Smart People. Expert Solutions.

SIEMENS OPEN LIBRARY

6 – PID Configuration

JULY 11, 2016

Contents

1. Purpose

The purpose of this document is to assist with configuration of the PID Open Library Object. In order to use the features of the built in Technology Object for the PID Compact Block, the library object requires special setup. The PID Comact Block only exists on the S7-1200 and S7-1500. The Open Library Object was written to be compatible with PID_Compact V2.2, and capabilities cannot be guaranteed for other versions of the PID Compact block.

2. Intended Use

This document is to be used by anyone utilizing the Siemens Open Library fbPID_CompactInterface. This document is used to configure the PLC and HMI objects for the library, as the configuration of these objects is not standard with the rest of the library.

3. Revision History

Version	Date	Author	Comments
1.0	2016-05-23	DMC	Initial Release
1.1	2016-07-11	DMC	Updated screenshots for PID_Compact and the interface function block

4. Open Library License

5. Hardware and Software Compatibility

This library was developed in TIA Portal V13 SP1. It was tested on the S7-1200 and S7-1500 platforms, and untested modifications have been made for compatibility with S7-300 and S7-400. The PLC objects can be used with any HMI, however, the configuration of the faceplates is only available using a Comfort Panel or WinCC Advanced, and have been tested on a 7" Comfort Panel.

6. Open Library PID Compact Setup

The following steps walk through the configuration of the PID Open Library Object. In order to use the features of the built-in Technology Object settings for the PID_Compact Block, the library object requires special setup.

The benefit of using the Library Interface block are as follows:

1. It provides a block to call and execute logic for the PID control in the Main Scan, which limits the logic located in the Cyclic OB.
2. It provides an easy interface for operation and changes to the PID from an HMI faceplate.
3. It utilizes the technology object, so all of the PID functions built in to portal can still be used.

6.1. Initial Setup

Before starting this document please make sure you have set up your project following the steps in the Basic Setup document. Each block is dependent on global constants and clock memory bits, and will not compile without correctly completing the initial setup. The following steps need to be performed:

- Enable system and clock memory bytes on the CPU.
- Retrieve the Open Library.
- Pull the Open Library PLC tags into the project.
- Setup Mode Control, or understand of how Open Library Modes function.

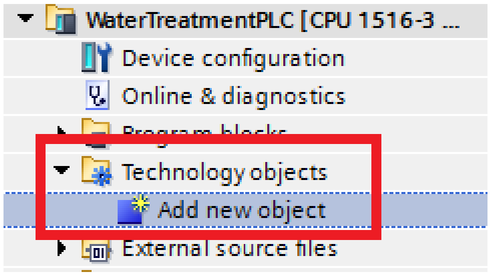
6.2. PLC

This section covers the setup required for the PLC programming. The following steps are required:

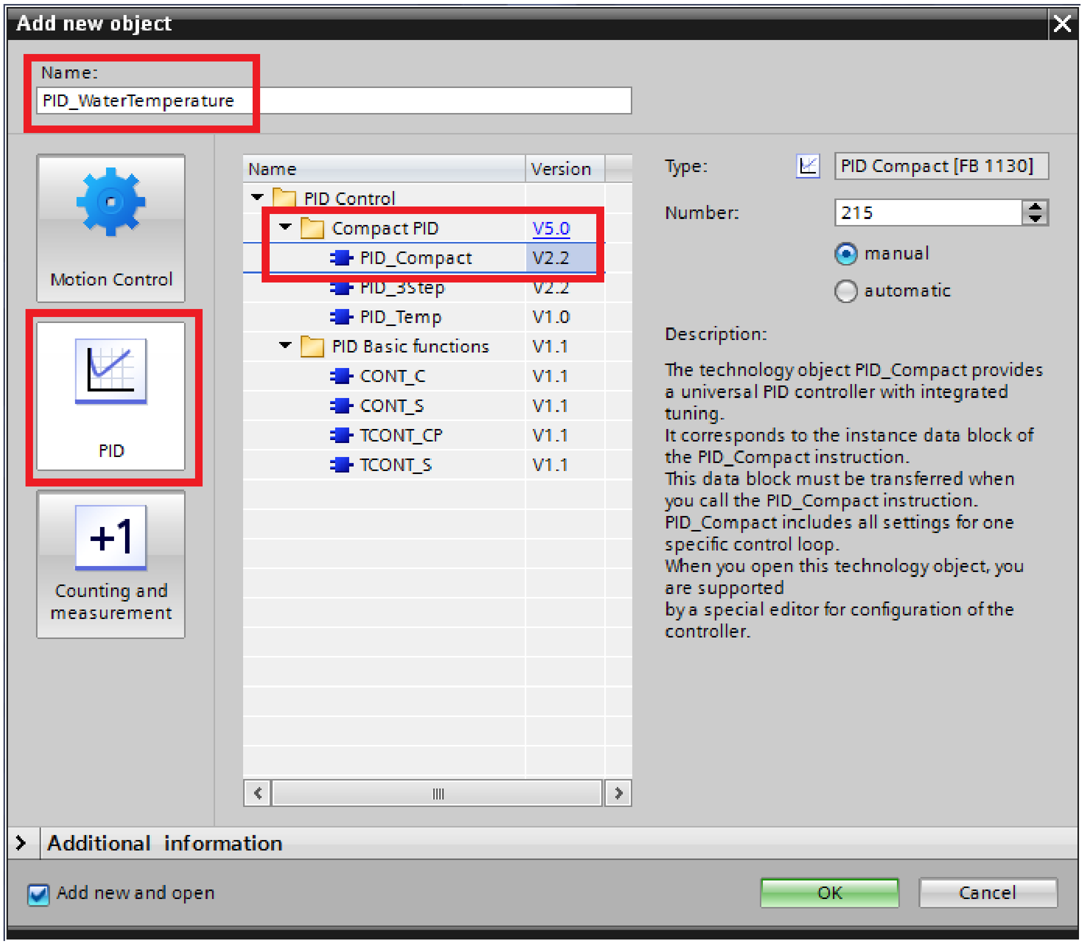
1. Create the PID Technology object
2. Create a Function Block to call the fbPID_CompactInterface block (this block is typically called in the main scan)
3. Create an FB and OB to call the PID_Compact Technology Object
4. Map the fbPID_CompactInterface to the PID_Compact Technology Object.

6.2.1. Creating PID Technology Object

1. Start by creating a PID_Compact technology object. Expand the Technology Objects folder and click Add new object.



2. In the new object dialog box, give your object a name and select the PID_Compact v2.2 block.



Name: PID_WaterTemperature

Name	Version
PID Control	
Compact PID	V5.0
PID_Compact	V2.2
PID_3Step	V2.2
PID_Temp	V1.0
PID Basic functions	V1.1
CONT_C	V1.1
CONT_S	V1.1
TCONT_CP	V1.1
TCONT_S	V1.1

Type: PID Compact [FB 1130]

