Library for Basic Processes (LBP) – Implementation

STEP 7 Basic / Professional V16, WinCC Comfort / Professional V16, WinCC V7.5 SP1, WinCC Open Architecture 3.16

https://support.industry.siemens.com/cs/ww/en/view/109749508

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG (“Siemens”). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.
By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.
The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens’ products and solutions constitute one element of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.
For additional information on industrial security measures that may be implemented, please visit **Fehler! Linkreferenz ungültig.**.
Siemens’ products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer’s exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: **Fehler! Linkreferenz ungültig.**.

Table of Contents

1 Introduction

1.1 Overview

The Library for Basic Processes provides support in visualizing various functions.

Advantages of the library:

- Uniform design
- Less engineering work

Many projects have the same elements that have to be visualized. These include motors, analog values and valves, for example. You can find the following blocks in the library:

Table 1-1

Block	Brief description
LBP_Aggr8	Control of up to 8 Aggregates
LBP_AnaAvg	Average calculation with plausibility check
LBP_AnaRead	Analog value processing
LBP_ClctVal	Group display
LBP_CntrA	Integrating counter
LBP_CntrD	Difference counter
LBP_CntrP	Pulse counter
LBP_CtrlPID	Continuous PID controller
LBP_CtrlStp	Step controller
LBP_Intlk8	Interlock for up to 8 signals
LBP_Msg8	Message block for 8 messages
LBP_Mtr	Simple motor
LBP_MtrDS	Motor with 2 speeds and directions of rotation
LBP_MtrF	Speed-controlled motor
LBP_OpAna	Switching of analog values
LBP_OpDig	Switching of a digital signal
LBP_Polygon	Conversion of values over 8 interpolation points
LBP_Select	Selection of values
LBP_SetCrv	Time-dependent set point curve
LBP_TimeSw	Time switch
LBP_Vlv	Simple valve
LBP_VlvA	Analog controlled valve
LBP_3wVlv	3-way valve

Innovations with V2.0

You obtain a fundamentally revised library with V2.0.

Aims of the revision

- Performance improvement
- Improvement of usability during configuration
- Improvement for more intuitive operation
- Sustainable programming and configuration of the library

1.2 Hardware and Software Requirements

Requirements for this library

In order to use the functionality of the library described here, the following hardware and software requirements must be met.

Hardware

All blocks (FB, FC, DB, etc.) in the library can be used universally with the following controllers:

- S7-1200 and S7-1200 F product family
- S7-1500 and S7-1500 F product family
- Simulation with S7-PLCSIM (as of V14)
- Comfort Panels

When using WinCC Professional, WinCC V7 or
WinCC Open Architecture a computer is required.

Software

- STEP 7 (TIA Portal) Basic or Professional V15

At least one of the following WinCC versions is required:

- WinCC Professional and Runtime Professional V15
- WinCC Comfort/Advanced V15 and Comfort Panels
- WinCC Advanced V15 and WinCC Runtime Advanced V15
- WinCC V7.5, Update 3 or higher
- WinCC Open Architecture 3.16

2 Integrating the PLC Function Blocks

2.1 Structure of the PLC Function Blocks

2.1.1 Nominators Across Modules

“indXXX”

If a tag name starts with “ind”, it gives feedback via an actuator.
Example: With the Mtr, the LBP block uses this input to check whether the motor has actually been switched on after a switch-on signal has been output.

“cmdXXX”

If a tag name starts with “cmd”, the actuator is controlled by this tag. These inputs are mainly required for automatic operation.

2.1.2 Cross-Block Input “identName”

The identification name (“identName”) is displayed as a header in the visualization. This is also used to filter the messages in the block-specific message windows. For this reason, each identification name should only exist once.

Note

You save lots of engineering work during the SiVArc generation of symbols and alarms in WinCC Comfort if you name the instance of the LBP block according to its “identName”. You have the following naming options:

- instIdentName
- InstIdentName
- IdentName

2.1.3 User-Defined Data Types for Communication with the Visualization System

settingsHMI

The user-defined data type “settingsHMI” only contains tags that can be written by the HMI system.

The following tags are included across all blocks:

- “note”: This tag contains the comment which is displayed below the header.
- “opStation”: This tag contains the operating station and is used for locking the control. If you want to be the only one on the HMI system who can currently operate, the station name of the HMI system is stored here.
- “overwrite”: For some values, the HMI system can be used to determine whether the value at the input of the block in the PLC program or the value which can be written from the HMI system should be used. For a value, some bits of the “overwrite” tag specify whether the PLC value or the HMI value is active. Other bits include commands such as “run” and “stop”. The following section describes, among others, the bit assignment of the “overwrite” tags. Bits that determine whether a PLC or HMI value is active are marked with “over”. “True” here means that the HMI value is active. Bits that are commands are marked with “cmd”.

settingsPLC

The user-defined data type “settingsPLC” contains the values of the tags that were interconnected as the input at the block in the PLC program.
It always contains the “identName” input.

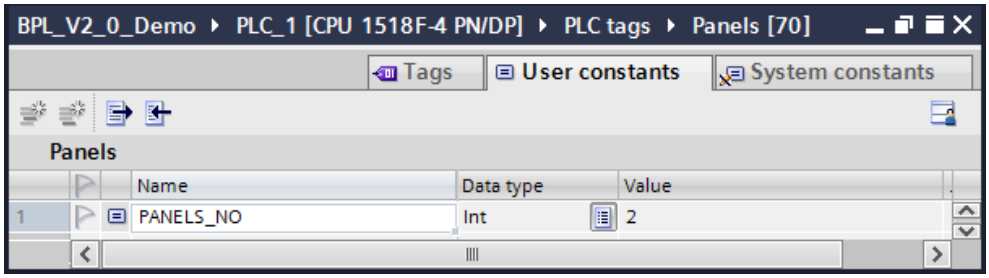
statusHMI

The user-defined data type “statusHMI” contains the values with which the block currently works and the values that the block outputs at the outputs in the PLC program.

2.1.4 Global Constant “PANELS_NO”

The global constant “PANELS_NO” has to be created so that the function blocks of the LBP operate correctly. With this constant, you determine how many Comfort Panels are connected to your control system to visualize LBP blocks. If you do not use any Comfort Panels, specify “0” as the static value.

Figure 2-1



2.1.5 Input/Output for Panel Communication

The following input/output parameter is used so that only one dataset must be integrated into the HMI tags per panel for each LBP block—even if the block is called up more often in the program.

Table 2-2

Parameters	Data type	Description
panels	Array [*] of UDT	Array for all Panels that are connected. For each connected panel, the array index of the connected array has to be increased by 1.
identName	String[30]	Identification name of the instance that was called up at the panel
opStation	String[16]	Operating station name
note	String[80]	Comment that is shown below the header
log15	UDT	Log data for HMI
3wVlv	UDT	Currently called-up instance of 3wVlv
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program

Parameters	Data type	Description
alarms1	Word	Bit alarm list for Comfort/Advanced
alarms2	Word	Bit alarm list for Comfort/Advanced
aggr8		Currently called-up instance of aggr8
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
alarms1	Word	Bit alarm list for Comfort/Advanced
alarms2	Word	Bit alarm list for Comfort/Advanced
anaAvg		Currently called-up instance of anaAvg
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program.
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
alarms1	Word	Bit alarm list for Comfort/Advanced
alarms2	Word	Bit alarm list for Comfort/Advanced
anaRead		Currently called-up instance of anaRead
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
alarms1	Word	Bit alarm list for Comfort/Advanced
Cntr	UDT	Currently called-up instance of Cntr
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
crtlPID	UDT	Currently called-up instance of ctrlPID
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
crtlStp	UDT	Currently called-up instance of ctrlStp
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the

Parameters		Data type	Description
			values that the block outputs at the outputs in the PLC program
intlk8		UDT	Currently called-up instance of intlk8
	settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
	settingsHMI	UDT	Tags that can be written by the HMI system
	statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
msg8		UDT	Currently called-up instance of msg8
	settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
	settingsHMI	UDT	Tags that can be written by the HMI system
	statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
mtr		UDT	Currently called-up instance of mtr
	settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
	settingsHMI	UDT	Tags that can be written by the HMI system
	statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
	alarms1	Word	Bit alarm list for Comfort/Advanced
	alarms2	Word	Bit alarm list for Comfort/Advanced
mtrDS		UDT	Currently called-up instance of mtrDS
	settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
	settingsHMI	UDT	Tags that can be written by the HMI system
	statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
	alarms1	Word	Bit alarm list for Comfort/Advanced
	alarms2	Word	Bit alarm list for Comfort/Advanced
mtrF		UDT	Currently called-up instance of mtrF
	settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
	settingsHMI	UDT	Tags that can be written by the HMI system
	statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
	alarms1	Word	Bit alarm list for Comfort/Advanced
opAna		UDT	Currently called-up instance of opAna
	settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
	settingsHMI	UDT	Tags that can be written by the HMI system
	statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
opDig		UDT	Currently called-up instance of opDig

Parameters	Data type	Description
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
polygon	UDT	Currently called-up instance of polygon
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
select	UDT	Currently called-up instance of select
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
setCvr	UDT	Currently called-up instance of setCvr
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
timeSw	UDT	Currently called-up instance of timeSw
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
vlv	UDT	Currently called-up instance of vlv
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program
alarms1	Word	Bit alarm list for Comfort/Advanced
alarms2	Word	Bit alarm list for Comfort/Advanced
vlvA	UDT	Currently called-up instance of vlvA
settingsPLC	UDT	Values of the tags that were connected as an input at the block in the PLC program
settingsHMI	UDT	Tags that can be written by the HMI system
statusHMI	UDT	Values with which the block currently operates and the values that the block outputs at the outputs in the PLC program

Parameters			Data type	Description
		alarms1	Word	Bit alarm list for Comfort/Advanced
		alarms2	Word	Bit alarm list for Comfort/Advanced

2.2 LBP_Aggr8 – Control of up to 8 Aggregates

Brief description

The block manages the automatic on/off and switching over of up to eight Aggregates.

The following modes are possible:

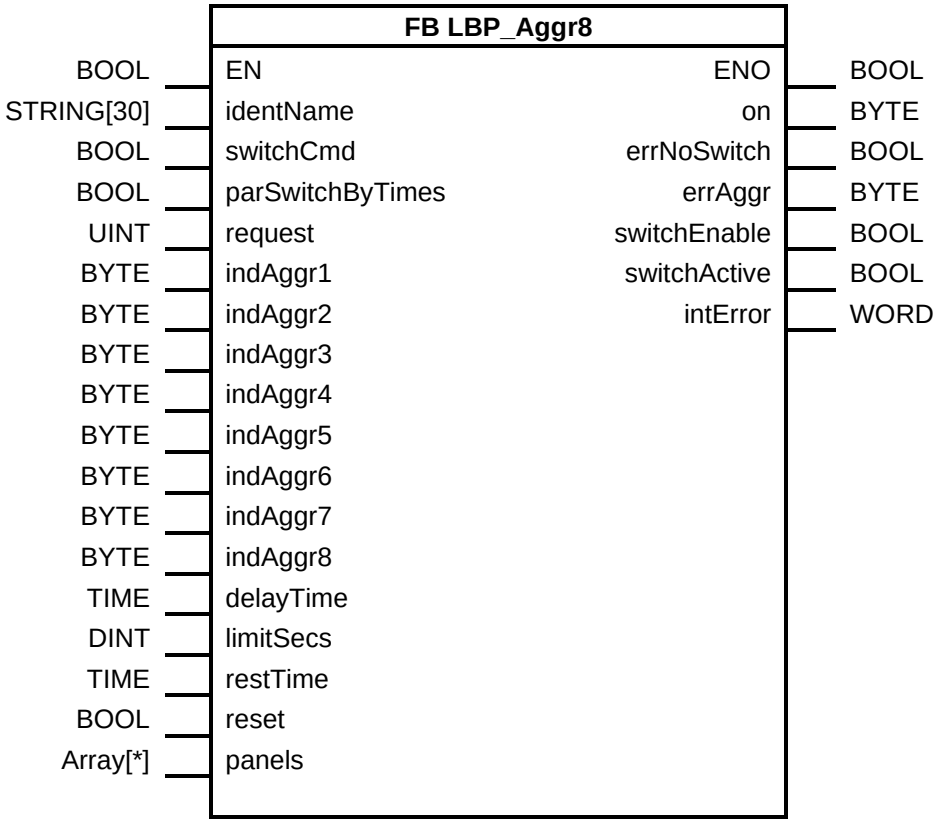
- Sequential switching according to priority
  - Switching on: The released Aggregate with the highest priority
  - Switching off: The Aggregate in operation with the lowest priority

  Priority-controlled actuation is active as soon as different priorities are parameterized.
- Changeover taking into account the operating hours
  - Switching on: The released Aggregate with the shortest total runtime
  - Switching off: The Aggregate in operation with the longest total running time

  This operating mode is active if the input “parSwitchByTime” is TRUE and only applies for the Aggregates that have the same priority.
- Changeover taking into account the cycle times
  - Switching on: The released Aggregate with the shortest last cycle time
  - Switching off: The Aggregate in operation with the longest current cycle time

Block

Figure 2-2: LBP_Aggr8



2.2.1 Interface Description of the PLC Block

Input parameters

Table 2-1

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName").
switchCmd	Bool	Switching is carried out to each rising edge
parSwitchByTimes	Bool	true: Switching is time-controlled (see parameter limitSecs)
request	UInt	Number of requested Aggregates (0 to 8)
indAggr1	Byte	Status Aggregate 1 Bit 0 – true: Aggregate is ready Bit 1 – true: Aggregate is switched on Bit 4 to 7 – Position in the switch-on sequence (priority)
indAggr2	Byte	Status Aggregate 2 (bit assignment see indAggr1)
indAggr3	Byte	Status Aggregate 3 (bit assignment see indAggr1)
indAggr4	Byte	Status Aggregate 4 (bit assignment see indAggr1)
indAggr5	Byte	Status Aggregate 5 (bit assignment see indAggr1)
indAggr6	Byte	Status Aggregate 6 (bit assignment see indAggr1)
indAggr7	Byte	Status Aggregate 7 (bit assignment see indAggr1)
indAggr8	Byte	Status Aggregate 8 (bit assignment see indAggr1)
delayTime	Time	Delay time (idle time until switching on again is possible)
limitSecs	DInt	Maximum runtime for time-controlled switching (if parSwitchByTimes = true)
restTime	Time	Rest period after switch-off (if parSwitchByTimes = true)
reset	Bool	Reset errors

Output parameters

Table 2-2

Parameters	Data type	Description
on	Byte	Bit 0–7 – Command to switch on (true) or off (false) for Aggregates 1–8
errNoSwitch	Bool	true: no Aggregate available
errAggr	Byte	Error at one or more Aggregates
switchEnable	Bool	true: Switching operation possible
switchActive	Bool	true: Switching operation active
intError	Word	Bit 0 – true: System error (reading the system time has failed)

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.2.2 Interface Description for HMI Communication

settingsHMI

Table 2-3

Parameters	Data type	Description
parSwitchByTimes	Bool	true: Switch time-controlled
request	UInt	Number of Aggregates required
prios	Array[0..7] of UInt	Position in the switch-on sequence for the Aggregates 1 to 8
delayTime	Time	Delay time (idle time until switching on again is possible)
limitSecs	Time	Maximum runtime for time-controlled switching
restTime	Time	Remaining run time for time-controlled switchover
note	String[80]	See “note
reset	Bool	Resetting the errors, switching commands, and remaining times
opStation	String[16]	See “opStation
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Switch command/switch by time/switch all (“over”) Bit 1 – Switch command (“cmd”) Bit 2 – Switch by time (“cmd”) Bit 3 – Request (“over”) Bit 4 – Delay time (“over”) Bit 5 – Rest time (“over”) … Bit 7 – Limit time (“over”) Bit 8 – Positions in the switch-on sequence (“over”)
opSecs	Array[0..7] of Time	Operating times for the Aggregates 1 to 8
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-4

Parameters	Data type	Description
parSwitchByTimes	Bool	true: Switch time-controlled.
request	UInt	Number of Aggregates required.
prios	Array[0..7] of UInt	Position in the switch-on sequence for the Aggregates 1 to 8.
delayTime	Time	Delay time (idle time until switching on again is possible).
limitSecs	Time	Maximum runtime for time-controlled switching.
restTime	Time	Rest period after switch-off.

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-5

Parameters	Data type	Description
identName	String[30]	Identification name (see 2.1.2 Cross-Block Input “identName”)
nextUp	Int	Number of the Aggregate to be switched on next
nextDown	Int	Number of the Aggregate to be switched on next
diffLimit	Time	Difference between runtime and maximum execution time
cycleSecs	Array[0..7] of Time	Runtimes of the Aggregates 1 to 8 (Array field 0 corresponds to Aggregate 1)
restTimes	Array[0..7] of Time	Remaining rest time of the Aggregates 1 to 8 (Array field 0 corresponds to Aggregate 1)
readys	Array[0..7] of Bool	true: Aggregate is ready to start (Array field 0 corresponds to Aggregate 1)
errors	Array[0..7] of Bool	true: Aggregate is interrupted (Array field 0 corresponds to Aggregate 1)
runCmds	Array[0..7] of Bool	Switching on commands at the Aggregates 1 to 8 (Array field 0 corresponds to Aggregate 1)
ons	Array[0..7] of Bool	true: Aggregate is switched on (Array field 0 corresponds to Aggregate 1)
request	UInt	Number of requested Aggregates
errNoSwitch	Bool	true: no Aggregate available
runUpDown	Bool	Aggregate(s) are started up or shut down
switchenable	Bool	true: Switching operation possible

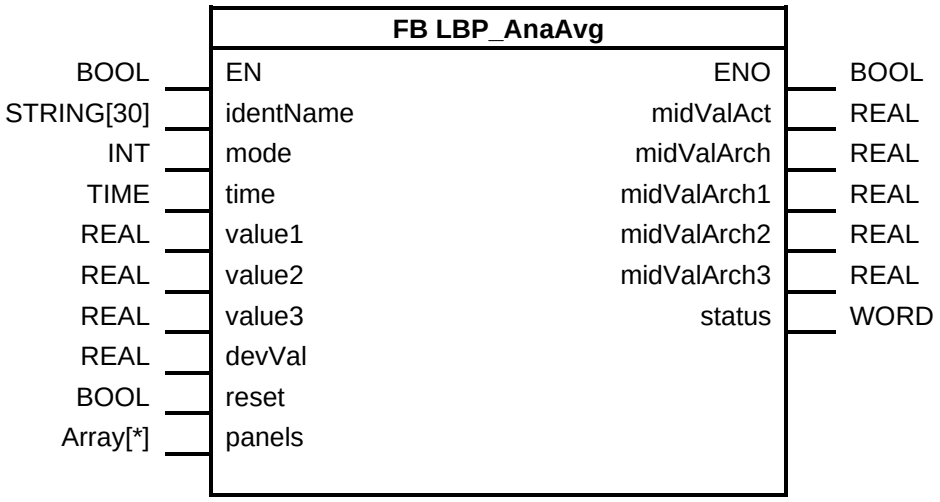
2.3 LBP_AnaAvg – Average Calculation with Plausibility Check

Brief description

The function calculates an average of up to three input values. An accepted deviation can be parameterized. Error messages are generated if a value exceeds the acceptable deviation.

Block

Figure 2-3: LBP_AnaAvg



2.3.1 Interface Description of the PLC Block

Input parameters

Table 2-6

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
mode	Int	Mode: "0" – Block is inactive "1" – "value1" is active "2" – "value1" and "value2" are active "3" – "value1", "value2", and "value3" are active
time	Time	Time span over which the historic mean value is formed
value1	Real	Value 1
value2	Real	Value 2
value3	Real	Value 3
devVal	Real	Permitted deviation in percentage
reset	Bool	true: All values are reset

Output parameters

Table 2-7

Parameters	Data type	Description
midValAct	Real	Current average value for all parameterized inputs
midValArch	Real	Historic average value for all parameterized inputs
midValArch1	Real	Historic mean value of "value1"
midValArch2	Real	Historic mean value of "value2"
midValArch3	Real	Historic mean value of "value3"
status	Word	State Bit assignment:

Parameters	Data type	Description
		Bit 0 – true: Mode 1, 2, or 3 is active and “value1” is not plausible Bit 1 – true: Mode 2 or 3 is active and “value2” is not plausible Bit 2 – 2 – true: Mode 3 is active and “value3” is not plausible

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.3.2 Interface Description for HMI Communication

settingsHMI

Table 2-8

Parameters	Data type	Description
mode	Int	Mode: “0” – Block is inactive “1” – “value1” is active “2” – “value1” and “value2” are active “3” – “value1”, “value2”, and “value3” are active
time	Time	Time span over which the historic mean value is formed
devVal	Real	Deviation in percentage
rangeMin	Real	Start of the measuring range (for scaling the displays)
rangeMax	Real	End of the measuring range (for scaling the displays)
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Mode (“over”) Bit 1 – Time (“over”) Bit 2 – Reset (“cmd”) Bit 3 – devVal (“over”)
note	String[80]	See “note
opStation	String[16]	See “opStation
format	String[10]	Display format on the HMI system (example: s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit	String[10]	Unit that is to be displayed on the HMI system
alarmsInfo	UDInt	Alarm information
AlarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-9

Parameters	Data type	Description
mode	Int	Mode: “0” – Block is inactive “1” – “value1” is active “2” – “value1” and “value2” are active “3” – “value1”, “value2”, and “value3” are active
time	Time	Time span over which the historic mean value is formed

Parameters	Data type	Description
devVal	Real	Deviation in percentage
value1	Real	Value 1
value2	Real	Value 2
value3	Real	Value 3

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-10

Parameters	Data type	Description
identName	String[30]	Identification name (see 2.1.2 Cross-Block Input “identName”)
status	Array[0..2] of Bool	Array field 0 – true: Mode 1, 2, or 3 is active and “value1” is not plausible Array field 1 – true: Mode 2 or 3 is active and “value2” is not plausible Array field 2 – true: Mode 3 is active and “value3” is not plausible
midValArch1	Real	Historic mean value of “value1”
midValArch2	Real	Historic mean value of “value2”
midValArch3	Real	Historic mean value of “value3”
midValAct	Real	Current average value for all parameterized inputs
midValArch	Real	Historic average value for all parameterized inputs

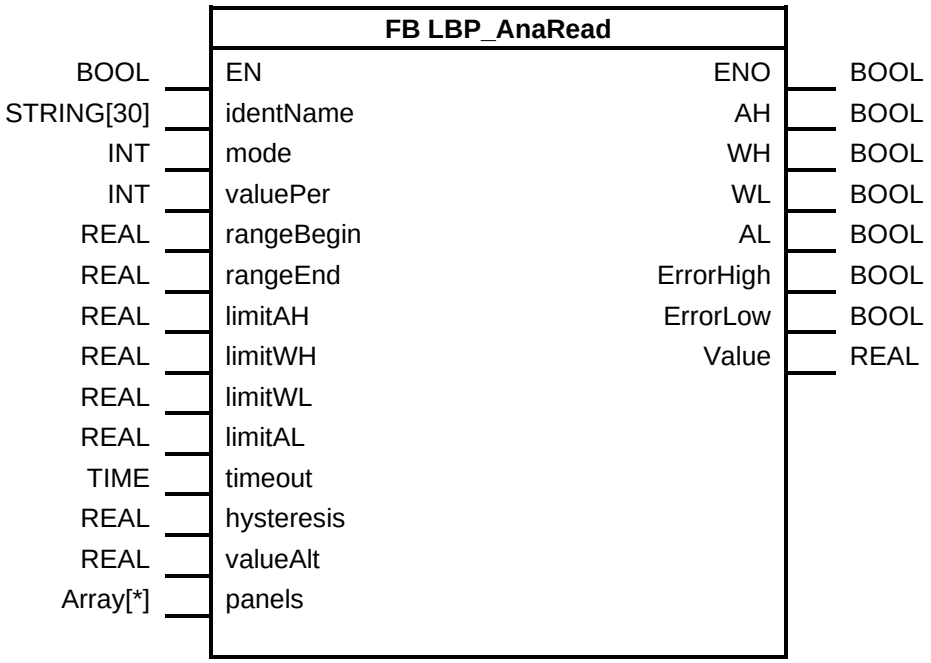
2.4 LBP_AnaRead – Analog Value Processing

Brief description

The block processes an analog value and displays it graphically. The value coming from the input module is displayed scaled. The limit ranges for warning and alarm can be parameterised.

Block

Figure 2-4: LBP_AnaRead



2.4.1 Interface Description of the PLC Block

Input parameters

Table 2-11

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
mode	Int	Mode: “0” – Alternative value “1” – Peripherals value unipolar “2” – Peripherals value bipolar “3” – Peripherals value with factor 0.1 “4” – Peripherals value with factor 0.01 “5” – Peripherals value with factor 1 “6” – Peripherals value with factor 10 “7” – Peripherals value with factor 100
valuePer	Int	Peripherals value
rangeBegin	Real	Start of the measuring range
rangeEnd	Real	End of the measuring range
limitAH	Real	Alarm high limit
limiWH	Real	Warning high limit
limitWL	Real	Warning low limit
limitAL	Real	Alarm low limit
timeout	Time	Tolerance time before an error message is generated when an error occurs
hysteresis	Real	Value for the absolute hysteresis: Limit alarms/warnings are only reset when the measured value falls below/exceeds the limit value plus/minus hysteresis.

Parameters	Data type	Description
valueAlt	Real	Alternative value

Output parameters

Table 2-12

Parameters	Data type	Description
AH	Bool	Alarm active for high limit violated
WH	Bool	Warning active for high limit violated
WL	Bool	Warning active for low limit violated
AL	Bool	Alarm active for low limit violated
ErrorHigh	Bool	Measurement error in the high range (measuring range of the peripherals exceeded)
ErrorLow	Bool	Measurement error in the low range (measuring range of the peripherals undershot or wire break)
Value	Real	Current analog value

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.4.2 Interface Description for HMI Communication

settingsHMI

Table 2-13

Parameters	Data type	Description
mode	Int	Mode: “0” – Alternative value “1” – Peripherals value unipolar “2” – Peripherals value bipolar “3” – Peripherals value with factor 0.1 “4” – Peripherals value with factor 0.01 “5” – Peripherals value with factor 1 “6” – Peripherals value with factor 10 “7” – Peripherals value with factor 100
rangeBegin	Real	Start of the measuring range
rangeEnd	Real	End of the measuring range
limitAH	Real	Alarm high limit
limiWH	Real	Warning high limit
limitWL	Real	Warning low limit
limitAL	Real	Alarm low limit
timeout	Time	Tolerance time before an error message is generated when an error occurs
Hysteresis	Real	Hysteresis
valueAlt	Real	If mode “0 – alternative value” is active, this is the alternative value that is used as the measured value without scaling.
Overwrite	Word	See “overwrite Bit assignment: Bit 0 – Mode (“over”)

Parameters	Data type	Description
		Bit 1 – rangeBegin (“over”) Bit 2 – rangeEnd (“over”) Bit 3 – limitAH (“over”) Bit 4 – limitWH (“over”) Bit 5 – limitWL (“over”) Bit 6 – limitAL (“over”) Bit 7 – timeout (“over”) Bit 8 – hysteresis (“over”) Bit 9 – valueAlt (“over”) Bit 15 – Creation of a test entry for the log (“cmd”)
note	String[80]	See “note
opStation	String[16]	See “opStation
format	String[10]	Display format on the HMI system (example: s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit	String[10]	Unit that is to be displayed on the HMI system
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-14

Parameters	Data type	Description
mode	Int	Mode: “0” – Alternative value “1” – Peripherals value unipolar “2” – Peripherals value bipolar “3” – Peripherals value with factor 0.1 “4” – Peripherals value with factor 0.01 “5” – Peripherals value with factor 1 “6” – Peripherals value with factor 10 “7” – Peripherals value with factor 100
rangeBegin	Real	Start of the measuring range
rangeEnd	Real	End of the measuring range
limitAH	Real	Alarm high limit
limiWH	Real	Warning high limit
limitWL	Real	Warning low limit
limitAL	Real	Alarm low limit
timeout	Time	Tolerance time before an error message is generated when an error occurs
hysteresis	Real	Hysteresis
valueAlt	Real	Alternative value

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-15

Parameters		Data type	Description
identName		String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
status		UDT	
	AH	Bool	Alarm active for high limit violated
	WH	Bool	Warning active for high limit violated
	WL	Bool	Warning active for low limit violated
	AL	Bool	Alarm active for low limit violated
	errorHigh	Bool	Measurement error in the high range
	errorLow	Bool	Measurement error in the low range
value		Real	Current analog value
rangeBegin		Real	Start of the measuring range
rangeEnd		Real	End of the measuring range
limitAH		Real	Alarm high limit
limiWH		Real	Warning high limit
limitWL		Real	Warning low limit
limitAL		Real	Alarm low limit

LBP_typeLog15MHIData

Table 2-16

Parameters		Data type	Description
data		Array[0..14] of "UDT"	
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

2.5 LBP_ClctVal – Group Display (without WinCC Display)

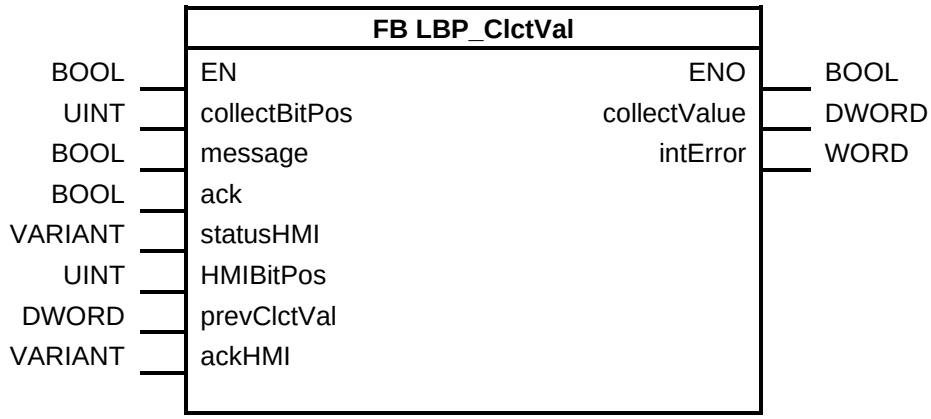
Brief description

The block generates a WinCC-compliant group display. It is possible to combine group displays via corresponding wiring at the “prevClctVal” input.

The acknowledgement status of the message from WinCC alarm logging can be processed via the “statusHMI” input. The wiring of the “ackHMI” parameter enables the forwarding of acknowledgements to WinCC.

Block

Figure 2-5: LBP_ClctVal



2.5.1 Interface Description of the PLC Block

Input parameters

Table 2-17

Parameters	Data type	Description
collectBitPos	UInt	Position of the message bit in the group display (0–15)
message	Bool	Current message status (true = active)
ack	Bool	Acknowledge command (rising edge)
statusHMI	Variant	When wired with a DWORD or WORD tag, this is interpreted as a status tag linked with WinCC alarm logging and the acknowledgement status is derived from this. Otherwise, the block's own acknowledgement status is used.
HMIBitPos	UInt	When using the “statusHMI” parameter, the corresponding bit position is specified here.
prevClctVal	DWord	For sequential grouping of several group displays, the Predecessor group display can be switched on here. The output “collectValue” is then the result of linking the current group display with “prevClctVal”.

Output parameters

Table 2-18

Parameters	Data type	Description
collectValue	DWord	Collective message generated
intError	Word	Fault detection of the block

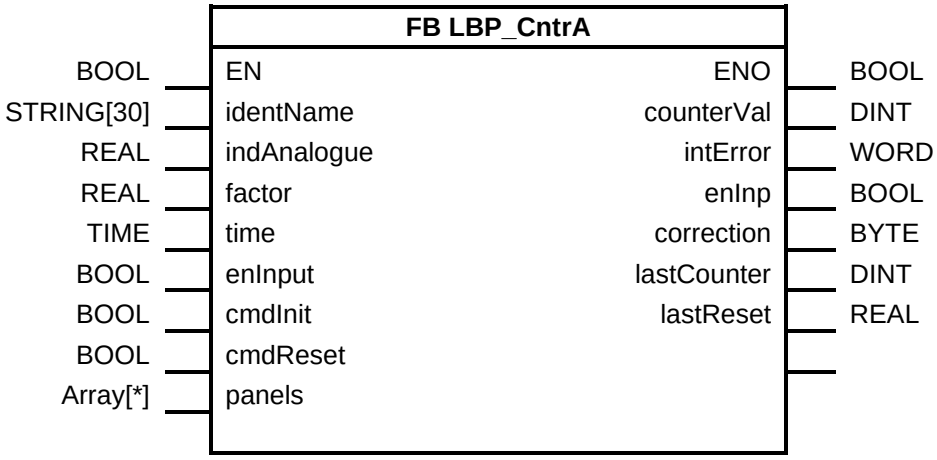
2.6 LBP_CntrA – Integrating Counter

Brief description

The block forms a counter value through the integration of an analog value. Both up and down counting (with negative analog value) is possible.

Block

Figure 2-6: LBP_CntrA



2.6.1 Interface Description of the PLC Block

Input parameters

Table 2-19

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indAnalogue	Real	Analog value from which the counter value is derived
factor	Real	Factor for pulse value determination
time	Time	Time interval for difference calculation
enInput	Bool	Enable calculation of meter reading
cmdInit	Bool	true: Initialize the counter (reset the internal values)
cmdReset	Bool	true: Reset counter value

Output parameters

Table 2-20

Parameters	Data type	Description
counterVal	DInt	Current counter value
intError	Word	Error code Bit assignment: Bit 0 – true: System error (error when reading the system time)
enInp	Bool	Enabling of counter reading calculation is active
correction	Byte	Last correction:

Parameters	Data type	Description
		“0”: Initialization “1”: Reset “2”: Change of counter “3”: Change of residual value
lastCounter	DInt	Counter value before correction
lastReset	Real	Residual value before correction

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.6.2 Interface Description for HMI Communication

The following user-defined data types are the same for all counters. However, not each tag is required for each counter.

settingsHMI

Table 2-21

Parameters		Data type	Description
enInp		Bool	Enable calculation of meter reading
note		String[80]	See “note
opStation		String[16]	See “opStation
format		String[10]	Display format on the HMI system (example: s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit		String[10]	Unit that is to be displayed on the HMI system.
overwrite		Word	See “overwrite Bit assignment: Bit 0 – Factor (“over”) (applies to CntrA, CntrD, CntrP) Bit 1 – Time (“over”) (applies to CntrA) Bit 2 – Limit (“over”) (applies to CntrD) Bit 3 – EnInput (“over”) (applies to CntrA, CntrD, CntrP) Bit 4 – EnInput (“cmd”) (applies to CntrA, CntrD, CntrP)
factor		Real	Factor for pulse value determination
time (CntrA)		Time	Time interval for difference calculation
limit (CntrD)		DInt	Overflow value
counter		DInt	Current counter value
rest		Real	Current residual value
cmds		UDT	-
	reset	Bool	true: Reset the counter value.
	takeOvEditCounterValue	Bool	Overwrite preceding count value with the current count value
	takeOvEditRestValue	Bool	Overwrite preceding residual value with the current residual value
	initialize	Bool	true: Initialize the counter (reset the internal values)
alarmsInfo		UDInt	Alarm information

Parameters	Data type	Description
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-22

Parameters	Data type	Description
factor	Real	Factor for pulse value determination
time (CntrA)	Time	Time interval for difference calculation
limit (CntrD)	DInt	Overflow value
enInput	Bool	true: Count upwards

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-23

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
status	Word	Status word Bit assignment: Bit 0: Initialization was active Bit 1: Reset Bit 2: “takeOvEditCounterValue” was active (the previous counter value was overwritten) Bit 3: “takeOvEditRestValue” was active (the previous residual value was overwritten) Bit 4: Newer differential value is calculated (CntrD) Bit 5: enInput is active Bit 8: enInput at the block input is active
statusHMI	Word	Status in the preceding cycle (bit assignment see “status”)
counterHMI	Dint	Current counter value
dataType	Int	Type of counter
rest	Real	Alarm high limit
correction	UDT	Warning high limit
timeStamp	DTL	Timestamp of last correction
flags	Byte	Last correction: “1”: Initialization “2”: Reset “3”: “takeOvEditCounterValue” was active (the previous counter value was overwritten) “4”: “takeOvEditRestValue” was active (the previous residual value was overwritten)
counter	Dint	Previous counter value
rest	Real	Previous remaining value

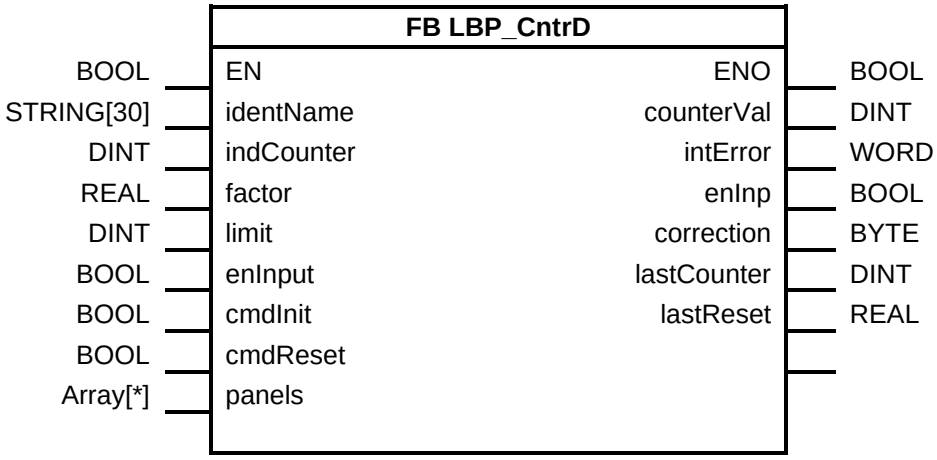
2.7 LBP_CntrD – Difference Counter

Brief description

The block forms a counter value by forming a difference from an input counter value. Both up and down counting (decreasing input value) is possible.

