Chapter 22 Ball-in-Tube - PID

Ball-in-Tube Lab

This lab is one of the more successful labs. It is used by the PLC courses as well as the automatic controls course. These courses can use the lab separately in the same day and not interfere with the other. These are extremely good characteristics for a lab and much planning has gone onto the present labs to accomplish these goals. The cost is approximately $500. This cost has dropped somewhat in the past years due to the decreased cost of lasers used as the feedback device. The planning for this lab has been ongoing. The most recent change included a separate PLC for the auto-controls lab. While the tube is very cumbersome and difficult to store, protecting the ball in the tube is an important part of the lab. The ball must be suspended slightly above the fan in order for it to take off. The tube can be removed from the base for shipment as necessary.

Inserting the PID instruction and technological object

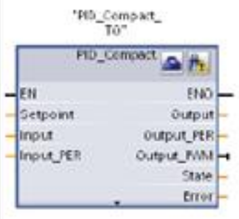
STEP 7 provides two instructions for PID control. Use the PID_Compact instruction for the lab in this course, please!

The PID_Compact instruction and its associated technological object provide a universal PID controller with tuning. The technological object contains all of the settings for the control loop.

The PID_3Step instruction and its associated technological object provide a PID controller with specific settings for motor-activated valves. The technological object contains all of the settings for the control loop. The PID_3Step controller provides two additional Boolean outputs.

After creating the technological object, you must configure the parameters. You also adjust the autotuning parameters ("pretuning" during startup or manual "fine tuning") to commission the operation of the PID controller.

LAD / FBD	SCL	Description

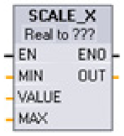
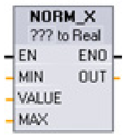
 "PID_Compact_T0"
PID_Compact
EN
ENO
Setpoint
Output
Input
Output_PER
Input_PER
Output_PWM
State
Error
	`"PID_Compact_1"(`
	`Setpoint:=_real_in_,`
	`Input:=_real_in_,`
	`Input_PER:=_word_in_,`
	`ManualEnable:=_bool_in_,`
	`ManualValue:=_real_in_,`
	`Reset:=_bool_in_,`
	`ScaledInput=>_real_out_,`
	`Output=>_real_out_,`
	`Output_PER=>_word_out_,`
	`Output_PWM=>_bool_out_,`
	`SetpointLimit_H=>_bool_out_,`
	`SetpointLimit_L=>_bool_out_,`
	`InputWarning_H=>_bool_out_,`
	`InputWarning_L=>_bool_out_,`
	`State=>_int_out_,`
	`Error=>_dword_out_);`
		PID_Compact provides a PID controller with self-tuning for automatic and manual mode. PID_Compact is a PIDT1 controller with anti-windup and weighting of the P- and D-component.

[1] STEP 7 automatically creates the technological object and instance DB when you insert the instruction. The instance DB contains the parameters of the technological object.

[2] In the SCL example, "PID_Compact_1" is the name of the instance DB.

When programming the inputs and outputs, the following two instructions are used to scale and normalize the analog value. Use the NORM_X function first to convert the number to a real in the range 0-1 and then use SCALE_X to scale the normalized value to a range for the real value.

Table 6- 6 SCALE_X and NORM_X instructions

LAD / FBD	SCL	Description
 SCALE_X Real to ??? EN ENO MIN OUT VALUE MAX	`out := SCALE_X( min,:=_undef_in_ value:=_real_in_, max:=undef_in_);`	Scales the normalized real parameter VALUE where (0.0 <= VALUE <= 1.0) in the data type and value range specified by the MIN and MAX parameters: OUT = VALUE (MAX - MIN) + MIN
 NORM_X ??? to Real EN ENO MIN OUT VALUE MAX	`out := NORM_X( min:=_,undef_in_ value:=_undef_in_, max:=_undef_in_);`	Normalizes the parameter VALUE inside the value range specified by the MIN and MAX parameters: OUT = (VALUE - MIN) / (MAX - MIN), where (0.0 <= OUT <= 1.0)

