Smart People. Expert Solutions.

SIEMENS OPEN LIBRARY

4 – Detailed Blocks Overview

OCTOBER 11, 2016

Contents

1. Purpose

This document details the inputs, outputs, user defined types, and functionality of the Siemens Open Library Function Blocks and HMI Faceplates. It should be used as a reference when using or configuring any of the blocks.

2. Intended Use

This document is intended to be used by anyone utilizing the Open Library for PLC and HMI Development. This document should be used after reviewing the following documents:

1. 1- Siemens Open Library – Library Overview and Architecture
2. 2- Siemens Open Library – Initial Setup
3. 3- Siemens Open Library – Example Object Configuration

3. Revision History

Version	Date	Author	Comments
1.0	2016-05-23	DMC	Initial Release
1.1	2016-06-20	DMC	Updates to Faceplates, function blocks, and datatypes
1.2	2016-08-23	DMC	Updating information regarding S7-1200 use of the GSeries VFD.
1.3	2016-10-11	DMC	No Changes

4. Open Library License

5. Hardware and Software Compatibility

This library was developed in TIA Portal V13 SP1. It has been 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.

6. Section Overview

This section provides a general overview of the library objects and what it contains.

6.1. Function Block

Each library object will have a Function block associated with it. This function block will have various inputs and outputs for the control of the device. Each block has a unique interface which will be outlined below. If the functionality of the block does not meet the requirements of your system, then it can be used as a starting point for developing custom libraries.

6.2. HMI_*Devicetype* User Defined Type

Each function block will have an In/Out UDT for the HMI interface and will be named HMI_*DeviceType*. This UDT serves as the interface with the HMI and can be used to control logic in the PLC. Each function block is connected to a specific version of the UDT, so if items are added to the UDT for custom development, the Function Block and HMI objects will need to be updated to reference the new UDT version.

A Global Data Block variable will need to be created with this UDT for each instance of the device type function block.

6.3. ERROR_*Devicetype* User Defined Type

Each function block that has errors will pass out a UDT named ERROR_*DeviceType*. A Global Data Block variable will need to be created with this UDT for each instance of the device type function block. Error UDT variables are intended to be grouped with other error structures and errors so that the data block can be used by an Excel macro to automatically generate alarms for the HMI. Errors inside this UDT should be used for appropriate PLC logic, and is the easiest way to interface to the alarms from a given device.

6.4. HMI Icon Faceplate

Most objects have an associated Icon Faceplate in the library. The Icon faceplate provides basic information on the main HMI screen and will contain images of objects like valves, pumps, and motors. Many of these objects utilize the built-in WinCC Symbol Library, so if different appearances are required for an application, it is quick to duplicate the type, rename it, and change the visual appearance of the object. Please note that the appearance for new objects will need to be set to 'Shaded' or 'Solid' in order for the color animations to function.

6.5. HMI Pop-up Faceplate

Each object has an associated Pop-up faceplate in the library. Each device in the programmed system will need to be configured with a separate Pop-up to prevent the need to multiplex tags. In addition to pop-up screens, these faceplates can be displayed on any screen desired. Custom versions of the pop-ups can also be created to add or remove functionality or diagnostic information as desired.

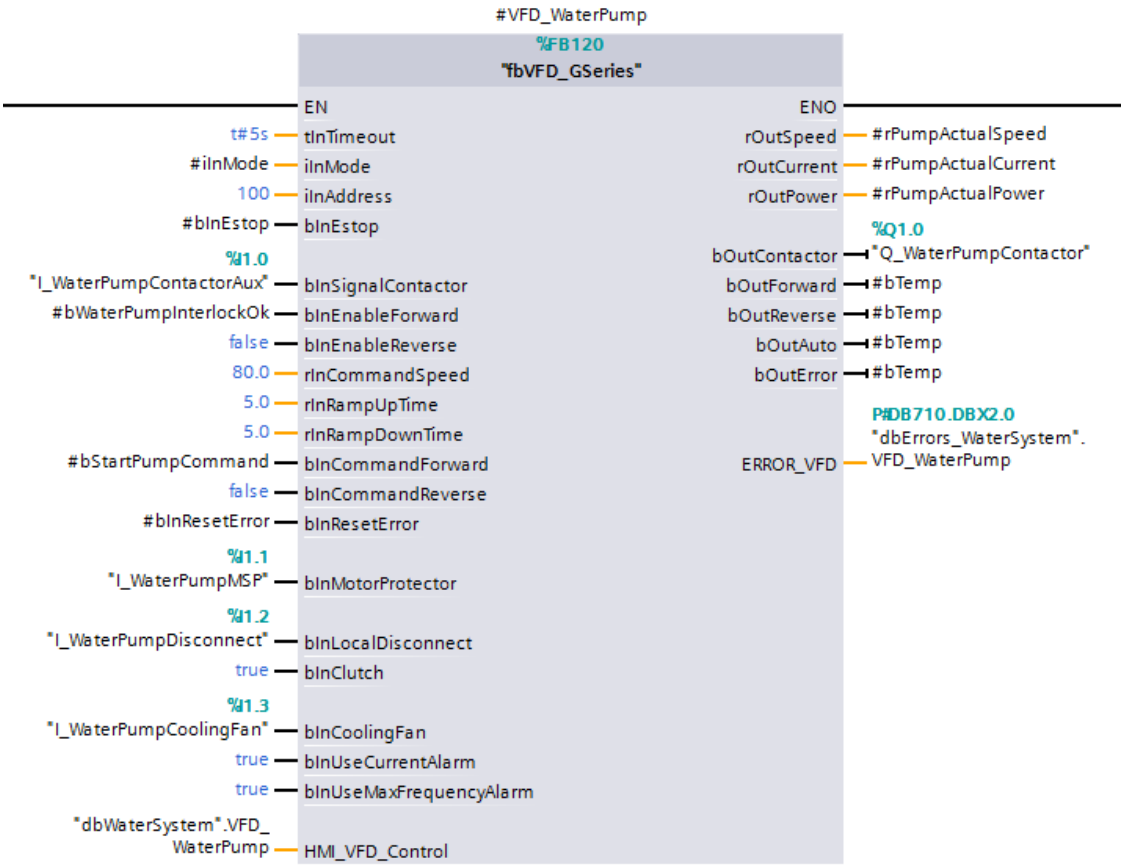
7. Devices

This section covers all objects that are in the Devices Folder of the open library.

7.1. G Series Motor Control – fbVFD_GSeries

7.1.1. Description

The G Series VFD Control Function Block controls Siemens G Series VFD or Micromaster VFDs using Standard Telegram 1. A technology object is not used by this block, which allows for a lower CPU requirement for each drive and avoids technology object count limitations. It utilizes Standard Telegram 1 and has been tested on a G120, however, it will function with any drive using Standard Telegram 1 on either Profibus or Profinet. NOTE: Due to a limitation of the S7-1200 processor, there are certain functions, detailed below, that are only supported on the S7-1500 processor.



7.1.2. Function Block Interface

7.1.2.1. Input Parameters

Input Variables	Type	Description
tInTimeout	Time	The amount of time before an error condition triggers an error
iInMode	Int	Device mode input. See Section 12.1.2 of the Library Overview and Architecture document of the Library Overview and Architecture document for further details
bInEstop	Bool	Emergency stop input
iInAddress	Int	The VFD IO start address. Input and Output IO addresses need to start at the same value
bInSignalContactor	Bool	Contactor feedback. Map the Output bOutContactor if no Contactor exists.
bInEnableForward	Bool	Forward interlock. The motor is allowed to run in the forward direction if this condition has been met. If the motor should always be able to move in this direction, this input should be set to true. To disable the motor from ever traveling in this direction, this input should be set to false
bInEnableReverse	Bool	Reverse interlock. The motor is allowed to run in the reverse direction if this condition has been met. If the motor should always be able to move in this direction, this input should be set to true. To disable the motor from ever traveling in this direction, this input should be set to false
rInCommandSpeed*	Real	Speed set point for automatic mode in percent. (0.0-100.0%)
rInRampUpTime	Real	The ramp up time for the VFD in seconds (S7-1500 ONLY)
rInRampDownTime	Real	The ramp down time for the VFD in seconds (S7-1500 ONLY)
bInCommandForward*	Bool	This is the forward command for auto mode. If this input is set high in auto mode the motor will run in this direction until the forward interlock becomes false, or this input is set low
bInCommandReverse*	Bool	This is the reverse command for auto mode. If this input is set high in auto mode the motor will run in this direction until the reverse interlock becomes false, or this input is set low
bInResetError	Bool	If an error condition exists, the error must first be fixed, then set this bit high to reset the internal error
bInMotorProtector	Bool	This input should be mapped directly to the motor's circuit breaker input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered
bInLocalDisconnect	Bool	This input should be wired directly to the motor's safety disconnect switch input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered

bInClutch	Bool	This input should be wired directly into the motor's clutch overload input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered
bInCoolingFan	Bool	This input is used to tell the VFD that the cooling fan is running, if available. If not available, this input should be set to true. When false for 30 seconds after VFD starts, the motor is disabled and an error is triggered
bInUseCurrentAlarm	Bool	This input enables or disables an alarm for drive overcurrent. If an overcurrent alarm should be used, then this input should be true. If the current alarm is to be disabled, then this input should be false, or left unwired
bInUseMaxFrequency	Bool	This input enables or disables an alarm for max frequency. If a frequency alarm should be used, then this input should be true. If the frequency alarm is to be disabled, then this input should be false, or left unwired

*Only valid in Automatic Mode

7.1.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_VFD_Control	udtHMI_VFD_Control	This in/out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to control the motor in manual mode, including forward, reverse, and stop commands. Also inside this UDT are all feedback signals including actual speed, actual current, and the forward/reverse signals

7.1.2.3. Out Parameters

Output Variables	Type	Description
bOutContactor	Bool	Contactor output. Map to Output to turn on contactor if used
bOutForward	Bool	Motor running forward feedback from VFD
bOutReverse	Bool	Motor running reverse feedback from VFD
rOutSpeed	Real	Actual speed from VFD
rOutCurrent	Real	Actual current from VFD (S7-1500 ONLY)
rOutPower	Real	Actual power from VFD (S7-1500 ONLY)
bOutError	Bool	This output indicates whether there's an error condition in the VFD
bOutAuto	Bool	Output indicating whether the block is in auto mode
ERROR_VFD	udtError_VFD	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state

7.1.3. User Defined Types

7.1.3.1. udtHMI_VFD_Control

Name	Data Type	Description
iMode	Int	Current mode
iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details
iStatus	Int	Status for HMI display. See Section 12.3.1 of the Library Overview and Architecture document for further details
rManualSpeedSP	Real	Speed set point for manual mode
rAutoSpeedSP	Real	Speed set point for automatic mode
rActualSpeed	Real	Actual motor speed
rActualCurrent	Real	Actual motor current
rActualPower	Real	Actual motor power
bPB_ResetError	Bool	Pushbutton Reset errors
bPB_Forward	Bool	Pushbutton Move forward in manual mode
bPB_Reverse	Bool	Move reverse in manual mode pushbutton
bPB_Stop	Bool	Stop in manual mode pushbutton
bPBEN_ResetError	Bool	Reset errors pushbutton enabled
bPBEN_Forward	Bool	Forward pushbutton enabled
bPBEN_Reverse	Bool	Reverse pushbutton enabled
bPBEN_Stop	Bool	Stop pushbutton enabled
bForwardOn	Bool	Run forward command is on
bReverseOn	Bool	Run reverse command is on
bSignalForward	Bool	Forward signal
bSignalReverse	Bool	Reverse signal
bError	Bool	Overall error
bInterlock	Bool	VFD Interlocked

7.1.3.2. udtError_VFD

Errors	Description
Current	Motor Protector tripped
Local_Disconnect	Motor Local switch OFF

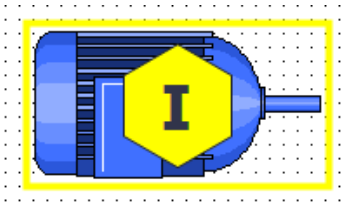
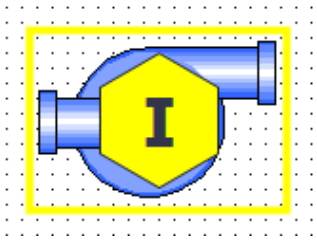
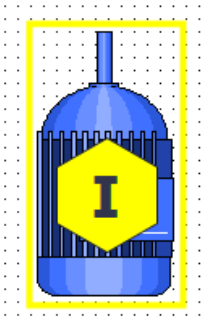
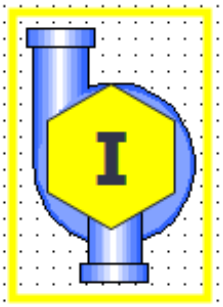
Clutch	Clutch error
Signal_Contactor_On	No feedback from motor contactor
Signal_Contactor_Off	Contactor still ON
Signal_Forward	No SIGNAL Forward from VFD
Signal_Reverse	No SIGNAL Reverse from VFD
Not_Stopped	Motor doesn't stop
VFD_Max_Freq	Max frequency reached
VFD_Current	VFD overcurrent
VFD_Motor_Overload	Motor overload
VFD_Overload	VFD overload
VFD_Fault	VFD fault
Cooling_Fan_On	Cooling fan did not come on within 30 seconds

7.1.4. HMI Icon Display

The HMI Icon is used on the main HMI screen to provide a brief overview of the status of the device. Custom devices can be made to match specific user application styles.

7.1.4.1. HMI Icon Display Objects

The library contains the following objects to be used on the HMI screen:

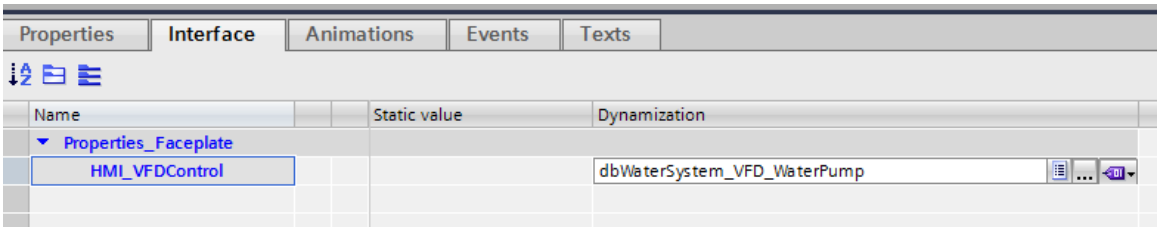
Motor Horizontal	Pump Horizontal	Motor Vertical	Pump Vertical
			

7.1.4.2. HMI Icon Appearance and Functionality

Indicator	Meaning
Device Color	Blue: Stopped Green: Running Yellow: Start Red: Error state Red Flashing: E-stop pressed
Yellow Border	Manual mode
Yellow Hexagon with 'I' in center	Interlocked

7.1.4.3. HMI Icon Interface

The device interface is shown below. It requires only one property linked, the HMI_VFDControl property. The HMI_VFDControl needs to be mapped to the same 'udtHMI_VFDControl' used by the Function Block.

