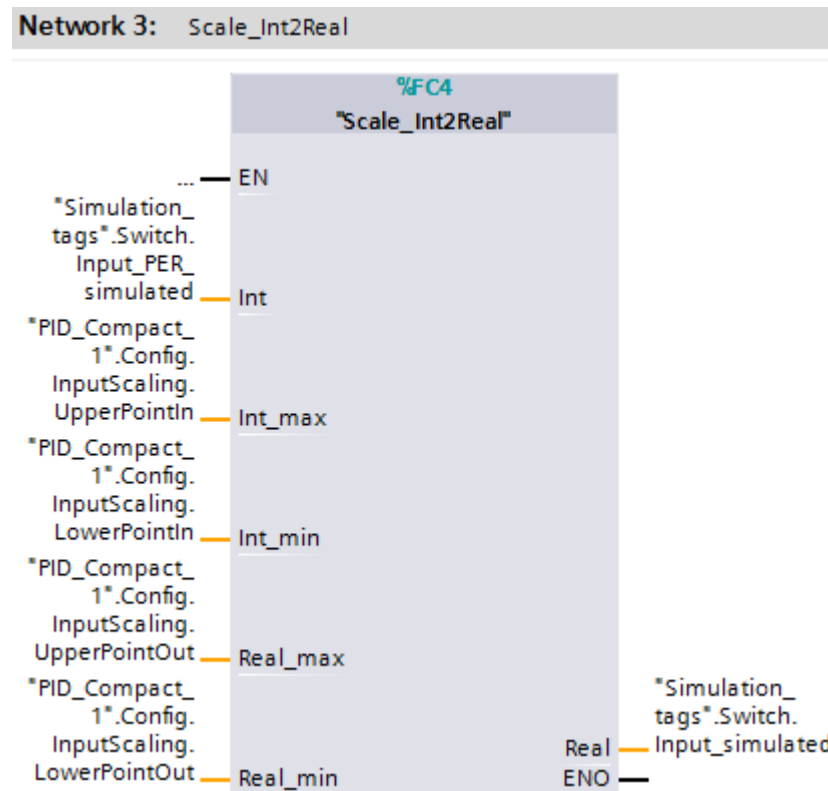


Table 3-5

	Name	Data type	Description
Input	Int	Int	Analog value to be converted
	Int_max	Real	Upper limit of the analog value
	Int_min	Real	Lower limit of the analog value
	Real_max	Real	Upper limit of the floating-point output value
	Real_min	Real	Lower limit of the floating-point output value
Output	Real	Real	Floating-point output value

**Note**

The specified input limits "Int\_max" and "Int\_min" were intentionally defined as "Real", in order to guarantee the compatibility with the specified limits in the "PID\_Compact" instance data block.

Converting the simulated analog actual value into the simulated floating-point actual value is necessary to be able to simulate the behavior in the case of an error even if the actual value "input" is selected.

Therefore, even if the actual value "Input" is selected, the overwriting of the analog actual value with the value -32768 in case of error will be applied (see Figure 3-3).

## 4 Installation and Commissioning

### 4.1 Hardware adaptation

This application has been realized with a CPU of the SIMATIC S7-1200 product family. Each S7-1200 CPU has 2 integrated analog inputs for receiving voltage signals from 0 to 10 V.

Depending on the design of your selected actuator, the hardware configuration of your S7-1200 might need adjustment.

The configuration options of S7-1200 for operating the "PID\_Compact" compact controller are introduced below.

#### 4.1.1 Input signal

The controlled variable is acquired as an adapted floating-point number "Input" or as an analog value from the "Input\_PER" I/O. The "PID\_Compact" offers the conversion of the analog value into the physical unit in the configuration screen.

The modules for the analog value acquisition are listed below.

#### Controlled variable acquisition

Table 4-1

Analog inputs	Article number	Type		Resolution
		Voltage	Current	
2 x onboard	all CPUs	0 to 10V	no	10 bits
<b>SB</b> 1231 AI 1 x 12 bit	6ES7231-4HA30-0XB0	±10V, ±5V, ±2.5V	0 to 20 mA	11 bits + sign
<b>SM</b> 1231 AI 4 x 13 bit	6ES7231-4HD32-0XB0	±10V, ±5V, ±2.5V	0 to 20 mA 4 to 20 mA	12 bits + sign
<b>SM</b> 1231 AI 8 x 13 bit	6ES7231-4HF32-0XB0	±10V, ±5V, ±2.5V	0 to 20 mA 4 to 20 mA	12 bits + sign
<b>SM</b> 1234 AI 4 x 13 bit / AO 2 x 14 bit	6ES7234-4HE32-0XB0	±10V, ±5V, ±2.5V	0 to 20 mA 4 to 20 mA	12 bits + sign
<b>SM</b> 1231 AI 4 x 16 bit	6ES7231-5ND32-0XB0	±10V, ±5V, ±2.5V, ±1.25V	0 to 20 mA 4 mA to 20 mA	15 bits + sign

#### Controlled variable acquisition (temperature)

Table 4-2

Thermocouple	Article number	Type	Resolution	
		Temperature / Voltage	Temperature	Voltage
<b>SB</b> 1231 AI 1 x 16 bit TC	6ES7231-5QA30-0XB0	Electrically isolated, TC and mV	0.1 °C/0.1 °F	15 bits + sign
<b>SM</b> 1231 AI 4 x 16 bit TC	6ES7231-5QD32-0XB0	Electrically isolated, TC and mV	0.1 °C/0.1 °F	15 bits + sign
<b>SM</b> 1231 AI 8 x 16 bit TC	6ES7231-5QF32-0XB0	Electrically isolated, TC and mV	0.1 °C/0.1 °F	15 bits + sign

## 4 Installation and Commissioning

### 4.1 Hardware adaptation

Table 4-3

Resistance thermometer	Article number	Type	Resolution	
		Temperature / resistance	Temperature	Resistance
<b>SB</b> 1231 AI 1 x 16 bit RTD	6ES7231-5PA30-0XB0	Module reference RTD and $\Omega$	0.1 °C/0.1 °F	15 bits + sign
<b>SM</b> 1231 AI 4 x RTD x 16 bit	6ES7231-5PD32-0XB0	Module reference RTD and $\Omega$	0.1 °C/0.1 °F	15 bits + sign
<b>SM</b> 1231 AI 8 x RTD x 16 bit	6ES7231-5PF32-0XB0	Module reference RTD and $\Omega$	0.1 °C/0.1 °F	15 bits + sign

#### 4.1.2 Output signal

The “PID\_Compact” controller provides the control of the actuator via an analog output or via a digital pulse-width modulated output.

#### Analog outputs

Table 4-4

Analog outputs	Article number	Type	
		Voltage (resolution)	Current (resolution)
CPU 1215C/ CPU 1217C	all controllers 1215C/1217C	no	0 to 20 mA (10 bits)
<b>SB</b> 1232 AO 1 x 12 bit	6ES7232-4HA30-0XB0	±10 V (12 bits)	0 to 20 mA (11 bits)
<b>SM</b> 1232 AO 2 x 14 bit <b>SM</b> 1232 AO 4 x 14 bit <b>SM</b> 1234 AI 4 x 13 bit / AO 2 x 14 bit	6ES7232-4HB32-0XB0 6ES7232-4HD32-0XB0 6ES7234-4HE32-0XB0	±10 V (14 bits)	0 to 20 mA or 4 to 20 mA (13 bits)

#### Digital outputs

Depending on the power consumption of your digital valve control, you can choose between S7-1200 controllers with transistor or relay outputs:

Table 4-5

Digital outputs	Voltage range	Current (max.)
Transistor	20.4 to 28.8 V	0.5 A
Relay	5 to 250 V	2 A

#### Note

**SB** = signal board (each CPU has a slot for a signal board)  
**SM** = signal module

- up to 2 signal modules can be connected to CPU 1212C
- up to 8 signal modules can be connected to CPU 1214C/1215C/1217C.

Further information on the selection and wiring of peripheral components is available in [Chapter A “Technical Specifications”](#) in the S7-1200 manual ([13](#)).

