



---

# SIEMENS OPEN LIBRARY

---

6 – PID Configuration  
OCTOBER 11, 2016

---

## Contents

1. Purpose .....	3
2. Intended Use.....	3
3. Revision History.....	3
4. Open Library License.....	3
5. Hardware and Software Compatibility .....	4
6. Open Library PID Compact Setup.....	5
6.1. Initial Setup .....	5
6.2. PLC .....	5
6.2.1. Creating PID Technology Object .....	5
6.3. HMI.....	17

---

## 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 Compact 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	2016-06-20	DMC	Updated screenshots for PID_Compact and the interface function block
1.2	2016-08-23	DMC	No Changes
1.3	2016-10-11	DMC	No Changes

---

## 4. Open Library License

Copyright (c) 2016 DMC, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 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.

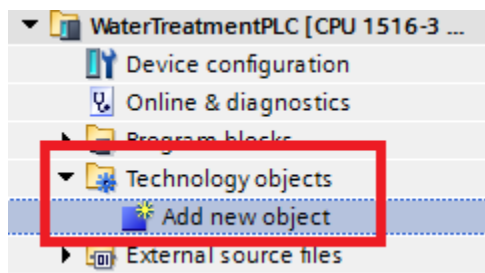
### 6.2. PLC

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

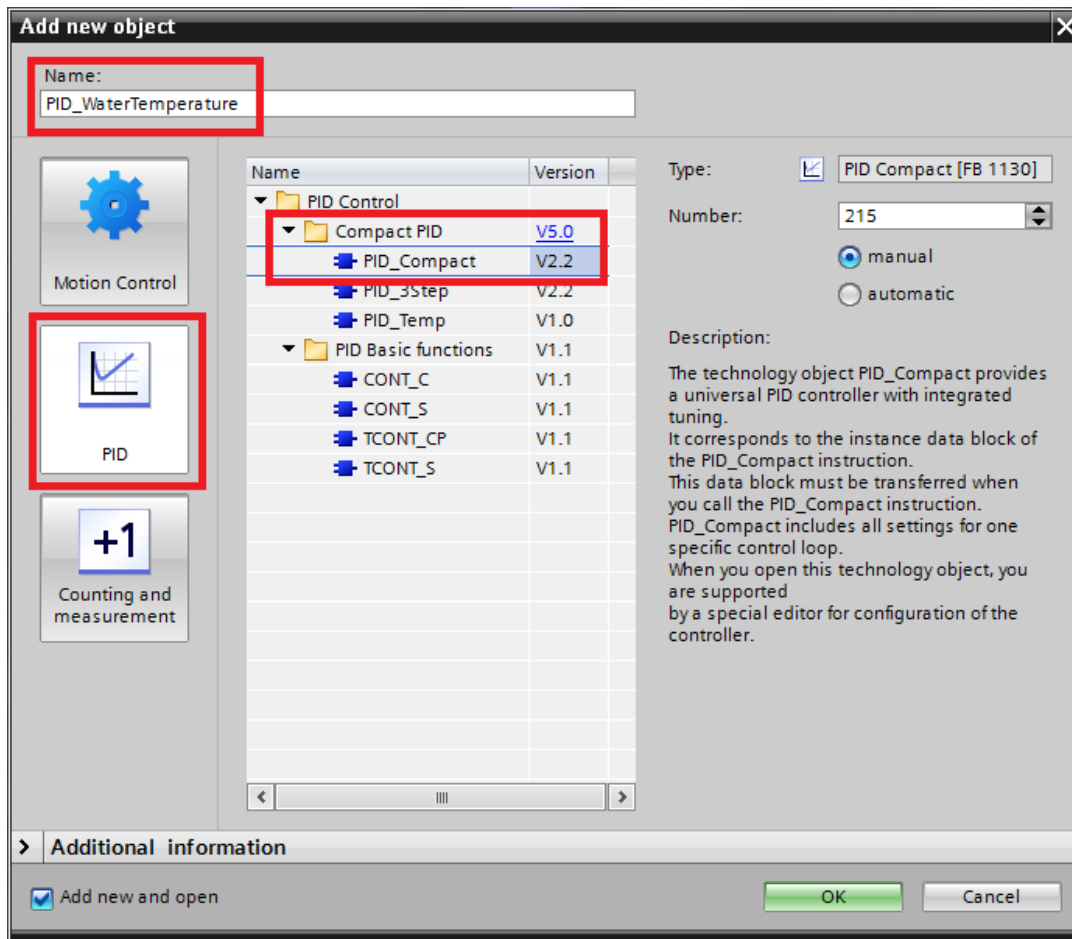
1. Create the PID Technology object
2. Create a Function Block to call the fbPID\_CompactInterface block (this block is typically called in the main scan)
3. Create an FB and OB to call the PID\_Compact Technology Object
4. Map the fbPID\_CompactInterface to the PID\_Compact Technology Object.