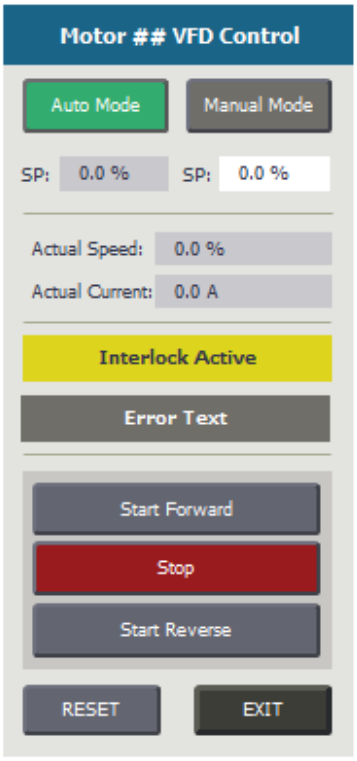


7.1.5. HMI Pop-up Faceplate

The HMI pop up provides a detailed display of the status of the VFD. Additionally, it allows for control on the VFD when the System Mode is Independent or Manual.

7.1.5.1. HMI Pop-up Display

The library contains a single pop-up faceplate shown below:



7.1.5.2. HMI Pop-up Appearance and Functionality

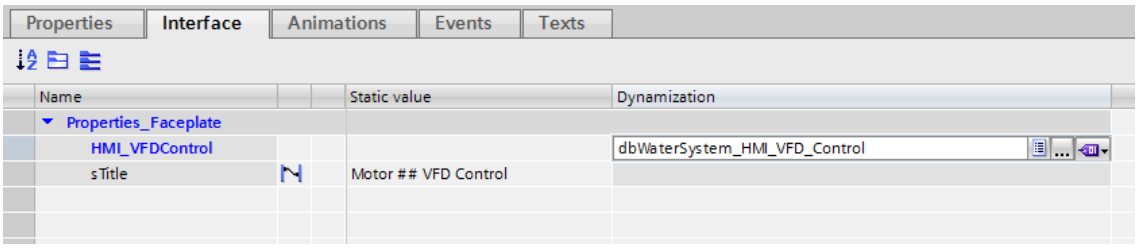
The table below gives a feature description for the pop up faceplate:

Object	Type	Description
Auto Mode	Button	Sets device to Auto Mode. Only available when global mode is Independent. Button is green when devices is in Auto Mode. Button is green when device is in Auto Mode.
Manual Mode	Button	Sets device to Manual Mode. Only available when global mode is Independent. Button is green when devices is in Auto Mode. Button is green when device is in Manual Mode.
Auto Mode SP	Output	Displays PLC set point to be used in Auto Mode
Manual Mode SP	Input	Adjusts the speed set point when in Manual Mode
Actual Speed	Output	Displays the current speed of the motor (%)
Actual Current	Output	Displays motor current (A)
Interlock Status Field	Output	Displays current status of device interlock (yellow for interlock, grey when not)

Error Status Field	Output	Displays current error status. Field box turns red if any errors are present, and displays active error text. It will scroll through multiple alarms.
Start Forward Button	Button	Commands motor forward when in Manual Mode
Stop Button	Button	Stops the device
Start Reverse Button	Button	Commands motor in reverse when in Manual Mode
RESET Button	Button	Resets errors
EXIT Button	Button	Exits the pop up screen

7.1.5.3. HMI Pop-up Interface

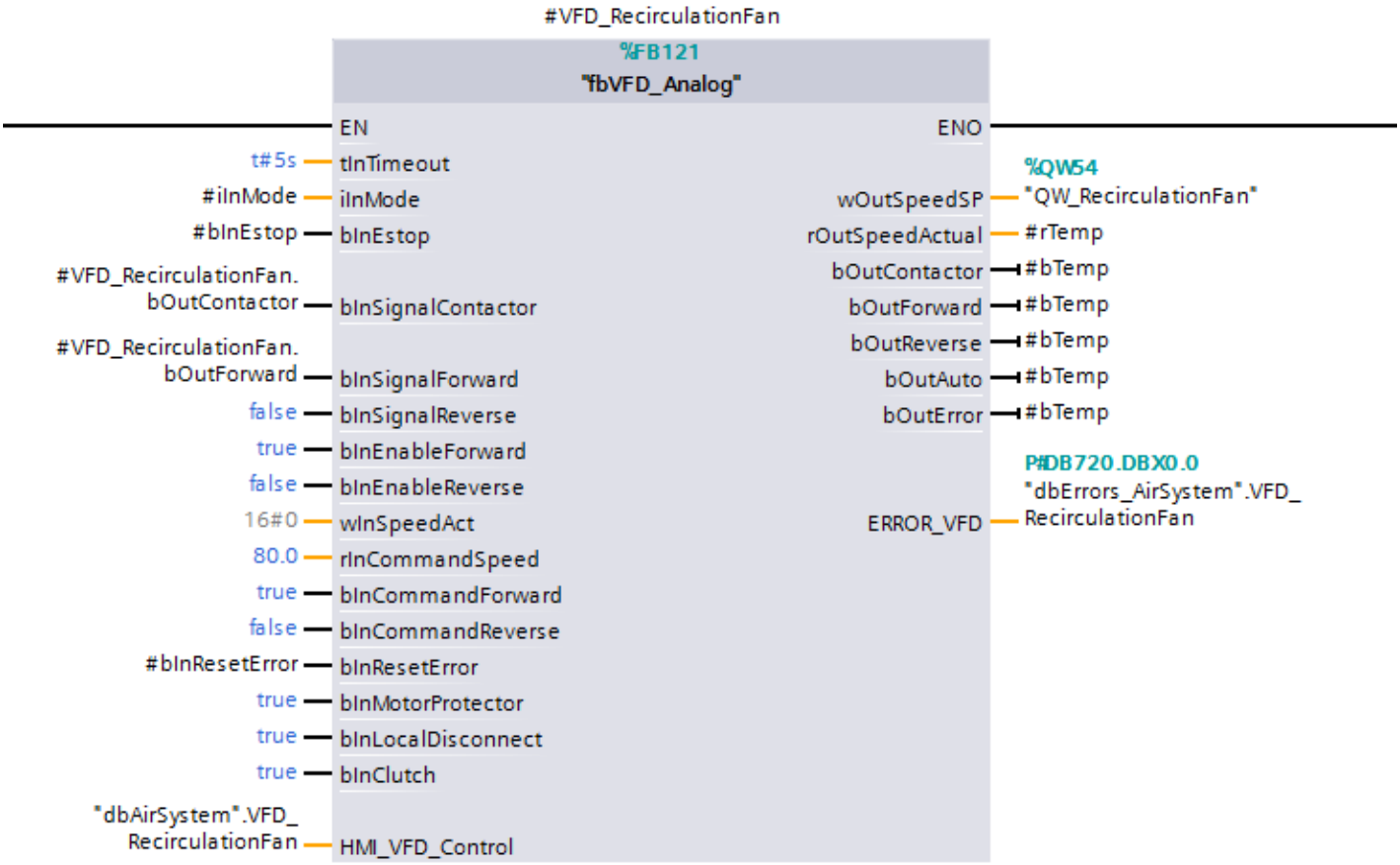
The device interface is shown below. Two properties are linked in this interface: HMI_VFDControl and sTitle. The HMI_VFDControl needs to be mapped to the same 'udtHMI_VFD_Control' used by the Function Block. The sTitle property requires a string by which to identify the device (such as a device name), and this value will display at the top of the Pop-up.



7.2. Analog VFD Control – fbVFD_Analog

7.2.1. Description

The fbVFD_Analog Motor Control Function Block is utilized for use of a VFD controlled with digital and analog signals. The error UDT is kept identical to the GSeries error UDT, however, some of the errors are not utilized by the block in this revision.



7.2.2. Function Block Interface

7.2.2.1. Input Parameters

Input Variables	Type	Description
tInTimeout	Time	The amount of time before an error condition triggers an error
iInMode	Int	Device mode input. See Section 12.1.2 of the Library Overview and Architecture document of the Library Overview and Architecture document for further details
bInEstop	Bool	Emergency stop input
bInSignalContactor	Bool	Contactor feedback

bInSignalForward	Bool	Forward feedback bit. This tells the motor function block whether the motor is going in the forward direction. If no feedback input is available, wire the forward output to the forward feedback input
bInSignalReverse	Bool	Reverse feedback bit. This tells the motor function block whether the motor is going in the reverse direction. If no feedback input is available, wire the forward output to the forward feedback input
wInSpeedAct	Word	Actual speed feedback from an analog speed sensor. If there is no speed feedback, then wOutSpeedSP should be mapped to this input
bInEnableForward	Bool	Forward interlock. The motor is allowed to run in the forward direction if this condition has been met. If the motor should always be able to move in this direction, this input should be set to true. To disable the motor from ever traveling in this direction, this input should be set to false
bInEnableReverse	Bool	Reverse interlock. The motor is allowed to run in the reverse direction if this condition has been met. If the motor should always be able to move in this direction, this input should be set to true. To disable the motor from ever traveling in this direction, this input should be set to false
rInCommandSpeed*	Real	Speed set point for automatic mode in percent (0.0-100.0%)
rInRampUpTime	Real	The ramp up time for the VFD in seconds
rInRampDownTime	Real	The ramp down time for the VFD in seconds
bInCommandForward*	Bool	This is the forward command for auto mode. If this input is set high in auto mode the motor will continue to run in this direction until the forward interlock becomes false, or this input is set low
bInCommandReverse*	Bool	This is the reverse command for auto mode. If this input is set high in auto mode the motor will continue to run in this direction until the reverse interlock becomes false, or this input is set low
bInResetError	Bool	If an error condition exists, the error must be fixed first, then set this bit high to reset the internal error
bInMotorProtector	Bool	This input should be wired directly to the motor's circuit breaker input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered
bInLocalDisconnect	Bool	This input should be wired directly to the motor's safety disconnect switch input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered
bInClutch	Bool	This input should be wired directly into the motor's clutch overload input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered

*Only valid in Automatic Mode

7.2.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_VFD_Control	udtHMI_VFD_Control	This In/Out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to control the motor in manual mode, including forward, reverse, and stop commands. Also inside this UDT are all feedback signals including actual speed, actual current, and the forward/reverse signals

7.2.2.3. Out Parameters

Output Variables	Type	Description
bOutContactor	Bool	Contactor output. Map to Output to turn on contactor if used
bOutForward	Bool	Motor is running in forward. Forward running signal based on input to block
bOutReverse	Bool	Motor is running in reverse. Reverse running signal based on input to block
wOutSpeedSP	Word	Speed value to Map to Output to control speed. Map to analog output.
rOutSpeedActual	Real	Actual Speed based on speed feedback signal
bOutError	Bool	This output indicates whether there's an error condition in the VFD
bOutAuto	Bool	Output indicating whether the block is in auto mode
ERROR_VFD	udtError_VFD	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state

7.2.3. User Defined Types

7.2.3.1. udtHMI_VFD_Control

Name	Data Type	Description
iMode	Int	Current mode
iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details
iStatus	Int	Status for HMI display. See Section 12.3.1 of the Library Overview and Architecture document for further details
rManualSpeedSP	Real	Speed set point for manual mode
rAutoSpeedSP	Real	Speed set point for automatic mode
rActualSpeed	Real	Actual motor speed

rActualCurrent	Real	Actual motor current
rActualPower	Real	Actual motor power
bPB_ResetError	Bool	Reset errors pushbutton
bPB_Forward	Bool	Move forward in manual mode pushbutton
bPB_Reverse	Bool	Move reverse in manual mode pushbutton
bPB_Stop	Bool	Stop in manual mode pushbutton
bPBEN_ResetError	Bool	Reset Errors pushbutton enabled
bPBEN_Forward	Bool	Forward pushbutton enabled
bPBEN_Reverse	Bool	Reverse pushbutton enabled
bPBEN_Stop	Bool	Stop pushbutton enabled
bForwardOn	Bool	Forward command is on
bReverseOn	Bool	Reverse command is on
bSignalForward	Bool	Forward signal
bSignalReverse	Bool	Reverse signal
bError	Bool	Overall error
bInterlock	Bool	VFD Interlocked

7.2.3.2. udtError_VFD

Errors	Description
Current	Motor Protector tripped
Local_Disconnect	Motor Local switch OFF
Clutch	Tool (clutch) error
Signal_Contactor_On	No feedback from motor contactor
Signal_Contactor_Off	Contactor still ON
Signal_Forward	No SIGNAL Forward from VFD
Signal_Reverse	No SIGNAL Reverse from VFD
Not_Stopped	Motor doesn't stop
VFD_Max_Freq	Part of UDT, but not used by fbVFD_Analog
VFD_Current	Part of UDT, but not used by fbVFD_Analog
VFD_Motor_Overload	Part of UDT, but not used by fbVFD_Analog
VFD_Overload	Part of UDT, but not used by fbVFD_Analog
VFD_Fault	Part of UDT, but not used by fbVFD_Analog

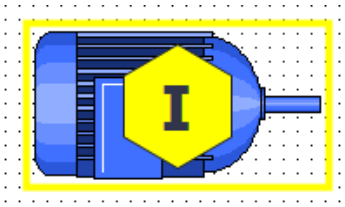
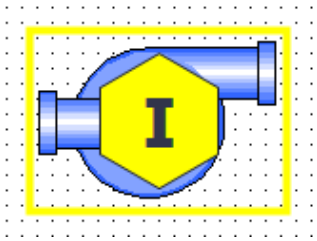
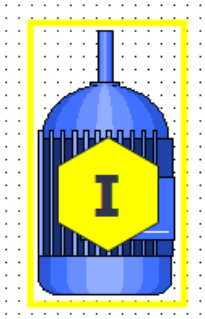
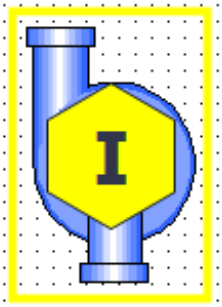
Cooling_Fan_On	Cooling fan did not come on within 30 seconds

7.2.4. HMI Icon Display

The HMI Icon is used on the main HMI screen to provide a brief overview of the status of the device. Custom devices can be made to match specific applications. Note: As the VFD_Analog faceplate uses the same UDT as VFD_GSeries they share the same faceplate.