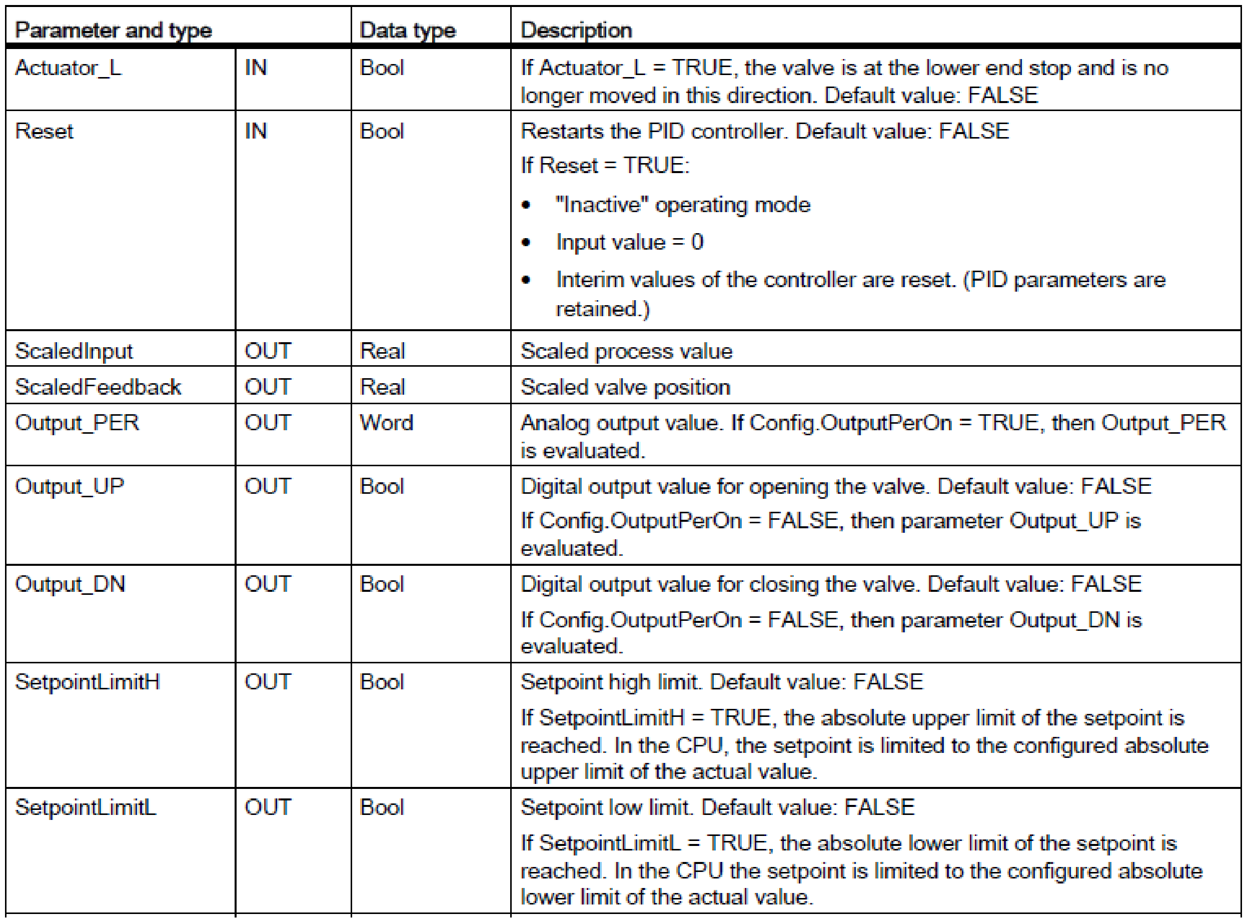
1 Equivalent SCL: `out := value (max-min) + min;`2 Equivalent SCL: `out := (value-min)/(max-min);`

Descriptions of various parameters in the PID block are found below:

Parameter and type		Data type	Description
Setpoint	IN	Real	Setpoint of the PID controller in automatic mode. Default value: 0.0
Input	IN	Real	Process value. Default value: 0.0 You must also set Config.InputPEROn = FALSE.
Input_PER	IN	Word	Analog process value (optional). Default value: W#16#0 You must also set Config.InputPEROn = TRUE.
ManualEnable	IN	Bool	Enables or disables the manual operation mode. Default value: FALSE • On the edge of the change from FALSE to TRUE, the PID controller switches to manual mode, State = 4, and Retain.Mode remains unchanged. • On the edge of the change from TRUE to FALSE, the PID controller switches to the last active operating mode and State = Retain.Mode.
ManualUP	IN	Bool	In manual mode, every rising edge opens the valve by 5% of the total actuating range, or for the duration of the minimum motor actuation time. ManualUP is evaluated only if you are using OutputPer **and** if position feedback is available. Default value: FALSE • If Output_PER is FALSE, the manual input turns Output_UP on for the time that corresponds to a movement of 5% of the device. • If Config.ActuatorEndStopOn is TRUE, then Output_UP does not come on if Actuator_H is TRUE.
ManualDN	IN	Bool	In manual mode, every rising edge closes the valve by 5% of the total actuating range, or for the duration of the minimum motor actuation time. ManualDN is evaluated only if you are using OutputPer **and** if position feedback is available. Default value: FALSE • If Output_PER is FALSE, the manual input turns Output_DN on for the time that corresponds to a movement of 5% of the device. • If Config.ActuatorEndStopOn is TRUE, then Output_DN does not turn on if Actuator_L is TRUE.
ManualValue	IN	Real	Process value for manual operation. Default value: 0.0 In manual mode, you specify the absolute position of the valve. ManualValue is evaluated only if you are using OutputPer, **or** if position feedback is available. Default value: 0.0
Feedback	IN	Real	Position feedback of the valve. Default value: 0.0 To use Feedback, then set Config.FeedbackPerOn = FALSE.

The values in the table above are necessary to make the PID block work correctly. Some may be set once and not included in the program as variables. Others must be included as programmed variables. For example, if Input_PER is used, this input must be represented as a percent from 0 to 100.0. This value is the value fed to the PID block from the analog process variable, in this case the laser. The variable must be represented in Input_PER as a ratio from 0 to 100.

Other variables in the table above are useful when coordinating with the faceplate. For example, if the PID algorithm is set to manual, the ManualValue variable must be set to the desired state of the output of the PID. The variable is moved to this location and the output is set to this value.