Number: 215

● manual

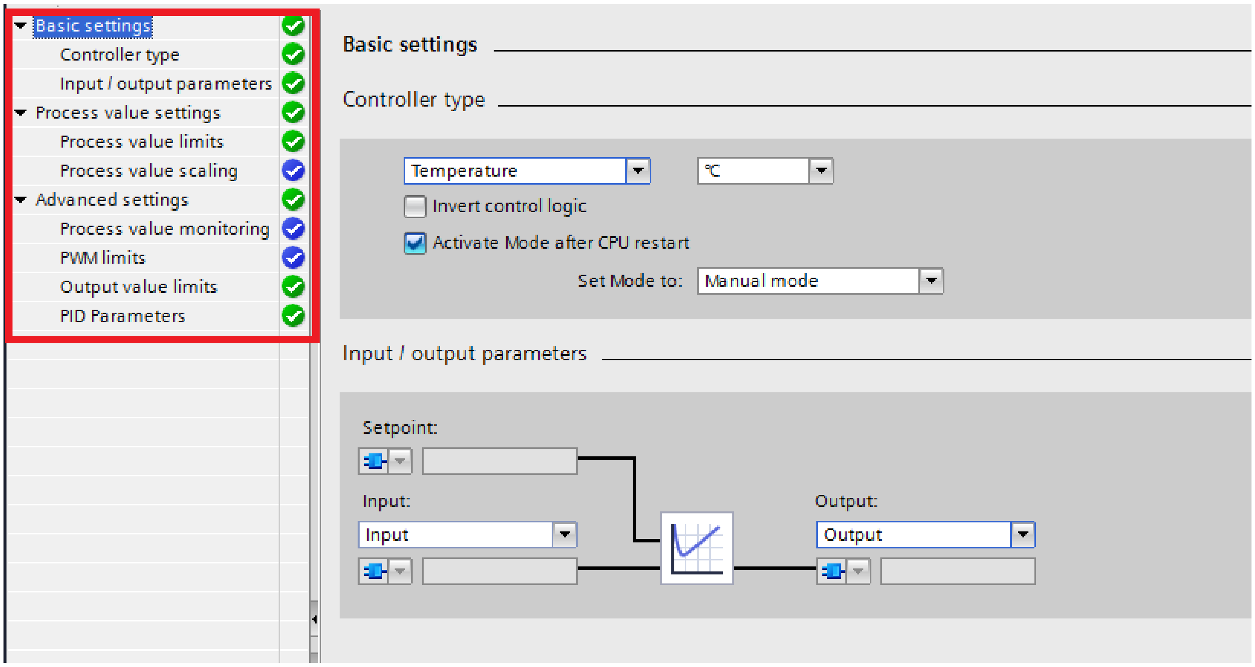
○ automatic

Description:

The technology object PID_Compact provides a universal PID controller with integrated tuning.
It corresponds to the instance data block of the PID_Compact instruction.
This data block must be transferred when you call the PID_Compact instruction.
PID_Compact includes all settings for one specific control loop.
When you open this technology object, you are supported
by a special editor for configuration of the controller.

☑ Add new and open

3. Configure your PID_Compact technology block to fit your physical system.



Controller type: Temperature ℃

☐ Invert control logic

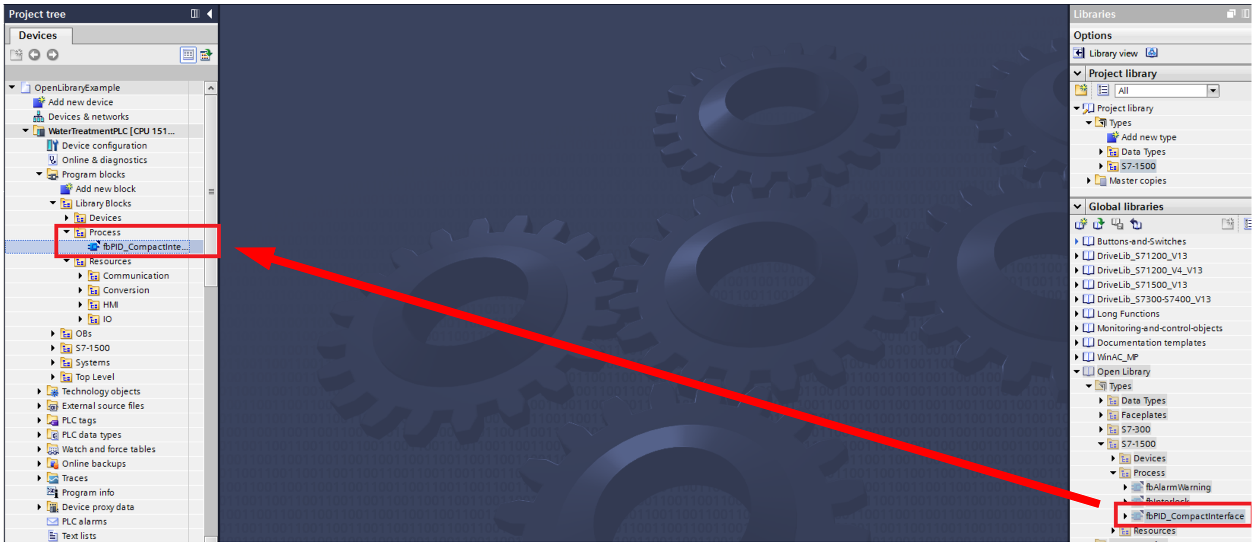
☑ Activate Mode after CPU restart

Set Mode to: Manual mode

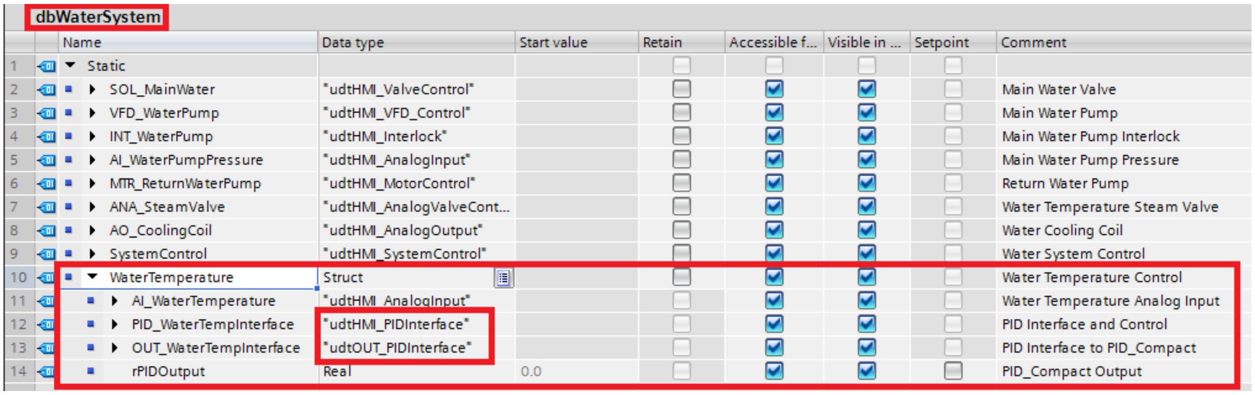
Input: Input

Output: Output

4. Pull fbPID_CompactInterface from the Open Library Types group into the Library Blocks group to use it in our project.



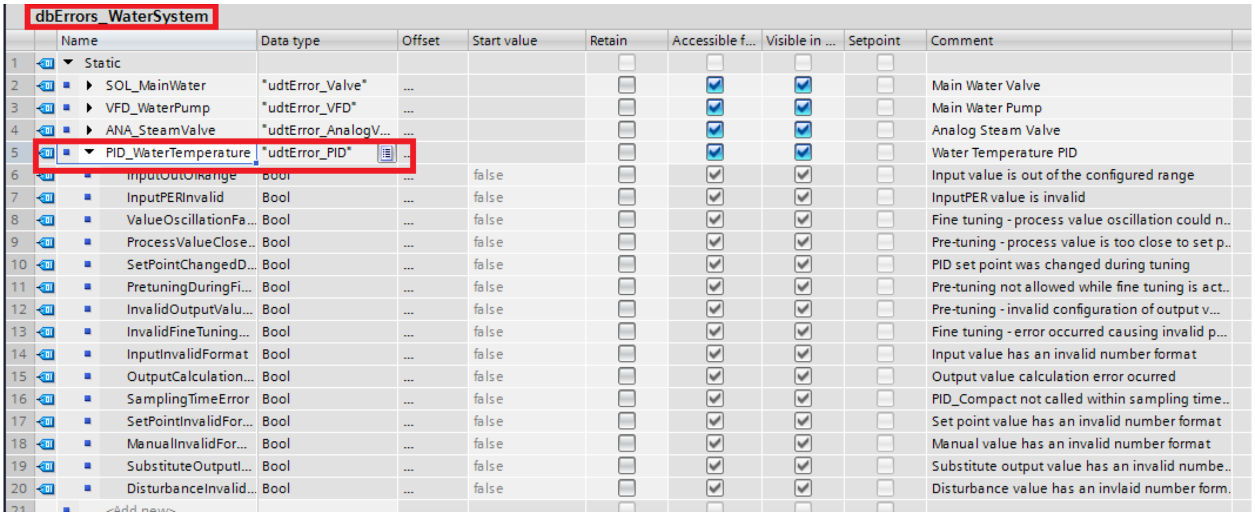
5. In a global data block (dbWaterSystem in this example), add a structure for all of your PID related variables. The required variables are seen in the screenshot below. This structure should include you HMI control for the PID system (udtHMI_PIDInterface). It should also include a structure for interfacing with the PID_Compact block (udtOUT_PIDInterface).