7.2.4.1. HMI Icon Display Objects

The library contains the following objects to be used on the HMI screen:

Motor Horizontal	Pump Horizontal	Motor Vertical	Pump Vertical
			

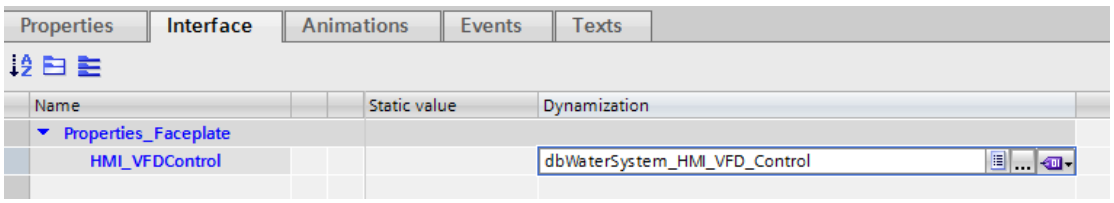
7.2.4.2. HMI Icon Appearance and Functionality

The device appearance changes depending on the state it is in, see table below for description:

Indicator	Meaning
Device Color	Blue: Stopped Green: Running Yellow: Start Red: Error state Red Flashing: E-stop pressed
Yellow Border	Manual mode
Yellow Hexagon with 'I' in center	Interlocked

7.2.4.3. HMI Icon Interface

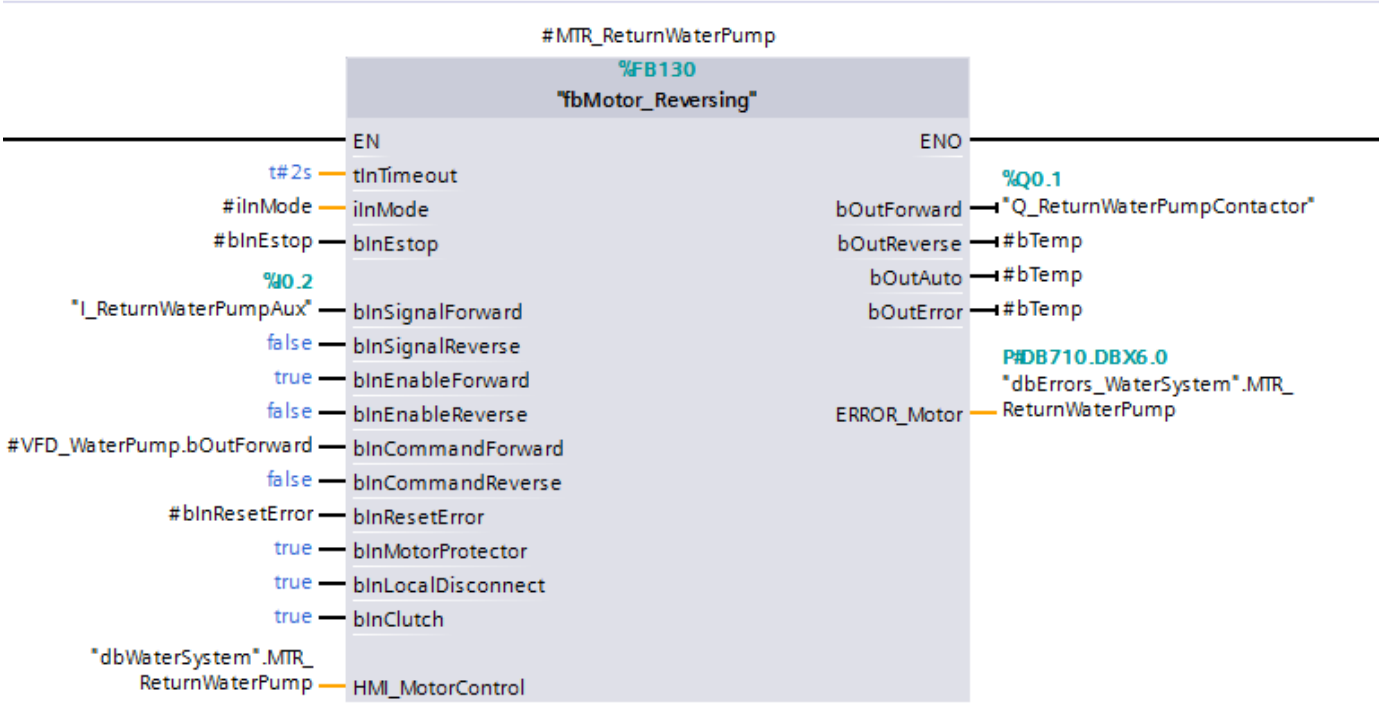
The device interface is shown below. It requires only one property linked, the HMI_VFD_Control property. The HMI_VFD_Control needs to be mapped to the same 'udtHMI_VFD_Control' used by the Function Block.



7.3. Digital Single Speed Motor – fbMotor_Reversing

7.3.1. Description

The Reversing Motor Control Function Block is utilized for motors started and stopped using digital outputs. If the Motor is only one direction, then map False into bInEnableReverse, and this will perform as a single direction motor.



7.3.2. Function Block Interface

7.3.2.1. Input Parameters

Input Variables	Type	Description
tInTimeout	Time	The amount of time before an error condition triggers an error
iInMode	Int	Device mode input. See Section 12.1.2 of the Library Overview and Architecture document for further details
bInEstop	Bool	Emergency stop input
bInSignalForward	Bool	Forward feedback bit. This tells the motor function block whether the motor is going in the forward direction. If no feedback input is available, wire the forward output to the forward feedback input
bInSignalReverse	Bool	Reverse feedback bit. This tells the motor function block whether the motor is going in the reverse direction. If no feedback input is available, wire the forward output to the forward feedback input
bInEnableForward	Bool	Forward interlock. The motor is allowed to run in the forward direction if this condition has been met. If the motor should always be able to move in this direction, this input should be set to true. To disable the motor from ever traveling in this direction, this input should be set to false
bInEnableReverse	Bool	Reverse interlock. The motor is allowed to run in the reverse direction if this condition has been met. If the motor should always be able to move in this direction, this input should be set to true. To disable the motor from ever traveling in this direction, this input should be set to false

bInCommandForward*	Bool	This is the forward command for auto mode. If this input is set high in auto mode the motor will continue to run in this direction until the forward interlock becomes false, or this input is set low
bInCommandReverse*	Bool	This is the reverse command for auto mode. If this input is set high in auto mode the motor will continue to run in this direction until the reverse interlock becomes false, or this input is set low
bInResetError	Bool	If an error condition exists, the error must be fixed first, then set this bit high to reset the internal error
bInMotorProtector	Bool	This input should be wired directly to the motor's circuit breaker input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered
bInLocalDisconnect	Bool	This input should be wired directly to the motor's safety disconnect switch input, if available. If not available, this input should be set to true. When false, the motor is disabled and an error is triggered
bInClutch	Bool	This input should be wired directly to the motor's clutch input. When false, the motor is disabled and an error is triggered

*Only valid in Automatic Mode

7.3.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_MotorControl	udtHMI_MotorControl	This in/out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to control the motor in manual mode, including forward, reverse, and stop commands

7.3.2.3. Out Parameters

Output Variables	Type	Description
bOutForward	Bool	This output should be wired to the motor's forward contact
bOutReverse	Bool	This output should be wired to the motor's reverse contact
bOutError	Bool	This output indicates whether there's an error condition in the motor
ERROR_Motor	udtError_Motor	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state
bOutAuto	Bool	This output indicates whether the block is in auto mode

7.3.3. User Defined Types

7.3.3.1. udtHMI_MotorControl

Name	Data Type	Description
iMode	Int	Current mode
iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details

iStatus	Int	Status for HMI display. See Section 12.3.1 of the Library Overview and Architecture document for further details
bPB_ResetError	Bool	Reset errors pushbutton
bPB_Forward	Bool	Move forward in manual mode pushbutton
bPB_Reverse	Bool	Move Reverse in manual mode pushbutton
bPB_Stop	Bool	Stop in manual mode pushbutton
bPBEN_ResetError	Bool	Reset Errors pushbutton enabled
bPBEN_Forward	Bool	Forward pushbutton enabled
bPBEN_Reverse	Bool	Reverse pushbutton enabled
bPBEN_Stop	Bool	Stop pushbutton enabled
bForwardOn	Bool	Forward command is on
bReverseOn	Bool	Reverse command is on
bSignalForward	Bool	Forward signal
bSignalReverse	Bool	Reverse signal
bError	Bool	Overall error
bInterlock	Bool	Motor interlocked

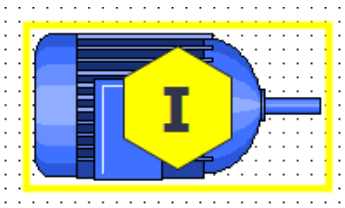
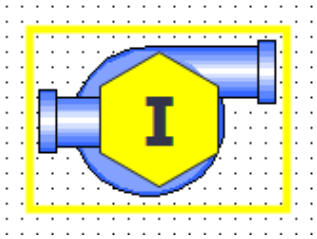
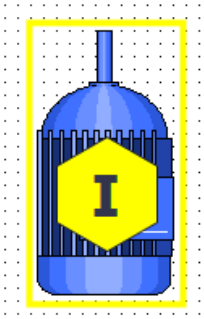
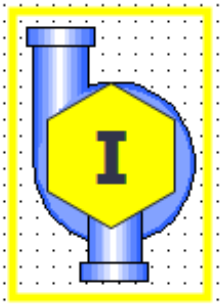
7.3.3.2. udtError_Motor

Name	Description
MotorProtectorTripped	Motor protector tripped
LocalDisconnectOff	Local disconnect off
ClutchTripped	Clutch tripped
NoSignalForward	No signal from forward contactor
NoSignalReverse	No signal from Reverse contactor
MotorNotStopped	Motor doesn't stop

7.3.4. HMI Icon Display

7.3.4.1. HMI Icon Display Objects

The library contains the following objects to be used on the HMI screen:

Motor Horizontal	Pump Horizontal	Motor Vertical	Pump Vertical
			

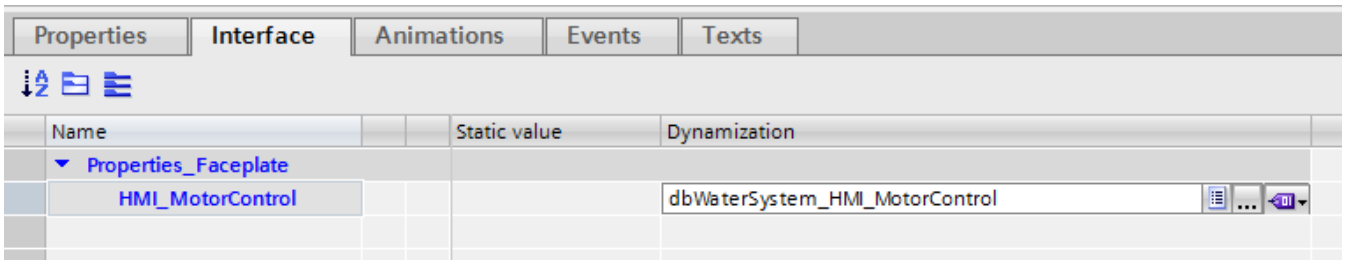
7.3.4.2. HMI Icon Appearance and Functionality

The device appearance's changes depending on the state it is in, see table below for description:

Indicator	Meaning
Device Color	Blue: Stopped Green: Running Yellow: Start Red: Error State Red Flashing: E-stop pressed
Yellow Border	Manual Mode
Yellow Hexagon with 'I' in center	Interlocked

7.3.4.3. HMI Icon Interface

The device interface is shown below. It requires only one property linked, the HMI_MotorControl property. The HMI_MotorControl property needs to be mapped to the same 'udtHMI_MotorControl' used by the Function Block.

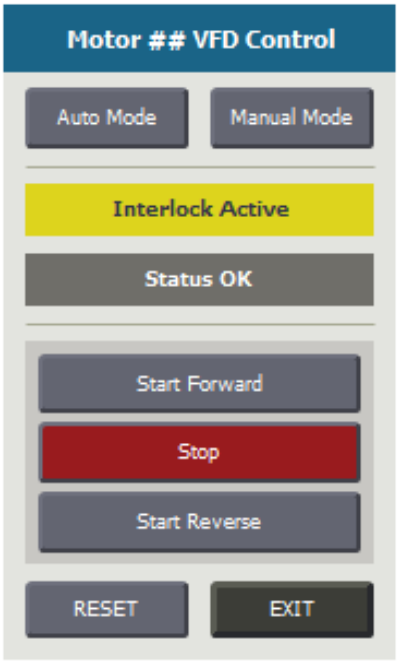


7.3.5. HMI Pop-up Faceplate

The HMI pop up provides a detailed display of the status of the motor. Additionally, it allows for control on the motor when the System Mode is Independent or Manual modes.

7.3.5.1. HMI Pop-up display

The library contains a single pop-up faceplate shown below:



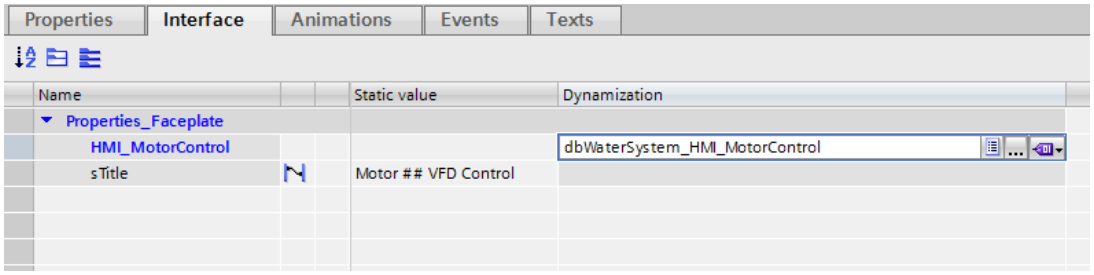
7.3.5.2. HMI Pop-up Appearance and Functionality

The table below gives a feature description for the pop up faceplate:

Object	Type	Description
Auto Mode Button	Button	Sets device to Auto Mode. Only available when global mode is Independent. Button is green when devices is in Auto Mode.
Manual Mode	Button	Sets device to Manual Mode. Only available when global mode is Independent. Button is green when device is in Manual Mode.
Interlock Status Field	I/O Field	Displays current status of device interlock (yellow for interlock, grey when not)
Error Status Field	I/O Field	Displays current error status. Field box turns red if any errors are present, and displays active error text. It will scroll through multiple alarms.
Start Forward Button	Button	Drives motor forward when in Manual Mode
Stop Button	Button	Stops the device
Start Reverse Button	Button	Drives motor in reverse when in Manual Mode
RESET Button	Button	Resets errors
EXIT Button	Button	Exits the pop up screen

7.3.5.3. HMI Pop-up Interface

The device interface is shown below. Two properties are linked in this interface: HMI_MotorControl and sTitle. The HMI_MotorControl needs to be mapped to the same 'udtHMI_MotorControl' used by the Function Block. The sTitle property requires a string passed into it, and this value will display at the top of the Pop-up.

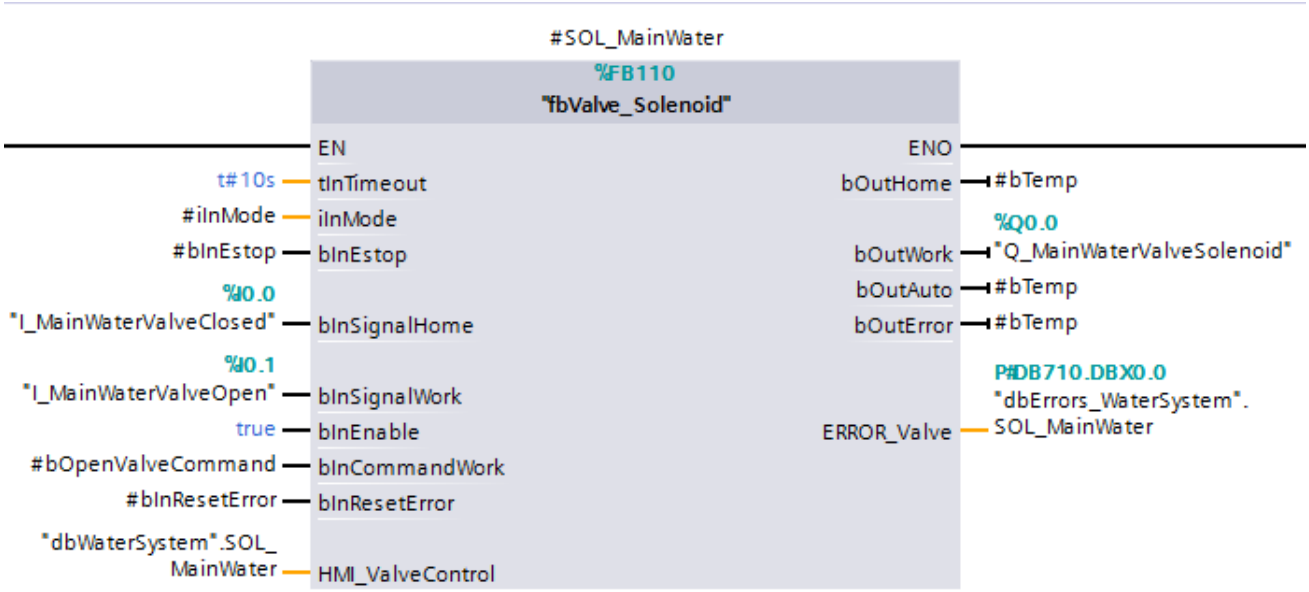

Properties
Interface
Animations
Events
Texts
Name
Static value
Dynamization
Properties_Faceplate
HMI_MotorControl
dbWaterSystem_HMI_MotorControl
sTitle
Motor ## VFD Control

8. Valve Control

8.1. Two State Solenoid Valve – fbValve_Solenoid

8.1.1. Description

This library item is to be utilized with two state solenoid valve applications. It can be used for spring close, spring open, or double acting solenoids.