Block

Figure 2-7: LBP_CntrD



2.7.1 Interface Description of the PLC Block

Input parameters

Table 2-24

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
indCounter	DInt	Input counter value from which the output counter value is to be derived
factor	Real	Factor for pulse value determination
limit	DInt	If a value > 0 is parameterized here, a negative difference is evaluated as an overflow of “indCounter”. In this case, the calculation of the new output counter value takes “limit” into account as the maximum value of the input counter.
enInput	Bool	Enable calculation of meter reading
cmdInit	Bool	true: Initialize the counter (reset the internal values)
cmdReset	Bool	true: Reset the counter value

Output parameters

Table 2-25

Parameters	Data type	Description
counterVal	DInt	Current counter value
intError	Word	Error code Bit assignment: Bit 0 – true: System error (error when reading the system time)
enInp	Bool	Enabling of counter reading calculation is active

Parameters	Data type	Description
correction	Byte	Last correction: “0”: Initialization “1”: Reset “2”: Change of counter “3”: Change of residual value
lastCounter	DInt	Counter value before correction
lastReset	Real	Residual value before correction

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.7.2 Interface Description for HMI Communication

See 2.6.2 Interface Description for HMI Communication

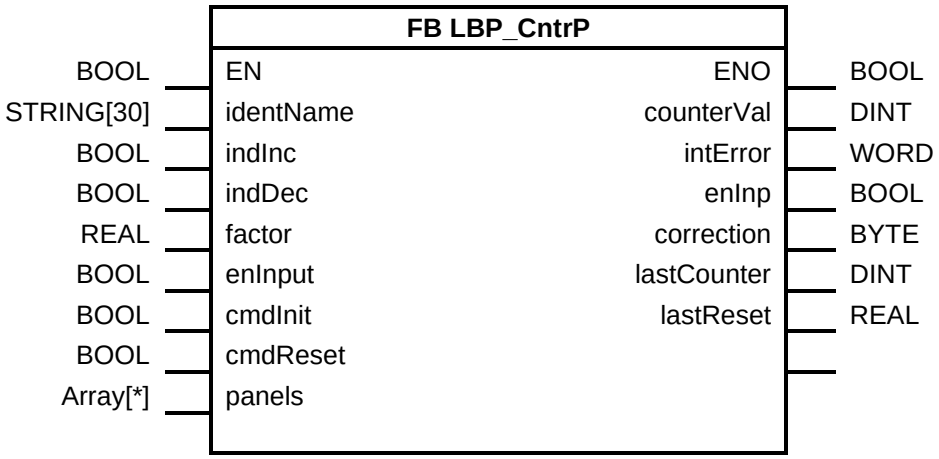
2.8 CntrP – Pulse Counter

Brief description

This function can be used to count up or down. This block can be used e.g. in a fast OB for counting pulses.

Block

Figure 2-8: LBP_CntrP



2.8.1 Interface Description of the PLC Block

Input parameters

Table 2-26

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
indInc	Bool	Counting impulse forwards
indDec	Bool	Counting impulse backwards.
factor	Real	Factor for pulse value determination (pulse valency)
enInput	Bool	Enable calculation of meter reading

Parameters	Data type	Description
cmdInit	Bool	true: Initialize the counter (reset the internal values)
cmdReset	Bool	true: Reset counter value

Output parameters

Table 2-27

Parameters	Data type	Description
counterVal	DInt	Current counter value
intError	Word	Error code Bit assignment: Bit 0 – true: System error (error when reading the system time)
enInp	Bool	Enabling of counter reading calculation is active
correction	Byte	Last correction: “0”: Initialization “1”: Reset “2”: Change of counter “3”: Change of residual value
lastCounter	DInt	Counter value before correction
lastReset	Real	Residual value before correction

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.8.2 Interface Description for HMI Communication

See 2.6.2 Interface Description for HMI Communication

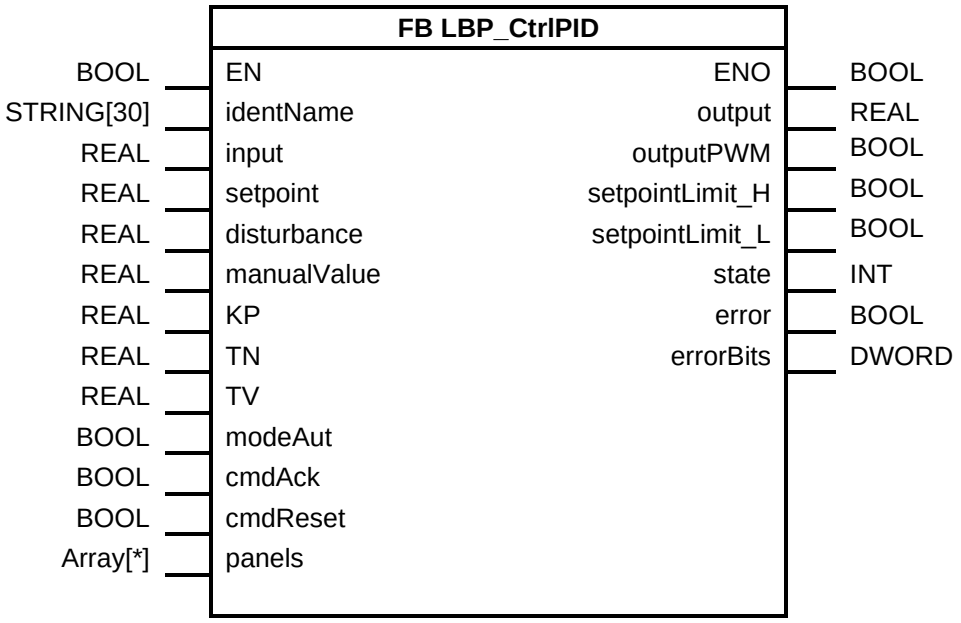
2.9 LBP_CtrlPID – Continuous PID Controller

Brief description

It is a configurable PID controller. The important parameters of the controller can be parameterized via faceplate. The controller can be used in manual and automatic operation.

Block

Figure 2-9: LBP_CtrlPID



2.9.1 Interface Description of the PLC Block

Input parameters

Table 2-28

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
input	Real	Current value
setpoint	Real	Set point for PID controller in automatic operation
disturbance	Real	Disturbance or pilot control value
manualValue	Real	This value is used as output value in manual operation
KP	Real	Proportional gain
TN	Real	Integration time
TV	Real	Differentiation time
modeAut	Bool	true: Automatic mode false: Manual operation
cmdAck	Bool	Acknowledgment of the errors at the PID controller (STEP 7 block PID_Compact)
cmdReset	Bool	Resetting the PID controller (STEP 7 block PID_Compact)

Output parameters

Table 2-29

Parameters	Data type	Description
output	Real	Output value in the REAL format
outputPWM	Bool	Pulse width modulated output value
setpointLimit_H	Bool	true: Absolute upper limit of the set point has been reached
setpointLimit_L	Bool	true: Absolute lower limit of the set point has been reached
state	Int	Operating mode: "0": Inactive "1": Initial optimization "2": Post-optimization "3": Automatic operation "4": Manual operation "5": Replacement output value with error monitoring
error	Bool	Error detected
errorBits	DWord	Error code of PID_Compact Bit assignment, see TIA Portal help under PID_Compact V2 → Parameter ErrorBits

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.9.2 Interface Description for HMI Communication

settingsHMI

Table 2-30

Parameters	Data type	Description
input	Real	Current value
setpoint	Real	Set point for PID controller in automatic operation
disturbance	Real	Disturbance or pilot control value
manualValue	Real	This value is used as output value in manual operation.
KP	Real	Proportional gain
TN	Real	Integration time
TV	Real	Differentiation time
rangeBegin	Real	Lower control limit
rangeEnd	Real	Upper control limit
output	Real	Output value
overwrite	Word	See "overwrite Bit assignment: Bit 0 – Allows the operation of mode, acknowledgment, and reset with the HMI system Bit 1 – input ("over") Bit 2 – setpoint ("over") Bit 3 – disturbance ("over") Bit 4 – manualValue ("over") Bit 15 – Creation of a test entry for the log ("cmd")
cmd	Word	Commands: Bit 0 – Deactivate

Parameters	Data type	Description
		Bit 1 – Initial optimization Bit 2 – Reoptimize Bit 3 – Setting the automatic operation Bit 4 – Setting the manual operation Bit 5 – Acknowledgment of the PID controller Bit 6 – Resetting of the PID controller Bit 7 – Apply the values of the block inputs for KP, TN, and TV Bit 8 – Apply the values from “settingsHMI” for KP, TN, and TV Bit 9 – Save the current values for KP, TN, and TV in “settingsHMI”
note	String[80]	See “note
opStation	String[16]	See “opStation
format	String[10]	Display format on the HMI system (example: s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit	String[10]	Unit that is to be displayed on the HMI system
CancelTuningLevel	Real	Parameters for self-optimization The higher the value, the longer and more exactly it is optimized.
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-31

Parameters	Data type	Description
input	Real	Current value
setpoint	Real	Set point for PID controller in automatic operation
disturbance	Real	Disturbance or pilot control value
manualValue	Real	This value is used as output value in manual operation.
KP	Real	KP value (proportional factor)
TN	Real	TN value (integration time constant)
TV	Real	TV value (differentiation constant)

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-32

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
status	Word	Operating mode: "0": Inactive "1": Initial optimization "2": Post-optimization "3": Automatic operation "4": Manual operation "5": Replacement output value with error monitoring
PIDCompactTeil	UDT	-
Progress	Real	Progress of the optimization as a percentage (0.0–100.0)
Gain	Real	Active proportional gain, Gain is retentive
Ti	Real	r_Ti > 0.0: Active integration time r_Ti = 0.0: I-component is switched off r_Ti is retentive
Td	Real	r_Td > 0.0: Active differentiation time r_Td = 0.0: D component is switched off r_Td is retentive
value	UDT	-
valInput	Real	Input value
setpoint	Real	Set point
feedback	Real	Feedback
msg	DWord	Error code of PID_Compact Bit assignment, see TIA Portal help under PID_Compact V2 → Parameter ErrorBits

LBP_typeLog15MHIData

Table 2-33

Parameters	Data type	Description
data	Array[0..14] of "UDT"	-
ts	DTL	Timestamp (local date and time)
code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

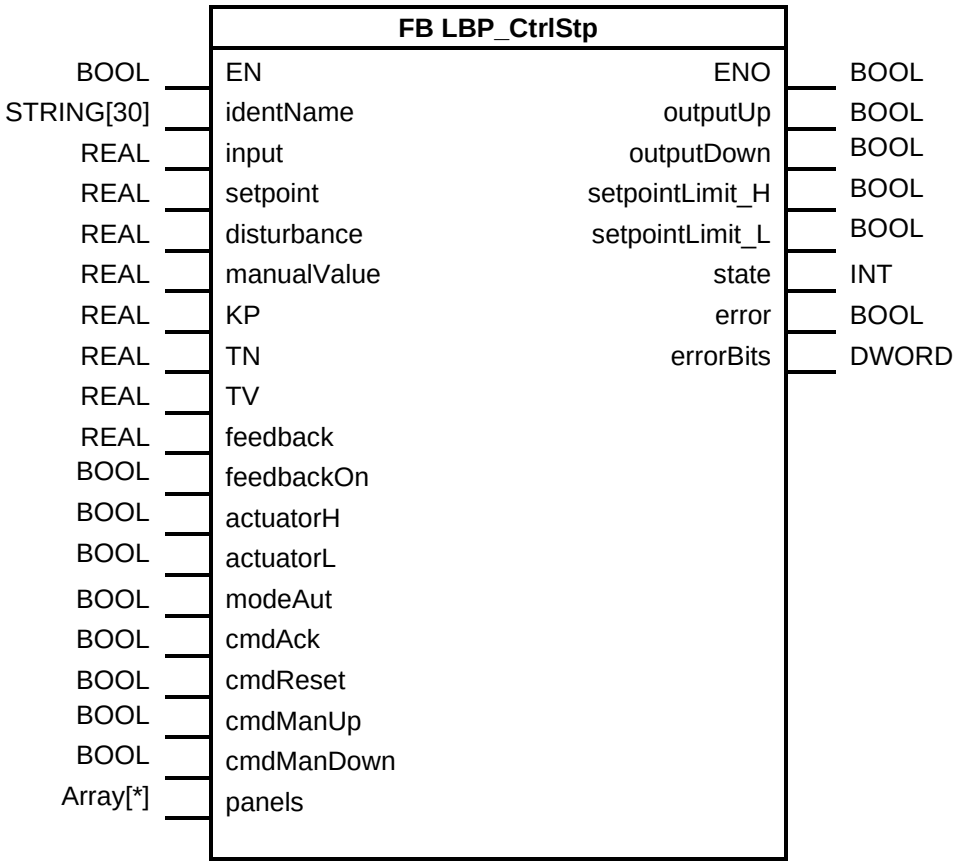
2.10 LBP_CtrlStp – Step Controller

Brief description

The block implements a PID controller with self-optimization for valves or actuators with integrating behavior. The controller can be operated with or without position feedback.

Block

Figure 2-10: LBP_CtrlStp



2.10.1 Interface Description of the PLC Block

Input parameters

Table 2-34

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
input	Real	Current value
setpoint	Real	Set point for PID controller in automatic operation
disturbance	Real	Disturbance or pilot control value
manualValue	Real	This value is used as output value in manual operation
KP	Real	Proportional gain
TN	Real	Integration time
TV	Real	Differentiation time
feedback	Real	Position feedback

Parameters	Data type	Description
feedbackOn	Bool	Position feedback should be used
actuatorH	Bool	Upper stop of the control position reached
actuatorL	Bool	Lower stop of the control position reached
modeAut	Bool	true: Automatic mode false: Manual operation
cmdAck	Bool	Acknowledgment of the errors at the PID controller (STEP 7 block PID_3Step)
cmdReset	Bool	Resetting the PID controller (STEP 7 block PID_3Step)
cmdManUp	Bool	Command for opening actuator (manual operation)
cmdManDown	Bool	Command for closing actuator (manual operation)

Output parameters

Table 2-35

Parameters	Data type	Description
outputUp	Bool	Actuation command for opening the actuator
outputDown	Bool	Actuation command for closing the actuator
setpointLimit_H	Bool	true: Absolute upper limit of the set point has been reached
setpointLimit_L	Bool	true: Absolute lower limit of the set point has been reached
state	Int	Operating mode: “0”: Inactive “1”: Initial optimization “2”: Post-optimization “3”: Automatic operation “4”: Manual operation “5”: Replacement output value with error monitoring
error	Bool	Error detected
errorBits	DWord	Error code of PID_Compact Bit assignment, see TIA Portal help under PID_3Step → Parameter ErrorBits

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.10.2 Interface Description for HMI Communication

settingsHMI

Table 2-36

Parameters	Data type	Description
input	Real	Current value
setpoint	Real	Set point for PID controller in automatic operation
disturbance	Real	Disturbance or pilot control value
manualValue	Real	This value is used as output value in manual operation
KP	Real	KP value (proportional factor)
TN	Real	TN value (integration time constant)

Parameters	Data type	Description
TV	Real	TV value (differentiation constant)
rangeBegin	Real	Lower control limit
rangeEnd	Real	Upper control limit
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Allows the operation of mode, acknowledgment, and reset with the HMI system Bit 1 – input (“over”) Bit 2 – setpoint (“over”) Bit 3 – disturbance (“over”) Bit 4 – manualValue (“over”) Bit 15 – Creation of a test entry for the log (“cmd”)
cmd	Word	Commands: Bit 0 – Deactivate Bit 1 – Initial optimization Bit 2 – Reoptimize Bit 3 – Setting the automatic operation Bit 4 – Setting the manual operation Bit 5 – Acknowledgment of the PID controller Bit 6 – Resetting of the PID controller Bit 7 – Apply the values of the block inputs for KP, TN, and TV Bit 8 – Apply the values from “settingsHMI” for KP, TN, and TV Bit 9 – Save the current values for KP, Tn, and TV in “settingsHMI”
note	String[80]	See “note
opStation	String[16]	See “opStation
format	String[10]	Display format on the HMI system (example: s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit	String[10]	Unit that is to be displayed on the HMI system
CancelTuningLevel	Real	Parameters for self-optimization The higher the value, the longer and more exactly it is optimized.
alarmsInfo1	UDInt	Alarm information 1
alarmsInfo2	UDInt	Alarm information 2
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-37

Parameters	Data type	Description
input	Real	Current value
setpoint	Real	Set point for PID controller in automatic operation
disturbance	Real	Disturbance or pilot control value
manualValue	Real	This value is used as output value in manual operation
KP	Real	Proportional gain
TN	Real	Integration time
TV	Real	Differentiation time

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-38

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
status	Word	Operating mode: “0”: Inactive “1”: Initial optimization “2”: Post-optimization “3”: Automatic operation “4”: Manual operation “5”: Replacement output value with error monitoring
PID3StepPart	UDT	-
Progress	Real	Progress of the current phase as a percentage (for optimization)
Actuator_H	Bool	Digital position feedback of the valve for the upper stop. If “Actuator_H” = TRUE, the position of the valve is at the upper stop and the valve is not moved further in this direction.
Actuator_L	Bool	Digital position feedback of the valve for the lower stop. If “Actuator_L” = TRUE, the position of the valve is at the lower stop and the valve is not moved further in this direction.
Output_UP	Bool	Digital output value for opening the valve. If “Config.OutputPerOn” (parameter of the block PID_3Step) = FALSE, the parameter “Output_UP” is used.
Output_DN	Bool	Digital output value for closing the valve. If “Config.OutputPerOn” (parameter of the block PID_3Step) = FALSE, the parameter “Output_DN” is used.
Gain	Real	Active proportional gain Gain is retentive
Ti	Real	Ti > 0.0: Active integration time

Parameters	Data type	Description
		Ti = 0.0: I-component is switched off Ti is retentive
Td	Real	Td > 0.0: Active differentiation time Td = 0.0: D component is switched off Td is retentive
value	UDT	-
valInput	Real	Input value
setpoint	Real	Set point
feedback	Real	Feedback
msg	DWord	Error code “of PID_3Step” Bit assignment, see TIA Portal help under PID_3Step → Parameter ErrorBits

LBP_typeLog15MHIData

Table 2-39

Parameters	Data type	Description
data	Array[0..14] of "UDT"	-
ts	DTL	Timestamp (local date and time)
code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

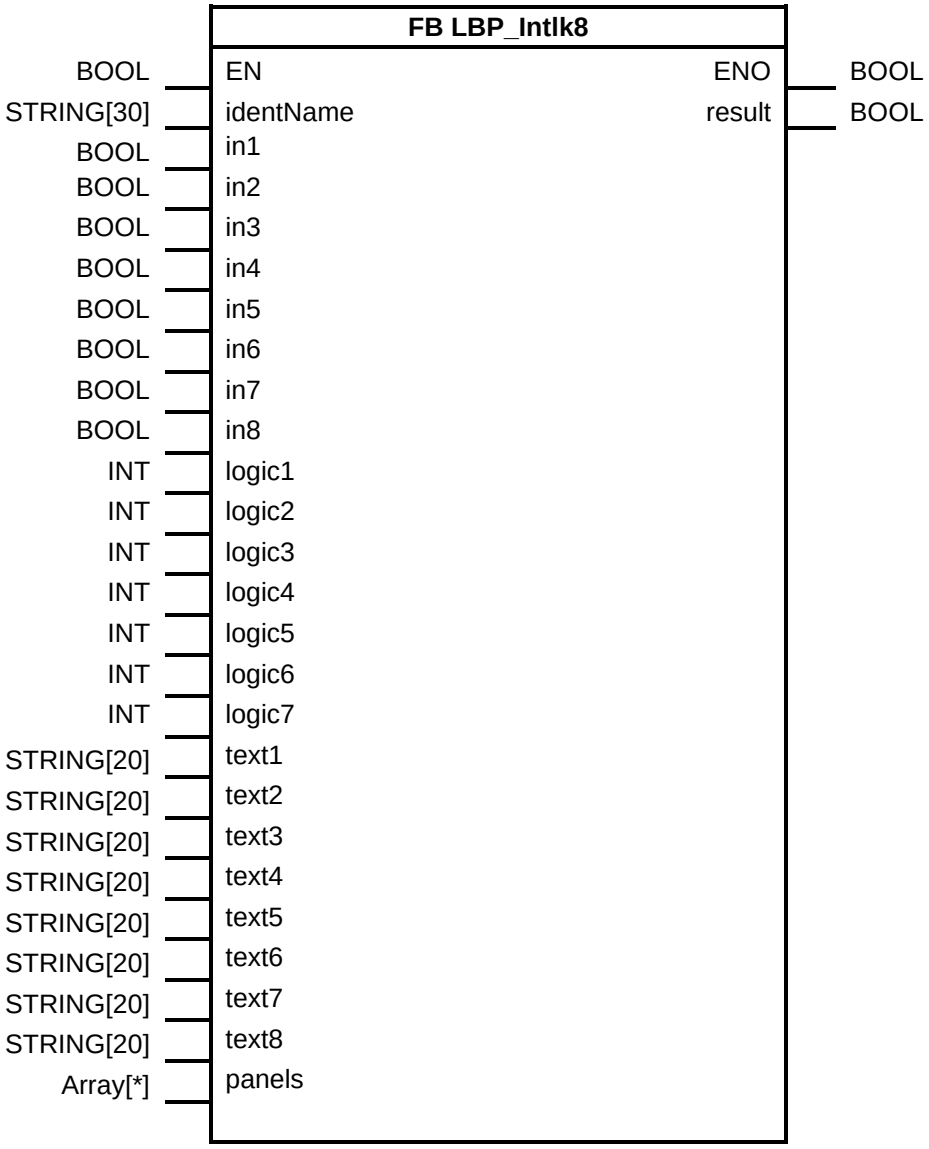
2.11 LBP_Intlk8 – Interlock for up to 8 Signals

Brief description

The block generates an output signal from up to eight binary inputs from a programmable logic.

Block

Figure 2-11: LBP_Intlk8



Note

Observe the following naming conventions for the Intlk8 to interlock a motor or a valve and open the Intlk8 window directly in WinCC from the motor window:

When using WinCC Professional or WinCC V7, name the instance in accordance with the scheme “Instance name of the motor or valve” and “LockOn”. Example: “instMtrLockOn”.
When using WinCC Comfort, name the “identName” in accordance with the scheme “identName of the motor or valve” and “LockOn”. Example: “instMtrLockOn”.

If you are using the interlock to assign a release to a motor or valve, use “RelOn” instead of “LockOn”.

2.11.1 Interface Description of the PLC Block

Input parameters

Table 2-40

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
in1 bis 8	Bool	Inputs 1–8
logic1 bis 7 logic1 – Logik zwischen in1 und in2 logic2 – Logik zwischen dem Ergenis von logic1 und in3 usw.	Int	“0”: bypassed “1”: OR “2”: AND “3”: OR NOT “4”: AND NOT)
text1 bis 8	Int	Default texts/title of the inputs 1-8

Output parameters

Table 2-41

Parameters	Data type	Description
result	Bool	Result of the logic

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.11.2 Interface Description for HMI Communication

settingsHMI

Table 2-42

Parameters	Data type	Description
input	Byte	Inputs Bit assignment: Bit 0 – in1 Bit 1 – in2, etc.
logics	Array[0..6] of Int	Logics: logic1 – Logic between in1 and in2 logic2 – Logic between the result of logic1 and in3,

Parameters	Data type	Description
		etc.
texts	Array[0..7] of String[20]	Texts/titles of inputs 1-8
note	String[80]	See “note
opStation	String[16]	See “opStation
overwrite	Word	See “overwrite Bit assignment: Bit 0 – in1 (“over”) Bit 1 – in2 (“over”) Bit 2 – in3 (“over”) Bit 3 – in4 (“over”) Bit 4 – in5 (“over”) Bit 5 – in6 (“over”) Bit 6 – in7 (“over”) Bit 7 – in8 (“over”) Bit 8 – logic1 (“over”) Bit 9 – logic2 (“over”) Bit 10 – logic3 (“over”) Bit 11 – logic4 (“over”) Bit 12 – logic5 (“over”) Bit 13 – logic6 (“over”) Bit 14 – logic7 (“over”)
colors	UDT	Colors
resultBulbOn	Int	Color of the result light in the activated status. “0”: light blue “1”: green “2”: yellow “3”: red
resultBulbOff	Int	Color of the result light in the deactivated status: “0”: light blue “1”: green “2”: yellow “3”: red
signalColorOn	DInt	Color of the logical connections if the preceding logic results in “true”: “9069843”: dark blue “0”: black “5026082”: light green “2366701”: light red “62207”: yellow “2588671”: orange
signalColorOff	DInt	Color of the logical connections if the preceding logic results in “false”: “14997947”: light blue “12829635”: light gray “27648”: dark green “133”: dark red “45493: dark yellow “1657769”: brown

settingsPLC

Table 2-43

Parameters	Data type	Description
input	Byte	Inputs Bit assignment: Bit 0 – in1 Bit 1 – in2, etc.
logics	Array[0..6] of Int	Logics: logic1 – Logic between in1 and in2m logic2 – Logic between the result of logic1 and in3, etc.
texts	Array[0..7] of String[20]	Default texts/titles of the inputs 1-8

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-44

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
status	Word	Status of the inputs and results Bit assignment: Bit 0 – in1 Bit 1 – in2 Bit 2 – in3 Bit 3 – in4 Bit 4 – in5 Bit 5 – in6 Bit 6 – in7 Bit 7 – in8 Bit 8 – Result logic1 Bit 9 – Result logic2 Bit 10 – Result logic3 Bit 11 – Result logic4 Bit 12 – Result logic5 Bit 13 – Result logic6 Bit 14 – Result logic7 Bit 15 – Output

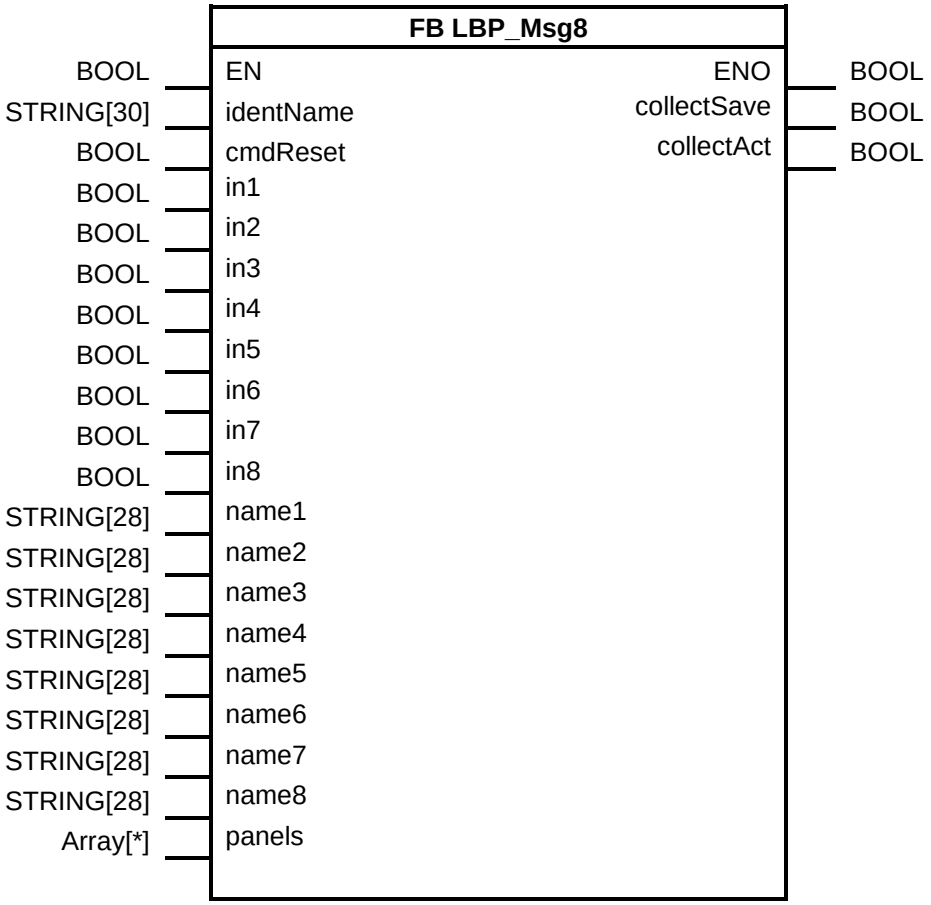
2.12 LBP_Msg8 – Message Block for 8 Messages

Brief description

The block creates messages based on an eight-digit binary code and stores a maximum of eight messages. Messages received can be reset via “reset”. If the messages are not to be saved, “reset” can be set to “True”.

Block

Figure 2-12: LBP_Msg8



Note

Observe the following naming conventions for the Msg8 if you to want visualize the faults of a motor or valve and want to open the Msg8 window in WinCC directly from the motor window:

When using WinCC Professional or WinCC V7, name the instance in accordance with the scheme “Instance name of the motor or valve” and “ErrExt”. Example: “instMtrErrExt”.
When using WinCC Comfort, name the “identName” in accordance with the scheme “identName of the motor or valve” and “ErrExt”. Example: “instMtrErrExt”.

If you are using the Msg8 to visualize warnings of a motor or valve, use “WarnExt” instead of “ErrExt”.

2.12.1 Interface Description of the PLC Block

Input parameters

Table 2-45

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
cmdReset	Bool	Reset the messages
in1 bis 8	Bool	Inputs for messages 1 to 8
name1 bis 8	String[28]	Name of the messages 1 to 8

Output parameters

Table 2-46

Parameters	Data type	Description
collectSave	Bool	Collective message of the stored messages
collectAct	Bool	Collective message of the current. Alarms

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.12.2 Interface Description for HMI Communication

settingsHMI

Table 2-47

Parameters	Data type	Description
command	Byte	Commands Bit assignment: Bit 0 – Resetting the stored messages
note	String[80]	See "note
opStation	String[16]	See "opStation
msgTest	Byte	Bits for testing the messages Bit assignment: Bit 0 – Message 1 (in1) Bit 1 – Message 2 (in2) Bit 2 – Message 3 (in3) Bit 3 – Message 4 (in4) Bit 4 – Message 5 (in5) Bit 5 – Message 6 (in6) Bit 6 – Message 7 (in7) Bit 7 – Message 8 (in8)
msgTypes	Byte	Types or colors of the lights for the messages. Bit assignment: Bit 0 – true: The light for message 1 is orange in the activated status (false: red) Bit 1 – true: The light for message 2 is orange in the activated status (false: red) etc.

settingsPLC

Table 2-48

Parameters	Data type	Description
names	Array[0..7] of String[28]	Name of the messages.

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-49

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
values	Word	Status of the messages Bit assignment: Bit 0 – Message1 is or was active … Bit 7 – Message8 is or was active Bit 8 – Message1 is active … Bit15 – Message 8 is active

2.13 LBP_Mtr – Simple Motor

Brief description

The “Mtr” block can be used to operate motors with one direction of rotation and one speed.

The engine can be used in the following modes:

- Manual operation: The engine is operated via the HMI
- Automatic operation: Motor is controlled via the “cmdAutOn” input
- Local operation: Motor is controlled via the “cmdLocOpen” input (local operation has the highest priority)
- Repair operation: Input “indRepair” is set. No control is possible and no display of errors and warnings

The block controls the reaction of the motor to a switch-over from ON or OFF. Error and status messages are automatically generated. Detailed information can be displayed through interconnection with blocks of the type “Msg8” and “Intlk8”. In addition, diagnostics entries of the block can be displayed in the HMI (log).

In all operating modes, the motor can only be switched on if the switch-on enable (“indRelease”) is set. If the motor is switched on, a missing switch-on enable does not result in a switch-off.

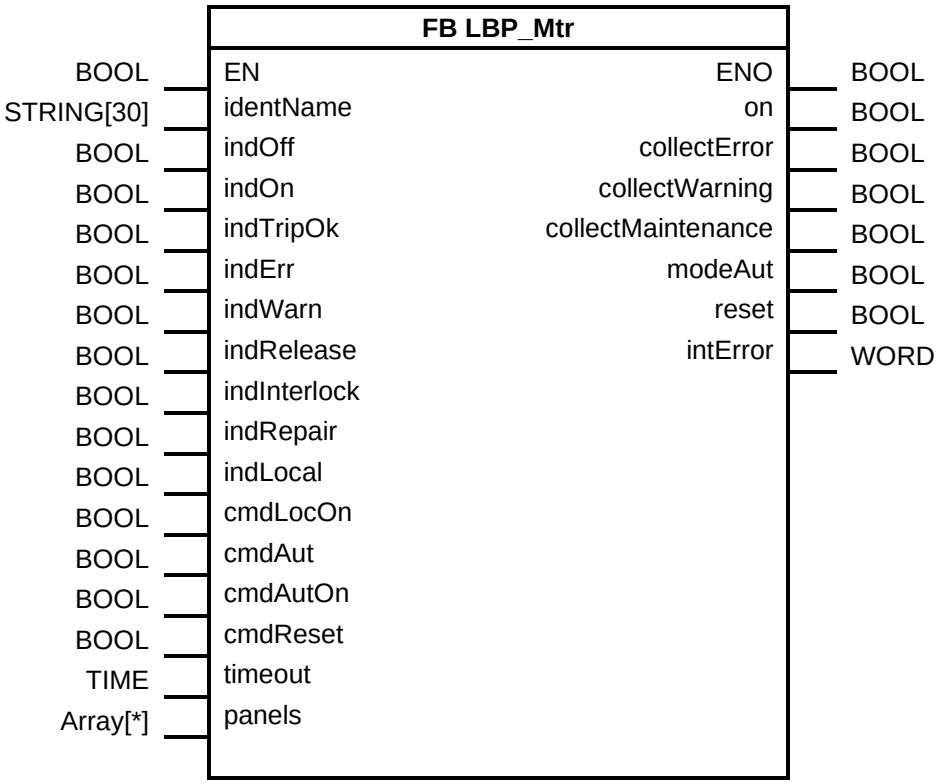
The following functions can be used for maintenance:

- Runtime meter
- Limit for operating hours (monitoring with limit greater than zero)

- Counter change controls
- Limit counter controls (monitoring with limit greater than zero)

Block

Figure 2-13: LBP_Mtr



2.13.1 Interface Description of the PLC Block

Input parameters

Table 2-50

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indOff	Bool	Feedback engine true: Engine is off
indOn	Bool	Feedback engine true: Engine is on
indTripOk	Bool	true: OK, false: Emergency stop
indErr	Bool	true: External error
indWarn	Bool	true: External Warning
indRelease	Bool	true: Switch on release for the motor.
indInterlock	Bool	true: Interlocking active (switches the motor off or prevents it from being switched on)
indRepair	Bool	true: Activate repair mode (motor can only be controlled externally; errors and warnings are not displayed via faceplate)
indLocal	Bool	true: Activate local operation (motor is in local operation and can only be activated via cmdLocOpen)

Parameters	Data type	Description
cmdLocOn	Bool	true: Motor is switched on when it is in local operation
cmdAut	Bool	Switching in external mode. The automatic mode is set with a positive edge.
cmdAutOn	Bool	true: Motor is switched on when it is in automatic operation.
cmdReset	Bool	true: Reset errors
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”

Output parameters

Table 2-51

Parameters	Data type	Description
on	Bool	Switch-on signal for motor.
collectError	Bool	Collective error.
collectWarning	Bool	Collective warning.
collectMaintenance	Bool	Collective maintenance message.
modeAut	Bool	true: Automatic operating mode active.
reset	Bool	true: “Reset” active.
intError	Word	Internal error in the block. Bit assignment: Bit 0 – System error (error when reading the system time).

