SIEMENS
Ingenuity for life

PID control with PID_Compact

SIMATIC S7-1200 / S7-1500 + TIA Portal V15.1

https://support.industry.siemens.com/cs/ww/en/view/100746401

Siemens Industry Online Support

Legal information

Use of application examples

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

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

Disclaimer of liability

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

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

Other information

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

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

Security information

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

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

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

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

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

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

Table of contents

1 Task

1.1 Overview

Introduction

For the targeted influencing of certain quantities in a technical system, these quantities must be controlled. Controllers are also used in a wide variety of applications in automation technology, for instance speed control.

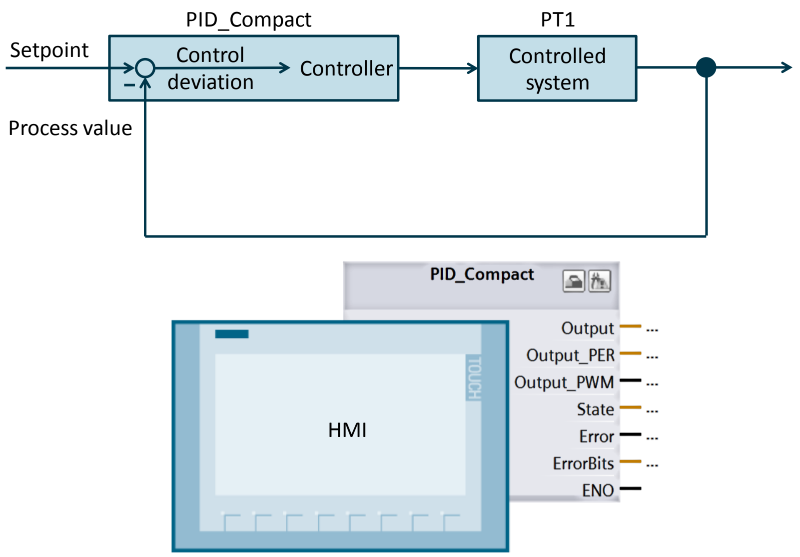
For the SIMATIC S7-1200/S7-1500, the technology object "PID_Compact" is provided for proportional actuators.

Description of the automation task

The automation task consists of setting up a control loop to influence physical variables in a technical process.
The control loop should consist of the following elements:

- "PID_Compact" as controller
- Simulated technical processes as a controlled system

Figure 1-1



The following points are described in the application example

- Configuration and parameter assignment of the software controller ("PID_Compact")
- Tuning options for "PID_Compact"
- Operation and monitoring of the control process via HMI

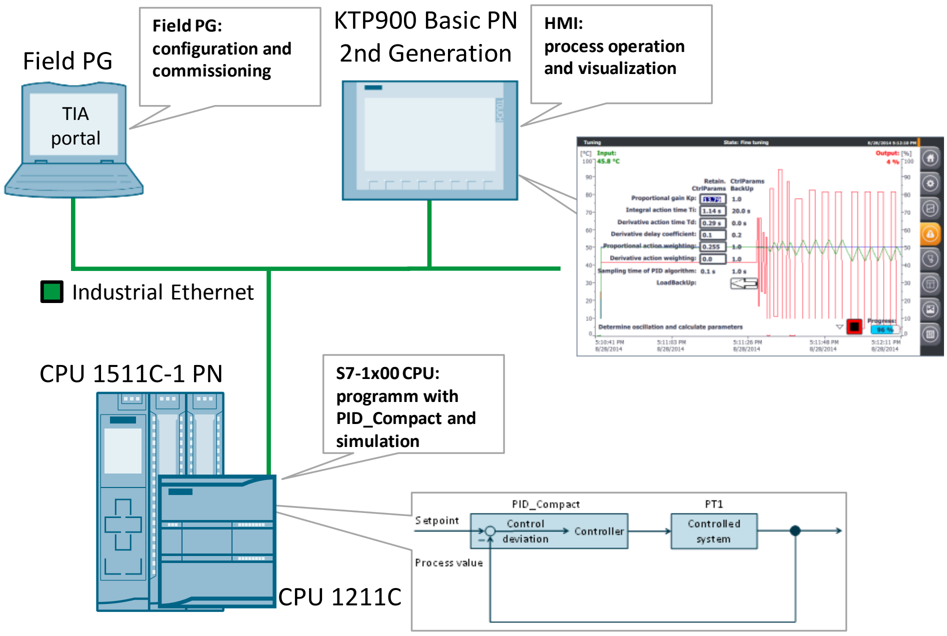
2 Solution

2.1 Overview

Diagram

The following diagram shows the most important components of the solution:

Figure 2-1



The "PID_Compact" technology object reads the measured process value and compares it with the setpoint (in this example, the setpoint is set via HMI). From the resulting control deviation, the controller calculates an output value in order to adjust the setpoint deviation or the disturbance variable if necessary. The output value of the PID controller consists of three components:

- P component
  The P component of the output value is proportional to the control deviation.
- I component
  The I component of the output value is the integral component. This increases as long as there is a control deviation.
- D component
  The D component is the differential component and increases as the rate of change of the control deviation increases.

The "PID_Compact" technology object has the "tuning" commissioning functionality with which the P, I and D parameters can be calculated automatically depending on the controlled system. However, you can also specify the control parameters manually.

The automatic tuning is divided into tuning types:

1. Pretuning and
2. Fine tuning

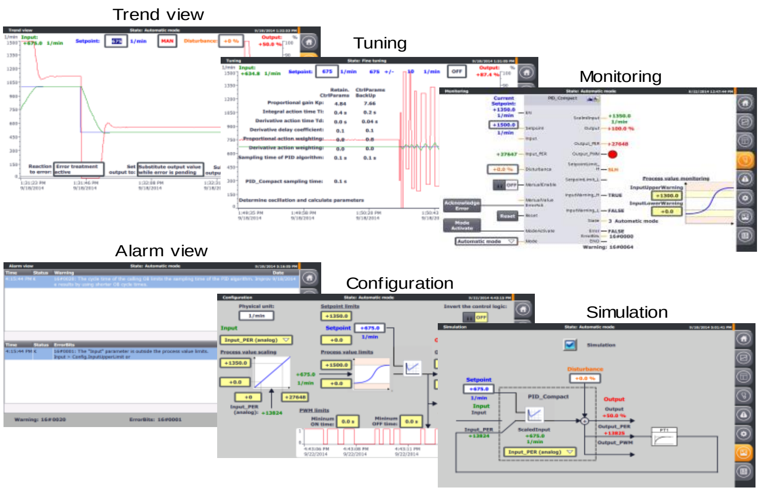
Both types of tuning are described below.

2.2 Description of the core functionality

The core functionality of the application example lies in the operation of the technology object "PID_Compact" via the HMI.

Overview and description of the interface

Figure 2-2



The operation of the application example consists of the following 6 screens:

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

The operation of the interfaces is described in more detail in the chapter Operation of the application example.

Advantages of this solution

The application example allows you to use all configuration options and commissioning features via an operator panel or HMI simulation.

This application example offers you the following advantages:

- Switchover between automatic and manual mode
- Trend view of setpoint, process and manipulated value via HMI
- Switchover between real controlled system and simulation
- Disturbance compensation in simulated operation
- Specification of the behavior in the event of an error and its simulation
- Manual controller parameter setting and automatic tuning
- Online monitoring of the "PID_Compact" controller module
- Configuration change at runtime

Delimitation

This application example provides an overview of the "PID_Compact" technology object for commissioning with the SIMATIC S7-1200/S7-1500.
You can use the application example to conveniently operate your control system via an operator panel and adapt it to your automation task.

The application example was tested by simulating the controlled system.
For real operation, you must adapt the application example to the actuator and process value sensor you are using:

- Analog control or control via a digital output using the pulse width modulated signal?
- Required voltage and power for control?
- What are the signal characteristics of the process value sensor used?

Note The application example is not a replacement for the configuration mask of the PID_Compact Assistant, as it is used to define the start values in the instance data block, which are decisive for restarting after a power failure.

In addition to the "PID_Compact" control block, STEP 7 (TIA Portal) also provides the following compact controllers with automatic tuning for the SIMATIC S7-1200/S7-1500:

- Modulating controller "PID_3Step" for valves or actuators with integrating behavior (\10\)
- Temperature controller "PID_Temp" for pure heating or heating/cooling applications (\4\, \5\)

Note You can find more information on the technology objects

- in the STEP 7 Professional manual (\6\) → chapter on "PID Control" or
- in the function manual on PID control (\13\)

Required knowledge

Basic knowledge of control engineering is required.

2.3 Hardware and software components

2.3.1 Validity

This application example was created and tested with

- STEP 7 V15.1 Update 1 and higher
- S7-1200 CPU Firmware V4.3 and higher / S7-1500 CPU Firmware 2.6 and higher
- Technology object "PID_Compact" V2.3 for S7-1200 / V2.4 for S7-1500

Note The version differences of the controller can be found in the chapter "New features of PID_Compact" in the function manual for PID control (\13\).

2.3.2 Components used

This application example was created using the following components:

Hardware components

Table 2-1

Components	Qty.	Article number	Note
SIMATIC HMI KTP900 BASIC	1	6AV2123-2JB03-0AX0	Optional (can also be simulated in WinCC)
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 V4.3
Fan/motor with analog speed control (0 to 10V / 0 to 20mA)	1	Fan/motor manufacturer	- Without integrated speed control electronics - Optionally with integrated actual speed feedback
HTL INCREMENTAL ENCODER 1000 I/U, OP. VOLT. 10-30V CLAMPING FLANGE, SHAFT 10 MM FLANGE SOCKET RADIAL	1	e.g. 6FX2001-4QB00	Optional if fan/motor does not provide integrated actual speed feedback
SIGNAL BOARD SB 1232, 1 AQ, (12 bit resolution)	1	6ES7232-4HA30-0XB0	Optional (for fan/motor control with 0 to 20 mA current output)
Programming device	1		With Ethernet connection
CPU 1511C-1 PN, 175 KB prog, 1 MB data	1	6ES7511-1CK01-0AB0	Firmware V2.6
SIMATIC S7 Memory Card, 24 MB	1	6ES7954-8LF03-0AA0	Can be plugged into the S7-1500 as load memory

Software components

Table 2-2

Components	Qty.	Article number	Note
SIMATIC STEP 7 Professional V15.1	1	6ES7822-1AA05-0YA5	• Contains WinCC Basic V15.1 • With update 1 (\8\) and HSP0276 (\12\) • for S7-1200 and S7-1500
SIMATIC STEP 7 Basic V15.1	1	6ES7822-0AA05-0YA5	• Contains WinCC Basic V15.1 • With update 1 (\8\) • With HSP0276 (\12\) for S7-1200 Firmware V4.3

Example files and projects

The following list contains all files and projects used in this example.