8.1.2. Function Block Interface

8.1.2.1. Input Parameters

Input Variables	Type	Description
tInTimeout	Time	The amount of time before an error condition triggers an error
iInMode	Int	Device mode input. See Section 12.1.2 of the Library Overview and Architecture document for further details
bInEstop	Bool	Emergency stop input
bInSignalHome	Bool	Home position feedback
bInSignalWork	Bool	Work position feedback
bInEnable	Bool	Valve interlock. The valve is allowed to operate if this condition has been met
bInCommandWork*	Bool	Move to work position in automatic mode
bInResetError	Bool	If an error condition exists, the error must be fixed first, then set this bit high to reset the internal error

*Only valid in Automatic Mode

8.1.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_CylinderControl	udtHMI_CylinderControl	This in/out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to control the cylinder in manual mode

8.1.2.3. Out Parameters

Output Variables	Type	Description
bOutHome	Bool	Output for home position
bOutWork	Bool	Output for work position
bOutAuto	Bool	This output indicates whether the block is in auto mode
bOutError	Bool	This output indicates whether there's an error condition in the cylinder
ERROR_Cylinder	udtError_Cylinder	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state

8.1.3. User Defined Types

8.1.3.1. udtError_Valve

Errors	Description
NoHomeFeedback	Home position feedback not active
NoWorkFeedback	Work position feedback not active
HomeFeedbackStillActive	Home position feedback still active when commanded to Work Position
WorkFeedbackStillActive	Work position feedback still active when commanded to Home Position

8.1.3.2. udtHMI_ValveControl

Name	Data Type	Description
iMode	Int	Current mode
iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details
iStatus	Int	Status for HMI display. See Section 12.3.1 of the Library Overview and Architecture document for further details
bPB_ResetError	Bool	Reset errors pushbutton
bPB_Home	Bool	Move to home in manual mode pushbutton
bPB_Work	Bool	Move to work in manual mode pushbutton

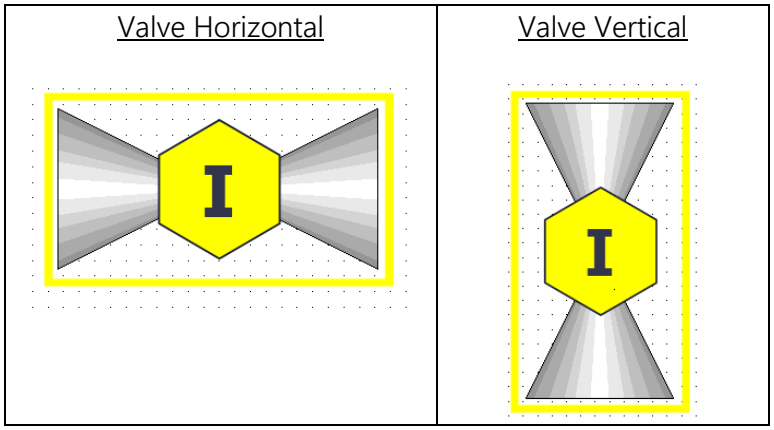
bPBEN_ResetError	Bool	Reset Error pushbutton enabled
bPBEN_Home	Bool	Home pushbutton enabled
bPBEN_Work	Bool	Work pushbutton enabled
bHomeOn	Bool	Home command is on
bWorkOn	Bool	Work command is on
bSignalHome	Bool	Home feedback
bSignalWork	Bool	Work feedback
bError	Bool	Error status
bInterlock	Bool	Valve interlocked

8.1.4. HMI Icon Display

The HMI Icon is used on the main HMI screen to provide a brief overview of the status of the device. Custom devices can be made to match specific applications.

8.1.4.1. HMI Icon Display Objects

The library contains the following objects to be used on the HMI screen:



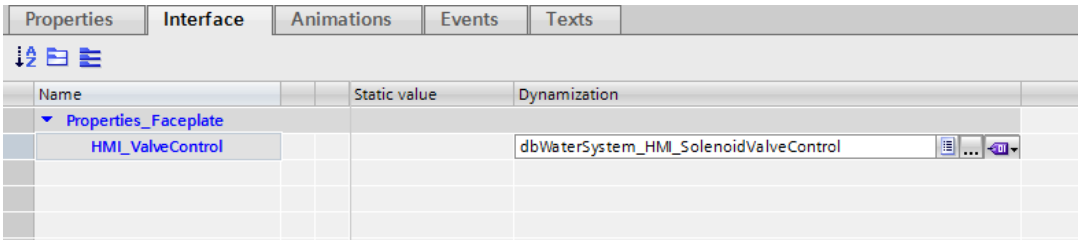
8.1.4.2. HMI Icon Appearance and Functionality

The device appearance's changes depending on the state it is in, see table below for description:

Indicator	Meaning
Device Color	Green: Open Yellow: Valve Opening/Closing Red: Error state Red Flashing: E-stop pressed
Yellow Border	Manual mode
Yellow Hexagon with 'I' in center	Interlocked

8.1.4.3. HMI Icon Interface

The device interface is shown below. It requires only one property linked, the HMI_ValveControl property. The HMI_ValveControl property requires a datablock of the 'udtHMI_ValveControl' type to be linked: in the example below it is dbWaterSystem_HMI_SolenoidValveControl.

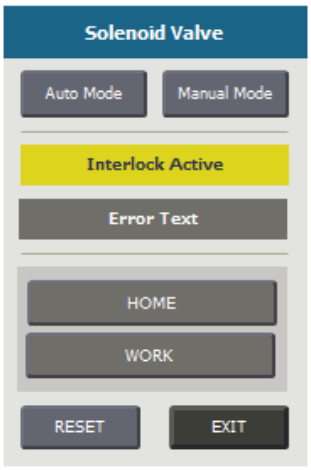


8.1.5. HMI Pop-up Faceplate

The HMI pop up provides a detailed display of the status of the valve. Additionally, it allows for control on the valve when the System Mode is Independent or Manual.

8.1.5.1. HMI Pop-up Display

The library contains a single pop-up faceplate shown below:



8.1.5.2. HMI Pop-up Appearance and Functionality

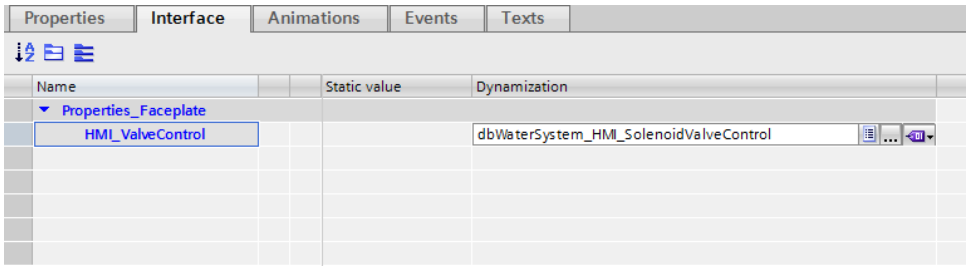
The table below gives a feature description for the pop up faceplate:

Object	Type	Description
Auto Mode Button	Button	Sets device to Auto Mode. Only available when global mode is Independent. Button is green when device is in Auto Mode.
Manual Mode	Button	Sets device to Manual Mode. Only available when global mode is Independent. Button is green when device is in Manual Mode.
Interlock Status Field	I/O Field	Displays current status of device interlock (yellow for interlock, grey when not)
Error Status Field	I/O Field	Displays current error status. Field box turns red if any errors are present, and displays active error text. It will scroll through multiple alarms
HOME Button	Button	Sets valve to HOME position

WORK Button	Button	Sets valve to WORK position
RESET Button	Button	Resets errors
EXIT Button	Button	Exits the pop up screen

8.1.5.3. HMI Pop-up Interface

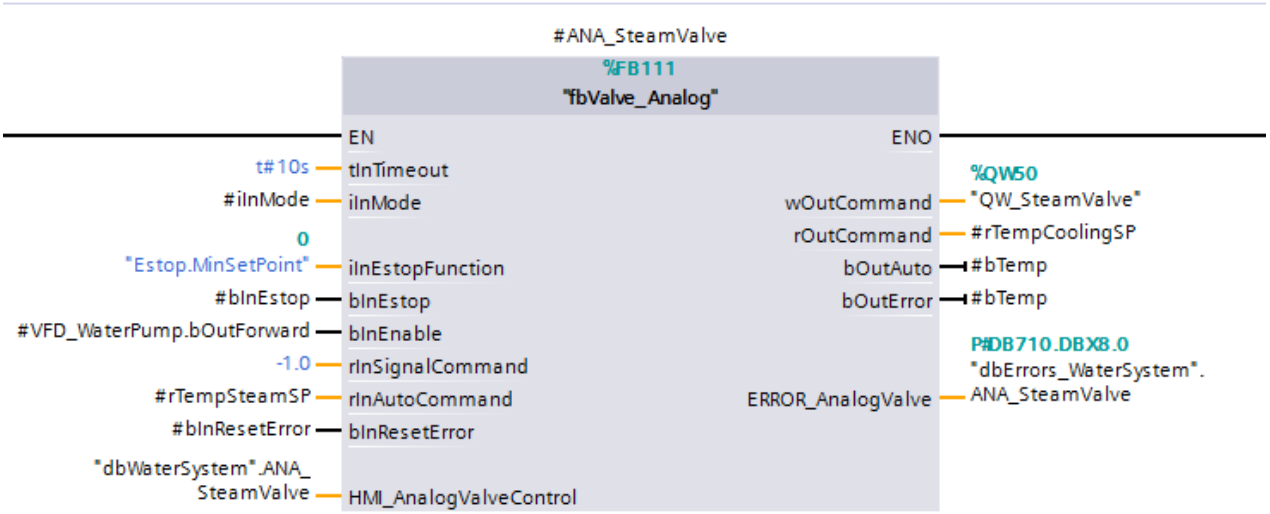
The device interface is shown below. The HMI_ValveControl property The HMI_ValveControl needs to be mapped to the same 'udtHMI_ValveControl' used by the Function Block.



8.2. Analog Valve – fbValve_Analog

8.2.1. Description

This library item is to be utilized with a valve controlled using an analog output. This block will work with any valve that is controlled through an analog output from the PLC.



8.2.2. Function Block Interface

8.2.2.1. Input Parameters

Input Variables	Type	Description
tInTimeout	Time	The amount of time before an error condition triggers an error
iInMode	Int	Device mode input. See Section 12.1.2 of the Library Overview and Architecture document for further details
iInEstopFunction	Int	Enumerated value of E-Stop response constant. 'Estop.MinSetPoint' sets rOutCommand to the minimum set point (0%), 'Estop.MaxSetPoint' sets to the maximum set point (100%), and 'Estop.HMISetPoint' sets to an HMI-specified value that can be in between the min and max.
bInEstop	Bool	Emergency stop input
bInEnable	Bool	Valve interlock. The valve is allowed to operate if this condition has been met
rInSignalCommand	Real	Actual % open from valve (-1 to disable)
rInAutoCommand	Real	Auto mode set point command
bInResetError	Bool	If an error condition exists, the error must be fixed first, then set this bit high to reset the internal error

8.2.2.2. InOut Parameters

In/Out Variables	Type	Description

HMI_CylinderControl	udtHMI_AnalogValveControl	This in/out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to control the cylinder in manual mode and provide status for the HMI

8.2.2.3. Out Parameters

Output Variables	Type	Description
wOutCommand	Word	Value to be mapped to output used to control cylinder
rOutCommand	Real	Output value from 0-100% used for other PLC logic
bOutAuto	Bool	This output indicates whether the block is in auto mode
bOutError	Bool	This output indicates whether there's an error condition in the cylinder
ERROR_AnaValve	udtError_AnaValve	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state

8.2.3. User Defined Types

8.2.3.1. udtHMI_VFD_Control

Name	Data Type	Description
iMode	Int	Current mode
iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details
iStatus	Int	Color status for HMI. See Section 12.3.1 of the Library Overview and Architecture document for further details
rManualSP	Real	Set point for manual mode
rAutoSP	Real	Set point for automatic mode
rEstopSP	Real	Set point for an e-stop condition, if selected
rActual	Real	Actual value
bPB_ResetError	Bool	Reset errors pushbutton
bPBEN_ResetError	Bool	Reset errors pushbutton enable
bError	Bool	Overall error
bInterlock	Bool	Valve Interlocked
iEstopFunction	Int	Enumerated value of E-Stop response constant.

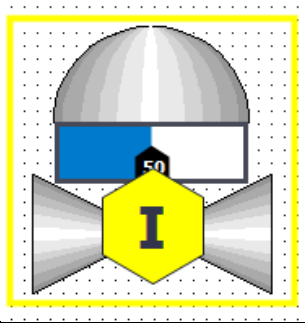
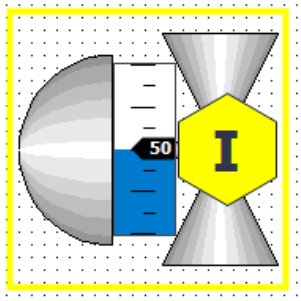
8.2.3.2. udtError_AnalogValve

Name	Data Type	Description
Invalid	Bool	Invalid input given

8.2.4. HMI Icon Display

8.2.4.1. HMI Icon Display Objects

The library contains the following objects to be used on the HMI screen:

Analog Valve Horizontal	Analog Valve Vertical
	

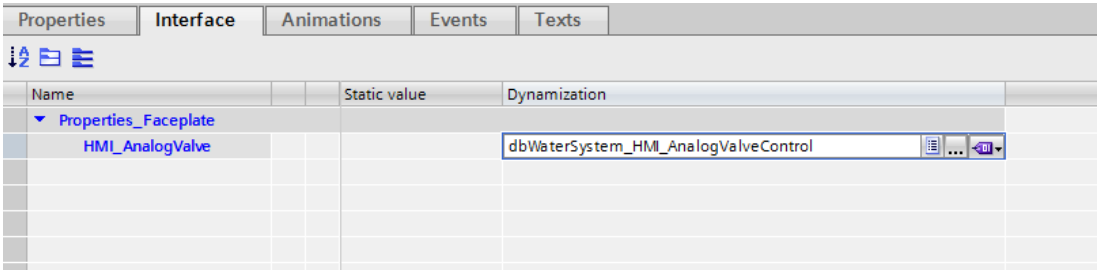
8.2.4.2. HMI Icon Appearance and Functionality

The device appearance's changes depending on the state it is in, see table below for description:

Indicator	Meaning
Device Color	Green: Opening Red: Error state
Yellow Border	Manual mode
Yellow Hexagon with 'I' in center	Interlocked

8.2.4.3. HMI Icon Interface

The device interface is shown below. It requires only one property linked, the HMI_AnalogValveControl property. The HMI_AnalogValveControl needs to be mapped to the same 'udtHMI_AnalogValveControl' used by the Function Block.