dbWaterSystem

	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static			☐	☐	☐	☐	
2	SOL_MainWater	"udtHMI_ValveControl"		☐	☑	☑	☐	Main Water Valve
3	VFD_WaterPump	"udtHMI_VFD_Control"		☐	☑	☑	☐	Main Water Pump
4	INT_WaterPump	"udtHMI_Interlock"		☐	☑	☑	☐	Main Water Pump Interlock
5	AI_WaterPumpPressure	"udtHMI_AnalogInput"		☐	☑	☑	☐	Main Water Pump Pressure
6	MTR_ReturnWaterPump	"udtHMI_MotorControl"		☐	☑	☑	☐	Return Water Pump
7	ANA_SteamValve	"udtHMI_AnalogValveCont...		☐	☑	☑	☐	Water Temperature Steam Valve
8	AO_CoolingCoil	"udtHMI_AnalogOutput"		☐	☑	☑	☐	Water Cooling Coil
9	SystemControl	"udtHMI_SystemControl"		☐	☑	☑	☐	Water System Control
10	WaterTemperature	Struct		☐	☑	☑	☐	Water Temperature Control
11	AI_WaterTemperature	"udtHMI_AnalogInput"		☐	☑	☑	☐	Water Temperature Analog Input
12	PID_WaterTempInterface	"udtHMI_PIDInterface"		☐	☑	☑	☐	PID Interface and Control
13	OUT_WaterTempInterface	"udtOUT_PIDInterface"		☐	☑	☑	☐	PID Interface to PID_Compact
14	rPIDOutput	Real	0.0	☐	☑	☑	☐	PID_Compact Output

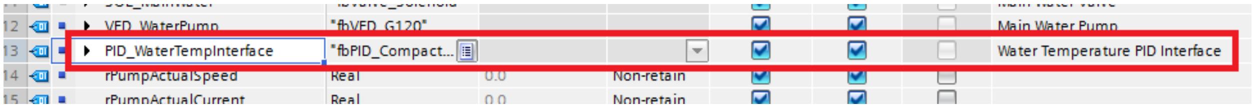
6. Similarly, add the error structure to the errors Data Block, dbErrors_WaterSystem in this example.



dbErrors_WaterSystem

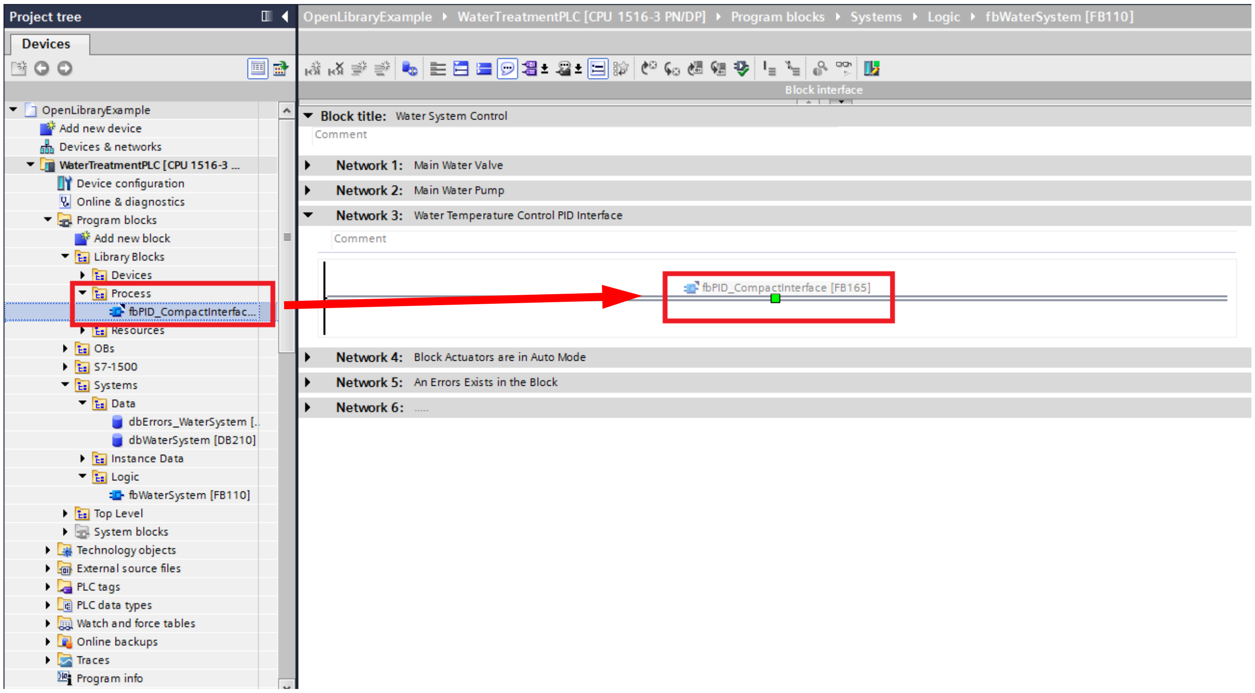
	Name	Data type	Offset	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static				☐	☐	☐	☐	
2	SOL_MainWater	"udtError_Valve"	...		☐	☑	☑	☐	Main Water Valve
3	VFD_WaterPump	"udtError_VFD"	...		☐	☑	☑	☐	Main Water Pump
4	ANA_SteamValve	"udtError_AnalogV...	...		☐	☑	☑	☐	Analog Steam Valve
5	PID_WaterTemperature	"udtError_PID"	...		☐	☑	☑	☐	Water Temperature PID
6	InputOutOfRange	Bool	...	false	☐	☑	☑	☐	Input value is out of the configured range
7	InputPERInvalid	Bool	...	false	☐	☑	☑	☐	InputPER value is invalid
8	ValueOscillationFa...	Bool	...	false	☐	☑	☑	☐	Fine tuning - process value oscillation could n...
9	ProcessValueClose...	Bool	...	false	☐	☑	☑	☐	Pre-tuning - process value is too close to set p...
10	SetPointChangedD...	Bool	...	false	☐	☑	☑	☐	PID set point was changed during tuning
11	PretuningDuringFi...	Bool	...	false	☐	☑	☑	☐	Pre-tuning not allowed while fine tuning is act...
12	InvalidOutputValu...	Bool	...	false	☐	☑	☑	☐	Pre-tuning - invalid configuration of output v...
13	InvalidFineTuning...	Bool	...	false	☐	☑	☑	☐	Fine tuning - error occurred causing invalid p...
14	InputInvalidFormat	Bool	...	false	☐	☑	☑	☐	Input value has an invalid number format
15	OutputCalculation...	Bool	...	false	☐	☑	☑	☐	Output value calculation error ocurred
16	SamplingTimeError	Bool	...	false	☐	☑	☑	☐	PID_Compact not called within sampling time...
17	SetPointInvalidFor...	Bool	...	false	☐	☑	☑	☐	Set point value has an invalid number format
18	ManualInvalidFor...	Bool	...	false	☐	☑	☑	☐	Manual value has an invalid number format
19	SubstituteOutputI...	Bool	...	false	☐	☑	☑	☐	Substitute output value has an invalid numbe...
20	DisturbanceInvalid...	Bool	...	false	☐	☑	☑	☐	Disturbance value has an invlaid number form...
21	<Add new>								