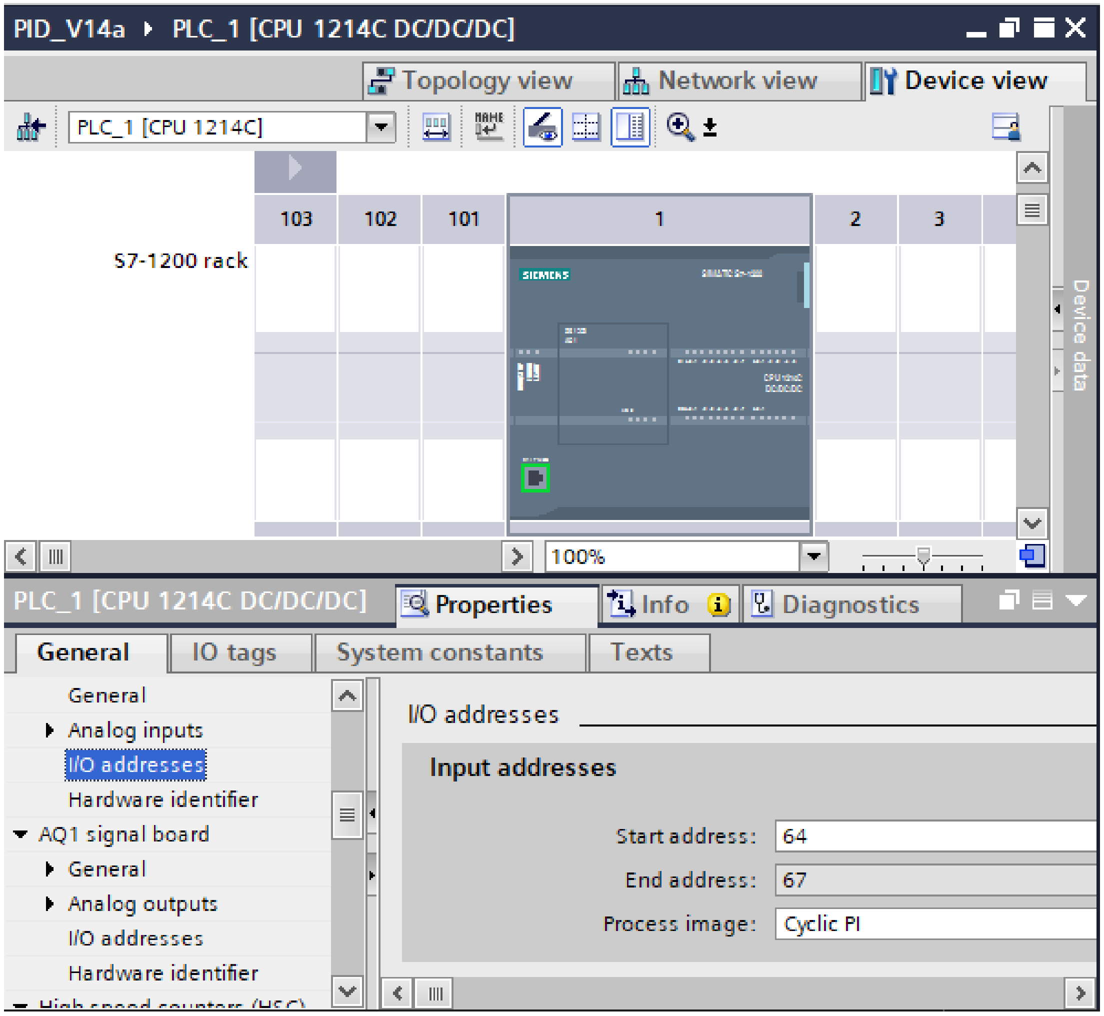


Parameter and type		Data type	Description
Actuator_L	IN	Bool	If Actuator_L = TRUE, the valve is at the lower end stop and is no longer moved in this direction. Default value: FALSE
Reset	IN	Bool	Restarts the PID controller. Default value: FALSE If Reset = TRUE: • "Inactive" operating mode • Input value = 0 • Interim values of the controller are reset. (PID parameters are retained.)
ScaledInput	OUT	Real	Scaled process value
ScaledFeedback	OUT	Real	Scaled valve position
Output_PER	OUT	Word	Analog output value. If Config.OutputPerOn = TRUE, then Output_PER is evaluated.
Output_UP	OUT	Bool	Digital output value for opening the valve. Default value: FALSE If Config.OutputPerOn = FALSE, then parameter Output_UP is evaluated.
Output_DN	OUT	Bool	Digital output value for closing the valve. Default value: FALSE If Config.OutputPerOn = FALSE, then parameter Output_DN is evaluated.
SetpointLimitH	OUT	Bool	Setpoint high limit. Default value: FALSE If SetpointLimitH = TRUE, the absolute upper limit of the setpoint is reached. In the CPU, the setpoint is limited to the configured absolute upper limit of the actual value.
SetpointLimitL	OUT	Bool	Setpoint low limit. Default value: FALSE If SetpointLimitL = TRUE, the absolute lower limit of the setpoint is reached. In the CPU the setpoint is limited to the configured absolute lower limit of the actual value.

InputWarningH	OUT	Bool	If InputWarningH = TRUE, the input value has reached or exceeded the upper warning limit. Default value: FALSE
InputWarningL	OUT	Bool	If InputWarningL = TRUE, the input value has reached or exceeded the lower warning limit. Default value: FALSE
State	OUT	Int	Current operating mode of the PID controller. Default value: 0 Use Retain.Mode to change the operating mode: • State = 0: Inactive • State = 1: Pretuning • State = 2: Manual fine tuning • State = 3: Automatic mode • State = 4: Manual mode • State = 5: Safety mode • State = 6: Output value measurement • State = 7: Safety mode monitoring with active trigger • State = 8: Inactive mode monitoring with active trigger
Error	OUT	Bool	If Error = TRUE, at least one error message is pending. Default value: FALSE
ErrorBits	OUT	DWord	Error message. Default value: DW#16#0000 (no error)

Likewise, these variables contain information to allow the PID algorithm to function properly. The state is a number from 0 to 8. We only use the values of 3 and 4 for the application given in the Ball-in-Tube program.