Table 2-3

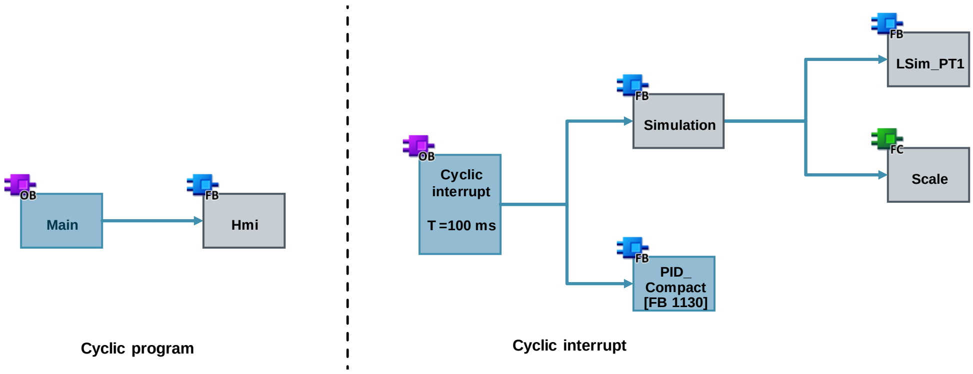
Components	Note
100746401_S71200_PidCompact_TiaV15.1_PROJ_V2.0.zip	TIA Portal project for S7-1200
100746401_S71500_PidCompact_TiaV15.1_PROJ_V2.0.zip	TIA Portal project for S7-1500
100746401_S71x00_PidCompact _DOC_V2.0_de.pdf	this document.

3 Principle of operation

3.1 Complete overview

Figure 3-1 shows the chronological sequence of the block calls in the control part of the TIA Portal project.

Figure 3-1



The example program is called in the following OBs:

- OB "Main", from which the FB is called for the HMI transfer.
- Cyclic interrupt OB "CyclicInterrupt", which cyclically calls the compact controller and the simulation blocks every 100 milliseconds.

The tag transfer between the functions takes place via the data block DB "Tags" and the instance data block of the controller DB "InstPidCompact".

3.2 OB "Main"

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

3.2.1 FB "Hmi"

Figure 3-2

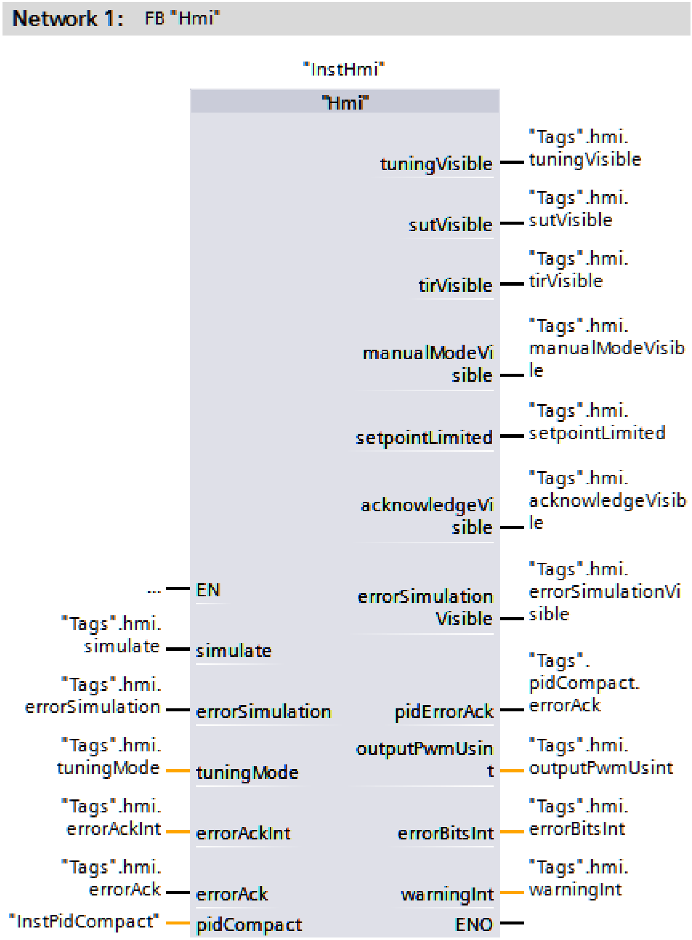


Table 3-1

	Name	Data type	Description
Input	simulate	Bool	Enabling simulation
	errorSimulation	Bool	Simulation of a sensor error
	tuningMode	Int	Tuning type selection (1=pre-/2=fine tuning)
	errorAckInt	Int	Acknowledgment tag for HMI bit messages
	errorAck	Bool	HMI request to clear error messages
Output	tuningVisible	Bool	Visibility of the tuning
	sutVisible	Bool	Visibility of the pretuning
	tirVisible	Bool	Visibility of the fine tuning
	manualModeVisible	Bool	Visibility of the switch to manual mode.
	setpointLimited	Bool	Violation of setpoint limit specifications
	acknowledgeVisible	Bool	Visibility of HMI request to clear error messages
	errorSimulationVisible	Bool	Visibility of the simulation of a sensor error

	Name	Data type	Description
	pidErrorAck	Bool	Deleting error messages (PID controller)
	outputPwmUsint	USInt	PWM signal as data type "USInt" for HMI display
	errorBitsInt	Int	Error message as data type "Int" for HMI display
	warningInt	Int	Warnings as data type "Int" for HMI display
InOut	pidCompact	PID_Compact	Transfer of the instance data block of the FB "PID_Compact".

Tags that the HMI requires for the visibility animation of objects and elements are defined in the "Hmi" function block.

More detailed descriptions can be found in the network headings.

3.3 OB "CyclicInterrupt"

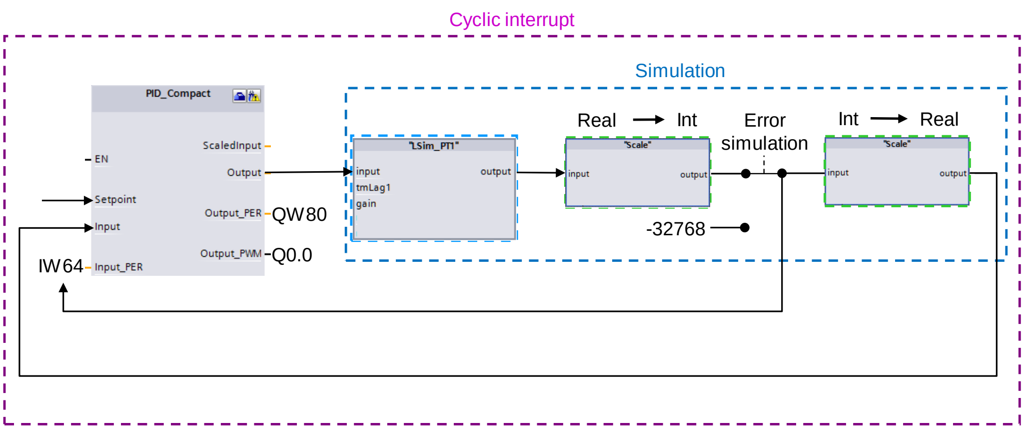
The actual program (the call of the compact controller "PID_Compact") takes place in the cyclic interrupt OB, since discrete software controls must be called at a defined time interval.

100ms was selected as the constant time interval of the sampling time of the OB "CyclicInterrupt".

Program Overview

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

Figure 3-3



The following peripheral connection for controlling a real system is made in the example project:

Table 3-2

Tag	S7-1200	S7-1500
"PID_Compact".Input_PER	IW64	IW0
"PID_Compact".Output_PER	QW80	QW0
"PID_Compact".Output_PWM	Q0.0	Q4.0

Description

In the example project, the compact controller "PID_Compact" accesses the peripheral signals from Table 3-2. This calculates the manipulated variable from the control deviation = setpoint - process value as a function of the PID parameters. The manipulated variable can be output either as an analog or digital pulse width modulated signal to the peripheral control outputs.
For simulation, the manipulated variable is transferred to the function block "LSim_PT1" as a floating point number.
The FB "LSim_PT1" simulates a controlled system with PT1 behavior and thus outputs the simulated process value as a floating point number.
This is converted into an analog value via the FC "Scale".
When the "Error simulation" is activated, the analog process value is overwritten with the erroneous value (-32768).
In addition, the simulated analog process value is converted via the FC "Scale" into the corresponding floating point number for the "Input" input of the FB "PID_Compact".

By deactivating the FB "Simulation" you can switch off the plant simulation and control a real controlled system (signal evaluation via the control periphery) with the FB "PID_Compact".

3.3.1 FB "PID_Compact"

STEP 7 V15.1 supplies the technology object "PID_Compact" in version 2.3 for the S7-1200 or version 2.4 for the S7-1500 with the installation.
This function block was specially developed for the control of proportional actuators.

