Learn-/Training Document

Siemens Automation Cooperates with Education (SCE) | From Version V14 SP1

TIA Portal Module 051-300
PID Controller for SIMATIC S7-1200

siemens.com/sce

SIEMENS
Global Industry Partner of WorldSkills International

worldskills

Matching SCE Trainer Packages for these Learn-/Training Document

- **SIMATIC S7-1200 AC/DC/RELAY (set of 6) "TIA Portal"**
  Order no.: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC (set of 6) "TIA Portal"**
  Order no.: 6ES7214-1AE30-4AB3
- **Upgrade SIMATIC STEP 7 BASIC V14 SP1 (for S7-1200) (set of 6) "TIA Portal"**
  Order no.: 6ES7822-0AA04-4YE5

Please note that these trainer packages are replaced with successor packages when necessary.
An overview of the currently available SCE packages is provided at: siemens.com/sce/tp

Continued training

For regional Siemens SCE continued training, please contact your regional SCE contact siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Information regarding use

We wish to thank the TU Dresden, particularly Prof. Dr.-Ing. Leon Urbas and the Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this Learn-/Training Document.

Table of contents

PID Controller for the SIMATIC S7-1200

1 Goal

In this chapter, you will become acquainted with the use of software PID controllers for the SIMATIC S7-1200 with the TIA Portal programming tool.

The module explains the call-up, connection, configuration and optimization of a PID controller for the SIMATIC S7-1200. It also shows the steps for calling the PID controller in the TIA Portal and integrating it into a user program.

The SIMATIC S7 controllers listed in Chapter 3 can be used.

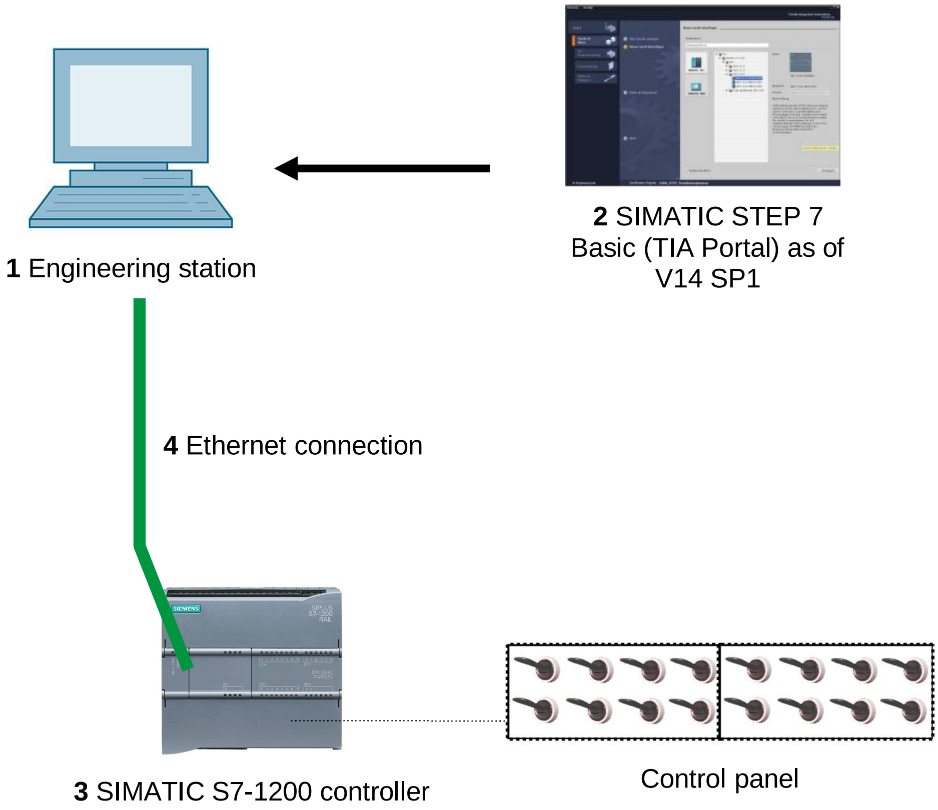
2 Prerequisite

This chapter builds on the chapter Analog Values with the SIMATIC S7 CPU1214C DC/DC/DC. You can use the following project for this chapter, for example: "SCE_EN_031-500_Analog_Values_S7-1200.zap14“.

3 Required hardware and software

1. Engineering station: requirements include hardware and operating system (for additional information, see Readme on the TIA Portal Installation DVDs)
2. SIMATIC STEP 7 Basic software in TIA Portal – as of V14 SP1
3. SIMATIC S7-1200 controller, e.g. CPU 1214C DC/DC/DC with ANALOG OUTPUT SB1232 signal board, 1 AO – Firmware as of V4.2.1

   Note: The digital inputs and analog inputs and outputs should be fed out to a control panel.
4. Ethernet connection between engineering station and controller



4 Theory of closed loop controls

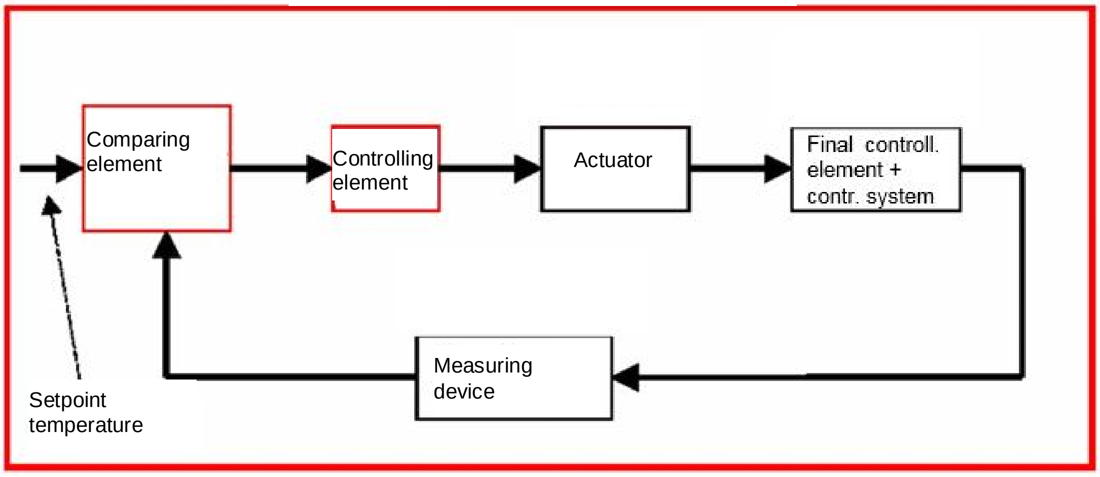
4.1 Tasks of closed loop controls

Closed loop control is a process in which the value of a variable is generated and maintained continuously through an intervention based on measurements of this variable.

This produces an action path that takes place in a closed loop – the control loop – because the process runs based on measurements of a variable that is, in turn, influenced by itself.

The variable to be controlled is continuously measured and compared with another preset variable of the same type. Depending on the result of this comparison, an adjustment of the variable to be controlled to the value of the preset variable is made.

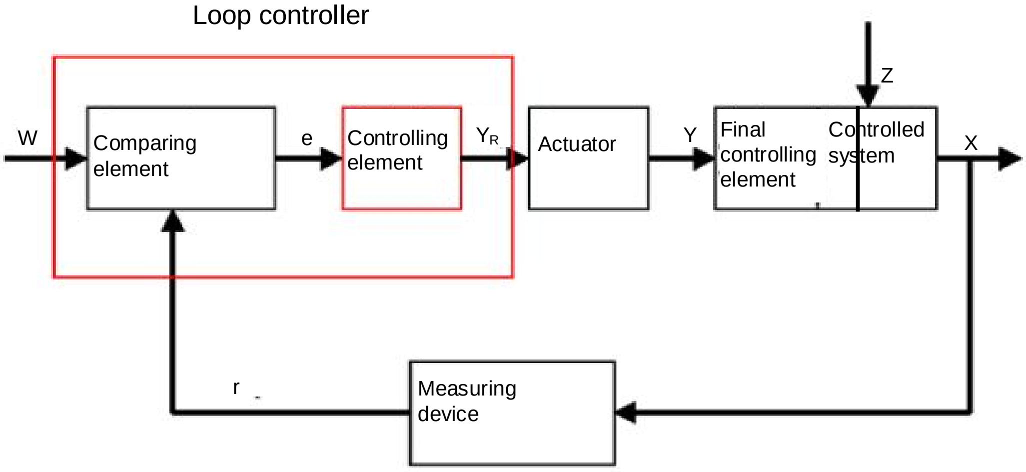
Diagrammatic representation of a closed loop control



4.2 Components of a control loop

The fundamental concepts of closed loop controls are explained in detail in the following.

An overview based on a diagram is presented here to start.



1. The controlled variable x

This is the actual "target" of the closed-loop control, namely the variable that is to be influenced or kept constant. In our example, this would be the room temperature. The instantaneous value of the controlled variable at a particular time is called the "actual value" at this time.

2. The feedback variable r

In a control loop, the controlled variable is continuously checked to enable a response to unwanted changes. The measured quantity proportional to the controlled variable is called the feedback variable. In the "Heating" example, it would correspond to the measured voltage of the inside thermometer.

3. The disturbance variable z

The disturbance variable is the variable that influences the controlled variable in an unwanted way and moves it away from the current setpoint. In the case of fixed setpoint control, this control is only necessary in the first place due to the existence of the disturbance variable. In the examined heating system, this would be, for example, the outside temperature or any other variable that causes the room temperature to move away from its ideal value.

4. The setpoint w

The setpoint at a given time is the value that the controlled variable should ideally have at this time. Note that the setpoint may vary continuously in a slave control. In our example, the setpoint would be the currently desired room temperature.

5. The comparing element

This is the point at which the current measured value of the controlled variable and the instantaneous value of the reference variable are compared. In most cases, both variables are measured voltages. The difference between the two variables is the "system error" e. This is passed to the controlling element and evaluated there (see below).

6. The controlling element

The controlling element is the actual heart of a closed loop control. It evaluates the system error, thus the information regarding whether, how and how much the controlled variable deviates from the current setpoint, as an input variable and derives from this the **"Controller output variable" $\mathbf{Y_R}$**, which is ultimately used to influence the controlled variable. In the heating system example, the controller output variable would be the voltage for the mixer motor.

The manner in which the controlling element determines the controller output variable from the system error is the main criterion of the closed-loop control.

7. The actuator

The actuator is, so to speak, the "executive organ" of the closed loop control. It receives information from the controlling element in the form of the controller output variable indicating how the controlled variable is to be influenced and translates this into a change of the "manipulated variable". In our example, this would be the mixer motor controller.

8. The final controlling element

This is the element of the control loop that influences the controlled variable (more or less directly) as a function of the **manipulated variable Y**. In the example, this would be the combination of the mixer, heating lines and radiators. The adjustment of the mixer (the manipulated variable) is made by the mixer motor (actuator) and influences the room temperature by means of the water temperature.

9. The controlled system

The controlled system is the system containing the variable to be controlled, thus the living space in the heating example.

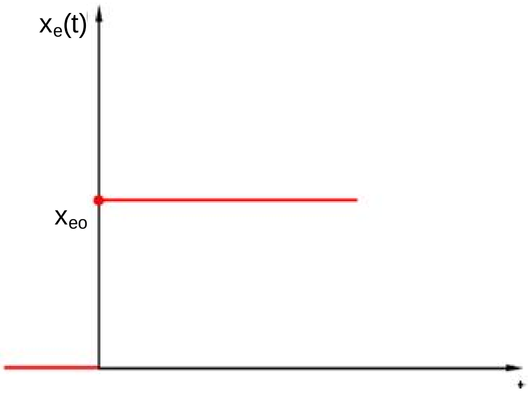
10. The dead time

The dead time refers to the time that elapses from a change in the controller output variable until there is a measurable response in the controlled system. In the example, this would be the time between a change in the voltage for the mixer motor and a measurable change in the room temperature resulting from this.

4.3 Step function for analysis of controlled systems

To analyze the response of controlled systems, controllers and control loops, a uniform function for the input signal is used – the step function.

Depending on whether a control loop element or the entire control loop is being analyzed, the controlled variable x(t), the manipulated variable y(t), the reference variable w(t) or the disturbance variable z(t) can be assigned the step function. The input signal is often designated xe(t) and the output signal xa(t).

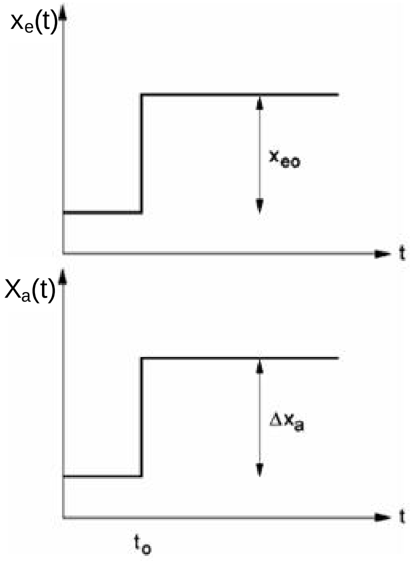


$$x_e(t) = \begin{cases} 0 & \text{for } t < 0 \\ x_{eo} & \text{for } t \geq 0 \end{cases}$$

4.4 Controlled systems with self-regulation

4.4.1 Proportional system without time delay

This controlled system is called a P system for short.



sudden change of the input variable at i t_0

Controlled variable/manipulated variable:

$$x = K_{ss} \bullet y$$

$$K_{ss} = \frac{\Delta x}{\Delta y} = \tan\alpha$$

Kss: Proportional coefficient for a manipulated variable change:

Controlled variable/disturbance variable:

$$x = K_{sz} \bullet z$$

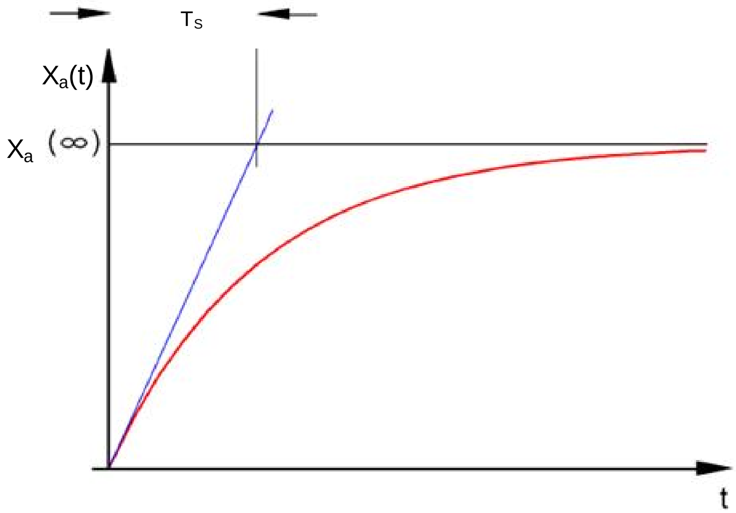
Ksz: Proportional value for a disturbance variable change

Range: $y_h = y_{max} - y_{min}$

Control range: $x_h = x_{max} - x_{min}$

4.4.2 Proportional system with time delay

This controlled system is called a P-T1 system for short.



Differential equation for a general input signal $x_e(t)$:

$T_S \bullet x_a(t) + x_a(t) = K_{PS} \bullet x_e(t)$

Solution of the differential equation for a step function at the input (step response)

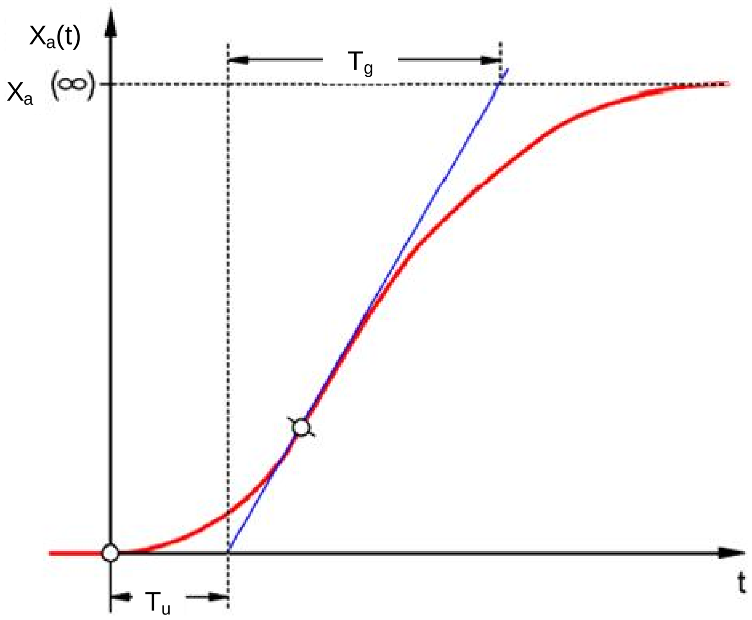
$x_a(t) = K_{PS}\ (1\text{-}e^{-t/TS}) \bullet x_{eo}$

$x_a\ (t = \infty) = K_{PS} \bullet x_{eo}$

T_S: Time constant

4.4.3 Proportional system with two time delays

This system is called a P-T2 system for short.