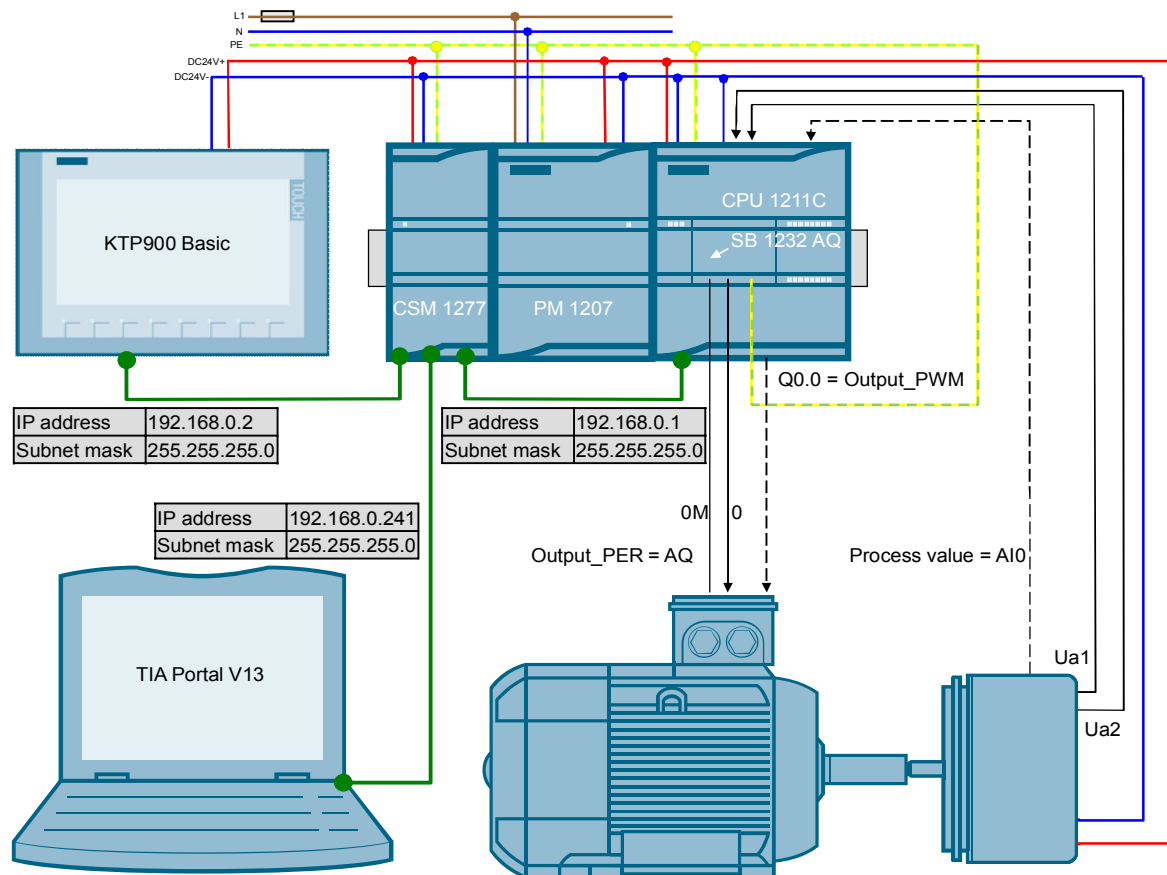
## 4 Installation and Commissioning

### 4.1 Hardware adaptation

#### Hardware installation

The figure below shows the hardware configuration of the application.

Figure 4-1



#### Installation of the hardware

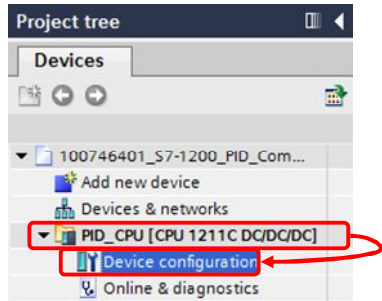
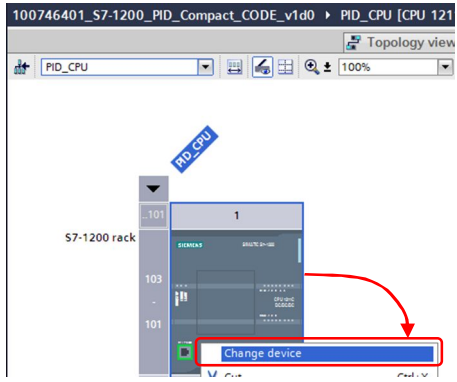
Table 4-6

No.	Action	Remarks
1	Adjust the periphery of S7-1200 to the actuator you are using.	See Chapter 4.1
2	Mount all required S7-1200 components on a top-hat rail.	See Chapter <b>Fehler!</b> <b>Verweisquelle konnte nicht gefunden werden.</b>
3	Wire and connect all required components as described.	S7-1200 manual ( <a href="#">\3\</a> ) <a href="#">Chapter A "Technical Specifications"</a>
4	Finally, activate the voltage supply for the SIMATIC PM 1207.	

## 4.2 Configuration guide

### Adjusting the device configuration

Table 4-7

No.	Action	Remarks
1.	Connect the S7-1200 controller with your programming device, assigning the Ethernet parameters shown in <a href="#">Figure 4-1</a> .	Assign IP address to S7-1200: S7-1200 manual (\3\) → <a href="#">Chapter 5.7.4</a>
2.	Open the project file (ap13) with STEP 7 V13.	Table 2-3
3.	Open the device configuration of the "PID_CPU" controller.	
4.	<p>Adjust the device configuration in the project to the real hardware configuration for your used actuator and actual value sensor (Chapter 4.1).</p> <p>If using a different CPU, select the configured CPU and activate "Change device..." via right-click.</p> <p>Further options for adding modules to the configuration are available in the S7-1200 Manual (\3\) → <a href="#">Chapter 5.3</a>.</p>	

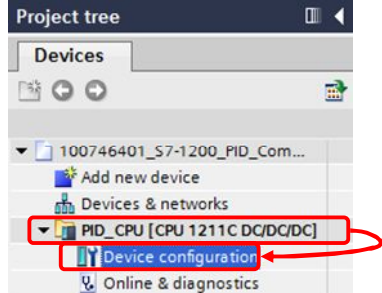
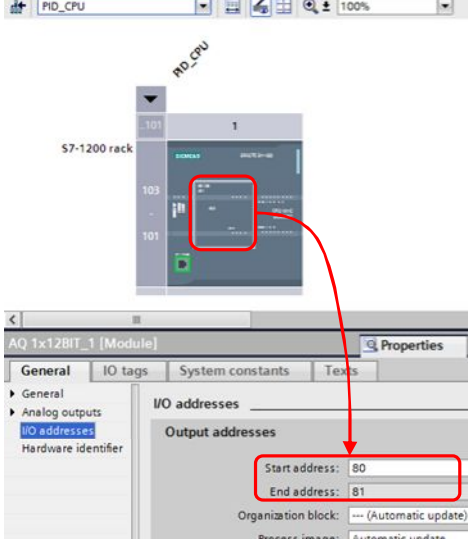
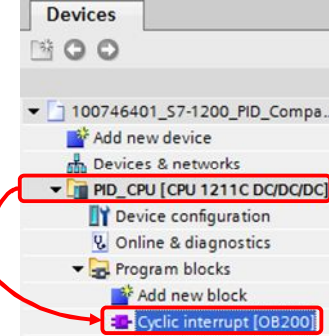


**Transferring the I/O addresses**

Depending on the changed configuration, the input or output addresses of the added hardware must be transferred to the program.

This is illustrated using a 1232 AQ 1x12 bit signal board as an example:

Table 4-8

No.	Action	Remarks
1.	Open the device configuration of the "PID_CPU" controller.	
2.	<p>In the device view of the CPU, select the 1232 AQ 1x12 bit signal board.</p> <p>Read the input address of the signal board in menu item "I/O addresses":</p> <ul style="list-style-type: none"> <li>Start address: 80</li> <li>End address: 81</li> </ul> <p>This means: The address through which the analog value of SB 1232 AQ 1x12 bit is output, is: <b>QW80</b></p>	
3.	Open the OB200 "Cyclic interrupt" in the control unit of the project.	

## 4.2 Configuration guide

No.	Action	Remarks
4.	<p>Since the analog manipulated variable is output to the actuator via the signal board, transfer the "tags".PID_Compact.Output_PER tag to the output word <b>QW80</b> in network 6.</p> <p>Here, in addition, you can adapt the digital output used in case you use the pulse-width modulated output signal "Output_PWM".</p>	
5.	<p>In network 2, transfer the address of the analog input to parameter "Input_PER_physical" of the "Switch" block, if you use an actual value sensor with analog signal evaluation.</p> <p>Here you can also transfer the adapted actual value as a floating-point number to parameter "Input_physical".</p>	

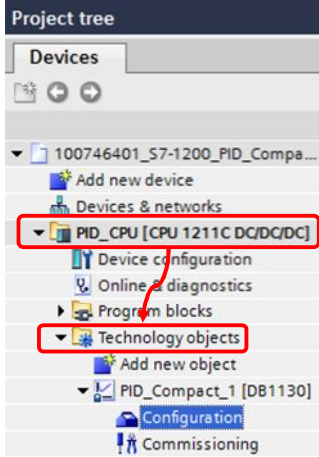
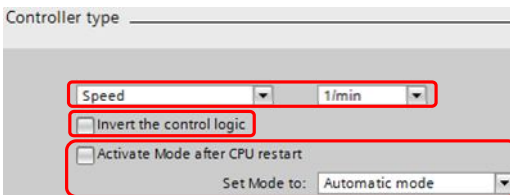
### Configuring the PID controller

The configuration of the “PID\_Compact” technology object defines the function principle of the compact controller.

The settings made determine the start values used by the PID controller when restarting after a cold or warm start (e.g. power failure).