#### 6.2.1. Creating PID Technology Object

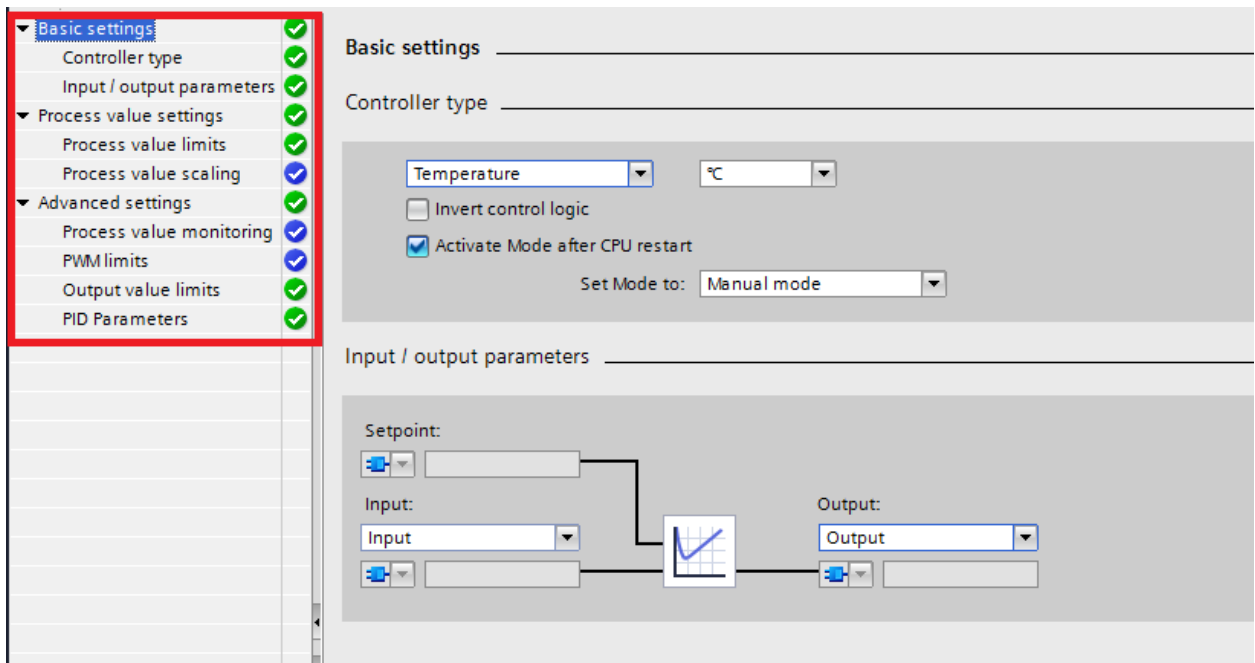
1. Start by creating a PID\_Compact technology object. Expand the Technology Objects folder and click Add new object.



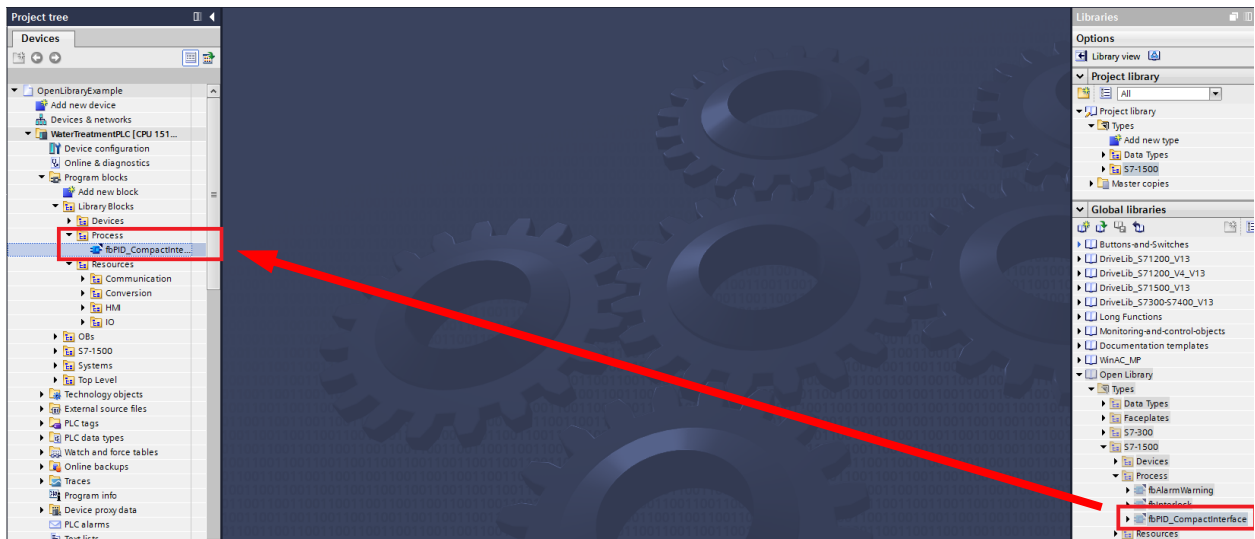
2. In the new object dialog box, give your object a name and select the PID\_Compact v2.2 block.



3. Configure your PID\_Compact technology block to fit your physical system.



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



- 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"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Main Water Valve	
3	VFD_WaterPump	*udtHMI_VFD_Control"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Main Water Pump	
4	INT_WaterPump	*udtHMI_Interlock"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Main Water Pump Interlock	
5	AI_WaterPumpPressure	*udtHMI_AnalogInput"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Main Water Pump Pressure	
6	MTR_ReturnWaterPump	*udtHMI_MotorControl"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Return Water Pump	
7	ANA_SteamValve	*udtHMI_AnalogValveCont...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Water Temperature Steam Valve	
8	AO_CoolingCoil	*udtHMI_AnalogOutput"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Water Cooling Coil	
9	SystemControl	*udtHMI_SystemControl"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Water System Control	
10	WaterTemperature	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Water Temperature Control	
11	AI_WaterTemperature	*udtHMI_AnalogInput"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Water Temperature Analog Input	
12	PID_WaterTempInterface	*udtHMI_PIDInterface"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		PID Interface and Control	
13	OUT_WaterTempInterface	*udtOUT_PIDInterface"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		PID Interface to PID_Compact	
14	rPIDOutput	Real	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		PID_Compact Output	

- Similarly, add the error structure to the errors Data Block, dbErrors\_WaterSystem in this example.