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.13.2 Interface Description for HMI Communication

settingsHMI

Table 2-52

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
note	String[80]	See “note
opStation	String[16]	See “opStation
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 7 – Disabling the input cmdAut Bit 15 – Creation of a test entry for the log (“cmd”)
switches	DInt	Maximum number of switching operations
opHours	Time	Maximum operating time
cmd	Word	Commands Bit assignment: Bit 0 – Off Bit 1 – On Bit 2 – Manual operation Bit 3 – Automatic operation Bit 4 – Reset

Parameters	Data type	Description
maintenance	UDT	Relevant values for a maintenance
switches	DInt	Current number of switching operations
opHours	Time	Operating time in hours
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-53

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-54

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	UDT	Status messages
off	Bool	Turned off
on	Bool	Turned on
auto	Bool	Automatic operation activated
local	Bool	Local operation activated
release	Bool	Release
runDown	Bool	Shutdown
runUp	Bool	Startup
errorTrip	Bool	Emergency off is or was active
errorExt	Bool	Error is or was active
interlock	Bool	Locking
repair	Bool	Repair mode
collectMaintenance	Bool	Collective message for maintenance messages
collectWarning	Bool	Collective message for warnings
collectError	Bool	Collective message for errors
errorFlagTrip	Bool	Emergency off is active
errorFlagExt	Bool	Fault is active
errorTimeout	Bool	Switching time exceeded.
errorPlaus	Bool	Activation not plausible (switched on or switched off at the same time or neither switched on nor switched off)
opHoursLimitReached	Bool	Maximum operating time exceeded
switchCounterLimitReached	Bool	Maximum number of switching procedures exceeded

Parameters		Data type	Description
	warningExt	Bool	External warning active
	autoCommand	Bool	Control for automatic operation active

“LBP_typeLog15MHIData”

Table 2-55

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

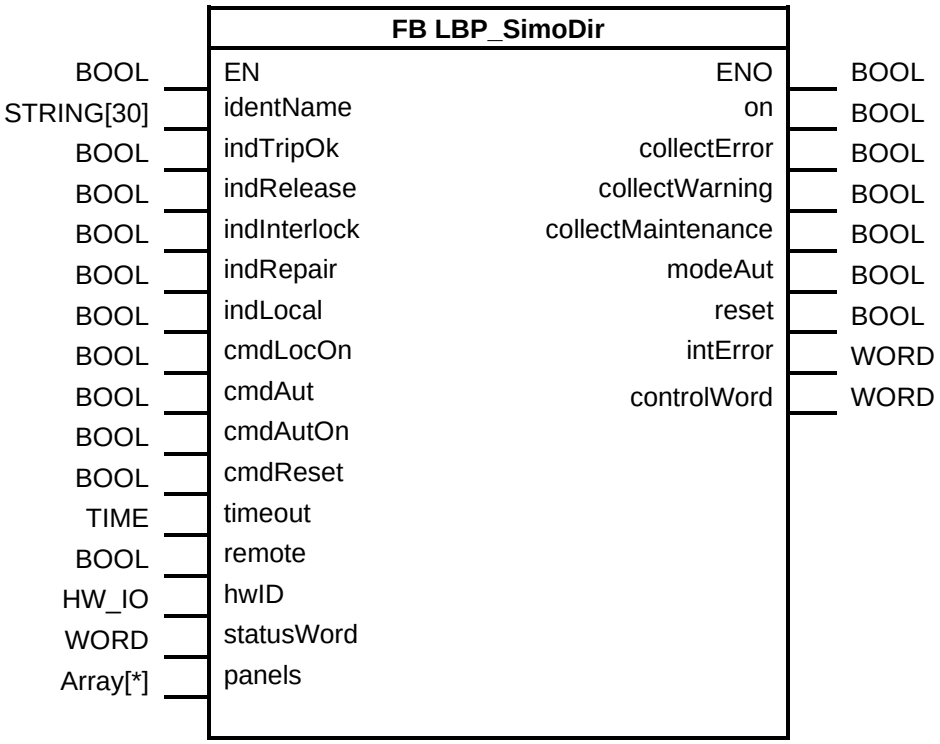
2.14 LBP_SimoDir – Simple Motor Controlled via a SIMOCODE

Brief description

The “SimoDir” block has the same functionality as the “Mtr” block, except that the latter communicates directly with the Simocode and thereby controls the motor.

Block

Figure 2-14: LBP_SimoDir



2.14.1 Interface Description of the PLC Block

Input parameters

Table 2-56

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
indTripOk	Bool	true: OK, false: Emergency stop
indRelease	Bool	true: Switch on release for the motor.
indInterlock	Bool	true: Interlocking active (switches the motor off or prevents it from being switched on)
indRepair	Bool	true: Activate repair mode (motor can only be controlled externally; errors and warnings are not displayed via faceplate)
indLocal	Bool	true: Activate local operation (motor is in local operation and can only be activated via cmdLocOpen)
cmdLocOn	Bool	true: Motor is switched on when it is in local operation
cmdAut	Bool	Switching in external mode. The automatic mode is set with a positive edge.
cmdAutOn	Bool	true: Motor is switched on when it is in automatic operation.
cmdReset	Bool	true: Reset errors
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
remote	Bool	Remote control of the Simocode active
hwID	HW_IO	Hardware ID of the Control device head of your Simocode module

Parameters	Data type	Description
		To connect them, proceed as follows: 1. Open “PLC > PLC tags > default tag table” 2. Click the “System constants” tab on the top right 3. Locate the hardware ID of the Control device head of your Simocode module. (Default name: “Control_device_1~Head”) 4. Move it to the input “hwID” using drag & drop
statusWord	Word	Output-Word of Simocode

Output parameters

Table 2-57

Parameters	Data type	Description
on	Bool	Switch-on signal for motor.
collectError	Bool	Collective error.
collectWarning	Bool	Collective warning.
collectMaintenance	Bool	Collective maintenance message.
modeAut	Bool	true: Automatic operating mode active.
reset	Bool	true: “Reset” active.
intError	Word	Internal error in the block. Bit assignment: Bit 0 – System error (error when reading the system time).
controlWord	Word	Input-Word of Simocode

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.14.2 Interface Description for HMI Communication

settingsHMI

Table 2-58

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
note	String[80]	See “note
opStation	String[16]	See “opStation
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 7 – Disabling the input cmdAut Bit 15 – Creation of a test entry for the log (“cmd”)
switches	DInt	Maximum number of switching operations
opHours	Time	Maximum operating time
cmd	Word	Commands Bit assignment: Bit 0 – Off Bit 1 – On Bit 2 – Manual operation Bit 3 – Automatic operation

Parameters	Data type	Description
		Bit 4 – Reset
maintenance	UDT	Relevant values for a maintenance
switches	DInt	Current number of switching operations
opHours	Time	Operating time in hours
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-59

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-60

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	UDT	Status messages
off	Bool	Turned off
on	Bool	Turned on
auto	Bool	Automatic operation activated
local	Bool	Local operation activated
release	Bool	Release
runDown	Bool	Shutdown
runUp	Bool	Startup
errorTrip	Bool	Emergency off is or was active
errorExt	Bool	Error is or was active
interlock	Bool	Locking
repair	Bool	Repair mode
collectMaintenance	Bool	Collective message for maintenance messages
collectWarning	Bool	Collective message for warnings
collectError	Bool	Collective message for errors
errorFlagTrip	Bool	Emergency off is active
errorFlagExt	Bool	Fault is active
errorTimeout	Bool	Switching time exceeded.
errorPlaus	Bool	Activation not plausible (switched on or switched off at the same time or neither switched on nor switched off)
opHoursLimitReached	Bool	Maximum operating time exceeded
switchCounterLimitReached	Bool	Maximum number of switching procedures

Parameters		Data type	Description
			exceeded
	warningExt	Bool	External warning active
	autoCommand	Bool	Control for automatic operation active

“LBP_typeLog15MHIData”

Table 2-61

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

Simocode datasets

You can find information on Simocode datasets in the “Function Manual for SIMOCODE pro – Communication”
Link: https://support.industry.siemens.com/cs/ww/en/view/109743960

The following Simocode specific UDTs are transferred to the HMI:

- DeviceDiag (“Function Manual for SIMOCODE pro – Communication” Chapter “3.4.2.7 Dataset 92 – Device Diagnostics”)
- Measurements (“Function Manual for SIMOCODE pro – Communication” Chapter “3.4.2.8 Dataser 94 – Measured Values”)
- Statistics (“Function Manual for SIMOCODE pro – Communication” Chapter “3.4.2.9 Dataset 95 – Service/Statistical Data”)

2.15 LBP_MtrDS – Motor with Two Speeds and Two Directions of Rotation

Brief description

This block controls a motor with two directions and two speeds. The engine can be used in the following operating modes:

- Manual operation: The engine is operated via the HMI
- Automatic operation: Motor is controlled via the inputs “cmdAutOn”, “cmdAutSpeed2” and “cmdAutReverse”
- Local operation: The motor is controlled via the inputs “cmdLocOn”, “cmdLocSpeed2”, and “cmdLocReverse” (local operation has the highest priority)
- Repair operation: Input “indRepair” is set. No control is possible and no display of errors and warnings

The block controls the response time when the motor is switched to ON or OFF. Error and status messages are automatically generated. Detailed information can be displayed through interconnection with blocks of the type “Msg8” and “Intlk8”. In addition, diagnostics entries of the block can be displayed in the HMI (log).

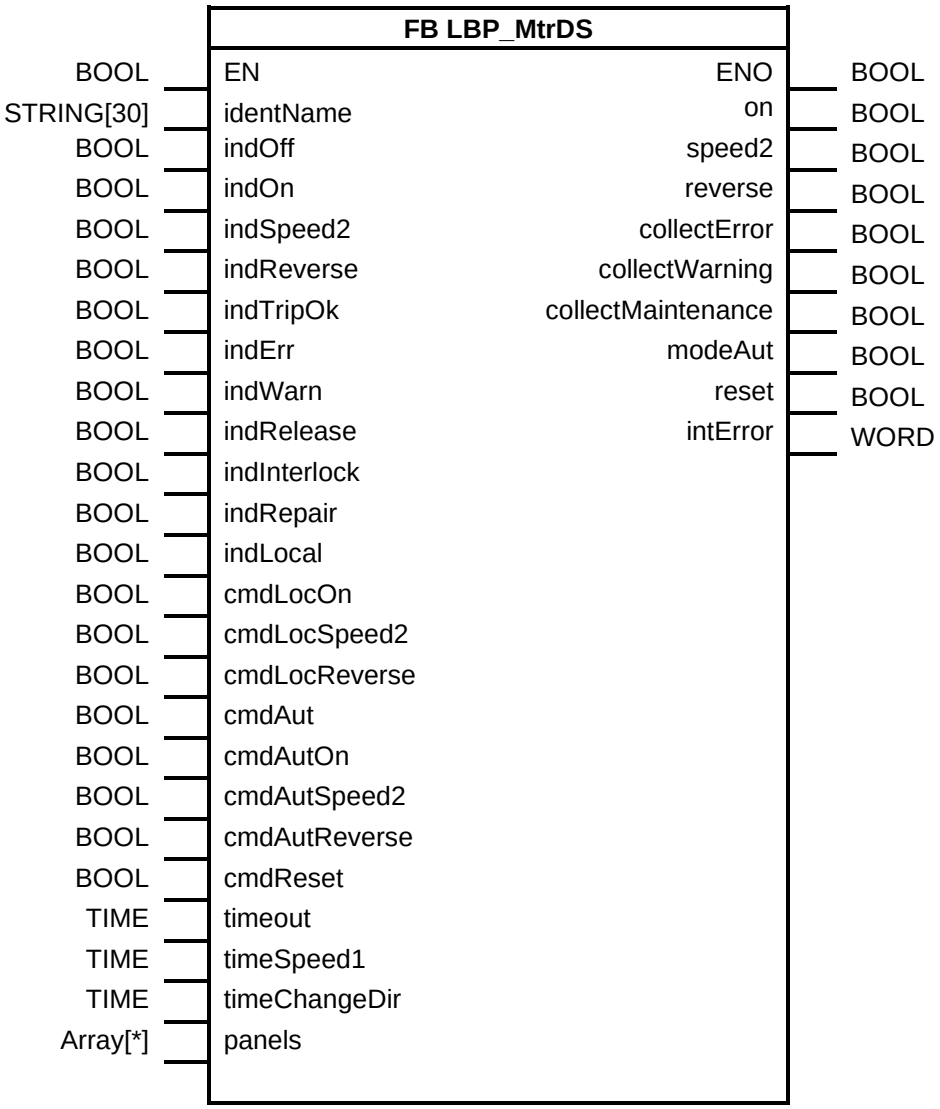
The following functions can be used for maintenance:

- Runtime meter
- Limit for operating hours (monitoring with limit greater than zero)

- Counter change controls
- Limit counter controls (monitoring with limit greater than zero)

Block

Figure 2-15: LBP_MtrDS



2.15.1 Interface Description of the PLC Block

Input parameters

Table 2-62

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indOff	Bool	Feedback engine: true: Engine is off
indOn	Bool	Feedback engine: true: Engine is on
indSpeed2	Bool	true: Second speed is active
indReverse	Bool	true: Motor runs in the opposite direction of rotation.
indTripOk	Bool	true: OK, false: Emergency stop
indErr	Bool	true: Active fault

Parameters	Data type	Description
indWarn	Bool	true: Warning active
indRelease	Bool	true: Release for activation
indInterlock	Bool	true: Interlocking active (switches the motor off or prevents it from being switched on)
indRepair	Bool	true: Repair mode active
indLocal	Bool	true: Local operation active
cmdLocOn	Bool	true: Motor is switched on when it is in local operation
cmdLocSpeed2	Bool	true: Motor is switched to speed 2 when it is in local operation
cmdLocReverse	Bool	true: Motor will change the direction of rotation if it is in local operation.
cmdAut	Bool	Switching in external mode. The automatic mode is set with a positive edge.
cmdAutOn	Bool	true: Motor is switched on when it is in automatic operation.
cmdAutSpeed2	Bool	true: Motor is switched to speed 2 when it is in automatic operation.
cmdAutReverse	Bool	true: Motor changes direction of rotation when it is in automatic operation.
cmdReset	Bool	true: Reset errors
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
timeSpeed1	Time	Time during which the motor must have run in speed 1 before the command for speed 2 becomes effective. A step-by-step run-up can be realized by means of a value >0.
timeChangeDir	Time	When the direction of rotation is switched over, the motor is initially switched off. The parameter indicates the time after which the motor starts up in the opposite direction of rotation.

Output parameters

Table 2-63

Parameters	Data type	Description
on	Bool	Switch-on signal for motor.
speed2	Bool	Signal for switching to speed 2
reverse	Bool	Signal for reversing the direction of rotation
collectError	Bool	Group error
collectWarning	Bool	Group warning
collectMaintenance	Bool	Maintenance required
modeAut	Bool	true: Automatic operating mode active.
reset	Bool	true: “Reset” active.
intError	Word	Error code

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.15.2 Interface Description for HMI Communication

settingsHMI

Table 2-64

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
switches	DInt	Maximum number of switching operations
opHours	Time	Maximum number of operating hours
maintenance	UDT	-
switches	DInt	Current number of switching operations
opHours	Time	Operating time in hours
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 1 – timeLevel (“over”) Bit 2 – timeDirection (“over”) Bit 15 – Creation of a test entry for the log (“cmd”)
note	String[80]	See “note
opStation	String[16]	See “opStation
cmd	Word	Commands Bit assignment: Bit 0 – Switching off motor in manual operation Bit 1 – Switching on motor in manual operation Bit 2 – Switching in manual operation Bit 3 – Switching to automatic operation Bit 4 – Resetting of the motor Bit 5 – Switching of velocity 1 to 2 or vice versa Bit 6 – Changing of the direction of rotation
timeLevel	Time	See “timeSpeed1” input
timeDirection	Time	See “timeChangeDir” input
config	Byte	Is only used at the HMI end
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-65

Parameters	Data type	Description
timeout	Time	Monitoring time (time after which there is a check as to whether the activation of the motor has functioned)
timeSpeed1	Time	See “timeSpeed1” input
timeChangeDir	Time	See “timeChangeDir” input

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-66

Parameters		Data type	Description
identName		String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus		UDT	-
	off	Bool	Motor off
	on	Bool	Motor on
	auto	Bool	Automatic operation active
	local	Bool	Local operation active
	release	Bool	Enable for activation given
	runDown	Bool	Startup active
	runUp	Bool	Motor is shut down
	autoCommand	Bool	Control for automatic operation active
	errorTrip	Bool	Emergency off is or was active
	errorExt	Bool	Error is or was active
	interlock	Bool	Locking
	repair	Bool	Repair mode
	collectMaintenance	Bool	Collective message for maintenance messages
	collectWarning	Bool	Collective message for warnings
	collectError	Bool	Collective message for errors
	errorFlagTrip	Bool	Emergency off is active
	errorFlagExt	Bool	Fault is active
	errorTimeout	Bool	Switching time exceeded
	errorPlaus	Bool	Activation not plausible (switched on or switched off at the same time or neither switched on nor switched off)
	opHoursLimitReached	Bool	Maximum operating time exceeded
	switchCounterLimitReached	Bool	Maximum switching procedures exceeded
	speedLevel2	Bool	Speed level 2 active.
	reverseDirection	Bool	Motor runs in reverse
	autoSpeedLevel2	Bool	The speed level 2 is actuated in automatic mode
	autoReverseDirection	Bool	A change of the direction of rotation is actuated in automatic mode
	warningExt	Bool	An external warning is active
	waitDirection	Bool	Waiting for direction of rotation change

LBP_typeLog15MHIData

Table 2-67

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

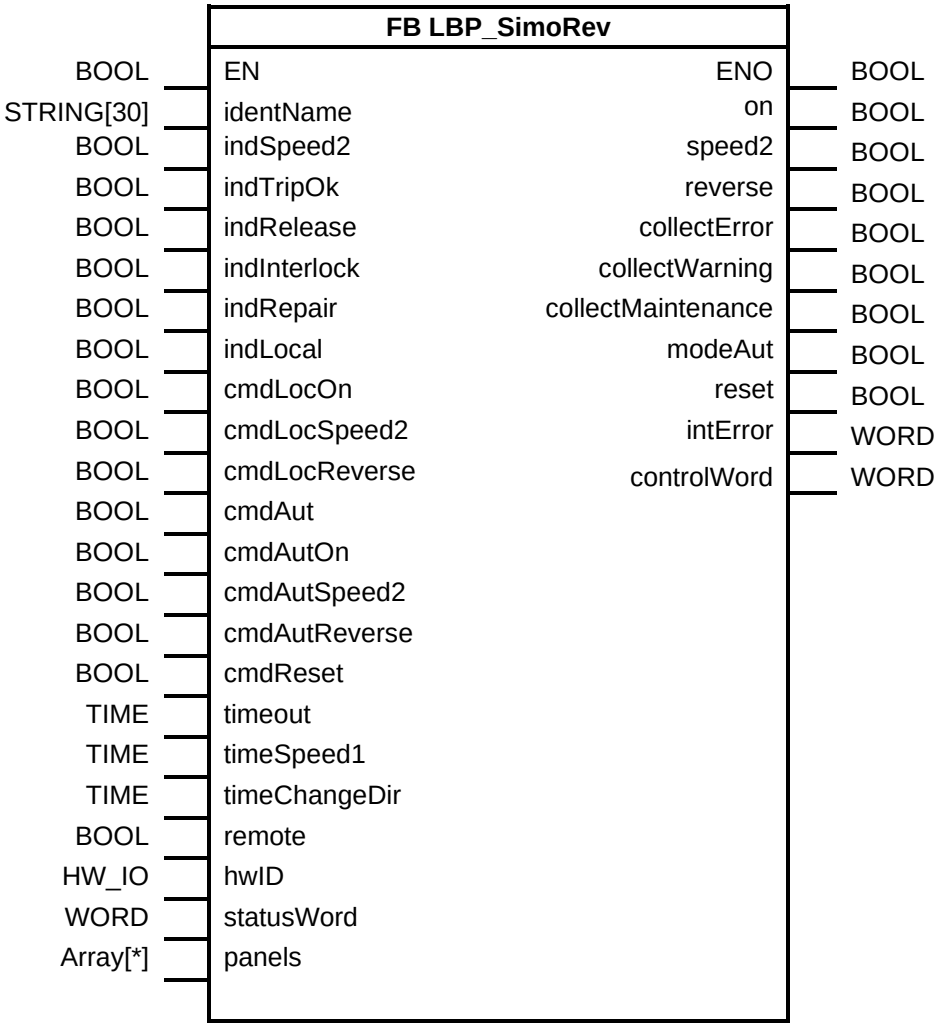
2.16 LBP_SimoRev – Motor Controlled by a SIMOCODE with Two Speeds and Two Directions of Rotation

Brief description

This block has the same functionality as the “MtrDS” block, except that it communicates directly with the Simocode and thereby controls the motor.

Block

Figure 2-16: LBP_SimoRev



2.16.1 Interface Description of the PLC Block

Input parameters

Table 2-68

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
indOff	Bool	Feedback engine: true: Engine is off
indOn	Bool	Feedback engine: true: Engine is on

Parameters	Data type	Description
indSpeed2	Bool	true: Second speed is active
indTripOk	Bool	true: OK, false: Emergency stop
indRelease	Bool	true: Release for activation
indInterlock	Bool	true: Interlocking active (switches the motor off or prevents it from being switched on)
indRepair	Bool	true: Repair mode active
indLocal	Bool	true: Local operation active
cmdLocOn	Bool	true: Motor is switched on when it is in local operation
cmdLocSpeed2	Bool	true: Motor is switched to speed 2 when it is in local operation
cmdLocReverse	Bool	true: Motor will change the direction of rotation if it is in local operation.
cmdAut	Bool	Switching in external mode. The automatic mode is set with a positive edge.
cmdAutOn	Bool	true: Motor is switched on when it is in automatic operation.
cmdAutSpeed2	Bool	true: Motor is switched to speed 2 when it is in automatic operation.
cmdAutReverse	Bool	true: Motor changes direction of rotation when it is in automatic operation.
cmdReset	Bool	true: Reset errors
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
timeSpeed1	Time	Time during which the motor must have run in speed 1 before the command for speed 2 becomes effective. A step-by-step run-up can be realized by means of a value >0.
timeChangeDir	Time	When the direction of rotation is switched over, the motor is initially switched off. The parameter indicates the time after which the motor starts up in the opposite direction of rotation.
remote	Bool	Remote control of the Simocode active
hwID	HW_IO	Hardware ID of the Control device head of your Simocode module To connect them, proceed as follows: 1. Open “PLC > PLC tags > default tag table” 2. Click the “System constants” tab on the top right 3. Locate the hardware ID of the Control device head of your Simocode module. (Default name: “Control_device_1~Head”) 4. Move it to the input “hwID” using drag & drop
statusWord	Word	Output-Word of Simocode

Output parameters

Table 2-69

Parameters	Data type	Description
on	Bool	Switch-on signal for motor.
speed2	Bool	Signal for switching to speed 2
reverse	Bool	Signal for reversing the direction of rotation
collectError	Bool	Group error

Parameters	Data type	Description
collectWarning	Bool	Group warning
collectMaintenance	Bool	Maintenance required
modeAut	Bool	true: Automatic operating mode active.
reset	Bool	true: “Reset” active.
intError	Word	Error code
controlWord	Word	Input-Word of Simocode

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.16.2 Interface Description for HMI Communication

settingsHMI

Table 2-70

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
switches	DInt	Maximum number of switching operations
opHours	Time	Maximum number of operating hours
maintenance	UDT	-
switches	DInt	Current number of switching operations
opHours	Time	Operating time in hours
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 1 – timeLevel (“over”) Bit 2 – timeDirection (“over”) Bit 15 – Creation of a test entry for the log (“cmd”)
note	String[80]	See “note
opStation	String[16]	See “opStation
cmd	Word	Commands Bit assignment: Bit 0 – Switching off motor in manual operation Bit 1 – Switching on motor in manual operation Bit 2 – Switching in manual operation Bit 3 – Switching to automatic operation Bit 4 – Resetting of the motor Bit 5 – Switching of velocity 1 to 2 or vice versa Bit 6 – Changing of the direction of rotation
timeLevel	Time	See “timeSpeed1” input
timeDirection	Time	See “timeChangeDir” input
config	Byte	Is only used at the HMI end
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-71

Parameters	Data type	Description
timeout	Time	Monitoring time (time after which there is a check as to whether the activation of the motor has functioned)
timeSpeed1	Time	See “timeSpeed1” input
timeChangeDir	Time	See “timeChangeDir” input

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-72

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	UDT	-
off	Bool	Motor off
on	Bool	Motor on
auto	Bool	Automatic operation active
local	Bool	Local operation active
release	Bool	Enable for activation given
runDown	Bool	Startup active
runUp	Bool	Motor is shut down
autoCommand	Bool	Control for automatic operation active
errorTrip	Bool	Emergency off is or was active
errorExt	Bool	Error is or was active
interlock	Bool	Locking
repair	Bool	Repair mode
collectMaintenance	Bool	Collective message for maintenance messages
collectWarning	Bool	Collective message for warnings
collectError	Bool	Collective message for errors
errorFlagTrip	Bool	Emergency off is active
errorFlagExt	Bool	Fault is active
errorTimeout	Bool	Switching time exceeded
errorPlaus	Bool	Activation not plausible (switched on or switched off at the same time or neither switched on nor switched off)
opHoursLimitReached	Bool	Maximum operating time exceeded
switchCounterLimitReached	Bool	Maximum switching procedures exceeded
speedLevel2	Bool	Speed level 2 active.
reverseDirection	Bool	Motor runs in reverse
autoSpeedLevel2	Bool	The speed level 2 is actuated in automatic mode
autoReverseDirection	Bool	A change of the direction of rotation is actuated in automatic mode

Parameters		Data type	Description
	warningExt	Bool	An external warning is active
	waitDirection	Bool	Waiting for direction of rotation change

LBP_typeLog15MHIData

Table 2-73

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

Simocode datasets

You can find information on Simocode datasets in the “Function Manual for SIMOCODE pro – Communication”
Link: https://support.industry.siemens.com/cs/ww/en/view/109743960

The following Simocode specific UDTs are transferred to the HMI:

- DeviceDiag (“Function Manual for SIMOCODE pro – Communication” Chapter “3.4.2.7 Dataset 92 – Device Diagnostics”)
- Measurements (“Function Manual for SIMOCODE pro – Communication” Chapter “3.4.2.8 Dataser 94 – Measured Values”)
- Statistics (“Function Manual for SIMOCODE pro – Communication” Chapter “3.4.2.9 Dataset 95 – Service/Statistical Data”)

2.17 LBP_MtrF – Speed-Controlled Motor

Brief description

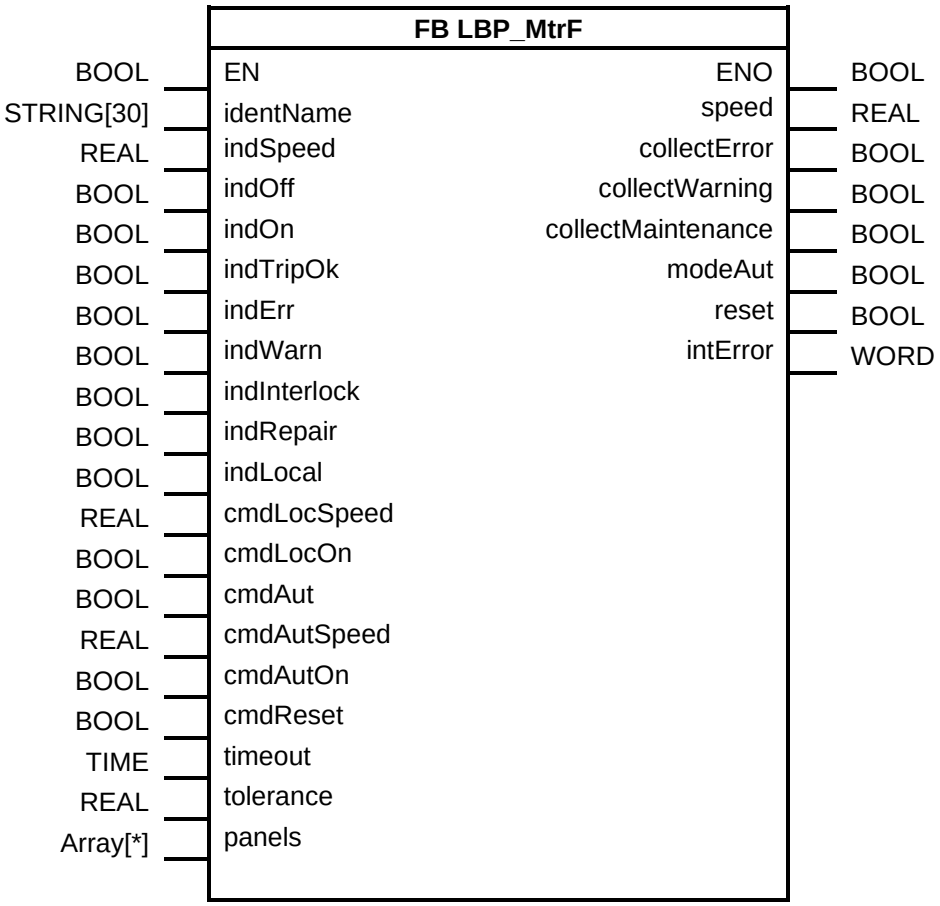
This block is used to control an engine with tag speed (frequency inverter) and analog speed feedback. The motor can be switched on and off. In addition, a speed value is also transferred. The faceplate can also be used to define a value after which the motor is considered to have stopped.

The engine can be used in the following modes:

- Manual operation: The engine is operated via the HMI
- Automatic operation: Motor is controlled via the “cmdAutSpeed” input
- Local operation: Motor is controlled via the “cmdLocSpeed” input (local operation has the highest priority)
- Repair operation: Input “indRepair” is set; no control is possible and no display of errors and warnings

Block

Figure 2-17: LBP_MtrF



2.17.1 Interface Description of the PLC Block

Input parameters

Table 2-74

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indSpeed	Real	Current motor speed
indOff	Bool	Feedback engine: True: Engine is off
indOn	Bool	Feedback engine: True: Engine is on
indTripOk	Bool	true: OK, false: Emergency stop
indErr	Bool	true: External error
indWarn	Bool	true: External Warning
indInterlock	Bool	true: Interlocking active (switches the motor off or prevents it from being switched on)
indRepair	Bool	true: Activate repair mode (motor can only be controlled externally; errors and warnings are not displayed via faceplate)

Parameters	Data type	Description
indLocal	Bool	true: Local operation active
cmdLocSpeed	Real	Velocity specification for local actuation
cmdLocOn	Bool	true: Motor is switched on when it is in local operation
cmdAut	Bool	Switching in external mode. The automatic mode is set with a positive edge.
cmdAutSpeed	Real	Velocity specification in automatic operation
cmdAutOn	Bool	true: Motor is switched on when it is in automatic operation.
cmdReset	Bool	true: Reset errors
timeout	Time	Monitoring time. In the case of a difference between set speed and actual speed that lies outside the tolerance, “errorTimeout” is set after this time.
tolerance	Real	Permissible tolerance of the deviation between setpoint and actual speed

Output parameters

Table 2-75

Parameters	Data type	Description
speed	Real	Velocity specification for motor
collectError	Bool	Group error
collectWarning	Bool	Group warning
collectMaintenance	Bool	Service necessary
modeAut	Bool	true: Automatic operating mode active.
reset	Bool	true: “Reset” active.
intError	Word	Error code

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.17.2 Interface Description for HMI Communication

settingsHMI

Table 2-76

Parameters	Data type	Description
timeout	Time	Monitoring time. In the case of a difference between set speed and actual speed that lies outside the tolerance, “errorTimeout” is set after this time.
cmd	Word	Command byte: Bit 2 – Operating mode “Manual” Bit 3 – “Automatic” operating mode Bit 4 – Reset
speedTolerance	Real	Permissible tolerance of the deviation between setpoint and actual speed
limitOff	Real	Speed value below which the motor is to be displayed in the “Off” status on the HMI. (See overwrite Bit 2)
limitOn	Real	Speed value above which the motor is to be displayed in the “On” status on the HMI. (See overwrite Bit 3)

Parameters	Data type	Description
opHours	Time	Maximum operating hours
note	String[80]	See “note
opStation	String[16]	See “opStation
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 1 – Speed tolerance (“over”) Bit 2 – Activate limit Off (“cmd”) Bit 3 – Activate limit On (“cmd”) Bit 7 – Disabling the input “cmdAut” Bit 15 – Creation of a test entry for the log (“cmd”)
maintenance	-	-
opHours	Time	Current operating hours
speed	Real	Velocity specification in manual operation
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-77

Parameters	Data type	Description
timeout	Time	Monitoring time. In the case of a difference between set speed and actual speed that lies outside the tolerance, “errorTimeout” is set after this time.
tolerance	Real	Permissible tolerance of the deviation between setpoint and actual speed
off	Bool	Motor is to be switched off
on	Bool	Motor is to be switched on

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-78

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	-	-
off	Bool	Motor is switched off
on	Bool	Motor is switched on
auto	Bool	Automatic operation is active
local	Bool	Local operation is active
runDown	Bool	Motor is shut down
runUp	Bool	Motor is booted
errorTrip	Bool	Emergency off is or was active
errorExt	Bool	Error is or was active

Parameters		Data type	Description
	interlock	Bool	Locking
	repair	Bool	Repair mode
	opHoursLimitReached	Bool	Collective message for maintenance message
	warningExt	Bool	Collective message for warnings
	collectError	Bool	Collective message for errors
	errorFlagTrip	Bool	Emergency off is active
	errorFlagExt	Bool	Fault is active
	errorTimeout	Bool	Monitoring time for impermissible difference between reference speed and actual speed has expired
	cmdOn	Bool	Switch-on command
cmdAutSpeed		Real	Velocity specification in automatic operation
speed		Real	Current velocity

LBP_typeLog15MHIData

Table 2-79

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

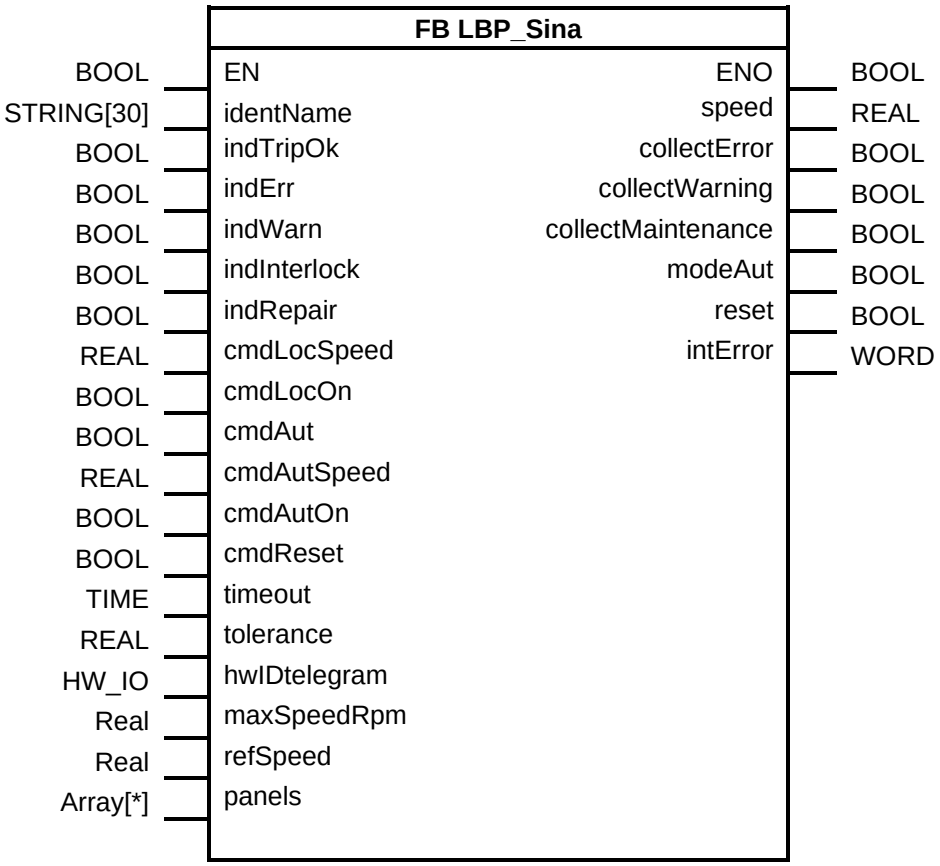
2.18 LBP_Sina – Via a G120 Speed-Controlled Motor

Brief description

This block has the same functionality as the “MtrF” block, except that it communicates directly with the G120 and thereby controls the motor.