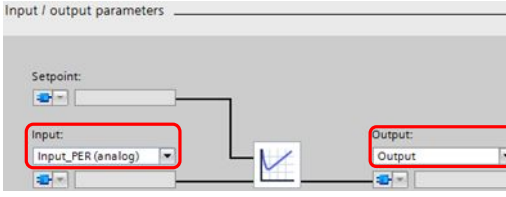
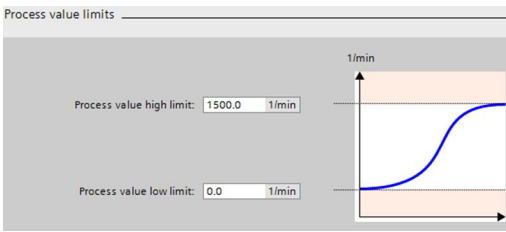
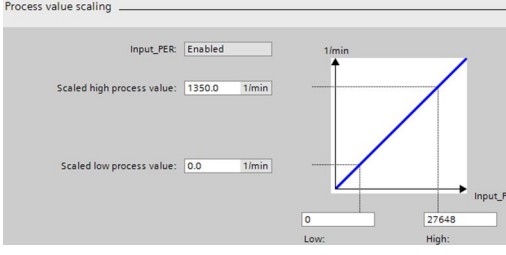
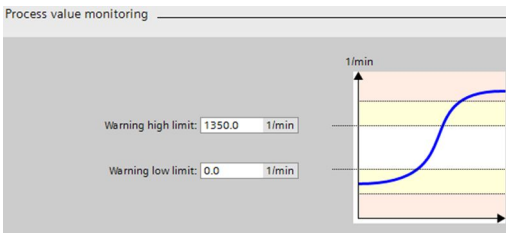
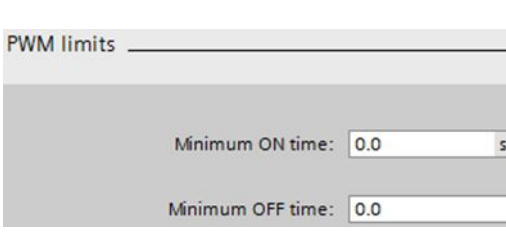
A more detailed description is available in the S7-1200 manual (\3\ → [Chapter 9.2.6](#) and in the STEP 7 Basic V13 manual (\6\ → [Chapter 11.1.3.2](#).

Table 4-9

No.	Action	Remarks
1.	Open the configuration editor by selecting CPU -> Technology objects -> PID_Compact_1 -> Configuration.	
2.	<p>Open the “<b>controller type</b>” submenu in the basic settings:</p> <p>Determine</p> <ul style="list-style-type: none"> <li>which physical unit to use for displaying setpoint and actual value</li> <li>whether the relation between the control deviation and the manipulated variable should be proportional or inversely proportional</li> <li>whether the controller, <b>after being fully loaded and with CPU restart</b>, should stay “inactive” or start up in the operating state which was selected under “Set mode to:” (otherwise the controller will always start in its most recent operating state, since “Mode” is retentive)</li> </ul> <p><b>Note:</b> The setting the controller type (here: speed) with the unit (here: 1/min) only serves for labeling the axis correctly in the commissioning wizard (see chapter 4.3).</p>	

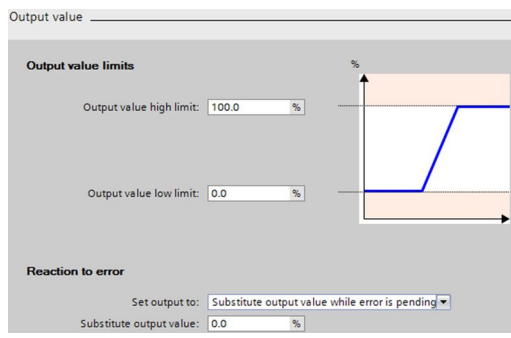
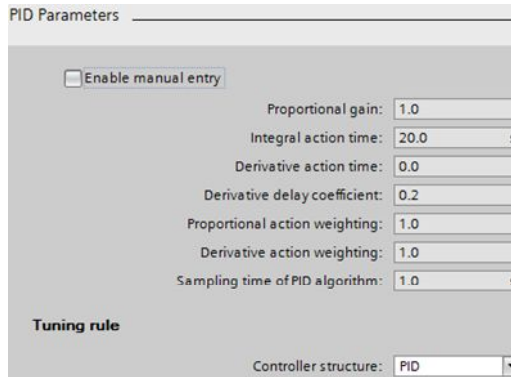
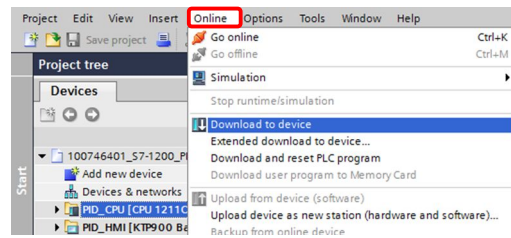
## 4 Installation and Commissioning

### 4.2 Configuration guide

No.	Action	Remarks
3.	<p>Adapt the basic settings of the “<b>input/output parameters</b>” submenu to the sensors/actors you are using:</p> <ul style="list-style-type: none"> <li>Actual value as an adapted floating-point number “Input” or analog value “Input_PER”</li> <li>Manipulated variable as floating-point number “Output”, as analog value “Output_PER” or as digital pulse-width modulated signal “Output_PWM”</li> </ul>	
4.	<p>In the “<b>Process value limits</b>” submenu, define the limits of the scaled process value in the process value settings.</p> <p><b>Note:</b> Please ensure the correct settings of upper and lower limits of the actual value, since the controller treats the violation of these limits as a fault and will respond accordingly!</p>	
5.	<p>In the “<b>Process value scaling</b>” submenu, define the value pairs for linear conversion into the scaled process value in the process value settings, if you use the analog process value “Input_PER”.</p>	
6.	<p>Open “<b>Process value monitoring</b>” in the Advanced Settings:</p> <p>Here you can specify warning limits where a respective warning bit will be activated when exceeded or fallen short of.</p>	
7.	<p>Open “<b>PWM limits</b>” in the Advanced Settings:</p> <p>For adjusting the actuator response time you can specify minimum ON/OFF times.</p> <p><b>Note:</b> The settings are also valid if a different manipulated variable signal (“Output” or “Output_PER”) is used!</p>	

## 4 Installation and Commissioning

### 4.2 Configuration guide

No.	Action	Remarks
8.	<p>Open the “<b>Output value</b>” submenu in the Advanced Settings:</p> <p><b>Output value limits</b> Define the percentage limits of the signal to be output to the actuator.</p> <p><b>Response to error</b> Determine whether in case of error</p> <ul style="list-style-type: none"> <li>the controller will be inactivated,</li> <li>the current manipulated variable will be retained for the time of error or</li> <li>a definable substitute output value shall be output as manipulated variable.</li> </ul>	
9.	<p>Open “<b>PID parameters</b>” in the Advanced Settings: Here you can specify the start values of the controller parameters manually. They will then be written to the instance data block of the “PID_Compact” and applied as actual values after a cold start (loading project into the controller).</p> <p><b>Rules for tuning</b> Depending on the controller structure selected, the start values for the setting rules of the pretuning or fine tuning will be set to</p> <ul style="list-style-type: none"> <li>“PID acc. to Chien, Hrones and Reswick” or “PID automatic” or</li> <li>“PI” acc. to Chien, Hrones and Reswick” or “Ziegler-Nichols PI”.</li> </ul>	
10.	<p>Save the project. Select the program folder of the S7-1200 and transfer the program to the controller via “Online/Download to device”.</p> <p>In “Load preview”, select the option “stop all” from “stop modules”.</p> <p>In “Loading process results”, select the option “start all” from “start modules” and finish the loading process.</p>	

#### Note

Any changes of the start values of a data block will only be applied as actual values with the next STOP/RUN transition (not with retentive data types).

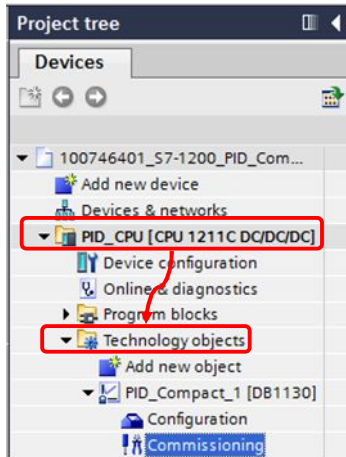
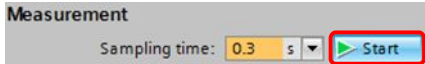
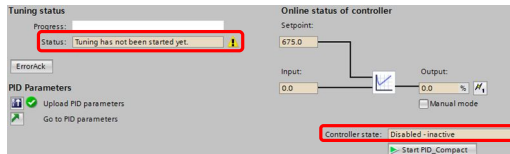

### 4.3 Commissioning the compact controller

In the commissioning editor, configure the compact controller for the automatic setting during startup and for the automatic setting during operation.

The settings made determine the start values used by the PID controller when restarting after a cold or warm start (e.g. power failure).