Tu: Delay time Tg: Compensation time

The system is generated through the reaction-free series connection of two P-T1 systems that have the time constants TS1 and TS2.

Controllability of P-Tn systems:

$$\frac{T_u}{T_g} < \frac{1}{10} \rightarrow \boxed{\text{easily controllable}} \qquad \frac{T_u}{T_g} \approx \frac{1}{6} \rightarrow \boxed{\text{still controllable}} \qquad \frac{T_u}{T_g} > \frac{1}{3} \rightarrow \boxed{\text{difficult to control}}$$

With the increasing ratio Tu/Tg, the system becomes less and less controllable.

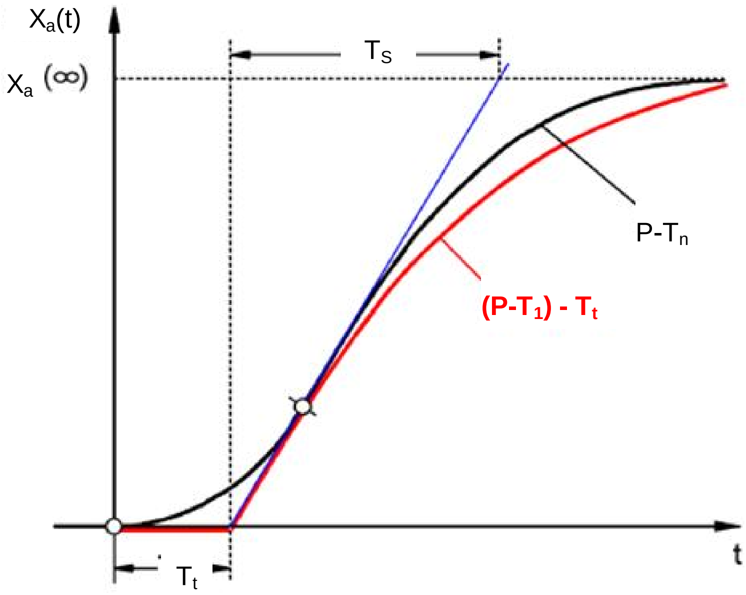
4.4.4 Proportional system with n time delays

This controlled system is called a P-Tn system for short.

The time response is described by an nth order differential equation. The step response characteristic is similar to that of the P-T2 system. The time response is described by Tu and Tg.

Substitute: An approximate substitution for the system with many delays is the series connection of a P-T1 system with a dead time system.

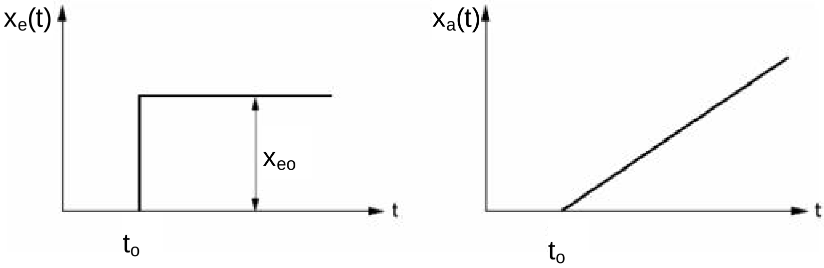
The following applies: Tt » Tu and TS » Tg.



4.5 Systems without self-regulation

This controlled system is called an I system for short.

After a disturbance, the controlled variable continues increasing steadily without striving for a fixed final value.

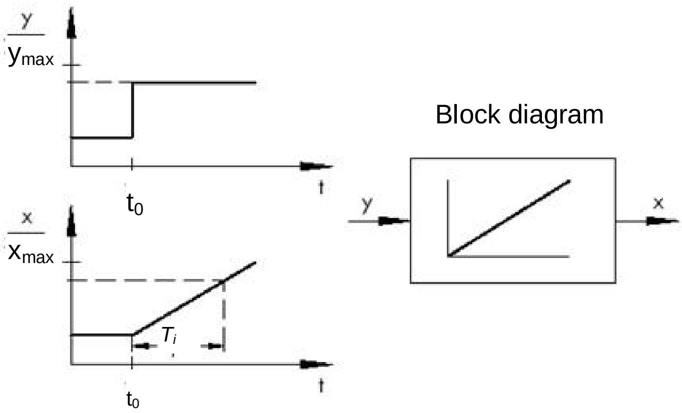


Example: Level control

For a tank with discharge outlet, whose incoming and outgoing flow rates are the same, there is a constant fill height. If the incoming or outgoing flow rate changes, the liquid level rises or falls. The level changes faster as the difference between the incoming flow rate and outgoing flow rate increases.

It is clear from this example that, in practice, the integral action has a limit in most cases. The controlled variable increases or decreases only until a system-inherent limit value is reached. A tank runs over or drains dry, pressure reaches the system maximum or minimum, etc.

The figure shows the time response of an I system to a step change in the input variable as well as the derived block diagram:



If the step function at the input changes to a function xe(t), then

$x_a(t)=K_{IS}\int x_e(t)\,dt$ ⇒ integrating controlled system

K_{IS}: Integral coefficient of the controlled system

* Figure from SAMSON Technical Information - L102 Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

4.6 Basic types of continuous controllers

Discrete controllers that only switch one or two manipulated variables on and off have the advantage of simplicity. Both the controller itself and the actuator and final controlling element are simpler in nature and thus less expensive than continuous controllers.

Discrete controllers have several disadvantages, however. For one thing, when large loads such as large electric motors or cooling units must be switched, high load peaks may occur at switch-on and overload the power supply, for example. For this reason, these often do not switch between "Off" and "On" but instead between full power ("full load") and a significantly lower power of the actuator or final controlling element ("base load"). Still, even with this improvement, a discrete closed-loop control is unsuitable for numerous applications. Consider an automobile engine whose speed is discreetly controlled. There would then be nothing between idle and full throttle. Apart from the fact that it would probably be impossible to properly transfer the forces from a sudden full-throttle to the road via the tires, such a vehicle would probably be unsuitable for road traffic.

Continuous controllers are therefore used for such applications. Theoretically, hardly any limits are placed on the mathematical relationship that establishes the controlling element between the system error and controller output variable. In practice, however, three classic basic types are differentiated. These will be described in more detail in the following.

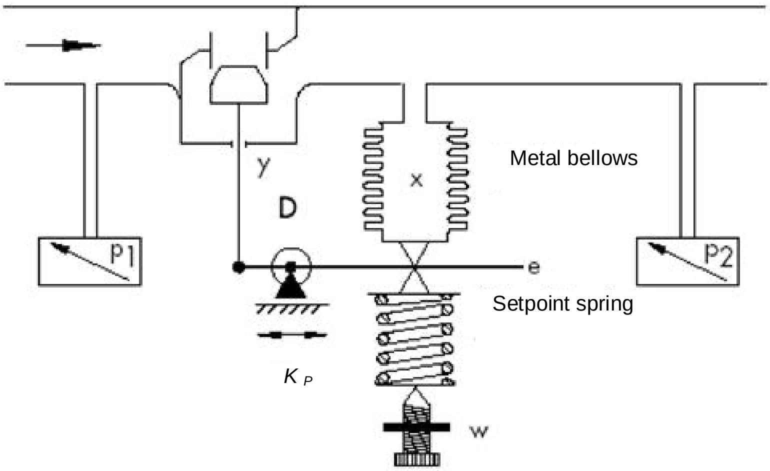
4.6.1 The proportional controller (P controller)

The manipulated variable y of a P controller is proportional to the measured error e. From this can be deducted that a P controller reacts to any deviation without lag and only generates a manipulated variable in case of system deviation.

The proportional pressure controller illustrated in the figure compares the force FS of the setpoint spring with the force FB created in the elastic metal bellows by the pressure p2. When the forces are off balance, the lever pivots about point D. This changes the position of the valve plug –and, hence, the pressure p2 to be controlled –until a new equilibrium of forces is restored.

The dynamic behavior of the P controller after a step change in the error variable is shown in the figure. The amplitude of the manipulated variable y is determined by the error e and the proportional-action coefficient Kp:

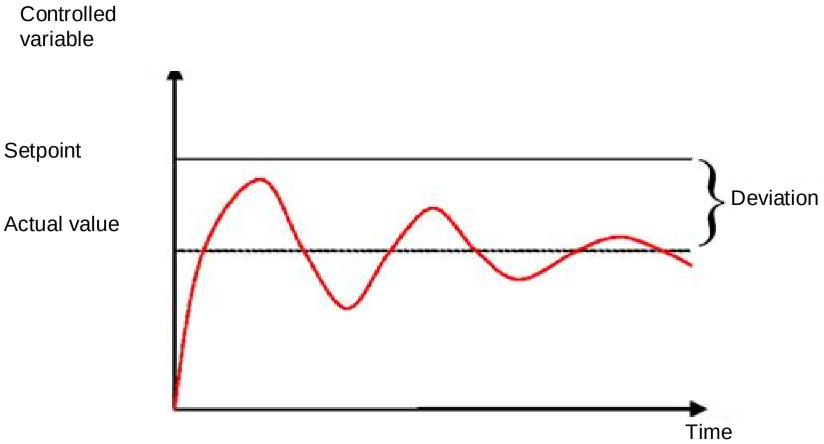
To keep the control deviation as small as possible, as large a proportional-action coefficient as possible must be selected. An increase in the factor causes the controller to react faster, but if the value is too high there is a risk of overshooting and a large "hunting" tendency of the controller.



$$y = K_{P} \cdot e$$

* Figure and text from SAMSON Technical Information - L102 Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

You see the response of the P controller in the diagram.



The advantages of this controller type lie, on the one hand, in its simplicity (in the simplest case, it can be implemented electronically with just a resistor) and, on the other hand, in its very prompt reaction compared to other controller types.

The main disadvantage of the P controller is its permanent system deviation. That is, the setpoint is never fully reached even over the long term. This disadvantage as well as the not yet ideal response speed cannot be minimized to a satisfactory extent through a larger proportional-action coefficient, because this leads to overshooting by the controller, or in other words an overreaction. In the worst case, the controller goes into a permanent oscillation in which the controlled variable is periodically moved away from the setpoint by the controller itself instead of by the manipulated variable.

The problem of permanent control deviation is best solved by an additional integral controller.

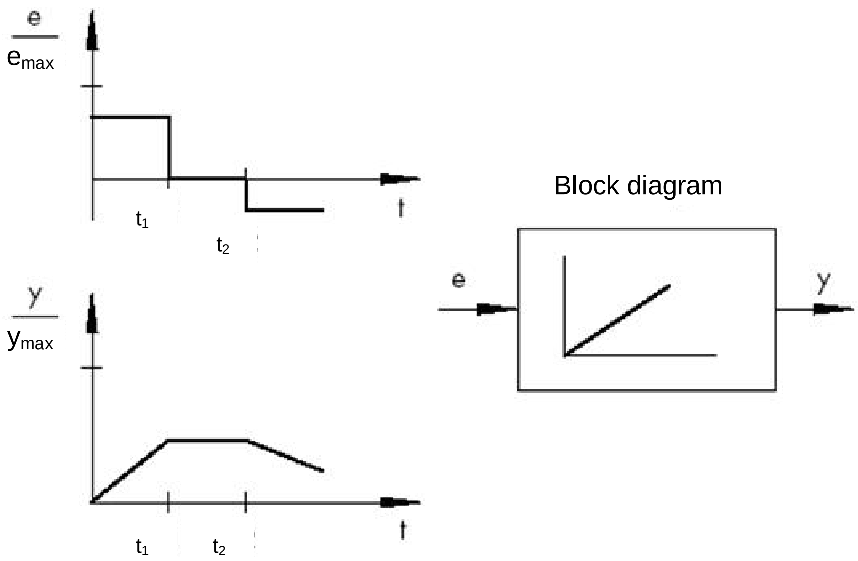
4.6.2 The integral controller (I controller)

Integral control action is used to fully correct system deviations at any operating point. As long as the error is nonzero, the integral action will cause the value of the manipulated variable to change. Only when reference variable and controlled variable are equally large –at the latest, though, when the manipulated variable reaches its system specific limit value (Umax, pmax, etc.)– is the control process balanced.

Mathematics expresses integral action as follows: the value of the manipulated variable is changed proportional to the integral of the error e.

$$y = K_i \int e \, dt \qquad \text{with} \qquad K_i = \frac{1}{T_n}$$

How rapidly the manipulated variable increases/decreases depends on the error and the integral time.

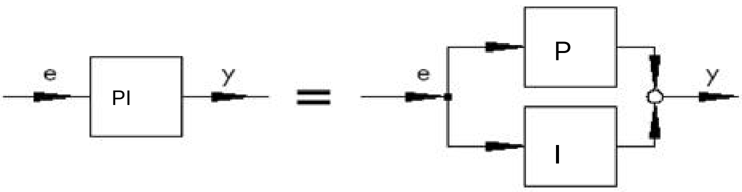


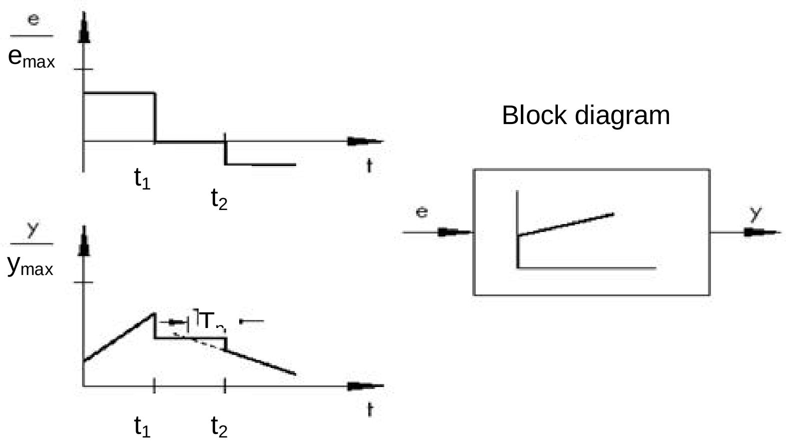
* Figure and text from SAMSON Technical Information - L102 Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

4.6.3 The PI controller

PI controllers are often employed in practice. In this combination, one P and one I controller are connected in parallel.

If properly designed, they combine the advantages of both controller types (stability and rapidity; no steady-state error), so that their disadvantages are compensated for at the same time.





The dynamic behavior is marked by the proportional-action coefficient Kp and the reset time Tn. Due to the proportional component, the manipulated variable immediately reacts to any error signal e, while the integral component starts gaining influence only after some time. Tn represents the time that elapses until the I component generates the same control amplitude that is generated by the P component (Kp) from the start. As with I controllers, the reset time Tn must be reduced if the integral-action component is to be amplified.

Controller dimensioning:

By adjusting the Kp and Tn values, oscillation of the controlled variable can be reduced, however, at the expense of control dynamics.

PI controller applications: Fast control loops allowing no steady-state error

Examples: pressure, temperature. ratio control, etc.