Figure 3-4

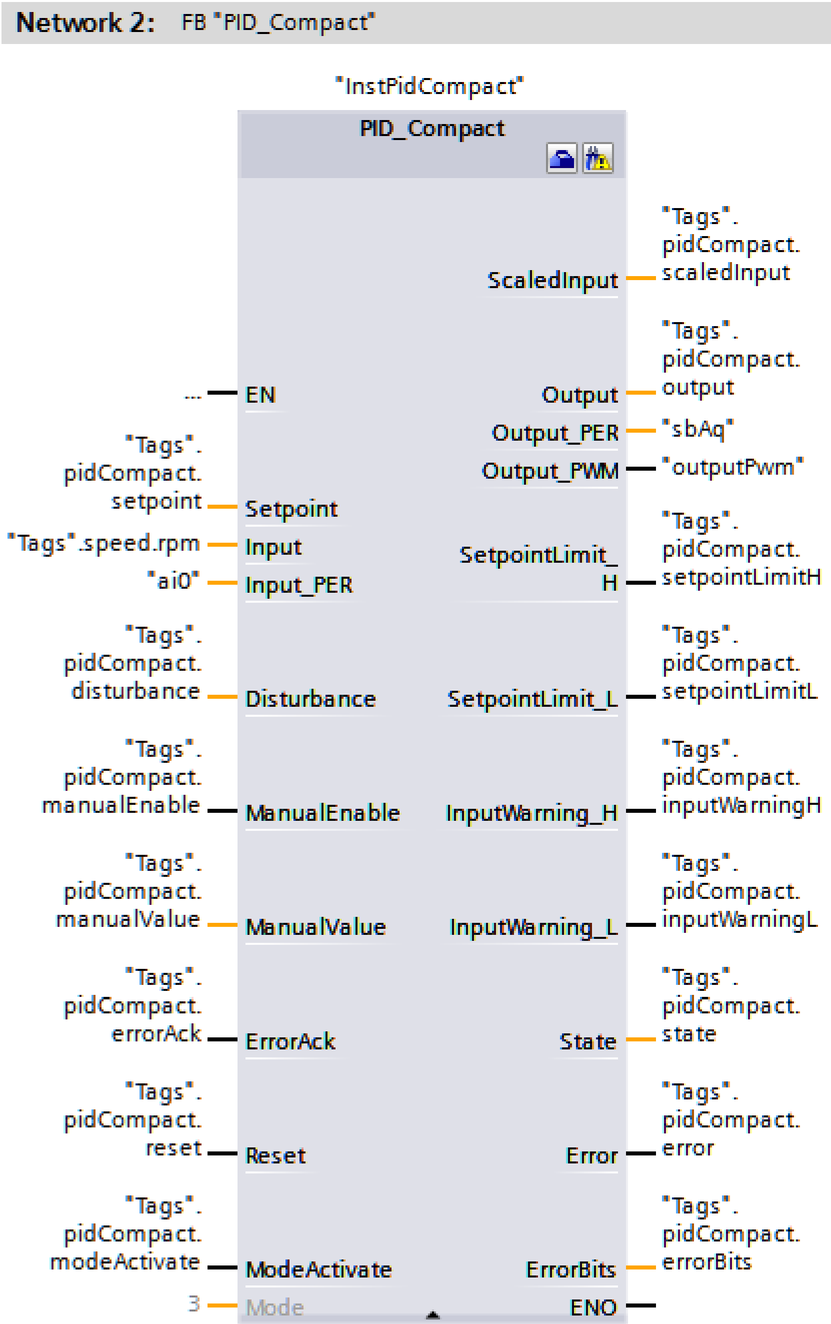


Table 3-3

	Name	Data type	Description
Input	Setpoint	Real	Setpoint input
	Input	Real	Process value in REAL format
	Input_PER	Int	Analog process value
	Disturbance	Real	Disturbance selection
	ManualEnable	Bool	Activating the "Manual mode" operating mode
	ManualValue	Real	Manual value
	errorAck	Bool	Deletion of error messages / warnings
	Reset	Bool	Resetting, restarting of the controller

	Name	Data type	Description
	ModeActivate	Bool	Enable "Mode" operating mode
Output	ScaledInput	Real	Scaled process value
	Output	Real	Output value in REAL format
	Output_PER	Int	Analog output value
	Output_PWM	Bool	Pulse width modulated output value
	SetpointLimit_H	Bool	Setpoint is fixed at the upper limit
	SetpointLimit_L	Bool	Setpoint is fixed at the lower limit
	InputWarning_H	Bool	Process value has exceeded upper warning limit
	InputWarning_L	Bool	Process value has undershot lower warning limit
	State	Int	Display of the current operating mode of the PID controller (0=Inactive,1=SUT,2=TIR,3=Automatic,4=Manual)
	Error	Bool	At least one error message present
	ErrorBits	DWord	Error message
InOut	Mode	Int	Operating mode default (see "State")

The FB "PID_Compact" is called in the alarm OB "CyclicInterrupt".
You can find the "InstPidCompact" instance data block in the "Technology Objects" folder:
This can be opened by right-clicking -> "Open DB editor".

Note A more detailed description of the compact controller can be found in the STEP 7 V15.1 online help. Select the function block "PID_Compact" in the program call (see Figure 3-4) and press F1.

3.3.2 FB "Simulation"

Figure 3-5

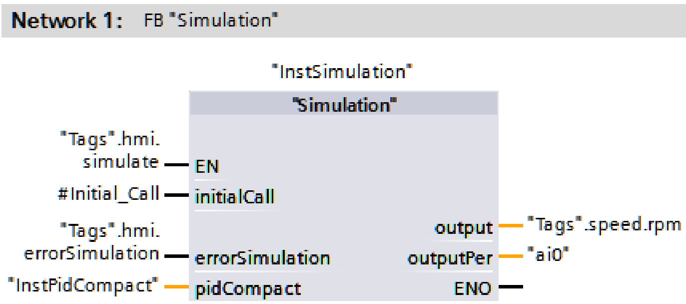


Table 3-4

	Name	Data type	Description
Input	initialCall	Bool	First call of the cyclic interrupt
	errorSimulation	Bool	Simulation of a sensor error
Output	output	Real	Simulated process value as REAL format
	outputPer	Int	Simulated analog process value
InOut	pidCompact	PID_Compact	Transfer of the instance data block of the FB "PID_Compact".

The FB "Simulation" simulates the system to be controlled as a PT1 element. In addition, the conversion to the process value as an analog value or floating point number takes place within the block. The FB "Simulation" writes directly to the selected peripheral input of the technology object "PID_Compact". When the FB "Simulation" is deactivated, the controller uses the process value of a connected sensor. The FB "Simulation" calls the following blocks:

- FB "LSim_PT1"
- FC "Scale"

The FB "Simulation" is called in the same cyclic interrupt as the compact controller "PID_Compact".

For more information, refer to the network headings and the following description.

FB "LSim_PT1"

The function block "LSim_PT1" simulates the continuous behavior of a PT1 system. This block comes from the library for controlled system simulation (\7\). You can also find a detailed description of the FB "LSim_PT1" here. In this application example, the "LSim_PT1" system simulation module is designed with a delay time of 3 seconds.

Note Please note that changes of the system parameters are only accepted after activation of the input "calcParam" (implemented in the example project as restart of the CPU).

FC "Scale"

The "Scale " function is used for linear conversion according to the following formula:

$$output = \frac{outputRef2 - outputRef1}{inputRef2 - inputRef1} \cdot (input - inputRef1) + outputRef1$$

Figure 3-6

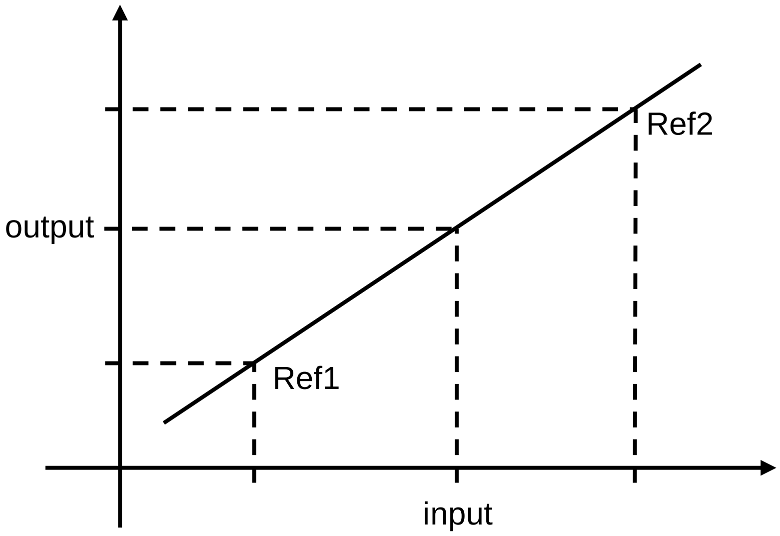


Figure 3-7

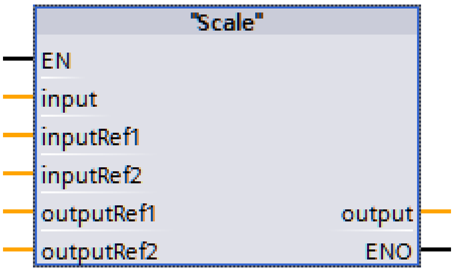


Table 3-5

	Name	Data type	Description
Input	input	LReal	Value to be converted
	inputRef1	LReal	Input value of reference point 1
	inputRef2	LReal	Input value of reference point 2
	outputRef1	LReal	Output value of reference point 1
	outputRef2	LReal	Output value of reference point 2
Output	output	LReal	Output value

By selecting the data type "LReal", the correct conversion of or into the attached actual parameters is ensured.

The FC "Scale" converts the process output into an analog value in order to simulate the behavior of the controller in the event of an error.
The fault occurs in a real system due to the failure of the process value sensor (e.g. due to wire breakage).
In the simulation, this is achieved by overwriting the analog process value with a value outside the measuring range (-32768) (see Figure 3-3).

Then FC "Scale" converts the resulting analog value into a floating point value for the process value selection "Input" of the FB "PID_Compact".

4 Installation and commissioning

4.1 Hardware adaptation

The following table provides information on the controllers used and the connection options to a real controlled system.

Table 4-1

Signal	CPU 1211C	CPU 1511C-1 PN
Analog process value (0-10V)	IW64	IW0
Analog manipulated value (0-10V)	QW80 (with SB 1232 AQ)	QW0
Digital PWM signal	Q0.0	Q4.0
Digital recording of the speed (A-B track) of the incremental encoder via fast counters	DI 0.0 DI 0.1	DI 10.0 DI 10.1

Depending on the version of your selected actuator, you may have to adjust the hardware configuration.
The configuration options for operating the "PID_Compact" compact controller are presented below.

Input signal

The controlled variable is recorded by the peripheral "Input_PER" as a processed floating point number "Input" or as an analog value. The "PID_Compact" offers the conversion of the analog value into the physical unit in the configuration mask.

For controlled variable acquisition, modules are offered for analog value recording as well as for temperature recording via thermocouples or resistance thermometers.

Output signal

The "PID_Compact" offers the control of the actuator via an analog output or via a digital pulse width modulated transistor output.

Note

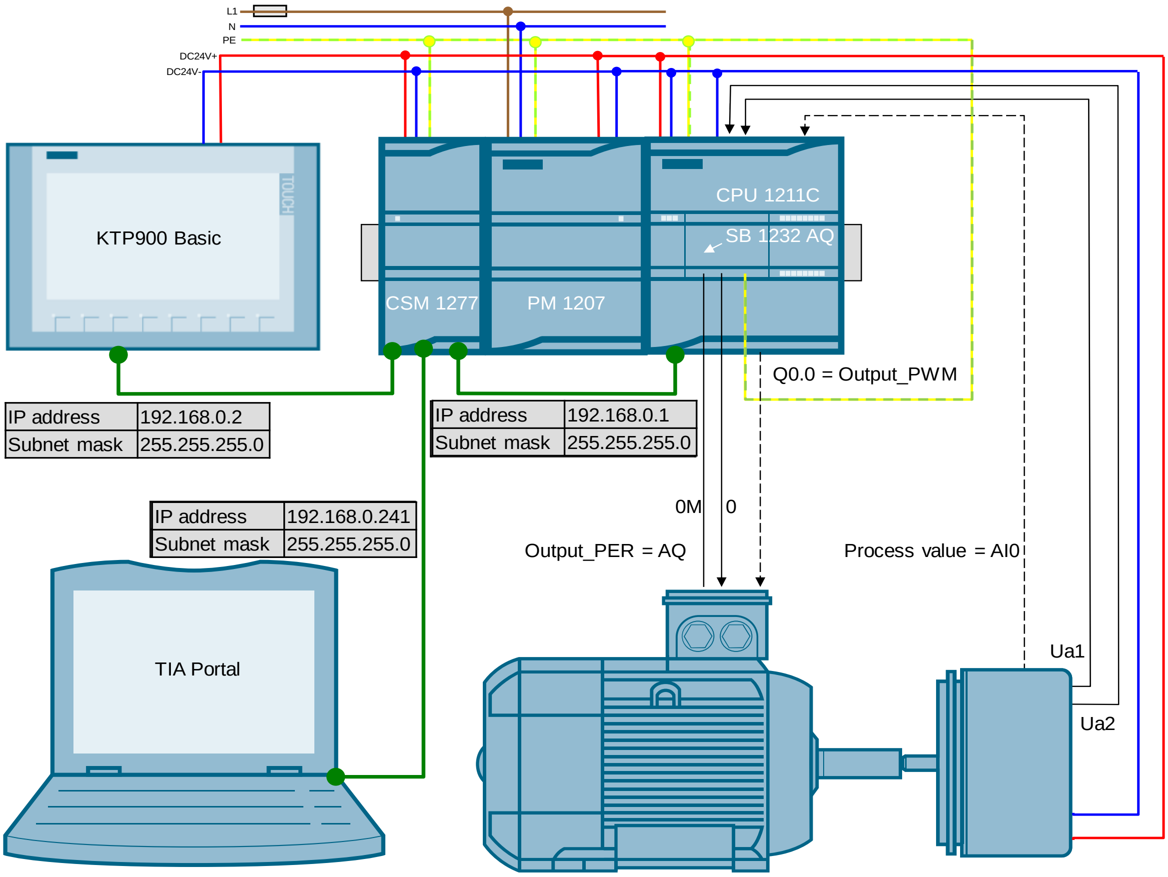
Further information on the selection of your peripherals or their wiring can be found in the hardware catalog in the TIA Portal or:

- in Section A "Technical data" in the S7-1200 manual (\3\)
- in the manual "SIMATIC S7-1500/ET 200MP Manual Collection" (\9\)
- via the TIA selection tool (\11\)

Hardware installation

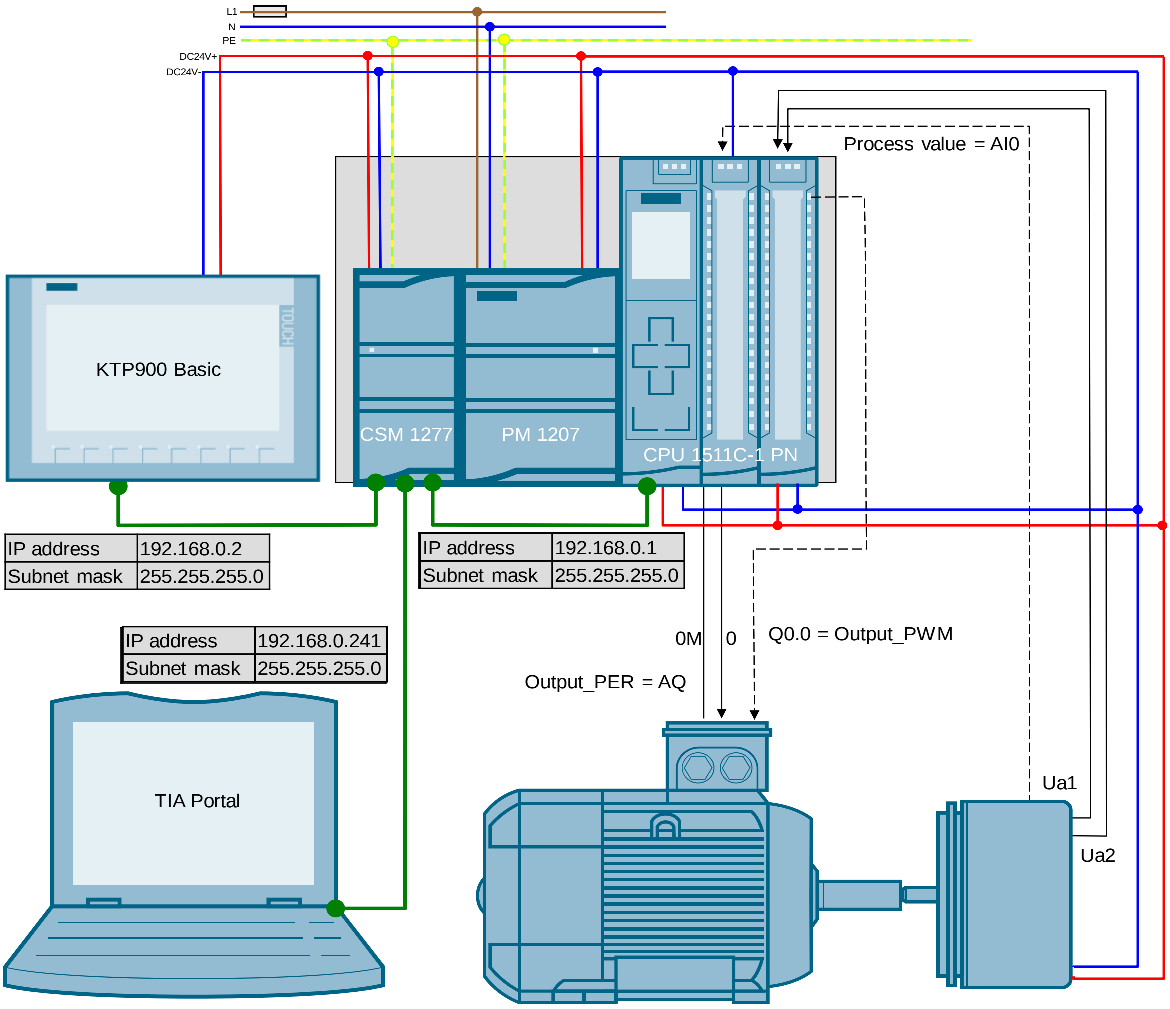
The following figure shows the hardware structure of the application example with a SIMATIC S7-1200.

Figure 4-1



The following figure shows the hardware structure of the application example with a SIMATIC S7-1500.

Figure 4-2



Installing the hardware

Table 4-2

No.	Action	Note
1	Adapt the peripherals of the S7-1200/S7-1500 to the actuator you are using.	See Chapter 4.1.
2	Mount all required components on a DIN rail (S7-1200) or S7-1500 profile rail.	
3	Wire and connect all required components as described.	S7-1200 Manual (\3\) Section A "Technical data" or S7-1500 Manual Collection (\9\)
4	Finally, activate the power supply for the SIMATIC PM 1207.	

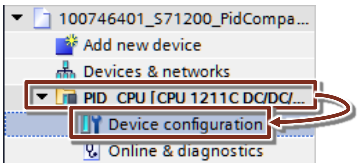
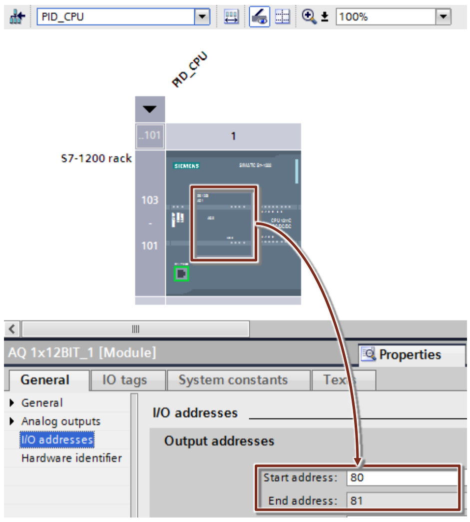
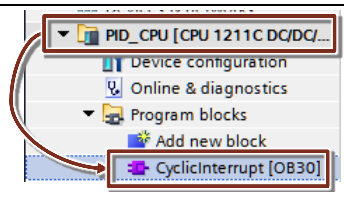
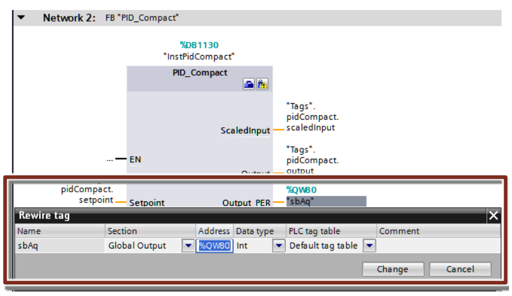
4.2 Configuration

4.2.1 Transferring 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 shown using the example of a signal board 1232 AQ 1x12 bit for the SIMATIC S7-1200:

Table 4-3

No.	Action	Note
1.	Open the device configuration of the controller "PID_CPU".	
2.	Mark the signal board 1232 AQ 1x12 Bit in the device view of the CPU. Read the output address of the signal board under the menu item "I/O addresses": • Start address: 80 • End address: 81 This means that: The address via which the analog value of the SB 1232 AQ 1x12 Bit is output is: **QW80**	
3.	Open the OB "CyclicInterrupt" in the control part of the project.	
4.	Since the analog manipulated variable is output to the actuator via the signal board, transfer the output "Output_PER" of the FB "PID_Compact" to the output word **QW80** in network 2. • To do this, select the linked tag "sbAq" and right-click to select "Rewire tag...". • Change the address of the tag. %QW80 Accordingly, you can also adapt the other peripheral links from Table 4-1 to your signal selection.	

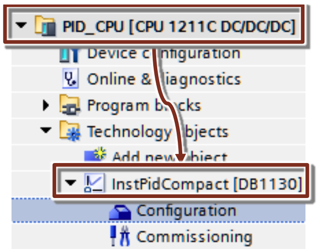
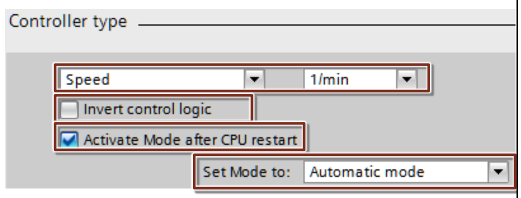
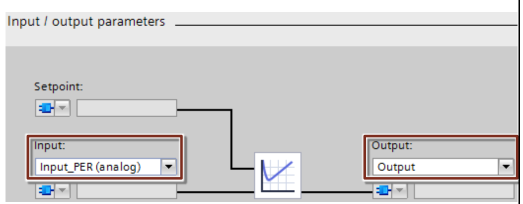
4.2.2 Configure PID controller

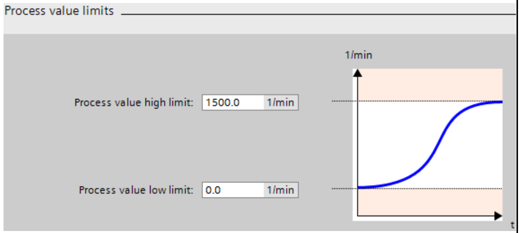
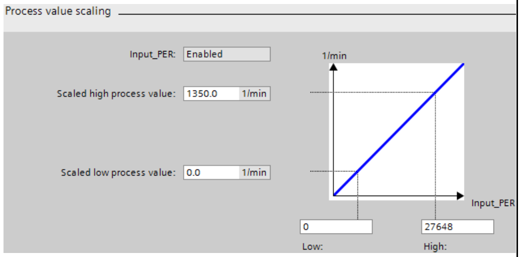
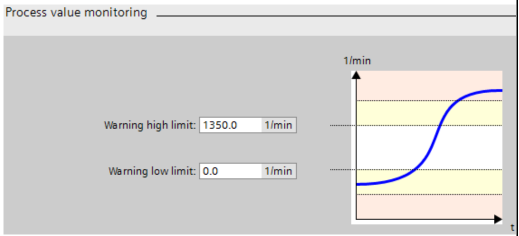
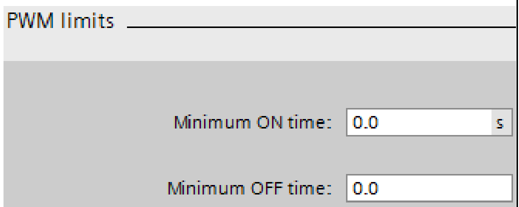
The configuration of the technology object "PID_Compact" determines the functionality of the compact controller.