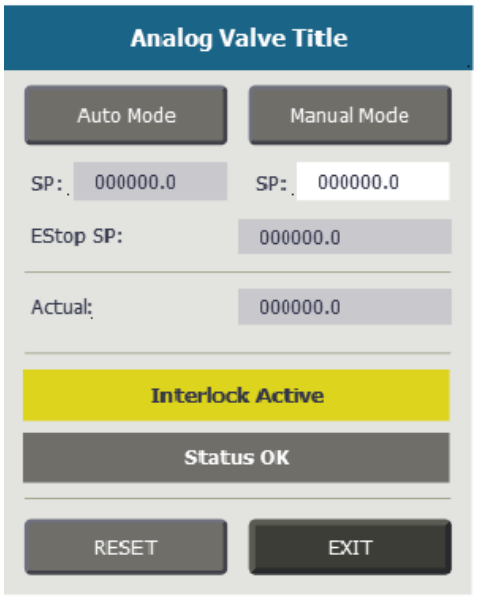


8.2.5. HMI Pop-up Faceplate

The HMI pop up provides a detailed display of the status of the valve. Additionally, it allows for control on the valve when the System Mode is Independent or Manual.

8.2.5.1. HMI Pop-up Display

The library contains a single pop-up faceplate shown below:



8.2.5.2. HMI Pop-up Appearance and Functionality

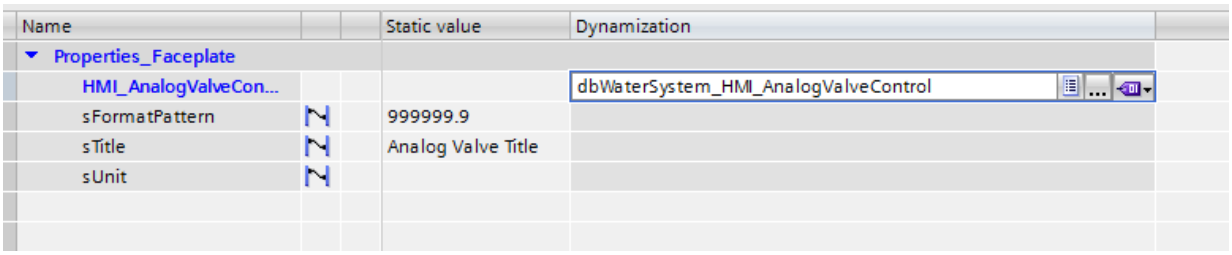
The table below gives a feature description for the pop up faceplate:

Object	Type	Description
Auto Mode Button	Button	Sets device to Auto Mode. Only available when global mode is Independent. Button is green when devices is in Auto Mode.
Manual Mode	Button	Sets device to Manual Mode. Only available when global mode is Independent. Button is green when device is in Manual Mode.
Auto Mode SP	Output	Displays PLC setpoint to be used in Auto Mode.
Manual Mode SP	Input	Adjusts the speed set point when in Manual Mode
Estop SP	Output	Displays the current speed of the motor
Actual	Output	Displays ratio of valve opened
Interlock Status Field	I/O Field	Displays current status of device interlock (yellow for interlock, grey when not)
Error Status Field	I/O Field	Displays current error status. Field box turns red if any errors are present, and displays active error text. It will scroll through multiple alarms
RESET Button	Button	Resets errors
EXIT Button	Button	Exits the pop up screen

8.2.5.3. HMI Pop-up Interface

The device interface is shown below. Four properties are linked in this interface:

- HMI_VFDControl: requires requires to be linked to the same 'udtHM_AnalogValveControl' used by the Function Block.
- sFormatPattern: Changes data format of values, setting amount of leading and trailing zeros of outputs. Adjusted directly in table
- sTitle: Sets title of the pop up display. Adjusted directly in table
- sUnit: sets the units used in the display

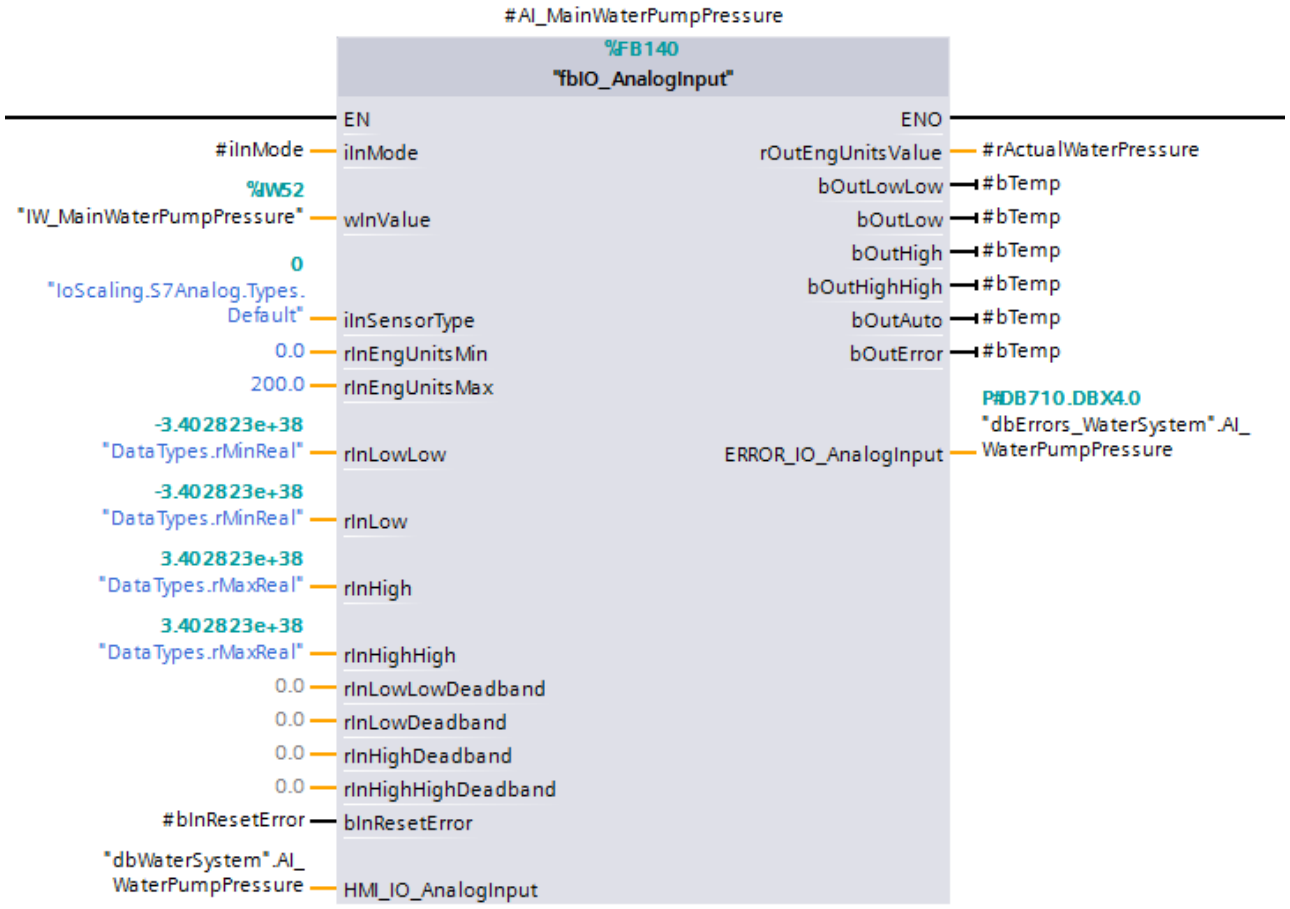


9. Input/Output Control

9.1. Analog Input with Scaling and Alarms – fbIO_AnalogInput

9.1.1. Description

This library object scales analog inputs and provides setups for alarms.



9.1.2. Function Block Interface

9.1.2.1. Input Parameters

Input Variables	Type	Description
iInMode	Int	Mode input. See Section 12.1.2 of the Library Overview and Architecture document for further details
wInValue	Word	Raw input signal from hardware input
rInEngUnitsMin	Real	Minimum Value for Engineering Units
rInEngUnitsMax	Real	Maximum Value for Engineering Units
rInLowLow	Real	Value for Low Alarm to trigger. Use IoScaling.S7Analog.iRawMin to disable

rInLow	Real	Value for Low Warning to trigger. Use IoScaling.S7Analog.iRawMin to disable
rInHigh	Real	Value for High Warning to trigger. Use IoScaling.S7Analog.iRawMax to disable
rInHighHigh	Real	Value for High Alarm to trigger. Use IoScaling.S7Analog.iRawMax to disable
rInLowLowDeadband	Real	Deadband for Low Low Alarm. Use 0 to disable
rInLowDeadband	Real	Deadband for Low Warning. Use 0 to disable
rInHighDeadband	Real	Deadband for High Warning. Use 0 to disable
rInHighHighDeadband	Real	Deadband for High High Alarm. Use 0 to disable
bInResetError	Bool	If an error condition exists, the error must be fixed first, then set this bit high to reset the internal error

*Only valid in Automatic Mode

9.1.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_IO_AnalogInput	udtHMI_AnalogInput	This In/Out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to configure additional settings and provide status to the HMI

9.1.2.3. Out Parameters

Output Variables	Type	Description
rOutEngUnitsValue	Real	Engineering units scaled value
bOutLowLow	Bool	Low low alarm is active
bOutLow	Bool	Low warning is active
bOutHigh	Bool	High warning is active
bOutHighHigh	Bool	High high alarm is active
bOutAuto	Bool	Input is in auto mode
bOutError	Bool	Error Exists
ERROR_IO_AnalogIn	udtError_AnalogInput	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state

9.1.3. User Defined Types

9.1.3.1. udtHMI_VFD_Control

Name	Data Type	Description
iMode	Int	Current mode

iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details
iStatus	Int	Overall HMI Status. See Section 12.3.1 of the Library Overview and Architecture document for further details
rActiveValue	Real	Used Value, Input or Override
rInputValue	Real	Input Value
rManualValue	Real	Manual mode simulated value
bPB_ResetError	Bool	Reset errors pushbutton
bPBEN_ResetError	Bool	Reset error pushbutton enabled
bError	Bool	Overall error

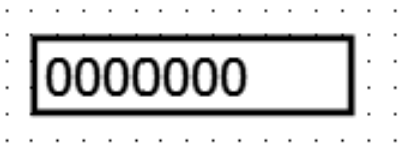
9.1.3.2. udtError_AnalogInput

Errors	Description
LowLowAlarm	Value below low low set point
HighHighAlarm	Value above high high set point
Invalid	Invalid input

9.1.4. HMI Icon Display

9.1.4.1. HMI Icon Display Object

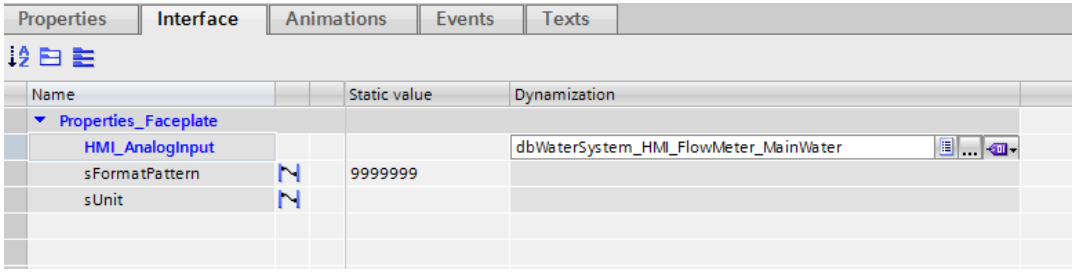
The library contains the following display to be used on the HMI screen:



9.1.4.2. HMI Icon Interface

The device interface is shown below. The following three properties need to be assigned to the object:

- HMI_AnalogInput: Control needs to be mapped to the same 'udtHMI_AnalogInput used by the Function Blocks
- sFormatPattern: Sets the amount of leading/trailing zeros of the display value
- sUnit: Sets the unit the value is displayed in (i.e. amps, centimeters etc.)

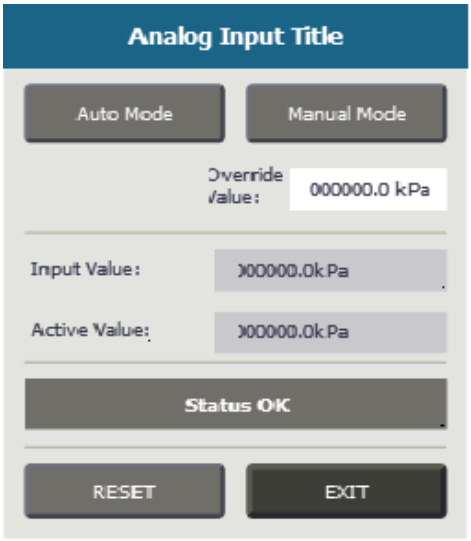


9.1.5. HMI Pop-up Faceplate

The HMI pop up provides a detailed display of the status of the Analog Input device. Additionally, it allows for override control of the device reading when the System Mode is Independent or Manual.