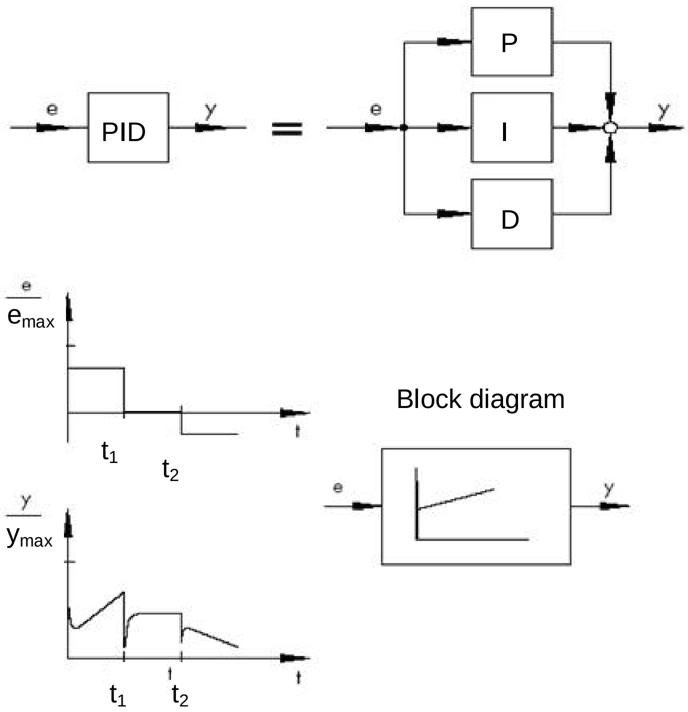
* Figure and text from SAMSON Technical Information - L102 Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

4.6.4 The derivative controller (D controller)

D controllers generate the manipulated variable from the rate of change of the error and not – – as P controllers –– from their amplitude. Therefore, they react much faster than P controllers: even if the error is small, derivative controllers generate– by anticipation, so to speak –large control amplitudes as soon as a change in amplitude occurs. A steady-state error signal, however, is not recognized by D controllers, because regardless of how big the error, its rate of change is zero. Therefore, derivative-only controllers are rarely used in practice. They are usually found in combination with other control elements, mostly in combination with proportional control.

4.6.5 The PID controller

If a D component is added to PI controllers, the result is an extremely versatile PID controller. As with PD controllers, the added D component –if properly tuned –causes the controlled variable to reach its setpoint more quickly, thus reaching steady state more rapidly.



$$y = K_p \cdot e + K_i \int e\,dt + K_D \frac{de}{dt} \quad \text{with} \quad K_i = \frac{K_P}{T_n};\ K_D = K_P \cdot T_V$$

* Figure and text from SAMSON Technical Information - L102 Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

4.7 Controller tuning using the oscillation test

For a satisfactory control result, the selection of a suitable controller is an important aspect. It is even more important that the control parameters Kp, Tn and TV be appropriately adjusted to the system response. Mostly, the adjustment of the controller parameters remains a compromise between a very stable, but also very slow control loop and a very dynamic, but irregular control response which may easily result in oscillation, making the control loop instable in the end.

For nonlinear systems that should always work in the same operating point, e.g. fixed setpoint control, the controller parameters must be adapted to the system response at this particular operating point. If a fixed operating point cannot be defined, such as with follow-up control systems ñ, the controller must be adjusted to ensure a sufficiently rapid and stable control result within the entire operating range.

In practice, controllers are usually tuned on the basis of values gained by experience.

Should these not be available, however, the system response must be analyzed in detail, followed by the application of several theoretical or practical tuning approaches in order to determine the proper control parameters.

One approach is a method first proposed by Ziegler and Nichols, the so-called ultimate method. It provides simple tuning that can be applied in many cases. This method, however, can only be applied to controlled systems that allow sustained oscillation of the controlled variable.

For this method, proceed as follows:

– At the controller, set Kp and Tv to the lowest value and Tn to the highest value (smallest possible influence of the controller).

– Adjust the controlled system manually to the desired operating point (start up control loop).

– Set the manipulated variable of the controller to the manually adjusted value and switch to automatic operating mode.

– Continue to increase Kp (decrease Xp) until the controlled variable encounters harmonic oscillation. If possible, small step changes in the setpoint should be made during the Kp adjustment to cause the control loop to oscillate.

– Take down the adjusted Kp value as critical proportional-action coefficient Kp,crit. Determine the time span for one full oscillation amplitude as Tcrit, if necessary by taking the time of several oscillations and calculating their average.

– Multiply the values of Kp,crit and Tcrit by the values according to the table and enter the determined values for Kp, Tn and Tv at the controller.

	K_p	T_n	T_v
P	0.50 x $K_{p.\ crit.}$	-	-
PI	0.45 x $K_{p.\ crit.}$	0.85 x $T_{crit.}$	-
PID	0.59 x $K_{p.\ crit.}$	0.50 x $T_{crit.}$	0.12 x $T_{crit.}$

* Figure and text from SAMSON Technical Information - L102 Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

4.8 Controller tuning with T_u-T_g approximation

The tuning of the controlled systems will be performed here using the example of a P-T2 system.

T_u-T_g approximation

The Ziegler-Nichols method and the Chien, Hrones and Reswick method are based on the T_u-T_g approximation in which the transfer coefficient of the system K_s, delay time T_u and balancing time T_g parameters are determined from the system step response.

The tuning rules, which are described below, are the result of experiments using analog computer simulations.

P-T_N systems can be described with sufficient accuracy with a so-called T_u-T_g approximation, that is, through approximation using a P-T_1-T_L system.

The starting point is the system step response with input step height K. The required parameters (transfer coefficient of the system K_s, delay time T_u and balancing time T_g) are determined as shown in the figure.

The transfer function must be measured up to the final steady-stated value (K*Ks) so that the transfer coefficient of the system Ks required for the calculation can be determined.

The main advantage of this method is that the approximation can also be used when an analytical description of the system is not possible.

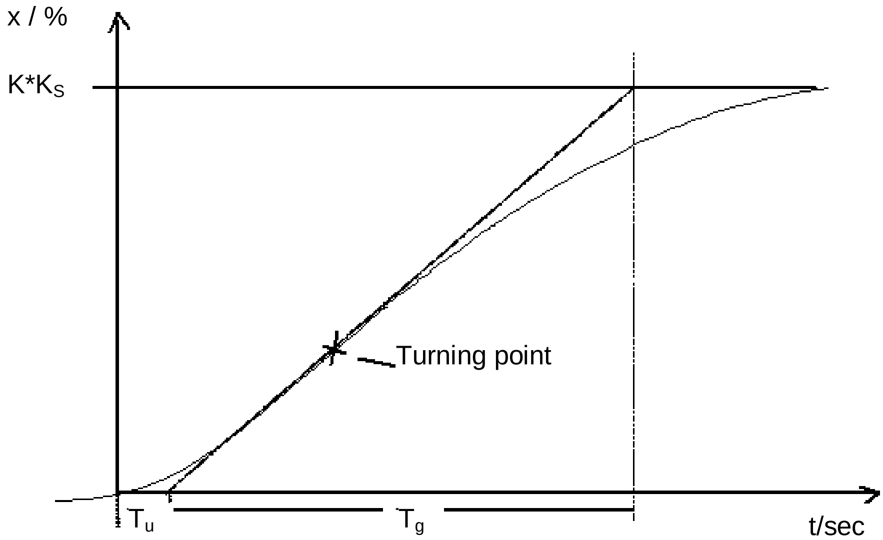


Figure: **T_u-T_g-Approximation**

4.8.1 Tuning the PI controller according to the Ziegler-Nichols method

Based on experiments on P-T_1-T_L systems, Ziegler and Nichols have identified the following optimal controller adjustments for fixed setpoint control:

$$K_{PR} = 0.9 \frac{T_g}{K_S T_u}$$

$$T_N = 3.33\, T_u$$

Use of these tuning values generally results in very good response to disturbances.

4.8.2 Tuning the PI controller according to the Chien, Hrones and Reswick method

Both the response to disturbances and response to setpoint changes were examined in order to achieve the most favorable controller parameters. Different values are yielded for the two cases. In addition, two different adjustments are specified in each case that meet different control performance requirements.

This resulted in the following adjustments:

· For response to disturbances:

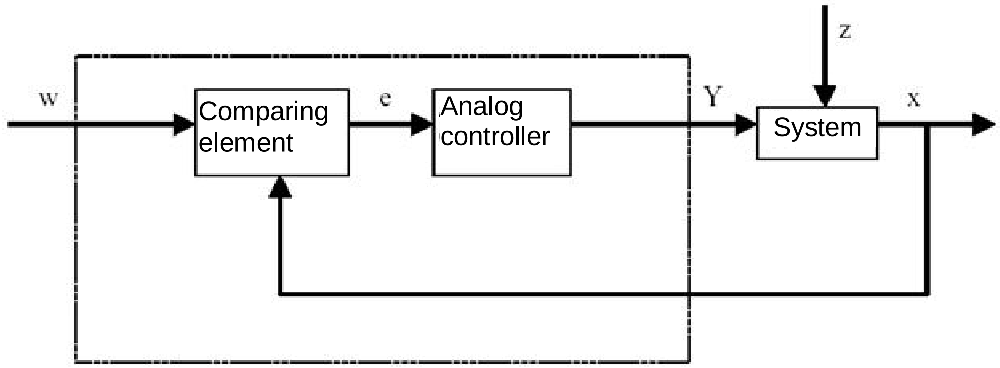
Aperiodic transient reaction with the shortest duration	20 % overshoot minimum oscillation period
$K_{PR} = 0.6 \frac{T_g}{K_S T_u}$	$K_{PR} = 0.7 \frac{T_g}{K_S T_u}$
$T_N = 4\, T_u$	$T_N = 2.3\, T_u$

· For response to setpoint changes:

Aperiodic transient reaction with the shortest duration	20 % overshoot minimum oscillation period
$K_{PR} = 0.35 \frac{T_g}{K_S T_u}$	$K_{PR} = 0.6 \frac{T_g}{K_S T_u}$
$T_N = 1.2\, T_g$	$T_N = T_g$

4.9 Digital controllers

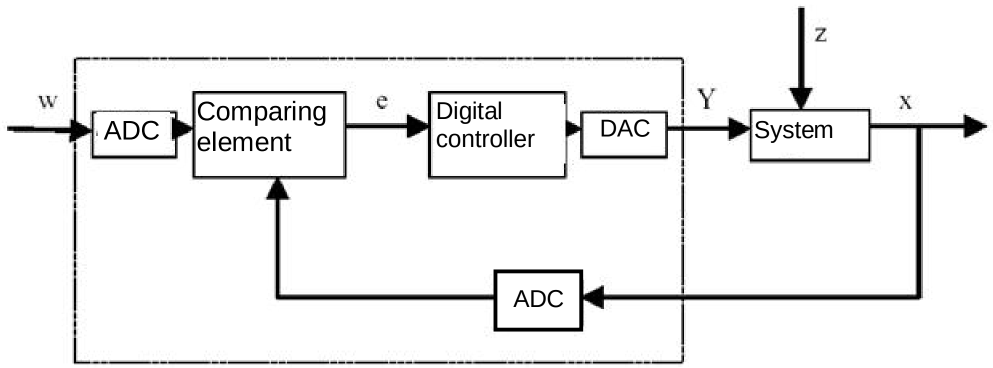
Up to now, the main focus was on analog controllers, in other words, controllers that use the system error, which exists as an analog value, to derive the controller output variable in an analog manner. The diagram of this type of control loop is now well-known:



Often, however, it is advantageous to perform the actual evaluation of the system error digitally. For one thing, the relationship between the system error and controller output variable can be defined much more flexibly when it can be defined by an algorithm or formula that can be used in each case to program a computer than when it has to be implemented in the form of an analog circuit. For another, digital technology enables significantly greater integration of circuits so that multiple controllers can be accommodated in the smallest space. Finally, by dividing the computing time when there is a sufficient amount of computing capacity, it is even possible to use an individual computer as a controller for multiple control loops.

To enable digital processing of the variables, both the reference variable and the feedback variable are first converted to digital values in an analog-to-digital converter (ADC). These are then subtracted from one another by a digital comparing element and the difference is passed to the digital controlling element. Its controller output variable is then converted back to an analog value in a digital-to-analog converter (DAC). From the outside, the combined unit of converters, comparing element and controlling element resembles an analog controller.

We will examine the structure of a digital controller based on a diagram:



The advantages resulting from digital implementation of the controller are accompanied by various problems. For this reason, the size of some variables related to the digital controller must be chosen large enough to prevent the accuracy of the closed loop control from suffering too much from digitization.

Quality criteria for digital computers are:

– The quantization resolution of the digital-to-analog converter

  This specifies how fine the continuous value range is digitally mapped. The chosen resolution must be high enough that none of the finer points important for the closed loop control are lost.

– The sampling rate of the analog-to-digital converter.

  This is the frequency at which the analog values present at the converter are measured and digitized. This must be high enough that the controller can also still respond to step changes in the controlled variable in a timely manner.

– The cycle time

  Unlike an analog closed-loop controller, each digital computer works in clock cycles. The speed of the utilized computer must be high enough that a significant change of the controlled variable cannot occur during a single clock cycle (in which the output value is calculated and no input value is queried).

The performance of the digital controller must be high enough that its response is apparently as prompt and precise as an analog controller.

5 Task

In this chapter, a PID controller for speed control will be added to the program from chapter "SCE_EN_031-500 Analog Values_S7-1200". The call-up of the "MOTOR_SPEEDCONTROL" [FC10] function must be deleted for this.

6 Planning

The PID_Compact technology object is available in the TIA Portal for closed loop controls.

For closed-loop control of the motor speed, this technology object replaces the "MOTOR_SPEEDCONTROL" [FC10] block.

This will be carried out as an expansion of the "031-500_Analog_Values_S7-1200" project. This project must be retrieved from the archive beforehand.

The call-up of the "MOTOR_SPEEDCONTROL" [FC10] function must be deleted in the "Main" [OB1] organization block before the technology object can be called and connected in a cyclic interrupt OB.

The PID_Compact technology object must then be configured and commissioned.

6.1 PID_Compact closed-loop control block

The PID_Compact technology object provides a PID controller with integrated tuning for proportional-action final controlling elements.

The following operating modes are possible:

– Inactive

– Pretuning

– Fine tuning

– Automatic mode

– Manual mode

– Substitute output value with error monitoring

Here, the connection, parameter assignment and commissioning of this controller will be for automatic mode

During commissioning we will use the integrated tuning algorithms and record the control response of the controlled system.

The PID_Compact technology object is always called from a cyclic interrupt OB whose fixed set cycle time is 50 ms here.

The speed setpoint is set as a constant at the "Setpoint" input of the PID_Compact technology object in revolutions per minute (range: +/- 50 rpm). The data type is 32-bit floating-point number (Real).

The actual speed value -B8 (sensor actual value speed of the motor +/-10V corresponds to +/- 50 rpm) will be entered at the "Input_PER" input.

The output of the controller "Output_PER" will then be connected directly with signal -U1 (manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm).

The controller will only be active as long as output -Q3 (conveyor motor -M1 variable speed) is set. If this is not set, the controller will be deactivated by connection of the "Reset" input.

6.2 Technology diagram

Here you see the technology diagram for the task.

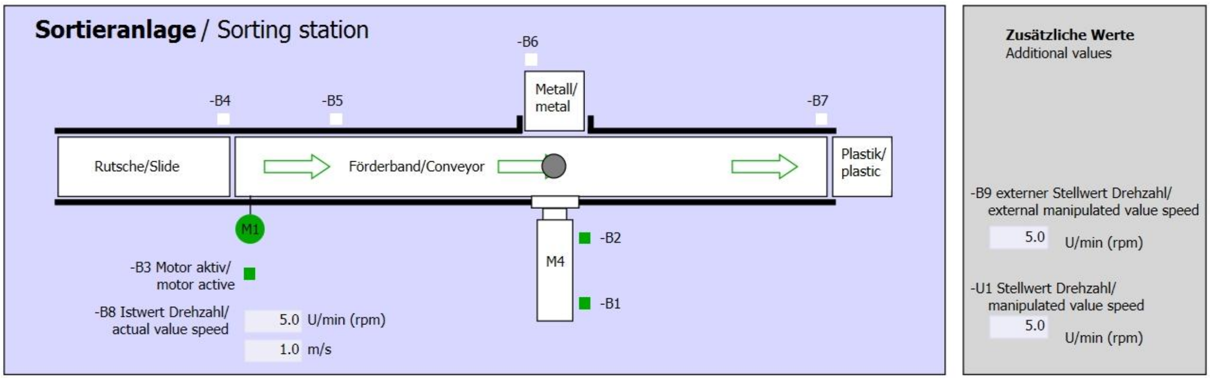


Figure 1: Technology diagram

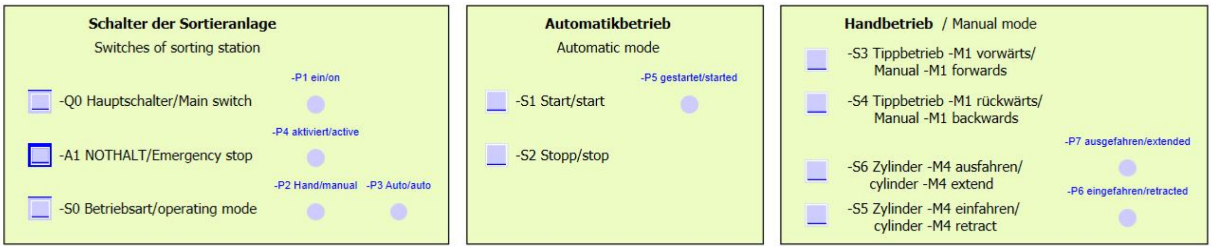


Figure 2: Control panel

6.3 Reference list

The following signals are required as global operands for this task.