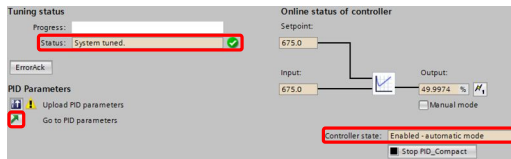

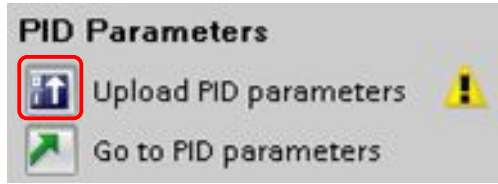
A more detailed description is available in the S7-1200 manual (V3) → [Chapter 9.2.7](#) and in the STEP 7 Basic V13 manual (V6) → [Chapter 11.1.3.2](#).

Table 4-10

No.	Action	Remarks
1.	Open the commissioning editor by selecting CPU -> Technology objects -> PID_Compact_1 -> Commissioning.	
2.	Start the measurement.	
3.	The tuning status indicates that tuning has not yet been started, and after the first startup of the CPU the controller is in "Disabled - inactive" mode (see <a href="#">Table 4-9</a> Step 2).	
4.	You will get the best results if you perform a pretuning process followed by fine tuning after the first startup from inactive state. Enter a setpoint ideally in the central area of the actual value range (e.g. via a monitoring table; in the project the start value of the setpoint has already been predefined). Start the pretuning process.	

## 4 Installation and Commissioning

### 4.3 Commissioning the compact controller

No.	Action	Remarks
5.	<p>After successful pretuning, the controller goes to automatic mode.</p> <p>The determined values can be viewed via "Go to PID parameters".</p> <p>Selecting "Upload PID parameters" will write the determined values as start values into the instance data block of "PID_Compact".</p>	
6.	<p>Enter a setpoint ideally in the central area of the actual value range (e.g. via a monitoring table; in the project the start value of the setpoint has already been predefined).</p> <p>Then start the fine tuning process.</p>	
7.	<p>After successfully terminating the fine tuning you can in turn upload the PID parameters as start values to the instance data block of "PID_Compact".</p>	

#### Note


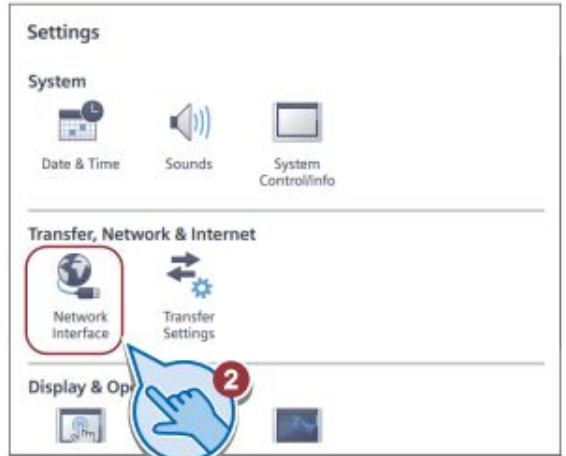
The PID parameters are stored retentively in the instance data block of the "PID\_Compact" compact controller. During a warm start (restored power) the last processed values remain. The start values are only loaded during cold start (transferring the project in STOP mode or memory reset via MRES).

## 4.4 HMI project part

### 4.4.1 Configuring the HMI

If the KTP900 Basic is used as an operator panel, the project-specific IP address (see [Figure 4-1](#)) must be set.

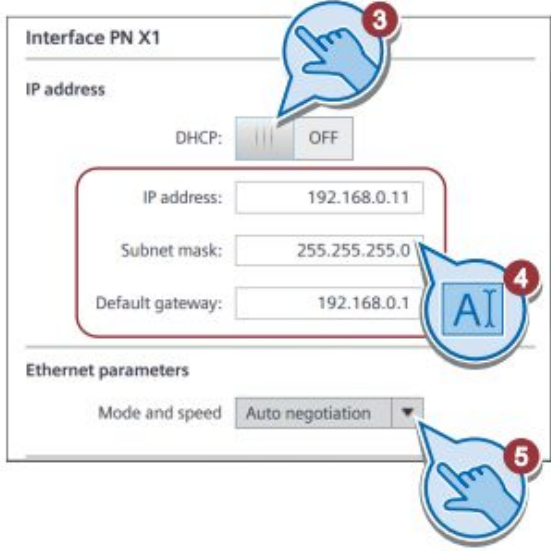
Table 4-11

No.	Action	Remarks
1.	<ul style="list-style-type: none"> <li>Connect the KTP900 Basic to the supply voltage.</li> <li>On the "Start Center" screen, go to settings by selecting "Settings".</li> </ul>	
2.	<ul style="list-style-type: none"> <li>Then open the network interface settings by selecting the "Network interface" button.</li> </ul>	



## 4 Installation and Commissioning

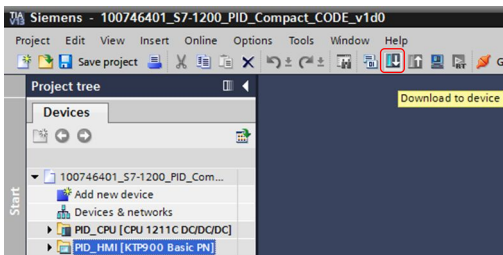
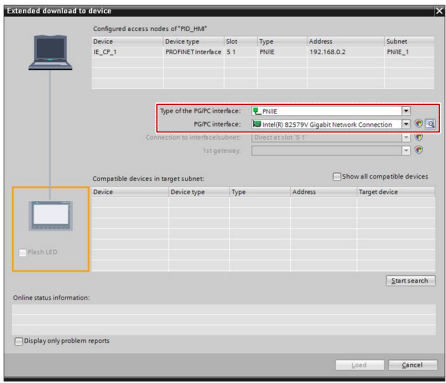
### 4.4 HMI project part

No.	Action	Remarks
3.	<ul style="list-style-type: none"> <li>Disable the “DHCP” option.</li> <li>Enter the network setting from <a href="#">Figure 4-1</a> for the HMI: <ul style="list-style-type: none"> <li>IP address: 192.168.0.2</li> <li>Subnet mask = 255.255.255.0</li> <li>“Default gateway” is irrelevant.</li> </ul> </li> <li>For the Ethernet parameters, select “Auto negotiation” for automatic transfer rate determination.</li> </ul>	

### Loading the HMI project part into the KTP900 Basic

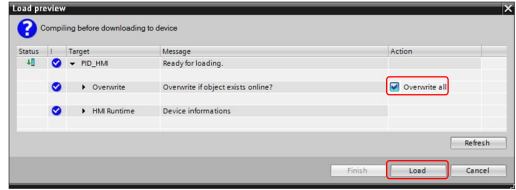

For the transfer, connect your PG/PC to the HMI either directly or using the CSM1277 switch.

Table 4-12

No.	Action	Remarks
1.	<ul style="list-style-type: none"> <li>Select the “PID_HMI [KTP900 Basic PN]” operator panel folder.</li> <li>Select the “Download to device” button to download the HMI project part to the KTP900 Basic.</li> </ul>	
2.	<ul style="list-style-type: none"> <li>From the “Extended download” screen, select “PN/IE” as the type of the PG/PC interface.</li> <li>Define your network card as the PG/PC interface.</li> </ul>	

## 4 Installation and Commissioning

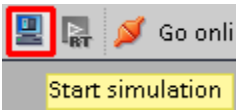
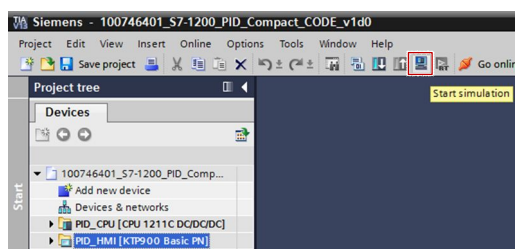
### 4.4 HMI project part

No.	Action	Remarks
3.	<ul style="list-style-type: none"> <li>If required, enable the “Overwrite all” option.</li> <li>Press the “Load” button.</li> </ul>	
4.	<ul style="list-style-type: none"> <li>Trigger the transfer in the Start Center of the KTP900 Basic using the “Transfer” button, depending on the operator panel setting.</li> </ul>	

### Starting PC runtime

To use the PG/PC as an operator panel, start the PC runtime as follows:

Table 4-13

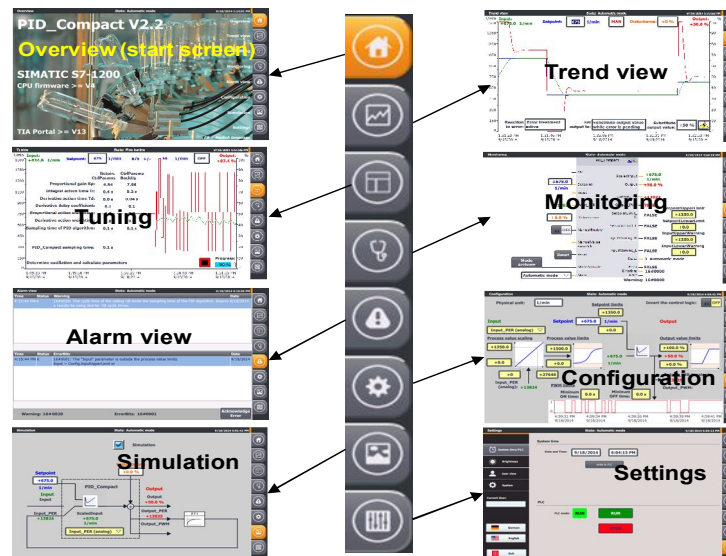
No.	Action	Remarks
1.	<ul style="list-style-type: none"> <li>Select the “PID_HMI [KTP900 Basic PN]” operator panel folder.</li> <li>Click the “Start simulation” button.</li> </ul> 	