9.1.5.1. Pop Up Object

The library contains a single pop-up faceplate shown below:



9.1.5.2. HMI Pop-up Appearance and Functionality

The table below gives a feature description for the pop up faceplate:

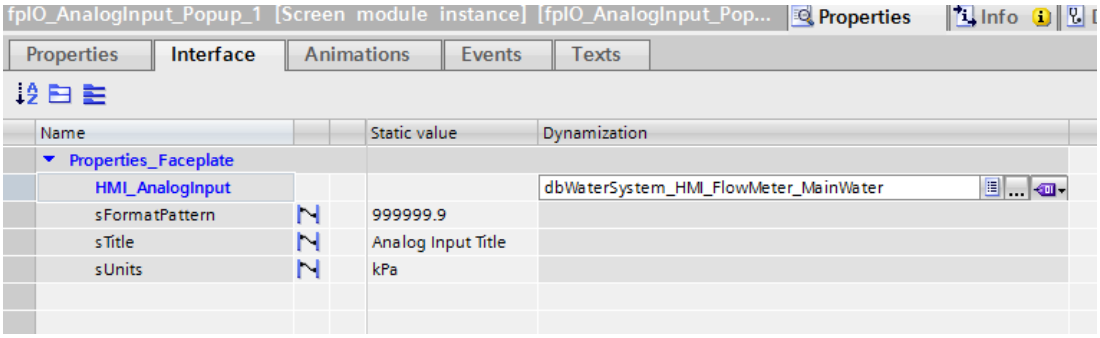
Object	Type	Description
Auto Mode Button	Button	Sets device to Auto Mode. Only available when global mode is Independent. Button is green when devices is in Auto Mode.
Manual Mode	Button	Sets device to Manual Mode. Only available when global mode is Independent. Button is green when device is in Manual Mode.
Override Value	Input	Adjusts the current value of the device
Input Value	Output	Displays the current reading of the device
Active Value	Output	Displays the value being currently used – the Override Value in Manual Mode, or Input Value in Auto Mode
Error Status Field	I/O Field	Displays current error status. Field box turns red if any errors are present, and displays active error text. It will scroll through multiple alarms
RESET Button	Button	Resets errors
EXIT Button	Button	Exits the pop up screen

9.1.5.3. Interface

The device interface is shown below. Four properties are linked in this interface:

- HMI_AnalogInput: Needs to be mapped to the same 'udt_HMI_AnalogInput' used by the Function Block

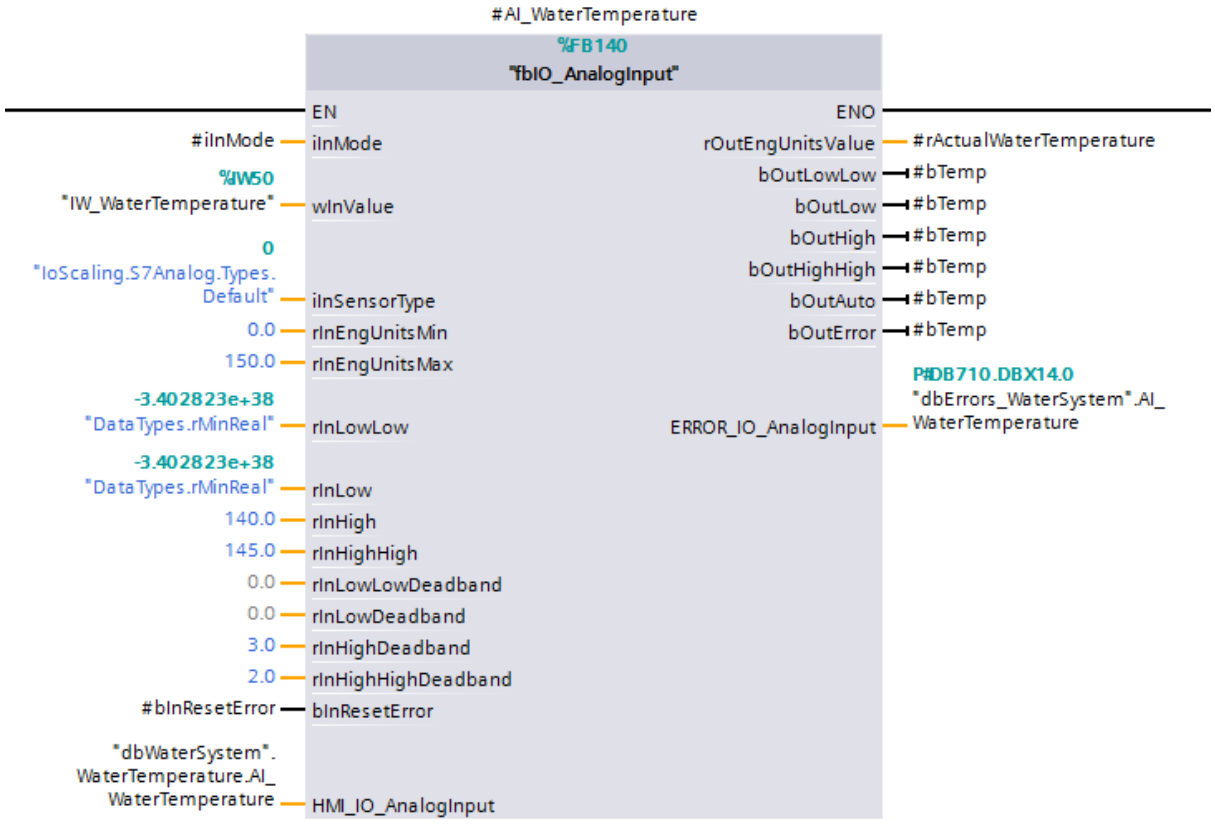
- sFormatPattern: Changes data format of values, setting amount of leading and trailing zeros of outputs. Adjusted directly in table
- sTitle: Sets title of the pop up display. Adjusted directly in table
- sUnits: Sets the units used in the display. Adjusted directly in table



9.2. Analog Output with Scaling – fbIO_AnalogOutput

9.2.1. Description

This library object scales process values to be used for Analog Outputs.



9.2.2. Function Block Interface

9.2.2.1. Input Parameters

Input Variables	Type	Description
iInMode	Int	Mode input. See Section 12.1.2 of the Library Overview and Architecture document for further details
iInSensorType	Int	Enumerated value of sensor types (IoScaling.S7Analog.Types)
iInEstopFunction	Int	Enumerated value of E-Stop response constant. 'Estop.MinSetPoint' sets wOutValue to the minimum set point (rInEngUnitsMax), 'Estop.MaxSetPoint' sets to the maximum set point (rInEngUnitsMin), and 'Estop.HMISetPoint' sets to an HMI-specified value that can be in between the min and max.
bInEstop	Bool	Emergency stop input
bInEnable	Bool	Interlock for the Analog Output. If this bit is false, then output will be 0.
rInValue*	Real	Auto mode value of output
rInEngUnitsMin	Real	Engineering units scale min
rInEngUnitsMax	Real	Engineering units scale max

bInResetError	Bool	If an error condition exists, the error must be fixed first, then set this bit high to reset the internal error

*Only valid in Automatic Mode

9.2.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_IO_AnalogOut	udtHMI_AnalogOutput	This In/Out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to configure additional settings and provide status to the HMI

9.2.2.3. Out Parameters

Output Variables	Type	Description
bOutAuto	Bool	Block in auto mode
bOutError	Bool	Error exists
wOutValue	Word	Value to output to Analog Output
ERROR_IO_AnalogOut	udtError_AnalogOut	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state

9.2.3. User Defined Types

9.2.3.1. udtHMI_AnalogOutput

Name	Type	Description
iMode	Int	Current mode
iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details
rValue	Real	Output Value
rAutoValue	Real	Auto mode value
rManualValue	Real	Manual mode value
rEstopValue	Real	Estop condition custom value
bPB_ResetError	Bool	Reset errors pushbutton
bPBEN_ResetError	Bool	Reset error pushbutton enabled
bError	Bool	Overall error
bInterlock	Bool	Analog output is interlocked
iEstopFunction	Int	Enumerated value of E-Stop response constant.

9.2.3.2. udtError_AnalogOutput

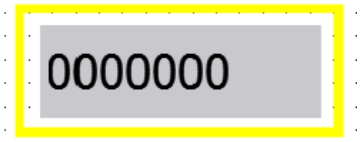
Errors	Description
Out_Of_Range	Input Value is outside range of Engineering Units
Configuration_Error	Engineering Units are not properly configured

9.2.4. HMI Icon Display

The HMI Icon is used on the main HMI screen to provide a brief overview of the status of the device. Custom devices can be made to match specific applications.

9.2.4.1. HMI Icon Display Objects

The library contains the following object to be used on the HMI screen:



9.2.4.2. HMI Icon Appearance and Functionality

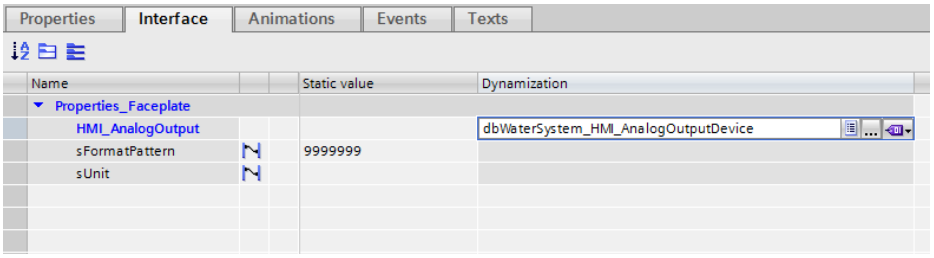
The device appearance's changes depending on the state it is in, see table below for description:

Indicator	Meaning
I/O Field Value	Current value sent to device
Yellow Border	Device interlocked

9.2.4.3. HMI Icon Interface

The device interface is shown below. It requires three properties to be linked:

- HMI_Analog_Output: Needs to be mapped to the same 'udtHMI_AnalogOutput' used by the Function Block
- sFormatPattern: Changes data format of values, setting amount of leading and trailing zeros of outputs. Adjusted directly in table
- sUnit: Sets the units used in the display

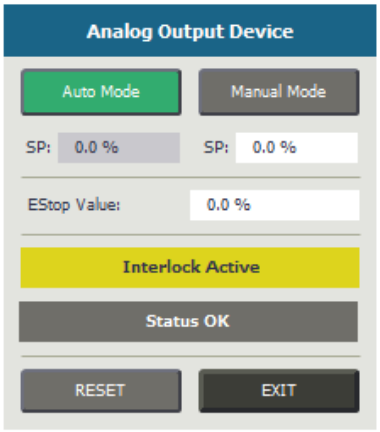


9.2.5. HMI Pop-up Faceplate

The HMI pop up provides a detailed display of the status of the Analog Output device. Additionally, it allows for control of the device when the System Mode is Independent or Manual.

9.2.5.1. HMI Pop-up Display

The library contains a single pop-up faceplate shown below:



9.2.5.2. HMI Pop-up Appearance and Functionality

The table below gives a feature description for the pop up faceplate:

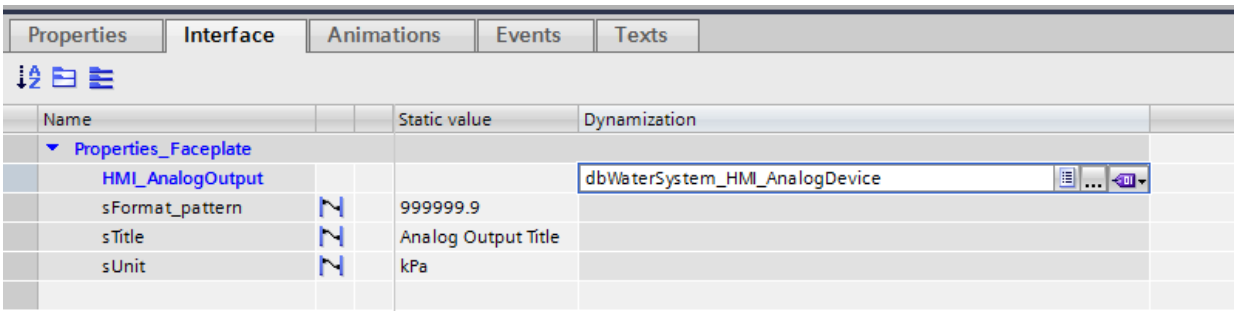
Object	Type	Description
Auto Mode Button	Button	Sets device to Auto Mode. Only available when global mode is Independent. Button is green when devices is in Auto Mode.
Manual Mode	Button	Sets device to Manual Mode. Only available when global mode is Independent. Button is green when device is in Manual Mode.
Auto Mode SP	Output	Displays PLC set point to be used in Auto Mode.
Manual Mode SP	Input	Adjusts the speed set point when in Manual Mode
EStop Value	I/O Field	Displays the current value of the Estop reading
Interlock Status Field	I/O Field	Displays current status of device interlock (yellow for interlock, grey when not)
Error Status Field	I/O Field	Displays current error status. Field box turns red if any errors are present, and displays active error text. It will scroll through multiple alarms
RESET Button	Button	Resets errors
EXIT Button	Button	Exits the pop up screen

9.2.5.3. HMI Pop-up Interface

The device interface is shown below. Two properties are linked in this interface:

- HMI_AnalogOutput: Needs to be mapped to the same 'udt_HMI_AnalogOutput' used by the Function Block

- sFormatPattern: Changes data format of values, setting amount of leading and trailing zeros of outputs. Adjusted directly in table
- sTitle: Sets title of the pop up display. Adjusted directly in table
- sUnit: Sets the units used in the display



10. General Control

10.1. System Control

10.1.1. Description

The System Control Library Object consists of a UDT, HMI Icon, and HMI Faceplate.

10.1.2. User Defined Types

10.1.2.1. udtHMI_SystemControl

Name	Data Type	Description
iMode	Int	System mode
bEStop	Bool	System hardware E-Stop
bError	Bool	Actuators error present
bAuto	Bool	All actuators in auto mode
bIndependentEnable	Bool	Independent mode enabled
bPB_ResetError	Bool	Reset error push button

10.1.3. HMI Icon Display

10.1.3.1. HMI Icon Display Objects

The library contains the following object to be used on the HMI screen:

10.1.3.2. HMI Icon Appearance and Functionality

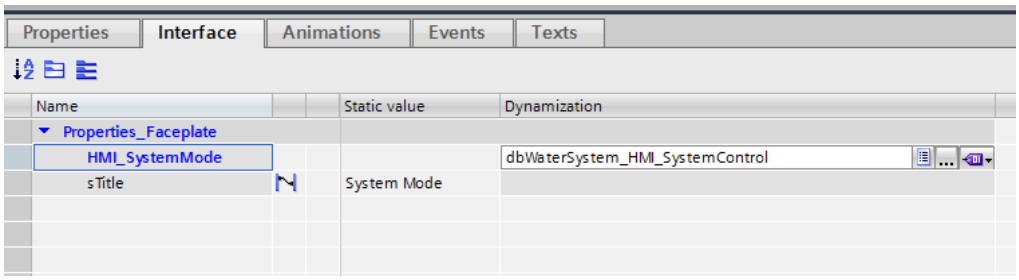
The status text and color updates according to the current running state of the device:

Device Mode	System Color
Stop	Grey
Auto	Green
Manual	Blue
Independent	Yellow

10.1.3.3. HMI Icon Interface

The device interface is shown below. The following properties are required to be linked:

- HMI_SystemMode: Needs to be mapped to the 'udt_HMI_SystemControl' for the given system.
- sTitle: Sets title of the pop up display. Adjusted directly in table.

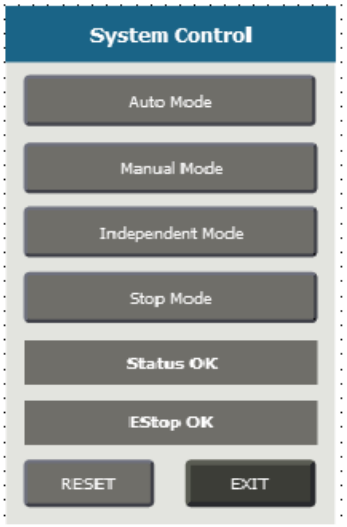


10.1.4. HMI Pop-up Faceplate

The HMI pop up provides a display of the system's status. Additionally, it provides operator control for the overall system mode.