7. Add a multiple instance static memory variable for the fbPID_CompactInterface block in the fbWaterSystem main block.

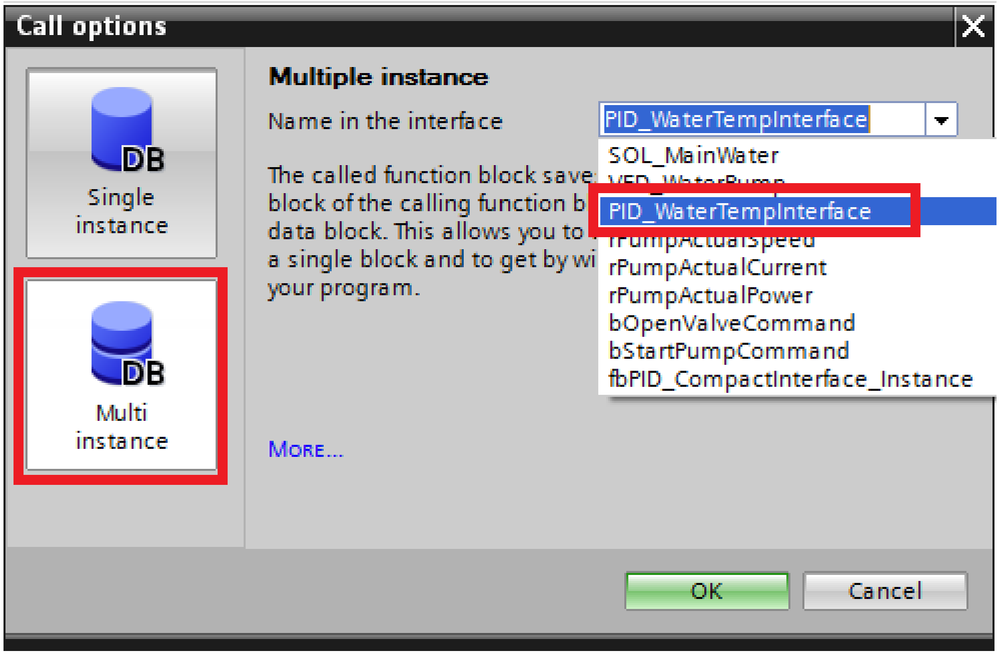


12	VFD_WaterPump	"fbVFD_G120"			☑	☑	☐	Main Water Pump
13	PID_WaterTempInterface	"fbPID_Compact...			☑	☑	☐	Water Temperature PID Interface
14	rPumpActualSpeed	Real	0.0	Non-retain	☑	☑	☐	
15	rPumpActualCurrent	Real	0.0	Non-retain	☑	☑	☐	

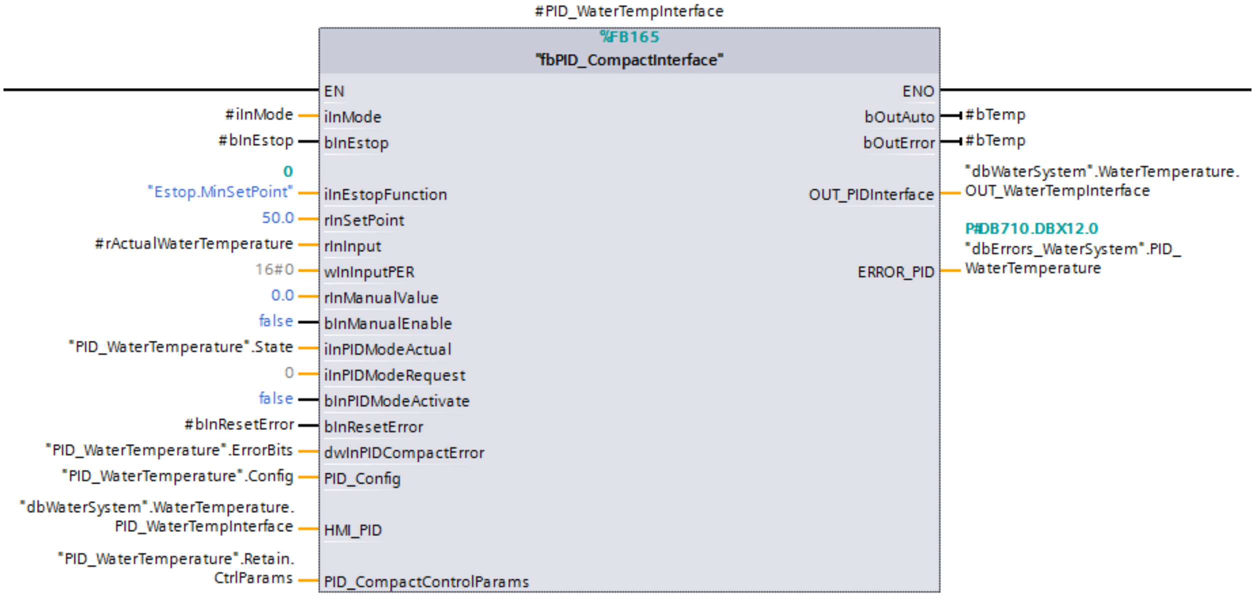
8. Drag an instance of fbPID_CompactInterface into a network in fbWaterSystem.



9. When the Call options dialog appears, choose the static memory multiple instance that you just created in the block interface.



10. Fill in the interface for your fbPID_CompactInterface instance. Notice the tag references to the PID_Compact technology data block as well as the HMI and Errors data block structures.

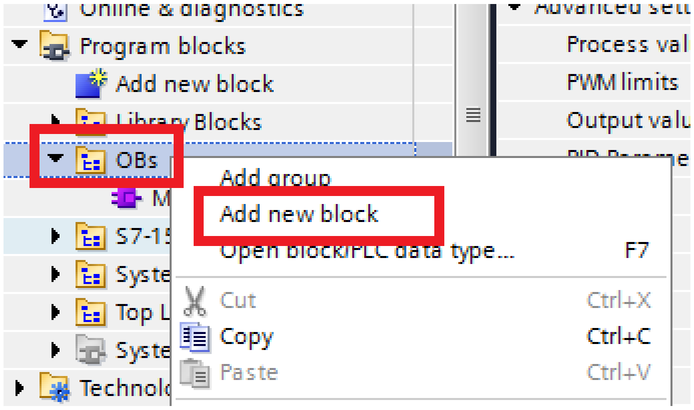


#PID_WaterTempInterface
%FB165
"fbPID_CompactInterface"