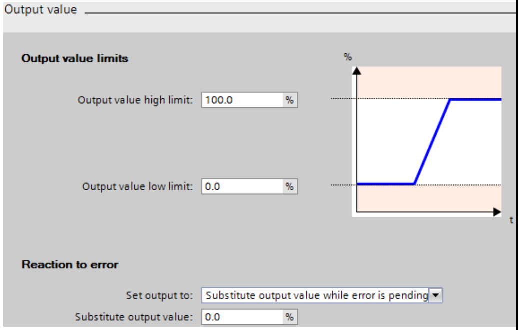
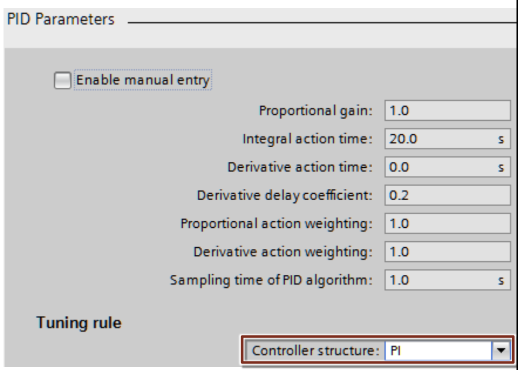
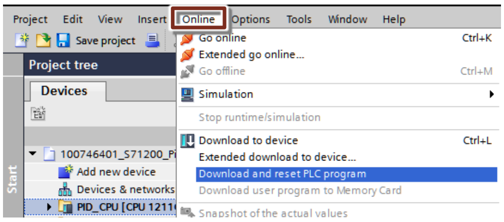
The settings made determine the starting values with which the PID controller restarts after a cold or warm start (e.g. power failure).

You can find a more detailed description in the S7-1200 manual (\3\) → Chapter 10.2.5 or in the function manual for PID control (\13\) → Chapter "Configuring PID_Compact V2".

Table 4-4

No.	Action	Note
1.	Open the configuration editor by selecting the CPU -> Technology objects -> InstPidCompact -> Configuration.	
2.	Open the "**Control mode**" submenu in the basic settings: Define • which physical unit is to be used for the display of setpoint and process value • whether the controller output is to be inverted • whether the controller is to remain "inactive" **after restarting the CPU** or changes to the operating mode stored at "Mode" • under "Set mode to:" the operating mode that is to be activated after a complete loading into the device. (The prerequisite for this is that the "Mode" parameter is not linked -> see Figure 3-4)	
3.	Adjust the "Input/Output parameters" submenu in the basic settings to the sensors/actuators used: • Process value as prepared floating point number "Input" or as analog value "Input_PER". • Manipulated variable as floating point number "Output", as analog value "Output_PER" or as digital pulse width modulated signal "Output_PWM".	

No.	Action	Note
4.	Determine the limits of the scaled process value in the submenu "Process value limits" in the process value settings. **Note:** Make sure that the upper and lower limits of the process value are set correctly, as the controller interprets violation of these limits as an error and reacts according to the settings "Behavior in the event of an error" (see no. 8)!	
5.	When using the analog process value "Input_PER" in the submenu "**Process value scaling**" in the process value settings, determine the value pairs for the linear conversion into the scaled process value.	
6.	Open the "**Process value monitoring**" in the Advanced settings: Here you can specify warning limits where a warning bit is activated if they are exceeded or not reached.	
7.	Open the "**PWM Limits**" in the Advanced settings: For adaptation to the actuator inertia, you can specify minimum switch-on or switch-off times here. **Note:** These settings are also effective when using another control variable signal ("Output" or "Output_PER")!	

<table>
<thead>
<tr><th>No.</th><th>Action</th><th>Note</th></tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td>Open the "Output value" in the Advanced settings:
Output value limits
Determine the percentage limits of the signal to be output to the actuator.
Response to errors
Check whether, in the case of an error,
the controller is switched to the inactive state,
the current manipulated variable is maintained for the duration of the error or
a substitute output value to be specified is to be output permanently as a manipulated variable or for the duration of the error.</td>
<td>
</td>
</tr>
<tr>
<td>9.</td>
<td>Open the "PID Parameters" in the Advanced settings:
Here you can enter the start values of the controller parameters manually.
They are then written as start values in the instance data block of the "PID_Compact" and transferred as actual values after a cold start (load project into the controller).
Rule for tuning
Depending on the selected controller structure, the initial values for the setting rules of the pre- or fine tuning are set to
"PID according to Chien, Hrones and Reswick" or "PID automatic" or
"PID according to Chien, Hrones and Reswick" or "Ziegler-Nichols PI".
Select "PI" as the controller structure for speed control.</td>
<td>
</td>
</tr>
<tr>
<td>10.</td>
<td>Save the project.
Mark the program folder of the S7-1x00 and transfer the program to the controller via "Online/PLC program load into device and reset", so that the settings made are used as start values of the technology object when the CPU is started.</td>
<td>
</td>
</tr>
</tbody>
</table>

Note

Changes to the start values of a data block are not accepted as actual values until the next STOP/RUN transition (for non-retentive data types).

4.3 CPU simulation with PLCSIM

You can simulate the technology object "PID_Compact" V2.x for CPU S7-1500 with PLCSIM. The simulation of PID_Compact V2.x with PLCSIM for CPU S7-1200 is not supported.

The actual timing of a cyclic interrupt OB may vary more with a simulated PLC than with "real" PLCs. In the standard configuration, PID_Compact automatically determines the time between calls and monitors them for fluctuations.

When simulating PID_Compact with PLCSIM, a sampling time error (ErrorBits = DW#16#00000800) can therefore be detected. This leads to the abort of ongoing tuning.

To prevent this, you should configure PID_Compact for simulation with PLCSIM as follows:

• CycleTime.EnEstimation = FALSE

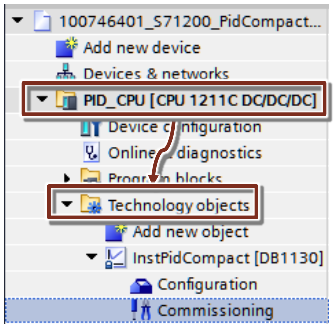
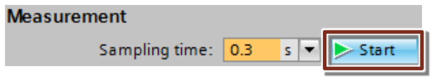
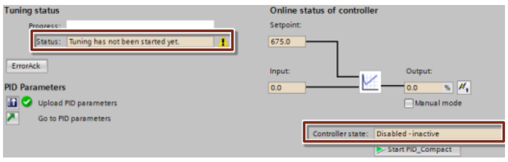
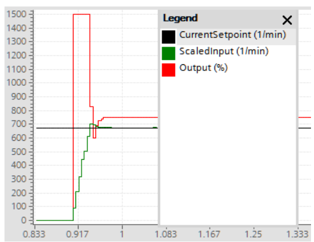
• CycleTime.EnMonitoring = FALSE

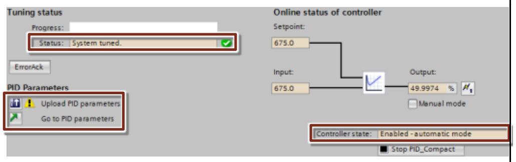
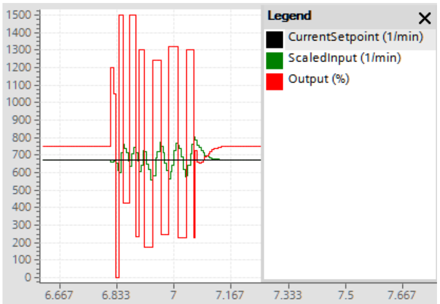
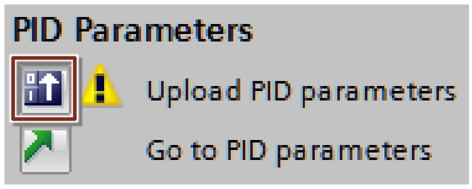
• CycleTime.Value: Assign the time cycle of the cyclic interrupt OB in seconds to this variable.

4.4 Commissioning the compact controller

Below you will learn how to optimize the PID_Compact with the aid of the commissioning wizard.

Table 4-5

No.	Action	Note
1.	Open the commissioning editor by selecting the CPU -> Technology objects -> InstPidCompact -> Commissioning.	
2.	Start the measurement.	
3.	The tuning status indicates that no tuning has been started yet and the controller is in the "Deactivated - Inactive" state after the first CPU start (see Table 4-4, No. 2).	
4.	You get the best PID parameters when you perform pretuning and fine tuning. Requirements for the pretuning are: • ManualEnable = FALSE, Reset = FALSE • PID_Compact is in "Manual", "Inactive" or "Automatic" mode. • The setpoint and the process value are within the configured limits (see Table 4-4, No. 4). • The difference between setpoint and process value is greater than 30 % of the difference between upper limit of process value and lower limit of process value. • The distance between setpoint and process value is > 50% of the setpoint. If possible, enter a target value in the middle field of the process value range (e.g. via an observation table; the start value of the target value is already predefined accordingly in the project). Start the pretuning.	
5.	The pretuning determines the process response to a jump in the output value and searches for the turning point. The PID parameters are calculated from the maximum gradient and the dead time of the controlled system. You can follow the pretuning and the subsequent adjustment to the setpoint with the found PID parameters via the trend.	

No.	Action	Note
6.	After successful pretuning, the controller switches to automatic mode. The determined values can be viewed via "Go to PID parameters". Via "Load PID parameter" the determined values are written to the project as start values in the instance data block of the "PID_Compact".	
7.	Requirements for fine tuning are: • ManualEnable = FALSE, Reset = FALSE • The setpoint and the process value are within the configured limits. • The control loop is steady at the operating point. The operating point is reached when the process value corresponds to the setpoint. • No interference is expected. • PID_Compact is in Inactive, Automatic or Manual mode. If possible, enter a target value in the middle field of the process value range (e.g. via an observation table; the start value of the target value is already predefined accordingly in the project). Now start the fine tuning.	
8.	The fine tuning generates a constant, limited oscillation of the process value. The amplitude and frequency of this oscillation are used to optimize the PID parameters for the operating point. All PID parameters are recalculated from the results. The PID parameters from fine tuning usually show better command and error behavior than the PID parameters from pretuning. You can track the fine tuning via the trend.	
9.	After successful completion of the fine tuning you can load the determined PID parameters into the project as start values in the instance data block of the "PID_Compact" in order to load and save the project changes into the CPU afterwards.	