## 5 Operation of the Application

### 5.1 Overview

#### Overview and description of the user interface

Figure 5-1



The user interface includes 8 menus:

- Start screen (overview)
- Trend view
- Tuning
- Monitoring
- Alarm view
- Configuration
- Simulation
- System functions (settings)

#### 5.1.1 Overview (start screen)

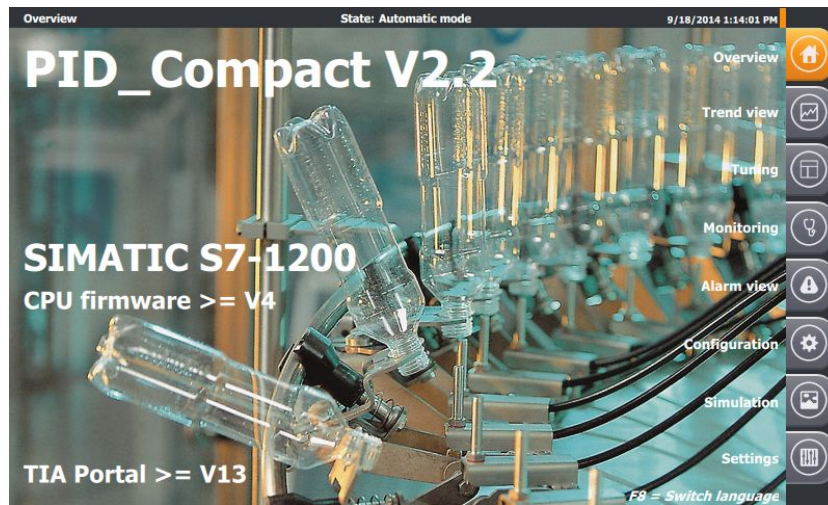
The overview screen gives information about the topic.









The technology object is presented: the "PID\_Compact" version 2.2 compact controller. It is available in the SIMATIC S7-1200 controller with CPU firmware V4.0 or higher. It is configured in STEP 7 (TIA Portal) with version V13 or higher.


In addition, the operation of the menu bar on the right is explained.

The menu bar is displayed on every screen.

Figure 5-2



-  takes you to the Overview screen (this screen).
  -  takes you to the Trend view screen ([Figure 5-3](#)).
  -  takes you to the Tuning screen ([Figure 5-4](#)).
  -  takes you to the Monitoring screen ([Figure 5-5](#)).
  -  takes you to the Alarm view screen ([Figure 5-6](#)).
  -  takes you to the Configuration screen ([Figure 5-8](#)).
  -  takes you to the Simulation screen ([Figure 5-10](#)).
  -  takes you to the System functions ([Figure 5-11](#)).
  - Press F8 to toggle between German and English language.
- You can see which menu is currently selected from the orange background color of

the menu icon: for example  (for the overview screen) or the menu name in the header (on the left): **Overview**.

In the middle of the header you can see which operating mode the controller is currently in: **State: Automatic mode** (for automatic mode)

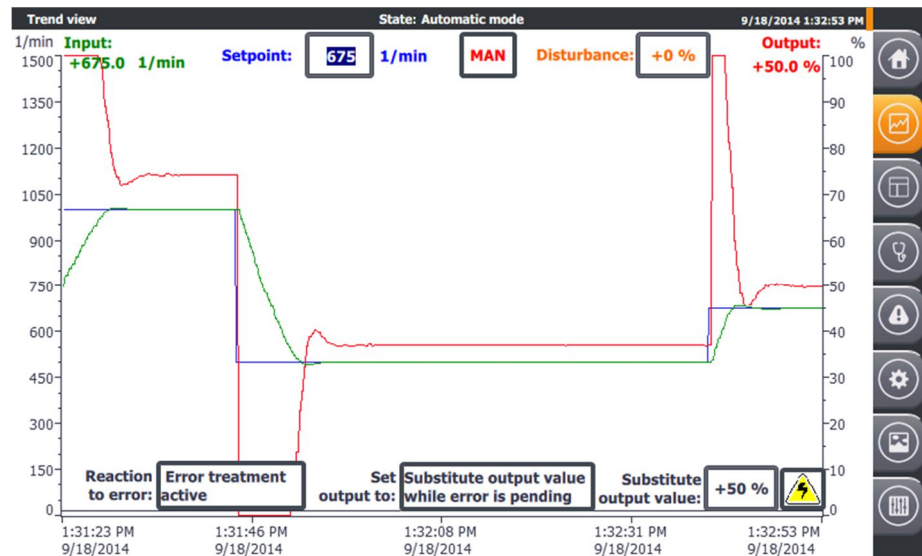
The current date and time is displayed in the header on the right side:

**9/19/2014 1:15:40 PM**

The header is also displayed in every screen.

#### 5.1.2 Trend view

Figure 5-3



The “Trend view” screen shows the time sequence over 90 seconds

- of the setpoint value **Setpoint** (left scale)
- of the actual value **Input** (left scale)
- of the manipulated variable **Output** (right scale)

#### Manual mode



Selecting **MAN** will switch to manual mode.

In manual mode you can enter the manipulated variable directly via the manual value (value range 0 to 100%).

#### Note

Here, manual mode is activated through the value specification at the “Mode” parameter, in connection with the activation through “ModeActivate” (not through “ManualEnable”).

#### Automatic mode



Selecting **AUTO** will switch to automatic mode.

**Setpoint: 675 1/min**

With this you can specify the setpoint value in automatic mode.

**Disturbance: +100 %**




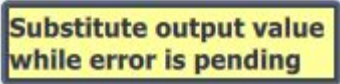

With this you can specify the disturbance that is added directly to the manipulated variable.

This way you can observe the disturbance compensation in the Trend view in automatic mode.

#### Response to error

You can predetermine the response of the controller in case the actual value is exceeded (e.g. due to failure of the sensor).

Determine whether in case of error


- the controller will be inactivated 
- or, with active error handling, 
  - the current manipulated variable will be retained for the time of error  or
  - a substitute output value , definable via , shall be output as manipulated variable.


In the event of an error, with active error handling, the manipulated variable (output) will be set to the current value or the substitute output value for the time of error.

This response will occur in the modes: Pre- or fine tuning, as well as in automatic mode.

The controller will additionally change to inactive in manual mode if this response is selected for the event of an error.

In controlled system simulation operation, the selected response to error can be

simulated via  in automatic mode (switching on and off). If the fault condition

is switched on, the icon will have a red background: 

#### NOTICE

**These selection boxes have a yellow background since these are non-retentive data in the instance data block of the "PID\_Compact". You can change these via the operator panel or their simulation in order to test the function.**

**In order to save these presettings also during a power failure, these values must be written to the instance data block of "PID\_Compact" as start values.**

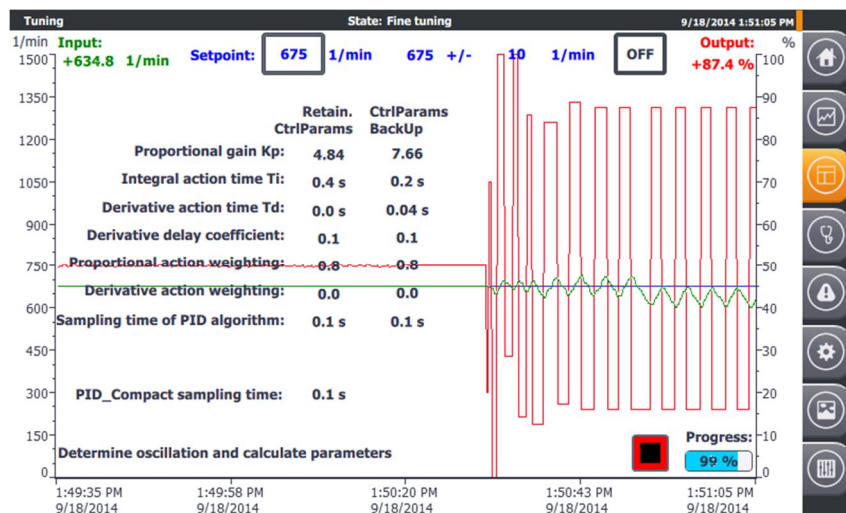
**The configuration wizard offers this function (Table 4-9, Step 8) with subsequent transmission of the instance data block.**



#### 5.1.3 Tuning

In the “Tuning” menu you can specify the control parameter setting automatically or manually.

Figure 5-4



The Tuning screen provides the option

- for pretuning or
- for fine tuning

from the inactive controller state, manual or automatic mode.

#### Note

For this, do not select manual mode via the “ManualEnable” input!