Block

Figure 2-18: LBP_Sina



2.18.1 Interface Description of the PLC Block

Input parameters

Table 2-80

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indTripOk	Bool	true: OK, false: Emergency stop
indErr	Bool	true: External error
indWarn	Bool	true: External Warning
indInterlock	Bool	true: Interlocking active (switches the motor off or prevents it from being switched on)
indRepair	Bool	true: Activate repair mode (motor can only be controlled externally; errors and warnings are not displayed via faceplate)
cmdLocSpeed	Real	Velocity specification for local actuation
cmdLocOn	Bool	true: Motor is switched on when it is in local operation
cmdAut	Bool	Switching in external mode. The automatic mode is set with a positive edge.
cmdAutSpeed	Real	Velocity specification in automatic operation
cmdAutOn	Bool	true: Motor is switched on when it is in automatic

Parameters	Data type	Description
		operation.
cmdReset	Bool	true: Reset errors
timeout	Time	Monitoring time. In the case of a difference between set speed and actual speed that lies outside the tolerance, “errorTimeout” is set after this time.
tolerance	Real	Permissible tolerance of the deviation between setpoint and actual speed
hwIDtelegram	HW_IO	Hardware ID of the G120 module Telegram To connect them, proceed as follows: 1. Open “PLC > PLC tags > default tag table” 2. Click the “System constants” tab in the top right 3. Locate the hardware ID of SINAMICS Telegram 1. (Possible name: “SINAMICS_CU240E~PROFINET-interface~Standard_Telegramm_1”) Move it to the input “hwIDtelegram” using drag & drop
maxSpeedRpm	Real	Maximum motor speed in revolutions per minute
refSpeed	Real	Nominal speed of the drive àp2000

Output parameters

Table 2-81

Parameters	Data type	Description
speed	Real	Velocity specification for motor
collectError	Bool	Group error
collectWarning	Bool	Group warning
collectMaintenance	Bool	Service necessary
modeAut	Bool	true: Automatic operating mode active.
reset	Bool	true: “Reset” active.
intError	Word	Error code

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.18.2 Interface Description for HMI Communication

settingsHMI

Table 2-82

Parameters	Data type	Description
timeout	Time	Monitoring time. In the case of a difference between set speed and actual speed that lies outside the tolerance, “errorTimeout” is set after this time.
cmd	Word	Command byte: Bit 2 – Operating mode “Manual” Bit 3 – “Automatic” operating mode Bit 4 – Reset
speedTolerance	Real	Permissible tolerance of the deviation between setpoint and actual speed
limitOff	Real	Speed value below which the motor is to be displayed in the “Off” status on the HMI. (See overwrite Bit 2)
limitOn	Real	Speed value above which the motor is to be

Parameters	Data type	Description
		displayed in the “On” status on the HMI. (See overwrite Bit 3)
opHours	Time	Maximum operating hours
note	String[80]	See “note
opStation	String[16]	See “opStation
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 1 – Speed tolerance (“over”) Bit 2 – Activate limit Off (“cmd”) Bit 3 – Activate limit On (“cmd”) Bit 7 – Disabling the input “cmdAut” Bit 15 – Creation of a test entry for the log (“cmd”)
maintenance	-	-
opHours	Time	Current operating hours
speed	Real	Velocity specification in manual operation
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-83

Parameters	Data type	Description
timeout	Time	Monitoring time. In the case of a difference between set speed and actual speed that lies outside the tolerance, “errorTimeout” is set after this time.
tolerance	Real	Permissible tolerance of the deviation between setpoint and actual speed
off	Bool	Motor is to be switched off
on	Bool	Motor is to be switched on

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-84

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	-	-
off	Bool	Motor is switched off
on	Bool	Motor is switched on
auto	Bool	Automatic operation is active
local	Bool	Local operation is active
runDown	Bool	Motor is shut down
runUp	Bool	Motor is booted

Parameters		Data type	Description
	errorTrip	Bool	Emergency off is or was active
	errorExt	Bool	Error is or was active
	interlock	Bool	Locking
	repair	Bool	Repair mode
	opHoursLimitReached	Bool	Collective message for maintenance message
	warningExt	Bool	Collective message for warnings
	collectError	Bool	Collective message for errors
	errorFlagTrip	Bool	Emergency off is active
	errorFlagExt	Bool	Fault is active
	errorTimeout	Bool	Monitoring time for impermissible difference between reference speed and actual speed has expired
	cmdOn	Bool	Switch-on command
cmdAutSpeed		Real	Velocity specification in automatic operation
speed		Real	Current velocity

LBP_typeLog15MHIData

Table 2-85

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

Sinamics datasets

You can find information on the SinaSpeed block in the Online Help section of the TIA Portal. Search for “SinaSpeed Function Block” in the Online Help section

The structures “sxSendBuf” and “sxRecvBuf” are sent to the HMI. These two structures consist of a velocity value (either a set point or an actual value) and a tag that corresponds to the Standard Telegram 1.
For more information, click “Standard Telegram 1” in the Online Help section.

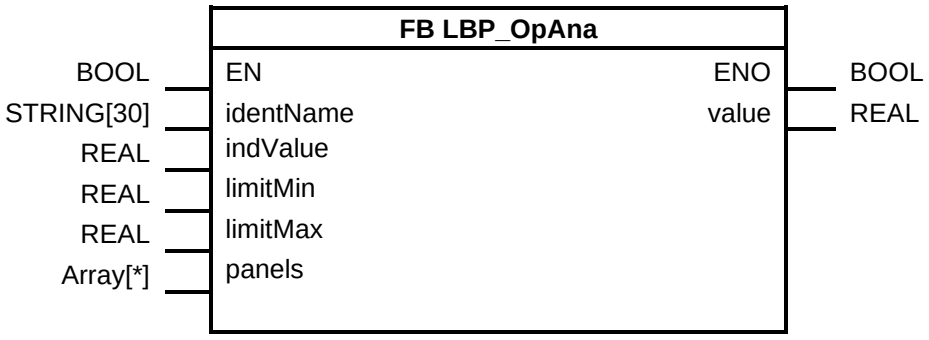
2.19 LBP_OpAna – Switching Analog Values

Brief description

This function can be used for switching analog values. This value can be monitored via defined limits. A value is only switched through if it is between the defined limits. Instead of a specified value, a process value can also be connected and monitored. If the parameterized value violates the ranges, warnings and errors are generated.

Block

Figure 2-19: LBP_OpAna



2.19.1 Interface Description of the PLC Block

Input parameters

Table 2-86

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
indValue	Real	Current analog value
limitMin	Real	Minimum value of the analog value
limitMax	Real	Maximum value of the analog value

Output parameters

Table 2-87

Parameters	Data type	Description
value	Real	Analog value.

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.19.2 Interface Description for HMI Communication

“settingsHMI

Table 2-88

Parameters	Data type	Description
Overwrite	Word	See “overwrite Bit assignment: Bit 0 – Value (“over”) Bit 1 – limitMin (“over”) Bit 2 – limitMax (“over”) Bit 3 – Track HMI value (“cmd”) (“true”: If overwrite-Bit 0 is set, the PLC value is transferred into the HMI value a single time)
value	Real	Analog value from HMI
limitMin	Real	Minimum of the analog value
limitMax	Real	Maximum of the analog value.

Parameters	Data type	Description
note	String[80]	See “note
caption	String[30]	Name
opStation	String[16]	See “opStation
format	String[10]	Display format on the HMI system (example: “s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit	String[10]	Unit that is to be displayed on the HMI system

“settingsPLC

Table 2-89

Parameters	Data type	Description
value	Real	Analog value
limitMin	Real	Minimum of the analog value
limitMax	Real	Maximum of the analog value

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

“statusHMI”

Table 2-90

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
value	Real	Current analog value

2.20 LBP_OpDig – Switching a Digital Signal

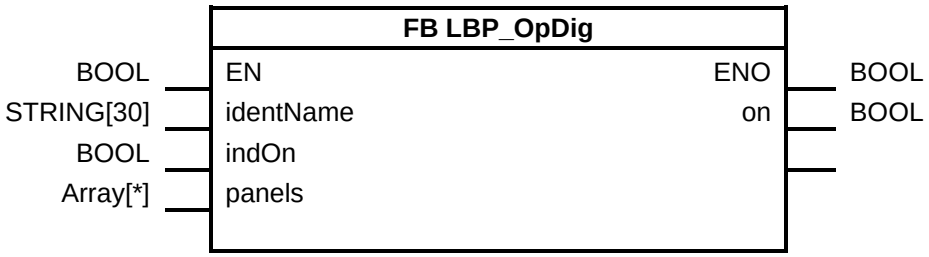
Brief description

The block allows the switching of a digital signal. Three operating modes are possible:

- Static: The block operates like an on/off switch
- Rising impulse: The block operates like a closing pushbutton, i.e. a pulse is output when actuated, otherwise the output is FALSE.
- Falling impulse: The block operates like an opening pushbutton, i.e. one cycle is output FALSE, otherwise the output is TRUE.

Block

Figure 2-20: LBP_OpDig



2.20.1 Interface Description of the PLC Block

Input parameters

Table 2-91

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indOn	Bool	Current digital value

Output parameters

Table 2-92

Parameters	Data type	Description
On	Bool	Specified digital value

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.20.2 Interface Description for HMI Communication

settingsHMI

Table 2-93

Parameters	Data type	Description
overwrite	Word	See "overwrite Bit assignment: Bit 0 – Value ("over") Bit 1 – value ("cmd")
on	Bool	Overwritten digital value from HMI
note	String[80]	See "note
opStation	String[16]	See "opStation
caption	String[30]	Name
pulseTime	Time	Switch-on time (applies to the types Positive edge and Negative edge)
type	Int	Type: "0": Static "1": Rising flank "2": Falling flank

settingsPLC

Table 2-94

Parameters	Data type	Description
on	Bool	Digital value.

Note All tags of "settingsPLC" are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-95

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
status	Word	State Bit assignment: Bit 0 – Output value Bit 1 – Current PLC value (value at input of the STEP 7 block) Bit 2 – Current HMI value (Overwrite bit 1) Bit 3 – overwrite Bit 0 ("true": You are currently working with the PLC value, "false": You are currently working with the HMI value)

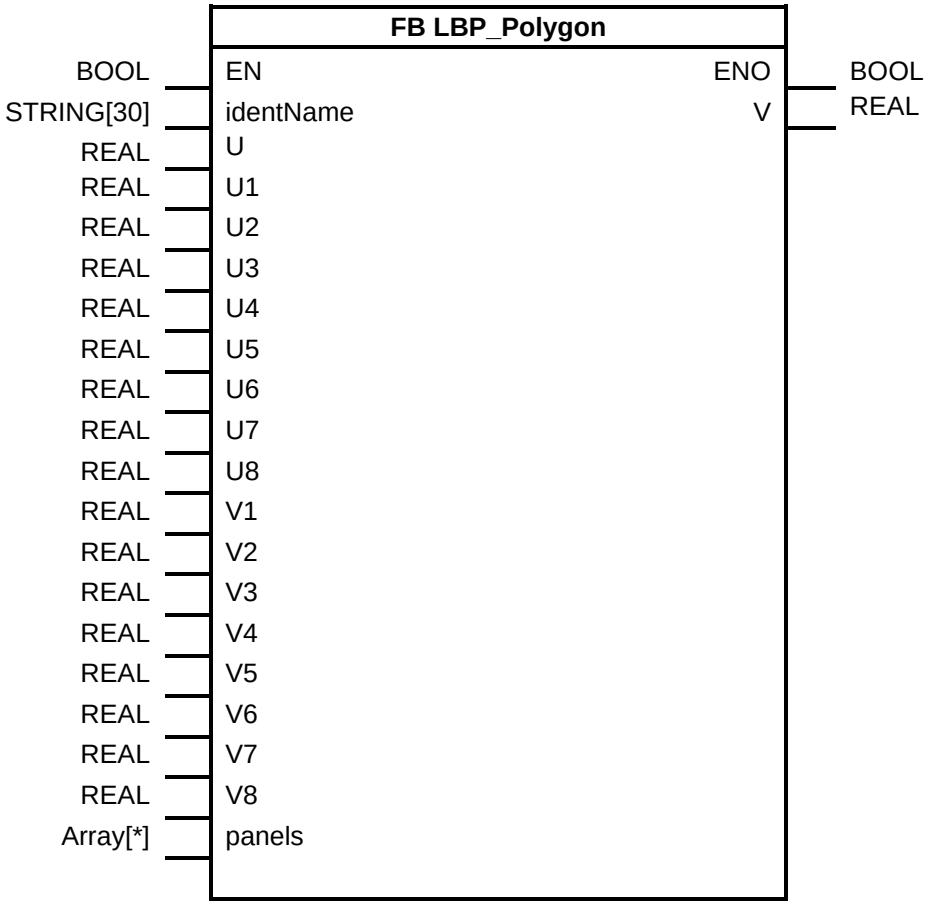
2.21 LBP_Polygon – Conversion of Values over 8 Interpolation Points

Brief description

The block creates a graph with 8 coordinates. The input value is converted to the output value by means of the graph. The block is used for the calculation of units.

Block

Figure 2-21: LBP_Polygon



2.21.1 Interface Description of the PLC Block

Input parameters

Table 2-96

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
U	Real	Input value
U1 bis 8	Real	U/V coordinate system U1 to 8 represent the U-values of the 8 coordinates from which the graph is formed.
V1 bis 8	Real	U/V coordinate system V1 to 8 represent the V-values of the 8 coordinates from

Parameters	Data type	Description
		which the graph is formed.

Output parameters

Table 2-97

Parameters	Data type	Description
V	Real	Output value (calculated V-value for the input value U)

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.21.2 Interface Description for HMI Communication

settingsHMI

Table 2-98

Parameters	Data type	Description
overwrite	Word	See "overwrite Bit assignment: Bit 0 – U(input) ("over") Bit 1 bit 8 – U1 to 8 ("over") Bit 9 to 16 – V1 to 8 ("over") Bit 17 – V(output) ("over")
note	String[80]	See "note
formatU	String[10]	Display format on the HMI system (example: "s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
formatV	String[10]	Display format on the HMI system (example: "s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unitU	String[10]	Unit that is to be displayed on the HMI system
unitV	String[10]	Unit that is to be displayed on the HMI system
coordinates	-	Coordinates
U	Real	Value to be converted (U)
U1 bis 8	Real	U/V coordinate system U1 to 8 represent the U-values of the 8 coordinates from which the graph is formed.
V1 bis 8	Real	U/V coordinate system V1 to 8 represent the V-values of the 8 coordinates from which the graph is formed.
V	Real	Converted value (V)
opStation	String[16]	See "opStation

settingsPLC

Table 2-99

Parameters	Data type	Description
U	Real	Input value
U1 bis 8	Real	U/V coordinate system

Parameters	Data type	Description
		U1 to 8 represent the U-values of the 8 coordinates from which the graph is formed.
V1 bis 8	Real	U/V coordinate system V1 to 8 represent the V-values of the 8 coordinates from which the graph is formed.

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-100

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
status	Word	State Bit assignment: Bit 0 – “true”: The HMI value is used for U (input) Bit 1 – “true”: The HMI value is used for V (output)
valueIn	Real	Current input value
valueOut	Real	Current output value

2.22 LBP_Select – Selection of Values

Brief description

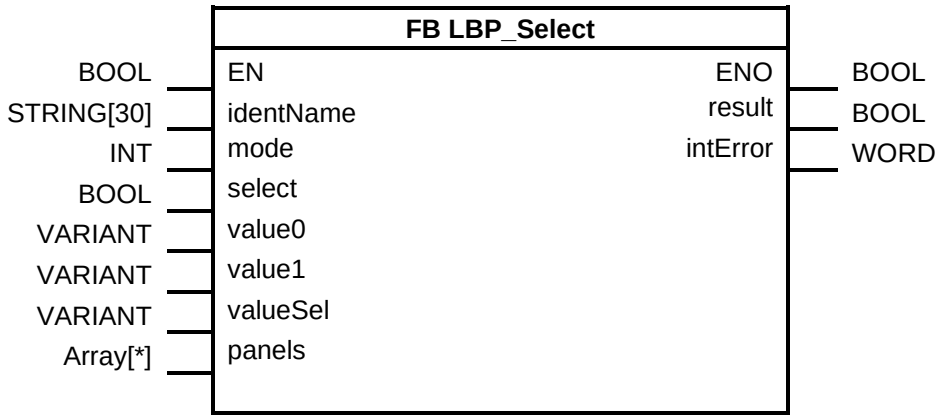
The block reads in two values and outputs one of them according to the following possible selection criteria:

- mode = “SEL”: The value is determined by the “select” input (select=FALSE: valueSel = value0, select = TRUE: valueSel = value1)
- mode = “MIN”: The smaller value is output
- mode = “MAX”: The greater value is output

The output “result” indicates which value is selected (result = FALSE: value0, result = TRUE: value1).

Block

Figure 2-22: LBP_Select



2.22.1 Interface Description of the PLC Block

Input parameters

Table 2-101

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
mode	Int	Selection mode: "1": Selection via switch "2": Selection of the greater value "3": Selection of the lesser value
select	Bool	Switch (affects mode 1)
value0	Variant	Value 1
value1	Variant	Value 2

Output parameters

Table 2-102

Parameters	Data type	Description
Result	Bool	Selected value
intError	Word	Error code

Input/output parameters

Table 2-103

Parameters	Data type	Description
panels	Array [*] of UDT	See 2.1.5 Input/Output for Panel Communication
valueSel	Variant	Selected value. The data types Bool, Byte, Int, DInt, Word, DWord, Real, and LReal are possible.

2.22.2 Interface Description for HMI Communication

settingsHMI

Table 2-104

Parameters	Data type	Description
note	String[80]	See "note
opStation	String[16]	See "opStation
format	String[10]	Display format on the HMI system (example: "s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit	String[10]	Unit that is to be displayed on the HMI system
overwrite	Word	See "overwrite Bit assignment: Bit 0 – Result ("over") Bit 1 – Result ("cmd") Bit 2 – Selected value ("over")
resultDWord	DWord	Selected value for the used data type DWord
resultReal	Real	Selected value for the used data type Real

settingsPLC

Table 2-105

Parameters	Data type	Description
mode	Int	Selection mode: "1": Selection via switch "2": Selection of the greater value "3": Selection of the lesser value
dataType	Int	Data type
resultPlc	Bool	Result when using the PLC values
result	Bool	Effective result

Note All tags of "settingsPLC" are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-106

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
value	LReal	Value
dataReals	UDT	-
dataRealValue0	Real	Value 1 for data type Real
dataRealValue1	Real	Value 2 for data type Real
dataRealResultValue	Real	Selected value for the used data type Real
dataDWords	UDT	-
dataDWordValue0	DWord	Value 1 for data type DWord

Parameters		Data type	Description
	dataDWordValue1	DWord	Value 2 for data type DWord
	dataDWordResultValue	DWord	Selected value for the used data type DWord

2.23 LBP_SetCrv – Time-Dependent Set Point Curve

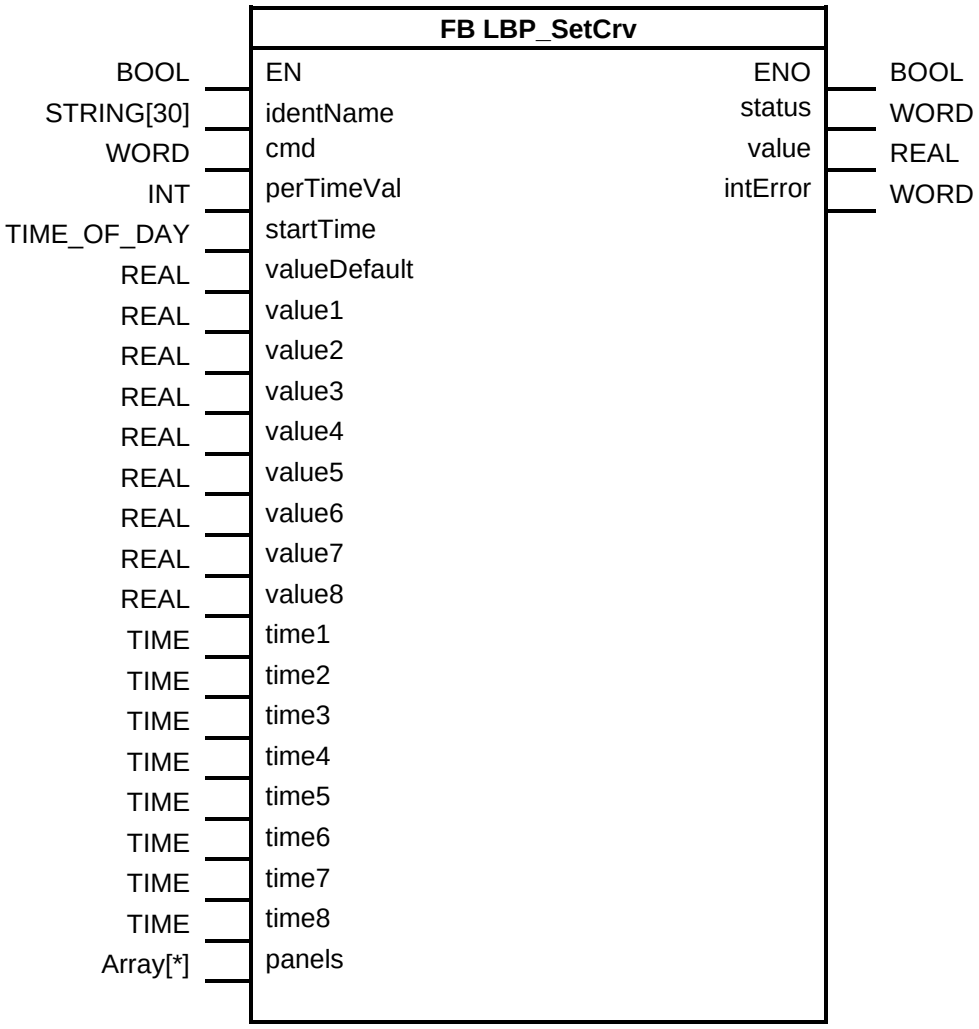
Note This function is not supported by a S7-1200 controller. To change that change the Datatype “Date-and-time” of the UDT “LBP_typeSetCrvStatus” parameter “actualTimeStart” to “DTL”, This parameter then won't be shown on Comfort panel anymore.

Brief description

This block can be used to create a time-dependent set point curve.

Block

Figure 2-23: LBP_SetCrv



2.23.1 Interface Description of the PLC Block

Input parameters

Table 2-107

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
cmd	Word	Command word: Bit 0 – Hold (freeze value) Bit 1 – Start with percentage time value (see perTimeVal) Bit 2 – Start time-independent (see startTime) Bit 3 – Interpolation Bit 8–15 – Percentage time value for the activation (1–100)
perTimeVal	Int	Percentage time value for start
startTime	Time_Of_Day	Time to start
valueDefault	Real	Start or default value
value1 bis 8	Real	Values of the 8 coordinates on the basis of which the

Parameters	Data type	Description
		curve is created
time1 bis 8	Time	Times for the 8 values of the 8 coordinates which were used to create the curve

Output parameters

Table 2-108

Parameters	Data type	Description
status	Word	Status word: Bit 0 – Execution of the curve is active Bit 2 – Start time reached (for time-dependent start) The high byte contains the current percentage time value.
value	Real	Output value V
intError	Word	Fault code: Unequal 0 – Error when reading the system time

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.23.2 Interface Description for HMI Communication

“settingsHMI

Table 2-109

Parameters		Data type	Description
note		String[80]	See “note
opStation		String[16]	See “opStation
format		String[10]	Display format on the HMI system (example: “s9999.999) (is automatically modified for WinCC Professional, however, not for WinCC Comfort)
unit		String[10]	Unit that is to be displayed on the HMI system
overwrite		Word	See “overwrite Bit assignment: Bit 0 – Allow switch by operator Bit 1..8 – value and time 1..8 (“over”) Bit 9 – start by time(“over”) Bit 10 – start by time (“value”) Bit 11 – start time(“over”) Bit 12 – interpolation (“over”) Bit 13 – interpolation (“value”) Bit 14 – value Default (“value”)
cmd		UDT	-
	deactivate	Bool	Holding processing
	actPercerntSetpoint	Bool	Starting with percentage time value
	strByTime	Bool	Activate time-dependent actuation
	interpolation	Bool	The curve is to be interpolated
perTimeVal		Int	Percentage time value for start
startTime		Time_Of_Day	Time to start
valueDefault		Real	Start or default value

Parameters	Data type	Description
values	Array[0..7] of Real	Values of the 8 coordinates on the basis of which the curve is created
times	Array[0..7] of Time	Times for the 8 coordinates on the basis of which the curve is created

settingsPLC

Table 2-110

Parameters		Data type	Description
cmd		UDT	-
	deactivate	Bool	Holding processing
	actPercerntSetpoint	Bool	Starting with percentage time value
	strByTime	Bool	Activate time-dependent actuation
	interpolation	Bool	The curve is to be interpolated
perTimeVal		Int	Percentage time value for start
startTime		Time_Of_Day	Time to start
valueDefault		Real	Start or default value
values		Array[0..7] of Real	Values of the 8 coordinates on the basis of which the curve is created
times		Array[0..7] of Time	Times for the 8 coordinates on the basis of which the curve is created

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-111

Parameters		Data type	Description
identName		String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
value		Real	Output value V
actualTime		Time	Current time value
actualTimeMax		Time	Full-scale value of the time axis
actualTmeStart		Time	Start value of the time axis
actualTimeRemain		Time	Current remaining time
actualStatus		Int	See “status” output
cmd		UDT	-
	deactivate		Holding processing
	actPercerntSetpoint		Starting with percentage time value
	strByTime		Activate time-dependent actuation
	interpolation	Bool	The curve is to be interpolated
perTimeVal		Int	Percentage time value for start

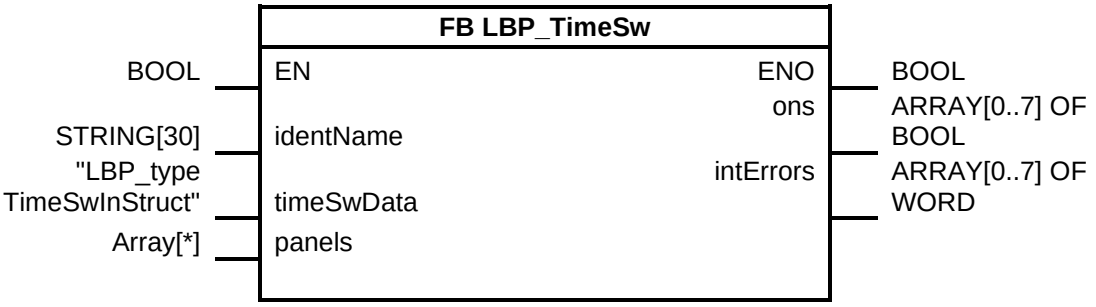
2.24 LBP_TimeSw – Time Switch

Brief description

The block realizes a time switch for different parameterizable cycles. For connection to the HMI, the block can be connected with “LibLBP_HMI_TimeSw”.

Block

Figure 2-24: LBP_TimeSw



2.24.1 Interface Description of the PLC Block

Input parameters

Table 2-112

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
timeSwData	UDT	Data of the time switches
timeSwDatas	Array[0..7] of UDT	-
typeCycle	Char	Type of time switch: “1”: Every week on Sunday “2”: Every week on Monday “3”: Every week on Tuesday “4”: Every week on Wednesday “5”: Every week on Thursday “6”: Every week on Friday “7”: Every week on Saturday “D”: Daily “H”: Hourly “M”: Once every minute
timeOn	Time_Of_Day	Switch-on time (time)
duration	Time	Runtime

Output parameters

Table 2-113

Parameters	Data type	Description
ons	Array[0..7] of Bool	Output signals for the 8 time switches
intErrors	Array[0..7] of Word	Internal error in the block: 56 – Error when reading the system time

Parameters	Data type	Description
		57 – Invalid value for type

Input/output parameters

See 2.1.5 Input/Output for Panel Communication

2.24.2 Interface Description for HMI Communication

settingsHMI

Table 2-114

Parameters	Data type	Description
note	String[80]	See “note
opStation	String[16]	See “opStation
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Type, start time and runtime for time switch with index 0 (“over”) Bit 1 – Type, start time and runtime for time switch with index 1 (“over”) Bit 2 – Type, start time and runtime for time switch with index 2 (“over”) Bit 3 – Type, start time and runtime for time switch with index 3 (“over”) Bit 4 – Type, start time and runtime for time switch with index 4 (“over”) Bit 5 – Type, start time and runtime for time switch with index 5 (“over”) Bit 6 – Type, start time and runtime for time switch with index 6 (“over”) Bit 7 – Type, start time and runtime for time switch with index 7 (“over”) Bit 8 to 15 – Overwrite the output signal with the value from the HMI system (overwrite Bit 16-23) (“over”) Bit 16 to 23 – Setting the output signal via the HMI system (“cmd”)
startTimes	Array[0..7] of Time_Of_Day	Switch-on times (time)
typeCycles	Array[0..7] of Byte	Types of time switches: “1”: Every week on Sunday “2”: Every week on Monday “3”: Every week on Tuesday “4”: Every week on Wednesday “5”: Every week on Thursday “6”: Every week on Friday “7”: Every week on Saturday “D”: Daily “H”: Hourly “M”: Once every minute
durationTimes	Array[0..7] of Time	Runtimes of the time switches

settingsPLC

Table 2-115

Parameters	Data type	Description
typeCycles	Array[0..7] of Byte	Types of time switches: “1”: Every week on Sunday “2”: Every week on Monday “3”: Every week on Tuesday “4”: Every week on Wednesday “5”: Every week on Thursday “6”: Every week on Friday “7”: Every week on Saturday “D”: Daily “H”: Hourly “M”: Once every minute
durationTimes	Array[0..7] of Time	Runtimes of the time switches
startTimes	Array[0..7] of Time_Of_Day	Switch-on times (time)
Ons	Array[0..7] of Bool	Output signals of the time switches using the input values at the STEP 7 block

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-116

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
typeCycles	Array[0..7] of Byte	Types of time switches: “1”: Every week on Sunday “2”: Every week on Monday “3”: Every week on Tuesday “4”: Every week on Wednesday “5”: Every week on Thursday “6”: Every week on Friday “7”: Every week on Saturday “D”: Daily “H”: Hourly “M”: Once every minute
startTimes	Array[0..7] of Time_Of_Day	Switch-on times (time)
durationTimes	Array[0..7] of Time	Runtimes of the time switches
ons	Array[0..7] of Bool	Outputs of the time switches
intErrors	Array[0..7] of Word	0 – no error 56 – Error when reading the system time 57 – Invalid value for type

2.25 LBP_Vlv – Simple Valve

Brief description

This block can be used to control valves with 2 position feedback signals (open/closed). The starting position can be declared as open or closed. Both feedback signals can be monitored.

The valve can be used in the following modes:

- Manual operation: Valve is operated via the HMI
- Automatic operation: Valve is controlled via the “cmdAutOpen” input
- Local operation: Valve is controlled via the “cmdLocOpen” input (local operation has the highest priority)
- Repair operation: Input “indRepair” is set. No control is possible and no display of errors and warnings

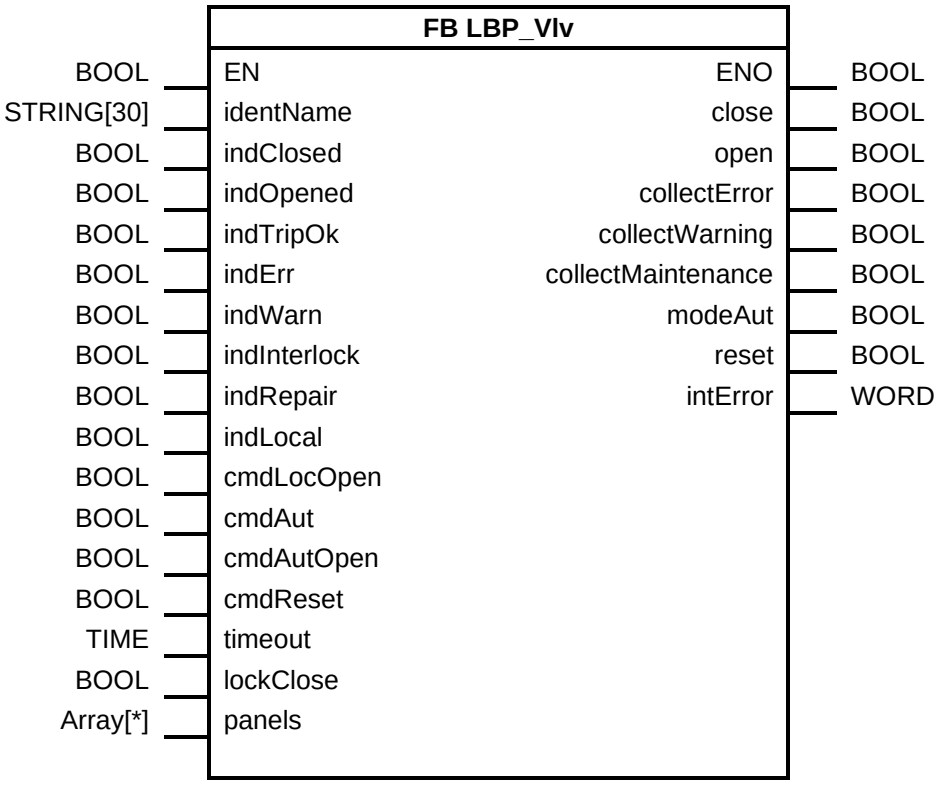
Error messages, status messages and group messages can be generated to monitor the valve.