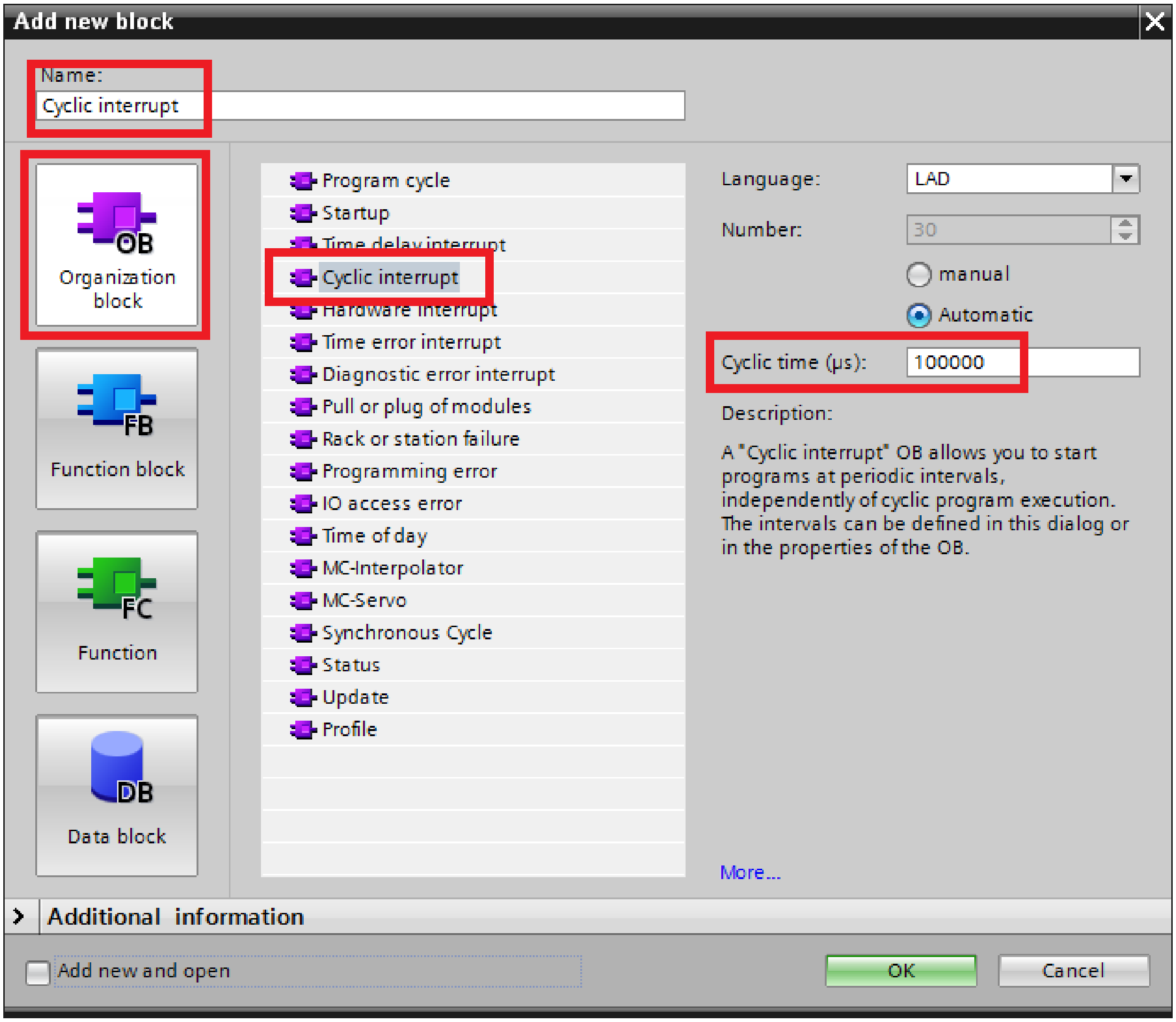
	EN
#iInMode	iInMode
#bInEstop	bInEstop
0 "Estop.MinSetPoint"	iInEstopFunction
50.0	rInSetPoint
#rActualWaterTemperature	rInInput
16#0	wInInputPER
0.0	rInManualValue
false	bInManualEnable
"PID_WaterTemperature".State	iInPIDModeActual
0	iInPIDModeRequest
false	bInPIDModeActivate
#bInResetError	bInResetError
"PID_WaterTemperature".ErrorBits	dwInPIDCompactError
"PID_WaterTemperature".Config	PID_Config
"dbWaterSystem".WaterTemperature.PID_WaterTempInterface	HMI_PID
"PID_WaterTemperature".Retain.CtrlParams	PID_CompactControlParams

ENO	
bOutAuto	#bTemp
bOutError	#bTemp
OUT_PIDInterface	"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface
ERROR_PID	P#DB710.DBX12.0 "dbErrors_WaterSystem".PID_WaterTemperature

11. Add a new Cyclic Interrupt OB that will call our time sensitive cyclic operations.



12. Give the cyclic interrupt OB a name and set the cycle time to something appropriate for the application. In this case, temperature does not need a fast cycle so it was set to 100ms (100,000µs).



Name: Cyclic interrupt

Language: LAD

Number: 30

○ manual

● Automatic

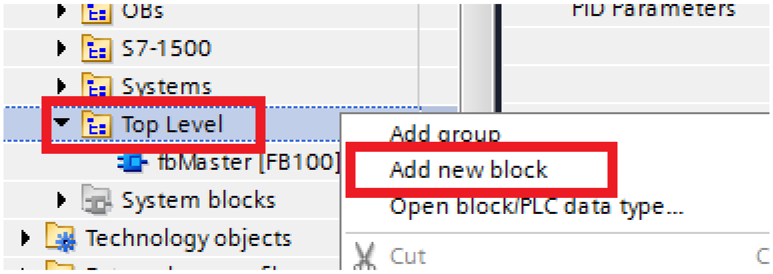
Cyclic time (µs): 100000

Description:

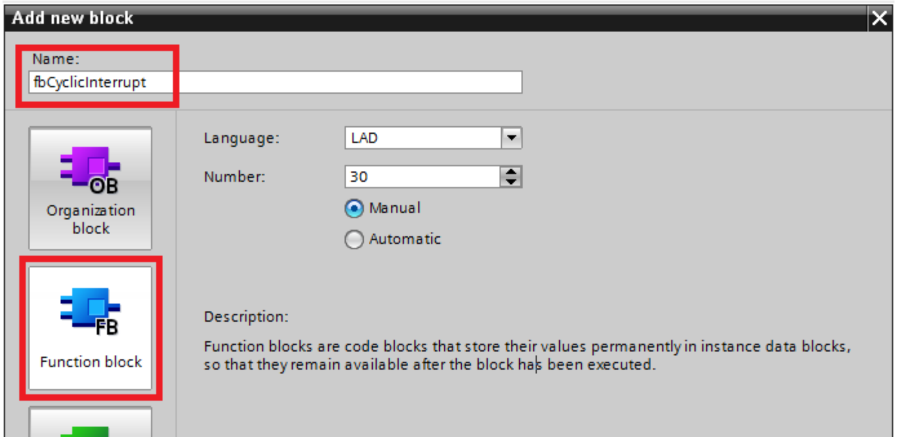
A "Cyclic interrupt" OB allows you to start programs at periodic intervals, independently of cyclic program execution. The intervals can be defined in this dialog or in the properties of the OB.

☐ Add new and open

13. Add a new function block that will call our PID_Compact and will be called from the cyclic interrupt OB. It is recommended best practice to limit logic inside an OB, and the library object requires instance memory, so we will use the FB to encapsulate our cyclic operation logic.



