



WAGO-I/O-SYSTEM 750 CODESYS 2 Library **WagoBuilding_01.lib** Building Automation Functions

Version: July 15, 2019



© 2019 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

Email: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55

Email: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

Table of Contents	3
Information about This Documentation	5
Copyright	5
Symbols.....	5
Number Notation	5
Font Conventions	6
Important Notes	7
Subject to Change	7
Personnel Qualification.....	7
Intended Use	7
Technical Condition of Specified Devices	7
Function block Overview	8
Lighting	11
Latched relay (FbLatchedRelay).....	11
Advanced Latched Relay (FbAdvancedLatchedRelay).....	12
Dimmer Double Button (FbDimmerDoubleButton).....	14
Dimmer Single Button (FbDimmerSingleButton).....	18
Automatic Light (FbMacroAutomaticLights).....	22
Constant Light Control (FbMacroConstantLightControl)	24
Daylight dependent lighting (FbMacroDaylightDependentLighting)	29
Light Control (FbMacroLightControl)	33
Stairwell Light Control (FbMacroStairwellLightControl).....	35
Twilight Control (FbMacroTwilightControl)	37
Sunshade	39
Sunshade actuator (FbSunshadeActuator)	39
Safety Commands	44
Frost protection alarm (FbFrostAlarm)	44
Wind Alarm (FbWindAlarm)	46
Weather protection (FbBasicWeatherProtection)	48
Weather Protection With Weather Assessment (FbAdvanceWeatherProtection)	50
Automatic Thermal Control	53
Sunshade as heating support (FbSunshadeHeatingSupport)	53
Sunshade as Cooling Support (FbSunshadeCoolingSupport)	56
Automatic Commands	59
Automatic sunlight sensor (FbSunshadeAutomaticSolarControl)	59
Automatic Twilight Control (FbSunshadeAutomaticTwilightControl)	62
Slat Tracking (FbSunshadeSlatTracking)	65
Other Functions.....	69
Position command transfer for sunshade (FuTypSunshade)	69

Position transfer for sunshade (FuTypSunshadePosition)	70
Multiplexer Sunshade (FuMUX_2_Sunshade, FuMUX_3_Sunshade)	71
Sensor Functions	72
Presence detection (FbPresenceSensor)	72
Controlling the Room Usage Type	74
Scene Control for lighting (FbControlLightScene)	74
Selecting a Group from the Scene Control for Lighting (FuGetLightSceneValue)	76
Scene Control for Sunshade (FbControlSunshadeScene)	77
Segment Control	79
Segment control for lighting (FbLightControl_X_Segments)	79
Segment Control for Sunshade (FbSunshadeControl_X_Segments)	81
Signal Transfer to the Segment Control for Sunshade (FuTypSunshadeSegmentControl)	84
Auxiliary Functions	86
Pushbutton Evaluation	86
Evaluation of "Multiple" Clicks (FbWB_EvaluateMultipleClick)	86
Evaluation of "short"/"long" Key Action (FbWB_EvaluateShortLongPress)	88
Analog Signals	89
Scaling the Input Values 0 to 32767 (FuWB_AI)	89
Scaling of Temperature Values In °C (FuWB_AI_Temp)	90
Scaling of Output Values 0 to 32767 (FuWB_AO)	91
Low Pass Filter, 1st Order (FbWB_LowPassFilter)	92
Low Pass Filter, 1st Order for AI 0 - 32767 (FbWB_LowPassFilterAI)	94
Low Pass Filter, 1st Order for Temperatures (FbWB_LowPassFilterTemp)	96
Low Pass Filter, 1st Order for Bus Signals (FbWB_LowPassFilterBus)	98
Additional Functions	100
Sun Position Calculation (FbWB_CalculateSunPosition)	100
Sunrise and Sunset Calculation (FbWB_CalculateSunriseSunset) ..	101
Hysteresis (FbWB_Hysteresis)	102
Operating Hours Counter (FbWB_OperatingHours_01)	104
Characteristics	105
Two-Point Characteristics (FuWB_2Point)	105
Ten-Point Characteristics (FuWB_10Point)	107
Four-Point Characteristic (FuWB_4Point)	109
Appendix	110
Structures	110
typLight	110
typSunshade	110
Initial Values for Variables	111
Calibration of the Brightness Measurement	112

Information about This Documentation

Note:**Always retain this documentation!**

This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user of the product. In addition, ensure that any supplement to this documentation is included, if necessary.

Copyright

This document, including all figures and illustrations contained therein, is subject to copyright. Any use of this document that infringes upon the copyright provisions stipulated herein is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g. photocopying), as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

Symbols

Attention**Attention!**

Boundary conditions that must always be observed to ensure smooth operation.

Note**Important note!**

Routines or advice for efficient use of a device and software optimization.

Information**Additional Information**

Refers to additional information which is not an integral part of this documentation (e.g. the Internet).

Number Notation

Table 1: Number Notation

Number code	Example	Remark
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated by a period

Font Conventions

Table 2: Font Conventions

Font type	Explanation
<i>italic</i>	Names of the paths and files are displayed in italics, e.g.: <i>C:\Programs\WAGO Software</i>
Menu	Menu options are displayed in bold e.g. Save
>	A “greater than” symbol between two names denotes the selection of a menu option from a menu, e.g.: File > New
Input	Designation of input or optional fields are displayed in bold; e.g.: Start of measurement range
“Value”	Input or selection values are displayed in quotation marks, e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Button labels in the dialogs are displayed in bold and enclosed in square brackets, e.g.: [input]
[Key]	Key labels on the keyboard are displayed in bold and enclosed in square brackets, e.g.: [F5]

Important Notes

To ensure fast installation and start-up of the units, we strongly recommend that the following information and explanations are carefully read and adhered to.

Subject to Change

WAGO Kontakttechnik GmbH & Co. KG reserves the right to make any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards. WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

Intended Use

For each individual application, the components are supplied from the factory with a dedicated hardware and software configuration. Modifications are only admitted within the framework of the possibilities documented in this document. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

Technical Condition of Specified Devices

For each individual application, the devices are supplied from the factory with a dedicated hardware and software configuration. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG. Please direct any requests for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

Function block Overview

The guideline VDI 3813 “Building automation and control systems (BACS)” published in May 2011 describes basic functions and macro-functions for the room automation. The guideline targets the efficient room control functionality in basic and final design.

The function blocks of the WAGO building library (WagoBuilding_01.lib) are based on this guideline. The functions and macro-functions of the library are according to the room control functions of VDI 3813. However some parameters of the functions and macro-functions are added other than the mentioned parameters in VDI 3813 guideline.