DI	Type	Identifier	Function	NC/NO
I 0.0	BOOL	-A1	Return signal emergency stop OK	NC
I 0.1	BOOL	-K0	Main switch "ON"	NO
I 0.2	BOOL	-S0	Mode selector manual (0)/ automatic (1)	Manual = 0 Auto = 1
I 0.3	BOOL	-S1	Pushbutton automatic start	NO
I 0.4	BOOL	-S2	Pushbutton automatic stop	NC
I 0.5	BOOL	-B1	Sensor cylinder -M4 retracted	NO
I 1.0	BOOL	-B4	Sensor part at slide	NO
I 1.3	BOOL	-B7	Sensor part at end of conveyor	NO
IW64	BOOL	-B8	Sensor actual value speed of the motor +/-10V corresponds to +/- 50 rpm	

DO	Type	Identifier	Function	
Q 0.2	BOOL	-Q3	Conveyor motor -M1 variable speed	
QW 64	BOOL	-U1	Manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm	

Legend for reference list

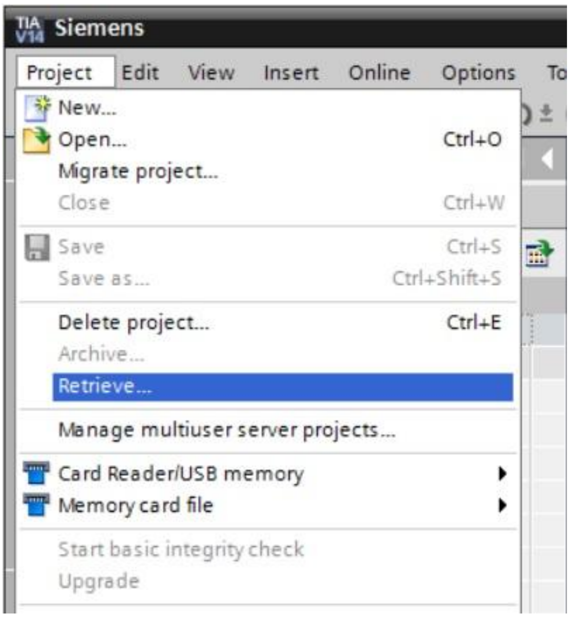
DI	Digital Input	DO	Digital Output
AI	Analog Input	AO	Analog Output
I	Input	Q	Output
NC	Normally Closed		
NO	Normally Open		

7 Structured step-by-step instructions

You can find instructions on how to carry out planning below. If you already have a good understanding of everything, it will be sufficient to focus on the numbered steps. Otherwise, simply follow the detailed steps in the instructions.

7.1 Retrieve an existing project

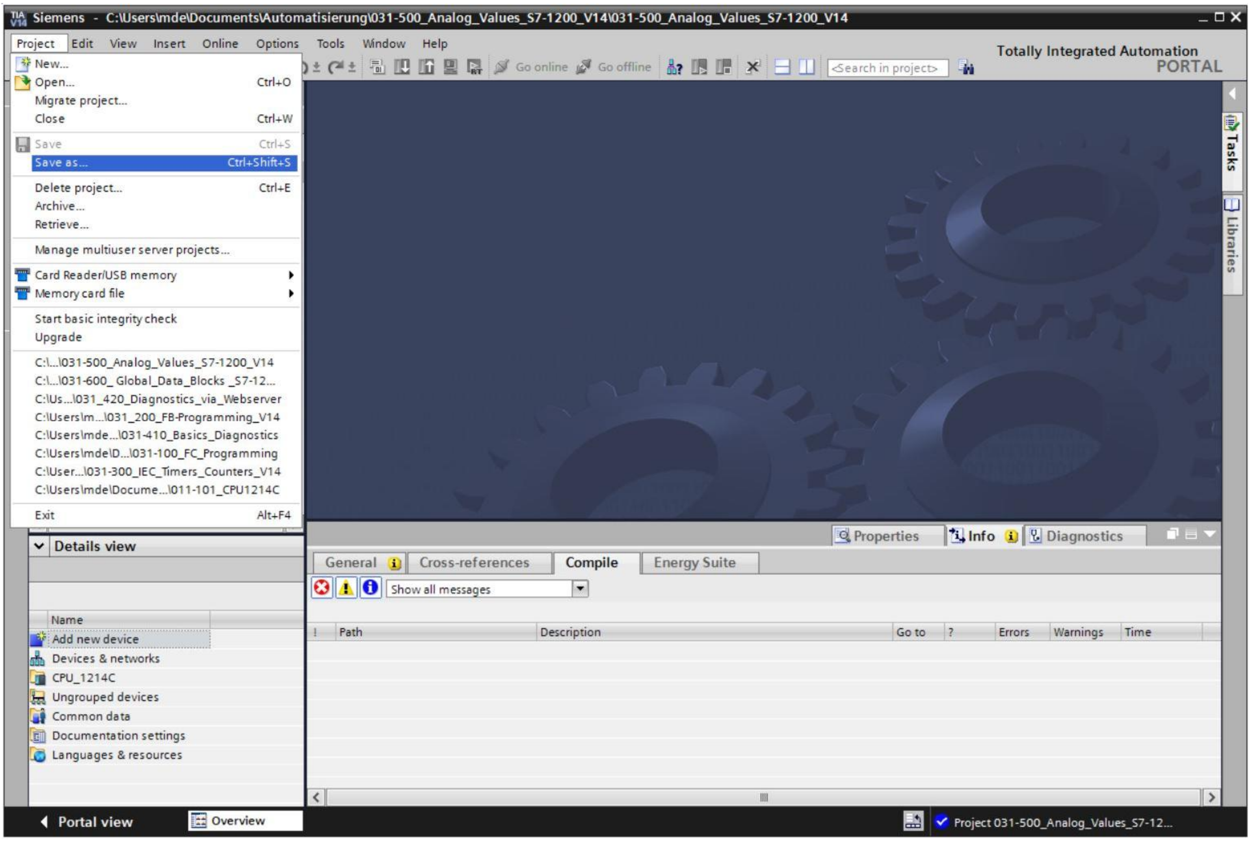
® Before we can expand the "SCE_EN_031-500_Analog_Values_S7-1200.zap14" project from chapter "SCE_EN_031-500 Analog Values_S7-1200", we must retrieve this project from the archive. To retrieve an existing project that has been archived, you must select the relevant archive with ® Project ® Retrieve in the project view. Confirm your selection with Open. (® Project ® Retrieve ® Select a .zap archive ® Open)



® The next step is to select the target directory where the retrieved project will be stored. Confirm your selection with "OK".

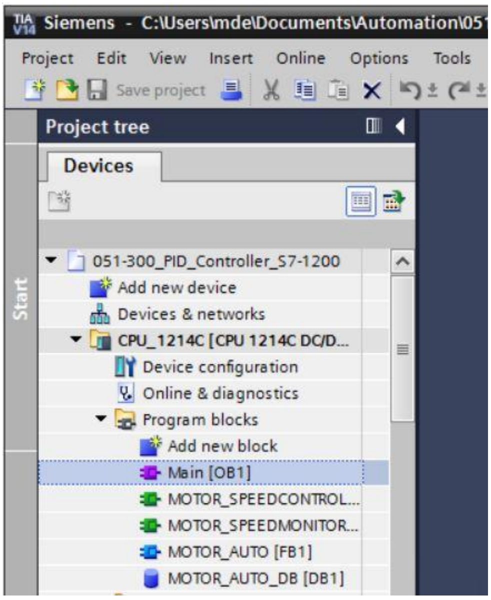
(® Target directory ® OK)

® Save the opened project under the name 051-300_PID_Controller_S7-1200. (® Project ® Save as … ® 051-300_PID_Controller_S7-1200 ® Save)



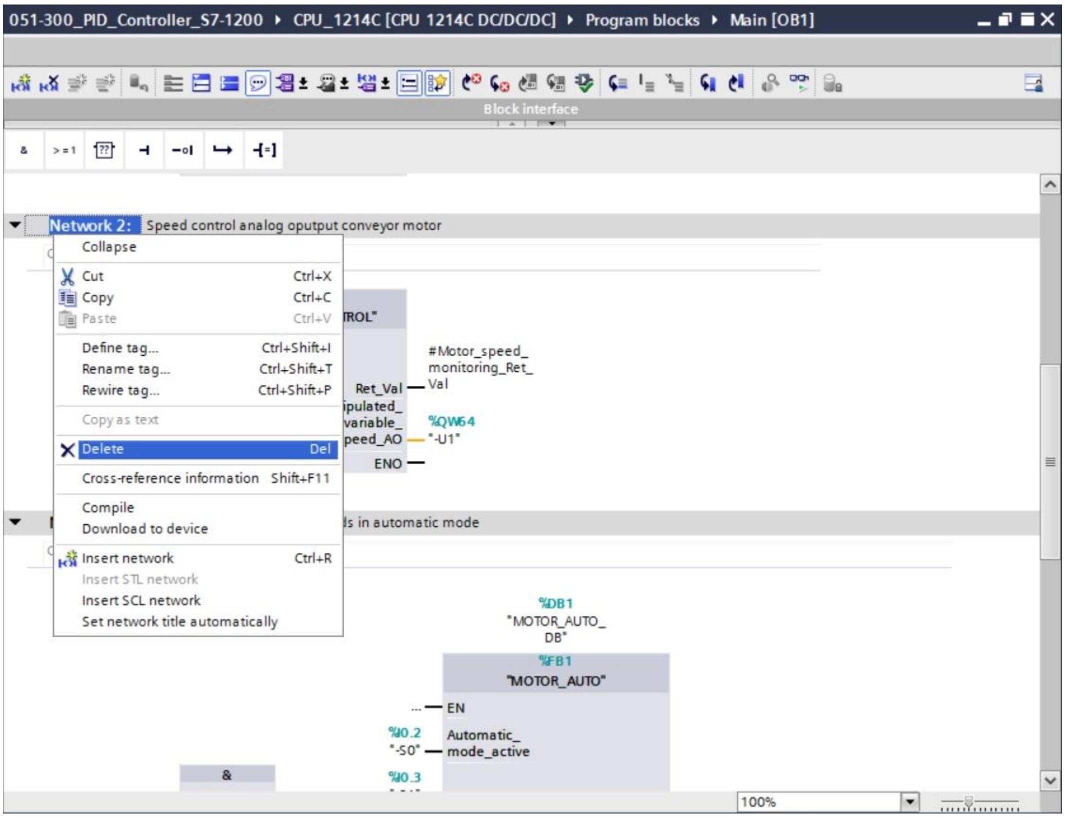
7.2 Call PID_Compact controller in a cyclic interrupt OB

® Open the “Main” [OB1] organization block with a double-click.



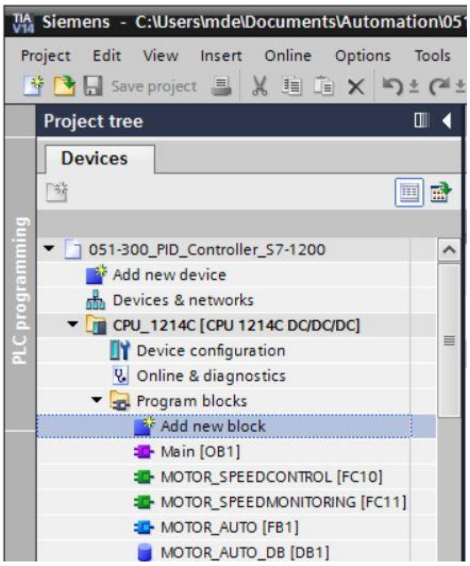
® Delete Network 2 with the no longer needed call-up of the "MOTOR_SPEEDCONTROL" [FC10] function.

(® Network 2 ® Delete)



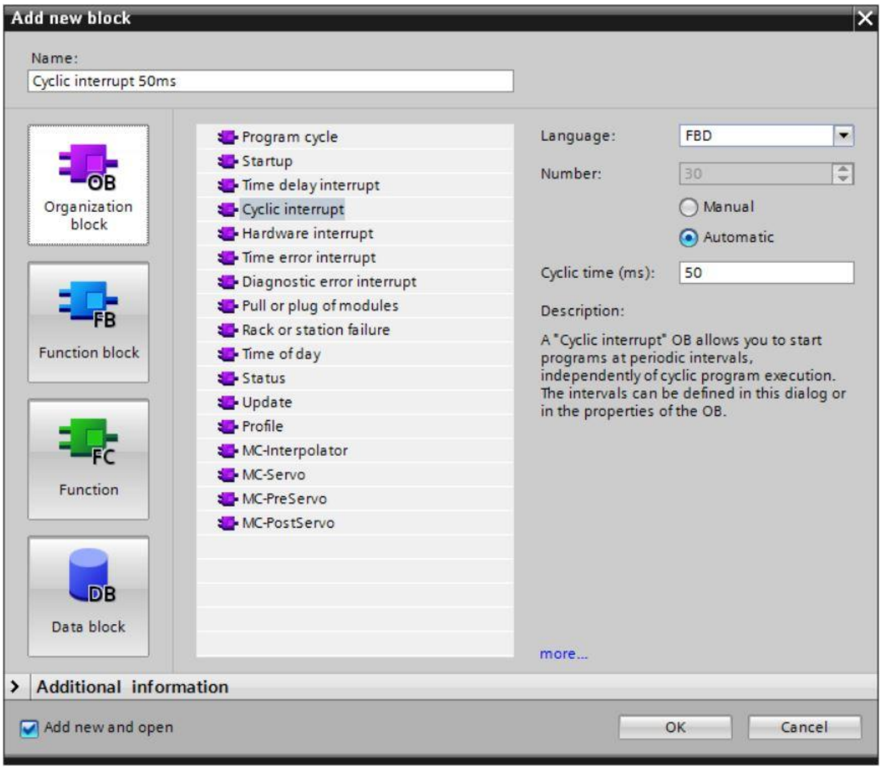
® We need a cyclic interrupt OB for calling the PID_Compact controller. Therefore, select the 'Add new block' item in the Program blocks folder.