For pretuning you can choose between the following tuning methods:

- Chien, Hrones, Reswick PID
- Chien, Hrones, Reswick PI

For fine tuning you can choose between the following tuning methods:


- PID automatic
- PID fast
- PID slow
- Ziegler-Nichols PID
- Ziegler-Nichols PI
- Ziegler-Nichols P

With **Cancel TuningLevel:** **10** 1/min you can determine the maximum setpoint change during tuning. If this value is exceeded, the tuning process will be interrupted and, depending on the response to error selected, the controller will switch to inactive or will return to the operating state that the tuning process was started from.

During tuning, the setpoint is frozen. The specified limits (frozen setpoint value with maximum setpoint change) are displayed: **675 +/- 10 1/min**

**Note** The “CancelTuningLevel” parameter ensures tuning even with signal noise at the setpoint (e.g. if a potentiometer is used).




Pressing  will start the selected tuning method.

**Note** Enter a setpoint value preferably in the middle of the actual value range to prevent the tuning process from being canceled if the limit is reached.

During tuning, the tuning status and the percentage of progress are displayed.

**Progress:**

**60 %**

Pressing  will cancel the tuning process and you will return to the operating state which the tuning process was started from.

After successful tuning, the determined controller parameters are displayed in the

**Retain.**

**CtrlParams** column and the controller parameters prior to tuning are moved to the

**CtrlParams**

**BackUp** column.

The secured “BackUp” parameter set can be loaded back into the controller with

**LoadBackUp**

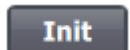
The current controller parameters (“Retain.CtrlParams”) can also be edited

manually (indicated by the frame around the parameters **1.0**).

**Note** The current controller parameter set (“Retain.CtrlParams”) is retentive and will also be retained after power failure.  
In order to use these parameters also for starting after a cold start, they must be written to the instance DP of “PID\_Compact” as start values.  
The commissioning wizard offers this function (Table 4-10, Step 7).



The default values of the PLC data type “PID\_CompactRetain” are loaded through



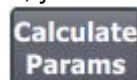
	Retain. CtrlParams
Proportional gain Kp:	1.0
Integral action time Ti:	20.0 s
Derivative action time Td:	0.0 s
Derivative delay coefficient:	0.2
Proportional action weighting:	1.0
Derivative action weighting:	1.0

The sampling time of the controller **PID\_Compact sampling time: 0.1 s** corresponds to the time clock of the cyclic interrupt organization block in which the “PID\_Compact” is called.

The sampling time of the PID algorithm

**Sampling time of PID algorithm: 0.1 s** corresponds to a multiple of the controller sampling time and does not depend on the PWM limitation.

After successful tuning, depending on the tuning type, you can calculate the



controller parameters for other tuning methods with **Calculate Params** without having to repeat the tuning process.



Use **OFF** if you want to inactivate the controller. This operating state is particularly suitable for pretuning. During pretuning the controller parameters are determined from the response to a setpoint step.

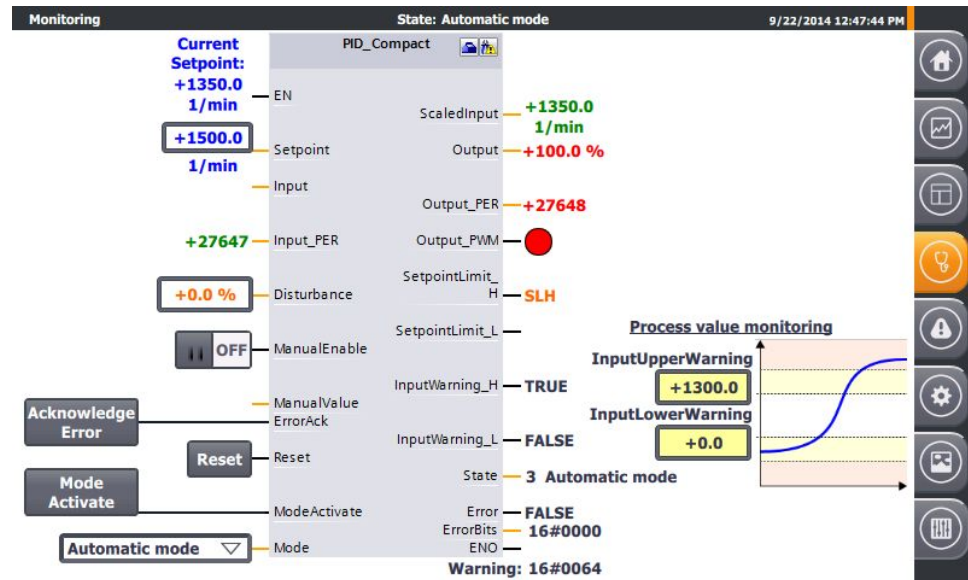
For this, the actual value must not be too close to the setpoint:

- $|\text{Setpoint} - \text{Input}| > 0.3 * |\text{Config.InputUpperLimit} - \text{Config.InputLowerLimit}|$  and
- $|\text{Setpoint} - \text{Input}| > 0.5 * |\text{Setpoint}|$

### 5.1.4 Monitoring

The Monitoring screen shows the online status of the “PID\_Compact” compact controller.

Figure 5-5



You can:

- view all inputs and outputs
- edit the following parameters:
  - Setpoint in automatic mode (“Setpoint”)
  - Disturbance value compensation in automatic mode (“Disturbance”)
  - Enable/disable manual mode (“ManualEnable”)
  - Manually specify the manipulated variable (“ManualValue”)
  - Acknowledgement (reset) of the “ErrorBits” and “Warning” (“Acknowledge Error”) messages
  - Reset of the compact controller (“Reset”)
  - Change of operating mode by selecting from the “Mode” parameter and activating via “ModeActivate”.
- Test the configuration of the actual value monitoring.
  - Editing the upper (“InputUpperWarning”) and lower (“InputLowerWarning”) warning limits
  - Direct monitoring at the “InputWarning\_H” or “InputWarning\_L” outputs.

#### Note

In this application, the Reset button initiates a restart of the “PID\_Compact” and “PROC\_C” blocks, with the controller changing to the “inactive” operating mode. The “ErrorBits” and “Warning” messages will be reset. Then the controller will start in the operating mode selected at the “Mode” parameter.

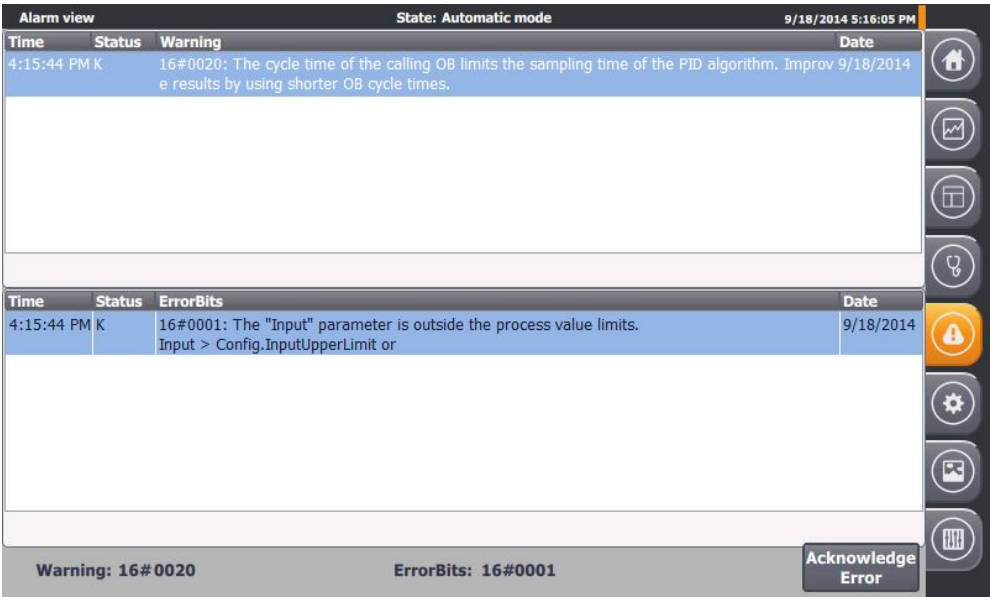
5.1 Overview

**Note** In manual mode (activated via “ManualEnable” = “ON”), the operating mode cannot be changed via “Mode” and “ModeActivate”.

5.1.5 Alarm view

The “Alarm view” menu shows the current alarms at the “ErrorBits” output and at the static variable “Warning” of the “PID\_Compact” as a hexadecimal error code and as a text with time stamp and status.

Figure 5-6



The screenshot shows the 'Alarm view' interface. At the top, it says 'State: Automatic mode' and '9/18/2014 5:16:05 PM'. There are two main sections: 'Warning' and 'ErrorBits'. The 'Warning' section shows a message: '16#0020: The cycle time of the calling OB limits the sampling time of the PID algorithm. Improve results by using shorter OB cycle times.' with a time stamp of '4:15:44 PM K' and date '9/18/2014'. The 'ErrorBits' section shows a message: '16#0001: The "Input" parameter is outside the process value limits. Input > Config.InputUpperLimit or' with a time stamp of '4:15:44 PM K' and date '9/18/2014'. At the bottom, there is a status bar showing 'Warning: 16#0020' and 'ErrorBits: 16#0001', and an 'Acknowledge Error' button.