The corresponding VDI 3813 functions and function blocks are as follows:

Table 3: Lighting

Function block Name	Building Automation Functions	Standard
Latched relay (FbLatchedRelay)	Light actuator/Actuate light	VDI 3813 part 2 – 6.2.2 / 6.4.2
Advanced Latched Relay (FbAdvancedLatchedRelay)	Light actuator/Actuate light	VDI 3813 part 2 – 6.2.2 / 6.4.2
Dimmer double Button (FbDimmerDoubleButton)	Light actuator/Actuate light	VDI 3813 part 2 – 6.2.2 / 6.4.2
Dimmer single button (FbDimmerSingleButton)	Light actuator/Actuate light	VDI 3813 part 2 – 6.2.2 / 6.4.2
Automatic light (FbMacroAutomaticLights)	Light actuator/Actuate light/Automatic light	VDI 3813 part 2 – 6.2.2 / 6.4.2 / 6.5.8
Constant light control (FbMacroConstantLightControl)	Light actuator/Actuate light/Constant-light control	VDI 3813 part 2 – 6.2.2 / 6.5.10
Daylight dependent lighting (FbMacroDaylightDependentLighting)	Light actuator/Actuate light/Daylight-dependent lighting	VDI 3813 part 2 – 6.2.2 / 6.4.2 / 6.5.9
Light control (FbMacroLightControl)	Light actuator/Actuate light/Light control	VDI 3813 - 6.2.2 / 6.4.2 / 6.5.6
Stairwell light control (FbMacroStairwellLightControl)	Light actuator/Actuate light/Stairwell light control	VDI 3813 part 2 – 6.2.2 / 6.4.2 / 6.5.7
Twilight control (FbMacroTwilightControl)	Light actuator/Actuate light/Twilight control	VDI 3813 part 2 – 6.2.2 / 6.4.2 / 6.5.11

Table 4: Sunshade

Function block Name	Building Automation Functions	Standard
Sunshade actuator (FbSunshadeActuator)	Sunshade actuator/Actuate sunshade/Priority control	VDI 3813 part 2 – 6.2.3 / 6.4.3 / 6.5.12
Position command transfer for sunshade (FuTypSunshade)	Actuate sunshade	VDI 3813 part 2 – 6.4.3
Weather protection (FbBasicWeatherProtection)	Weather protection	VDI 3813 part 2 – 6.5.18
Weather protection with weather assessment (FbAdvanceWeatherProtection)	Weather protection	VDI 3813 part 2 – 6.5.18
Sunshade as heating support (FbSunshadeHeatingSupport)	Automatic thermal control	VDI 3813 part 2 – 6.5.17
Sunshade as cooling support (FbSunshadeCoolingSupport)	Automatic thermal control	VDI 3813 part 2 – 6.5.17
Automatic sunlight sensor (FbSunshadeAutomaticSolarControl)	Automatic solar control	VDI 3813 part 2 – 6.5.14
Automatic twilight control (FbSunshadeAutomaticTwilightControl)	Automatic twilight control	VDI 3813 part 2 – 6.5.13
Slat tracking (FbSunshadeSlatTracking)	Slat tracking	VDI 3813 part 2 – 6.5.15

Table 5: Sensor Functions

Function block Name	Building Automation Functions	Standard
Presence detection (FbPresenceSensor)	Presence detection/Occupancy evaluation	VDI 3813 part 2 – 6.1.2 VDI 3813 part 2 – 6.5.2

Table 6: Scene Control

Function block Name	Building Automation Functions	Standard
Scene control for lighting (FbControlLightScene)	Select room utilization type/Control via room utilization types	VDI 3813 part 2 – 6.4.6 / 6.5.3
Scene control for sunshade (FbControlSunshadeScene)	Select room utilization type/Control via room utilization types	VDI 3813 part 2 – 6.4.6 / 6.5.3

Table 7: Partition Wall Control

Function block Name	Building Automation Functions	Standard
Segment control for lighting (FbLightControl_X_Segments)	Partition wall control	VDI 3813 part 2 – 6.5.5
Segment control for sunshade (FbSunshadeControl_X_Segments)	Partition wall control	VDI 3813 part 2 – 6.5.5

Lighting

Latched relay (FbLatchedRelay)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbLatchedRelay	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xButton	BOOL	Touch signal
xCentralOnxCentralOn	BOOL	Switch on central
xCentralOff	BOOL	Switch off central
Return value:	Data type:	Comment:
xActuator	BOOL	Output switching signal
Graphical illustration:		
<div><div>FbLatchedRelay</div><div><div>xButton</div><div>xCentralOn</div><div>xCentralOff</div></div><div>xActuator</div></div>		
Function description:		
<p>The FbLatchedRelay function block maps the function of a latched relay. The switching function corresponds to a toggle flip-flop.</p> <p>The function block reacts to rising switching signals at the “xButton” input. With every positive switching signal at the “xButton” input, the latched relay switches its status value at the “xActuator” output.</p> <p>The “xCentralOn” and “xCentralOff” inputs serve to connect central ON/OFF commands for the “xActuator” output. The “xCentralOn” input sends an ON command in the case of a rising edge. The “xCentralOff” input sends an OFF command in the case of a rising edge.</p>		

Advanced Latched Relay (FbAdvancedLatchedRelay)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbAdvancedLatchedRelay	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xButton		BOOL	Touch signal
xCentralOnxCentralOn		BOOL	Switch on central
xCentralOff		BOOL	Switch off central
typL_SCENE		typLight	Parameter input for scene recall
xFeedback		BOOL	Status feedback of the segment control
Input/Output parameters:		Data type:	Comment:
xRecoveryValue		BOOL	Switching value with voltage recovery
Return value:		Data type:	Comment:
xActuator		BOOL	Output switching signal
typL_Segment		typLight	Parameter output for segment control
Graphical illustration:			
<div><div>FbAdvancedLatchedRelay</div><div><div>xButton</div><div>xCentralOn</div><div>xCentralOff</div><div>typL_SCENE</div><div>xFeedback</div><div>xRecoveryValue</div><div>xActuator</div><div>typL_Segment</div></div></div>			

Function description:

The **FbAdvancedLatchedRelay** function block maps the function of a latched relay. The switching function corresponds to a toggle flip-flop. The function block makes it possible to define a switching value with voltage recovery.

The function block reacts to rising switching signals at the **"xButton"** input. With every positive switching signal at the **"xButton"** input, the latched relay switches its status value at the **"xActuator"** output.

The **"xCentralOn"** and **"xCentralOff"** inputs serve to connect central ON/OFF commands. The **"xCentralOn"** input sends an ON command in the case of a rising edge. The **"xCentralOff"** input sends an OFF command in the case of a rising edge.

The **"typL_SCENE"** input is used for scene control and can be linked to a scene function block. When an update signal is received, the transmitted switching value will be evaluated. If a scene switching value is greater than 0, **"xActuator"** switches to TRUE.

The **"xRecoveryValue"** input/output variable maps the switching behavior after voltage recovery. The following states can be defined for the switching behavior:

- Always switch off after voltage recovery:
Initialize the variable **"xRecoveryValue"** with FALSE
- Always switch on after voltage recovery:
Initialize the variable **"xRecoveryValue"** with TRUE
- Always recover the last value after voltage recovery:
Declare the variable **"xRecoveryValue"** as RETAIN PERSISTENT without initialization.