(® Program blocks ® Add new block)



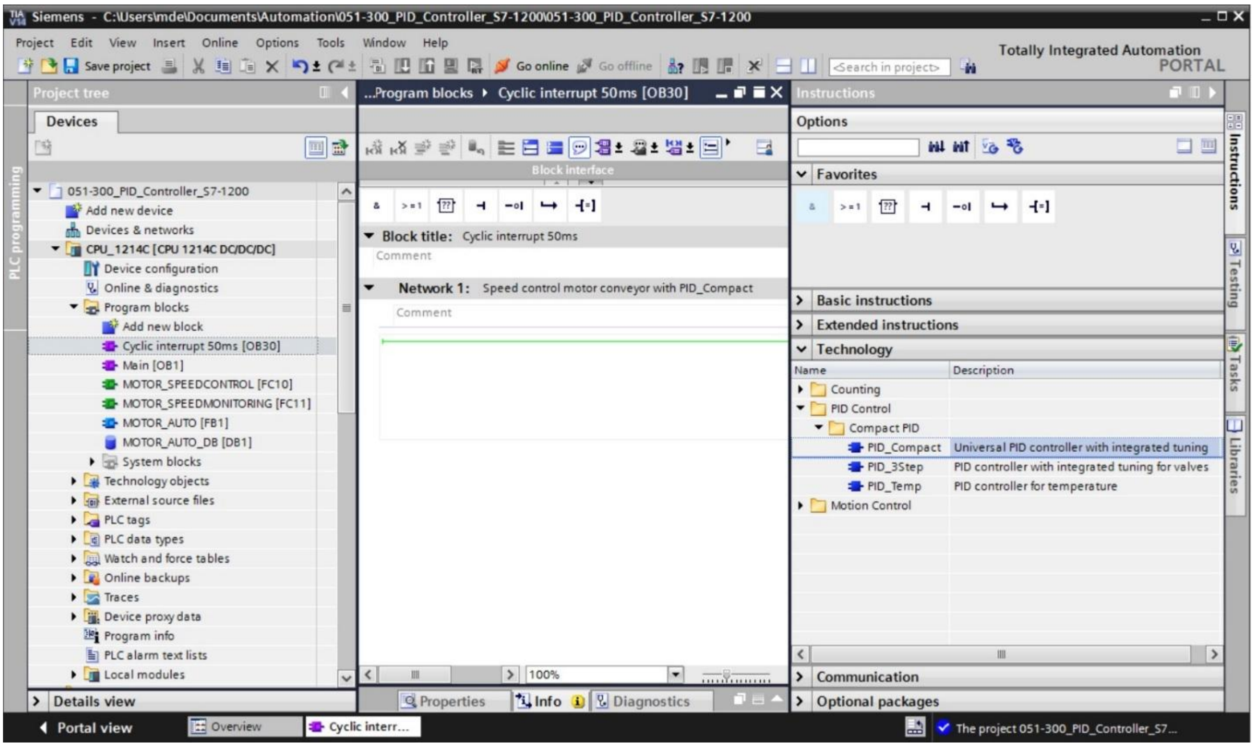
® Select OB in the next dialog and rename the cyclic interrupt OB to: "Cyclic interrupt 50ms". Set the language to FBD and assign "50 ms" as the cyclic time. Select the "Add new and open" check box. Click "OK".

(® OB ® Name: Cyclic interrupt 50ms ® Language: FBD ® Cyclic time (ms): 50 ® ☑ Add new and open ® OK)



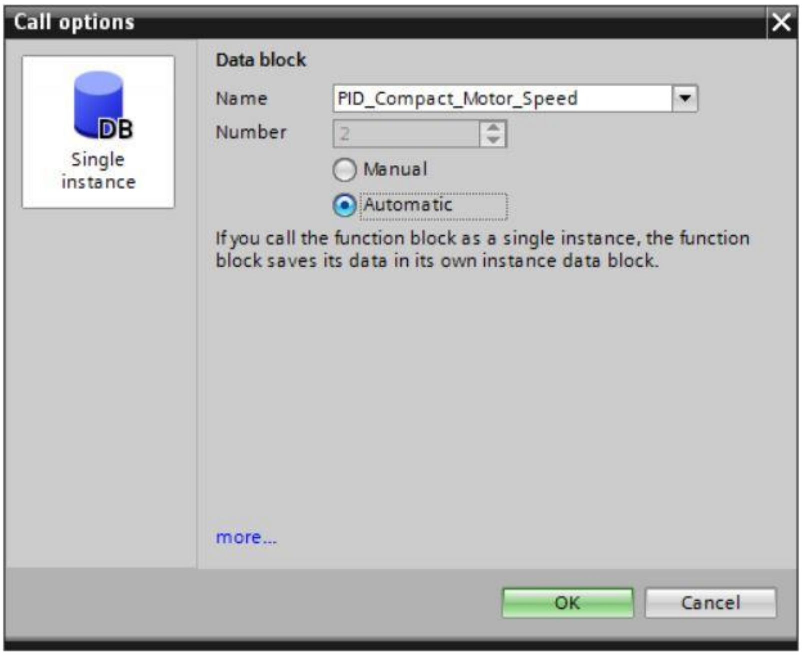
® The block is then directly opened. Enter meaningful comments and move the 'PID_Compact' technology object to Network 1 using drag & drop.

(® Technology ® PID Control ® Compact PID ® PID_Compact)

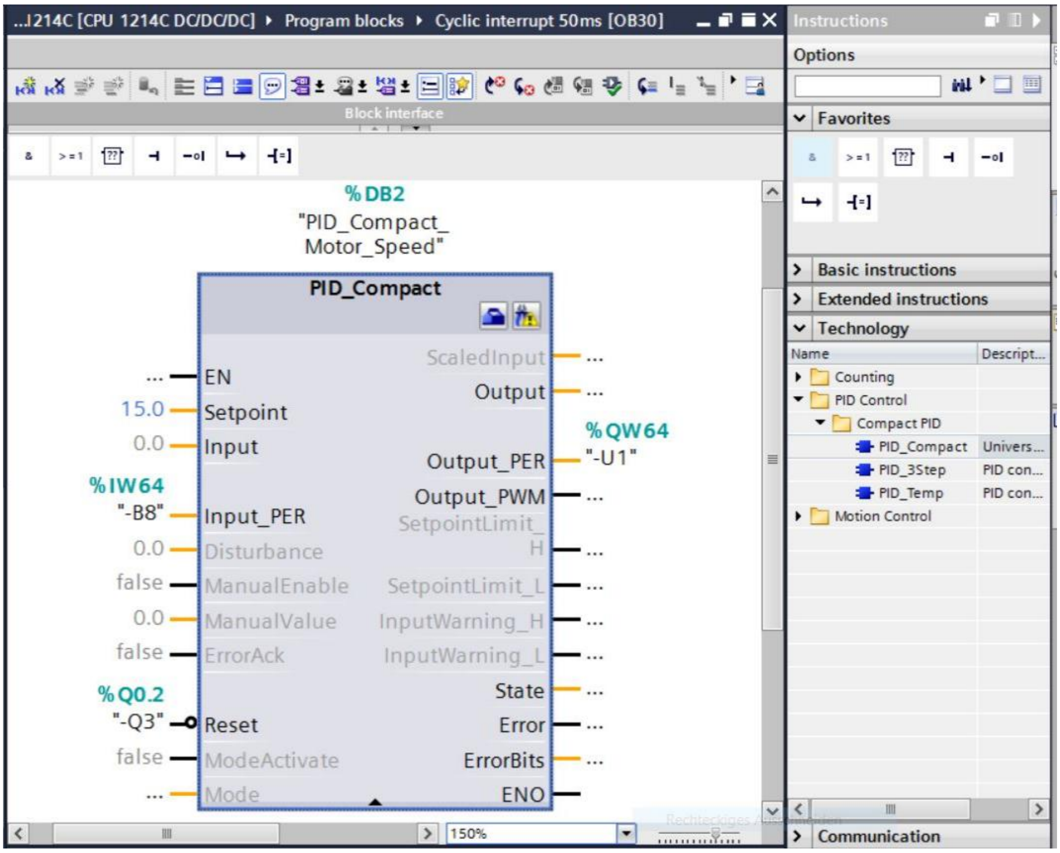


® Assign a name for the instance data block and apply it with OK.

(® PID_Compact_Motor_Speed ® OK)

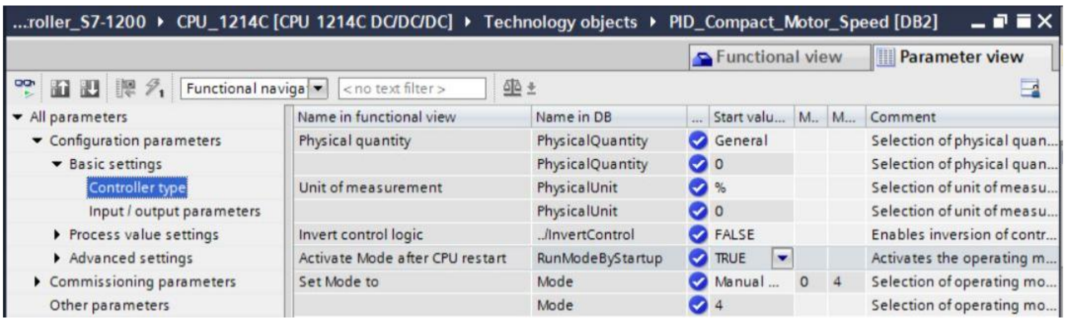


® Expand the view of the block by clicking the '▲' arrow. Interconnect this block as shown here with setpoint (constant: 15.0), actual value (global tag "-B8"), manipulated variable (global tag "-U1") and Reset input for deactivating the controller (global tag "-Q3"). Negate the 'Reset' input. The configuration mask '' of the controller can then be opened. (® ▲ ® 15.0 ® "-B8" ® "-U1" ® -Q3 ® −o| ®)



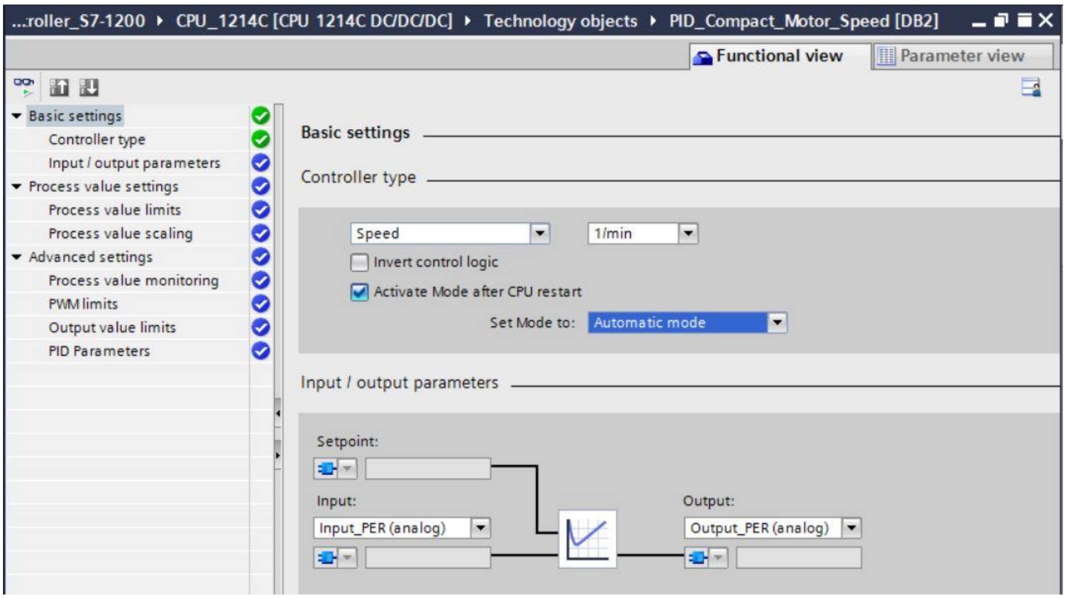
® There are 2 views for configuration of the controller: Parameter view and Functional view. Here we will use the easier-to-understand 'Functional view'.

(® Functional view)



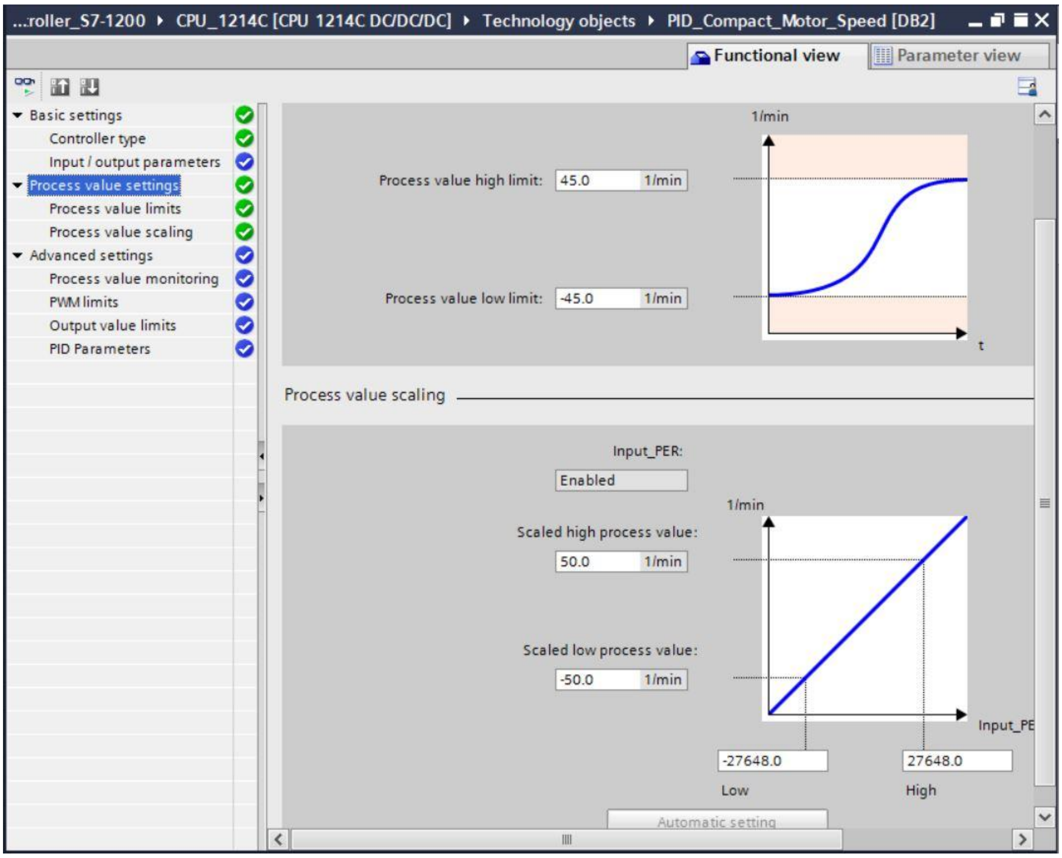
® In the 'Basic settings', the 'Controller type' and the interconnection of the 'Input / output parameters' are entered. Set the values as shown here.

(® Basic settings® Controller type ® Input / output parameters)



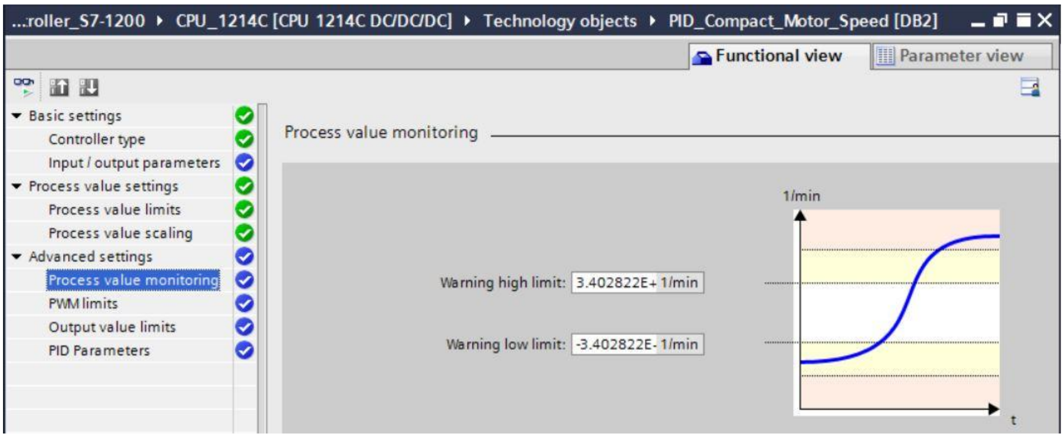
® In 'Process value settings' we scale to the range +/- 50 rpm and define the 'Process value limits' of +/- 45 rpm.

(® Process value settings ® Process value limits ® Process value scaling)



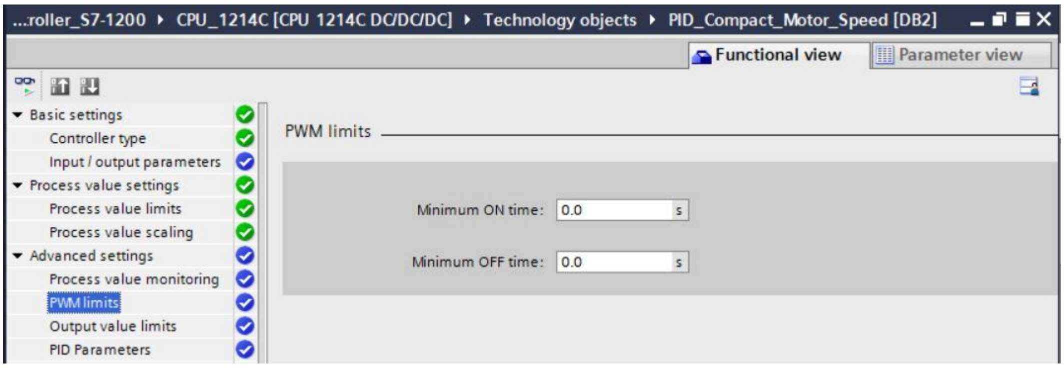
® In the 'Advanced settings', a process value monitoring would be possible but we don't want to deal with that here.