14. Name the cyclic function block.



Name: fbCyclicInterrupt

Language: LAD

Number: 30

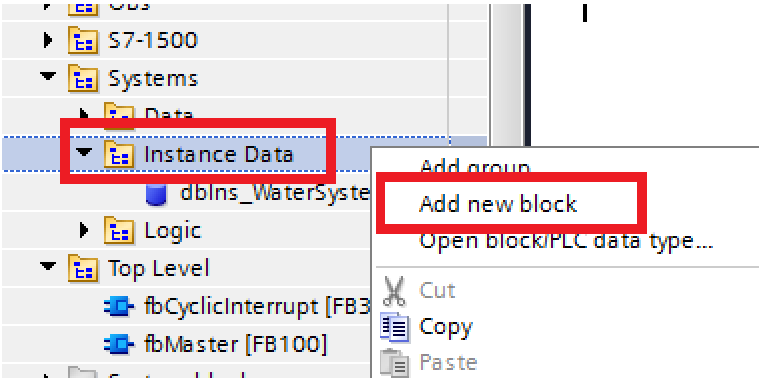
● Manual

○ Automatic

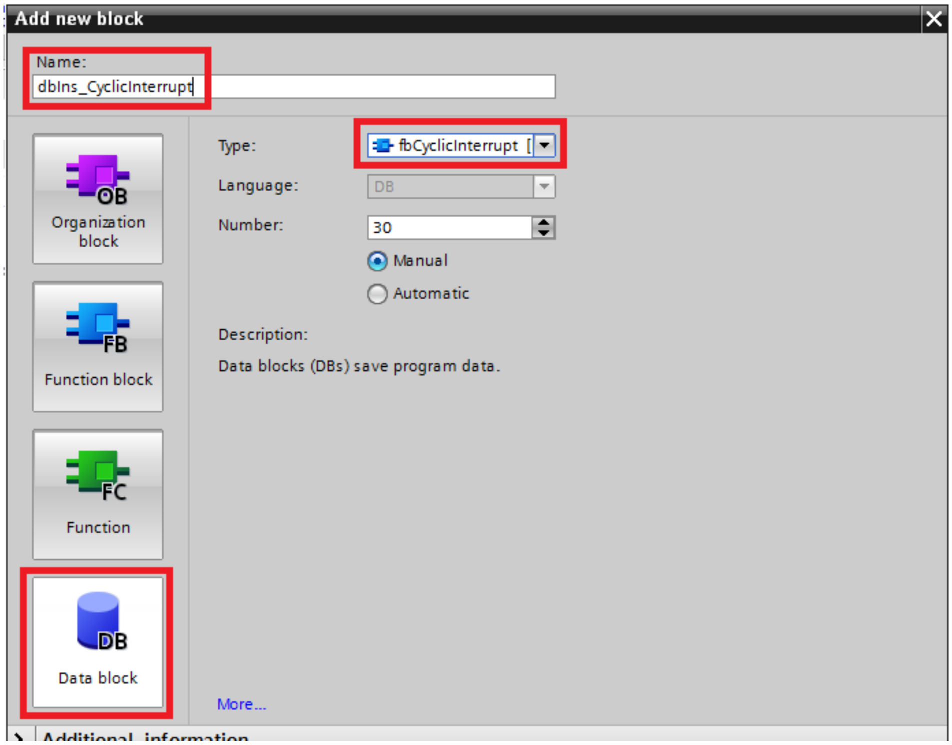
Description:

Function blocks are code blocks that store their values permanently in instance data blocks, so that they remain available after the block has been executed.

15. Create an instance data block to accompany the cyclic function block.



16. Name the instance data block and give it the type of your cyclic function block.



Add new block

Name: dbIns_CyclicInterrupt

Type: fbCyclicInterrupt

Language: DB

Number: 30

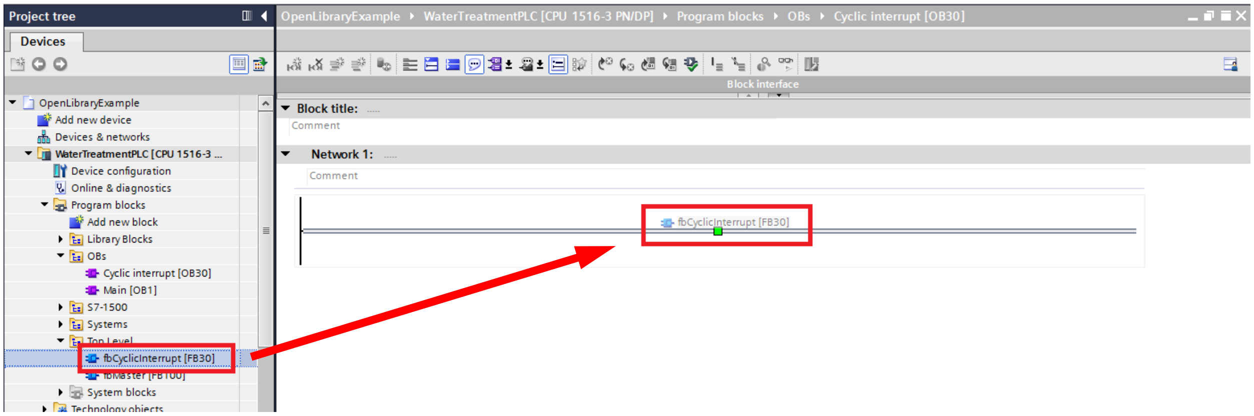
● Manual

○ Automatic

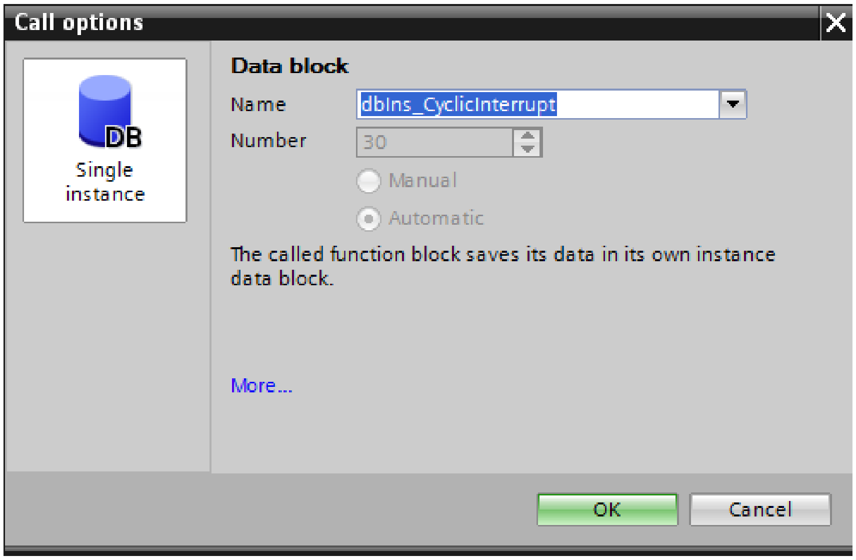
Description:

Data blocks (DBs) save program data.

17. Inside of the cyclic interrupt OB, drag in an instance of the cyclic FB you created.



18. Assign the instance data block as a single instance.



Call options

Single instance

Data block

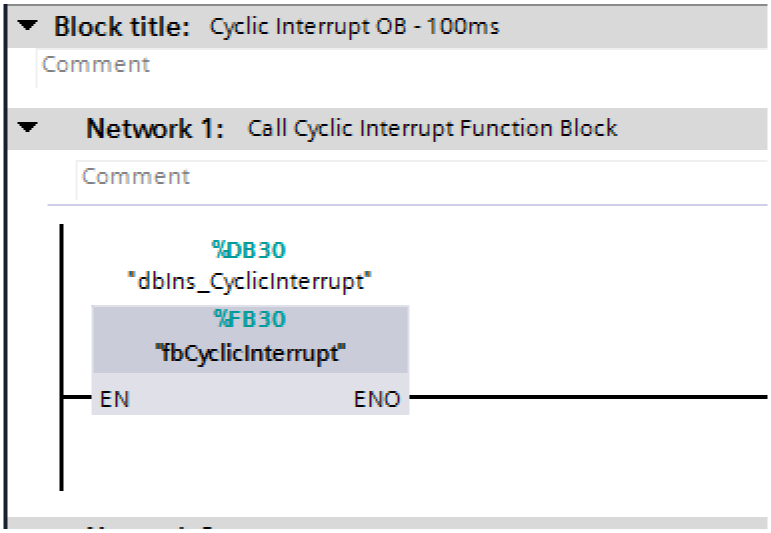
Name: dbIns_CyclicInterrupt

Number: 30

○ Manual

● Automatic

The called function block saves its data in its own instance data block.