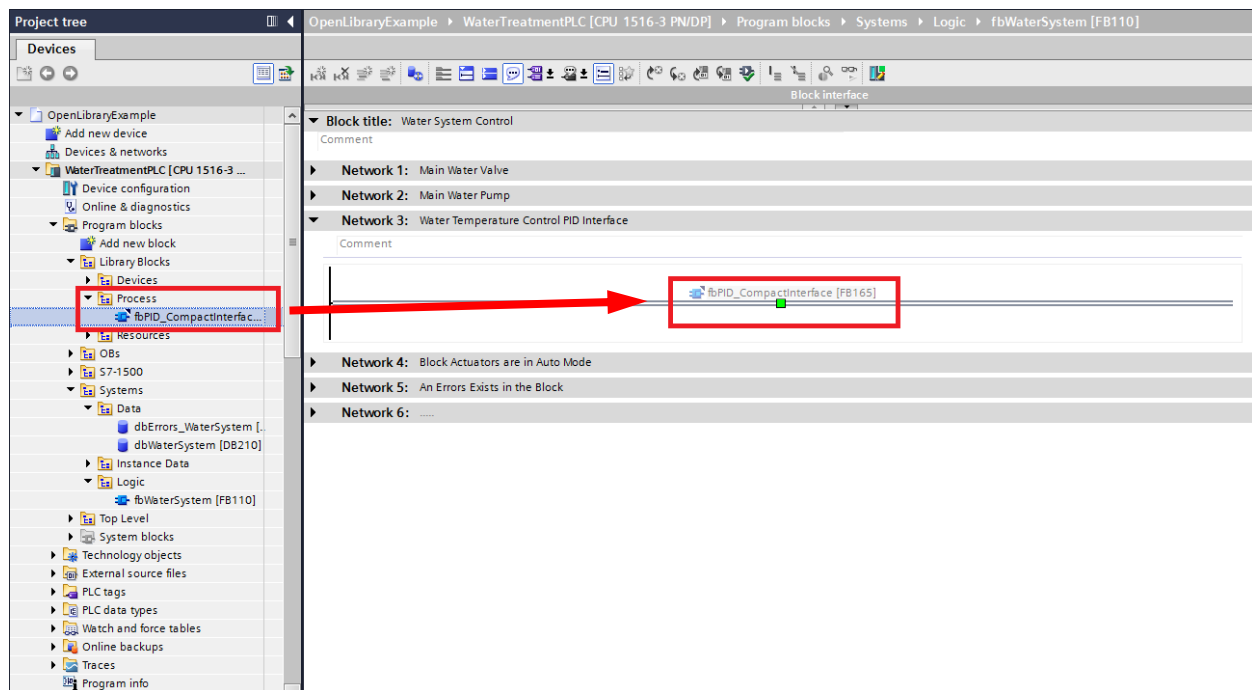
dbErrors_WaterSystem									
	Name	Data type	Offset	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static								
2	SOL_MainWater	*udtError_Valve"	...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Main Water Valve
3	VFD_WaterPump	*udtError_VFD"	...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

- Add a multiple instance static memory variable for the fbPID\_CompactInterface block in the fbWaterSystem main block.

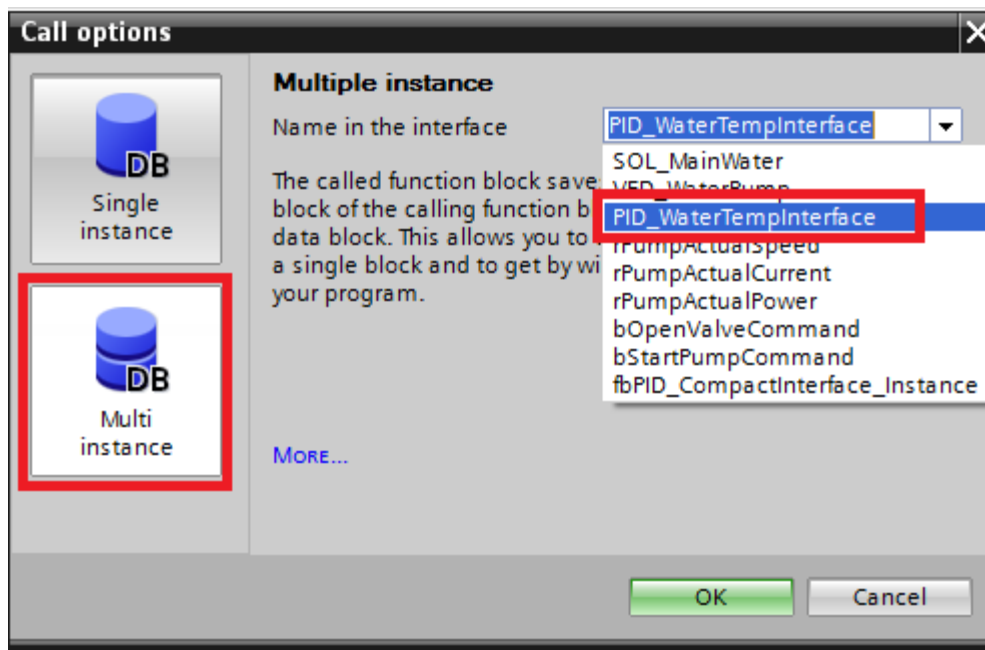
12	VFD_WaterPump	*fbVFD_G120"				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Main Water Pump
13	PID_WaterTempInterface	*fbPID_Compact...				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Water Temperature PID Interface
14	rPumpActualSpeed	Real	0.0	Non-retain		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
15	rPumpActualCurrent	Real	0.0	Non-retain		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		