The I/O address of the analog input point is shown in the analog input addresses of the base processor unit. If additional analog points beyond two or if these points need a floating neutral, then an additional analog input card is needed. In our example for the ball-in-tube lab, the input addresses start at I:64. The first address is bytes I:64 and 65. The second input address is bytes I:66 and 67



I/O addresses

Input addresses

Start address: 64

End address: 67

Process image: Cyclic PI

Display of the analog points is done on a historical data plot shown below.

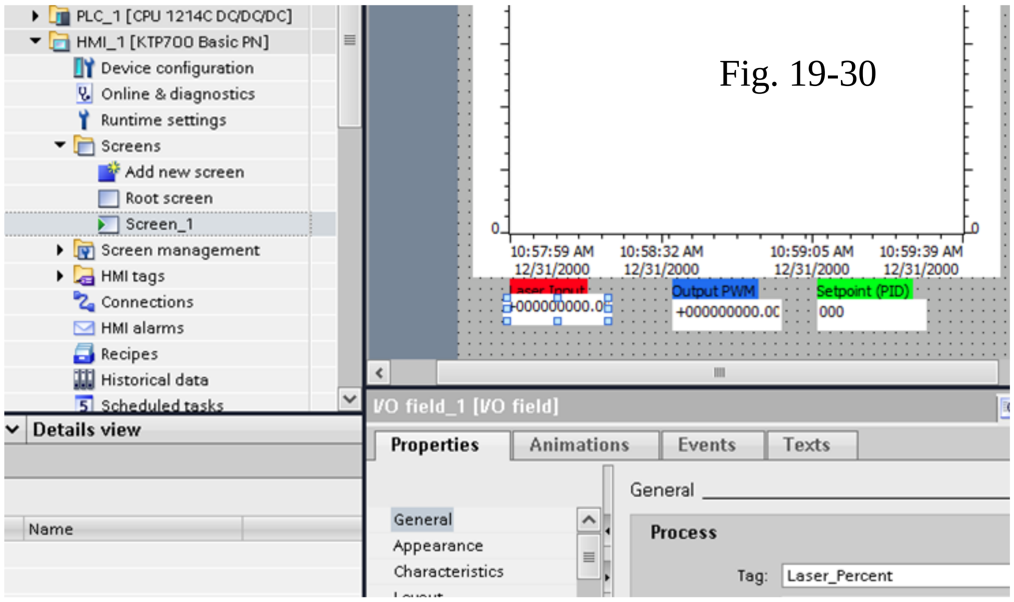
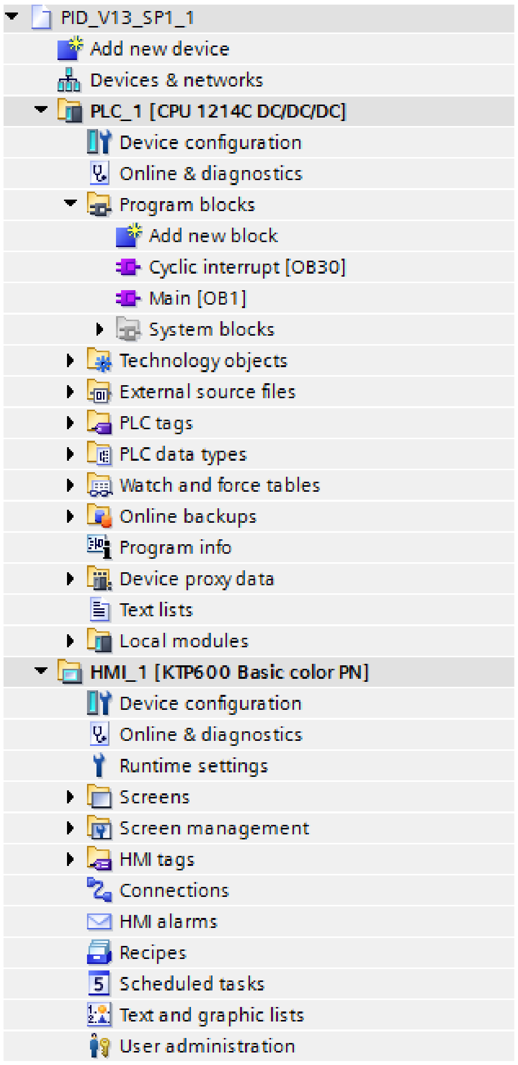