The following functions can be used for maintenance:

- Runtime meter
- Counter change controls
- Limit counter controls (monitoring with limit greater than zero)

Block

Figure 2-25: LBP_Vlv



2.25.1 Interface Description of the PLC Block

Input parameters

Table 2-117

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indClosed	Bool	Feedback valve: true: Valve is closed
indOpened	Bool	Feedback valve: true: Valve is open
indTripOk	Bool	true: OK, false: Emergency stop
indErr	Bool	External error
indWarn	Bool	External Warning.
indInterlock	Bool	Interlocking active (switches the valve off or prevents it from being switched on)
indRepair	Bool	Activate repair mode (valve can only be controlled externally; errors and warnings are not displayed via faceplate)

Parameters	Data type	Description
indLocal	Bool	Activate on-site operation (valve is in local operation and can only be activated via cmdLocOpen)
cmdLocOpen	Bool	The valve is to be opened in local operation
cmdAut	Bool	Switching in external mode. The automatic mode is set with a positive edge.
cmdAutOpen	Bool	Open in automatic operation
cmdReset	Bool	Reset errors
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
lockClose	Bool	“false”: Valve closes at interlocking “true”: Valve opens at interlocking

Output parameters

Table 2-118

Parameters	Data type	Description
close	Bool	Control command “Close”
open	Bool	Control command “Open”
collectError	Bool	Collective error pending
collectWarning	Bool	Group warning pending
collectMaintenance	Bool	Maintenance required
modeAut	Bool	Automatic operation active
reset	Bool	Reset active
intError	Word	1: Error when reading the system time

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.25.2 Interface Description for HMI Communication

settingsHMI

Table 2-119

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 7 – Disabling the input “cmdAut” Bit 15 – Creation of a test entry for the log (“cmd”)
note	String[80]	See “note
opStation	String[16]	See “opStation

Parameters	Data type	Description
cmd	Word	Commands Bit assignment: Bit 0 – Close Bit 1 – Open Bit 2 – Manual operation Bit 3 – Automatic operation Bit 4 – Reset
opHours	Time	Maximum operating time
switches	DInt	Maximum number of switching operations
maintenance	UDT	Relevant values for a maintenance
switches	DInt	Current number of switching operations
opHours	Time	Operating time in hours
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-120

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-121

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	UDT	-
closed	Bool	Valve is closed
opened	Bool	Valve is open
auto	Bool	Automatic operation
local	Bool	Local operation activated
closing	Bool	Valve just closing
opening	Bool	Valve just opening
cmdAutOpen	Bool	The command for opening is pending for automatic operation
errorTrip	Bool	Emergency off is or was active
errorExt	Bool	Error is or was active
interlock	Bool	Locking
repair	Bool	Repair mode
collectMaintenance	Bool	Collective message for maintenance messages
collectWarning	Bool	Collective message for warnings

Parameters		Data type	Description
	collectError	Bool	Collective message for errors
	errorFlagTrip	Bool	Emergency off is active
	errorFlagExt	Bool	Fault is active
	errorTimeout	Bool	Switching time exceeded
	errorPlaus	Bool	Feedback messages not plausible (valve is closed and open at the same time, or neither closed nor open)
	opHoursLimitReached	Bool	Maximum operating time exceeded
	switchCounterLimitReached	Bool	Maximum switching procedures exceeded
	lockClose	Bool	“false”: Valve closes at interlocking “true”: Valve opens at interlocking
	warningExt	Bool	External warning is active

LBP_typeLog15MHIData

Table 2-122

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

2.26 LBP_VlvA – Analog Controlled Valve

Brief description

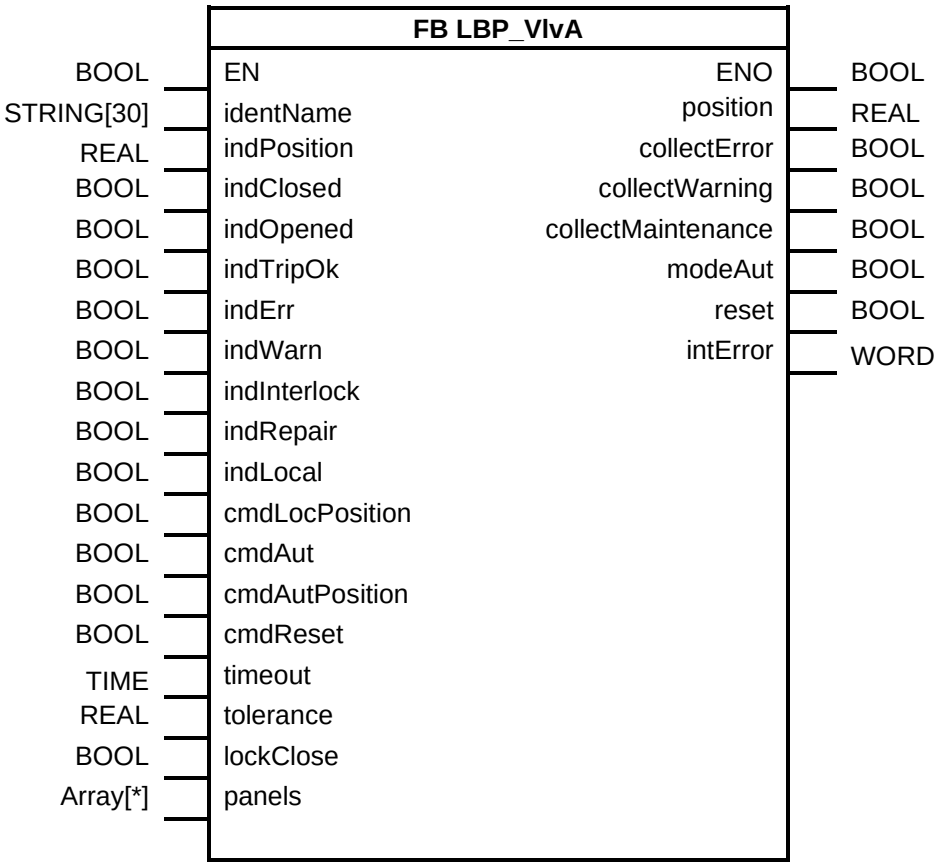
This block can be used to control an analog valve (damper) with an analog control and feedback signal. An analog actuation command can be set and is monitored. In addition, a monitoring time can be parameterized.

The valve can be used in the following modes:

- Manual operation: Valve is operated via the HMI
- Automatic operation: Valve is controlled via the “cmdAutPosition” input
- Local operation: Valve is controlled via the “cmdLocPosition” input (local operation has the highest priority)
- Repair operation: Input “indRepair” is set. No control is possible and no display of errors and warnings

Block

Figure 2-26: LBP_VlvA



2.26.1 Interface Description of the PLC Block

Input parameters

Table 2-123

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
indPosition	Real	Position feedback [0–100 %]
indClosed	Bool	“true”: currently closed
indOpened	Bool	“true”: currently opened
indTripOk	Bool	“true”: OK, “false”: Emergency stop
indErr	Bool	External error
indWarn	Bool	External Warning
indInterlock	Bool	Interlock active (see “lockClose” input)
indRepair	Bool	Activated repair mode (valve can only be controlled externally; errors and warnings are not generated)
indLocal	Bool	On-site operation activated (valve is in local operation and can only be activated via “cmdLocOpen”)
cmdLocPosition	Real	Position setting in local operation
cmdAut	Bool	Activate automatic operation (positive edge)
cmdAutPosition	Real	Position setting in automatic operation
cmdReset	Bool	The valve is to be reset

Parameters	Data type	Description
timeout	Time	Monitoring time. In the case of a difference between set position and actual position that lies outside the tolerance, “errorTimeout” is set after this time.
Tolerance	Real	Permissible tolerance of the deviation between setpoint and actual position
lockClose	Bool	“false”: Valve closes at interlocking “true”: Valve opens at interlocking

Output parameters

Table 2-124

Parameters	Data type	Description
position	Real	Set point for controlling the valve [0–100 %]
collectError	Bool	Group error
collectWarning	Bool	Group warning
collectMaintenance	Bool	Service necessary
modeAut	Bool	Automatic operation active
reset	Bool	Reset active
intError	Word	1 – Error when reading the system time

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.26.2 Interface Description for HMI Communication

settingsHMI

Table 2-125

Parameters	Data type	Description
timeout	Time	Monitoring time. In the case of a difference between set position and actual position that lies outside the tolerance, “errorTimeout” is set after this time. Monitoring time for the messages “timeout” and “plausibility”
posTolerance	Real	Permissible tolerance of the deviation between setpoint and actual position
limitClosed	Real	Specification from when the valve is considered to be closed (for display on HMI)
limitOpened	Real	Specification from when the valve is considered to be open (for display on HMI)
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 1 – Tolerance (“over”) Bit 2 – Position limit for closed is to be used (“cmd”) Bit 3 – Position limit for opened is to be used (“over”) Bit 7 – Block the input cmdAut (“cmd”) Bit 15 – Creation of a test entry for the log (“cmd”)

Parameters	Data type	Description
cmd	Word	Commands Bit assignment: Bit “2”: Active manual operation Bit “3”: Activate automatic operation Bit “4”: Reset valve
opHours	Time	Maximum operating time
maintenance	-	Relevant values for a maintenance
switches	DInt	Current number of switching operations
opHours	Time	Operating time in hours
note	String[80]	See “note
opStation	String[16]	See “opStation
position	Real	Position
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-126

Parameters	Data type	Description
timeout	Time	Monitoring time. In the case of a difference between set position and actual position that lies outside the tolerance, errorTimeout is set after this time.
Tolerance	Real	Permissible tolerance of the deviation between setpoint and actual position
closed	Bool	Closed
opened	Bool	Open

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values)

statusHMI

Table 2-127

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	-	-
closed	Bool	Valve is closed
opened	Bool	Valve is open
auto	Bool	Valve is in automatic operation
local	Bool	Valve is in local operation
closing	Bool	Valve closing
opening	Bool	Valve opening
errorTrip	Bool	Emergency off is or was active
errorExt	Bool	Error is or was active
interlock	Bool	Locking
repair	Bool	Repair mode

Parameters		Data type	Description
	opHoursLimitReached	Bool	Maximum operating time exceeded
	warningExt	Bool	External warning active
	collectError	Bool	Collective error active
	errorFlagTrip	Bool	Emergency off is active
	errorFlagExt	Bool	Fault is active
	errorTimeout	Bool	Switching time exceeded
	lockClose	Bool	“false”: Valve closes at interlocking “true”: Valve opens at interlocking
cmdAutPosition		Real	Position setting in automatic mode
position		Real	Current position

LBP_typeLog15MHIData

Table 2-128

Parameters		Data type	Description
data		Array[0..14] of "UDT"	-
	ts	DTL	Timestamp (local date and time)
	code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

2.27 LBP_3wVlv – Three-Way Valve

Brief description

This block can be used to control three-way valves. The starting position can be declared. The feedback signal for the position can be monitored.

The valve can take on the following positions:

Table 2-129

Positions	
	Position 1
	Position 2
	Position 3 (everything open)
	Position 0 (closed)

The valve can be used in the following modes:

- Manual operation: Valve is operated via the HMI
- Automatic operation: Valve is controlled via the “cmdAutOpen” input

- Local operation: Valve is controlled via the “cmdLocOpen” input (local operation has the highest priority)
- Repair operation: Input “indRepair” is set. No control is possible and no display of errors and warnings

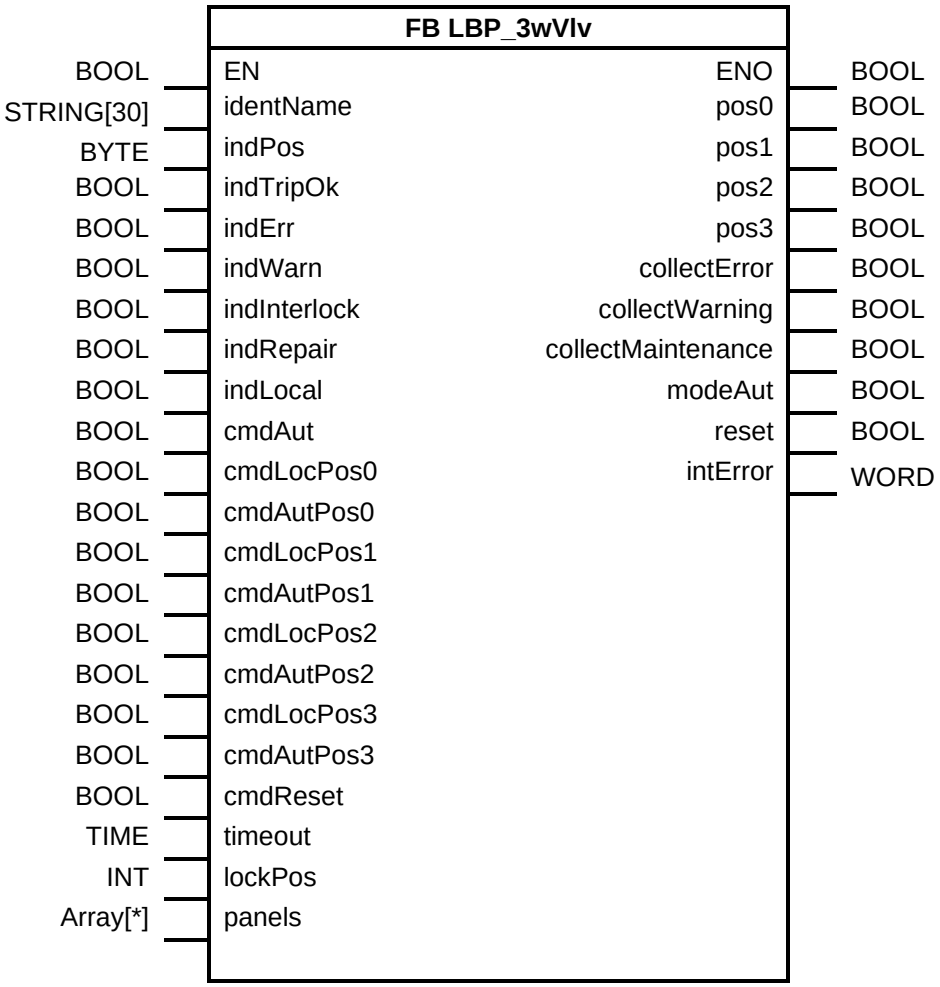
Error messages, status messages and group messages can be generated to monitor the valve.

The following functions can be used for maintenance:

- Runtime meter
- Counter change controls
- Limit counter controls (monitoring with limit greater than zero)

Block

Figure 2-27: LBP_3wVlv



2.27.1 Interface Description of the PLC Block

Input parameters

Table 2-130

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input "identName")
indPos	Byte	Current position
indTripOk	Bool	Emergency off pending
indErr	Bool	Active fault
indWarn	Bool	Warning active
indInterlock	Bool	Interlock active
indRepair	Bool	Repair mode active
indLocal	Bool	Local operation active
cmdAut	Bool	Automatic operation is actuated
cmdLocPos0	Bool	Position 0 is to be actuated in local operation
cmdAutPos0	Bool	Position 0 is to be actuated in automatic operation
cmdLocPos1	Bool	Position 1 is to be actuated in local operation
cmdAutPos1	Bool	Position 1 is to be actuated in automatic operation
cmdLocPos2	Bool	Position 2 is to be actuated in local operation
cmdAutPos2	Bool	Position 2 is to be actuated in automatic operation
cmdLocPos3	Bool	Position 3 is to be actuated in local operation
cmdAutPos3	Bool	Position 3 is to be actuated in automatic operation
cmdReset	Bool	The valve is to be reset
timeout	Time	Monitoring time for the messages "timeout" and "plausibility"
lockPos	Int	"0": Pos3 becomes active at interlocking "1": Pos0 becomes active at interlocking "2": Pos1 becomes active at interlocking "3": Pos2 becomes active at interlocking

Output parameters

Table 2-131

Parameters	Data type	Description
pos0	Bool	Position 0 is actuated
pos1	Bool	Position 1 is actuated
pos2	Bool	Position 2 is actuated
pos3	Bool	Position 3 is actuated
collectError	Bool	Collective error active
collectWarning	Bool	Collective warning active
collectMaintenance	Bool	Maintenance required
modeAut	Bool	Automatic operation is actuated
reset	Bool	Valve is to be reset
intError	Word	Bit 0 – true: System error (reading the system time has failed)

Input/output parameters

See 2.1.5 Input/Output for Panel Communication.

2.27.2 Interface Description for HMI Communication

settingsHMI

Table 2-132

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”
overwrite	Word	See “overwrite Bit assignment: Bit 0 – Timeout (“over”) Bit 7 – Block the input “cmdAut” (“cmd”) Bit 15 – Creation of a test entry for the log (“cmd”)
note	String[80]	See “note
opStation	String[16]	See “opStation
cmd	Word	Commands Bit assignment: Bit 0 – Position 0 is set Bit 1 – Position 1 is set Bit 2 – Position 2 is set Bit 3 – Position 3 is set Bit 4 – Manual operation is activated Bit 5 – Automatic operation is activated Bit 6 – Reset is performed
opHours	Time	Maximum number of switching operations
switches	DInt	Relevant values for a maintenance
maintenance	UDT	-
switches	DInt	Current number of switching operations
opHours	Time	Operating time in hours
alarmsInfo	UDInt	Alarm information
alarmsAck	UDInt	Alarm acknowledgment

settingsPLC

Table 2-133

Parameters	Data type	Description
timeout	Time	Monitoring time for the messages “timeout” and “plausibility”

Note All tags of “settingsPLC” are used for the transmission of input values to the HMI system (same tag name and values).

statusHMI

Table 2-134

Parameters	Data type	Description
identName	String[30]	Identification name (see Chapter 2.1.2 Cross-Block Input “identName”)
msgStatus	UDT	-
pos0	Bool	Position 0 active
pos1	Bool	Position 1 active
pos2	Bool	Position 2 active
pos3	Bool	Position 3 active
auto	Bool	Automatic operation active
local	Bool	Local operation active
positioning0	Bool	Positioning to Position 0
positioning1	Bool	Positioning to Position 1
positioning2	Bool	Positioning to Position 2
positioning3	Bool	Positioning to Position 3
cmdAutOpen	Bool	The command for opening is pending for automatic operation
errorTrip	Bool	Emergency off is or was active
errorExt	Bool	Error is or was active
interlock	Bool	Locking
repair	Bool	Repair mode
collectMaintenance	Bool	Collective message for maintenance messages
collectWarning	Bool	Collective message for warnings
collectError	Bool	Collective message for errors
errorFlagTrip	Bool	Emergency off is active
errorFlagExt	Bool	Fault is active
errorTimeout	Bool	Switching time exceeded
errorPlaus	Bool	Feedback messages not plausible (closed and open at the same time, or neither closed nor open)
opHoursLimitReached	Bool	Maximum operating time exceeded
switchCounterLimitReached	Bool	Maximum number of switching procedures exceeded
lockClose	Bool	“false”: Valve closes at interlocking “true”: Valve opens at interlocking
warningExt	Bool	external warning is active
status	UDT	-
ctrl	UDT	Current control signals
pos0	Bool	Control of position 0
pos1	Bool	Control of position 1
pos2	Bool	Control of position 2
pos3	Bool	Control of position 3
ctrlMan	UDT	Control signals in manual operation
pos0	Bool	Control of position 0
pos1	Bool	Control of position 1
pos2	Bool	Control of position 2
pos3	Bool	Control of position 3

Parameters	Data type	Description
ctrlAut	UDT	Control signals in automatic operation
pos0	Bool	Control of position 0
pos1	Bool	Control of position 1
pos2	Bool	Control of position 2
pos3	Bool	Control of position 3
ctrlLoc	UDT	Control signals in local operation
pos0	Bool	Control of position 0
pos1	Bool	Control of position 1
pos2	Bool	Control of position 2
pos3	Bool	Control of position 3
commands	UDT	-
manual	Bool	Activate manual operation
auto	Bool	Activation of the automatic operation
reset	Bool	Reset errors
resetRepeat	Bool	Repeats resetting

LBP_typeLog15MHIData

Table 2-135

Parameters	Data type	Description
data	Array[0..14] of "UDT"	-
ts	DTL	Timestamp (local date and time)
code	Word	Error code (see 2.28.3 Cross-Block Error Codes)

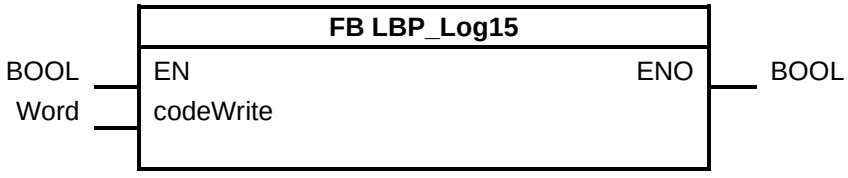
2.28 LBP_Log15 – Logging Data

Function description

This function saves errors in a log. The log corresponds to a ring buffer with a max. of 15 entries. This function is integrated in each block for the traceability of events.

Block

Figure 2-28: LBP_Log15



2.28.1 Interface Description of the PLC Block

Input parameters

Table 2-136

Parameters	Data type	Description
codeWrite	Word	Identification error depending on block

2.28.2 Interface Description for HMI Communication

"statData"

Table 2-137

Parameters			Data type	Description
statData			UDT	-
	data		Array[0..14] von UDT	Current number of switching operations
		ts	DTL	Current date and time
		code	Word	Identification number for error description

2.28.3 Cross-Block Error Codes

Table 2-138

Error code	Abbreviation	Description
1	< CHECK >	Check item
2	CMD MAN	Manual operation was activated
3	CMD AUT	Automatic operation was activated
4	CMD RESET	Block was reset
5	REPAIR OFF	Repair mode was deactivated
6	REPAIR ON	Repair mode was activated
7	LOCAL ON	Local mode was activated
8	LOCAL OFF	Local mode was deactivated
9	INTLK OFF	Interlock was released
10	INTLK ON	Interlock was activated
11	ERR TRIP	Emergency off was triggered
12	ERR EXT	External error was triggered
13	ERR TIMEOUT	Actuation has not been executed in the specified time
14	ERR PLAUS	Plausibility error
15	OFF	Was switched off
16	ON	Was switched on
17	NO RELEASE	Switch-on command but no switch-on release
18	RELEASE OFF	Switch-on release Off
19	RELEASE ON	Switch-on release On
20	WAIT SPEED	Waiting time for speed changeover is running
21	WAIT DIRECTION	Waiting time for change of rotation direction is running
22	ON2	Second speed is activated

Error code	Abbreviation	Description
23	ON1R	Speed 1 active in reverse mode
24	ON2R	Speed 2 active in reverse mode
25	CLOSE	Was closed
26	OPEN	Was opened
27	ERR HIGH	Measurement error in high range (peripheral measuring range not observed)
28	ERR LOW	Measurement error in low range (peripheral measuring range not observed)
29	ALARM HIGH	Alarm limit high violated
30	WARNING HIGH	High warning threshold violated
31	WARNING LOW	Low warning threshold
32	ALARM LOW	Alarm limit low violated
33	CMD OFF	Switch-off command
34	CMD OPTI	Should be initially optimized
35	CMD OPTA	Should be re-optimized
36	ERR INP	Input value lies outside the process value limits
37	ERR PER	Error at “Input_PER” input of the “PID_Compact” block called in the BPL block
38	ERR NOPT	Error during the re-optimization
39	ERR EOPT	Initial optimization/initialization could not be started
40	ERR SOPT	The set point was changed during an optimization
41	ERR ENOPT	An initial optimization/initialization was started during the re-optimization
42	ERR FEOPT	Error during initial optimization/initialization
43	ERR FNP	Error during re-optimization
44	ERR UWPI	Incorrect value for “Input”. Check tag format
45	ERR AWF	Calculation of the initial value failed. PID parameters have to be checked
46	ERR ATZF	Sampling time error has occurred. The block was not called within the sampling time of the cyclic interrupt OB.
47	ERR UWSP	Incorrect value as set point. Check tag format
48	ERR UWM	Incorrect value as manual value (“ManualValue”). Check tag format.
49	ERR UWSO	Incorrect value for “SubstituteOutput”. Check tag format.
50	ERR UWD	Incorrect value for “Disturbance”. Check tag format.
51	MOD OFF	Was deactivated
52	MOD OPTI	Was optimized
53	MOD OPTA	Was re-optimized
54	MOD AUT	Was in automatic mode
55	MOD MAN	Was in manual operation

3 Integrating the HMI Faceplates

3.1 WinCC (Runtime) Professional

3.1.1 General Steps

1. Open the library.
   a. The “Libraries” tab name is located at the left-hand edge in the TIA Portal. Select this object.
   b. Expand the “Global libraries” tab.
   c. To open the LBP, click the icon with the book and the green arrow.

Figure 3-1

   d. In the window that opens, select the file “LibraryOfBasicProcesses” with the type “Siemens TIA Portal V15 library”.
2. Copy the LBP into the project library.

Note When using SiVArc, it is important that the library structure of the Library for Basic Processes is also mapped in the project library in the same way. To this purpose follow the steps listed below.

   a. Expand the “Project library” tab in a way that you can see the project library on top and the global libraries below.
   b. Expand the “LibraryOfBasicProcesses” library.
   c. Expand the “Types” folder in the “LBasicProcessLibrary”.
   d. Select all the folders under “Types” and drag them into the project library into the “Types” folder.
   e. Repeat the two preceding steps for the “Master copies” folder.
3. Create the PLC program with the STEP 7 blocks from the library:
   The STEP 7 blocks are located under “Types > PLC > BlocksAndDatatypes”. If you drag a block into your program, the data types are automatically copied as well.

Note You can find additional information on the individual blocks in Chapter 2 Integrating the PLC Function Blocks.

4. Adapt the “Runtime settings”.
   a. You can find the item “Runtime settings” under your HMI Runtime in the project navigation. Open it by double-clicking it.
   b. You can find the “Use default design” point under “General > “Screen”. Remove the associated check mark and select “WinCC Dark” as the design. These design settings were used during the development of the LBP.
5. Copy the “Screenhandling” script.

a. You can find the folder “LBP_Skripts” in the project library under “Types > HMI > WinCCProfessional”. Drag this folder into the project navigation under your HMI Runtime into the “Scripts” folder.
b. Open the “Screenhandling” script.
c. If you use a different navigation than the navigation from the demo project, settings in the “Param-Area” of the script may be required. You can find more information on this under 3.1.4 Function of the “Screenhandling”.

6. Implement the User Administrator.
   a. You can find the folders “Authorizations” and “Usergroups” in the project library under “Master copies > HMI > WinCCProfessional > Usermanagement”. Select these two folders and drag them into the project navigation under your Runtime onto the “User administration” item.
   b. Create any users and assign them to a group.
   The following groups exist:

Table 3-1

User group	Authorization	Operator input options
LibLBP Administrator	Authorization_101 – LibLBP Operator Authorization_102 – LibLBP Engineer Authorization_104 – LibLBP Administrator	Administrators are allowed to operate everything.
LibLBP Engineer	Authorization_101 – LibLBP Operator Authorization_102 – LibLBP Engineer	Engineers are allowed to operate everything except changing the station names or depriving other users the exclusive operation possibility.
LibLBP Operator	Authorization_101 – LibLBP Operator	Operators are only authorized to operate the “Home” screens.

Note It is also possible to configure your own user groups with the respective permissions.

7. Harmonize the project

Note After all types from the project library have been transferred, it is necessary to perform a harmonization of the project with the library. All instance names are set correctly so that the internal access can take place correctly.

a. In the project library, right-click the “Types” folder and select “Library management”.
b. Click the “Harmonize project” button.

Figure 3-2

c. In the window that opens, remove the check mark for the PLC and select the option “Harmonize paths between project and library”. “Harmonize paths between project and library” must not be checked.

Note This point may not be omitted, the names of screens and scripts are otherwise not created correctly. As a result, the screens and functions are not referenced correctly anymore. This means that the function of the LBP would no longer exist.

8. Create the background screen
   a. You can find the “Main” screen in the project library under “Master copies > WinCCProfessional > DemoScreens”. Copy it to your project. This screen corresponds to the Start Screen of the demo project.
   b. If you like the Start Screen of the demo project, you can keep working with it. Otherwise, proceed as follows:
      i. Create a background screen as you like.
      ii. Copy the faceplate “swLimit” into your background screen.
      iii. Copy the empty faceplates from the Start Screen of the demo project into your background screen.
      iv. Your specific screens with the LBP symbols can be displayed within a faceplate on this background screen. To switch between the screens, create a navigation (The integration of the “Screens” symbols will be done later).
      v. With the navigation interfaces, the following script must be stored at the “Event”, so that—depending on the setting—all pop ups are closed or opened during the display selection:

Table 3-2

Code for navigation

```
If HMIRuntime.DataSet("settingCloseByScreenchange").Value = 1
Then
   If HMIRuntime.DataSet(ActiveScreenItem.Parent.ObjectName &
"array_1_screen") Is Nothing Then
      Exit Sub
   Else
      Dim counter
      For counter = 1 To
SmartTags("@NOTP::LBP_LimitFaceplates").Value
         HMIRuntime.DataSet(ActiveScreenItem.Parent.ObjectName &
"array_" & counter & "_screen").Value = 0
         HMIRuntime.DataSet(ActiveScreenItem.Parent.ObjectName &
"array_" & counter & "_parentScreen").Value = 0
         ActiveScreenItem.Parent.ScreenItems("swLBPScreen_" &
counter).Visible = False
      Next
   End If
End If
```

      vi. Create a button to open the pop-up for the general settings. This must call up the “Screenhandling” script as an event as follows.

Table 3-3

Call up of the general settings
`Screenhandling ", "LBP_Config"`

3.1.2 Continuing with SiVArc

Note

SiVArc is used in this library to reduce the configuration effort when implementing the symbols and alarms.

You can find general information on SiVArc in the manual under:

https://support.industry.siemens.com/cs/document/109755214

Steps for the integration with SiVArC

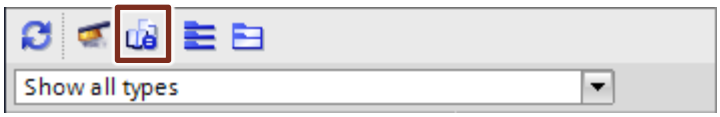
9. Copy with SiVArc
   a. You can find the “SiVArc” folder in the project navigation in your project in the “Shared data” folder. From the folder, open the list for “Copy rules”.
   b. You can find all SiVArc rules summarized into one group in your project library under “Master copies > WinCCComfort> SiVArc > SiVArcCopyrules”; these are necessary for copying screens, alarm classes, and tags.
   Drag them into the list for “Copy rules”.
   c. Right-click the header line, move the mouse to Show/Hide, and select the panel type and panel here.
   d. Ensure the check boxes of your panel type and panel are selected.
   e. Right-click in the project navigation on your HMI Runtime and then select “Generation of the visualization (SiVArc) > Generation of the visualization (SiVArc)”.
10. Harmonize the project.

Note

This is only necessary if you want to copy the folder structure from the library into your project

   a. In the project library, right-click the “Types” folder and select “Library management”.
   b. Click the “Harmonize project” button.

Figure 3-3



   c. In the window that opens, remove the check mark for the PLC and select the “Harmonize paths between project and library” option.

11. Generate the alarms.
   a. You can find the “SiVArc” folder in the project navigation in your project in the “Shared data” folder. Open the list for “Message rules” from the folder.

b. All the SiVArc rules that are required for the generation of alarms are summarized in one group in your project library under “Master copies > WinCCProfessional > SiVArc > SiVArcAlarmrules”.

c. Drag these into the list for “Message rules”.

d. Right-click the header line, move the mouse to Show/Hide, and select the Runtime type and Runtime station here.

e. Ensure the check boxes of your Runtime type and Runtime are selected.

f. In the project navigation, right-click your HMI Runtime and then select “Generation of the visualization (SiVArc) > Generation of the visualization (SiVArc)”.

g. Open the “HMI alarms” in the project navigation under your HMI Runtime. The generated bit messages open directly.

h. Open the “HMI alarms” again. This time, however, instead of the “Bit messages” tab, select the “Message groups” tab and then the subordinate “Class groups” tab.

i. Status tags and acknowledgment tags were also created for the LBP class groups. If you want to globally report whether an LBPP warning or an LBP alarm is pending, supplement the tags marked in red in your tag table. If not, delete these here.

Note

The status tag is structured as follows:
The first half of the bits of the status tag indicates whether the message or class group is active.
The second half of the bits of the status tag indicates whether the message has to be acknowledged.
Example:
Status tag = xyz with data type UINT
Status bit = 3
Bit 3 of the tag xyz indicates whether the message is active pending.
Bit 11 of the tag xyz indicates whether the message has been acknowledged.
You can find more information on the status tag in the online help of the TIA Portal.

12. Generate the symbols.

a. You can find the “SiVArc” folder in the project navigation in your project in the “Shared data” folder. Open the “Screen Rules” list from the folder.

b. You can find the rule group “Professional_LBPSymbols” in your project library under “Master copies > WinCCProfessional > SiVArc > SiVArcScreenrules”.

Note

These rules generate the “Symbols” screen with which you are familiar from your demo project, except that here, the only symbols that are generated are those for which the associated PLC blocks are available in your STEP 7 code. After the generation, you can drag or copy the symbols to the right positions in your project.

c. Select the “Professional_LBPSymbols” rule group and drag it into the list for “Screen rules”.

d. Right-click the header line, move the mouse to Show/Hide, and select the Runtime type and Runtime station here.

e. Ensure the check boxes of your Runtime type and Runtime are selected.

f. In the project navigation, right-click your HMI Runtime and then select “Generation of the visualization (SiVArc) > Generation of the visualization (SiVArc)”.

Note So that the symbols function properly, your background screen, in which the empty faceplates are stored, must be named “Main”.

During the symbol generation, symbols for the “Msg8” and “Intlk8” blocks are also generated, even if they belong to another block.

g. Create screens that you call up via the navigation in the background screen and drag or copy the symbols from the “DemoSymbols” screen into your own screens.

3.1.3 Continuing Manually

13. Copy all tags from LBP_GeneralTags.
    a. You can find all general HMI tags that are required for the use of the LBP in the project library under “Master copies > HMI > WinCCProfessional > GeneralTags”.
    b. Select these tags and drag them into any tag table in the project navigation under “HMI tags”.
    c. Select the “LBP_StationName” tag.
    d. You can find the “Properties” tab at the lower edge of the TIA Portal. Select these and select the “Properties” sub-tab.
    e. Click “Value” and specify the HMI station name as the start value. Each HMI station should have its own station name since this is used to lock the windows.
    f. Then select the "LBP_DoubleConfirmation" tag and as a start value assign a "1" if you want to be asked once again at important control elements whether you really want to control them.
    g. Then select the “LBP_LimitFaceplates” tag and assign the maximum number of faceplates which may be opened at the same time as a start value.

Note The number for “LBP_LimitFaceplates” may not exceed the number of empty faceplates that are stored on the Start Screen. This number is 16 in the demo project.

14. Copy the message classes.
    a. You can find all the alarm classes that you require for working with the LBP in the project library under “Master copies > HMI > WinCCProfessional > Alarming > Alarmclasses”.

b. Select all alarm classes and drag them to "HMI alarms" in the project navigation under your HMI Runtime.

15. Copy the screens.
    a. You can find the folder “HMI > WinCCProfessional > Screens” and the folder “LBP_PopUpScreens” under “Types” in the project library.
    b. Copy the folder “LBP_PopUpScreens” into the “Screens” folder in the project navigation under your HMI Runtime.
16. Copy the tags.
    a. Open the instances of the LBP blocks.
    b. Drag all “statDataXxx” structures of your LBP blocks into the HMI tags.
    c. For SimoDir and SimoRev, also drag the structures “DeviceDiag”, “Measurements”, and “Statistics” from the multi-instance “instSimocode” into the HMI tags.
    d. For the Sina, also drag the “ZSW1” tag (from “sxRecvBuf”) and the “STW1” tag (from “sxSendBuf”) from the “instSinaSpeed” multi-instance into the HMI tags.

17. Create the alarms
    a. Open the demo project next to your own project.
    b. Open the “HMI alarms” in the project navigation of the demo project under the HMI Runtime.
    c. Copy all the bit messages to your project that containe “Aggr8” in the name. If you do not use the “Aggr8” block in your project, take the messages of another block and observe the following note.

Note

In the case of blocks that are used more than once in the demo project, you have to ensure that only one set of messages is copied.
To do this, copy all messages that contain the block name (“Aggr8”) and have the same instance in the name.
The search function can help with this (selecting a message and pressing <Ctrl+F>).

    d. Right-click “ID” and click “Show/Hide”. Subsequently, set the check mark for “Status tag” and “Status bit”.
    e. From the first message, copy the message name of the instance name, for example “LBP_Aggr8_26_InstDemo_2_1_instAggr8” → instance name: “InstDemo_2_1_instAggr8”.
    f. On your keyboard, press the keys “<Ctrl+F>”. A search window opens on the right-hand side of the TIA Portal.
    g. Insert the copied instance name in the entry field under “Find” and enter the instance name that you use for this block in your own project under “Replace”. Subsequently the trigger tag, the status tag, and the acknowledgment tag should be recognized.
18. Connect the symbols.
    a. Open a screen (not a background screen or the Start Screen, but a screen on which you want LBP symbols displayed) in the project navigation under your HMI Runtime in the “Screens” folder.
    b. You can find all the symbols of the LBP in your project library under “Types > HMI > WinCCProfessional > Symbols”.
    Drag the desired symbols into your screen.

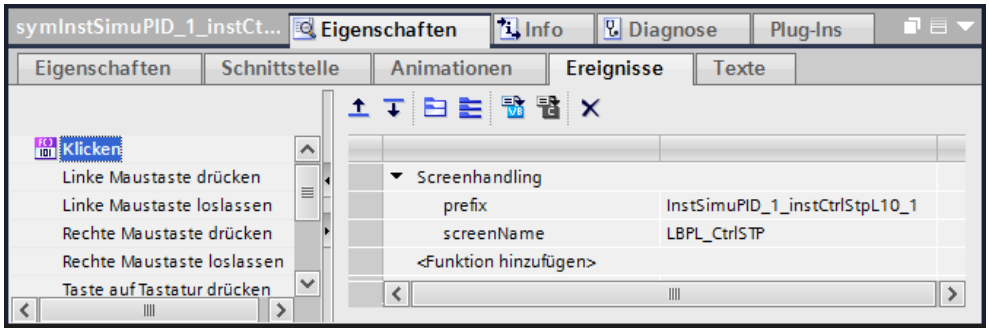
c. Select a symbol and open the “Properties” tab at the lower edge of the TIA Portal.
d. Select the subordinate “Interface” tab.
e. You must link the tag of the same name (UDT) from the PLC interface to this parameter.
f. Then select the “Events” tab.
g. Here under “Click”, call up the “Screenhandling” VB script.
h. Interconnect the parameters as follows:

Table 3-4

Screenhandling parameter	Interconnection
Prefix	Enter the name of the instance of the associated LBP block as a string.
screenName	Specify the LBP screen that is to open in the pop-up here. This is the screen that you have copied from the library to the “FirstScreens” folder.