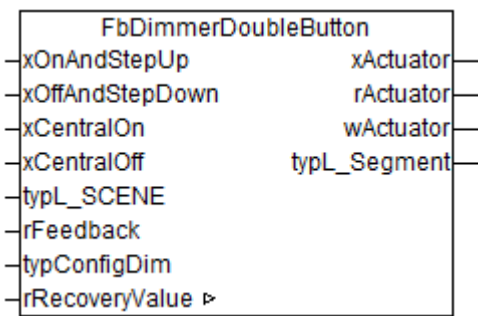
The **"xFeedback"** input is used as status feedback when connecting to the segment control. The current switching state of a segment must be passed to the **"xFeedback"** input. Hence the function block will receive the current switching state.

The **"xActuator"** output indicates the switching state of the latched relay.

The **"typL_Segment"** is used to connect the function block to the segment control.

Dimmer Double Button (FbDimmerDoubleButton)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbDimmerDoubleButton	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xOnAndStepUp	BOOL	Switch ON/step up button signal
xOffAndStepDown	BOOL	Switch OFF/step down button signal
xCentralOnxCentralOn	BOOL	Switch on central
xCentralOff	BOOL	Switch off central
typL_SCENE	typLight	Parameter for scene recall
rFeedback	REAL	Status feedback of the segment control [%] Value range: 0 – 100%
typConfigDim	typConfigDim	Parameter for the macro function block
.typDimmer	typDimmer	Parameter for dimming
.rMaximumDimLevel	REAL	Maximum dimming value [%] Value range: 1 – 100% Default setting: 100%
.rMinimumDimLevel	REAL	Minimum dimming value [%] Value range: 1 – 100% Default setting: 5%
.rSwitchOnDimLevel	REAL	Switch ON at dimming value [%] Value range: 1 – 100% Default setting: 90% Switch ON at last dimming value: 101%
.tShortPushButton	TIME	Maximum time for a brief button press Default setting: t#500ms
.xMinLevelAsOff	BOOL	The lighting is switched to “ <i>rMinimumDimLevel</i> ” instead of the switch-off command.
.xOnlyDimming	BOOL	The “ <i>xOnAndStepUp</i> ” and “ <i>xOffAndStepDown</i> ” inputs only accept dimming commands.
.xSwitchOnAndStepUp	BOOL	Switch ON before stepping up
.xStepDownAndSwitchOff	BOOL	Switch OFF when the minimum dimming value is reached

.typLightActuator	typLightActuator	Parameter for light actuator
.rSetRecoveryValue	REAL	Dimming value with voltage recovery Value range: 0 – 100% Use last dimming value: 101%
.tDimPeriod	TIME	Dimming period between minimum and maximum dimming value Default setting: t#5s
.bExponent	BYTE	Logarithmic dimming curve for adaptation to the human eye Linear dimming curve: 0% Logarithmic dimming curve: 1 – 100% Default setting: 0%
Input/Output parameters:		
rRecoveryValue	REAL	Dimming value with voltage recovery [%]
Return value:		
xActuator	BOOL	Output switching signal
rActuator	REAL	Output switching value [%] Value range: 0 – 100%
wActuator	WORD	Output switching value Value range = 0 – 32767
typL_Segment	typLight	Parameter output for segment control
Graphical illustration:		
		

Function description:

The **FbDimmerDoubleButton** function block can be used to dim a light with a double button.

The **“xOnAndStepUp”** and **“xOffAndStepDown”** button inputs evaluate short and long button commands. A short button press transmits an ON/OFF command. The switch-on value can be parameterized. A long button press transmits an UP/DOWN command. The dimming value can be dimmed between the limiting values.

The **“xCentralOn”** and **“xCentralOff”** inputs serve to connect central ON/OFF commands. The **“xCentralOn”** input sends an ON command to the maximum dimming value in the case of a rising edge. The **“xCentralOff”** input sends an OFF command in the case of a rising edge. The actuation time of the **“xCentralOn”** and **“xCentralOff”** inputs does not affect the switching behavior.

The **“typL_SCENE”** input is used for scene control and can be linked to a scene function block. When an update signal is received, the defined dimming value is transmitted.

The **“rFeedback”** input is used as status feedback when connecting to the segment control. The current dimming value of a segment must be passed to the **“rFeedback”** input. Hence the function block will receive the current dimming value.

The **“typConfigDim”** input contains all parameter values for the function block:

- **“.typDimmer”** contains the dimming parameters:
 - **“.rMaximumDimLevel”** defines the maximum dimming value as a percentage.
 - **“.rMinimumDimLevel”** defines the minimum dimming value as a percentage.
 - **“.rSwitchOnDimLevel”** defines a fixed switch-on value. If the dimming value is to be reset before switching off, set **“rSwitchOnDimLevel”** to 101%.
 - **“.tShortPushButton”** defines the maximum actuation time of the **“xOnAndStepUp”** and **“xOffAndStepDown”** inputs for a short button press. If the inputs are pressed longer, the input is interpreted as a long button press.
 - **“.xMinLevelAsOff”** is set to TRUE if the switch-off command does not switch off the light, but should be set to the **“.rMinimumDimLevel”** dimming value.
 - **“.xOnlyDimming”** is applied if the button inputs should only dim. Short button commands are not evaluated.
 - **“.xSwitchOnAndStepUp”** is applied if a switch-on command is sent before stepping up.
 - **“.xStepDownAndSwitchOff”** is applied if a switch-off command is sent when the minimum dimming value is not reached.

- **“.typLightActuator”** contains the light actuator parameters:
 - **“.rSetRecoveryValue”** defines the behavior of the light actuator with voltage recovery. The assignment is explained in the table below.
 - **“.tDimPeriod”** defines the dimming period in which the output signal is switched from **“.rMinimumDimLevel”** to **“.rMinimumDimLevel”**.
 - **“.bExponent”** allows the user to use a logarithmic dimming curve that is adapted to the human eye. If **“.bExponent”** is rising, the elongation of the dimming curve increases.

The **“.rRecoveryValue”** input/output variable maps the switching behavior after voltage recovery. The assignment is explained in the table below.

The **“xActuator”** output indicates the digital switching state. If a percentage dimming value is greater than 0, **“xActuator”** switches to TRUE.

The **“rActuator”** output indicates the percentage dimming value. The possible dimming value in the ON state is limited by the maximum and minimum dimming value. In the OFF state, the dimming value is 0.

The **“wActuator”** output indicates the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

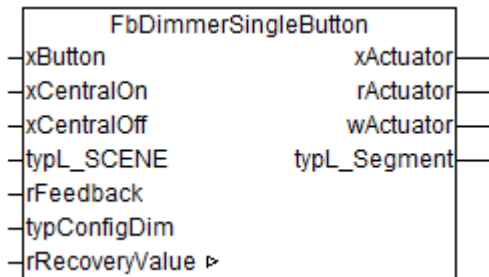
The **“typL_Segment”** is used to connect the function block to the segment control.