Fig. 19-30

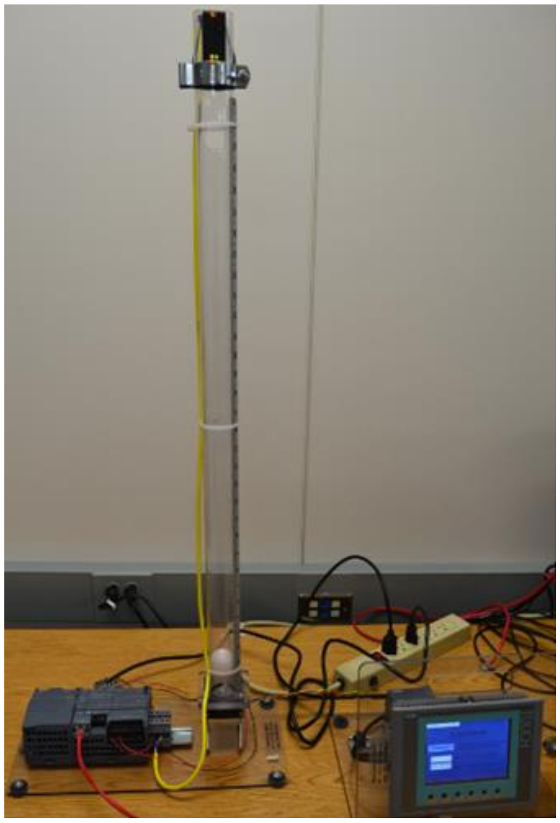
General

Process

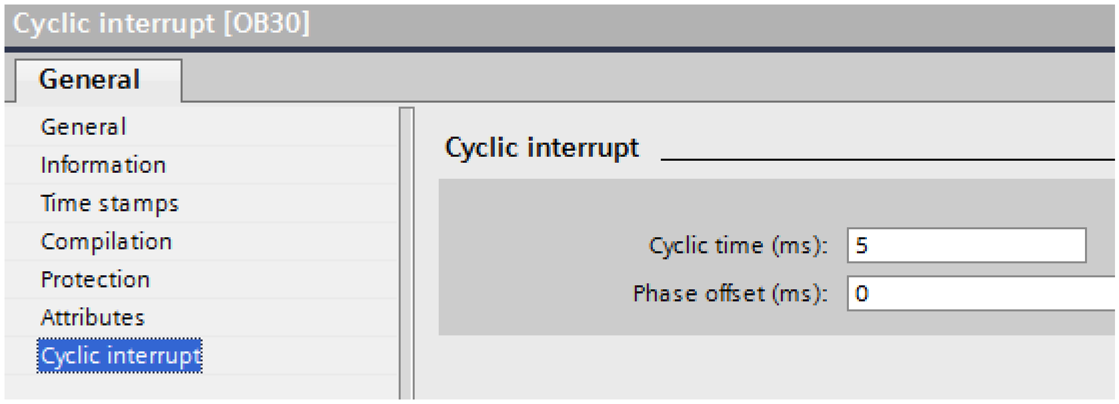
Tag: Laser_Percent


- PID_V13_SP1_1
  - Add new device
  - Devices & networks
  - PLC_1 [CPU 1214C DC/DC/DC]
    - Device configuration
    - Online & diagnostics
    - Program blocks
      - Add new block
      - Cyclic interrupt [OB30]
      - Main [OB1]
      - System blocks
    - Technology objects
    - External source files
    - PLC tags
    - PLC data types
    - Watch and force tables
    - Online backups
    - Program info
    - Device proxy data
    - Text lists
    - Local modules
  - HMI_1 [KTP600 Basic color PN]
    - Device configuration
    - Online & diagnostics
    - Runtime settings
    - Screens
    - Screen management
    - HMI tags
    - Connections
    - HMI alarms
    - Recipes
    - Scheduled tasks
    - Text and graphic lists
    - User administration

Project Tree for Ball-in-Tube



Physical Layout of Ball-in-TUbe


Cyclic interrupt [OB30]

General

- General
- Information
- Time stamps
- Compilation
- Protection
- Attributes
- Cyclic interrupt