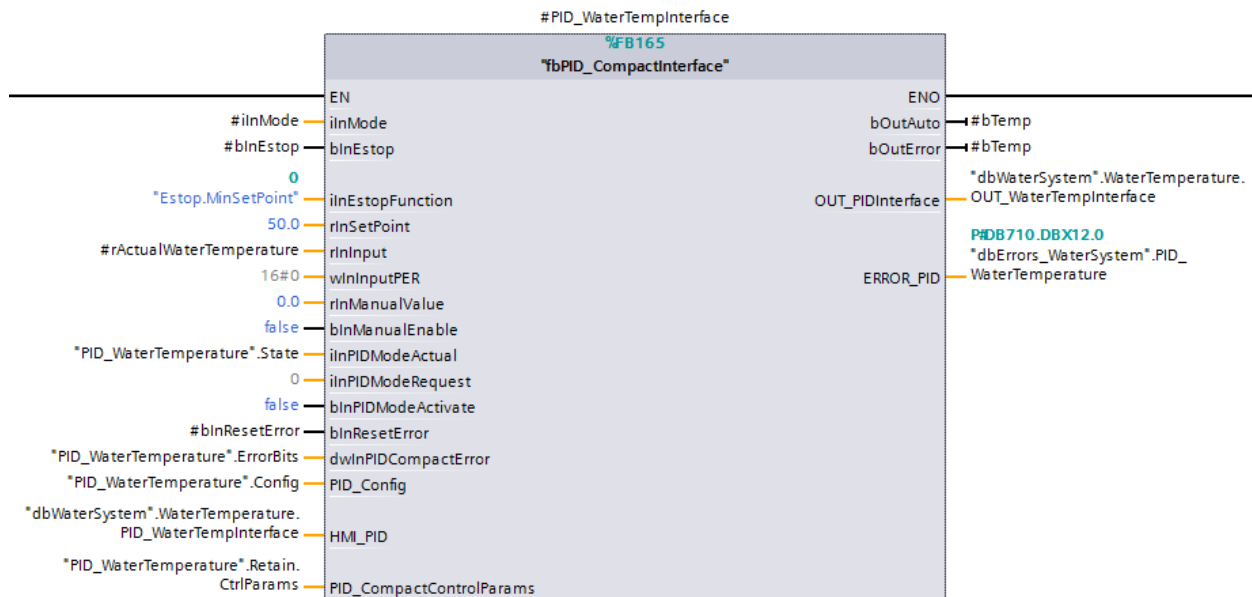
8. Drag an instance of fbPID\_CompactInterface into a network in fbWaterSystem.



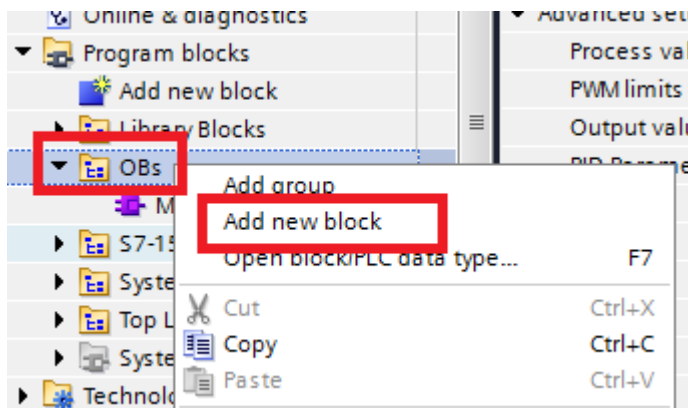
9. When the Call options dialog appears, choose the static memory multiple instance that you just created in the block interface.



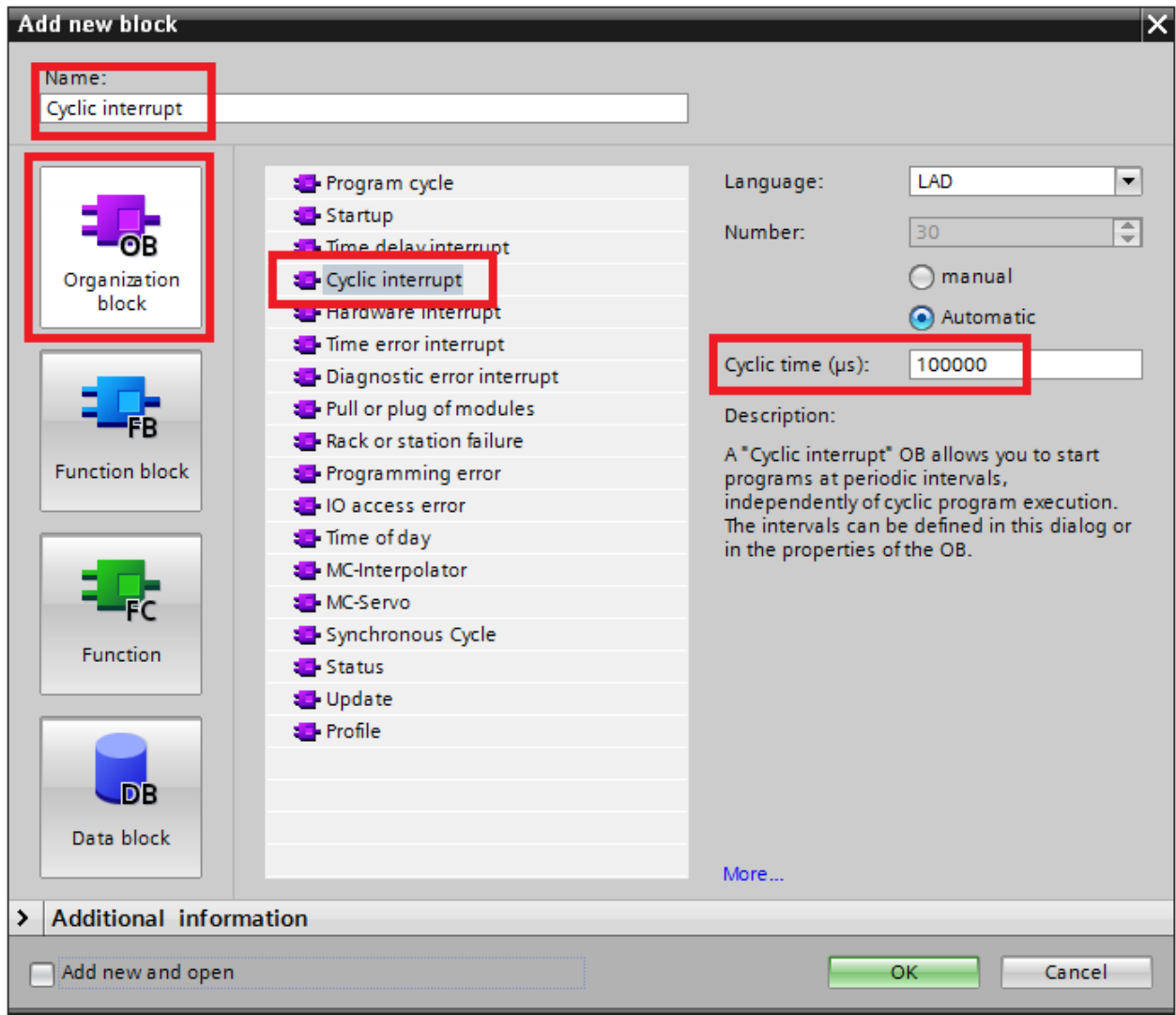
- Fill in the interface for your fbPID\_CompactInterface instance. Notice the tag references to the PID\_Compact technology data block as well as the HMI and Errors data block structures.