Behavior after voltage recovery:

Parameter “typConfigDim.typLightActuator.rSetRecoveryValue”	Actuator behavior after voltage recovery	Variable “rRecoveryValue”
0%	Power OFF	To save as a variable
1-100%	Switching on to parameterized value	To save as a variable
101%	Restore last dimming value	To save as RETAIN PERSISTENT variable

Dimmer Single Button (FbDimmerSingleButton)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbDimmerSingleButton	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xButton		BOOL	Switch and dim button signal
xCentralOn		BOOL	Switch on central
xCentralOff		BOOL	Switch off central
typL_SCENE		typLight	Parameter input for scene recall
rFeedback		REAL	Status feedback of the segment control [%] Value range: 0 – 100%
typConfigDim		typConfigDim	Parameter for the macro function block
.typDimmer		typDimmer	Parameter for dimming
.rMaximumDimLevel		REAL	Maximum dimming value [%] Value range: 1 – 100% Default setting: 100%
.rMinimumDimLevel		REAL	Minimum dimming value [%] Value range: 1 – 100% Default setting: 5%
.rSwitchOnDimLevel		REAL	Switch ON at dimming value [%] Value range: 1 – 100% Default setting: 90% Switch ON at last dimming value: 101%
.tShortPush Button		TIME	Maximum time for a brief button press Default setting: t#500ms
.xMinLevelAsOff		BOOL	The lighting is switched to “ <i>rMinimumDimLevel</i> ” instead of the switch-off command.
.xOnlyDimming		BOOL	The “ <i>xButton</i> ” input only accepts dimming commands.
.xSwitchOnAndStepUp		BOOL	Switch ON before stepping up
.xStepDownAndSwitchOff		BOOL	Switch OFF when the minimum dimming value is reached

.typLightActuator	typLightActuator	Parameter for light actuator
.rSetRecoveryValue	REAL	Dimming value with voltage recovery Value range: 0 – 100% Use last dimming value: 101%
.tDimPeriod	TIME	Dimming period between minimum and maximum dimming value Default setting: t#5s
.bExponent	BYTE	Logarithmic dimming curve for adaptation to the human eye Linear dimming curve: 0% Logarithmic dimming curve: 1 – 100% Default setting: 0%
Input/Output parameters:		
rRecoveryValue	REAL	Dimming value with voltage recovery [%]
Return value:		
xActuator	BOOL	Output switching signal
rActuator	REAL	Output dimming value [%] Value range: 0 – 100%
wActuator	WORD	Output dimming value Value range = 0 – 32767
typL_Segment	typLight	Parameter output for segment control
Graphical illustration:		
 <pre> graph LR subgraph FbDimmerSingleButton direction TB xButton xCentralOn xCentralOff typL_SCENE rFeedback typConfigDim rRecoveryValue end xButton --> xActuator xCentralOn --> rActuator xCentralOff --> wActuator typL_SCENE --> typL_Segment </pre>		

Function description:

The **FbDimmerSingleButton** function block can be used to dim a light with a single button.

The **“xButton”** button input evaluates short and long button commands. A short button press transmits an ON/OFF command. The switch-on value can be parameterized. A long button press transmits an UP/DOWN command. The light is dimmed up after switching on. The dimming value can be dimmed between the limiting values.

The **“xCentralOn”** and **“xCentralOff”** inputs serve to connect central ON/OFF commands. The **“xCentralOn”** input sends an ON command to the maximum dimming value in the case of a rising edge. The **“xCentralOff”** input sends an OFF command in the case of a rising edge. The actuation time of the **“xCentralOn”** and **“xCentralOff”** inputs does not affect the switching behavior.

The **“typL_SCENE”** input is used for scene control and can be linked to a scene function block. When an update signal is received, the defined dimming value is transmitted.

The **“rFeedback”** input is used as status feedback when connecting to the segment control. The current dimming value of a segment must be passed to the **“rFeedback”** input. Hence the function block will receive the current dimming value.

The **“typConfigDim”** input contains all parameter values for the function block:

- **“.typDimmer”** contains the dimming parameters:
 - **“.rMaximumDimLevel”** defines the maximum dimming value as a percentage.
 - **“.rMinimumDimLevel”** defines the minimum dimming value as a percentage.
 - **“.rSwitchOnDimLevel”** defines a fixed switch-on value. If the dimming value is to be reset before switching off, set **“rSwitchOnDimLevel”** to 101%.
 - **“.tShortPushButton”** defines the maximum actuation time of the **“xButton”** input for a short button press. If the input is pressed longer, the input is interpreted as a long button press.
 - **“.xMinLevelAsOff”** is set to TRUE if the switch-off command does not switch off the light, but should be set to the **“.rMinimumDimLevel”** dimming value.
 - **“.xOnlyDimming”** is applied if the button input should only dim. Short button commands are not evaluated.
 - **“.xSwitchOnAndStepUp”** is applied if a switch-on command is sent before stepping up.
 - **“.xStepDownAndSwitchOff”** is applied if a switch-off command is sent when the minimum dimming value is not reached.

- **“.typLightActuator”** contains the light actuator parameters:
 - **“.rSetRecoveryValue”** defines the behavior of the light actuator with voltage recovery. The assignment is explained in the table below.
 - **“.tDimPeriod”** defines the dimming period in which the output signal is switched from **“.rMinimumDimLevel”** to **“.rMinimumDimLevel”**.
 - **“.bExponent”** allows the user to use a logarithmic dimming curve that is adapted to the human eye. If **“.bExponent”** is rising, the elongation of the dimming curve increases.

The **“.rRecoveryValue”** input/output variable maps the switching behavior after voltage recovery. The assignment is explained in the table below.

The **“xActuator”** output indicates the digital switching state. If a percentage dimming value is greater than 0, **“xActuator”** switches to TRUE.

The **“rActuator”** output indicates the percentage dimming value. The possible dimming value in the ON state is limited by the maximum and minimum dimming value. In the OFF state, the dimming value is 0.

The **“wActuator”** output indicates the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

The **“typL_Segment”** is used to connect the function block to the segment control.