Cyclic interrupt

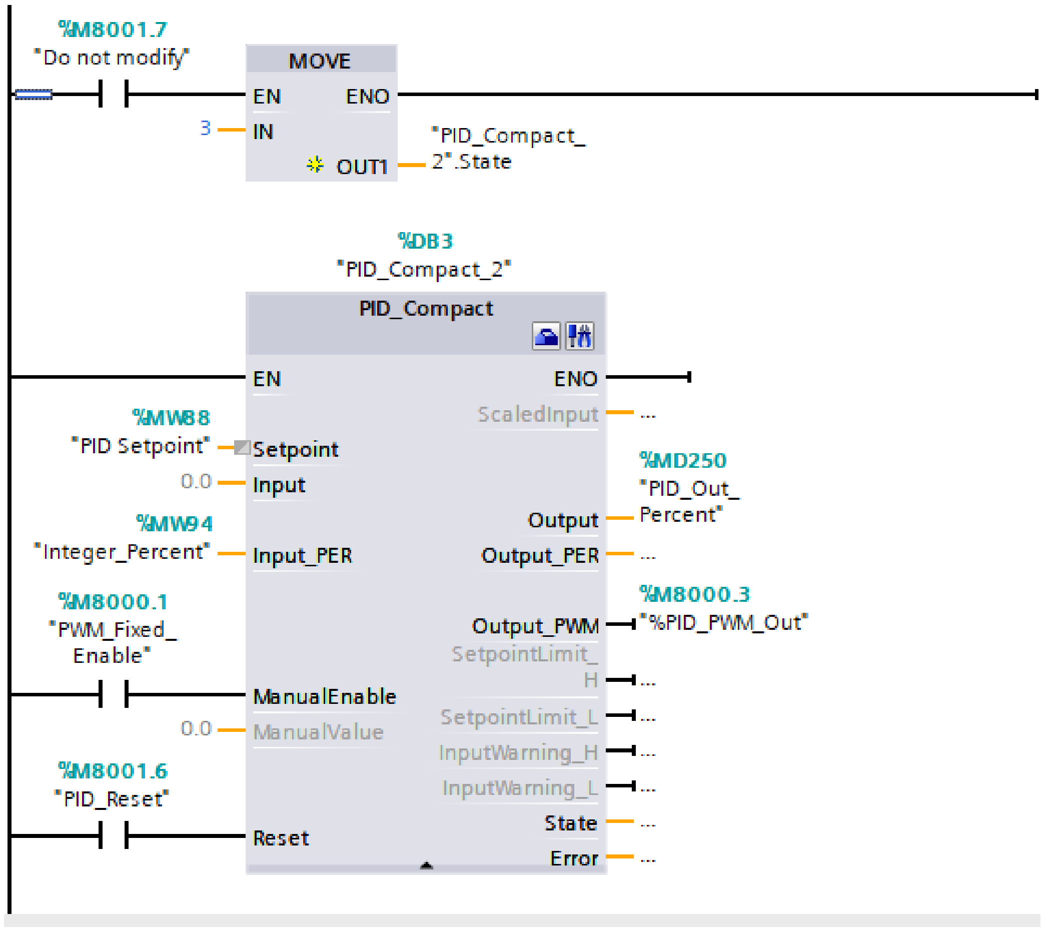
Cyclic time (ms): 5

Phase offset (ms): 0

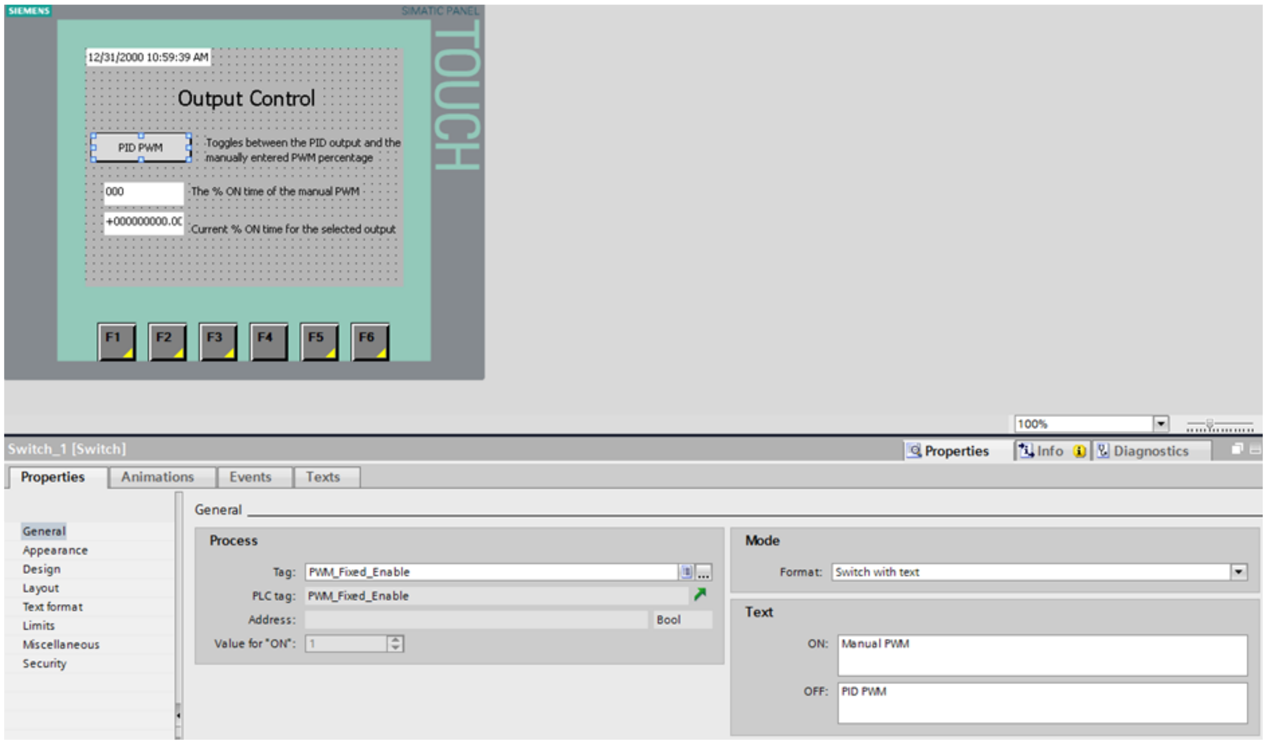
Setting up the Cyclic Interrupt (OB30)

A separate Cyclic Interrupt Program must be built to provide execution of the PID Block. The PID program executes the PID algorithm after reading the Process Variable input. After execution of the algorithm, the PWM output determines the state of the output to the fan.

The PID algorithm for the Ball-in-Tube program is shown below. The instruction is configured and set up from programming statements as well as constants entered into the tables above.



The HMI panel below has a button to choose between auto and manual. In PID_PWM, the button is in automatic. When in auto, the setpoint is entered on a separate page. The manual value for the PID output may be entered below the button in manual mode.



Switch_1 [Switch]