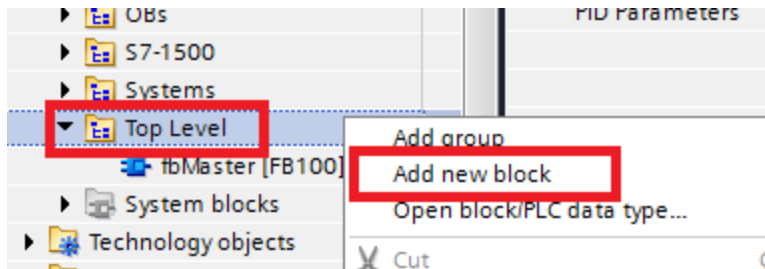
- Add a new Cyclic Interrupt OB that will call our time sensitive cyclic operations.



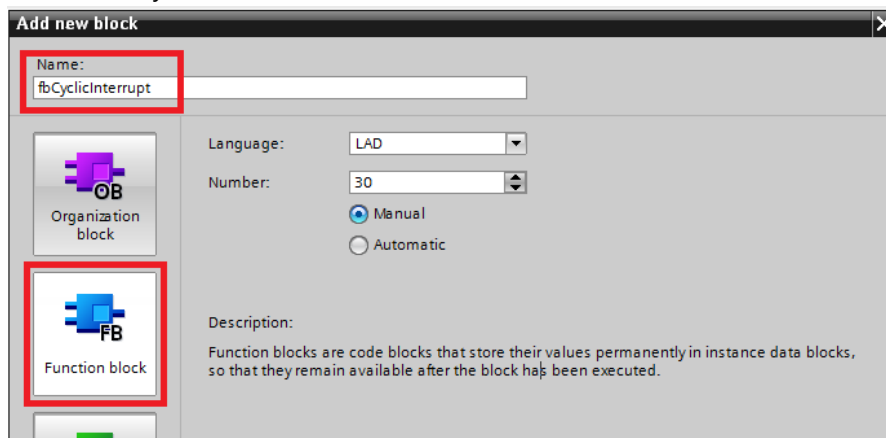
12. Give the cyclic interrupt OB a name and set the cycle time to something appropriate for the application. In this case, temperature does not need a fast cycle so it was set to 100ms (100,000µs).



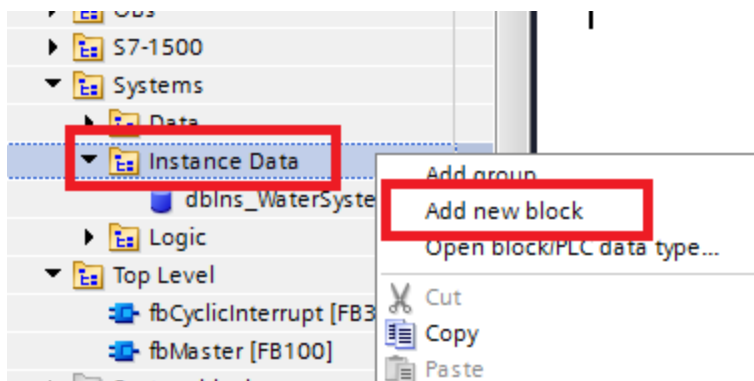
13. Add a new function block that will call our PID\_Compact and will be called from the cyclic interrupt OB. It is recommended best practice to limit logic inside an OB, and the library object requires instance memory, so we will use the FB to encapsulate our cyclic operation logic.