Behavior after voltage recovery:

Parameter “typConfigDim.typLightActuator.rSetRecoveryValue”	Actuator behavior after voltage recovery	Variable “rRecoveryValue”
0%	Power OFF	To save as a variable
1-100%	Switching on to parameterized value	To save as a variable
101%	Restore last dimming value	To save as RETAIN PERSISTENT variable

Automatic Light (FbMacroAutomaticLights)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbMacroAutomaticLights	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xPresence		BOOL	Occupancy status of the occupancy evaluation / presence detection (e.g. from FbPresenceSensor)
typConfigAutomaticLights		typConfigAutomaticLights	Parameter for the macro function block
.rSwitchOnValue		REAL	Switch-on value [%] Value range: 0 – 100% Default setting: 100%
.tDelayOff		TIME	Switch-off delay Default setting: t#5m
Return value:		Data type:	Comment:
xActuator		BOOL	Output switching signal
rActuator		REAL	Output switching value [%] Value range: 0 – 100%
wActuator		WORD	Output switching value Value range = 0 – 32767
Graphical illustration:			
<div><div>FbMacroAutomaticLights</div><div><div>xPresence</div><div>typConfigAutomaticLights</div></div><div><div>xActuator</div><div>rActuator</div><div>wActuator</div></div></div>			

Function description:

The **FbMacroAutomaticLights** function block switches the room lighting depending on presence detection. Natural lighting by daylight is ignored. The function block is particularly suited for rooms without direct sunlight, e.g. corridors and restrooms.

The **“xPresence”** input is connected to the presence detection. The light is switched ON when presence detection is enabled. If presence detection is disabled, the light is switched off after the OFF delay has elapsed.

The **“typConfigAutomaticLights”** input contains all parameter values for the function block:

- **“rSwitchOnValue”** defines the percentage switch-on value of the lighting.
- **“tDelayOff”** defines the switch-off delay. With renewed presence detection, the elapsed time is reset.

The **“xActuator”** output indicates the digital switching state. If a percentage dimming value is greater than 0, **“xActuator”** switches to TRUE.

The **“rActuator”** output indicates the percentage dimming value. The value is specified by the **“typAutomaticLights.rSwitchOnValue”** switch-on value. In the OFF state, the dimming value is 0.

The **“wActuator”** output indicates the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

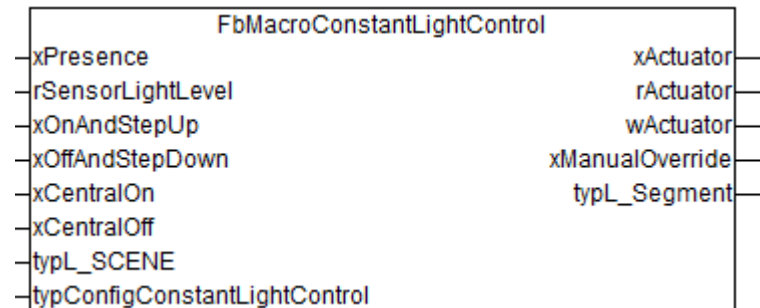
Constant Light Control (FbMacroConstantLightControl)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbMacroConstantLightControl	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xPresence	BOOL	Occupancy status of the occupancy evaluation / presence detection (e.g. from FbPresenceSensor)
rSensorLightLevel	REAL	Measured light intensity [lx]
xOnAndStepUp	BOOL	Switch ON/step up button signal
xOffAndStepDown	BOOL	Switch OFF/step down button signal
xCentralOnxCentralOn	BOOL	Switch on central
xCentralOff	BOOL	Switch off central
typL_SCENE	typLight	Parameter input for scene recall
typConfigContantLightControl	typConfigContantLightControl	Parameter for the macro function block
.typDimmer	typDimmer	Parameter for dimming
.rMaximumDimLevel	REAL	Maximum dimming value [%] Value range: 1 – 100% Default setting: 100%
.rMinimumDimLevel	REAL	Minimum dimming value [%] Value range: 1 – 100% Default setting: 5%
.rSwitchOnDimLevel	REAL	Manual switch ON at dimming value [%] Value range: 1 – 100% Default setting: 90% Switch ON at last dimming value: 101%
.tShortPushButton	TIME	Maximum time for a brief button press Default setting: t#500ms
.xMinLevelAsOff	BOOL	The lighting is switched to <i>“rMinimumDimLevel”</i> instead of the switch-off command.
.xOnlyDimming	BOOL	The <i>“xOnAndStepUp”</i> and <i>“xOffAndStepDown”</i> inputs only accept dimming commands.
.xSwitchOnAndStepUp	BOOL	Switch ON before stepping up
.xStepDownAndSwitchOff	BOOL	Switch OFF when the minimum dimming value is reached

.typConstantLightControl	typConstantLightControl	Parameter for constant light control
.rSetpointIlluminance	REAL	Minimum light intensity [lx] Default setting: 500 lx
.rSwitchOnValue	REAL	Switch-on dimming value of the control [%] Value range: 0 – 100% Default setting: 100%
.tDelayAutoSwitchOn	TIME	Switch-on delay of the control when changing the light intensity Default setting: t#10 s
.rDeviationIlluminance	REAL	Threshold value of the light intensity [lx] Default setting: 50 lx
.tOffDelayAtMinLevel	TIME	Switch-off delay when the minimum dimming value is reached Default setting: t#15m
.tTimeToDisableManual	REAL	Override time of the constant light control by manual input Default setting: t#15m
.rMinValueLightControl	REAL	Minimum dimming value of the control [%] Value range: 1 – 100% Default setting: 10%
.typBrightnessMeasurement	typBrightnessMeasurement	Parameter for the brightness measurement
.rGain	REAL	Gain factor for correct measurement of the light intensity at the workstation Default setting: 3
.rGainAdaptation	REAL	Compensation for the different influences from daylight and artificial light on the sensor Value range: 0 – 90% Default setting: 20%
.typLightActuator	typLightActuator	Parameter for light actuator
.rSetRecoveryValue	REAL	No function for the constant light control
.tDimPeriod	TIME	Dimming period between minimum and maximum dimming value Default setting: t#5s
.bExponent	BYTE	Logarithmic dimming curve for adaptation to the human eye Linear dimming curve: 0% Logarithmic dimming curve: 1 – 100% Default setting: 0%

Return value:	Data type:	Comment:
xActuator	BOOL	Output switching signal
rActuator	REAL	Output dimming value [%] Value range: 0 – 100%
wActuator	WORD	Output dimming value Value range = 0 – 32767
xManualOverride	BOOL	Manual override
typL_Segment	typLight	Parameter output for segment control

Graphical illustration:



Function description:

The **FbMacroConstantLightControl** function block is used for automatic control of room lighting to a minimum light intensity. Daylight is taken into account. A PID controller controls the lighting internally. The constant light control can be overridden by button inputs or a scene recall.

The “**xPresence**” input is connected to the presence detection. It defines the occupancy status of the room. A change to the occupancy status leads to instantaneous switching.

The measured light intensity of the sensor is connected to the “**rSensorLightLevel**” input. The sensor must be calibrated with parameters for the “**typConfigConstantLightControl.typBrightnessMeasurement**” brightness measurement. Calibration of the brightness measurement is described in [Appendix](#).

The “**xOnAndStepUp**” and “**xOffAndStepDown**” button inputs override the automatic constant light control for a configurable time. The button inputs evaluate short and long button commands. A short button press transmits an ON/OFF command. The switch-on dimming value can be parameterized. A long button press transmits an UP/DOWN command. The dimming value can be dimmed between the dimming value range.