You can find a more detailed description in the S7-1200 manual (\3\→ Chapter 10.2.7) and in the function manual on PID control (\13\ → Chapter "Commissioning PID_Compact V2").

Note

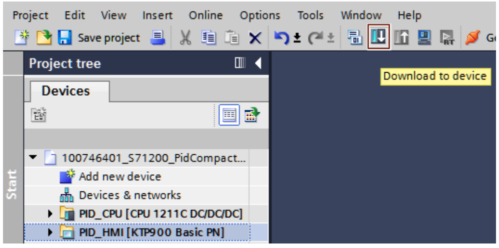
The PID parameters are retentively stored in the instance data module of the "PID_Compact" compact controller.
During a warm start (voltage recovery), the last values passed through are retained.
The start values are only loaded during a cold start (transfer of the project in the operating state STOP or overall reset of the memory via MRES).

4.5 HMI device

Load HMI project part into KTP900 Basic

Connect your PG/PC directly or via the switch CSM1277 to the HMI.

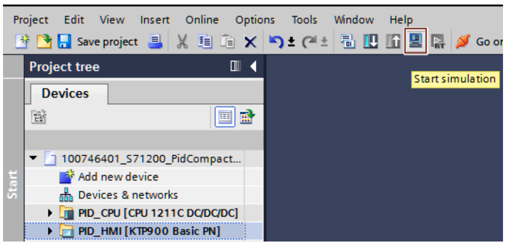
Table 4-6

No.	Action	Note
1.	• Mark the HMI folder "PID_HMI [KTP900 Basic PN]". • Click the "Load to device" button to download the HMI project part into the KTP900 Basic. • Follow the wizard to "Load into device".	

HMI simulation

If you want to use the PG/PC as an operating device, start the HMI simulation as follows:

Table 4-7

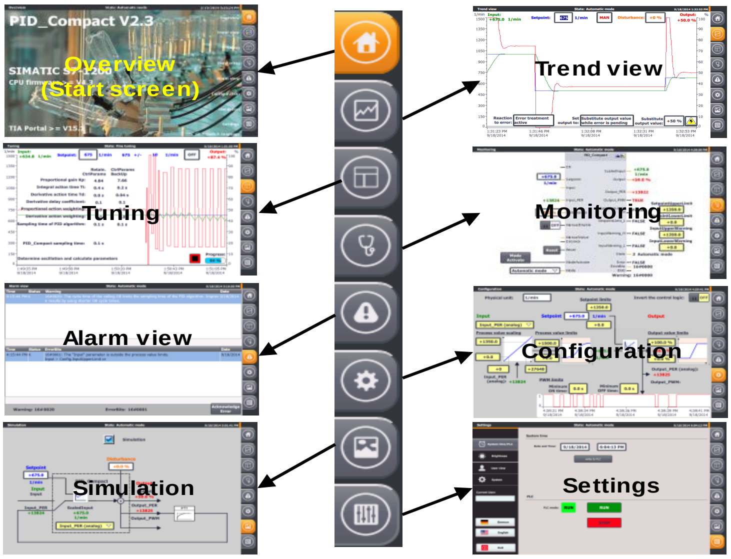
No.	Action	Note
1.	• Mark the HMI folder "PID_HMI [KTP900 Basic PN]". • Click the "Start simulation" button.	

5 Operation of the application example

5.1 Overview

Overview and description of the interface

Figure 5-1



The user interface is made up of 8 menus:

- Start screen (overview)
- Trend view
- Tuning
- Monitoring
- Alarm view
- Configuration
- Simulation
- Settings

5.1.1 Overview (start screen)

The overview screen provides information on the topic dealt with.
The technology object is presented: the compact controller "PID_Compact" available in the controllers:

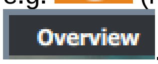
- SIMATIC S7-1200 CPU with Firmware V4.3 and PID_Compact V2.3
- SIMATIC S7-1500 CPU with Firmware V2.6 and PID_Compact V2.4

Configuration is performed with STEP 7 V15.1 (TIA Portal).

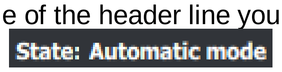
The operation of the right menu bar is also explained.
It is available in every screen

Figure 5-2

-  takes you to the overview screen (this screen).
-  takes you to the trend view (Figure 5-3).
-  takes you to the tuning (Figure 5-4).
-  takes you to the monitoring (Figure 5-5).
-  takes you to the alarm view (Figure 5-6).
-  takes you to the configuration (Figure 5-8).
-  takes you to the simulation (Figure 5-10).
-  takes you to the system functions (Figure 5-11).
- With F8 you can switch between German and English.

The currently selected menu is indicated by the orange background of the symbol: e.g.  (for the overview screen) or the title in the header line (left): 

.

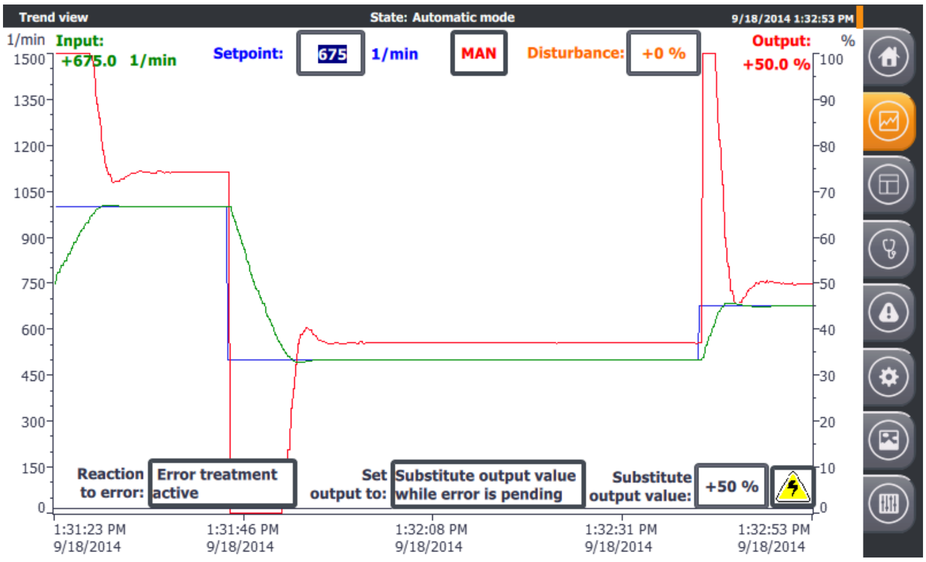
In the middle of the header line you can see which operating mode the controller is currently in: 

(for automatic mode)

The header line is also visible in each image.

5.1.2 Trend view

Figure 5-3



The "Trend view" image shows the time course over 90 seconds.

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

Manual mode

Use **MAN** to change to manual mode.

In manual mode, you can enter the manipulated variable directly via the manual value **Manual value:** **+50 %** (value range 0 to 100 %).

Note	Manual mode is activated here via the value setting at the "Mode" parameter in conjunction with activation via "ModeActivate" (not via "ManualEnable").

Automatic mode

Use **AUTO** to change to automatic mode.

Use **Setpoint:** **675** **1/min** to set the setpoint in automatic mode.

Use **Disturbance:** **+100 %** to specify the disturbance variable which is added directly to the manipulated variable.

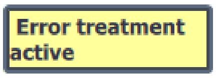
In automatic mode, you can follow the regulation of the disturbance variable in the trend view.

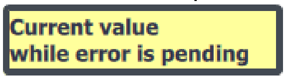
Response to errors

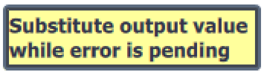
If the process value limits are exceeded (e.g. due to sensor failure), you can predefine the behavior of the controller.

Check whether, in the case of an error,

- the controller should be switched to the inactive state 

- or with active troubleshooting 

  - the current manipulated variable is maintained for the duration 

 or
  - a substitute value 

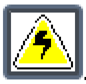
 should be set via 

 as a manipulated variable.

If troubleshooting is active, the manipulated variable (output) is set to the current value or to the substitute output value for the duration of the error in the event of an error.
This behavior occurs in the following operating modes:

- Pretuning
- Fine tuning
- Automatic mode

The controller also switches to inactive in manual mode when this behavior is selected in the event of an error.

In the system simulation mode, the selected behavior in the event of an error can be simulated (switched on and off) via . If an error is activated, the symbol is highlighted in red: 

Note

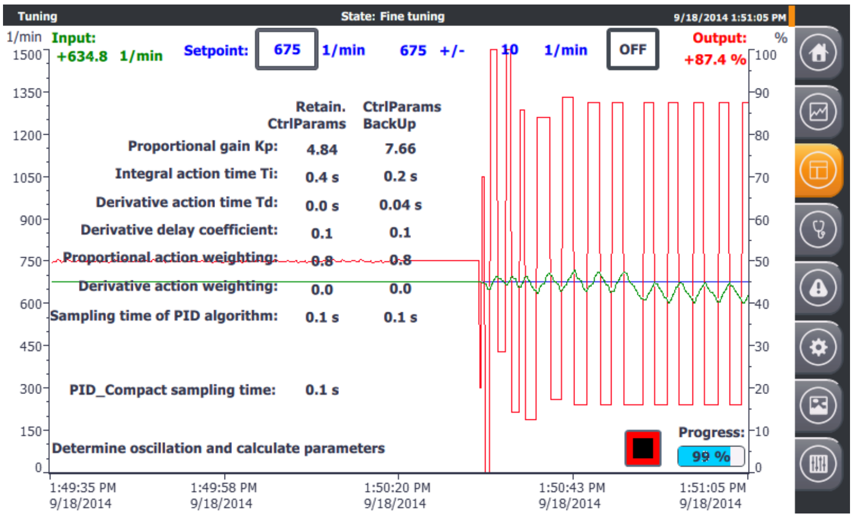
These selection fields are highlighted in yellow, since they are non-retentive data in the instance data block of the "PID_Compact".
You can change them via the operator interface or its simulation to test the function.

To save these default settings even in the event of a power loss, this value must be written as start values in the instance data module of the "PID_Compact".
The configuration wizard offers this function (Table 4-4) with subsequent transfer of the instance data block.

5.1.3 Tuning

In the "Tuning" menu you can set the control parameters automatically or manually.

Figure 5-4



The tuning screen lets you perform

- pretuning
- or fine tuning

from the inactive controller state, manual or automatic mode.