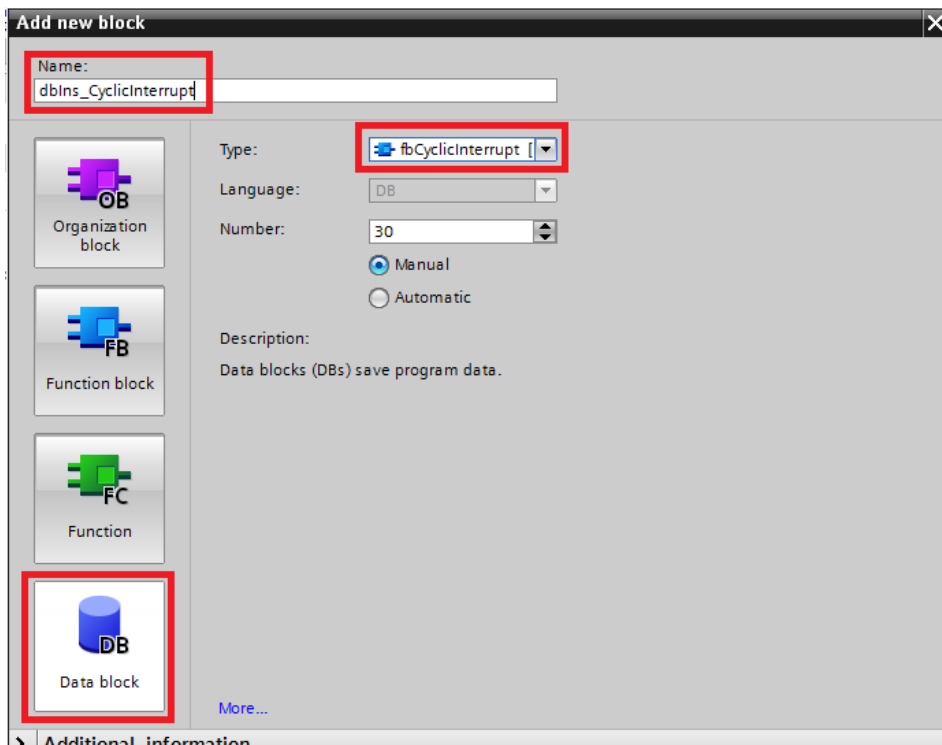
14. Name the cyclic function block.



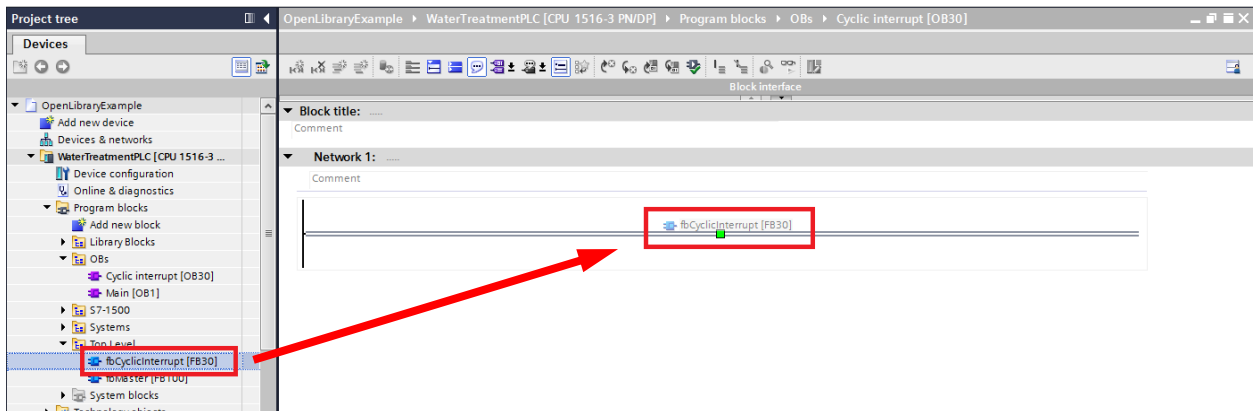
15. Create an instance data block to accompany the cyclic function block.



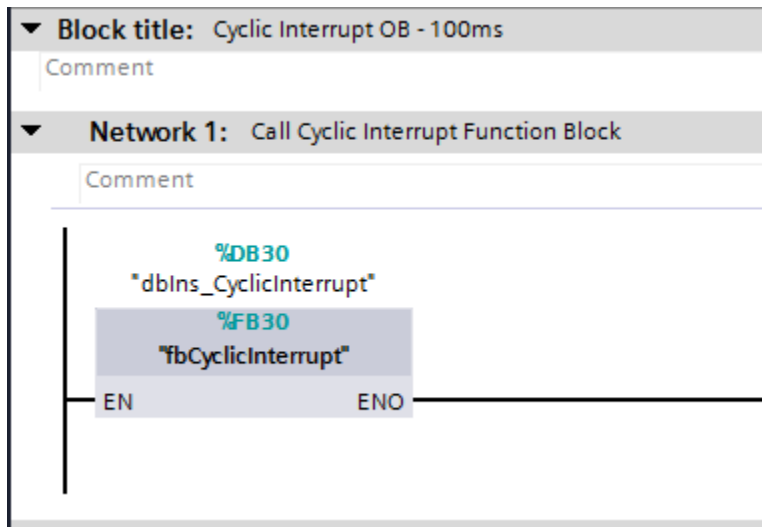
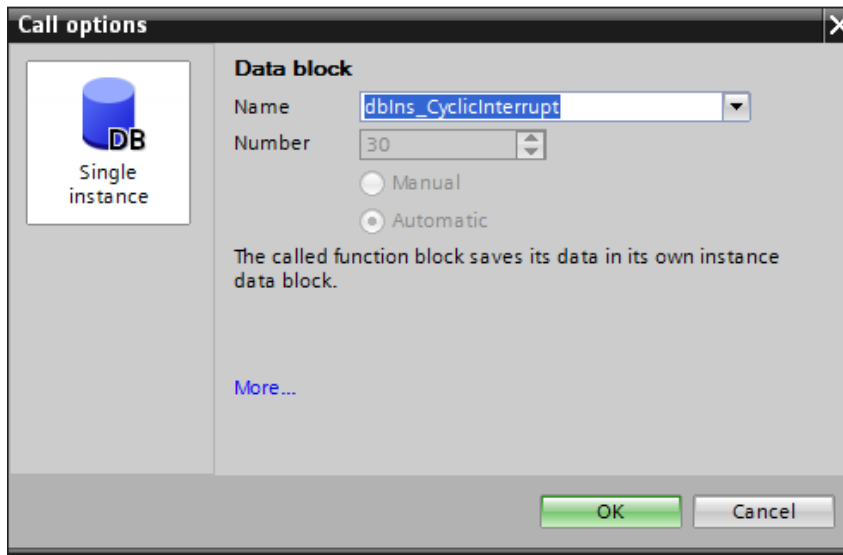
16. Name the instance data block and give it the type of your cyclic function block.