(® Advanced settings ® Process value monitoring)

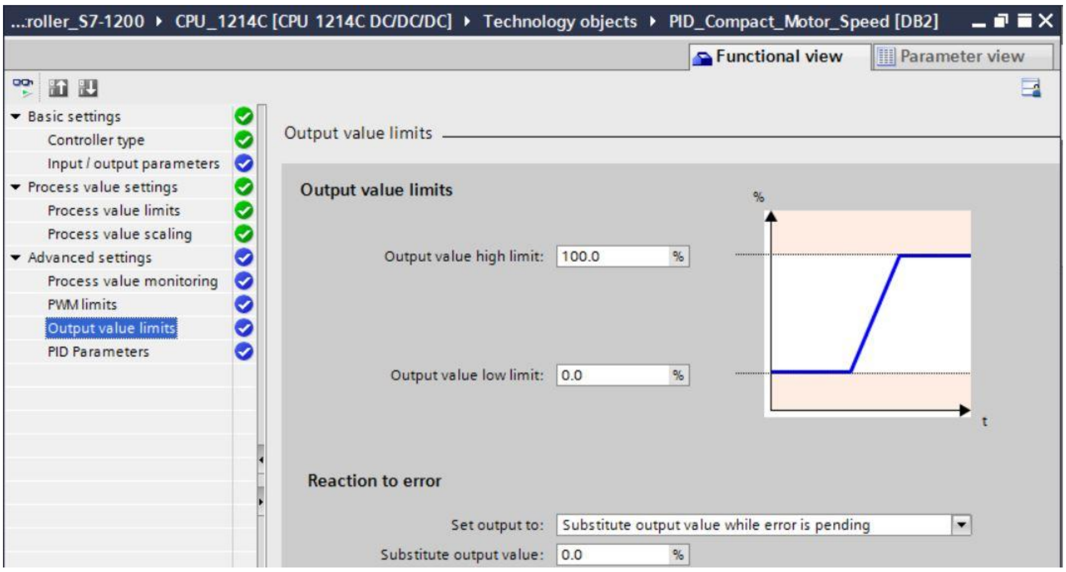


® In the 'Advanced settings' for 'PWM' (pulse width modulation), we will leave the default values since the output for this is not needed in our project.

(® Advanced settings ® PWM)

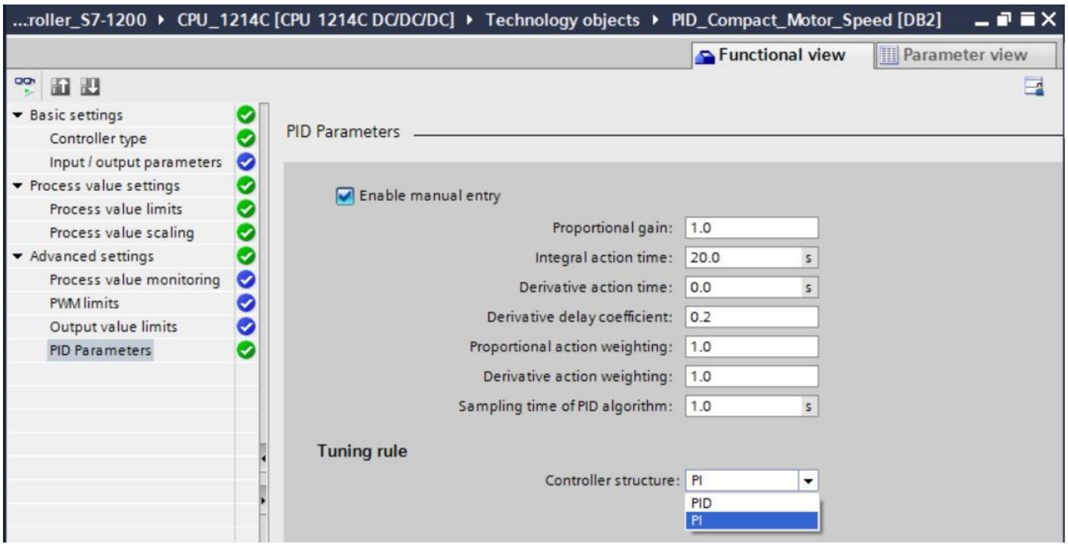


® In the 'Advanced settings', we define the 'Output value limits' of 0.0% to 100.0%. (® Advanced settings ® Output value limits)



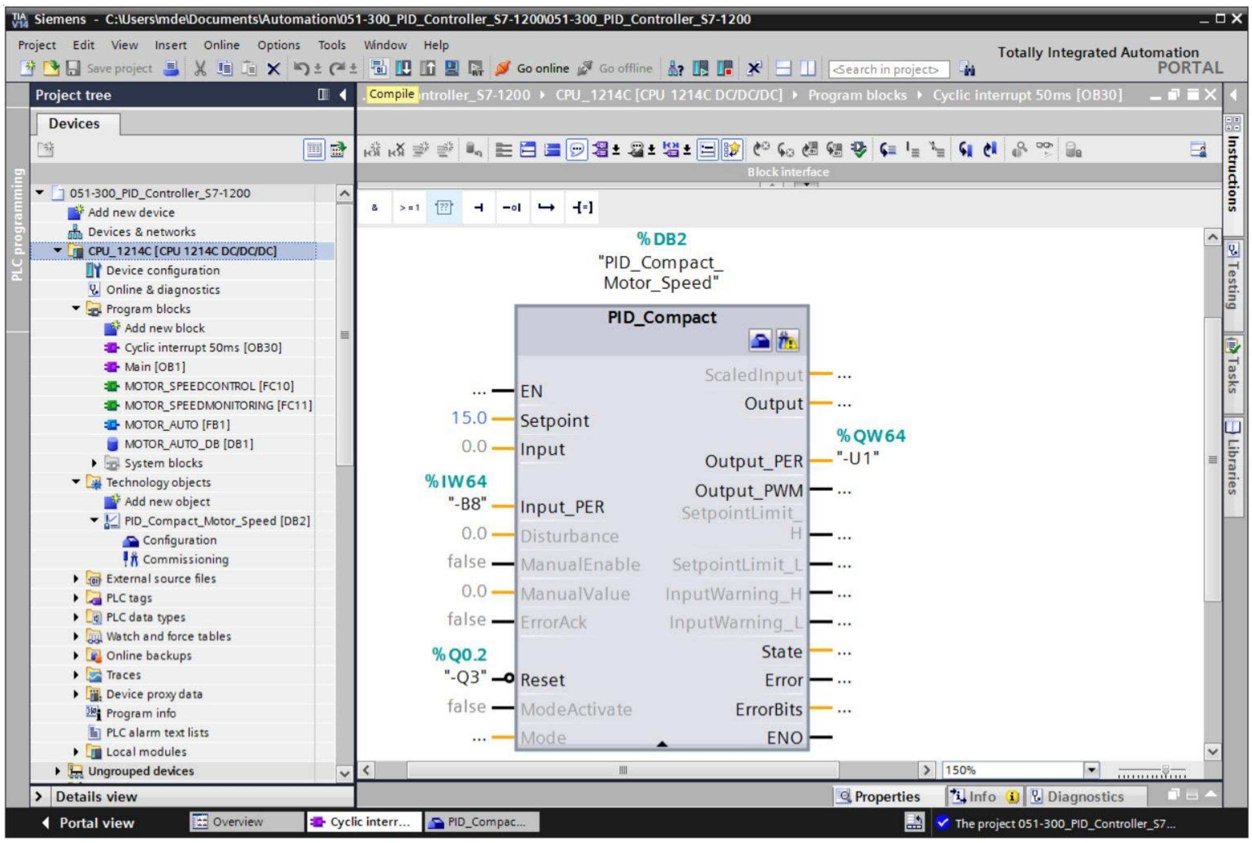
® In the 'Advanced settings', you will now also find a manual setting of the 'PID parameters'. Once we have changed the controller structure to 'PI', the configuration window is closed by clicking ✕ and we receive a finished product with a functional PID controller. This should, however, still be commissioned and tuned online during operation.

(® Advanced settings ® PID Parameters ® Controller structure: PI ® ✕)

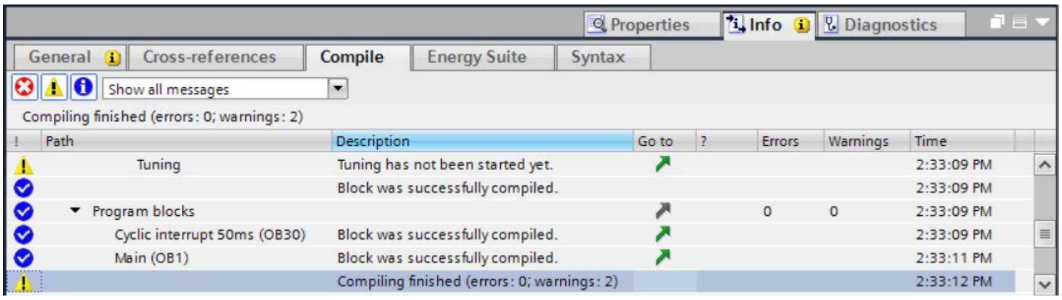


7.3 Save and compile the program

® To save your project, click the Save project button in the menu. To compile all blocks, click the "Program blocks" folder and select the  icon for compiling in the menu. (® Save project ® Program blocks ® )

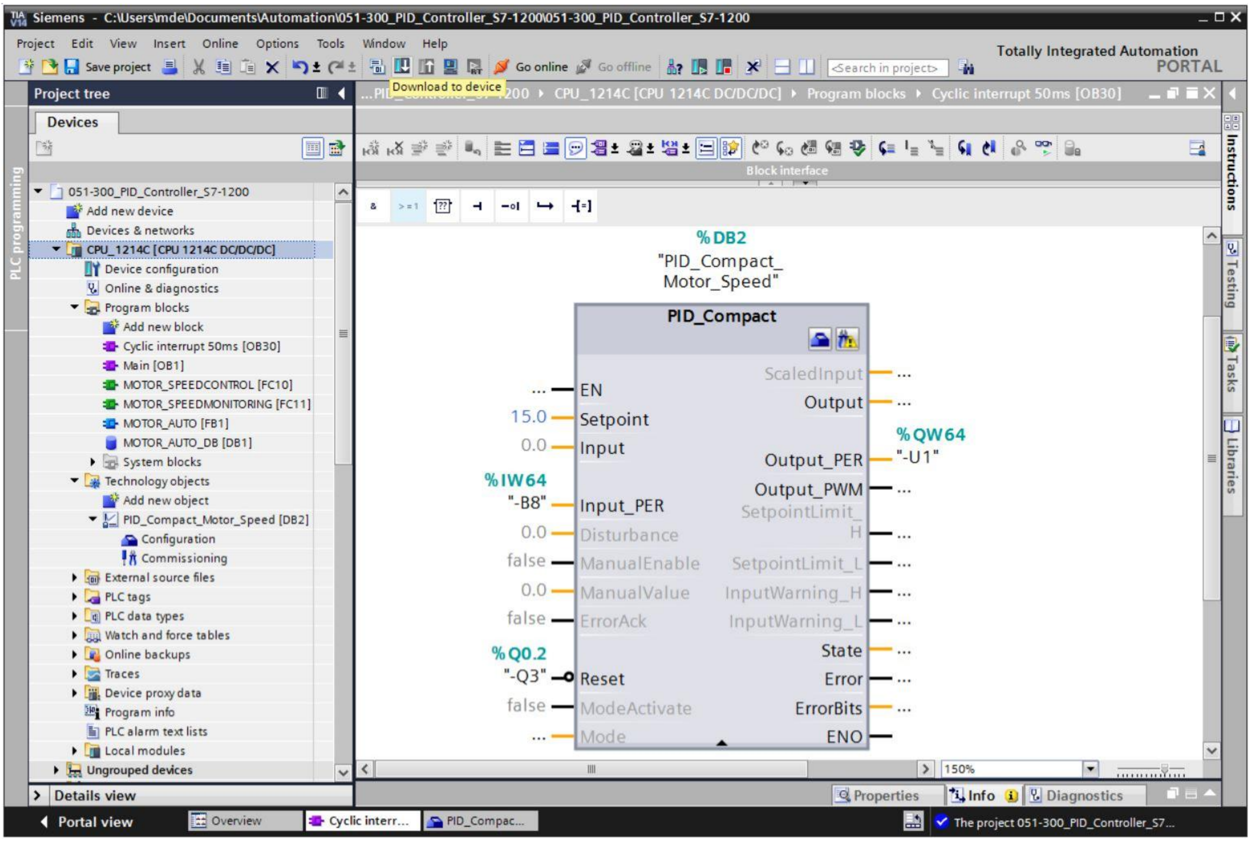


® The "Info", "Compile" area shows which blocks were successfully compiled.



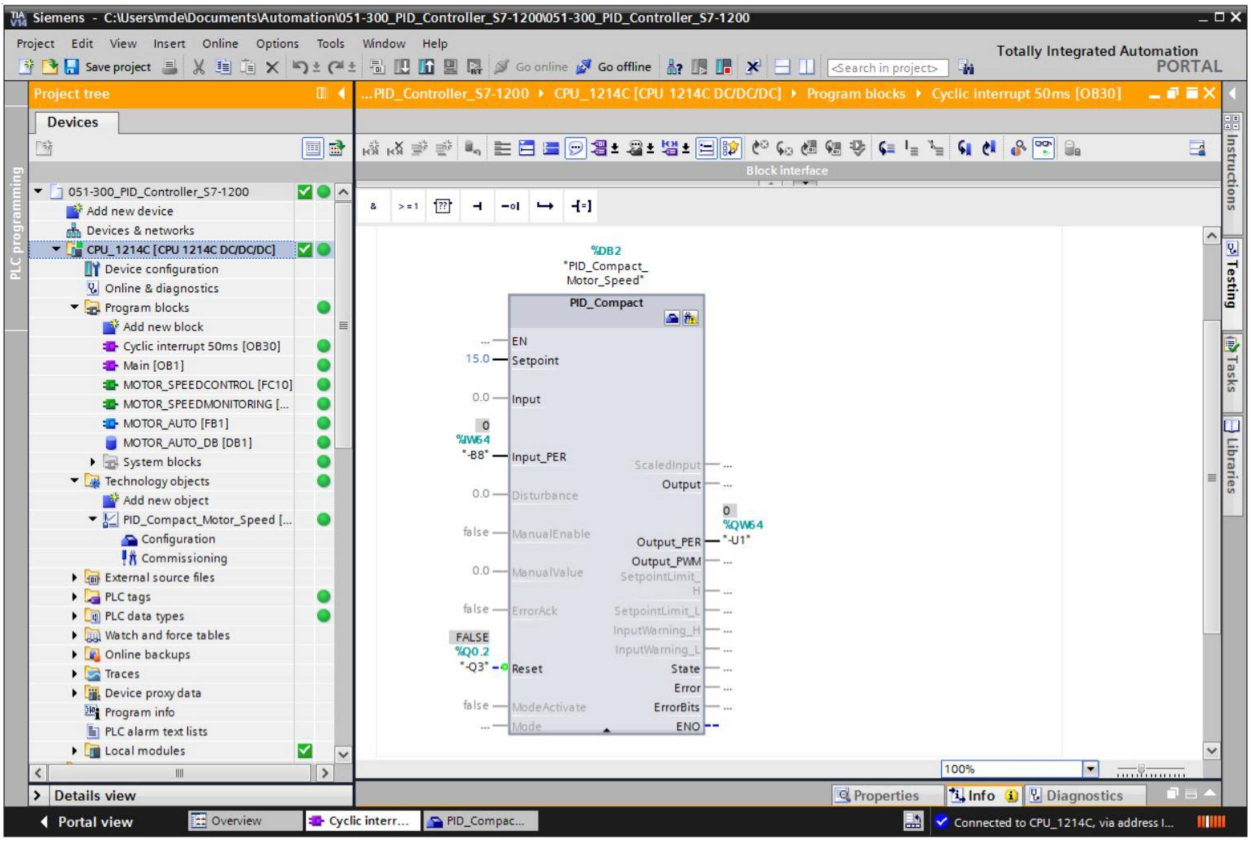
7.4 Download the program

® After successful compilation, the complete controller with the created program including the hardware configuration can, as described in the previous modules, be downloaded. (® )