The “**xCentralOn**” and “**xCentralOff**” inputs serve to connect central ON/OFF commands. The “**xCentralOn**” input sends an ON command to the maximum dimming value in the case of a rising edge. The “**xCentralOff**” input sends an OFF command in the case of a rising edge. The actuation time of the “**xCentralOn**” and “**xCentralOff**” inputs does not affect the switching behavior.

The “**typL_SCENE**” input is used for scene control and can be linked to a scene function block. When an update signal is received, the defined dimming value is transmitted and the automatic constant light control overridden.

The **“typConfigConstantLightControl”** input contains all parameter values for the function block:

- **“.typDimmer”** contains the dimming parameters:
 - **“.rMaximumDimLevel”** defines the maximum dimming value as a percentage.
 - **“.rMinimumDimLevel”** defines the minimum dimming value as a percentage.
 - **“.rSwitchOnDimLevel”** defines a fixed switch-on dimming value via the **“xOnAndStepUp”** button input. If the dimming value is to be reset before switching off, set **“.rSwitchOnDimLevel”** to 101%.
 - **“.tShortPushButton”** defines the maximum actuation time of the **“xOnAndStepUp”** and **“xOffAndStepDown”** inputs for a short button press. If the inputs are pressed longer, the input is interpreted as a long button press.
 - **“.xMinLevelAsOff”** is set to TRUE if the switch-off command does not switch off the light, but should be set to the **“.rMinimumDimLevel”** dimming value.
 - **“.xOnlyDimming”** is applied if the button inputs should only dim. Short button commands are not evaluated.
 - **“.xSwitchOnAndStepUp”** is applied if a switch-on command is sent before stepping up.
 - **“.xStepDownAndSwitchOff”** is applied if a switch-off command is sent when the minimum dimming value is not reached.
- **“.typConstantLightControl”** contains the parameters for constant light control:
 - **“.rSetpointIlluminance”** defines the threshold for the minimum light intensity at the workstation.
 - **“.rSwitchOnValue”** defines the switch-on value of the control as a percentage. The parameterized maximum dimming value cannot be overridden.
 - **“.tDelayAutoSwitchOn”** defines the switch-on delay of the constant light control after falling below the minimum light intensity.
 - **“.rDeviationIlluminance”** defines the threshold value of the light intensity. This must be overridden, so that the control switches on the lighting.
 - **“.tOffDelayAtMinLevel”** specifies the switch-off delay time, which switches off the lighting after reaching the minimum value of the control and overridden minimum light intensity. The conditions must be met over the entire period to send the switch-off command.
 - **“.tTimeToDisableManual”** defines the override time by button inputs or scene recall. With renewed override, the elapsed time is reset. If the time is set to zero seconds, the override is only reset after the presence information falls off.
 - **“.rMinValueLightControl”** specifies the minimum dimming value as a percentage at which the control can step down. The value may differ from the manual minimum dimming value.

- **“.typBrightnessMeasurement”** contains the sensor configuration parameters. The calibration process is described in [Appendix](#) in detail. The following parameters are defined:
 - **“.rGain”** specifies the gain factor of the light intensity at the workstation compared to the measured light intensity.
 - **“.rGainAdaptation”** is used to compensate for the different influences from daylight and artificial light on the sensor.
- **“.typLightActuator”** contains the light actuator parameters:
 - **“.rSetRecoveryValue”** has no function in the constant light control.
 - **“.tDimPeriod”** defines the dimming period in which the output signal is switched from **“.rMinimumDimLevel”** to **“.rMinimumDimLevel”**.
 - **“.bExponent”** allows the user to use a logarithmic dimming curve that is adapted to the human eye. If **“.bExponent”** is rising, the elongation of the dimming curve increases.

The **“xActuator”** output indicates the digital switching state. If a percentage dimming value is greater than 0, **“xActuator”** switches to TRUE.

The **“rActuator”** output indicates the percentage dimming value. The possible dimming value in the ON state is limited by the maximum and minimum dimming value. In the OFF state, the dimming value is 0.

The **“wActuator”** output indicates the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

The **“xManualOverride”** output signals that the automatic function is overridden.

The **“typL_Segment”** is used to connect the function block to the segment control.

Daylight dependent lighting (FbMacroDaylightDependentLighting)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbMacroDaylightDependentLighting	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xPresence		BOOL	Occupancy status of the occupancy evaluation / presence detection
rSensorLightLevel		REAL	Measured light intensity [lx]
xOn		BOOL	Manual switch ON
xOff		BOOL	Manual switch OFF
typL_SCENE		typLight	Parameter input for scene recall
typConfigDaylightDependentLighting		typConfigDaylightDependentLighting	Parameter for the macro function block
.typDaylightDependentLighting		typDaylightDependentLighting	Parameter for room lighting control
.rSetpointIlluminance		REAL	Minimum light intensity [lx] Default setting: 500 lx
.tDelayOn		TIME	Switch-on delay when falling below the minimum light intensity Default setting: t#20s
.tDelayOff		TIME	Switch-off delay when exceeding the adjusted switch-off threshold Default setting: t#20s
.tTimeToDisablingManual		TIME	Override time of the daylight dependent lighting by manual input Default setting: t#15m
.rSwitchOnValue		REAL	Switch-on value [%] Value range: 0 – 100% Default setting: 100%
.tWaitTime		TIME	Time until the new switch-off threshold is calculated Default setting: t#20s
.rOffset		REAL	Offset for adjusting the switch-off threshold [lx] Default setting: 50 lx

.typBrightnessMeasurement	typBrightnessMeasurement	Parameter for the brightness measurement
.rGain	REAL	Gain factor for correct measurement of the light intensity at the workstation Default setting: 3
.rGainAdaptation	REAL	Compensation for the different influences from daylight and artificial light on the sensor Value range: 0 – 90% Default setting: 20%
Return value:		
xActuator	BOOL	Output switching signal
rActuator	REAL	Output switching value [%] Value range: 0 – 100%
wActuator	WORD	Output switching value Value range = 0 – 32767
xManualOverride	BOOL	Manual override
typL_Segment	typLight	Parameter output for segment control
Graphical illustration:		
<pre> graph LR subgraph FbMacroDaylightDependentLighting direction TB xPresence rSensorLightLevel xOn xOff typL_SCENE typConfigDaylightDependentLighting end xPresence --- xActuator rSensorLightLevel --- rActuator xOn --- wActuator xOff --- xManualOverride typL_SCENE --- typL_Segment typConfigDaylightDependentLighting </pre>		

Function description:

The **FbMacroDaylightDependentLighting** function block is used for daylight dependent room lighting to ensure the minimum light intensity. Daylight is taken into account. If there is sufficient natural light, the artificial lighting is switched off. If the setpoint is not reached, the lighting is switched on. After the lighting is switched on and the configurable time has elapsed, the switch-off threshold for artificial light is automatically adjusted, so that the minimum light intensity is ensured continuously. The daylight dependent lighting can be overridden by the button inputs or a scene recall.