Note Manual operation must not be selected 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

Use **Cancel TuningLevel: 10 1/min** to determine the maximum permitted setpoint change during tuning. If this value is exceeded, the tuning is aborted and, depending on the selected error handling, the controller switches to inactive or returns to the operating mode from which the tuning was started.

The setpoint is frozen during tuning. The limit specifications (frozen setpoint with maximum permitted setpoint change) are displayed: **675 +/- 10 1/min**

Note The parameter "CancelTuningLevel" ensures tuning even with signal noise at the setpoint (e.g. when using a potentiometer).

Use ▶ to start the selected tuning.

Note If possible, enter a target value in the middle field of the process value range to avoid aborting the tuning by reaching the limit.

The tuning status and the percentage progress are displayed during tuning.

Progress:

60 %

Use ■ to cancel the tuning and return to the operating mode from which the tuning was started.

After successful tuning, the determined controller parameters are displayed in the column **Retain. CtrlParams** and the controller parameters are moved to the column **CtrlParams BackUp** before tuning.

The saved "BackUp" parameter set can be reloaded into the controller via **LoadBackUp**.

The current controller parameters ("Retain.CtrlParams") can also be edited manually (displayed by framing the parameter **1.0**).

Note The current controller parameter set ("Retain.CtrlParams") is retentive and remains even after voltage loss.
To start with these parameters even after a cold start, they must be written as start values in the instance DB of the "PID_Compact".
Der commissioning wizard provides this function (Table 4-5, Step 9).

Use **Init** to load the default values of the PLC data type "PID_CompactRetain":

	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 cycle 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 sampling time of the controller and depends on the PWM limitation.

After successful tuning, you can calculate the controller parameters using **Calculate Params** for other tuning methods without repeating the tuning process, depending on the type of tuning.

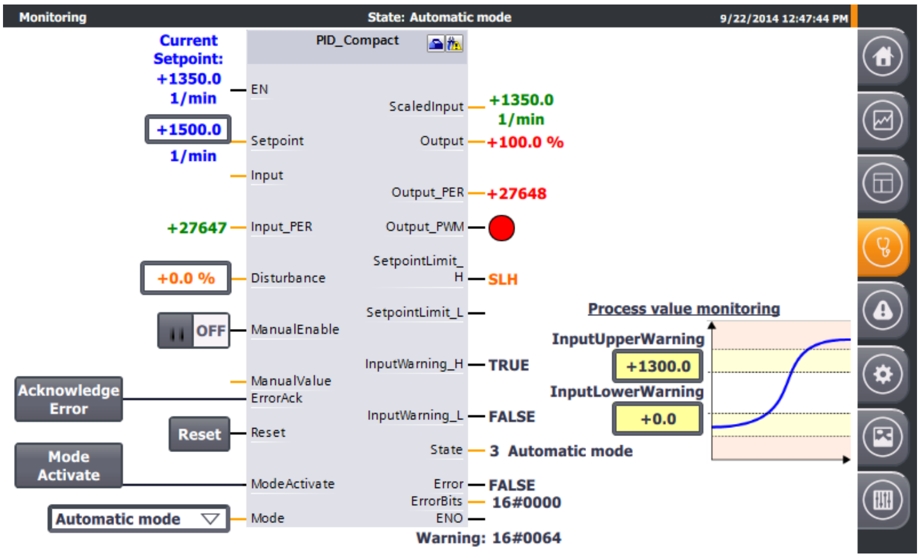
Use **OFF** to deactivate the controller. This operating state is particularly advantageous for pretuning. During pretuning, the control parameters are determined from the response to a setpoint jump.
The process value must not be too close to the setpoint:

- |Setpoint - Input| > 0.3 * |Config.InputUpperLimit - Config.InputLowerLimit| and
- |Setpoint - Input| > 0.5 * |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 input and output values
- edit the following parameters:
  - Setpoint in automatic mode ("Setpoint")
  - Feedforward control in automatic mode ("Disturbance")
  - On/Off switching of manual operation ("ManualEnable")
  - Manual manipulated variable setting in manual mode ("ManualValue")
  - Acknowledgment (reset) of the messages "ErrorBits" and "Warning" ("Acknowledge Error")
  - Resetting the compact controller ("Reset")
  - Changing the operating mode via the selection at the parameter "Mode" and activation via "ModeActivate".
- Test the process value monitoring configuration
  - Editing the upper ("InputUpperWarning") and lower ("InputLowerWarning") warning limits
  - Direct monitoring at the outputs "InputWarning_H" or "InputWarning_L".

Note

In this application example, the reset button restarts the blocks "PID_Compact" and "LSim_PT1". The controller switches to the "Inactive" operating mode. The messages "ErrorBits" and "Warnings" are reset. The controller then starts in the operating mode that is applied to the "Mode" parameter.

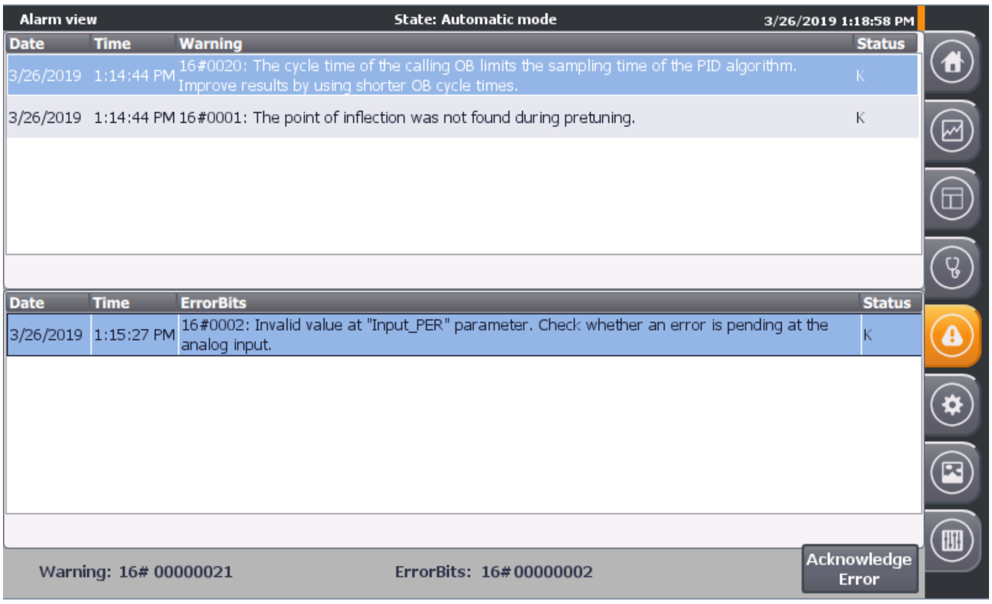
Note

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

5.1.5 Alarm view

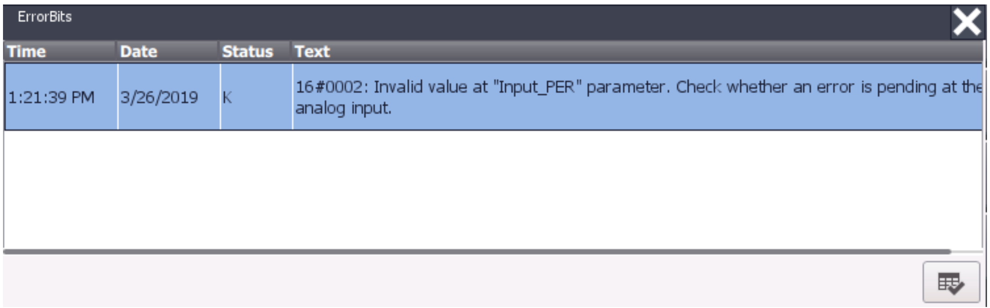
The "Message display" menu displays the current messages at the "ErrorBits" output and at the "Warning" static parameter of the "PID_Compact" as hexadecimal error codes, as well as in text form with time stamp and status.

Figure 5-6



The error messages "ErrorBits" are also displayed globally when they occur.

Figure 5-7



You have the option of acknowledging errors that are no longer pending via . This deletes all messages that are no longer present at "ErrorBits" and "Warning" via the "ErrorAck" input.

Within the message display and the other masks, execute this function via 

.

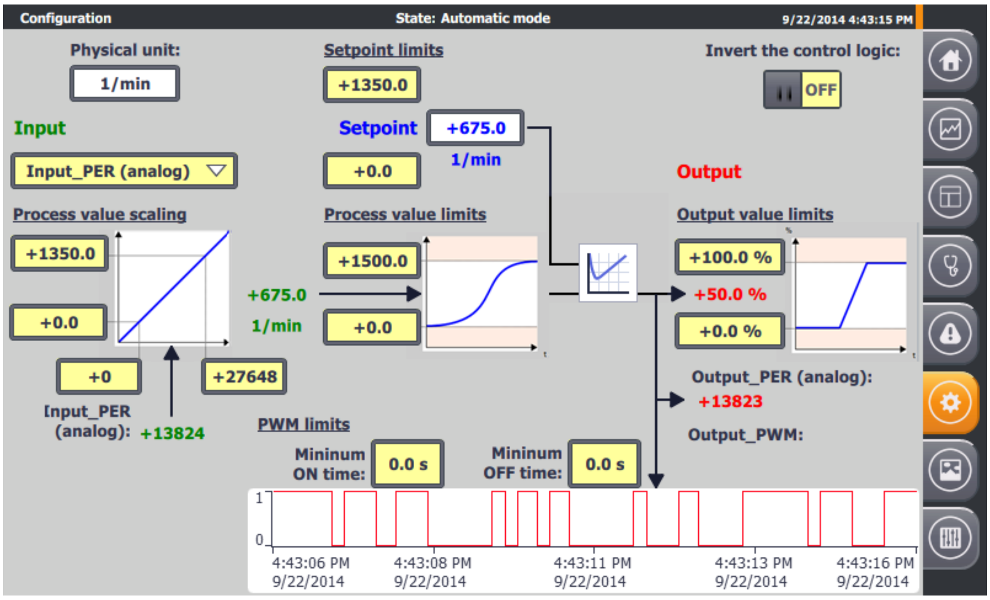
The button is only visible for pending messages ("ErrorBits" or "Warning").

Note The complete description of all error messages can be found in the online help of STEP 7 (TIA Portal). Select the function block "PID_Compact" in the program call (see Figure 3-4) and press F1.