Block title: Cyclic Interrupt OB - 100ms

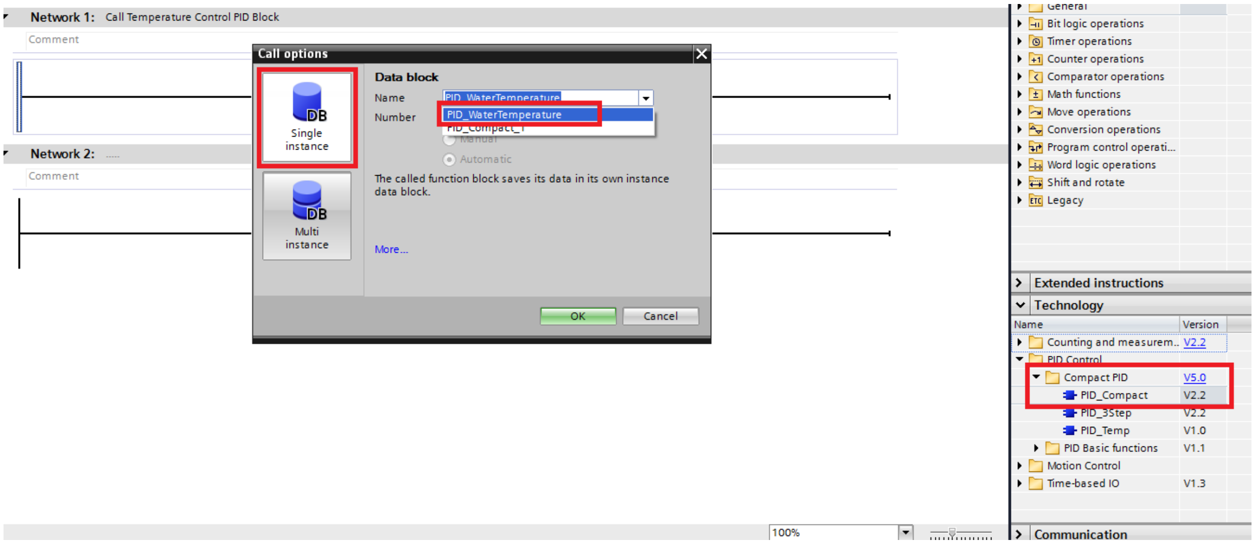
Comment

Network 1: Call Cyclic Interrupt Function Block

Comment

%DB30
"dbIns_CyclicInterrupt"
%FB30
"fbCyclicInterrupt"

19. Inside of your cyclic FB, drag in an instance of PID_Compact from the Siemens library. Assign the single instance to the data block of your technology object.



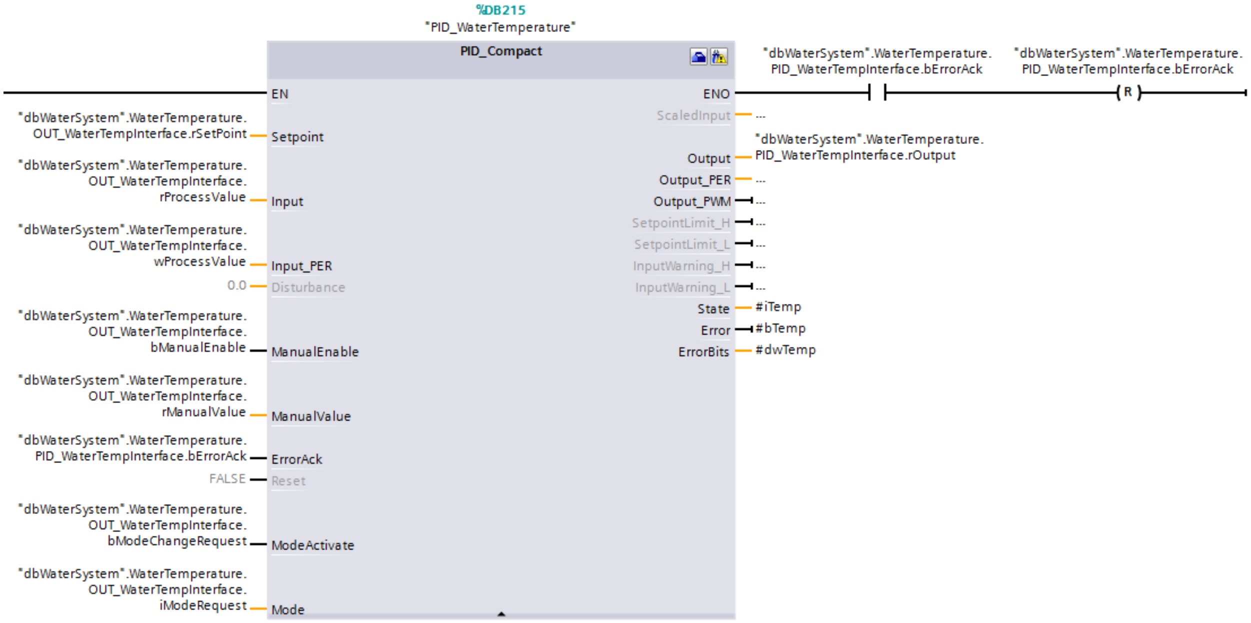
Network 1: Call Temperature Control PID Block

Data block

Name: PID_WaterTemperature

The called function block saves its data in its own instance data block.

20. Fill in the interface of the PID_Compact block with the dbWaterSystem variables created earlier.



%DB215
"PID_WaterTemperature"
PID_Compact

	EN
"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface.rSetPoint	Setpoint
"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface.rProcessValue	Input
"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface.wProcessValue	Input_PER
0.0	Disturbance
"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface.bManualEnable	ManualEnable
"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface.rManualValue	ManualValue
"dbWaterSystem".WaterTemperature.PID_WaterTempInterface.bErrorAck	ErrorAck
FALSE	Reset
"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface.bModeChangeRequest	ModeActivate
"dbWaterSystem".WaterTemperature.OUT_WaterTempInterface.iModeRequest	Mode

ENO	
ScaledInput	...
Output	"dbWaterSystem".WaterTemperature.PID_WaterTempInterface.rOutput
Output_PER	...
Output_PWM	...
SetpointLimit_H	...
SetpointLimit_L	...
InputWarning_H	...
InputWarning_L	...
State	#iTemp
Error	#bTemp
ErrorBits	#dwTemp

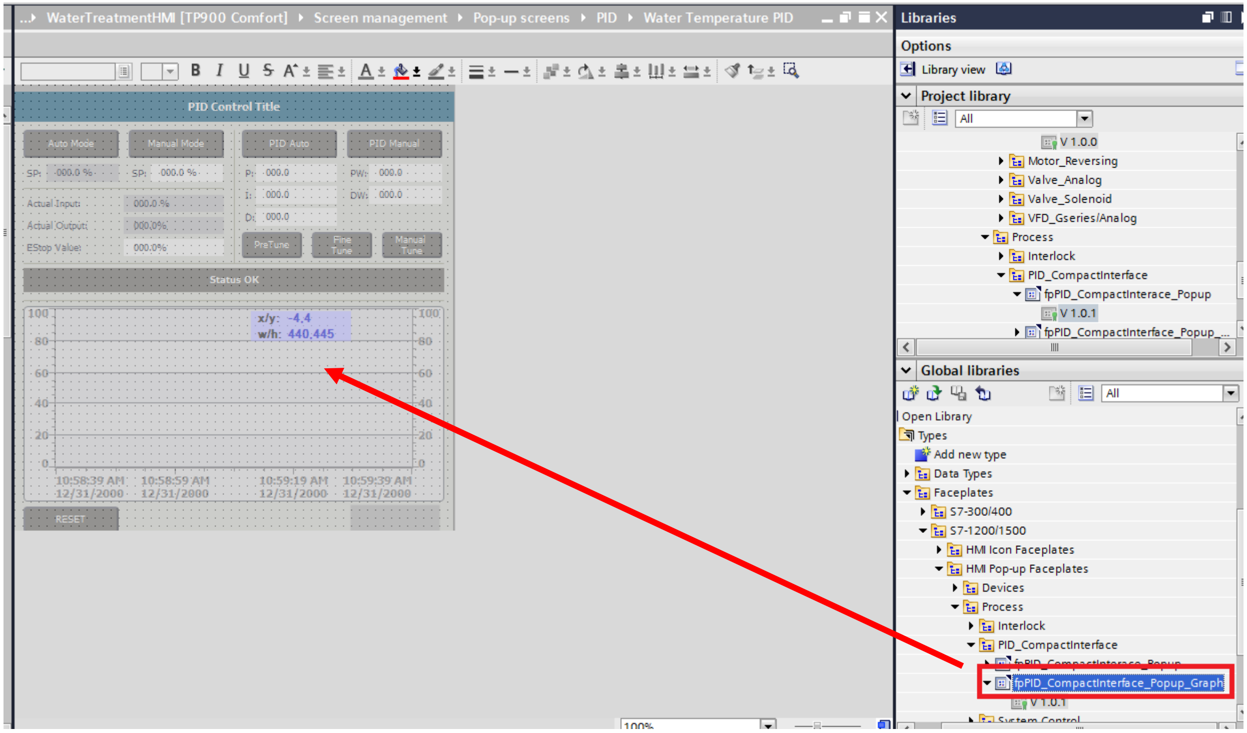
"dbWaterSystem".WaterTemperature.PID_WaterTempInterface.bErrorAck

"dbWaterSystem".WaterTemperature.PID_WaterTempInterface.bErrorAck

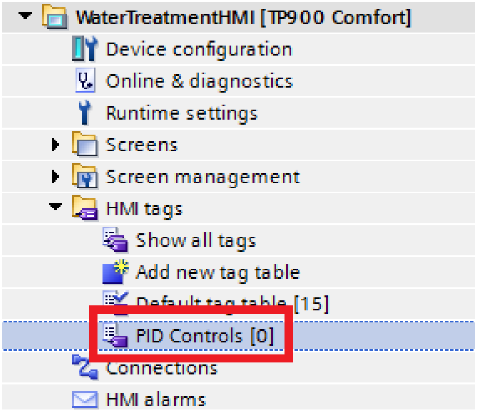
6.3. HMI

This section walks through the required steps to add the HMI pop-ups for the PID Interface Block.