10.1.4.1. HMI Pop-up Display

The library contains a single pop-up faceplate shown below:



10.1.4.2. HMI Pop-up Appearance and Functionality

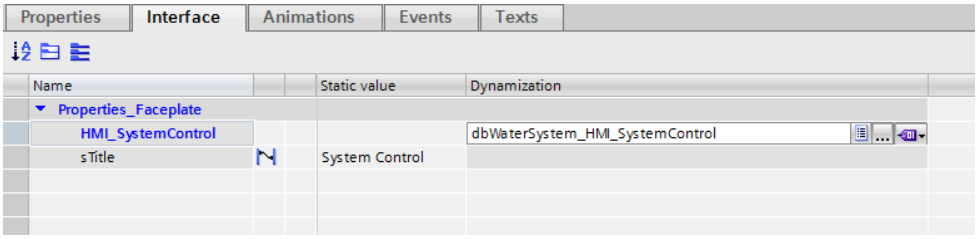
The table below gives a feature description for the pop up faceplate:

Object	Type	Description
Auto Mode	Button	Sets system to Auto Mode
Manual Mode	Button	Sets system to Manual Mode
Independent Mode	Button	Sets system to Independent Mode
Stop Mode	Button	Sets system to Stop Mode
Interlock Status Field	I/O Field	Displays current system alarm status (red for alarm, grey for System Ok)
E-Stop Status Field	I/O Field	Displays current E-stop status. Field box turns red if an E-stop error is present, and displays active error text.
RESET Button	Button	Resets errors
EXIT Button	Button	Exits the pop up screen

10.1.4.3. HMI Icon Interface

The device interface is shown below. The following properties are required to be linked:

- HMI_SystemMode: Needs to be mapped to the 'udt_HMI_SystemControl' for the given system
- sTitle: Sets title of the pop up display. Adjusted directly in table

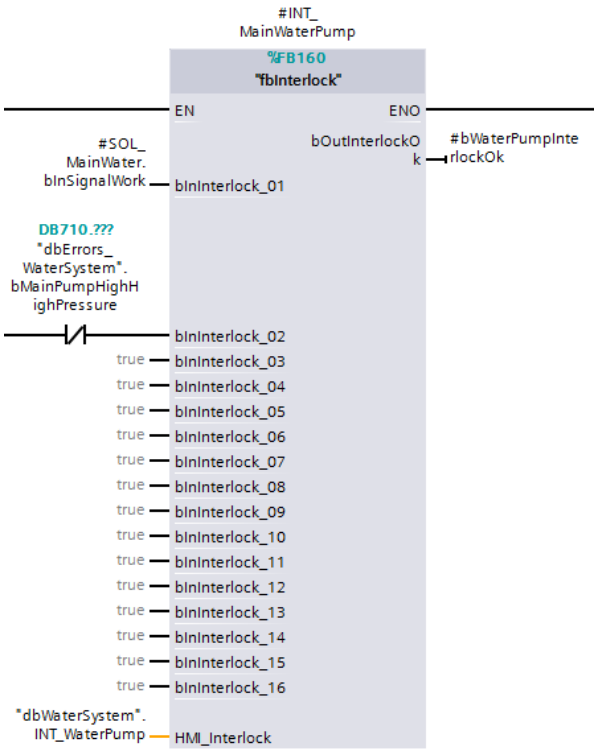


-

10.2. Interlock Function Block – fbInterlock

10.2.1. Description

This library object is used to provide operator information for multiple values that are interlocking a process. It has inputs for items that may be interlocking a process, and provides operator information to inform the operator of what interlocks are preventing operation.



10.2.2. Function Block Interface

10.2.2.1. Input Parameters

Input Variables	Type	Description
bInInterlock_01	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_02	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_03	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_04	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_05	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_06	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True

bInInterlock_07	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_08	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_09	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_10	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_11	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_12	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_13	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_14	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_15	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True
bInInterlock_16	Bool	Interlock for the system. When value is 1, system is okay to operate. Disable input by not mapping or putting in a value of True

10.2.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_Interlock	udtHMI_Interlock	This In/Out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to configure additional settings and provide status to the HMI

10.2.2.3. Out Parameters

Output Variables	Type	Description
bOutInterlockOK	Bool	Value of 1 means system is okay to operate

10.2.3. User Defined Types

10.2.3.1. udtHMI_Interlock

Name	Data Type	Description
abInterlocks	Array[1..16] of Bool	Interlock statuses

asInterlockNames	Array[1..16] of String[20]	Interlock names
bInterlockOk	Bool	System Interlock is ok

10.2.4. HMI Icon Display

The Interlock block is configured so that the Faceplate can be tied to a faceplate of another object, and allow viewing from other library pop-ups.

10.2.4.1. HMI Icon Display Objects

The library contains the following display to be used on the HMI screen:

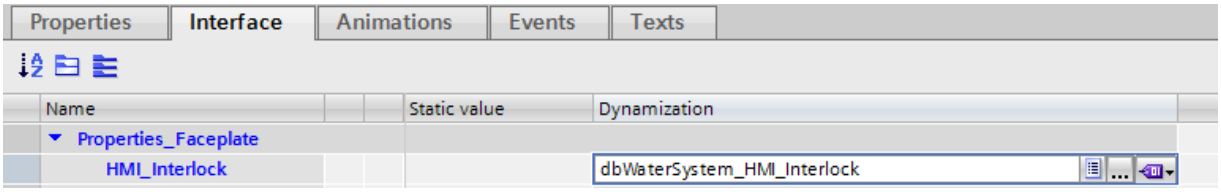
10.2.4.2. HMI Icon Appearance and Functionality

The device appearance's changes depending on the state it is in, see table below for description:

Indicator	Meaning
Device Color	Yellow: Device interlocked Grey: Device not interlocked

10.2.4.3. HMI Icon Interface

The device interface is shown below. Only one property is required to be linked: the HMI_Interlock property. It needs to be mapped to the same 'udtHMI_Interlock' used by the Function Block.



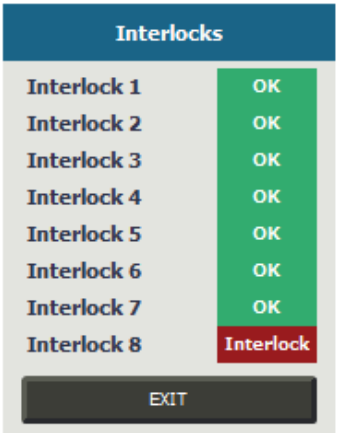
10.2.5. HMI Pop-up Faceplate

The HMI pop up provides a display of the status of each interlock in the system

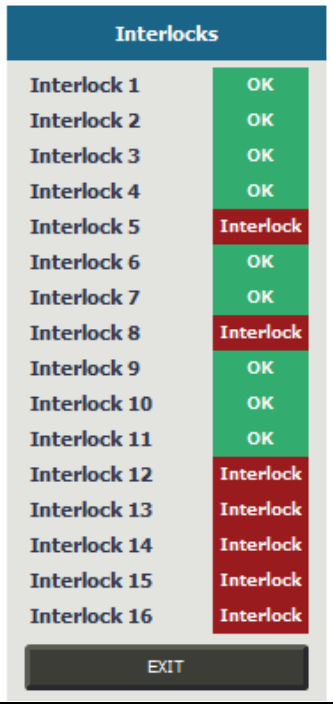
10.2.5.1. HMI Pop-up Display

The library contains two pop-up faceplates shown below:

Interlock8



Interlock16



Both faceplates use the same UDT- 'udtHMI_Interlock'. fpInterlock8 is to be used when you have 8 or fewer devices requiring interlocks.

10.2.5.2. HMI Pop-up Appearance and Functionality

The table below gives a feature description for the pop-up faceplate:

Object	Type	Description
Device Status Field	Output	Device OK: Displays 'OK' in a green box Device Interlocked, unable to run: Displays 'Interlock' displayed in Red Box
EXIT Button	Button	Exits the pop-up screen

10.2.5.3. HMI Pop-up Interface

The device interface is shown below. Two properties are linked in this interface: HMI_Interlock and sTitle. The HMI_Interlock needs to be mapped to the same 'udtHMI_Interlock' used by the Function Block. The sInterlockTitle property requires a string passed into it, and this value will display at the top of the Pop-up.

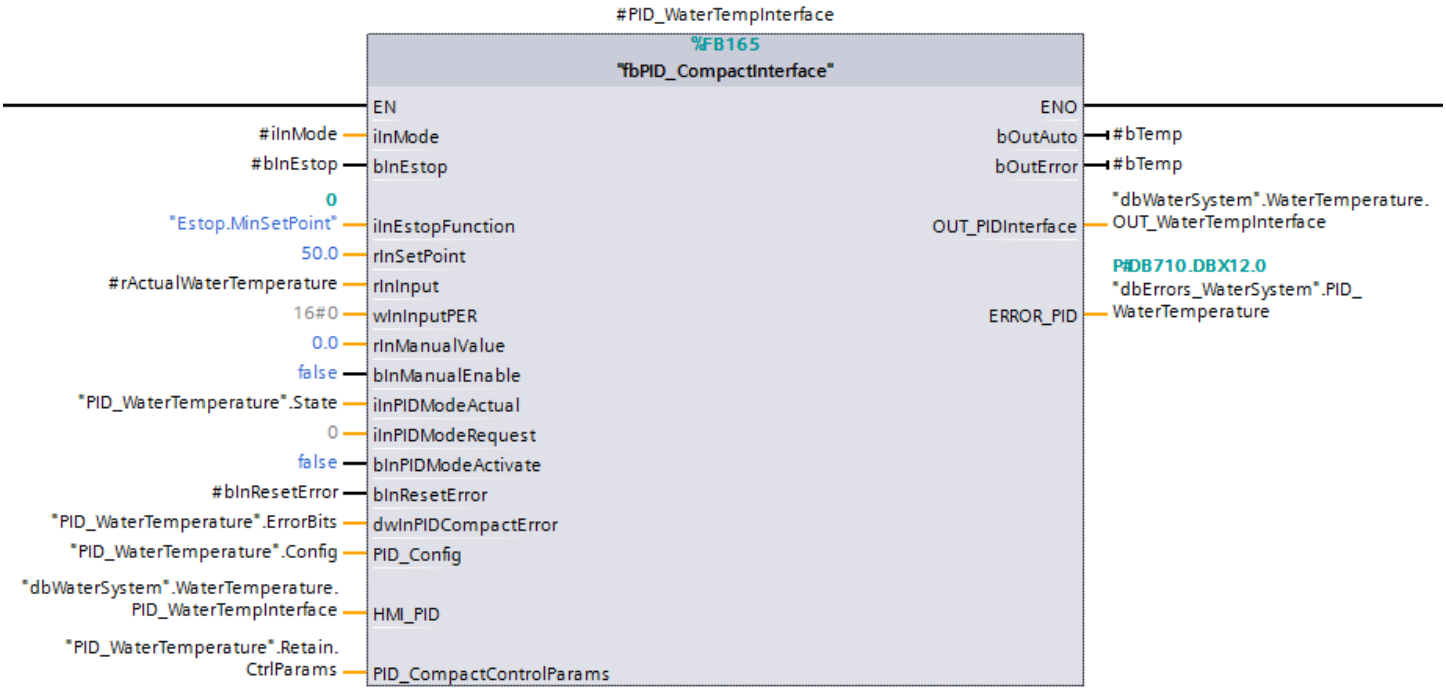
10.3. PID Interface - fbPID_CompactInterface

10.3.1. Description

The S7-1200 and S7-1500 provide PID Technology objects that are best practices for use, therefore this would not replace the blocks, but would provide an interface to the technology object on the HMI.

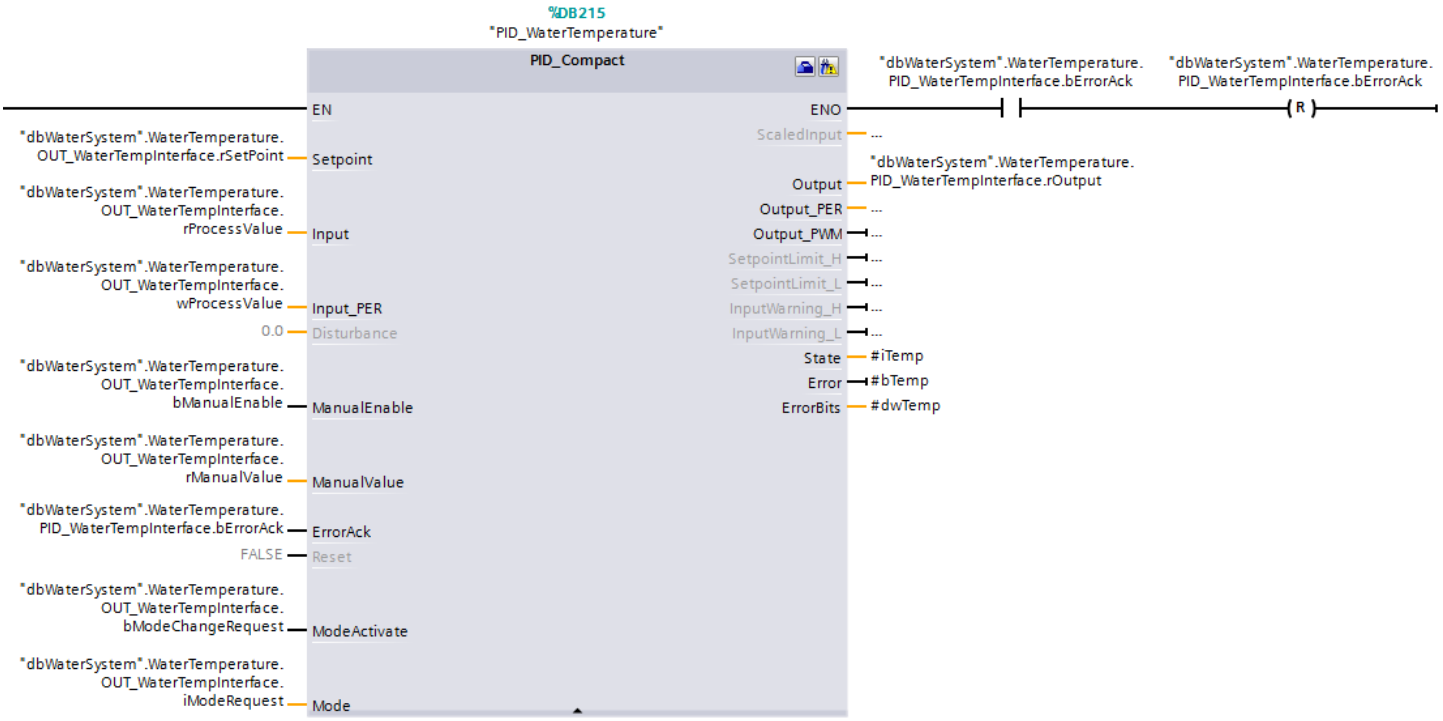
The interface will allow for modification of the PID loop and for manual override of the output from the HMI.

Unlike other library objects that are fully encapsulated, many of the output of this block will need to be mapped to the PID_Compact block.



This block is used to handle HMI mode selection and control as well as scale inputs, handle e-stop conditions, and allow manual and automatic tuning from the HMI.

The PID_Compact block must still be called from a cyclic interrupt OB. The interface tags used here should be passed from the output of the fbPID_CompactInterface block with the exception of the ErrorAck bit, which must be reset in the cyclic interrupt.



10.3.2. Function Block Interface

10.3.2.1. Input Parameters

Input Variables	Type	Description
iInMode	Int	Sets the mode of the block. When the block is in auto mode, it utilizes rInSetpoint. When the block is in manual mode, it reads commands from the HMI through HMI_PIDInterface. . See Section 12.1.2 of the Library Overview and Architecture document for further details
bInEstop	Bool	Emergency stop input
iInEstopFunction	Int	Enumerated value of E-Stop response constant. 'Estop.MinSetPoint' sets Output to the minimum set point (PID output maximum), 'Estop.MaxSetPoint' sets to the maximum set point (PID output minimum), and 'Estop.HMISetPoint' sets to an HMI-specified value that can be in between the min and max.
rInSetpoint*	Real	PID set point
rInInput	Real	Scaled Process value (0-100%) Use this value or wInInputPer
wInInputPER	Word	Raw Process value. Use this value or rInInput
rInManualValue*	Real	Used by automation logic to override PID and provide a manual value through PLC logic
bInManualEnable*	Bool	rInManualValue is used to override the PID output value when this is true
iInPIDModeActual	Int	Actual mode of the PID_Compact technology object. This should be mapped from the technology object data block
iInPIDModeRequest	Int	PID_Compact Mode as described by the PID compact. Used by automation logic to enable the override of the PID_Compact mode
bInPIDModeActivate	Bool	PID_Compact Mode will change on a rising edge of this bit based on iInPIDModeRequest
bInResetError	Bool	If an error condition exists, the error must be fixed first, then set this bit high to reset the internal error

dwInPIDCompactError	DWord	This is the output from the PID Compact Block and is used to create meaningful alarms. This should be mapped from the technology object data block
PID_Config	PID_CompactConfig	PID_Compact structure used to provide configuration data. This should be mapped from the technology object data block.