5.1.6 Configuration

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

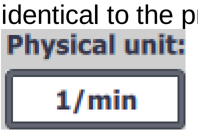
Figure 5-8



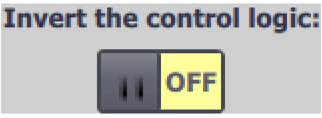
Here you can change the following settings during runtime:

Basic settings

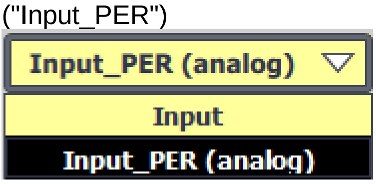
- Control type
  - Specification of the displayed physical unit (limited to 5 characters; not identical to the preselection in the configuration wizard)



  - Inversion of the rule sense (see Table 4-4, Step 2)



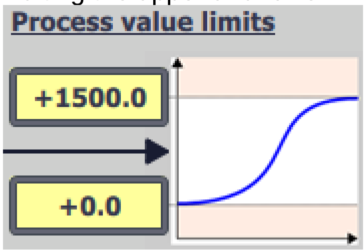
- Input/output parameters (see Table 4-4, Step 3)
  - Porcess value signal selection: Floating-point number ("Input") or analog ("Input_PER")



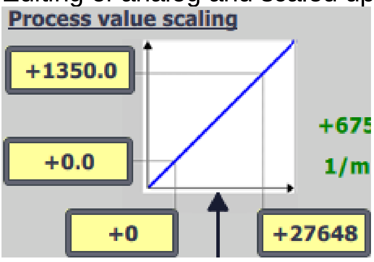
Process value settings

- Process value limits (see Table 4-4, Step 4)

- Editing the upper and lower limits of the process value



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



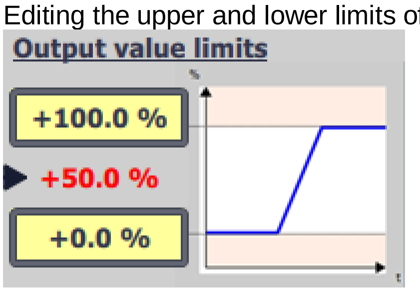
Note

The process value scaling serves for linear conversion of the analog value "Input_PER" into the scaled process value "ScaledInput".
In simulation mode, however, this conversion is also required when selecting the process value acquisition via the floating point value "Input" (see Figure 3-3).

Advanced settings

- PWM limits (see Table 4-4, Step 7)
  - Editing of minimum switch-on and switch-off time for adaptation to possible actuator inertia

- Output value limits (see Table 4-4, Step 8)
  - Editing the upper and lower limits of the output value

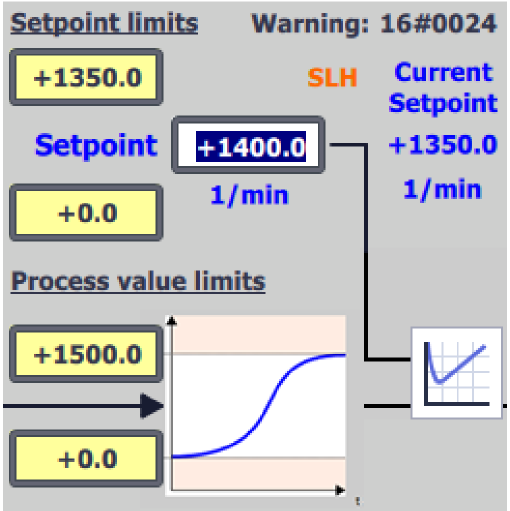


Setpoint limits

The compact controller "PID_Compact" automatically limits the setpoint to the "Process value limits".
However, you can also limit the setpoint to a smaller range above the setpoint limits.

"PID_Compact" automatically takes the narrower limit.

Figure 5-9



In the event of a limit violation, a corresponding internal restriction takes place. The actual setpoint "CurrentSetpoint" is displayed and the output parameter "SetpointLimit_H" SLH or "SetpointLimit_L" SLL indicates the limit violation. A corresponding warning message appears (16#0004).

This screen is used to get to know the compact controller settings and their characteristics (especially for simulation mode).

Note

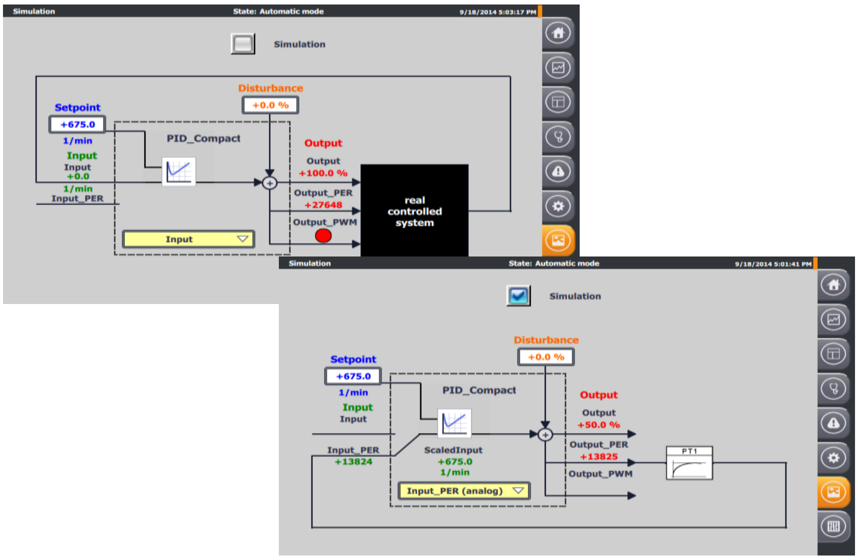
These input/output fields are highlighted in yellow, since they are non-retentive data in the instance data block of the "PID_Compact".
You can change them via the operator interface or its simulation to test the function.

To save these default settings even in the event of a power loss, these values must be written as start values in the instance data module of the "PID_Compact".
The configuration wizard offers this function (Table 4-4) with subsequent transfer of the instance data block.

5.1.7 Simulation

The simulation screen allows you to switch between a real and a simulated control system.

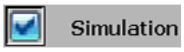
Figure 5-10



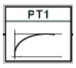
The block diagram of the PID control is shown with:

- the setpoint
- the process value display with process value signal selection
  - "Input" as floating-point number or
  - "Input_PER" as analog value with internal conversion ("ScaledInput")
- the disturbance specification "Disturbance".
- the output of the manipulated variable
  - as a percentage floating point number "Output"
  - as analog value "Output_PER"
  - as pulse-width modulated digital signal "Output_PWM"

If the simulation is not switched on, the FB "Simulation" is deactivated and the controller processes the signals via the control peripherals (Table 4-1).

When the simulation 

is switched on, the figure shows the block diagram structure, how the input signals for the controller are calculated:

The output of the PT1 system simulation block 

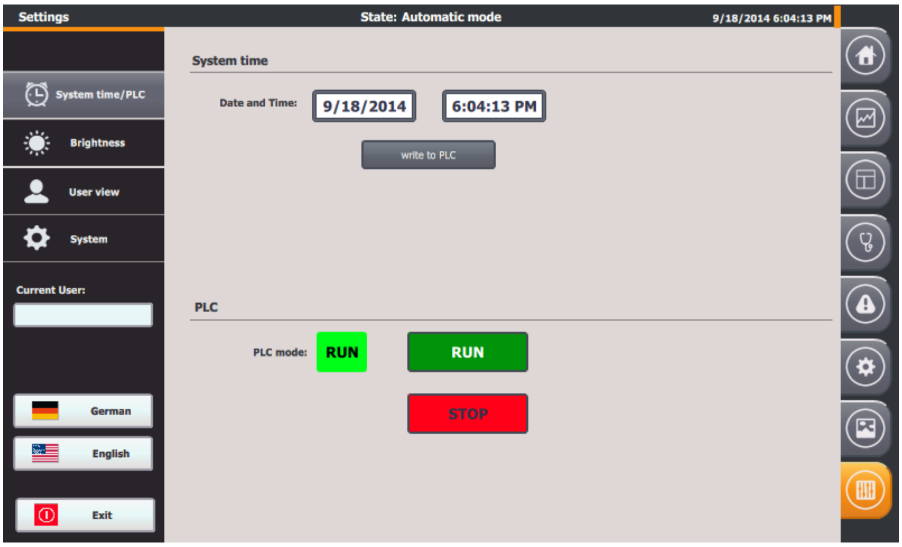
supplies the process value.

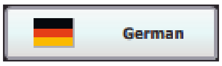
5.1.8 Settings

The settings menu consists of the following screens

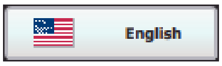
- System time / CPU
- Brightness
- User
- System

Figure 5-11



Via 

you can select "German" as the display language.

Via 

you can select "English" as the display language.

Via 

you end the HMI runtime.

Time setting/CPU

The application example has a time synchronization between CPU and HMI.

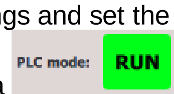
Via 

you can edit the date and via 

the time.

Via 

you accept these settings and set the CPU system time.

The current CPU operating state is displayed via 

.

Via 

you put the CPU into the operating state "RUN".

Via 

you put the CPU into the operating state "STOP".

In the CPU operating state 

the header and sidebar flash alternately orange:

6 Appendix

6.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

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

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

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:
https://support.industry.siemens.com

Technical Support

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

SITRAIN – Training for Industry

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

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:
www.siemens.com/sitrain

Service offer

Our range of services includes the following:

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

You can find detailed information on our range of services in the service catalog web page:
support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android and Windows Phone:
support.industry.siemens.com/cs/ww/en/sc/2067

6.2 Links and Literature

Table 6-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/100746401
\3\	SIMATIC S7-1200 programmable controller system manual https://support.industry.siemens.com/cs/ww/en/view/109764129
\4\	Single and multi-loop controller structures (cascade control) with PID_Temp https://support.industry.siemens.com/cs/ww/en/view/103526819
\5\	Multi-zone control with "PID_Temp" for SIMATIC S7-1200/S7-1500 https://support.industry.siemens.com/cs/ww/en/view/109740463
\6\	SIMATIC STEP 7 Basic/Professional V15.1 and SIMATIC WinCC V15.1 https://support.industry.siemens.com/cs/ww/en/view/109755202
\7\	Control of simulated controlled systems in the S7-1500 with PID_Compact V2 https://support.industry.siemens.com/cs/ww/en/view/79047707
\8\	Updates for STEP 7 V15.1 and WinCC V15.1 https://support.industry.siemens.com/cs/ww/en/view/109763890
\9\	SIMATIC S7-1500/ET 200MP Manual Collection https://support.industry.siemens.com/cs/ww/en/view/86140384
\10\	Three-step control with SIMATIC S7-1200/S7-1500 https://support.industry.siemens.com/cs/ww/en/view/62154322
\11\	TIA Selection Tool http://w3.siemens.com/mcms/topics/de/simatic/tia-selection-tool/Seiten/tab.aspx
\12\	Support Packages for the hardware catalog in the TIA Portal (HSP) https://support.industry.siemens.com/cs/ww/en/view/72341852
\13\	Function manual SIMATIC S7-1200, S7-1500 PID control https://support.industry.siemens.com/cs/ww/en/view/108210036

6.3 Change documentation

Table 6-2

Version	Date	Modifications
V1.0	11/2014	First version
V2.0	04/2019	Expanded entry on S7-1500 and TIA V15.1