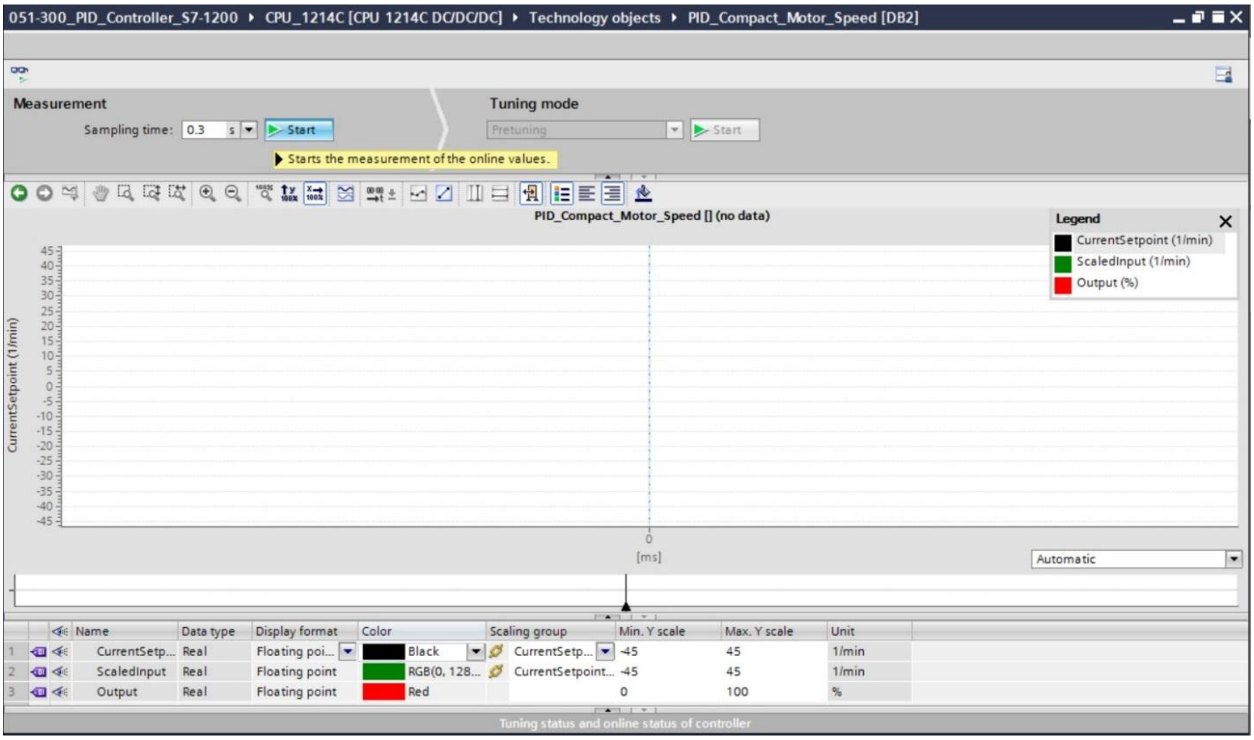


7.5 Monitor PID_Compact

® Click the Monitoring on/off icon to monitor the state of the blocks and tags when testing the program. At the first start of the CPU, however, the 'PID_Compact' controller is not yet tuned. We still have to start the tuning by clicking the ' Commissioning ' icon. (® Cyclic interrupt 50ms [OB30] ® ® PID_Compact ® Commissioning)

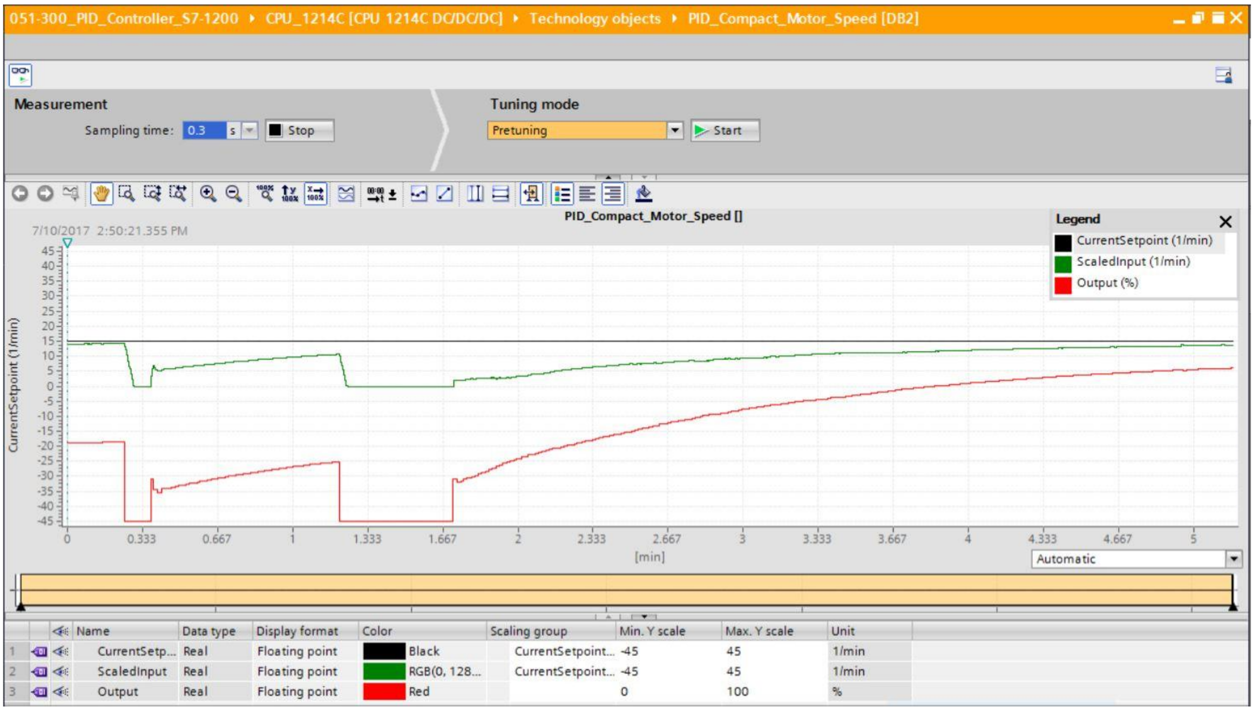


® If we click Start under 'Measurement', the values of the setpoint (Setpoint), actual value (ScaledInput) and manipulated variable (Output) can be displayed and monitored in a diagram. (® Start)



® The measurement can be stopped again by clicking 'Stop'.

(® Stop)



7.6 PID_Compact pretuning

The pretuning determines the process response to a step change of the output value and searches for the turning point. The PID parameters are calculated from the maximum slope and the dead time of the controlled system. The optimal PID parameters are obtained when you perform pretuning and fine tuning.

The more stable the actual value is, the easier and more accurately the PID parameters can be determined. Actual value noise is acceptable as long as the actual value rise is significantly greater than the noise. This is most likely the case in "Inactive" or "Manual mode" operating mode. The PID parameters are backed up before they are recalculated.

The following requirements must be met:

– The "PID_Compact" instruction is called in a cyclic interrupt OB.

– ManualEnable = FALSE

– Reset = FALSE

– PID_Compact is in "Manual mode", "Inactive" or "Automatic mode" operating mode.

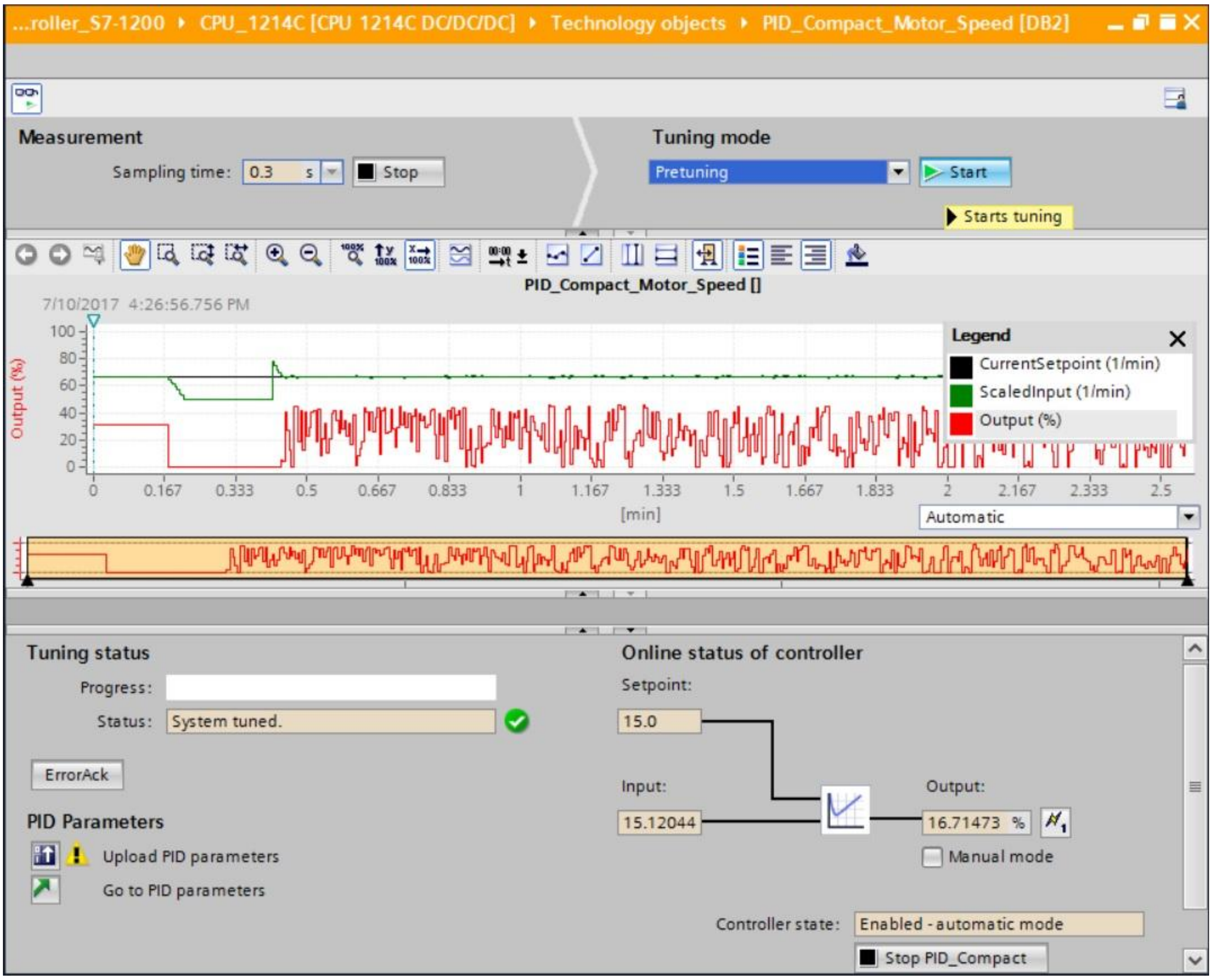
– The setpoint and actual value are within the configured limits (see "Process value monitoring" configuration).

– The difference between setpoint and actual value is greater than 30 % of the difference between the process value high limit and low limit.

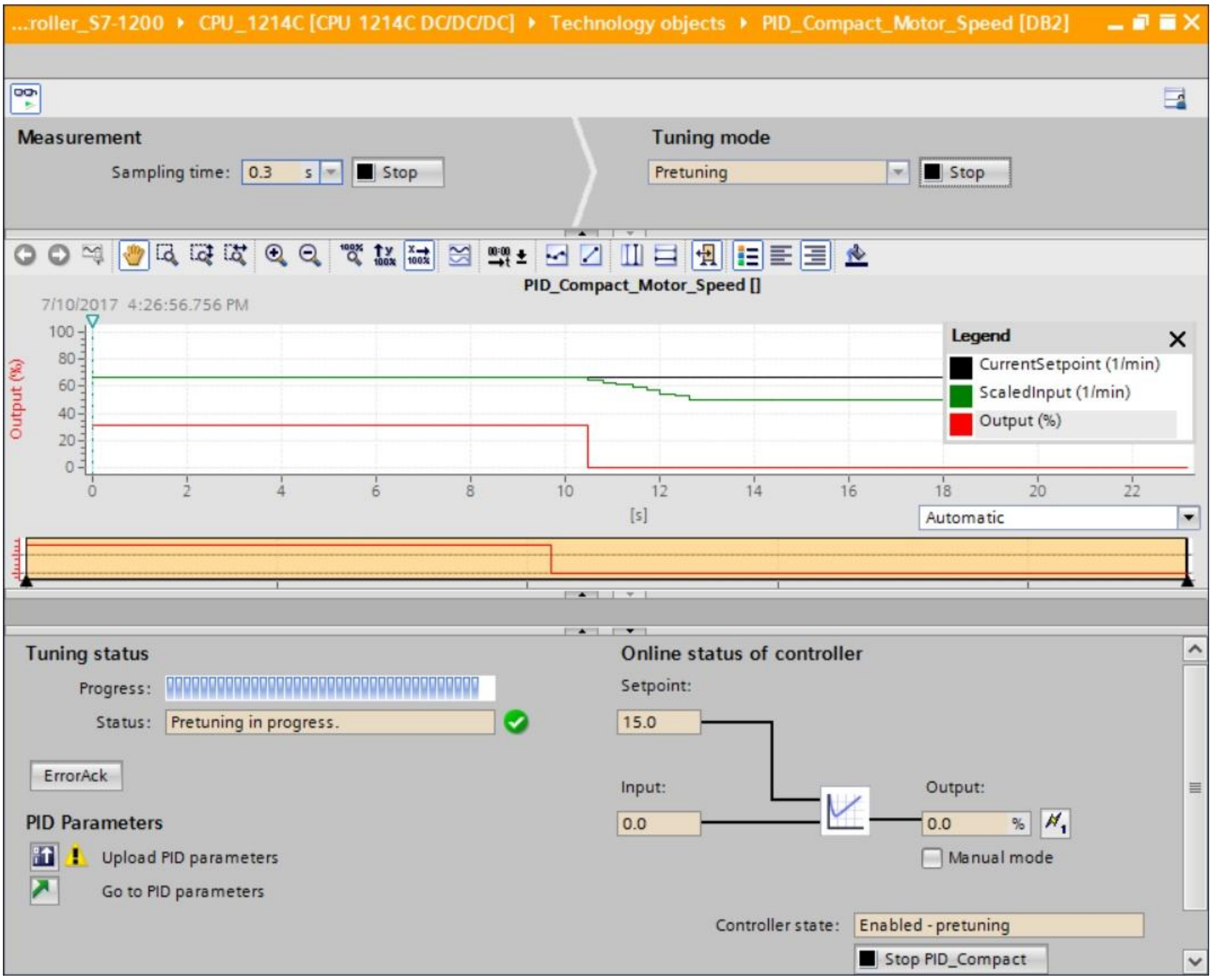
– The difference between setpoint and actual value is > 50 % of the setpoint.

® 'Pretuning' is selected as the 'Tuning mode' and this is then started.

(® Tuning mode ® Pretuning ® Start)



® The pretuning starts. The current work steps and any errors that occur are shown in the "Tuning status" field. The progress bar shows the progress of the current work step.



7.7 PID_Compact fine tuning

The fine tuning generates a constant, limited oscillation of the actual value. The PID parameters are optimized for the operating point based on the amplitude and frequency of this oscillation. All PID parameters are recalculated from the results. The PID parameters resulting from fine tuning generally produce a better response to setpoint changes and disturbances than the PID parameters from pretuning. The optimal PID parameters are obtained when you perform pretuning and fine tuning.

PID_Compact automatically attempts to generate an oscillation that is greater than the actual value noise. The fine tuning is influenced only slightly by the stability of the actual value. The PID parameters are backed up before they are recalculated.