Time	Status	Warning	Date
4:15:44 PM K		16#0020: The cycle time of the calling OB limits the sampling time of the PID algorithm. Improve results by using shorter OB cycle times.	9/18/2014

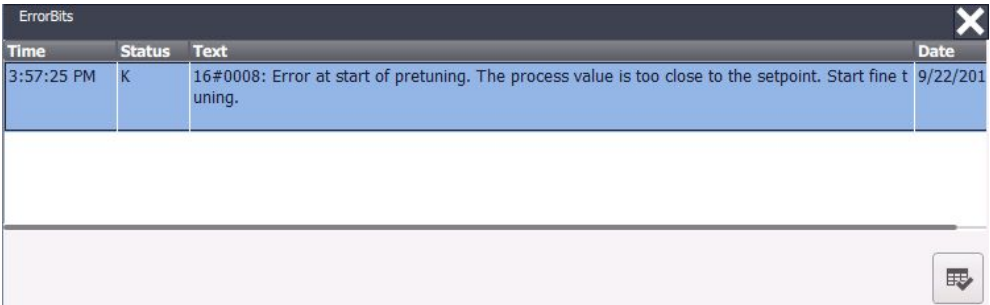
  

Time	Status	ErrorBits	Date
4:15:44 PM K		16#0001: The "Input" parameter is outside the process value limits. Input > Config.InputUpperLimit or	9/18/2014

Warning: 16#0020      ErrorBits: 16#0001      Acknowledge Error


The “ErrorBits” error messages are also displayed globally once they come up.

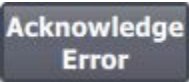
Figure 5-7



The screenshot shows the 'ErrorBits' dialog box. It has a title bar 'ErrorBits' and a close button 'X'. The table inside shows a message: '16#0008: Error at start of pretuning. The process value is too close to the setpoint. Start fine tuning.' with a time stamp of '3:57:25 PM K' and date '9/22/2014'. There is a scroll bar at the bottom and a button with a downward arrow icon.

Time	Status	Text	Date
3:57:25 PM K		16#0008: Error at start of pretuning. The process value is too close to the setpoint. Start fine tuning.	9/22/2014

You can acknowledge errors no longer pending by selecting . This will cause any messages no longer pending at “ErrorBits” and “Warning” to be deleted via the “ErrorAck” input.

Perform this function via  within the message display and the other screens.  
The button is only visible if messages are pending (“ErrorBits” or “Warning”).

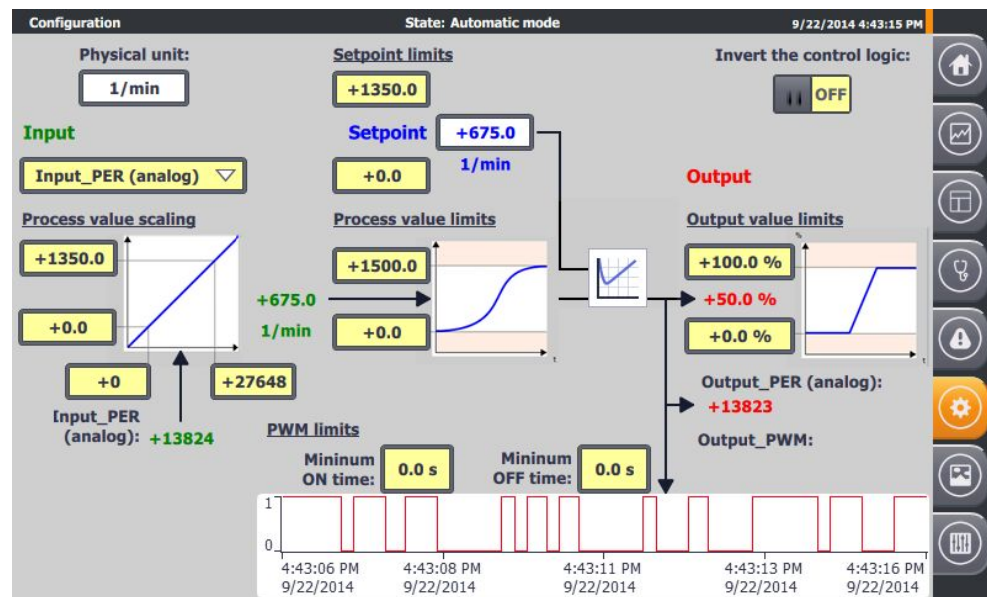
#### Note

You will find the full description of all error messages in the online help of STEP 7 Basic V13. Select function block "PID\_Compact" in the program call (see Figure 3-5) and press F1.

#### 5.1.6 Configuration

The Configuration screen is based on the basic settings of the configuration wizard (Table 4-9).

Figure 5-8



Here, you can change the following specifications during runtime:

#### Default settings

- Control mode
  - Specification of the physical unit displayed (limited to 5 characters; not identical with the preselection in the configuration wizard)

**Physical unit:**  
1/min

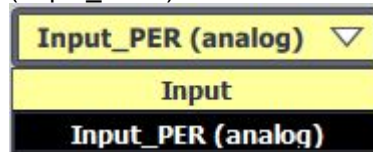
- Inverting the control direction (see Table 4-9, step 2)

**Invert the control logic:**  
OFF

- Input/output parameters (see Table 4-9, step 3)

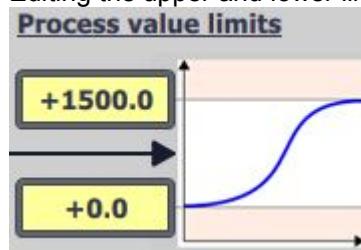
### 5.1 Overview

- Selection of process value signal: floating-point number ("Input") or analog ("Input\_PER")

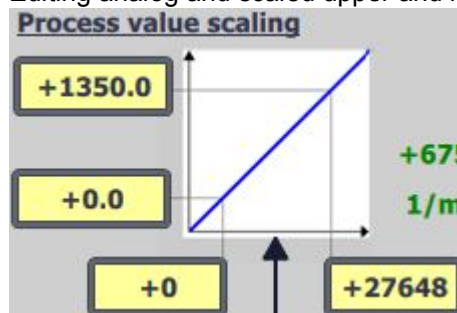


#### Process value settings

- Process value limits (see Table 4-9, step 4)
  - Editing the upper and lower limit of the process value



- Process value scaling (see Table 4-9, step 5)
  - Editing analog and scaled upper and lower process values



#### Note

Process value scaling is used for linear conversion of the "Input\_PER" analog value into the "ScaledInput" scaled process value. In simulation mode, this conversion is also required for selecting the process value acquisition via the floating-point value "Input" (see [Figure 3-3](#)).

#### Advanced settings

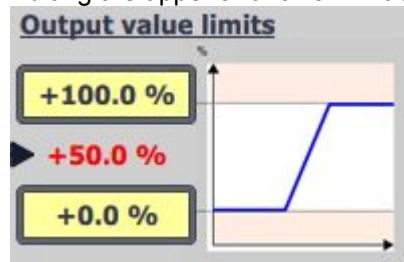
- PWM limits (see Table 4-9, step 7)
  - Editing minimum switch-on and switch-off time for adjusting to potential actuator response time



- Output value limits (see Table 4-9, step 8)

### 5.1 Overview

- Editing the upper and lower limit of the output value



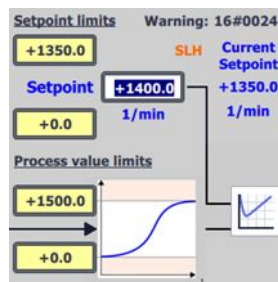
### Setpoint limits

The "PID\_Compact" compact controller automatically limits the setpoint value to the "Process value limits".

You can also limit the setpoint to a smaller range via the "Setpoint limits".

"PID\_Compact" will automatically take the closer limit.

Figure 5-9



In case of a limit violation there will be an internal limitation.

The actual setpoint value "CurrentSetpoint" is displayed and the output parameter

"SetpointLimit\_H" **SLH** or "SetpointLimit\_L" **SLL** indicates the limit violation.

A corresponding warning message will be issued (16#0004).

This screen is for making the user familiar with the settings of the compact controller and their characteristics (especially for the simulation mode).

#### NOTICE

These input/output fields have a yellow background since these are non-retentive data in the instance data block of the "PID\_Compact". You can change these via the operator panel or their simulation in order to test the function.