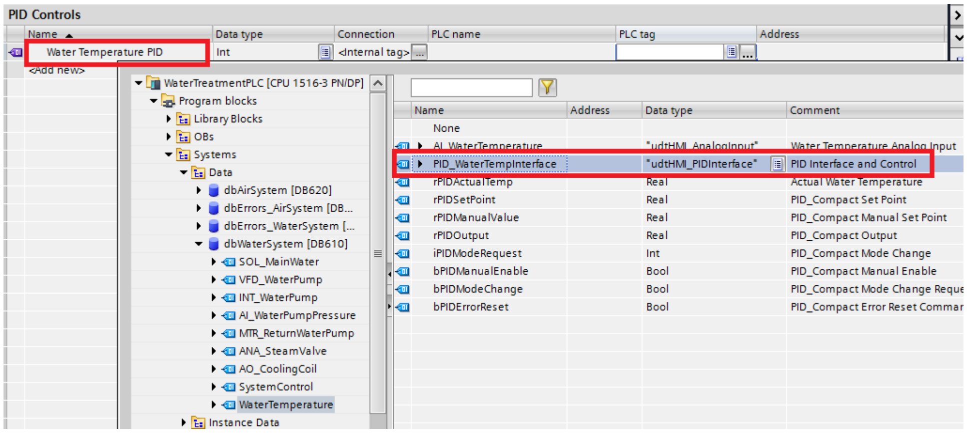
1. Create a new Pop-up screen for the PID Faceplate.
2. Pull the fpPID_CompactInterface_Popup_Graph into the new Pop-up screen.



3. Add a new HMI tag table called PID Controls.



4. Create the Water Temperature PID tag in the new table and map it to "dbWaterSystem".WaterTemperature.PID_WaterTemperature.

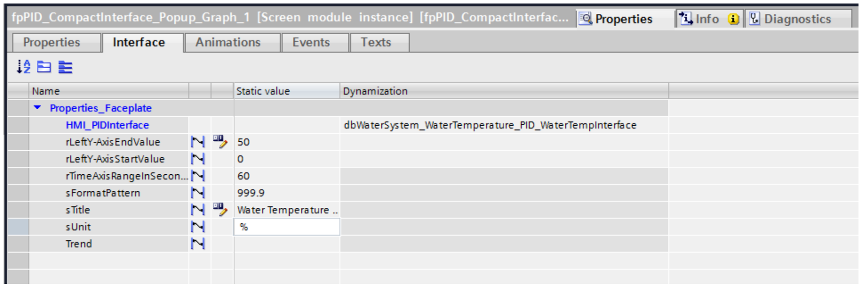


PID Controls

Name	Data type	Connection	PLC name	PLC tag	Address
Water Temperature PID	Int	<Internal tag>			
<Add new>					

Name	Address	Data type	Comment
None			
AI_WaterTemperature		"udtHMI_AnalogInput"	Water Temperature Analog Input
PID_WaterTempInterface		"udtHMI_PIDInterface"	PID Interface and Control
rPIDActualTemp		Real	Actual Water Temperature
rPIDSetPoint		Real	PID_Compact Set Point
rPIDManualValue		Real	PID_Compact Manual Set Point
rPIDOutput		Real	PID_Compact Output
iPIDModeRequest		Int	PID_Compact Mode Change
bPIDManualEnable		Bool	PID_Compact Manual Enable
bPIDModeChange		Bool	PID_Compact Mode Change Reque
bPIDErrorReset		Bool	PID_Compact Error Reset Commar

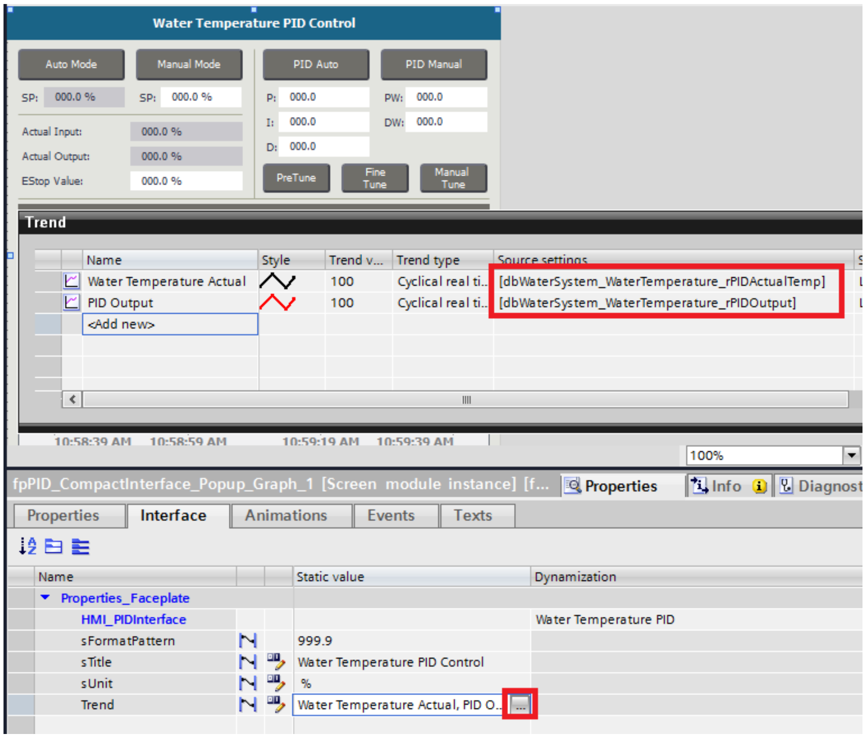
5. Fill in the interface of the popup with the newly created tag as well as static values.



fpPID_CompactInterface_Popup_Graph_1 [Screen module instance] [fpPID_CompactInterfac...
Properties
Info
Diagnostics
Properties
Interface
Animations
Events
Texts

Name	Static value	Dynamization
Properties_Faceplate		
HMI_PIDInterface		dbWaterSystem_WaterTemperature_PID_WaterTempInterface
rLeftY-AxisEndValue	50	
rLeftY-AxisStartValue	0	
rTimeAxisRangeInSecon...	60	
sFormatPattern	999.9	
sTitle	Water Temperature ..	
sUnit	%	
Trend		

6. Lastly, configure the Trend view with the tags you would like to trend.



Trend

Name	Style	Trend v...	Trend type	Source settings
Water Temperature Actual		100	Cyclical real ti...	[dbWaterSystem_WaterTemperature_rPIDActualTemp]
PID Output		100	Cyclical real ti...	[dbWaterSystem_WaterTemperature_rPIDOutput]
<Add new>				

Name	Static value	Dynamization
Properties_Faceplate		
HMI_PIDInterface		Water Temperature PID
sFormatPattern	999.9	
sTitle	Water Temperature PID Control	
sUnit	%	
Trend	Water Temperature Actual, PID O...	