*Only valid in Automatic Mode

10.3.2.2. InOut Parameters

In/Out Variables	Type	Description
HMI_PID	udtHMI_PIDInterface	This in/out UDT is used to communicate to the HMI. Inside the UDT are all the variables needed to control the PID in manual mode
PID_CompactControlParams	PID_CompactControlParams	PID_Compact structure used to reference and modify tuning parameters. This should be mapped from the technology object data block

10.3.2.3. Out Parameters

Output Variables	Type	Description
ERROR_PID	udtError_PID	This output UDT has one bit for every type of error. It should be mapped to the HMI's error list. It can also be used for advanced debugging of the error state
OUT_PIDInterface	udtOUT_PIDInterface	The output UDT is used to map standard variables to the inputs of the PID_Compact block

10.3.3. User Defined Types

10.3.3.1. udtHMI_PIDControlParameters

Name	Type	Description
rProportionalGain	Real	Proportional gain coefficient
rIntegralTime	Real	Integral control time
rDerivativeTime	Real	Derivative control time
rProportionalWeight	Real	Proportional control weighting
rDerivativeWeight	Real	Derivative control weighting

10.3.3.2. udtOUT_PIDInterface

Name	Type	Description
rProcessValue	Real	Process value
wProcessValue	Word	Peripheral process value
rSetPoint	Real	PID control set point
rManualValue	Real	PID manual override value
iModeRequest	Int	Requested PID_Compact override mode

bManualEnable	Bool	Enable PID manual override value
bModeChangeRequest	Bool	Request a PID_Compact mode change

10.3.3.3. udtHMI_PIDInterface

Name	Type	Description
iMode	Int	Current mode
iPIDMode	Int	PID compact mode
iErrorCode	Int	Error code. See Section 12.3.1 of the Library Overview and Architecture document for further details
iEstopFunction	Int	Enumerated value of E-Stop response constant.
rEstopValue	Real	PID value for an e-stop if custom value is selected
rManualSP	Real	PID set point for manual mode
rAutoSP	Real	PID set point for automatic mode
rActualInput	Real	Actual PID input value
wActualInputPeripheral	Word	Actual PID input peripheral value
rOutput	Real	PID output value
wOutputPeripheral	Word	PID output peripheral value
bPB_ResetError	Bool	Reset errors pushbutton
bPB_Pretune	Bool	Start pretuning pushbutton
bPB_FineTune	Bool	Start finetuning pushbutton
bPB_AutomaticMode	Bool	Switch to automatic mode pushbutton
bPB_ManualMode	Bool	Switch to manual mode pushbutton
bPB_ManualTuning	Bool	Switch to manual PID tuning pushbutton
bTOG_ManualTuning	Bool	TOG Manual tuning enabled
bPBEN_ResetError	Bool	Reset errors pushbutton enabled
bPBEN_Pretune	Bool	Pretuning pushbutton enabled
bPBEN_FineTune	Bool	Fine tuning pushbutton enabled
bPBEN_AutomaticMode	Bool	Automatic mode pushbutton enabled
bPBEN_ManualMode	Bool	Manual mode pushbutton enabled
bError	Bool	Overall error
bErrorAck	Bool	PID_Compact Error Reset Command

10.3.3.4. udtError_PID

Name	Description
InputOutOfRange	Input value is out of the configured range
InputPERInvalid	InputPER value is invalid

ValueOscillationFailed	Fine tuning - process value oscillation could not be maintained
ProcessValueCloseToSetPoint	Pre-tuning - process value is too close to set point
SetPointChangedDuringTuning	PID set point was changed during tuning
PretuningDuringFineTuning	Pre-tuning not allowed while fine tuning is active
InvalidOutputValueLimits	Pre-tuning - invalid configuration of output value limits
InvalidFineTuningParameter	Fine tuning - error occurred causing invalid parameters
InputInvalidFormat	Input value has an invalid number format
OutputCalculationError	Output value calculation error occurred
SamplingTimeError	PID_Compact not called within sampling time of cyclic interrupt OB
SetPointInvalidFormat	Set point value has an invalid number format
ManualInvalidFormat	Manual value has an invalid number format
SubstituteOutputInvalidFormat	Substitute output value has an invalid number format
DisturbanceInvalidFormat	Disturbance value has an invalid number format

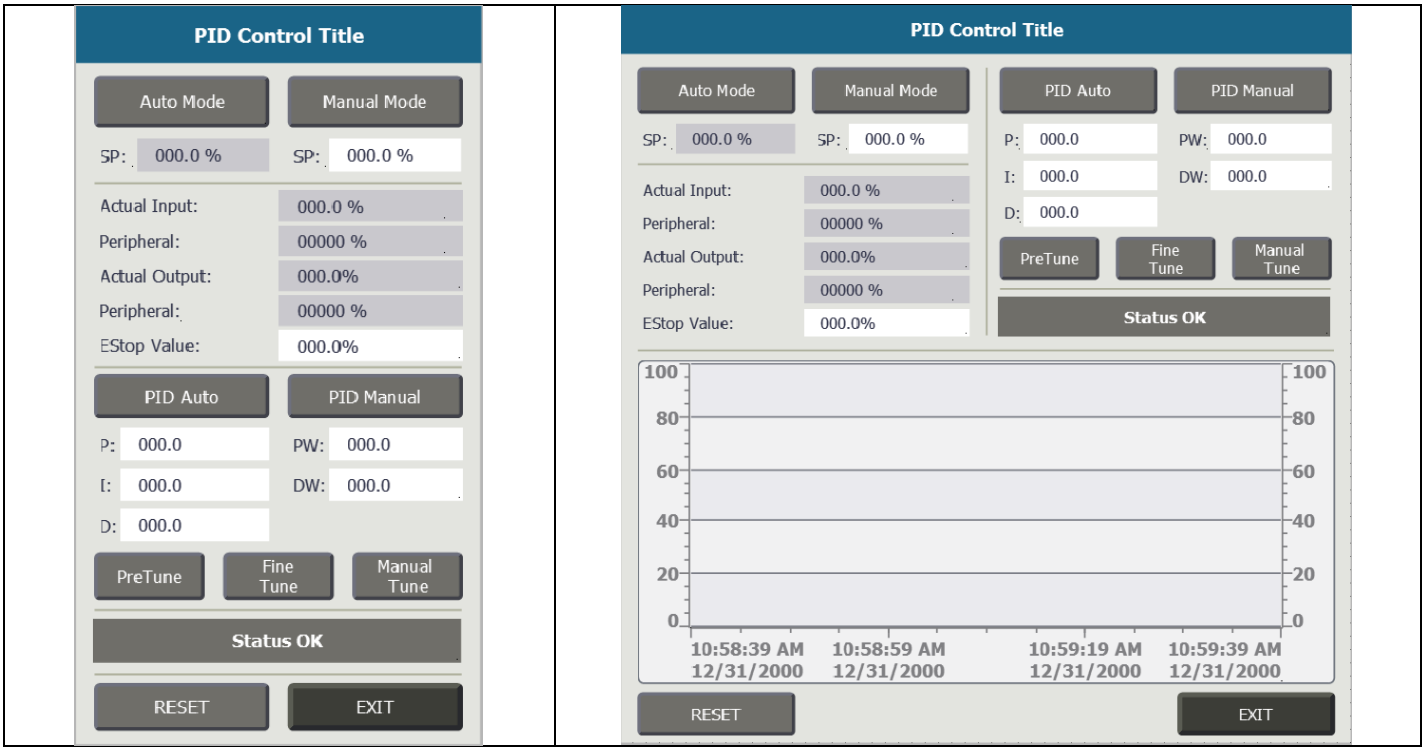
10.3.4. HMI Pop-up Faceplate

The HMI pop up provides a detailed panel for PID control. There are two pop-up faceplates available: fpPID_CompactInterface_Popup and fpPID_CompactInterface_Popup_Graph.

10.3.4.1. HMI Pop-up display

The library contains two pop-up faceplates, shown below:

PID Control Pop-up	PID Control Pop-up w/Trend Graph



10.3.4.2. HMI Pop-up Appearance and Functionality

The table below gives a feature description for the pop up faceplates:

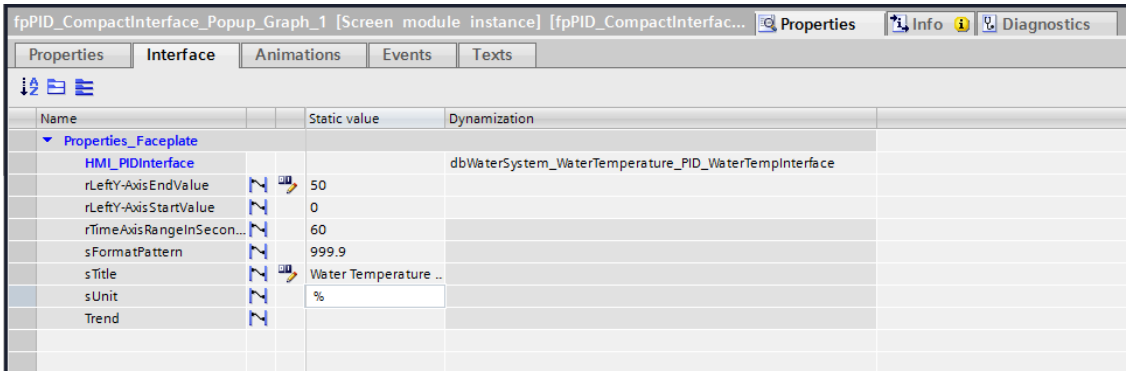
Object	Type	Description
Auto Mode Button	Button	Sets device to Auto Mode. Only available when global mode is Independent. Button is green when devices is in Auto Mode.
Manual Mode	Button	Sets device to Manual Mode. Only available when global mode is Independent. Button is green when device is in Manual Mode.
Auto Mode SP	Output	Displays PLC set point to be used in Auto Mode.
Manual Mode SP	Input	Adjusts the speed set point when in Manual Mode
Actual Input	I/O Field	Displays the current speed of the motor (%)
Peripheral	I/O Field	Displays motor current (A)
Estop Value	I/O Field	Displays the current value of the Estop reading
PID Auto Button	Button	Sets PID control to Auto
PID Manual Button	Button	Sets PID control to Manual
P Field	I/O Field	Proportional gain value
I Field	I/O Field	Integral gain value
D Field	I/O Field	Derivative gain value
PW Field	I/O Field	Proportional Weight
DW Field	I/O Field	Derivative Weight
PreTune Button	Button	Initializes a pre-tune in the PID_Compact Block

Fine Tune Button	Button	Initializes a fine tune in the PID_Compact Block
Manual Tune Button	Button	Initializes a manual tune in the PID_Compact Block
RESET Button	Button	Resets the device
EXIT Button	Button	Exits the pop up screen

10.3.4.3. HMI Pop-up Interface

The device interface is shown below. The following properties are linked to the interface:

- HMI_PIDInterface: Needs to be mapped to the same 'udtHMI_PIDInterface' used by the Function Block
- rLeftY-AxisEndValue: Maximum value for the left Y-Axis of the Trend Graph. This will typically be the maximum value of the variable that is being controlled, water temperature for this example. Adjusted directly in table. Note: This is only used for the PID Control pop-up with Trend Graph
- rLeftY-AxisStartValue: Minimum value for the left Y-Axis of the Trend Graph. This will typically be the minimum value of the variable that is being controlled, water temperature for this example. Adjusted directly in table. Note: This is only used for the PID Control pop-up with Trend Graph
- rTimeAxisRangeInSeconds: Defines the number of seconds being shown on the Trend graph. Adjusted directly in table. Note: This is only used for the PID Control pop-up with Trend Graph
- sFormatPattern: Changes data format of values, setting amount of leading and trailing zeros of outputs. Adjusted directly in table
- sTitle: Sets title of the pop-up display. Adjusted directly in table
- sUnit: Sets the units used in the display. Adjusted directly in table
- Trend: This is configured by right clicking on the trend on the pop up and adding the relevant tags to the trend. Note: This is only used for the PID Control pop-up with Trend Graph

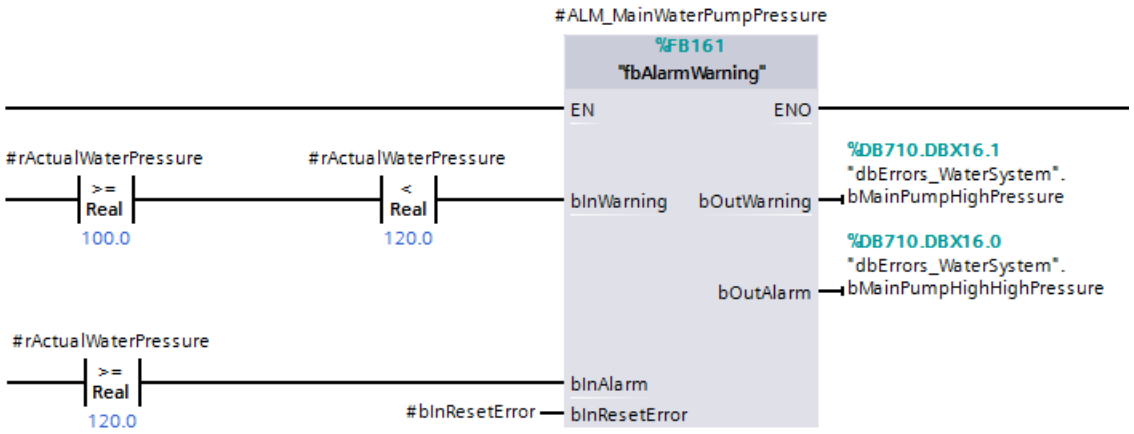


*Note: If custom values are required for both the right and left Y-Axis, use the PID control without the Trend Graph and add a new one in manually. The right Y-Axis for the pop-up with the Trend Graph is set to default to values between 0 and 100, representing the percentage of the PID control.

10.4. Standard Alarm Interface – fbAlarmWarning

10.4.1. Description

The Standard Alarm Interface provides an easy way to setup additional alarms. Alarms will latch until reset.



10.4.2. Function Block Interface

10.4.2.1. Input Parameters

Input Variables	Type	Description
bInWarning	Bool	Warning Signal. Provides warning when signal is true
bInAlarm	Bool	Alarm Signal. Provides alarm when signal is true
bInResetError	Bool	If an error condition exists, the error must be resolved first, then set this bit high to reset the internal error

10.4.2.2. Out Parameters

Out Variables	Type	Description
bOutWarning	Bool	Signal used to show warning is active
bOutAlarm	Bool	Signal used to show alarm is active and has not been reset

10.4.3. HMI Icon Display

The HMI interface is done through the Alarm banner. There is no faceplate for this object. For information on how to utilize alarms, see 'Siemens Open Library – Siemens Alarm Generation' document.