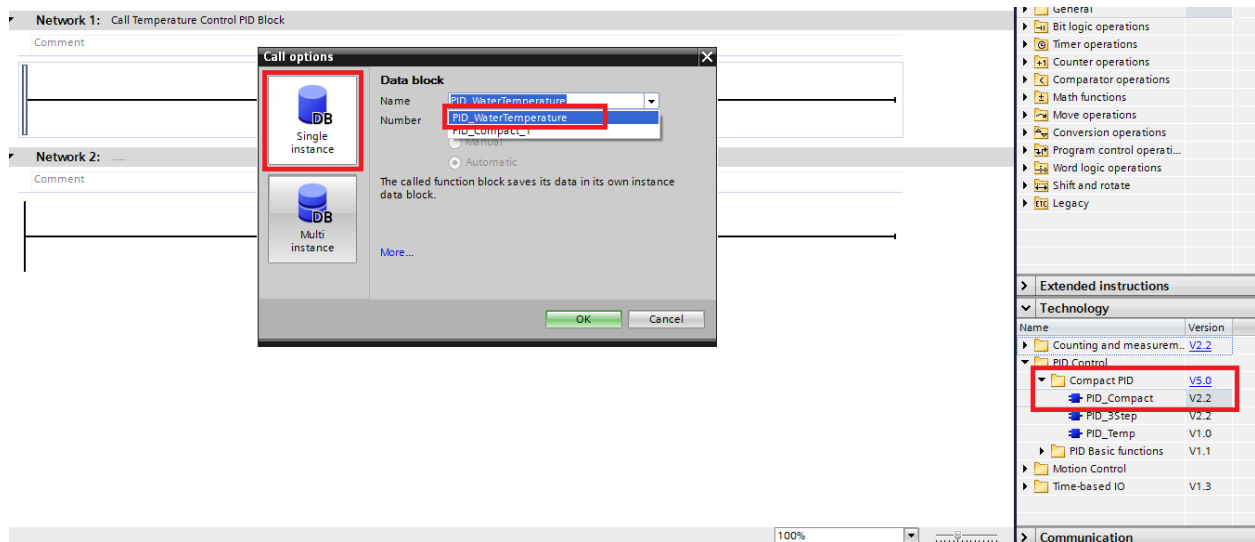
17. Inside of the cyclic interrupt OB, drag in an instance of the cyclic FB you created.



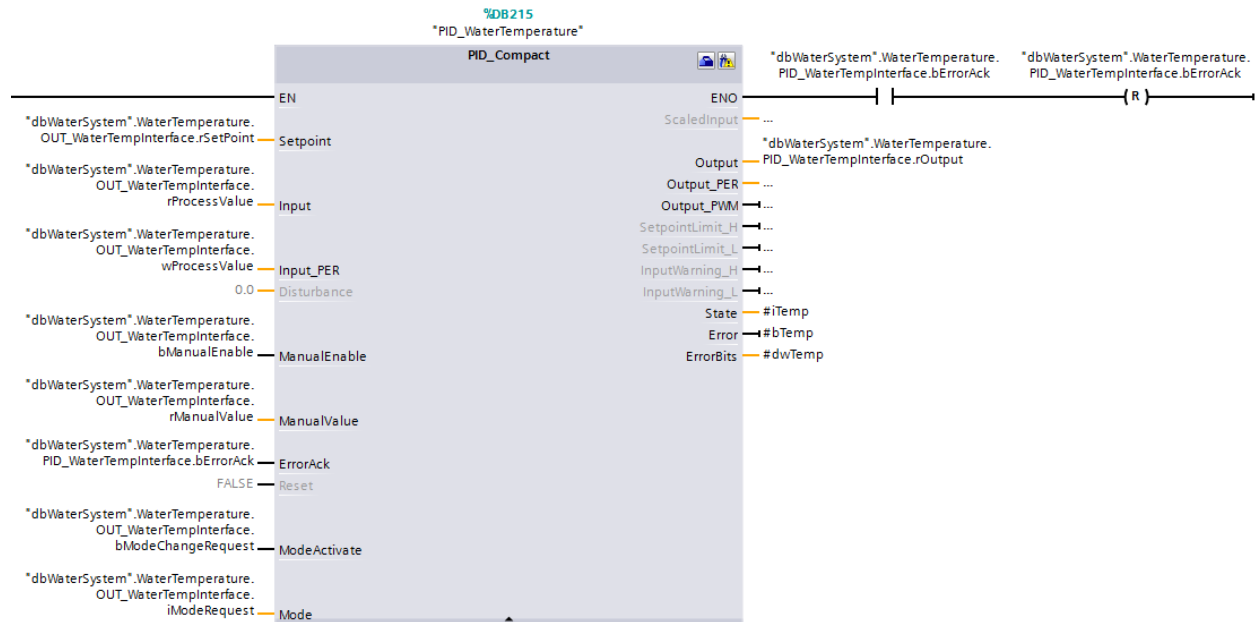
18. Assign the instance data block as a single instance.



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.