The following requirements must be met:

– The "PID_Compact" instruction is called in a cyclic interrupt OB.

– ManualEnable = FALSE

– Reset = FALSE

– The setpoint and actual value are within the configured limits.

– The control loop is stable at the operating point. The operating point is reached when the actual value is equal to the setpoint.

– No disturbances are expected.

– PID_Compact is in "Manual mode", "Inactive" or "Automatic mode" operating mode.

The fine tuning runs as follows when started in automatic mode:

When you want to improve the existing PID parameters by tuning them, start the fine tuning from automatic mode.

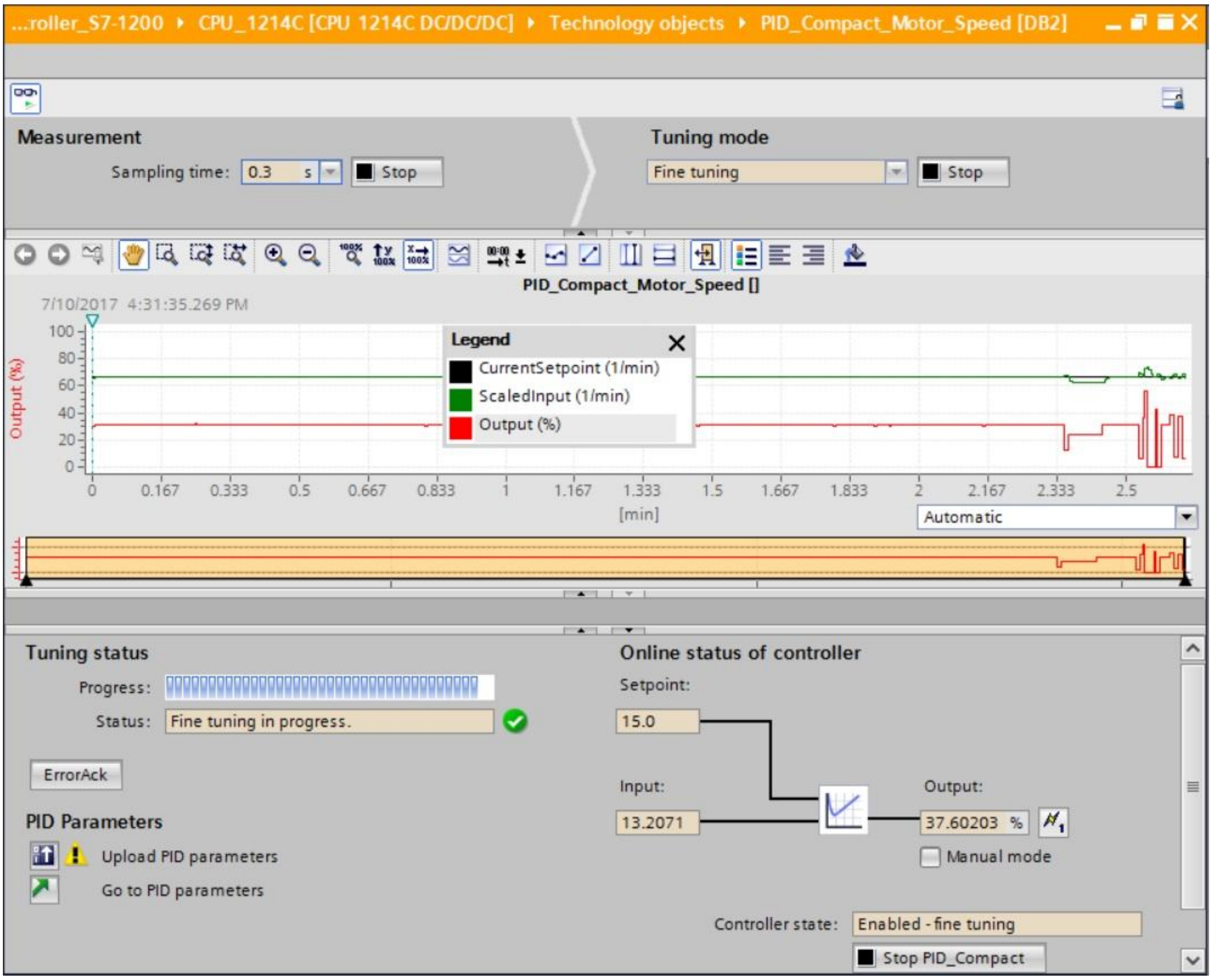
PID_Compact uses the existing PID parameters for controlling until the control loop is stable and the requirements for fine tuning are met. Only then does the fine tuning start.

The fine tuning runs as follows when started in inactive or manual mode:

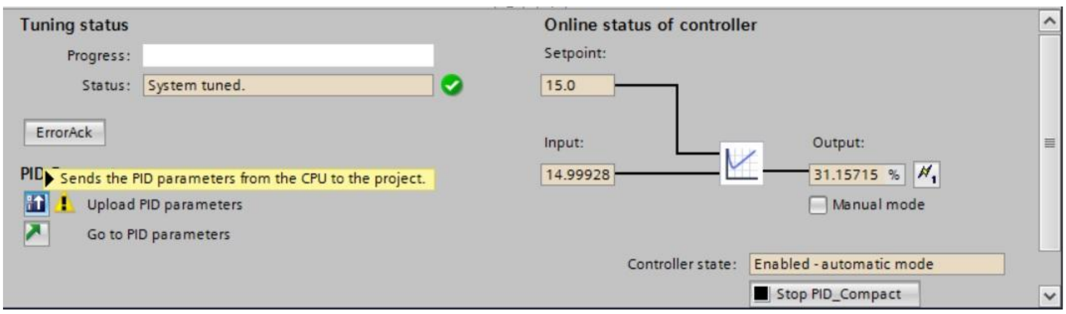
When the requirements for pretuning are met, pretuning is started. PID_Compact uses the determined PID parameters for controlling until the control loop is stable and the requirements for fine tuning are met. Only then does the fine tuning start. If pretuning is not possible, PID_Compact responds as configured in Response to error.

If the actual value is already too close to the setpoint for pretuning, an attempt is made to reach the setpoint with minimum or maximum output value. This can cause increased overshoot.

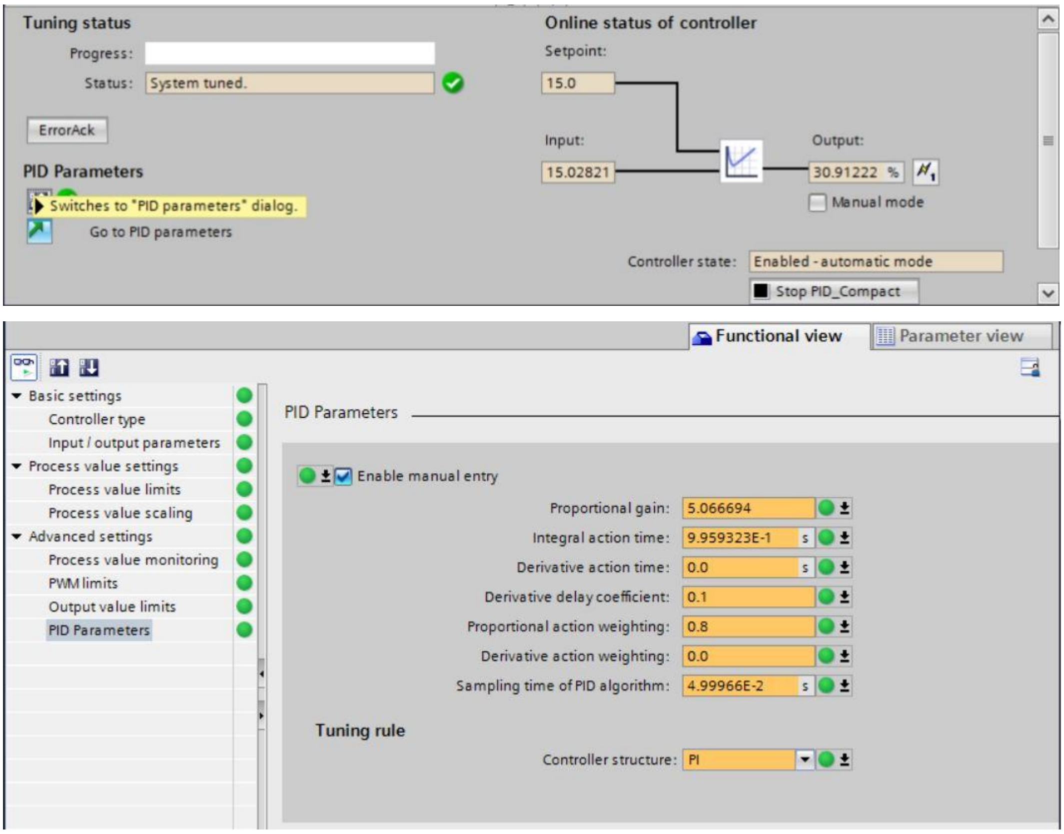
® 'Fine tuning' is selected as the 'Tuning mode' and this is then started.
(® Tuning mode ® Fine tuning ® Start)



® The fine tuning starts. The current work steps and any errors that occur are shown in the "Tuning status" field. If the self-tuning was completed without error message, the PID parameters have been tuned. The PID controller switches to automatic mode and uses the tuned parameters. The tuned PID parameters are retained at a Power ON and restart of the CPU. You can download the PID parameters from the CPU to your project with the ' ' button. (®)



® The PID parameters in the configuration can be displayed by clicking ''.
(®)



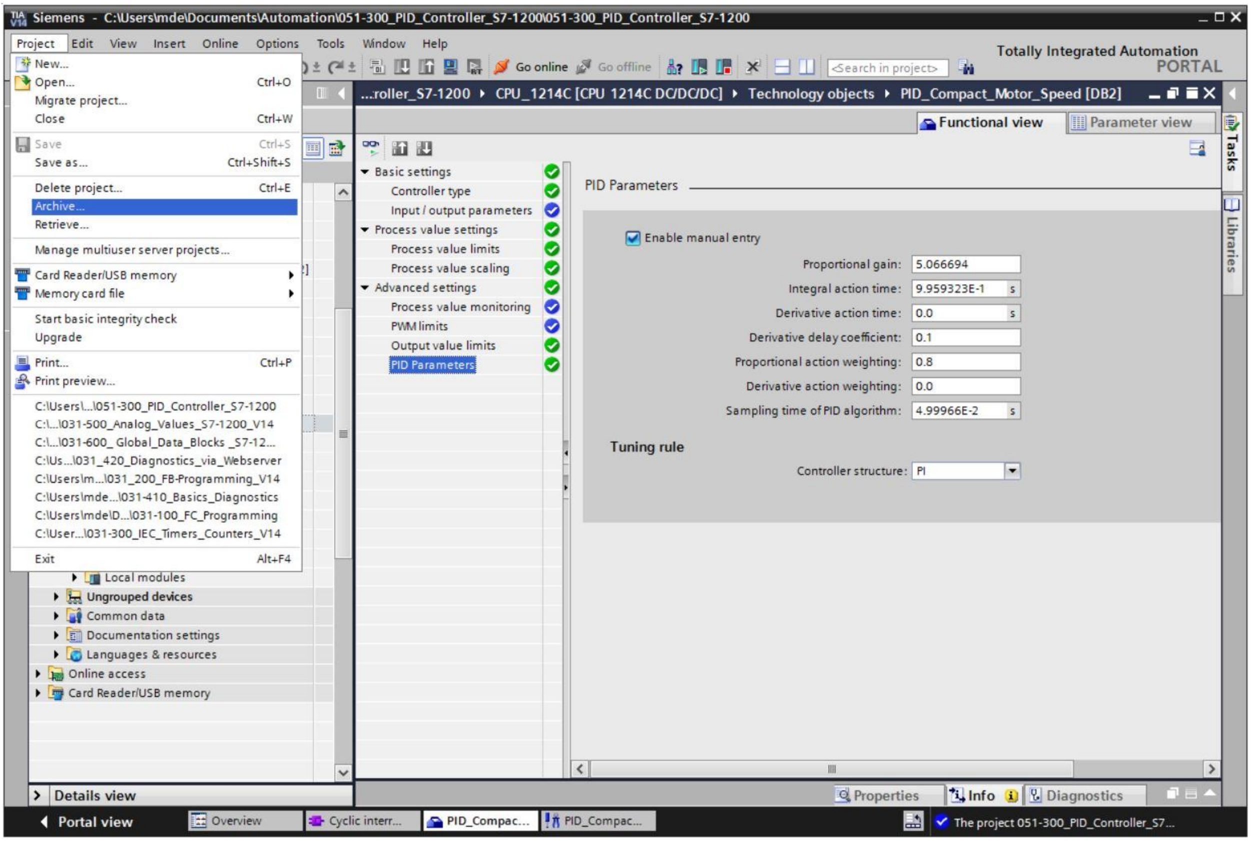
® As the final step, the online connection should be disconnected and the complete project should be saved.

(® Go offline ® Save project)

7.8 Archive the project

® Now we want to archive the complete project. Select the ® 'Archive ...' command in the ® 'Project' menu. Select a folder where you want to archive your project and save it with the file type "TIA Portal project archive".

(® Project ® Archive ® TIA Portal project archive ® 051-300_PID_Control_S7-1200…. ® Save)



8 Checklist

No.	Description	Completed
1	Cyclic interrupt OB Cyclic interrupt 50ms [OB30] successfully created.	
2	PID_Compact controller in cyclic interrupt OB Cyclic interrupt 50ms [OB30] called and connected.	
3	Configuration of the PID_Compact controller performed.	
4	Compiling successful and without error message	
5	Download successful and without error message	
6	Pretuning successful and without error message	
7	Fine tuning successful and without error message	
8	Switch on station (-K0 = 1) Cylinder retracted / Feedback activated (-B1 = 1) EMERGENCY OFF (-A1 = 1) not activated AUTOMATIC mode (-S0 = 1) Pushbutton automatic stop not actuated (-S2 = 1) Briefly press the automatic start pushbutton (-S1 = 1) Sensor part at slide activated (-B4 = 1) then Conveyor motor M1 variable speed (-Q3 = 1) switches on and stays on. The speed corresponds to the speed setpoint in the range +/- 50 rpm	
9	Sensor part at end of conveyor activated (-B7 = 1) ® -Q3 = 0 (after 2 seconds)	
10	Briefly press the automatic stop pushbutton (-S2 = 0) ® -Q3 = 0	
11	Activate EMERGENCY OFF (-A1 = 0) ® -Q3 = 0	
12	Manual mode (-S0 = 0) ® -Q3 = 0	
13	Switch off station (-K0 = 0) ® -Q3 = 0	
14	Cylinder not retracted (-B1 = 0) ® -Q3 = 0	
15	Speed > Motor_speed_monitoring_error_max ® -Q3 = 0	
16	Speed < Motor_speed_monitoring_error_min ® -Q3 = 0	
17	Project successfully archived	

9 Additional information

More information for further practice and consolidation is available as orientation, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software / firmware, under the following link:

www.siemens.com/sce/s7-1200

Preview „Additional information“

Getting Started, Videos, Tutorials, Apps, Manuals, Trial-SW/Firmware

- TIA Portal Videos
- TIA Portal Tutorial Center
- Getting Started
- Programming Guideline
- Easy Entry in SIMATIC S7-1200
- Download Trial Software/Firmware
- Technical Documentation SIMATIC Controller
- Industry Online Support App
- TIA Portal, SIMATIC S7-1200/1500 Overview
- TIA Portal Website
- SIMATIC S7-1200 Website
- SIMATIC S7-1500 Website

Further Information

Siemens Automation Cooperates with Education
siemens.com/sce

SCE Learn-/Training Documents
siemens.com/sce/documents

SCE Trainer Packages
siemens.com/sce/tp

SCE Contact Partners
siemens.com/sce/contact

Digital Enterprise
siemens.com/digital-enterprise

Industrie 4.0
siemens.com/future-of-manufacturing

Totally Integrated Automation (TIA)
siemens.com/tia

TIA Portal
siemens.com/tia-portal

SIMATIC Controller
siemens.com/controller

SIMATIC Technical Documentation
siemens.com/simatic-docu

Industry Online Support
support.industry.siemens.com

Product catalogue and online ordering system Industry Mall
mall.industry.siemens.com

Siemens AG
Digital Factory
P.O. Box 4848
90026 Nuremberg
Germany

siemens.com/sce