General

Process

Tag: PWM_Fixed_Enable
PLC tag: PWM_Fixed_Enable
Address: Bool
Value for "ON": 1

Mode

Format: Switch with text

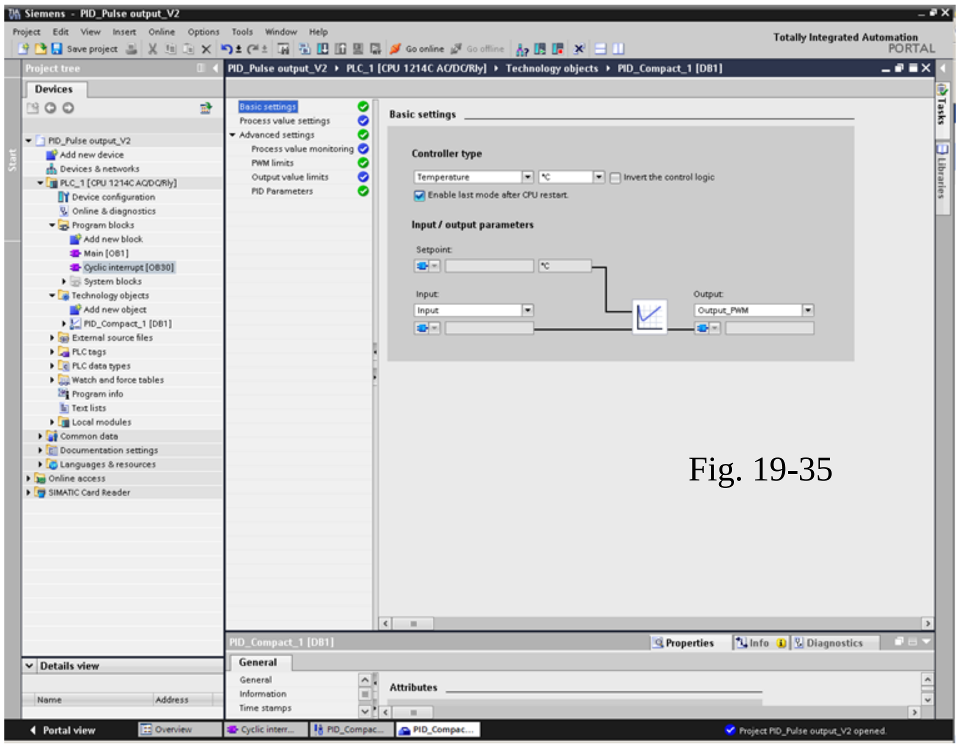
Text

ON: Manual PWM
OFF: PID PWM

Below the button is a data entry window for the value of percent on time for the fan. In thest window is the percent on time for the output.

The Configuration editor for PID_Compact shows the following screen. Here, the user selects the units such as temperature or pressure. The user also determines whether variables such as the PV are Input or Input_Per. Most users would select ‘general’ for controller type.

Use the commissioning editor to configure the controller for auto-tuning at startup and for auto-tuning during operation. To open the commissioning editor, click the icon on either the instruction or the project navigator.



Controller type: Temperature | °C | ☐ Invert the control logic

☑ Enable last mode after CPU restart.

Setpoint: °C

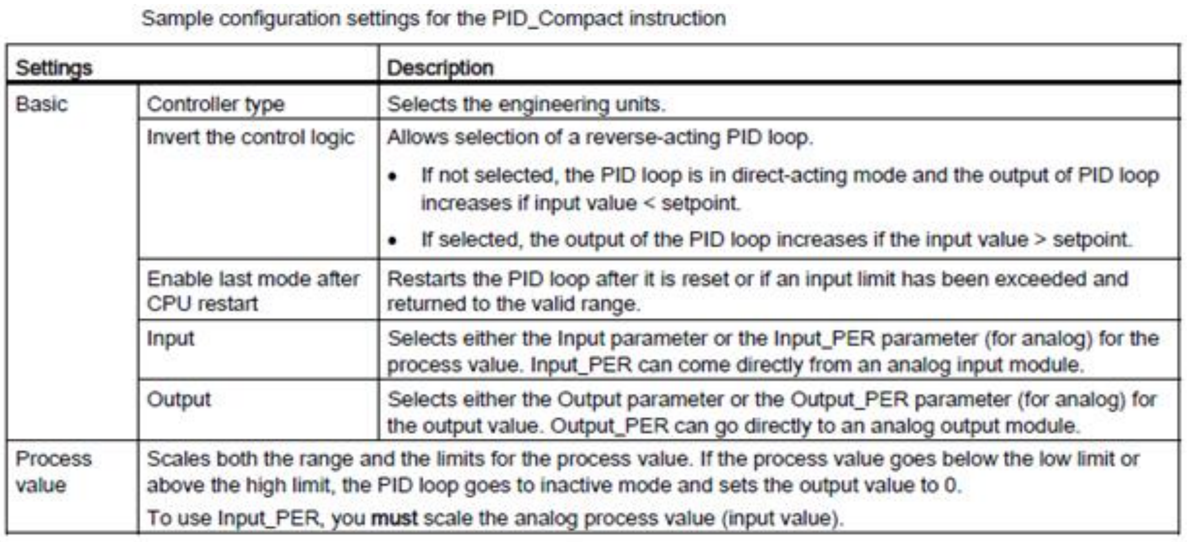
Input: Input

Output: Output_PWM

✓ Project PID_Pulse output_V2 opened.