20. Fill in the interface of the PID\_Compact block with the dbWaterSystem variables created earlier.

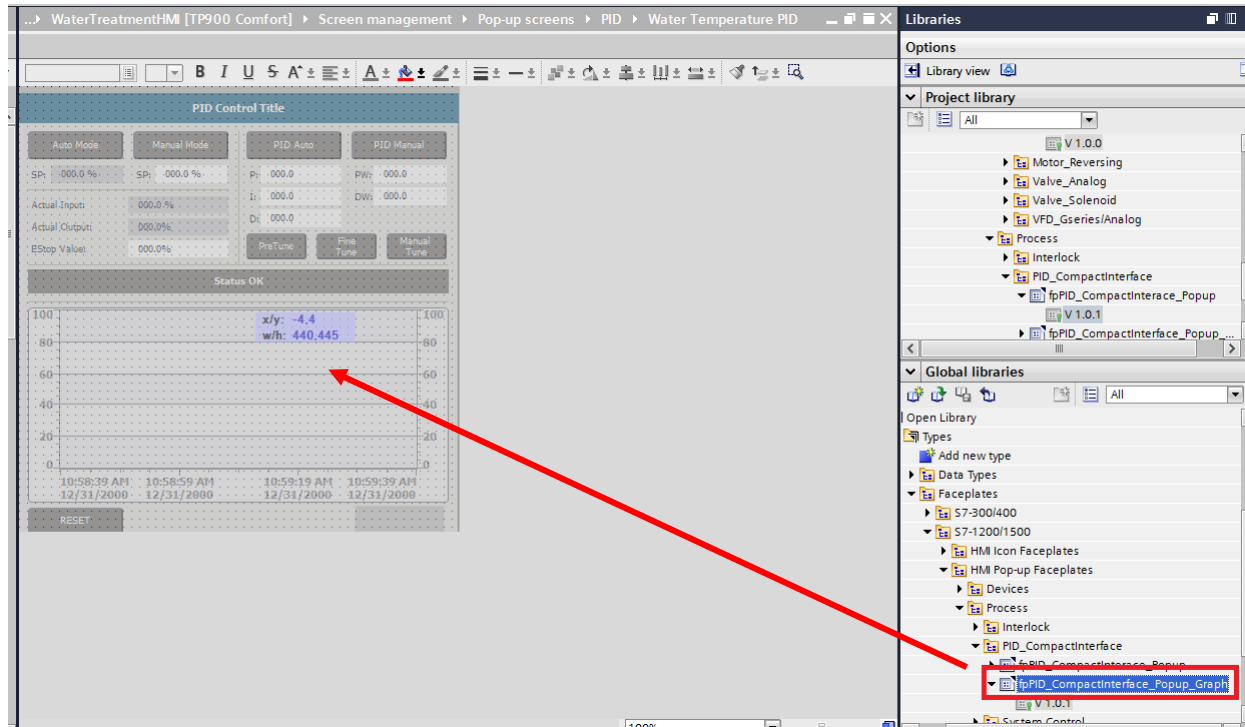




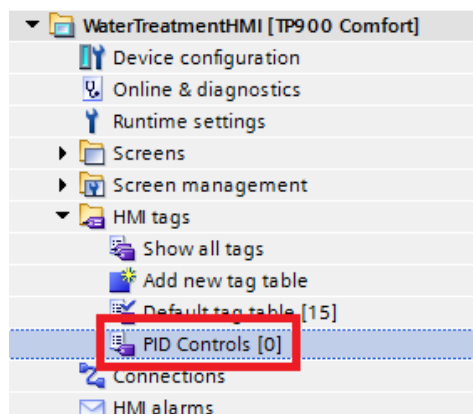
## 6.3. HMI

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

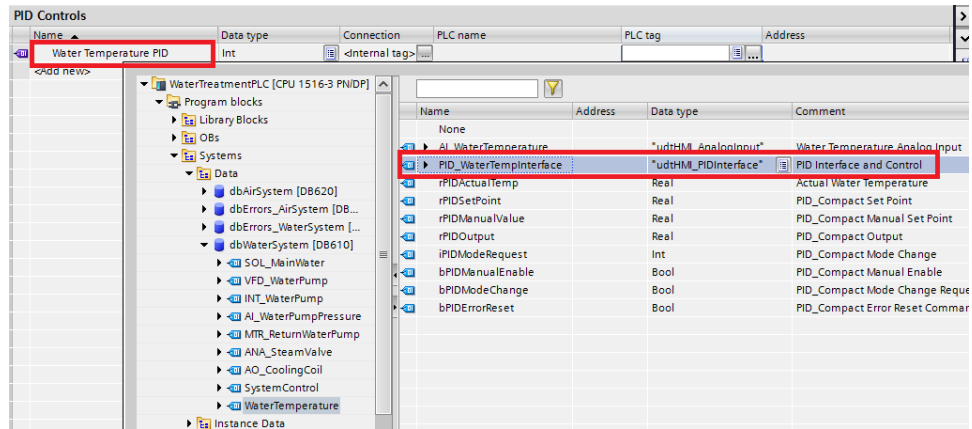
1. Create a new Pop-up screen for the PID Faceplate.
2. Pull the fpPID\_CompactInterface\_Popup\_Graph into the new Pop-up screen.



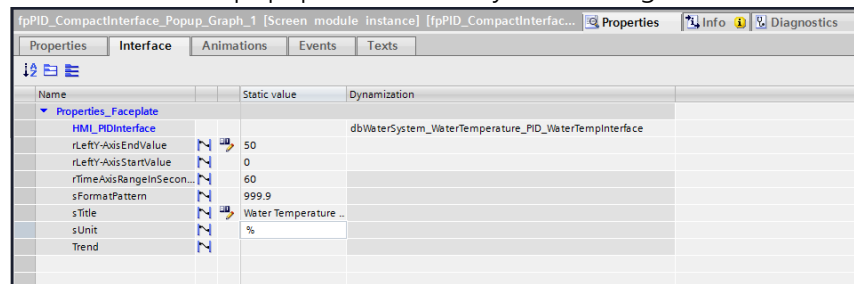
3. Add a new HMI tag table called PID Controls.



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



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



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