The **“xPresence”** input defines the occupancy status of the room. A change to the occupancy status leads to instantaneous switching.

The measured light intensity of the sensor is connected to the **“rSensorLightLevel”** input. The sensor must be calibrated with parameters for the **“typConfigDaylightDependentLighting.typBrightnessMeasurement”** brightness measurement. Calibration is described in [Appendix](#).

The **“xOn”** and **“xOff”** button inputs override automatic daylight dependent lighting. A rising edge at the **“xOn”** input switches the lighting on. A rising edge at the **“xOff”** input switches the lighting off. The override is reset after a configurable time.

The **“typConfigDaylightDependentLighting”** input contains all parameter values for the function block:

- **“.typDaylightDependentLighting”** contains the parameters of daylight dependent lighting:
 - **“.rSetpointIlluminance”** defines the minimum light intensity at the workstation.
 - **“.tDelayOn”** defines the switch-on delay time when the minimum light intensity is not reached.
 - **“.tDelayOff”** defines the switch-off delay when the adjusted switch-off threshold is exceeded.
 - **“.tTimeToDisableManual”** defines the override time by button inputs or scene recall. With renewed override, the elapsed time is reset.
 - **“.rSwitchOnValue”** defines the percentage dimming value when switching on.
 - **“.tWaitTime”** specifies the time after which the adjusted switch-off threshold is calculated. During this time, the lighting is to achieve full light intensity after switching on.
 - **“.rOffset”** makes it possible to increase the adjusted switch-off threshold.
- **“.typBrightnessMeasurement”** contains the sensor configuration parameters. The calibration process is described in [Appendix](#) in detail. The following parameters are defined:
 - **“.rGain”** specifies the gain factor of the light intensity at the workstation compared to the measured light intensity.
 - **“.rGainAdaptation”** is used to compensate for the different influences from daylight and artificial light on the sensor.

The **“xActuator”** output indicates the digital switching state. If a percentage dimming value is greater than 0, **“xActuator”** switches to TRUE.

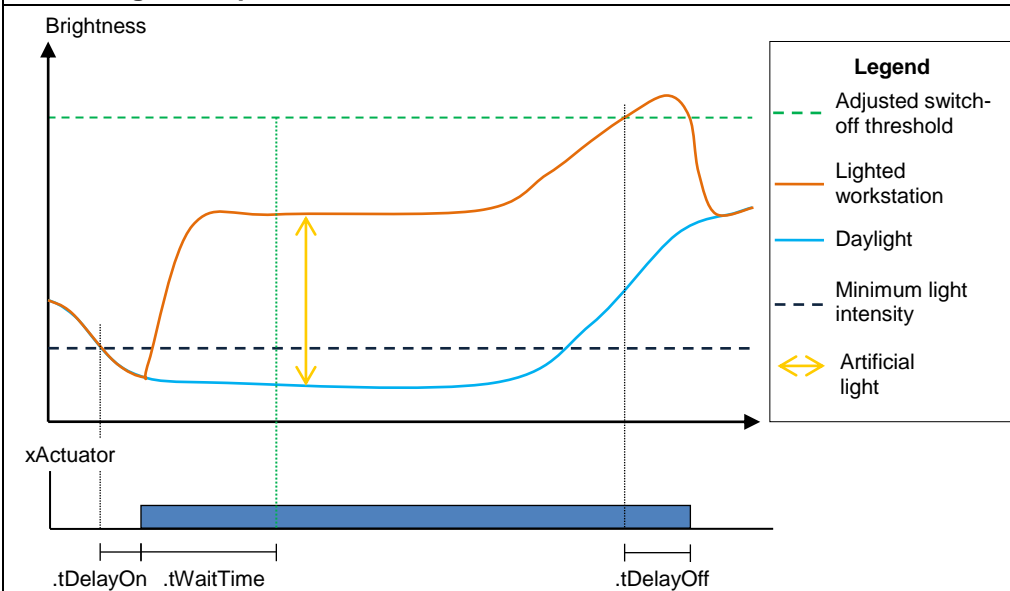
The **“rActuator”** output indicates the percentage dimming value. The value is specified by the **“typConfigDaylightDependentLighting.typDaylightDependentLighting.rSwitchOnValue”** switch-on-threshold or the **“typL_SCENE.rDimValue”** scene value. In the OFF state, the dimming value is 0.

The **“wActuator”** output indicates the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

The **“xManualOverride”** output signals that the automatic function is overridden.

The **“typL_Segment”** is used to connect the function block to the segment control.

Chronological sequence:



Calculating the adjusted switch-off threshold:

Adjustment = (current light intensity – switch-on light intensity) + minimum light intensity + offset

Light Control (FbMacroLightControl)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbMacroLightControl	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xSwitch	BOOL	Switching signal
typConfigLightControl	typConfigLightControl	Parameter for the macro function block
.rSwitchOnValue	REAL	Switch-on value [%] Value range: 0 – 100% Default setting: 100%
.typLightControl	typLightControl	Parameter for light control
.tDelayOn	TIME	Switch-on delay Default setting: t#1s
.tDelayOff	TIME	Switch-off delay Default setting: t#1s
Return value:	Data type:	Comment:
xActuator	BOOL	Output switching signal
rActuator	REAL	Output switching value [%] Value range: 0 – 100%
wActuator	WORD	Output switching value Value range = 0 – 32767
Graphical illustration:		
<div><div>FbMacroLightControl</div><div><div>xSwitch</div><div>typConfigLightControl</div></div><div><div>xActuator</div><div>rActuator</div><div>wActuator</div></div></div>		

Function description:

The **FbMacroLightControl** function block applicable to switching ON/OFF switchable and dimmable lighting systems.

The **“xSwitch”** input determines the switching behavior of the function block. A positive edge switches on, and negative edge switches off.

The **“typConfigLightControl”** input contains all parameter values for the function block:

- **“.rSwitchOnValue”** determines the percentage switch-on value.
- **“.typLightControl”** contains the parameters for light control:
 - **“.tDelayOn”** specifies the switch-on delay.
 - **“.tDelayOff”** specifies the switch-off delay.

The **“xActuator”** output indicates the digital switching state. If a percentage dimming value is greater than 0, **“xActuator”** switches to TRUE.

The **“rActuator”** output indicates the percentage dimming value. The value is specified by the **“typConfigLightControl.rSwitchOnValue”** switch-on value. In the OFF state, the dimming value is 0.