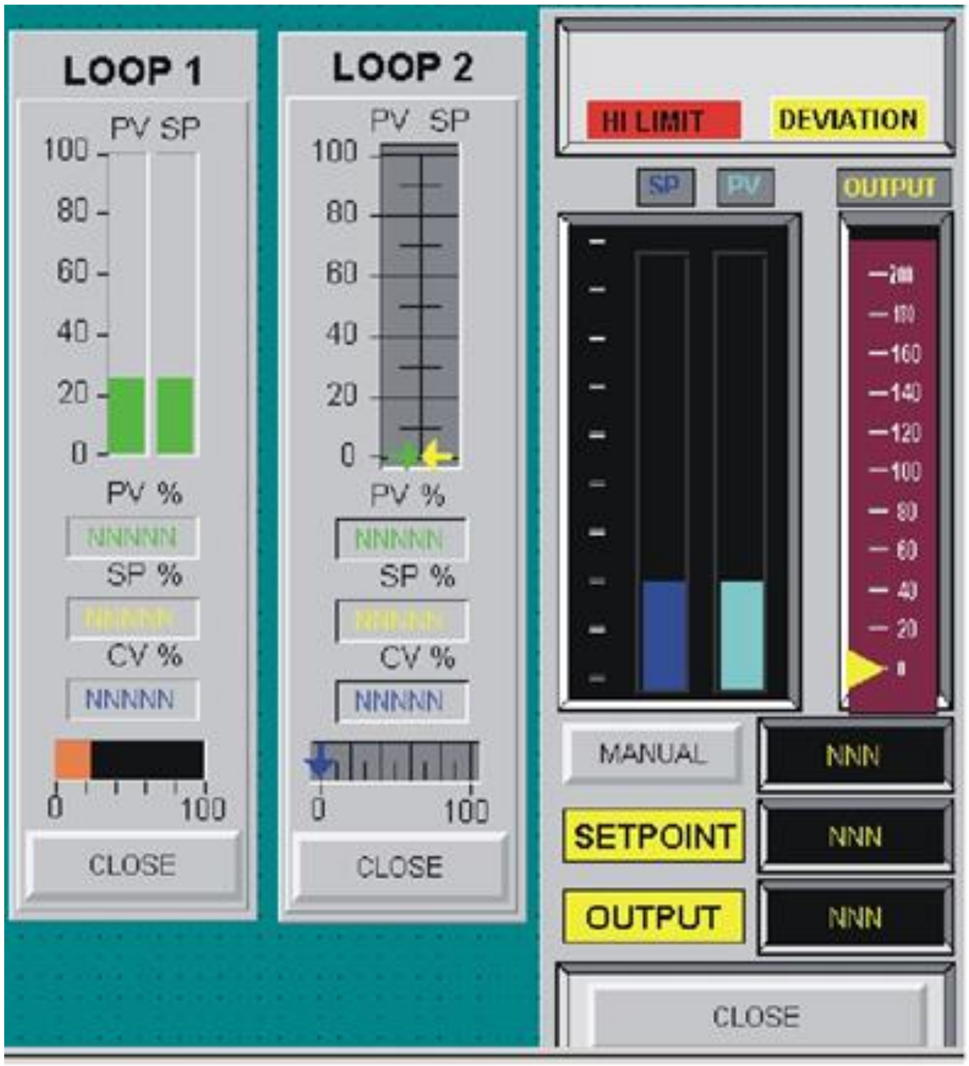
Fig. 19-35

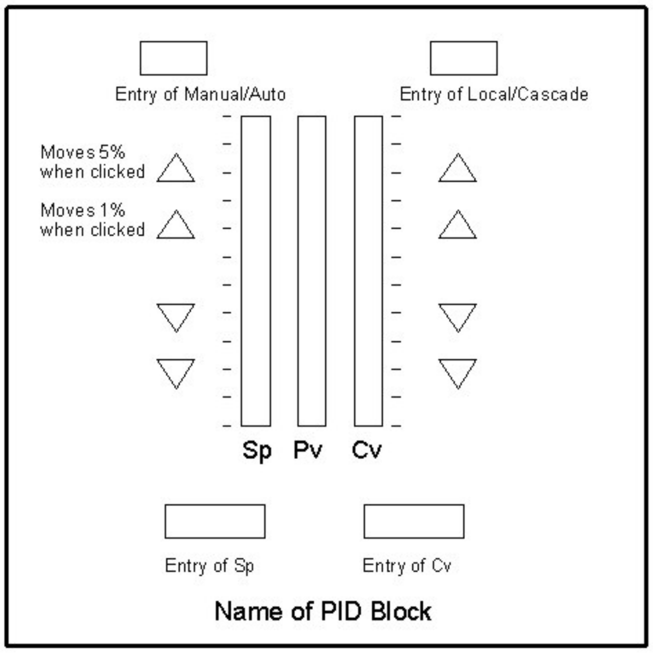
The following table lists some common suggested actions for assisting the set-up of the PID controller:



Sample configuration settings for the PID_Compact instruction

Settings		Description
Basic	Controller type	Selects the engineering units.
	Invert the control logic	Allows selection of a reverse-acting PID loop. • If not selected, the PID loop is in direct-acting mode and the output of PID loop increases if input value < setpoint. • If selected, the output of the PID loop increases if the input value > setpoint.
	Enable last mode after CPU restart	Restarts the PID loop after it is reset or if an input limit has been exceeded and returned to the valid range.
	Input	Selects either the Input parameter or the Input_PER parameter (for analog) for the process value. Input_PER can come directly from an analog input module.
	Output	Selects either the Output parameter or the Output_PER parameter (for analog) for the output value. Output_PER can go directly to an analog output module.
Process value	Scales both the range and the limits for the process value. If the process value goes below the low limit or above the high limit, the PID loop goes to inactive mode and sets the output value to 0. To use Input_PER, you **must** scale the analog process value (input value).	


LOOP 1
PV SP
100
80
60
40
20
0
PV %
NNNNN
SP %
NNNNN
CV %
NNNNN
0
100
CLOSE
LOOP 2
PV SP
100
80
60
40
20
0
PV %
NNNNN
SP %
NNNNN
CV %
NNNNN
0
100
CLOSE
HI LIMIT
DEVIATION
SP
PV
OUTPUT
200
180
160
140
120
100
80
60
40
20
0
MANUAL
NNN
SETPOINT
NNN
OUTPUT
NNN
CLOSE


Entry of Manual/Auto
Entry of Local/Cascade
Moves 5% when clicked
Moves 1% when clicked
Sp
Pv
Cv
Entry of Sp
Entry of Cv
Name of PID Block