Example:

Figure 3-4



3.1.4 Function of the “Screenhandling” Script

The “Screenhandling” function is responsible for opening the pop-up.

Note The function can only make faceplates visible; it cannot, however, generate new faceplates for the runtime. For this reason, the empty faceplates must be already stored on the “Start Screen”.

Setting options via general settings

The following setting options of the general settings determine how the pop-ups are opened or closed:

- Limit: Maximum number of opened faceplates
- Close faceplates at screen change
- Extd. Faceplates close with the “Home” faceplate. If the regular faceplate is closed, all windows that were opened by this faceplate close automatically as well—such as the message window, for example.
- Opening faceplates within the user interface
- Prevention of the opening of multiple faceplates with the same instance

Setting options within the script

When you open the script, you can find a description of the parameters at the beginning of the script and then a “Param-Area” in which you can make settings if necessary.

If you work with the navigation of the demo project, you must not make any changes here.

The following tags, to which static values are assigned, are located in the Param-Area:

- “offsetLeftOptional”
- “offsetTopOptional”

The Start Screen, in which the empty faceplates for the LBP are stored, is located in your project.

This screen includes a faceplate that represents the operating area and also shows the screens on which the symbols are displayed. The position of this faceplate has to be stored in “offsetLeftOptional” and “offsetTopOptional” relative to the Start Screen so that the position of the pop-up can be calculated correctly.

Figure 3-5

```
'Param-Area-------------------------------------------------------------------
'Define the coordinations (x, y) of the plant screen in the Main screen
Dim offsetLeftOptional, offsetTopOptional
offsetLeftOptional = 0
offsetTopOptional = 120
'-----------------------------------------------------------------------------
```

Nesting levels of screens

It is important for the functionality of the “Screenhandling” script that the empty faceplates are stored on the Start Screen and that they are always shown as the background.

However, it is possible to nest screens that contain symbols within other screens.

3.1.5 Working with Multiple Screens

You can find an FAQ that covers two variants for configuring multiple screens in the SIOS Portal under the URL “https://support.industry.siemens.com/cs/ww/de/view/109744837”. If you work with the LBP, you use the second variant “Splitting a Screen Across Multiple Monitors”.

Store the empty faceplates for the LBP on the background screen.

Since you now have several faceplates for the display of further screens with symbols, the position of the pop-up is calculated incorrectly. To prevent this dynamically, interconnect the tags in the “Param-Area” section in the “Screenhandling” script. The reason for this is that each faceplate has another position with regard to the background screen. Calculate the offsets according to the faceplate in which the symbols are clicked.

The following example queries on which screen the clicked symbol is located. Based on the assumption that the screen is always opened in the same faceplate, you can adjust the Screenhandling function as follows:

Figure 3-6

```
Sub Screenhandling(ByVal prefix, ByVal callItemObjectName, ByVal screenName, ByVal mainScreenName)
Dim offsetLeftOptional, offsetTopOptional
If  HMIRuntime.ActiveScreen.ObjectName = "Demo1" Then
    offsetLeftOptional = 0
    offsetTopOptional = 120
ElseIf HMIRuntime.ActiveScreen.ObjectName = "Demo2" Then
    offsetLeftOptional = 0
    offsetTopOptional = 120
End If
'-----------------------------------------------------------------------------------------------
```

Note When the “Screenhandling” script is updated, this information is lost and it must subsequently be added again.

3.2 WinCC Comfort/Advanced

3.2.1 General Steps

1. Open the library.
   a. The “Libraries” tab name is located at the left-hand edge in the TIA Portal. Select this object.
   b. Expand the “Global libraries” tab.
   c. To open the LBP, click the icon with the book and the green arrow.
   d. In the window, select the file “LibraryOfBasicProcesses” with the type “Siemens TIA Portal V15 library”.
2. Copy the LBP into the project library.

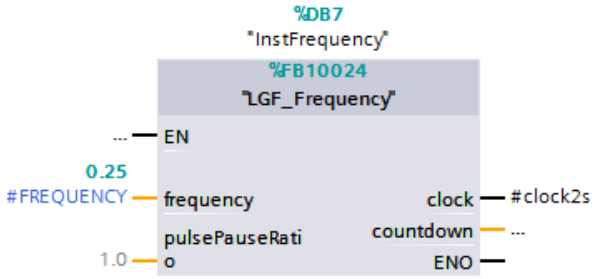
Note When using SiVArc, it is important that the library structure of the BasicProcessLibrary is mapped in the same way as the project library. To this purpose follow the steps listed below.

   a. Expand the “Project library” tab in a way that you can see the project library on top and the global libraries below.
   b. Expand the “LibraryOfBasicProcesses” library.
   c. Expand the “Types” folder in the “LibraryOfBasicProcesses”.
   d. Select all the folders under “Types” and drag them into the project library into the “Types” folder.
   e. Repeat the two preceding steps for the “Master copies” folder.
3. Create the PLC program with the STEP 7 blocks from the library.
   a. The STEP 7 blocks are located under “Types > PLC > BlocksAndDatatypes”. If you drag a block into your program, the data types are automatically copied as well.
   b. “PanelsBlock” is copied from the “Master copies\PLC” folder.

Note You can find additional information on the individual blocks in Chapter 2 Integrating the PLC Function Blocks.

4. Create a clock bit for triggering WinCC Comfort scripts of the blocks “SetCrv”, “Polygon”, “Intlk8”, and “Msg8”.
   a. In the project library under “Types > PLC > LGF > 7 signal generators”, you can find the function “LGF_Frequency “ from the “Library of general functions”.
   b. Call up the function cyclically in your PLC program.
   c. At the “frequency” input, enter a frequency of “0.5” and connect a static tag to the “Impulse” output that you then transfer using drag & drop into an HMI tag table.

Figure 3-7



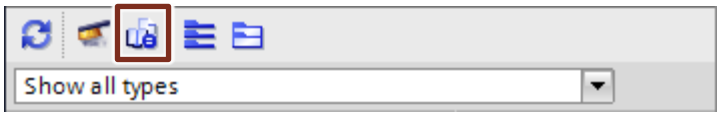
   d. “Select the tag in the HMI tag table and open the “Properties” of the tag at the lower edge in TIA Portal.
   e. If you want to visualize the functions on your Comfort Panel, select the “Events” tab and call up the LBP scripts “LBP_SetCrv”, “LBP_Polygon”, “LBP_Intlk8”, and “LBP_Msg8”.
5. Apply all scripts from the library (project library).
   a. You can find all scripts that are required for the use of the LBP in the project library under “Types > HMI > WinCC Comfort > Scripts”.
   b. Select these folders and drag them into the project navigation under “Scripts > VB scripts”.
6. Adapt the HMI to the “Runtime settings”.
   a. You can find the item “Runtime settings” under your HMI Runtime in the project navigation. Open it by double-clicking it. Under “General > Screen”, remove the check mark for “Default project style” and select “HMI Template Suite…” under “Operator Panel style”
7. Copy the pop-up screens
   b. You can find the “LBP_PopUpScreens” folder in the project library under “Master copies > HMI > WinCC Comfort”
   c. Copy the screens into the folder “Screen management > Pop-up screens”
8. Implement navigation
   a. Open the screen “Slide-in screen right” under “Screen management > Slide-in screens”
   b. Set the background color (32; 42; 46) under “General”
   c. Set the width to 220 under “Display”.
   d. You can find the “Navigation” copy template in the project library under “Master copies > HMI > WinCC Comfort > ScreenManagementComfort”. Drag it into the “Slide-in screen”.
   e. Customize the navigation to suit your project. The “Settings” navigation button must be kept.

9. Harmonize the project.

Note After all types from the project library have been transferred, it is necessary to perform a harmonization of the project with the library. All instance names are set correctly so that the internal access can take place correctly.

a. In the project library, right-click the “Types” folder and select “Library management”.
b. Click the “Harmonize project” button.

Figure 3-8



c. In the window that opens, remove the check mark for the PLC and select the option “Harmonize paths between project and library”. Do not check “Harmonize paths between project and library”.

Note This point may not be omitted, the names of screens and scripts are otherwise not created correctly. As a result, the screens and functions are not referenced correctly anymore. This means that the function of the LBP would no longer exist.

10. Implement the User Administrator
    a. You can find the “Authorizations” and “Usergroups” folders in the project library under “Master copies > HMI > WinCC Comfort > User management”. Select these two folders and drag them into the project navigation under your Runtime onto the “User administration” item.
    b. Create any users and assign them to one of the following groups:

Table 3-5

User group	Authorization	Operator input options
LBP Administrator	Authorization_101 – LBP Operator Authorization_102 – LBP Engineer Authorization_104 – LBP Administrator	Administrators are allowed to operate everything.
LBP Engineer	Authorization_101 – LBP Operator Authorization_102 – LBP Engineer	Engineers have all rights with the exception of changing the station names or depriving other users of the exclusive operation possibility.
LBP Operator	Authorization_101 – LBP Operator	Operators are only authorized to operate the Home screens.

Note It is also possible to configure your own user groups with the respective permissions.

11. Copy the tasks
    a. You can find two tasks under “Master copies > HMI > WinCC Comfort > ScheduledTasks” in the project library.

The task “User Name” is used to display the current user and is required if the Template Suite is used as the basis for design.
The task “LBP_SetStationName” sets the station name once at the start of Runtime. This is necessary because the station name cannot be given a default value. The station name can be changed at runtime.

b. Select the tasks and drag them to the “Task planner” item in the project navigation under your HMI Runtime.
c. Open the tag tables under “HMI tags” and adjust the connection if necessary.
d. If your data block is not called “PanelsBlock” or if you have to access “Panel[2]” with your panel to be configured, modify the “PLC” tag. The fastest way to do this is via the “Find & Replace” function with the key combination <Ctrl+F>.
e. Then select the “LBP_DoubleConfirmation” tag and assign a start value of “1” if you want to be asked once again at important control elements whether you really want to control them.

3.2.2 Continuing with SiVArc

Note

At this library, SiVArc is used to reduce the configuration work during the implementation of the symbols and alarms.

You can find general information on SiVArc in the manual under:

https://support.industry.siemens.com/cs/document/109755214.

Steps for the integration with SiVArc

12. Copy with SiVArc
    f. You can find the “SiVArc” folder in the project navigation in your project in the “Shared data” folder. From the folder, open the list for “Copy rules”.
    g. You can find all SiVArc rules summarized into one group in your project library under “Master copies > WinCCComfort> SiVArc > SiVArcCopyrules”; these are necessary for copying templates, screens, alarm classes, and tags.
    Drag them into the list for “Copy rules”.
    h. Right-click the header line, move the mouse to Show/Hide, and select the panel type and panel here.
    i. Ensure the check boxes of your panel type and panel are selected.
    j. Right-click in the project navigation on your HMI Runtime and then select “Generation of the visualization (SiVArc) > Generation of the visualization (SiVArc)”.
13. Harmonize the project.

Note

This is only necessary if you want to copy the folder structure from the library into your project

    k. In the project library, right-click the “Types” folder and select “Library management”.
    l. Click the “Harmonize project” button.

Figure 3-9

m. In the window that opens, remove the check mark for the PLC and select the “Harmonize paths between project and library” option.

14. Generate the alarms.
    a. You can find the “SiVArc” folder in the project navigation in your project in the “Shared data” folder. Open the list for “Message rules” from the folder.
    b. All the SiVArc rules that are required for the generation of alarms are summarized in one group in your project library under “Master copies > WinCCComfort > SiVArc > SiVArcAlarmrules”.
    Drag these into the list for “Message rules”.
    c. Right-click the header line, move the mouse to Show/Hide, and select the panel type and panel here.
    d. Ensure the check boxes of your panel type and panel are selected.
    e. Right-click in the project navigation on your HMI Runtime and then select “Generation of the visualization (SiVArc) > Generation of the visualization (SiVArc)”.
    f. Open the “HMI alarms” in the project navigation under your HMI Runtime. The generated bit messages open directly.

Note

If you have not given the same name to your instance DB as the “identName” (possible would be “InstIdentName”, “instIdentName”, or “IdentName”), the message text must be adapted. In this case, replace the instance name in your message text through the IdentName or directly through a tag reference to the “statSymbolXxxx.identName” tag.

With the messages of the “Msg8” function block, it is advisable to modify the message text after the generation. Replace “Message x” by a tag reference to the “statSymbolMsg8.names[x]” tag. To do this, proceed as follows:

1. Delete the “Message x” text.
2. Right-click and select “Insert dynamic parameter (tag)...”.
3. Select the “statSymbolMsg8.names[x]” tag of the respective instance. The name of the bit message was named after the instance during the generation.

15. Generate the symbols.
    a. You can find the “SiVArc” folder in the project navigation in your project in the “Shared data” folder. Open the “Screen Rules” list from the folder.
    b. You can find the rule group “Comfort_BPLSymbols” in your project library under “Master copies > WinCCComfort > SiVArc > SiVArcScreenrules”.

Note

These rules generate the “Symbols” screen with which you are familiar from your demo project, except that here, the only symbols that are generated are those for which the associated PLC blocks are available in your STEP 7 code. After the generation, you can drag or copy the symbols to the right positions in your project.

c. Select the “Comfort_BPLSymbols” rule group and drag it into the list for “Screen Rules”.

d. Right-click the header line, move the mouse to Show/Hide, and select the panel type and panel here.

e. Ensure the check boxes of your panel type and panel are selected.

f. Right-click in the project navigation on your HMI Runtime and then select “Generation of the visualization (SiVArc) > Generation of the visualization (SiVArc)”.

Note

So that the symbols function, your background screen on which the symbols are stored must be named “Main”. If this is not the case, the event for setting the “LBP_PlantScreen” tag has to be adapted. The name of the background screen has to be stored here.

If you have not given the same name to your instance DB as the “identName” (possible would be “InstIdentName”, “instIdentName”, or “IdentName”), the event for setting the tag “LBP_identName” must be adapted to the identName defined in your PLC program.

During the symbol generation, symbols for the “Msg8” and “Intlk8” blocks are also generated, even if they belong to another block.

g. Drag or copy the symbols from the “Symbols” screen into their own screens (in the ideal case, to a screen with the name “Main”).

3.2.3 Continuing Manually

16. Copy the HMI tags from the copy template (master copies).
    a. You can find all HMI tags that are required for the use of the LBP in the project library under “Master copies > HMI > WinCC Comfort > HMI Tags”.
    b. Select this tag table and drag them into the project navigation under “HMI tags”.
    c. Open the tag tables under “HMI tags” and adjust the connection if necessary.
    d. If your data block is not called “PanelsBlock” or if you have to access “Panel[2]” with your panel to be configured, modify the “PLC” tag. The fastest way to do this is via the “Find & Replace” function with the key combination <Ctrl+F>.
    e. Then select the “LBP_DoubleConfirmation” tag and assign a start value of “1” if you want to be asked once again at important control elements whether you really want to control them.
17. Transfer the symbol UDTs into the HMI tags using drag & drop.
    a. The instances of the LBP blocks contain a UDT with the name “statSymbolXxxxx” in the static area. Drag this UDT into one of your tag tables. Repeat this procedure for all of the LBP block instances that you want to visualize on this panel.
18. Apply the screens from the library.
    a. You can find all screens that are required for the use of the LBP in the project library under “Types > HMI > WinCC Comfort > Screens”.
    b. Select these folders and drag them into the project navigation under “Screens”.

c. For motors and valves, “Intlk8” and “Msg8” must be available if this is parameterized in the PLC program.

19. Copy the LBP templates
    a. You can find the “LBP_Templates” folder in the project library under “Master copies > HMI > WinCC Comfort”
    b. Copy the screens into the folder “Screen management > Templates”
20. Copy the “Screen management” from the “Master copies”.
    a. You can find the “Templates” folder in the project library under “Master copies > HMI > WinCC Comfort > ScreenManagementComfort”. Copy this into the “Screen management > Templates” folder.
21. Copy the message classes.
    a. The project library contains all the alarm classes that you require for working with the LBP under “Master copies > HMI > WinCCProfessional > Alarming > Alarmclasses”.
22. Select all alarm classes and drag them into the project navigation within your HMI Runtime to the “HMI alarms” item.
23. Create the alarms.
    a. Open the demo project next to your own project.
    b. Open the “HMI alarms” in the project navigation of the demo project under the HMI Runtime.
    c. Copy all the bit messages into your own project in which “Aggr8” is contained in the name. If you do not use the “Aggr8” block in your project, take the messages of another block and observe the following note.

Note

In the case of blocks that are used more than once in the demo project, only one set of messages has to be copied.

Support:
Copy all messages that contain the block name (“Aggr8”) and that have stored the same instance in the name.
The search function can help with this (selecting a message and pressing <Ctrl+F>).

    d. From the first message copy the message name of the instance name, for example “LBP_Aggr8_26_InstDemo_2_1_instAggr8” → instance name: “InstDemo_2_1_instAggr8”.
    e. On your keyboard, press the keys “<Ctrl+F> and a search window opens on the right-hand side in TIA Portal.
    f. Insert the copied instance name in the entry field under “Find” and enter the instance name that you use for this block in your own project under “Replace”. Subsequently, the trigger tag and acknowledgment tags are recognized.
    g. In the “Message text”, column replace the text before “–” through the stored identName at the associated FB or directly through a tag reference to the “statSymbolXxxx.identName” tag.
    h. At the messages of the function block “Msg8”, you adapt the tag reference to the “statSymbolMsg8.names[x]” tag from the corresponding instance.
24. Connect the symbols.
    a. In the project navigation under your HMI Runtime → Screens, open a screen on which you want to display the LBP symbols.
    b. You can find all the symbols of the LBP in your project library under “Types” → “HMI” → “WinCCComfort” → “Symbols”.

c. Drag the desired symbols into your screen.

d. Select a symbol. Open the “Properties” tab at the lower edge in TIA Portal.

e. Select the subordinate “Interface” tab.

f. At the parameter. connect the tag by the same name (UDT) of the PLC interface.

g. Then select the “Events” tab.

h. Add the following events:

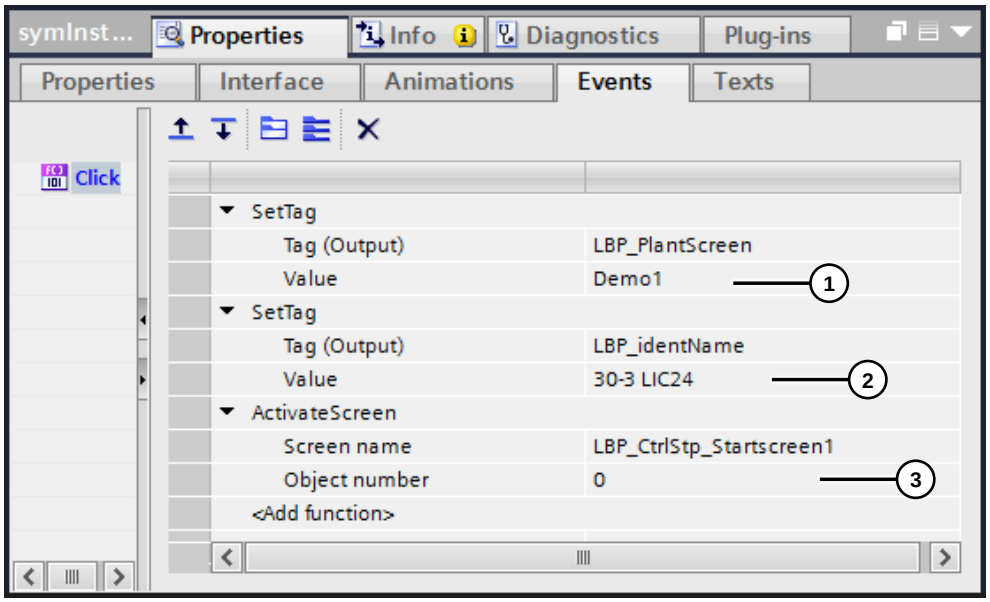
(i) “SetTag”, to write the screen name of the background screen to the “LBP_PlantScreen” tag.

(ii) “SetTag”, to write the “ident name” of the block associated with the symbol to the “LBP_identName” tag. This is necessary to request the correct data when the symbol is clicked. The name must be statically stored.

(iii) “ActivateScreen”, to open the respective screen when you click the symbol.

Example:

Figure 3-10



3.3 WinCC V7.5

Note

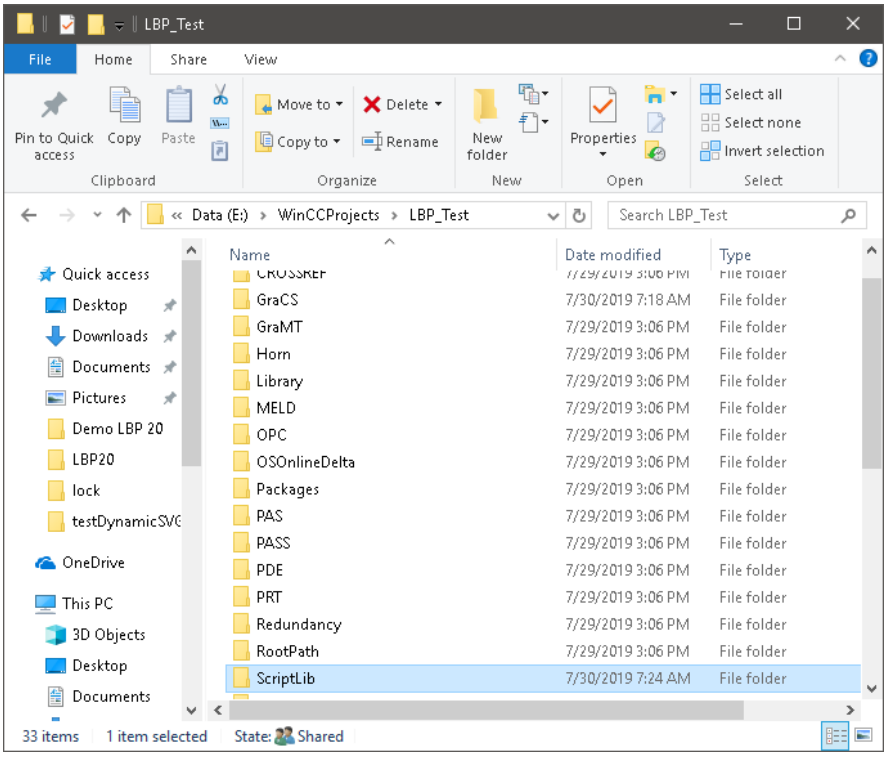
Importing the tags into WinCC V7.5 and dynamically adjusting the display format at runtime only works if WinCC V7.5 Upd3 or higher is installed.

The library elements are not designed for use in combination with the Basic Process Control.

3.3.1 Preparing the WinCC Project

1. Open your own WinCC project and the “LBP_WinCCV75MasterCopies” folder in separate Windows Explorer windows. The “LBP_WinCCV75MasterCopies” folder is available for download in the same SIOS entry.
2. Copy the screens.
   a. In the “LBP_WinCCV75MasterCopies” folder, navigate to the subfolder “00_ProcessScreens”, select the content, and press the key combination “<Ctrl+C>.”
   b. Navigate to the “GraCS” folder in your WinCC project folder and press the key combination “<Ctrl+V>”.
3. Copy the scripts.
   a. In the “LBP_WinCCV75MasterCopies” folder, navigate to the folder “01_Scripts”, select the “ScriptLib” folder, and press the key combination “<Ctrl+C>.”
   b. Navigate back from the “GraCS” folder into your higher-level project folder and press the key combination “<CTRL+V>”.

   Figure 3-11



4. Import the texts.

a. Open your own project in the WinCC Explorer.
b. Open the “Text Library”.
c. Click “Edit” and select “Import”.
d. In the opened window, navigate to the folder “LBP_WinCCV75MasterCopies > 02_Texts”.
e. Check whether the “*.txt” format type is selected on the right.
f. Select the file “LBP_TextLibrary.txt”.

5. Import the “Text and graphics lists”.
   a. Open “Text and graphics lists” in WinCC Explorer.
   b. Click “Edit” and select “Import”.
   c. In the opened window, navigate to the folder “LBP_WinCCV75MasterCopies > 02_Texts”.
   d. Select the “LBP_TextList.txt” file.
   e. Repeat the procedure for the “LBP_GraphicList.txt” file.
6. Import the User Administrator.
   a. Open “User administrator” in WinCC Explorer.
   b. Click “Edit” and select “Import”.
   c. In the window that opens, navigate to the folder “LBP_WinCCV75MasterCopies > 03_UserAdministration”.
   d. Select the “LBP_UserAdministration.txt” file.
   e. Through the import, a user is already created in each group as an example. If you want to use these users, adapt the passwords of the users. If you do not want to use them, delete them and create your own users.
7. Import the internal tags.
   a. Open “Tag Management” in WinCC Explorer.
   b. Click “Edit” and select “Import”.
   c. In the opened window, navigate to the folder “LBP_WinCCV75MasterCopies” → “04_InternalTags”.
   d. Select the “LBP_ InternalTags.txt” file.

3.3.2 Integrating the block-specific data

1. Export the tags from the TIA Portal.
   a. Open your TIA Portal project that contains the control system with the LBP blocks.
   b. Right-click the PLC and select “Export to SIMATIC SCADA”.
   c. Export the data.

Note You can find additional information on the topic in the SIOS Portal under the entry ID 109748955, SIMATIC SCADA Export for TIA Portal.

2. Create an S7-1500 driver.
   a. Open “Tag Management” in WinCC Explorer.
   b. In the navigation area, right-click “Tag Management”, select “Add new driver”, and click “SIMATIC S7-1200, S7-1500 Channel”.

c. Expand the structure of the driver, right-click “OMS+”, and select the “New Connection”.

d. Right-click your newly created connection and click “Connection Parameters”.

e. Set the connection parameters.

3. Import the AS symbols from the TIA Portal.

a. Right-click “Tag Management” in the navigation area on your added connection, go to “AS symbols”, and click “Load from file”.

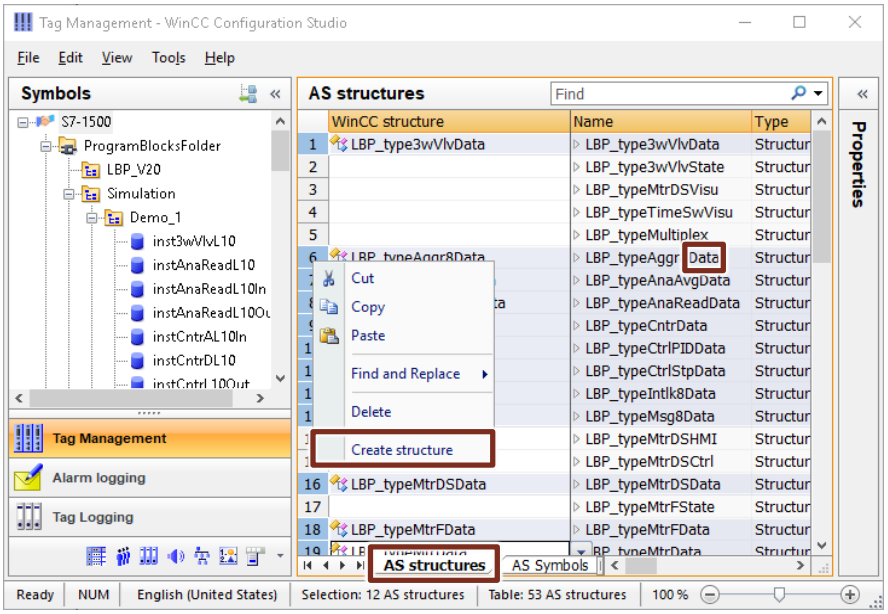
b. In the window that opens in the bottom right, select “TIA Portal Export File (*.zip)” in the file format selection and navigate to the zip file previously exported from the TIA Portal.

c. The “Symbols” open after the import. Select the worksheet “AS structures” (see Figure 3-12).

d. Select all lines whose name ends with “Data”.

e. Right-click into the first column (Number) of a selected line and click “Create structure”.

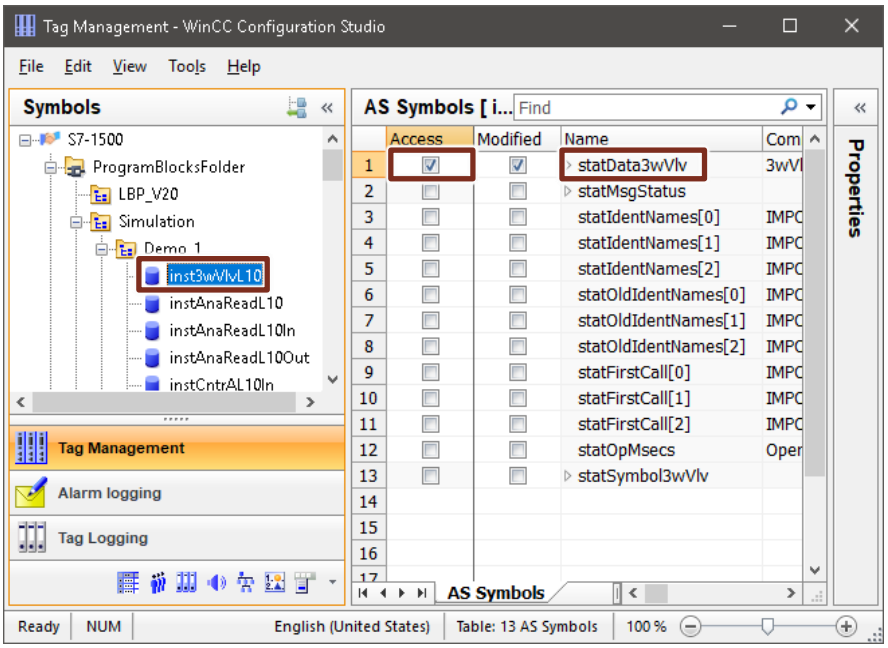
Figure 3-12



f. In the navigation, select an instance that is to be visualized on your WinCC system.

g. Set the check mark for “Access” for the structure with the name “statDataXxxx”.

Figure 3-13

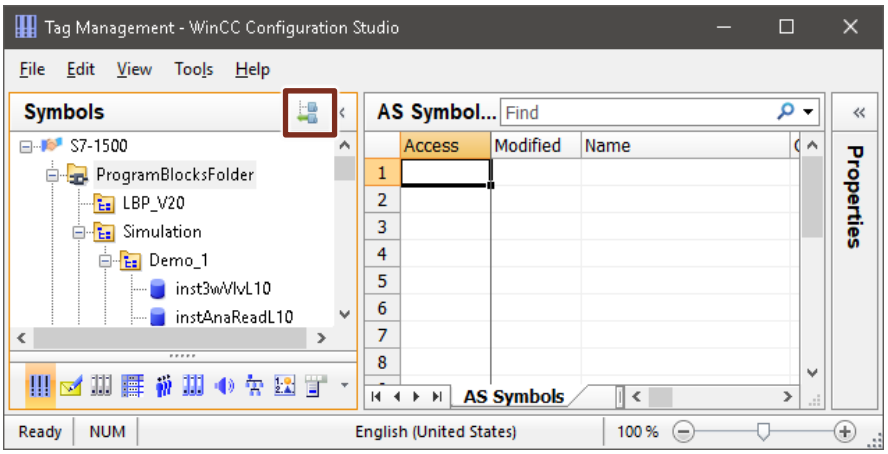


h. Repeat the procedure for all instances that you want to visualize.

4. Group the tags for better clarity (optional)

a. Switch from “Symbols” back to “Tag Management”.

Figure 3-14



b. Right-click your connection and click “New Group”.

c. Name the group as you want (example: “3wVlv”).

d. Select a structure under “structure tags”.

e. Assign a group to the displayed structures in the “Group” column.

5. Create the messages.

a. In the Windows Explorer, navigate to the folder “LBP_WinCCV75MasterCopies > 05_AlarmLogging” and open the “LBP_Messages.xlsx” file with Microsoft Excel.

b. A set of messages for each block is contained in the file. Adapt it to your program by opening the search via the key combination “<Ctrl+F>”.

c. Select the “Replace” tab in the search.

d. Under “Find what”, enter the currently used instance name (for example: “instMtr”), and under “Replace with”, enter the instance name that you assigned in the STEP 7 program.

e. If you use a block more regularly, copy the set of messages and insert it again below. The messages must then be adapted to the respective instance name again.
Make sure that each number is only assigned once in the “Number” column.

f. Open “Alarm Logging” in WinCC Explorer.

g. Click “Edit” and select “Import”.

h. In the opening window, select the file format “Excel Workbook (*.xslx)” in the bottom right.

i. Navigate to the previously edited Excel file.

j. Click “Import”.

6. Change/create the background screen.

a. In WinCC Explorer, select the “Graphics Designer” and double-click to open the “Main.Pdl” screen.

b. If you like the background screen of the demo project, you can keep working with it. Otherwise, proceed as follows:

i. Create a background screen as you like.

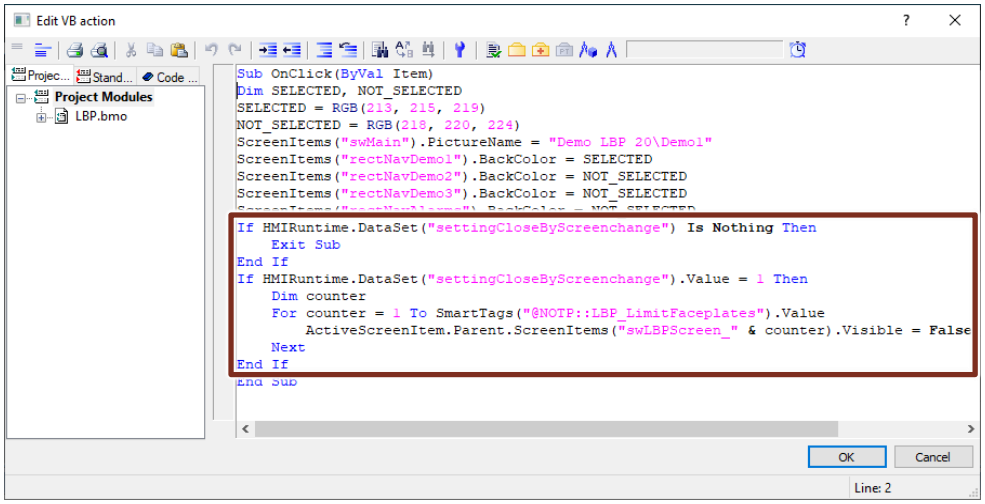
ii. Copy all the empty faceplates and the Config faceplate from the background screen of the demo project into your background screen.

iii. Your specific screens with the LBP symbols can be displayed within a faceplate on this background screen. Create this faceplate.

iv. To switch between the screens, create a navigation (integration of the symbols is done later).

v. For the navigation buttons, the following script, marked in red, must be executed:

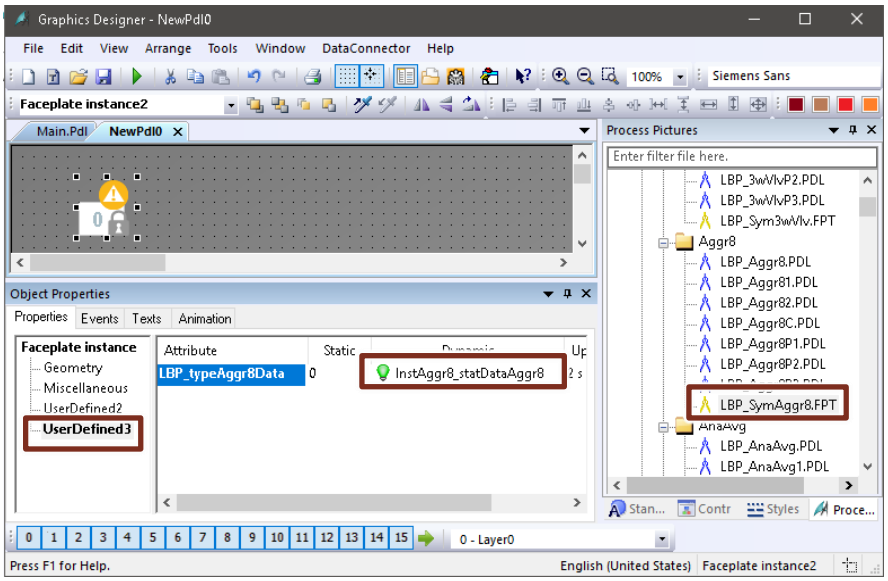
Figure 3-15



vi. Create a button to open the general settings. Switch the Config faceplate to visible with this button.