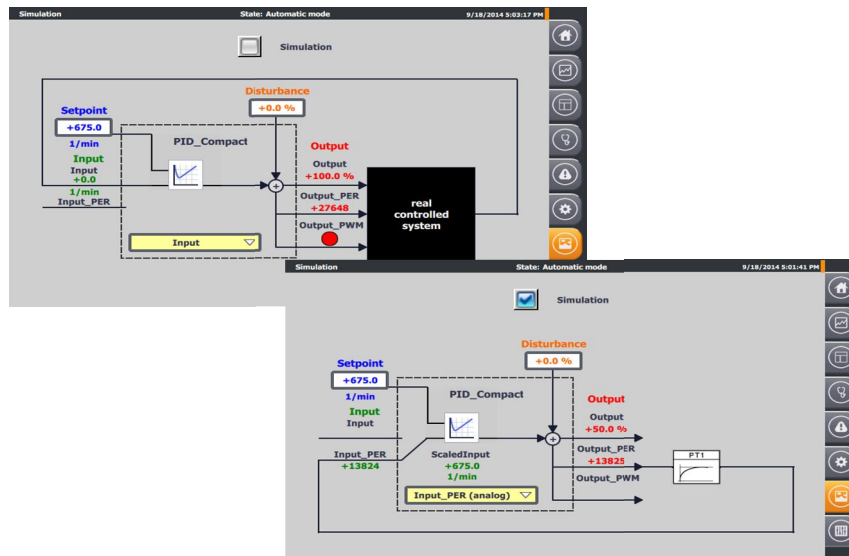
In order to save these presets also during a power failure, these values must be written to the instance data block of "PID\_Compact" as start values.

The configuration wizard offers this function (Table 4-9) with subsequent transmission of the instance data block.

#### 5.1.7 Simulation

In the Simulation screen you can switch between a real and a simulated control system.


Figure 5-10




The block diagram of the PID controller is displayed with:

- the “Setpoint” specification
- the actual value display with selection of actual value signal
  - “Input” as floating-point number or
  - “Input\_PER” as analog value with internal conversion (“ScaledInput”)
- the “Disturbance” specification
- the manipulated variable output
  - as a percentage floating-point number “Output”
  - as an analog value “Output\_PER”
  - as a pulse-width modulated digital signal “Output\_PWM”

If the simulation is not switched on, the controller receives the signals from the controller I/O (Table 4-8, step 5).

If the simulation is switched on (  Simulation ), the screen shows the block diagram structure for calculating the input signals for the controller:

The output of the controlled system simulation block PT1  supplies the actual value.

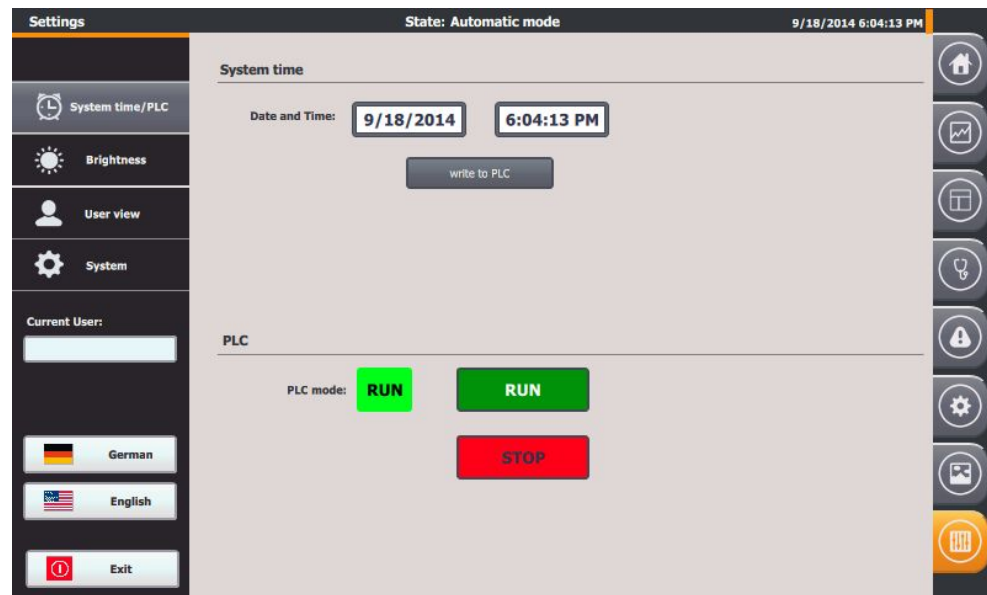


#### 5.1.8 Settings

The settings menu includes the screens

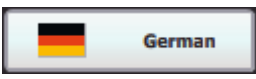
- System time/CPU
- Brightness
- User view
- System

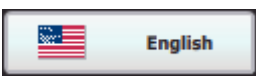
Figure 5-11



The user currently logged-on is displayed:

Current User:



Pressing  will select "German" as the display language.

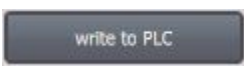
Pressing  will select "English" as the display language.


Pressing  will exit the HMI runtime.

#### Time setting/CPU

The application includes time synchronization between CPU and HMI.

Use  to edit the date and  to edit the time.


Use  to apply these settings and through this set the CPU system time.


 shows the current CPU operating state.



## 5 Operation of the Application

### 5.1 Overview

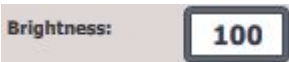
Use  to set the CPU to "RUN" mode.

Use  to set the CPU to "STOP" mode.

In the CPU operating state  the header and the side bar will alternately blink orange:



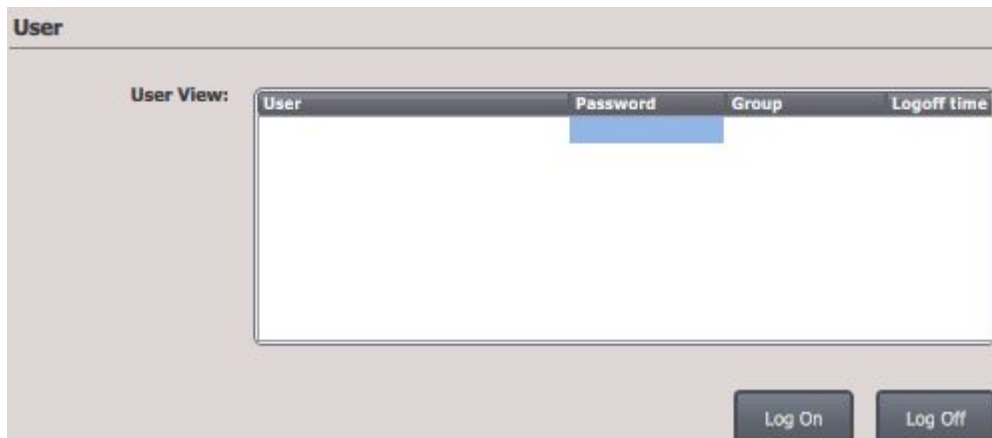
### Brightness

Use  to select the brightness in percent (setting range: 30 to 100%) of the operator panel (only possible if a real HMI is used).

### User view


The details of a user that is logging on are displayed here (user, password, group and logoff time). No authorization was connected with the application. Therefore users are not required to log on.

Figure 5-12



Log on with  or log off with .

### System

 will temporarily make the touch panel non-responsive so that you can clean it (only possible if a real HMI is used).

## 6 Related Literature

Table 6-1

	Subject	Title
\1\	Siemens Industry Online Support	<a href="http://support.automation.siemens.com">http://support.automation.siemens.com</a>
\2\	Download page of the entry	<a href="http://support.automation.siemens.com/WW/view/en/100746401">http://support.automation.siemens.com/WW/view/en/100746401</a>
\3\	SIMATIC S7-1200 Automation System System Manual	<a href="http://support.automation.siemens.com/WW/view/en/91696622">http://support.automation.siemens.com/WW/view/en/91696622</a>
\4\	WinCC Basic V13.0 System Manual	<a href="http://support.automation.siemens.com/WW/view/en/91379840">http://support.automation.siemens.com/WW/view/en/91379840</a>
\5\	Operating Instructions SIMATIC HMI Devices Basic Panels 2nd generation	<a href="http://support.automation.siemens.com/WW/view/en/90114350">http://support.automation.siemens.com/WW/view/en/90114350</a>
\6\	STEP 7 Basic V13.0 System Manual	<a href="http://support.automation.siemens.com/WW/view/en/89336297">http://support.automation.siemens.com/WW/view/en/89336297</a>
\7\	Controlling simulated controlled systems in S7-1500 with PID_Compact V2	<a href="http://support.automation.siemens.com/WW/view/en/79047707">http://support.automation.siemens.com/WW/view/en/79047707</a>
\8\	Updates for STEP 7 V13 and WinCC V13	<a href="http://support.automation.siemens.com/WW/view/en/90466591">http://support.automation.siemens.com/WW/view/en/90466591</a>
\9\	SIMATIC S7-1200 Easy Book	<a href="http://support.automation.siemens.com/WW/view/en/39710145">http://support.automation.siemens.com/WW/view/en/39710145</a>
\10\	3-Point Stepper Control with SIMATIC S7-1200 (Set 2)	<a href="http://support.automation.siemens.com/WW/view/en/62154322">http://support.automation.siemens.com/WW/view/en/62154322</a>
\11\	3-Point Stepper Control with SIMATIC S7-1500	<a href="http://support.automation.siemens.com/WW/view/en/68011827">http://support.automation.siemens.com/WW/view/en/68011827</a>
\12\	How are the automatic PID controller settings made for asymmetrical temperature control paths in STEP 7 V11 and higher?	<a href="http://support.automation.siemens.com/WW/view/en/66995021">http://support.automation.siemens.com/WW/view/en/66995021</a>

## 7 History

Table 7-1

Version	Date	Modifications
V1.0	11/2014	First version