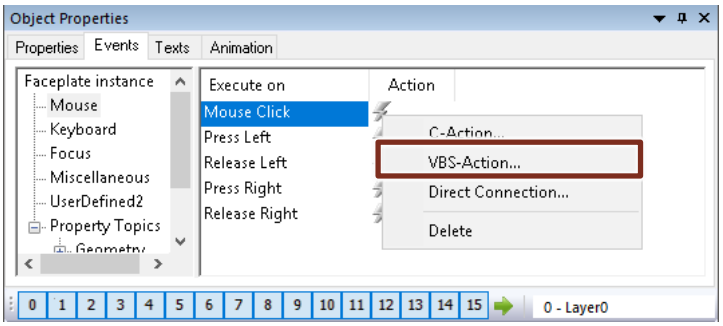
7. Integrate and connect the symbols.
   a. Open or create a screen in WinCC Explorer which is opened via the navigation in the background screen and is to show symbols. If you work with the background screen of the demo project, you can use the screens “Demo1” or “Demo2” for this purpose.
   b. The right-hand side of the “Graphic Designer” contains the symbols (“LBP_SymXxxx.FPT”) in the “Process Screens” tab.
   c. Drag the symbol into your screen.
   d. Open the “Properties” of the faceplate.
   e. Integrate the respective “statDataXxxx” structure under “Faceplate instance > UserDefined3”.

Figure 3-16



   f. Open the “Events” tab and select “Mouse” under “Faceplate instance”.
   g. Connect the “Mouse Click” event to “VBS action…”.

Figure 3-17



   h. Insert the following code line into the VBS action:
   Screenhandling “Instance name”, “Screen to be opened”
   Example:
   Screenhandling “instAggr8”, “LBP\Aggr8\LBP_Aggr8”

Note Before a new import of the AS symbols, delete the structure tags that change.

Multi-instances are only supported in WinCC V7.5 SP1 or higher.

3.4 WinCC Open Architecture

3.4.1 Scope of Delivery

The “Library for Basic Processes” (LBP) for WinCC OA comprises the following three packages:

1. **LBP_3.16**
   The folder contains all the information required for the engineering and the operation of a project. This includes the LBP symbols, their associated faceplates, as well as files through whose import the LBP data model is generated. The integration of the library, also referred to as “Subproject”, is described later on
2. **LBP_template**
   This package represents an executable template for an LBP project. It has to be registered for the operation with WinCC OA. By means of a Wizard, you can then parameterize projects from the connection(s) and generate the data points (devices) existing in the control system. You can then easily begin with the screen creation on this basis.
3. **LBP_DemoProject**
   A complete WinCC OA sample project, including the associated TIA project, can be found in this package. The PLC project can be transferred to a PLC (PLCSim/Advanced). After registration, start, and corresponding connection configuration of the WinCC OA project, the simulation can be viewed and operated.

3.4.2 Installation

At this position, an existing installation of WinCC OA 3.16 is assumed. Necessary fundamentals such as licensing, project generation or registration, and dealing with “managers” within the WinCC administrator panel or the console are required and can be found in the online help of WinCC OA.

Note During the installation of WinCC OA, the S7plus driver responsible for the communication with the control units of the CPU type 15xx and 12xx has to be selected and installed.

Registration of the library

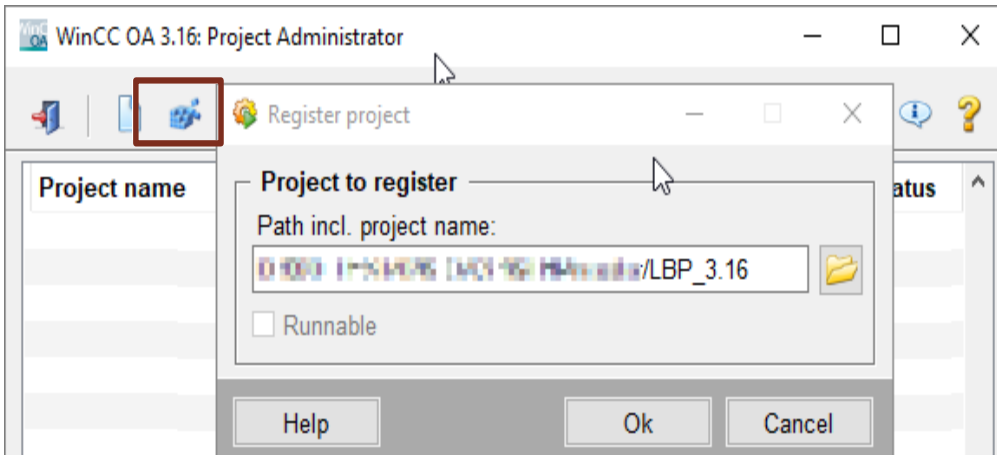
For the further steps, the “LBP_3.16” folder structure has to be located on the local hard disk of the WinCC OA computer.

Restart the WinCC OA project administration via the start menu.

There, the library “LBP_3.16”, as shown in the following figure, is first registered.

Open the associated dialog box either by clicking the selected icon in the screen or by right-clicking the project list and selecting the corresponding menu items in the shortcut menu.

Figure 3-18



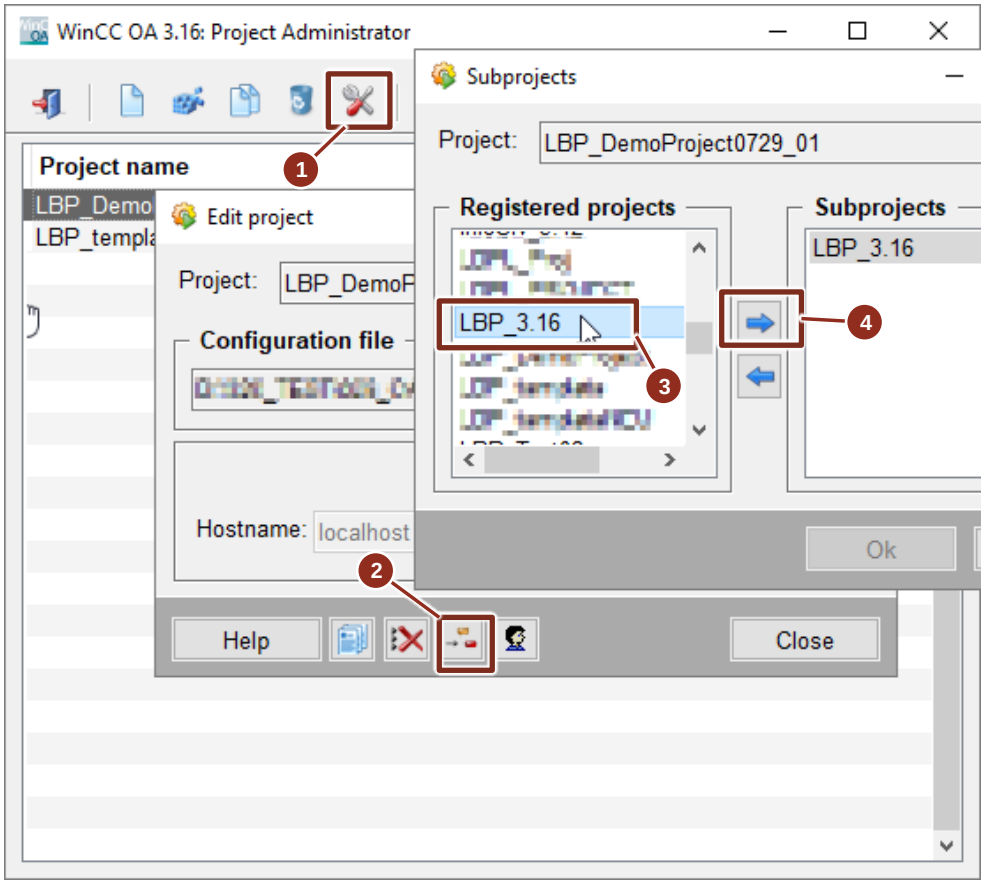
Therefore, it is now possible to include the library in your own projects or, as described below, in the “LBP_template” template project or the “LBP_Demoproject” demo project.

Registration of the template or demo project

Also register the actual executable project (template/demo).

Then select this project and use “Change project properties” (1) to select the function “Integrate subproject” (2). In the dialog that opens, select the library “LBP_3.16” (3) and move it into the list of subprojects by clicking the “Arrow to the right” (4).

Figure 3-19



The selected project can only be started via the WinCC OA administration or the WinCC OA console. All required drivers and control managers are already available in the two projects provided.

It is, of course, also possible to create a new project in accordance with your own requirements

Creating a new project

Note In order to handle own projects, profound knowledge of WinCC OA is required, which can be gained by attending the respective training courses.

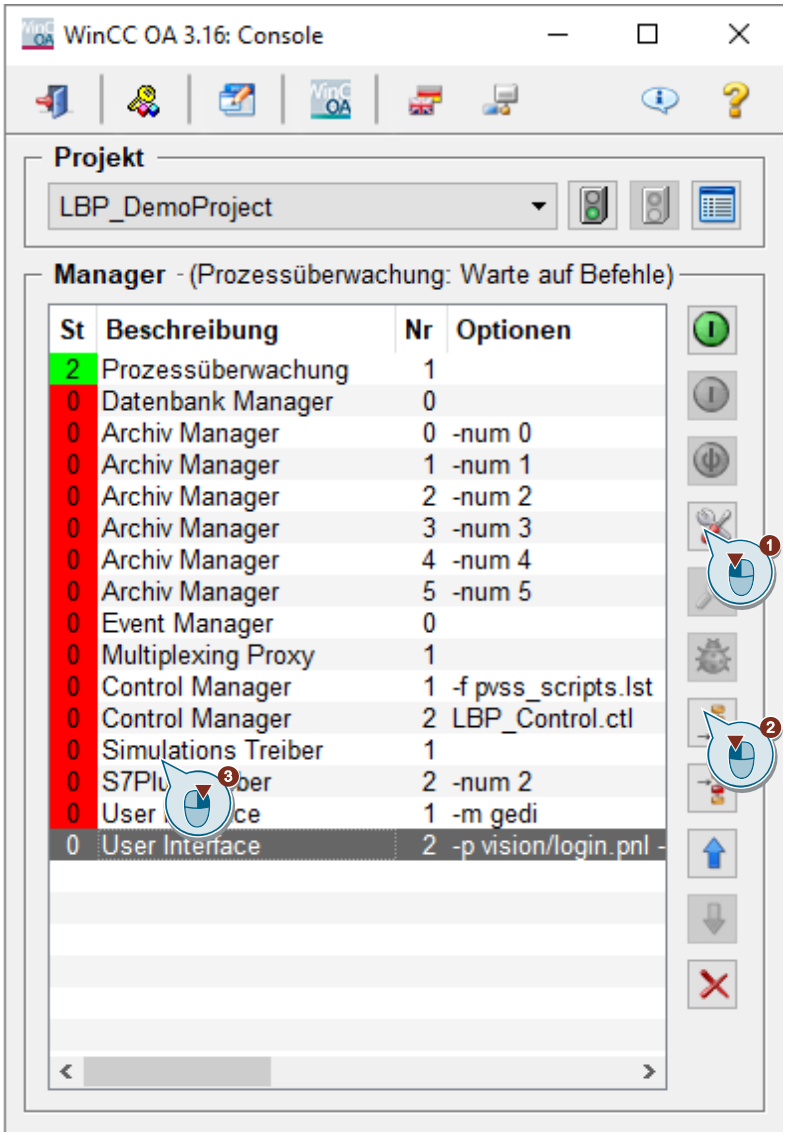
You can create a new project with the corresponding function of the WinCC OA Administration.

Note When creating a new project, the following three project languages must always be selected:

- de_AT.utf8
- en_US.utf8
- ru_RU.utf8

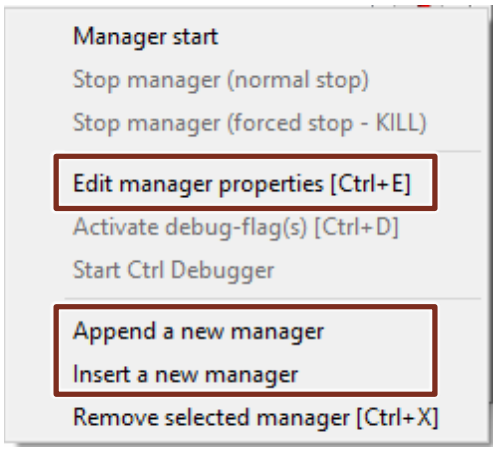
Connect the library as described above. In contrast to the template or demo project, you must add and parameterize additional managers in the WinCC OA console:

Figure 3-20 WinCC OA Console



1. Left-click this icon to allow changes to the properties of the selected manager.
2. Left-click this icon to add a new manager type.
3. Right-click the manager or the list respectively. A shortcut menu with the following functions opens:

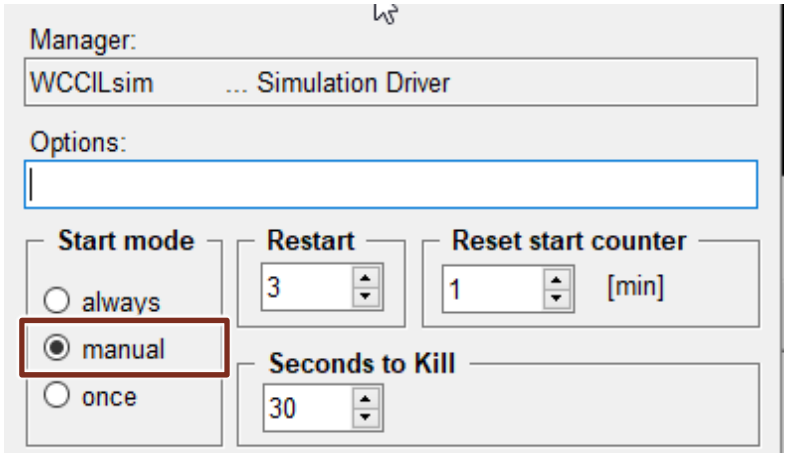
Figure 3-21



Using these functions perform the next steps:

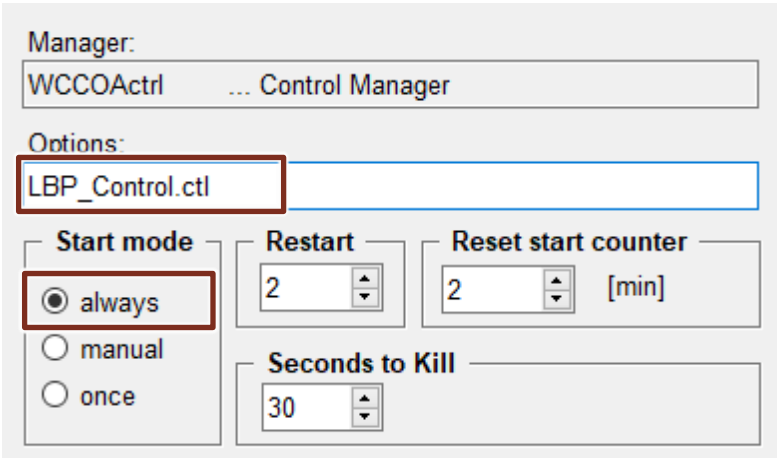
1. Set the “Start Mode” manager property of the “Simulation Driver” to “Manual”. This ensures that the manager is not started. Alternatively, the manager can also be deleted.

Figure 3-22



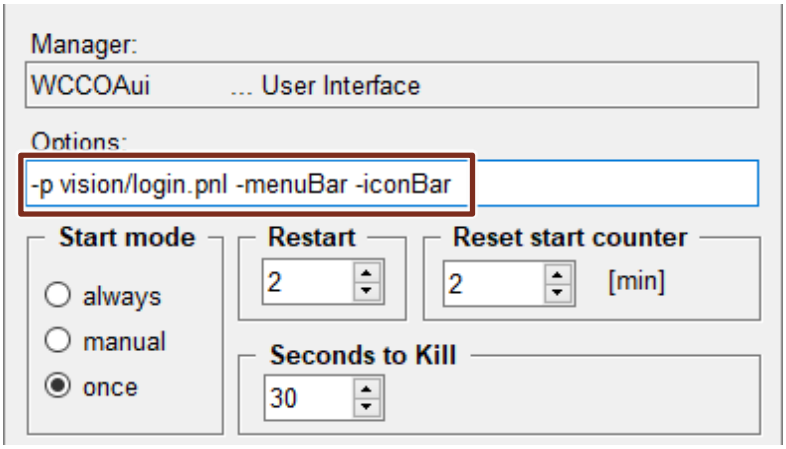
2. Add a new Control Manager with the following settings.

Figure 3-23 Control Manager properties



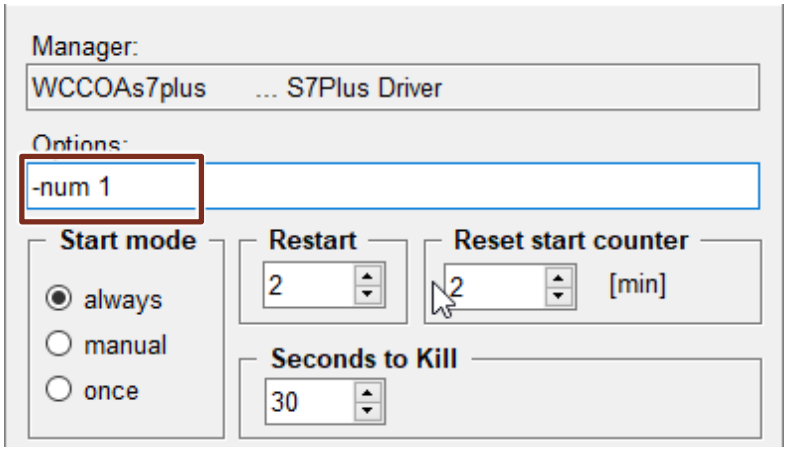
3. Add a new user interface manager with the following settings.

Figure 3-24 User interface properties



4. Add a new driver manager of the “S7Plus driver” type with the following settings:

Figure 3-25 S7+ driver properties



The manager number *(-num X*) can deviate depending on the desired configuration. However, it must be unique in each case.

3.4.3 Engineering

WinCC Open Architecture is an independent SCADA system and not an integral component of the TIA Portal. As a result, they are not immediately available for the communication with the project data required for the PLC and have to be generated separately. To this purpose, the mass parameter assignment of data is made easier through a tool that supplies all the required parameterization data from the PLC to WinCC Open Architecture. The installation path of WinCC Open Architecture contains installation files that install this tool in the TIA Portal. The next steps describe the procedure for the installation of the TIA Export Tool in the TIA Portal and the export of the parameter file.

Step 1: Creating an export file in the TIA Portal

All control data of the TIA project that the driver is to access must also be available as a TIA export file for the runtime. To do this, export the project data from the TIA Portal via the TIA Export Tool into a zip file and then copy the zip file into the WinCC Open Architecture project directory.

The export tool is stored during the installation of the S7Plus component (optional, can be selected from the WinCC Open Architecture A–installation) in the installation folder “<wincc_oa_path>/data/clsetup”.

1. In the directory, select the suitable installation file for your TIA Portal version. Install the corresponding tool (V13SP1, V14 or V15) on the system of your TIA Portal.
2. Start the TIA Portal, right-click the project or a specific PLC, and in the shortcut menu, select the “Export to SIMATIC SCADA” entry.
3. Place the exported project into the directory “<WinCC_OA_Proj>/data/TIA_Projects” on the computer on which the driver runs. In a redundant system, this also applies for the redundant driver.

This task, among others, is also taken over by the Wizard in the LBP. This is described from the next step (Figure 3-29 Import TIA file).

Figure 3-26

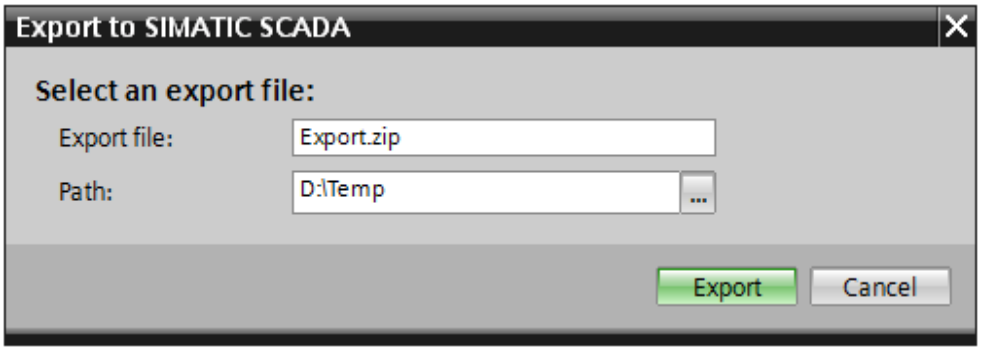
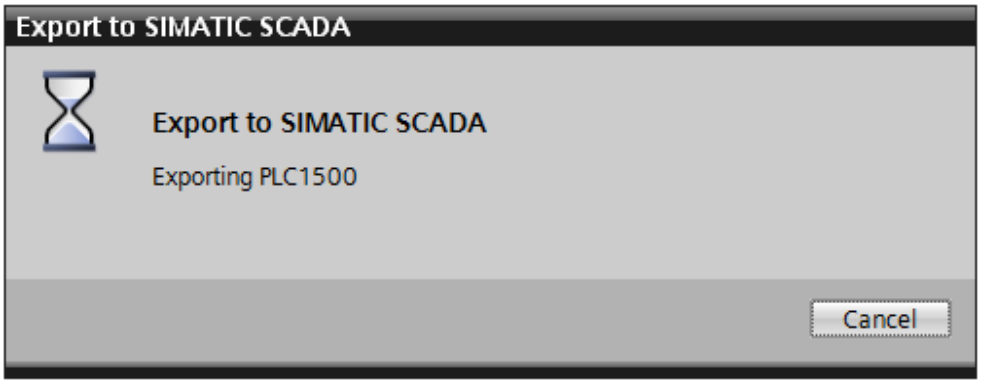


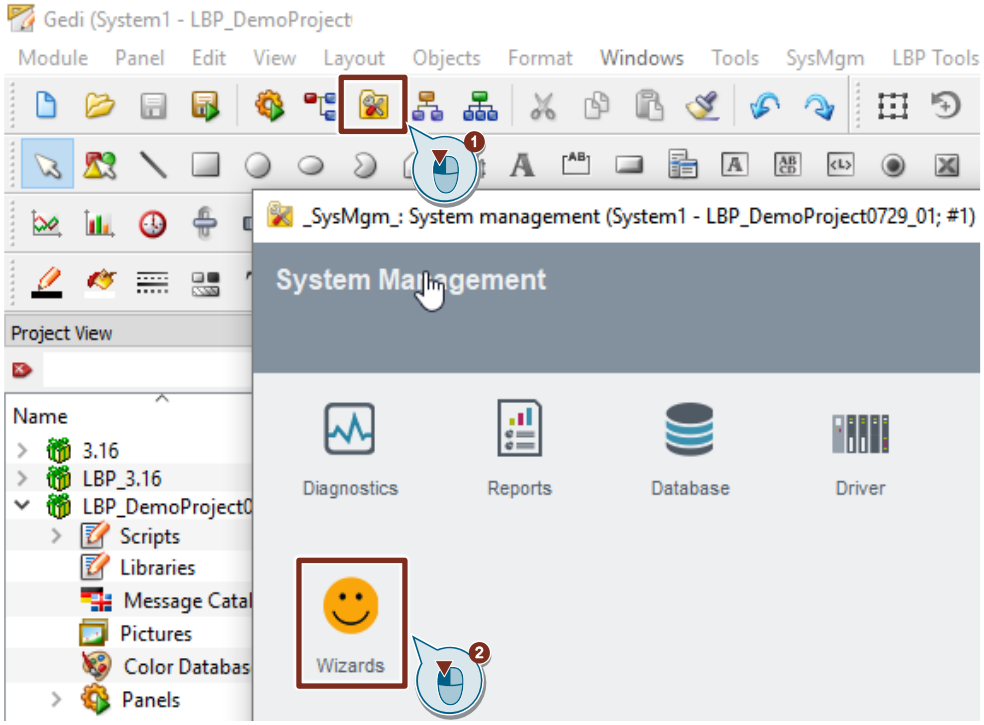
Figure 3-27



Step 2: Project start and data import

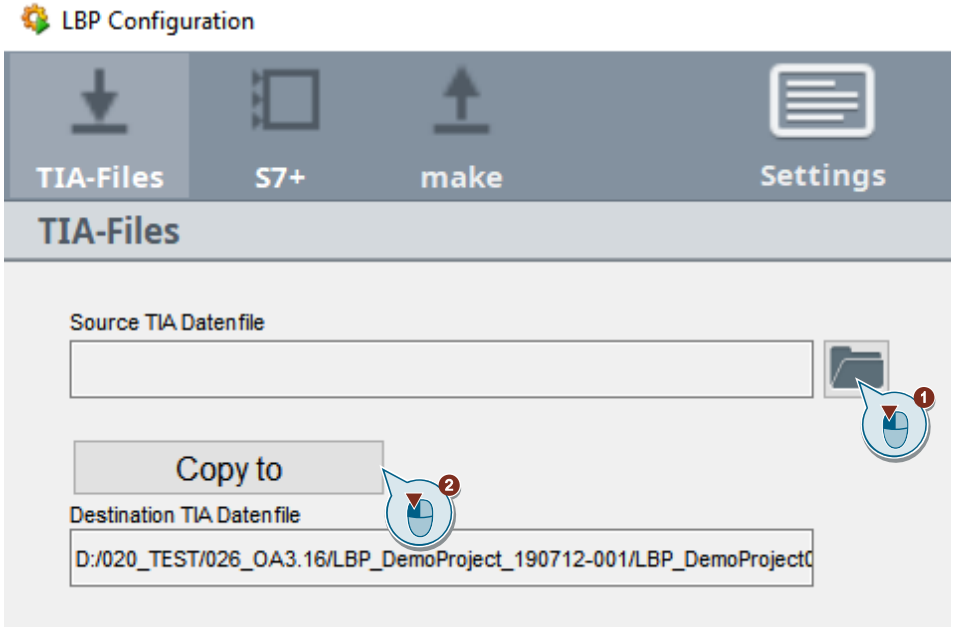
1. Start the project.
2. After the Graphical Editor of WinCC OA has been opened, start the system management there.

Figure 3-28



Clicking the Wizard opens a selection of existing Wizards. Select the “LBP configuration”.
In the opened panel, you are now offered four steps for generating your project.

Figure 3-29 Import TIA file

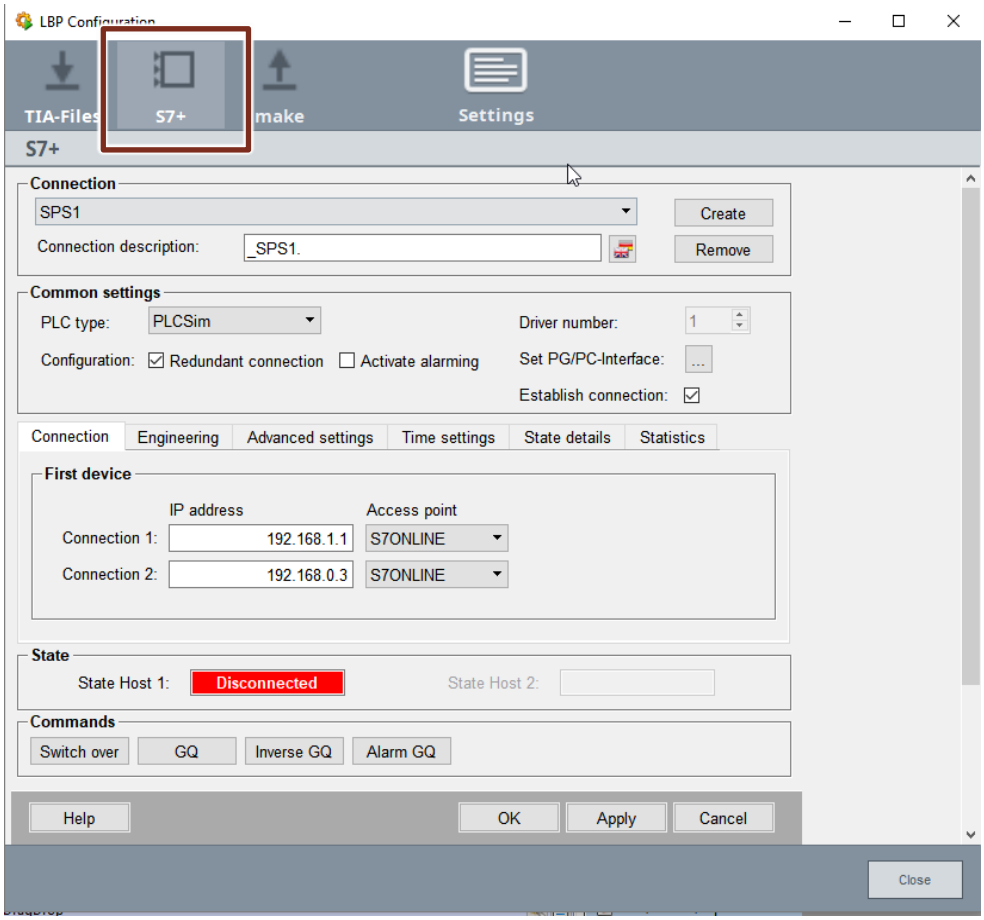


3. Select the export file of the TIA project in a file browser (1).
4. By clicking the “Copy to” (2) button, this file is automatically copied into the correct directory of the project.

Step 3: Configuring and establishing connection(s) to PLC

Select the “S7+” tab to open the following parameter screen.

Figure 3-30



Create and name a new connection and select the desired PLC type.

Specify the driver number (Figure 3-25 S7+ driver properties) and IP address of the PLC and in the “Engineering” tab select the associated PLC data.

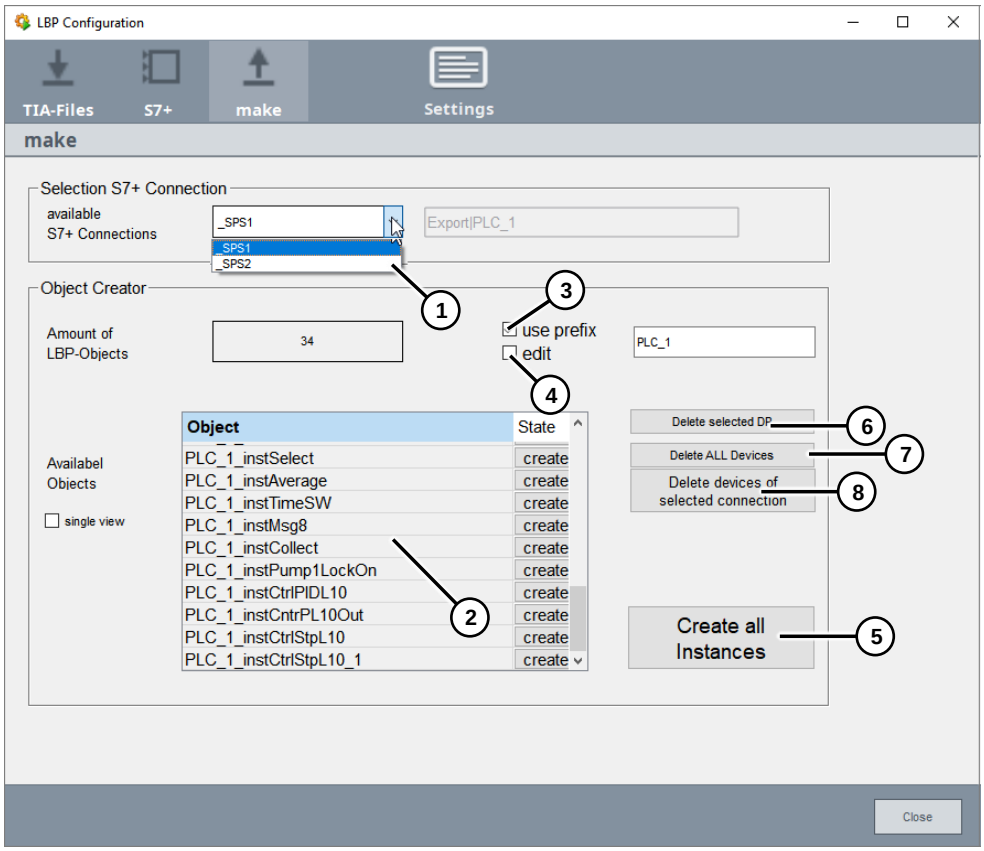
You can find detailed information on the driver parameter assignment in the section “Configuration of the S7Plus driver” of the WinCC Open Architecture online help.

Note To edit the lower part of the dialog, use the side scroll bar.

Step 4: Creating device instances automatically (mass parameter assignment)

With the “make” dialog, all required device data can be generated automatically in a short time.

Figure 3-31



1. From the drop-down list in the “Selection S7+ Connection” (1), select the correct station from the control systems or configured connections.
   After selecting a connection, the LBP devices configured in the control system that are available for the visualization are automatically displayed and listed in the table at the bottom (2).
2. To avoid naming conflicts for multiple control systems with the same content, a prefix is recommended. This is the control name as it is designated in the TIA Portal. Deactivate this option with “Use prefix” (3) to apply the original name from the control unit or leave it activated to use the proposed name as a prefix.
   Alternatively, you can specify your own prefix with the “edit” (4) option.

Note

Caution when using own prefixes: This could allow the generation of multiple devices that access the same address ranges of a control system—which inevitably results in an error message.

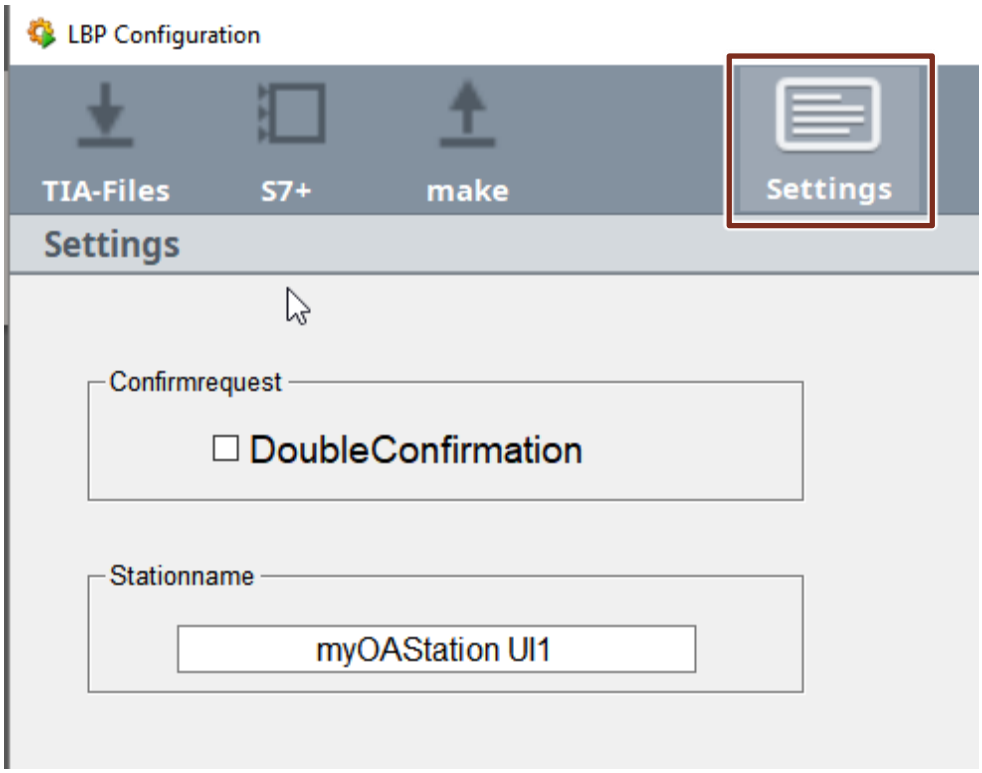
3. After specifying a prefix for the equipment name, generate a selected device via the “Create” interface in the table. Alternatively, all devices of the connection can be generated with the “Create all Instances” (5) button.

Subsequently, the data are available for selection in the Graphical Editor (GEDI).

Here, you can also delete either selected devices (6), all devices (7) or the devices of the currently selected connection (8) (take prefix into account!).

Step 5: Project settings

Figure 3-32



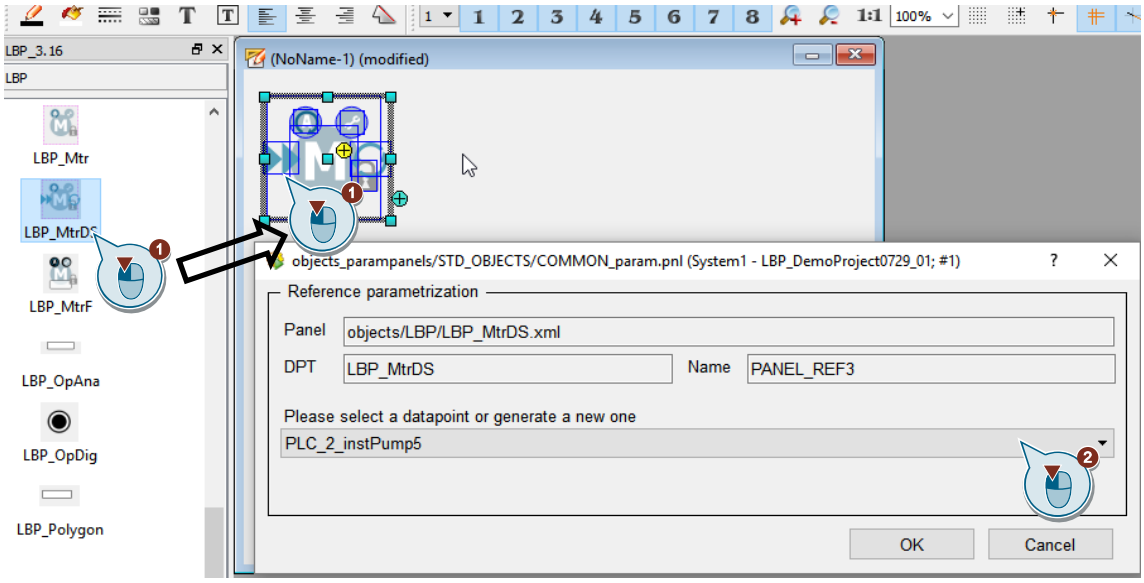
4. Set a check mark at “DoubleConfirmation” to specify whether the operating actions have to be confirmed in the running project via an additional request. Otherwise, the input may be carried out directly.
5. With “Station name”, you can specify a station name that is required for the interlocking of operating blockings. This name is displayed for other stations if a blockade of this station is activated.

3.4.4 Creating Screens

The screen creation takes place via the Graphical Editor (GEDI) of WinCC OA and is described in the online help.

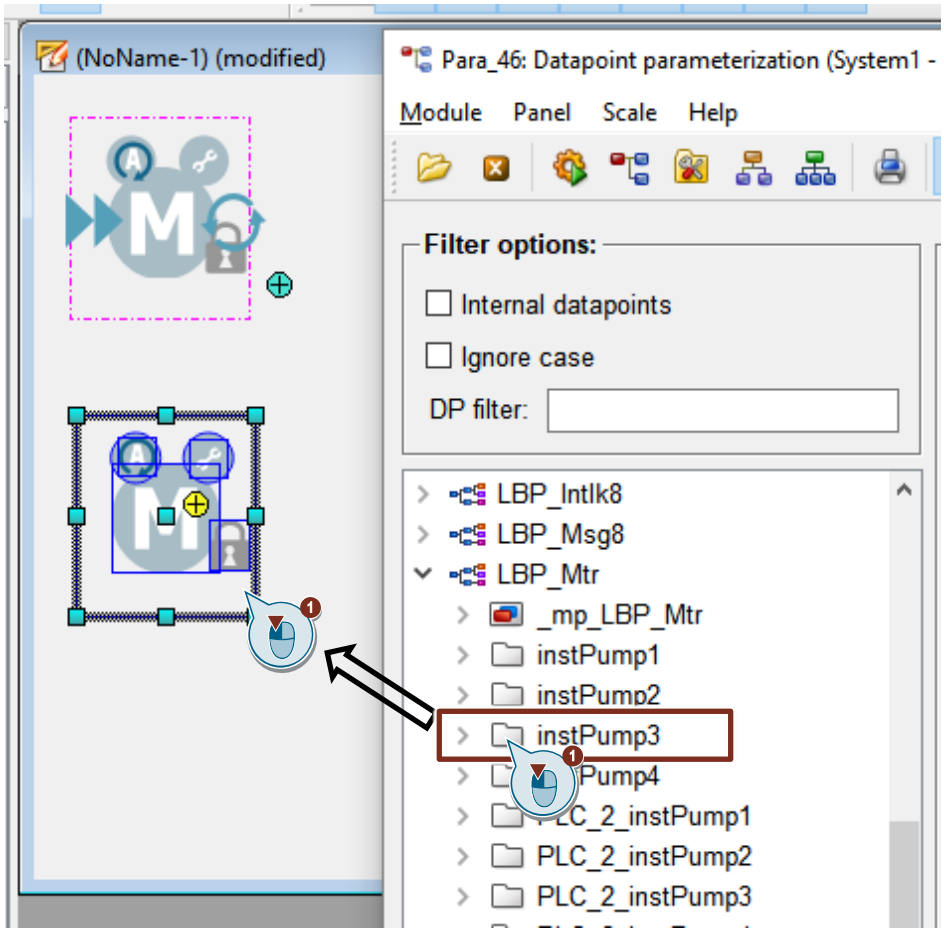
The LBP objects are available in the “LBP” catalog and can be moved using drag & drop (1) from there into the panel. After the placement, a dialog opens automatically in which a suitable data point is assigned to the graphical object.

Figure 3-33



A further possibility to drag objects into the graphical surface using drag & drop is the Database Parameter Assignment Tool (PARA).

Figure 3-34



4 Changes in V2.0

Since the interface between PLC and HMI has been completely revised, nothing can be imported from V1.0 from the HMI point-of view. All symbols and screens have to be implemented again.

4.1 STEP 7 Blocks

The following section describes which interfaces have changed.

There are no additional HMI blocks anymore with V2.0. For this reason, the “dataHMI” input is not required anymore for some blocks. However, tags have been added that were previously treated in the HMI block.

The “identName” has additionally been added everywhere.

CntrP, CntrA, CntrD

Table 4-1

Deleted inputs	Added inputs	Deleted outputs	Added outputs
dataHMI	identName	-	correction
-	-	-	lastCounter
-	-	-	lastReset
-	-	-	enInp

SetCrv / Select

Table 4-2

Deleted inputs	Added inputs	Deleted outputs	Added outputs
dataHMI	identName	-	-

Msg8 / OpAna / OpDig

Table 4-3

Deleted inputs	Added inputs	Deleted outputs	Added outputs
-	identName	intError	-

TimeSw

Table 4-4

Deleted inputs	Added inputs	Deleted outputs	Added outputs
-	identName	intError	-

TimeSw

Table 4-5

Deleted inputs	Added inputs	Deleted outputs	Added outputs
dataHMI	identName	-	-
typeCycle	timeSwData (this is an UDT which contains all the deleted entries with the exception of dataHMI.)	-	-
timeOn	-	-	-

Deleted inputs	Added inputs	Deleted outputs	Added outputs
duration	-	-	-
indexHMI	-	-	-

Mtr, MtrDS, MtrF, Vlv, VlvA, 3wVlv

Table 4-6

Changed inputs	Replace at	Reason
indTrip	indTripOk	The signals are “low active” “0”: Error Trip active “1”: Error Trip not active

5 Creating Own Blocks

5.1 Creating the PLC program

Note You can find information on the creation of the STEP 7 program code at: https://support.industry.siemens.com/cs/ww/en/view/81318674.

5.1.1 Tag Management of LBP Blocks

Inputs

identName (String[30]) (see 2.1.2 Cross-Block Input “identName”)

Static tags

statData*Block name*
This UDT has to be created once for each block. It represents the interface to the HMI system and contains the following structures:

- **settingsHMI**
  Contains all the tags that can be described from the HMI system.
  The following tags are included in every or nearly every block:
  - note (see 2.1.3 User-Defined Data Types for Communication with the Visualization System)
  - opStation (see 2.1.3 User-Defined Data Types for Communication with the Visualization System)
  - overwrite (see 2.1.3 User-Defined Data Types for Communication with the Visualization System)
  - alarmsAck (see 2.1.3 User-Defined Data Types for Communication with the Visualization System)
  - alarmsInfo (see 2.1.3 User-Defined Data Types for Communication with the Visualization System)
- **settingsPLC**
  Contains all the inputs of the STEP 7 block that is to be visualized on the HMI system (raw data).
- **statusHMI**

  Contains the values that the STEP 7 block outputs after the editing.

  This structure must contain the “identName” tag (see 2.1.2 Cross-Block Input “identName”).

  If the status is visualized on the HMI page, a UDT can be created from Boolean tags with the “msgStatus” name (for an example, see 2.13.2 Interface Description for HMI Communication).
- **log15** (optional)
  This UDT is required if the “LBP_Log15” function block is used. This is used for saving the status or error codes in a data buffer with 15 entries.
  Application:
  To create a new entry, write the new code on the input. Since the text is already assigned by a text list from the HMI point-of-view, it is important that each code only occurs once in the complete PLC program. The currently occupied codes are configured in the HMI system in the “LBP_Log” text list under “Text and graphics lists” (all values are occupied at least until 55).
  To read out the entry, copy the “statData” UDT from the “LBP_Log15” instance

into the “log15” UDT
(Example: `#statData3wVlv.statLog15 := #instFBLog15.statData`)
See also: 2.28 LBP_Log15 – Logging Data.

Outputs

“intError” (Word) for output of error codes.

The usage makes sense if STEP 7 functions that output error codes are used in the block.

5.2 Creating the Visualization in WinCC Professional

Copy the folder with screens of the LBP block that resembles your block the most. Then adapt the folder names and screen names to the name of your block.

5.2.1 Naming Elements

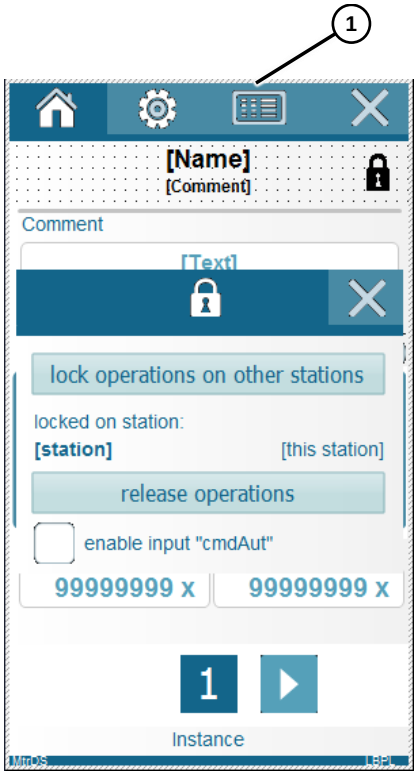
Table 5-1

Object	Labeling guideline
Input box	ioXxx
Frame/rectangle	rectXxx
Button	btnXxx
Texts	txtXxx
Groups	grpXxx
Circles	cirXxx
Lines	lineXxx
Figure	picXxx

5.2.2 Adjusting the Background Screens (LBP_Xyz)

Your header functions with one of the following adjustments:

Figure 5-1



1. If your block does not have any own messages, the tab for messages can be removed in the navigation. To do this, cancel the group of the tab and delete the “Report” graphic.
   If your block has messages, select the button, open the properties, and modify the “Click” event under “Events”. Here, “_statDataXyz” has to be replaced by the name of your structure.
2. Select the lock, open the properties, and then click “Property list”. A script was attached under “General > Process value”. Adapt the SmartTag for “activeStation” to the structure name of your block.
3. Click next to the screen within the gray area, right-click, and select “Change object references”. Then, replace the structure name of the tags with the structure name of your block.
4. Switch all layers of the layout to visible.
   You can find the “Layout” tab at the right-hand edge in TIA Portal. To switch the layers to visible, click the grayed-out eye next to the name of the layer.
5. The “Content” level contains all the faceplates that show the content of the individual tabs. Select each faceplate individually and adapt the displayed screen to the screen of your block in its properties under “General”.
   Also adapt the tag prefix to the name of the structure of your block.
6. Adapt the skirting board to your block name.

5.2.3 Adjusting the Home Screen (LBP_Xyz1)

Retain the line at the upper edge of the screens. You can adapt the rest at will. You can obtain ideas from other LBP screens.

5.2.4 Adjusting the Setting Pages (LBP_XyzP1)

The entry fields for Unit and Format can continue to be used without adjusting if you have named the associated tags in your UDT by the same name as the other LBP blocks.

A script is stored in the entry field for the comment that needs to be modified (see 5.2.7 Script for Switching Operability and Colors).

Note If you want to display the log screen (“General > LBP_Log”), it is important that the last setting page is named “LBP_XyzPLast” and that you have supplemented the “LBP_Log” text list with your error codes.

Adjusting the internal navigation in the setting pages

Figure 5-2

1. Select the internal navigation and open its properties.
2. Select the “Events” tab.
3. Open the “Click” event of the “picSwitchPage1” element.
4. Adjust the “SetPropertyCurrentWindow” function to fit your screen names.
5. Repeat steps 3 and 4 for the “picSwitchPage2” element.

5.2.5 Adjustments when Using the Double Confirmation (LBP_XyzC)

For buttons for which the Double Confirmation is relevant, a script is stored with the “Click” event. Example:

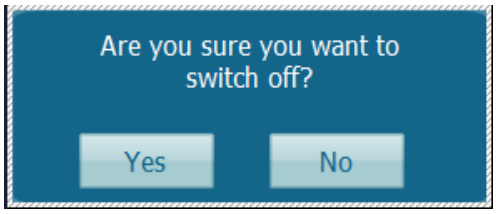
Figure 5-3

```
Sub OnClick(ByVal item)
Const offset = 3
Dim CmdBit
CmdBit = offset - SmartTags(""&".statusHMI.msgStatus.auto") 'cmd-Bit 3 = switch to auto, cmd-Bit 2 = Switch to Manual

If SmartTags("@NOTP::LBPL_DoubleConfiguration") Then
    If Parent.Parent.DataSet("txtDoubleConfirmation") Is Nothing Then 'Variable for selecting the text for double confirmation
        Parent.Parent.DataSet.Add "txtDoubleConfirmation",CmdBit
    End If
    Parent.Parent.DataSet("txtDoubleConfirmation").Value = CmdBit
    Parent.Parent.ScreenItems("swDoubleConfirmation").Visible = True
Else
    InvertBitInTag ""&".settingsHMI.cmd", CmdBit
End If
End Sub
```

The “LBP_DoubleConfirmation” tag determines whether the Double Confirmation is active or not. Subsequently, a check is carried out to determine whether the dataset tag is “txtDoubleConfirmation” already exists. If not, it is created. This tag is passed to the number of the text list entry to be displayed as a query. The number of the text list entry also determines the number of bits of the “cmd” tag that is to be set if the question is answered with “Yes”.

Figure 5-4



Create a text list with the questions that you require. If only different bits of the same tag are to be set depending on the question, the numbers of the text list entries should correspond with the bit numbers.

If you have to write different tags, you have the possibility of placing several buttons on top of each other and showing and hiding them depending on the text list entry.

5.2.6 Creation of Messages

Observe the following points for the creation of messages:

- Assign the message class “LBP_Status”, “LBP_Warning” or “LBP_Alarm” to your messages. Warnings and alarms need to be acknowledged as a result.
- Select the “alarmsAck” tag from your “settingsHmi” UDT as an acknowledgment tag. You can select the acknowledgment bit freely.
- Select the “alarmsInfo” tag from your “settingsHmi” UDT as the status tag. You can select the status bit freely.
- In order for the message filter to work, the “identName” tag from the “statusHMI” UDT must be selected as Parameter1.

5.2.7 Script for Switching Operability and Colors

You can find the script for switching the operability and the colors behind the first interface on your screen or behind the first element.

It usually starts with a query to determine whether the operation is currently locked or not.

Table 5-2

Querying the operation release

`Dim activeStation, thisStation, statelocked, stateLockedBool`
`activeStation = SmartTags("&".settingsHMI.opStation")`
`thisStation = SmartTags("@NOTP::LBP_StationName")`
`If (activeStation <> thisStation) And Not(activeStation = ") Then`
`  statelocked = 0`
`  stateLockedBool = False`
`Else`
`  statelocked = 1`
`  stateLockedBool = True`
`End If`
`ScreenItems("ioHmiTime").Enabled = stateLockedBool`
`ScreenItems("ioHmiMode").Enabled = stateLockedBool`

The “stateLockedBool” tag is “true” when operation is possible. The “Enabled” property must have the value of this tag at each button and each entry field.

Then, an array is defined in which the colors are determined depending on the switch position, the operability, and whether it is an HMI entry field or a PLC entry field. This array is only required, as a whole, if there are typical settings that are described in the document for controlling the LBP in Chapter “2.1.1.2 Operating the Settings Screens” under “Overwriting Values”.

Table 5-3

Color array

```
Dim colorArray(1,1,5)
Dim colorSwitch(1)
Const TXT_PLC = 1, TXT_HMI_BACK_PLC = 2, BACK_HMI = 3, BORDER_HMI
= 4, BORDER_PLC = 5
Const SWITCH_UP = 0, SWITCH_DOWN = 1
Const UNLOCKED = 1, LOCKED = 0
Dim LIGHT_GRAY, BLUE_GRAY, WHITE, BLUE, FRAME_GRAY
LIGHT_GRAY = RGB(223,223,223)
FRAME_GRAY = RGB(202,202,202)
BLUE_GRAY = RGB(167,188,197)
WHITE = RGB(255,255,255)
BLUE =  RGB(85,160,185)

colorSwitch(UNLOCKED) = BLUE
colorSwitch(LOCKED) = BLUE_GRAY
colorArray(SWITCH_UP,UNLOCKED,TXT_PLC) = LIGHT_GRAY
colorArray(SWITCH_UP,UNLOCKED,TXT_HMI_BACK_PLC) = BLUE
colorArray(SWITCH_UP,UNLOCKED,BACK_HMI) = WHITE
colorArray(SWITCH_UP,UNLOCKED,BORDER_HMI) = FRAME_GRAY
colorArray(SWITCH_UP,UNLOCKED,BORDER_PLC) = BLUE
colorArray(SWITCH_UP,LOCKED,TXT_PLC) = LIGHT_GRAY
colorArray(SWITCH_UP,LOCKED,TXT_HMI_BACK_PLC) = BLUE_GRAY
colorArray(SWITCH_UP,LOCKED,BACK_HMI) = WHITE
colorArray(SWITCH_UP,LOCKED,BORDER_HMI) = FRAME_GRAY
colorArray(SWITCH_UP,LOCKED,BORDER_PLC) = BLUE_GRAY
colorArray(SWITCH_DOWN,UNLOCKED,TXT_PLC) = BLUE_GRAY
colorArray(SWITCH_DOWN,UNLOCKED,TXT_HMI_BACK_PLC) = WHITE
colorArray(SWITCH_DOWN,UNLOCKED,BACK_HMI) = BLUE
colorArray(SWITCH_DOWN,UNLOCKED,BORDER_HMI) = BLUE
colorArray(SWITCH_DOWN,UNLOCKED,BORDER_PLC) = FRAME_GRAY
colorArray(SWITCH_DOWN,LOCKED,TXT_PLC) = BLUE_GRAY
colorArray(SWITCH_DOWN,LOCKED,TXT_HMI_BACK_PLC) = WHITE
colorArray(SWITCH_DOWN,LOCKED,BACK_HMI) = BLUE_GRAY
colorArray(SWITCH_DOWN,LOCKED,BORDER_HMI) = BLUE_GRAY
colorArray(SWITCH_DOWN,LOCKED,BORDER_PLC) = FRAME_GRAY
```

Subsequently, the colors are assigned to the screen elements. To begin, the “overwrite” bit, that determines which position the switch has, is evaluated. This takes the following form:

Table 5-4

Color assignment

```
Dim overwrite
overwrite = SmartTags("&".settingsHMI.overwrite").Value
Dim overBool, overInt

'Mode----------------------------------------
overBool = CBool(overwrite And 2^0)' if overwrite bit 0 --> true
overInt = (CInt(overBool))*(-1)

ScreenItems("picOnMode").Visible = overBool
```

Color assignment
`ScreenItems("picOffMode").Visible = Not overBool`
`ScreenItems("picOnMode").ProcessValue = statelocked`
`ScreenItems("picOffMode").ProcessValue = statelocked`
`ScreenItems("rectSliderMode").BackColor = colorSwitch(statelocked)`
`ScreenItems("rectSliderMode").BorderColor = colorSwitch(statelocked)`
`ScreenItems("ioPlcMode").ForeColor = colorArray(overInt, statelocked, TXT_PLC)`
`ScreenItems("ioHmiMode").ForeColor = colorArray(overInt, statelocked, TXT_HMI_BACK_PLC)`
`ScreenItems("ioHmiMode").BackColor = colorArray(overInt, statelocked, BACK_HMI)`
`ScreenItems("ioPlcMode").BackColor = colorArray(overInt, statelocked, TXT_HMI_BACK_PLC)`
`ScreenItems("ioHmiMode").BorderColor = colorArray(overInt, statelocked, BORDER_HMI)`
`ScreenItems("ioPlcMode").BorderColor = colorArray(overInt, statelocked, BORDER_PLC)`

5.2.8 Evaluation of the Status Tag (.settingsHMI.alarmsInfo)

The first half of the bits of the status tag indicates whether a message is active. The second half of the bits indicates whether the message has been acknowledged.

At the LPB, the warning triangles pulsate at the symbols if messages of the block are not yet acknowledged. The frames of fields flash when the error that they are showing has not been acknowledged yet.

A field that is to show the acknowledgement status is superimposed with a graphics list that represents the frame. This graphics list has the name “ioFrame_x”.

At the first frame on the screen, a script is stored which evaluates the acknowledgement status of all the messages.

To this purpose, Bits 16 to 32 of the “.settingsHMI.alarmsInfo” tag are read out. If the status bit “0” is specified in the HMI message, Bit 16 must be evaluated to read out the acknowledgement status.

5.3 Creation of the Visualization in WinCC Comfort

5.3.1 General Information

The easiest way to create your own screens for a function in WinCC Comfort is to copy the screens of an LBP function that is similar to your own function. In this case, the correct templates, faceplates, and parts of the LBP screens can be used directly.

Note Each LBP screen for WinCC Comfort contains a grey rectangle in the “GrayOut” layer (layer 20). This rectangle is used to make it grayed out whenever a pop-up image is displayed. This is implemented via the internal tag “PopUpBackground”. If this is set, the visibility of the rectangle is reset to “true”.

5.3.2 Naming Elements

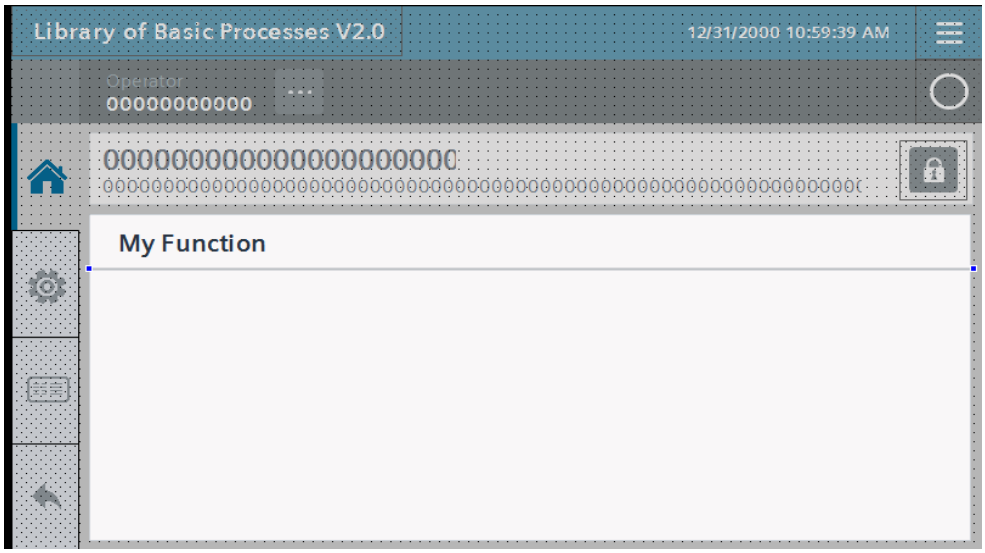
Table 5-5

Object	Labeling guideline
Input box	ioXxx
Frame/rectangle	rectXxx
Button	btnXxx
Texts	txtXxx
Groups	grpXxx
Circles	cirXxx
Lines	lineXxx
Figure	picXxx

5.3.3 Adjustments to the Home Screen (LBP_Xxx_StartScreen)

1. Copy the Start Screen of the LBP function that best fits the new function
2. Rename this.
3. In the screen, you can find the navigation “(faceplate “LBP_NavigationStartScreen”). Adjust the event of the “btnSettings” button for this faceplate. Change the screen name to the name of your (future) settings screen.

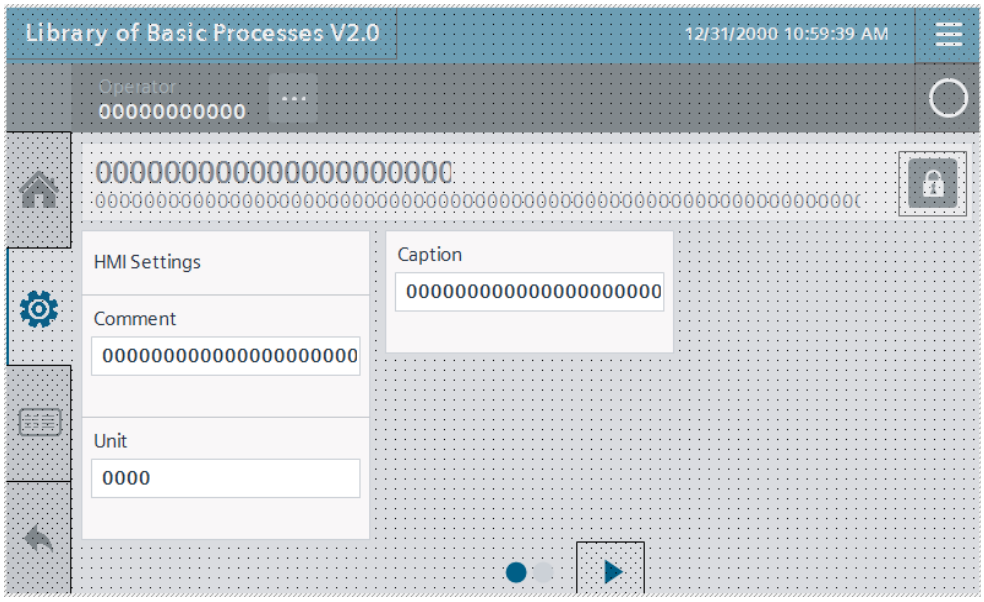
4. Arrange the rest of the screen.
   Screen 5-5



5.3.4 Adjusting the Settings Page (LBP_Xxxx_Settings1)

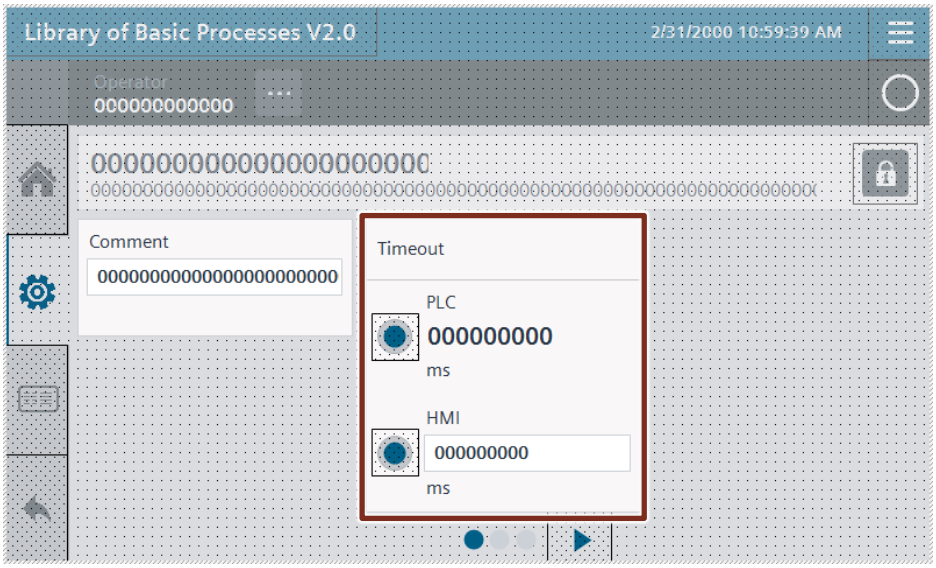
1. Copy the settings screen of the LBP function that best fits your new function.
2. Rename this.
3. Change the navigation
   a. In the screen, click the navigation faceplate “LBP_NavigateSettings”
   b. Select the “Events” tab and select “btnHome”
   c. Change the image name to the name of your Start Screen.
4. Change the contents of the settings screen. The content can be configured with the help of the prepared faceplates, which you can find in the library in the folder “Types > HMI > WinCCComfort”. These can be used, for example, for the comment and unit.

Figure 5-6



5. In the LBP, you have the option to switch between an HMI and PLC value for some parameters. For this type of switching option, proceed as follows:
   a. Copy all required elements from a different settings screen. Look for a faceplate where this value already contains the correct data type.
   b. Customize the interface of the faceplate. (The tag for “9_Locked” remains the same.)
   c. Select the radio buttons and change the animations
   d. Select the radio buttons and change the events

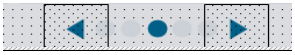
Figure 5-7



6. Circles, polygons, and invisible buttons are used to navigate between the settings screens. If you require this navigation, proceed as follows:
   a. Copy the navigation from another LBP screen. (If possible, from a function that contains the same number of settings pages)
   b. Select one of the invisible buttons and click “Events”
   c. Change the “screen name” parameter of the “activate screen” function to the new screen name.

d. Repeat the last 2 steps for the additional button.

Figure 5-8

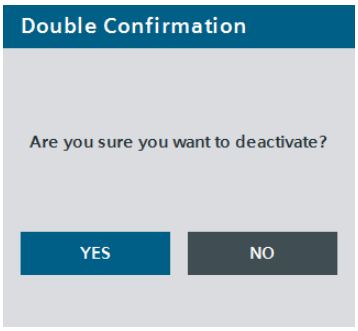


7. If you use the “Log” function, proceed as follows:
   a. Copy a “Log” image from a different LBP function.
   b. Adjust the navigation on the left and on the bottom side as described in points 3 and 6
   c. Select the “btnCheckItem” button
   d. Click “Events”
   e. Modify the tag to the “overwrite” tag of your function.

5.3.5 Adjustments when Using the Double Confirmation.

1. Copy one or both of the following scripts into your project:
   - “LBP_DoubleConf” for writing a value via the Double Confirmation Interfaces:
     - “cmd”: The value that is set in the event of a confirmation must be stored here. It is also the value which selects the text (the query) via a text list.
     - “cmd_Adr”: The tag to which the value is to be transferred in the event of a confirmation must be linked here.
   - “LBP_DoubleConfBitInTag” for setting a bit using the Double Confirmation
     - “cmd”: The binary code bit must be placed here. (e.g. Bit 0 → cmd is 1, Bit 1→ cmd is 2, Bit 2→ cmd is 4, Bit 3 → cmd is 8). It is also the value which selects the text (the query) via a text list.
     - “cmd_Adr”: The tag in which the bit is to be set must be bound here.
     - “objectVis”: Here, you can specify an object name of an object that should be visible so that the Double Confirmation is carried out. A blank name can also be specified (“”)
2. Call the script for each button that needs a Double Confirmation.
3. Copy the following elements from another LBP Start Screen (e.g. Mtr):
   - “rct_backgr”
   - “Feld_sure”
   - “txt_doubleConf”
   - “txt_doubleConf”
   - “butYes”

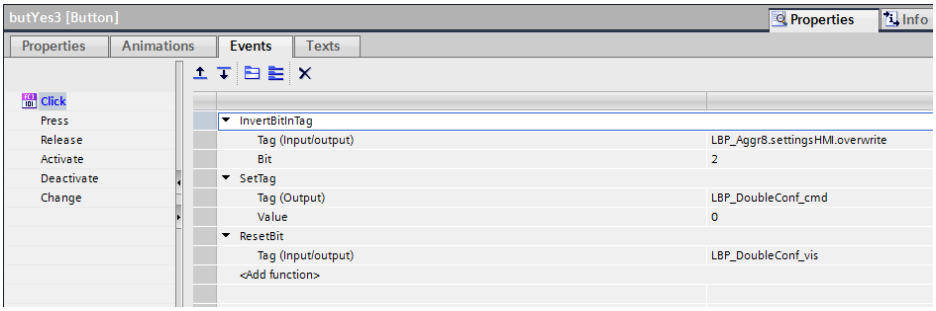
Figure 5-9



4. Create a text list with the queries for the Double Confirmation. The text is selected by the value which is passed to the script via the “cmd” interface.
5. Create a “YES” button for each button with a Double Confirmation.
   Each “YES” button has 3 events linked to it:
   - SetTag or InvertBitInTag to set the value or bit
   - SetTag to reset the “LBP_DoubleConf_cmd” tag (this tag was set by the respective script)
   - ResetBit for resetting the “LBP_DoubleConf” tag. (This tag is used to change the visibility of all Double Confirmation elements)

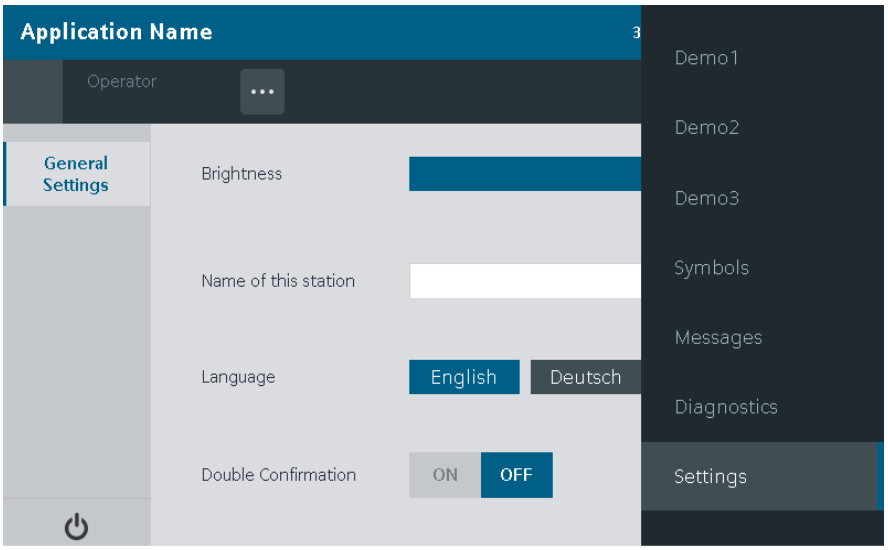
   Adjust the tag and value under “Animations > Visibility”.

   Figure 5-10



6. Activate Double Confirmation in Runtime
   a. Click the navigation symbol in the upper right corner
   b. Click “Settings” in the opened navigation
   c. Activate the Double Confirmation

Figure 5-11



5.3.6 Creation of Messages

See 5.2.6 Creation of Messages

NOTE Use tags from the UDT symbol as triggers for the messages. There is no status tag in Comfort.

6 Appendix

6.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:
https://support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:
www.siemens.com/industry/supportrequest

SITRAIN – Training for Industry

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:
www.siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:
Fehler! Linkreferenz ungültig.

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android and Windows Phone:
Fehler! Linkreferenz ungültig.

6.2 Links and Literature

Table 6-1

No.	Subject
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the article page of the application example https://support.industry.siemens.com/cs/ww/en/view/109749508
\3\	SIMATIC Visualization Architect Manual https://support.industry.siemens.com/cs/document/109755214
\4\	How do you use multiple monitors simultaneously with WinCC (TIA Portal) Runtime Professional? https://support.industry.siemens.com/cs/document/109744837
\5\	Programming Guide and Programming Style Guide for SIMATIC S7-1200 and S7-1500 https://support.industry.siemens.com/cs/document/81318674

6.3 Change documentation

Table 6-2

Version	Date	Change
V2.0	08/2019	Release for: • SIMATIC STEP 7 Basic/Professional V15 • WinCC Comfort/Advanced V15 • Comfort Panel • WinCC Runtime Advanced V15 • WinCC Runtime Professional V15 • WinCC V7.5 WinCC Open Architecture 3.16
V2.2	11/2019	Update for TIA Portal V15.1 Change of the symbol color of the valves from white-blue to blue-green
V2.4	03/2020	Update for TIA Portal V16 Additional functions: - SimoDir - SimoRev - Sina For MtrF was added an Power On/Off button