The **“wActuator”** output indicates the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

Stairwell Light Control (FbMacroStairwellLightControl)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbMacroStairwellLightControl	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xButton		BOOL	Touch signal
typConfigStairwellLightControl		typConfigStairwellLightControl	Parameter for the macro function block
.rSwitchOnValue		REAL	Switch-on value [%] Value range: 0 – 100% Default setting: 100%
.typStairwellLightControl		typStairwellLightControl	Parameter for stairwell light control
.tHoldingTime		TIME	Total lighting on-time Default setting: t#20s
.tPrewarning		TIME	Time of switch-off prewarning before switching off Default setting: t#5s
.rPrewarningLevel		REAL	Dimming value of the switch-off prewarning [%] Value range: 0 – 100% Default setting: 0%
Output parameter:		Data type:	Comment:
xActuator		BOOL	Output switching signal
rActuator		REAL	Output switching value [%] Value range: 0 – 100%
wActuator		WORD	Output switching value Value range = 0 – 32767
Graphical illustration:			
<div><div>FbMacroStairwellLightControl</div><div><div>xButton</div><div>typConfigStairwellLightControl</div></div><div><div>xActuator</div><div>rActuator</div><div>wActuator</div></div></div>			

Function description:

The **FbMacroStairwellLightControl** maps the function of a stairwell light control. A prewarning can be triggered before switching off.

The lighting is switched on by a positive edge at the **“xButton”** input for a configurable period. Another positive edge at **“xButton”** restarts the elapsed time.

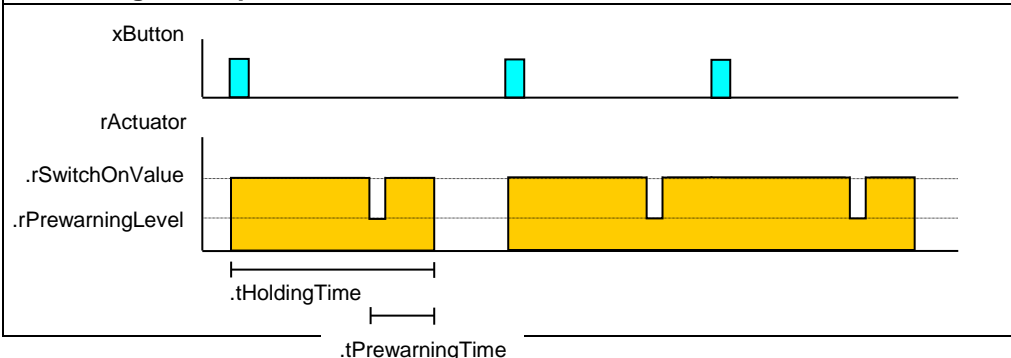
The **“typConfigStairwellLightControl”** input contains all parameter values for the function block:

- **“.rSwitchOnValue”** determines the percentage switch-on value.
- **“.typStairwellLightControl”** contains the parameters for stairwell light control:
 - **“.tHoldingTime”** specifies the total lighting on-time.
 - **“.tPrewarning”** defines the time before switching off at which a switch-off prewarning is triggered. The switch-off prewarning briefly switches the lighting to the configured **“.rPrewarningLevel”** value.
 - **“.rPrewarningLevel”** defines the percentage lighting dimming value of the switch-off prewarning.

The **“xActuator”** output indicates the digital switching state. If a percentage switching value is greater than 0, **“xActuator”** switches to TRUE.

The **“rActuator”** output indicates the percentage dimming value. The value is specified by the **“typConfigStairwellLightControl.rSwitchOnValue”** switch-on value. When switched off, the dimming value is 0. When the switch-off prewarning is used, the **“typConfigStairwellLightControl”** switching value is briefly output at the configured time.

The **“wActuator”** output indicates the switching value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

Chronological sequence:

Twilight Control (FbMacroTwilightControl)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbMacroTwilightControl	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xEnable	BOOL	Switching the twilight control ON/OFF Default setting: TRUE (executing function)
rIlluminance	REAL	Measured light intensity outside the building [lx]
typConfigTwilightControl	typConfigTwilightControl	Parameter for the twilight control
.rTwilightLimit	REAL	Switch-on threshold (at twilight) [lx] Default setting: 50 lx
.rSunriseLimit	REAL	Switch-off threshold (at sunrise) [lx] Default setting: 100 lx
.tTimeHysteresis	TIME	Time hysteresis of the switching operations Default setting: t#5m
.rDimValueAtTwilight	REAL	Switch-on threshold of the twilight control [%] Value range: 0 – 100% Default setting: 100%
.rDimValueAtSunrise	REAL	Switch-off threshold of the twilight control [%] Value range: 0 – 100% Default setting: 0%
Return value:	Data type:	Comment:
xActuator	BOOL	Output switching signal
rActuator	REAL	Output switching value [%] Value range: 0 – 100%
wActuator	WORD	Output switching value Value range = 0 – 32767
Graphical illustration:		
<div><div>FbMacroTwilightControl</div><div><div>xEnable</div><div>rIlluminance</div><div>typConfigTwilightControl</div><div>xActuator</div><div>rActuator</div><div>wActuator</div></div></div>		

Function description:

The **FbMacroTwilightControl** function block maps the function of the twilight control. The lighting is switched on depending on the outdoor light intensity. With low outdoor lighting, such as at dusk, the lighting is switched on. Conversely, the lighting is switched off when outdoor lighting is adequate, such as at daybreak.

The **“xEnable”** input can be used to disable twilight control. The information can be come from the time program or a command from the building management system.

The current measured intensity of the natural light is applied to the **“rIlluminance”** input.

The **“typConfigTwilightControl”** input contains all parameter values for the function block:

- **“.rTwilightLimit”** defines the threshold for switching on the lighting.
- **“.rSunriseLimit”** defines the threshold for switching off the lighting.
- **“.tTimeHysteresis”** sets a time hysteresis to prevent accidental switching after brief brightness variations. The limiting values of the light intensity must be met without interruption during the entire time interval to trigger a switching operation.
- **“.rDimValueAtTwilight”** specifies the percentage switch-on value when the **“.rTwilightLimit”** limiting value is not reached.
- **“.rDimValueAtSunrise”** specifies the percentage switch-off value when the **“.rSunriseLimit”** limiting value is exceeded.

The **“xActuator”** output indicates the digital switching state. If a percentage dimming value is greater than 0, **“xActuator”** switches to TRUE.

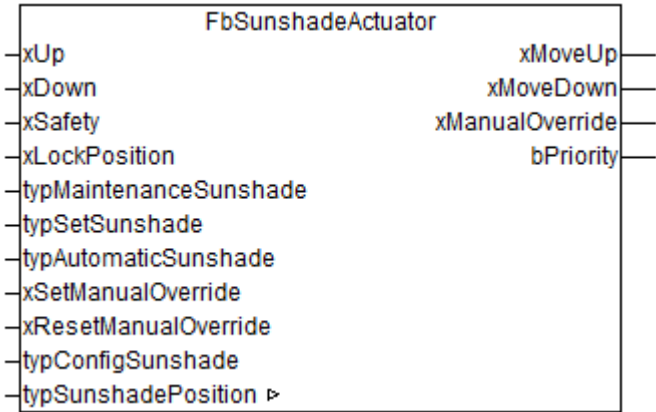
The **“rActuator”** output indicates the percentage dimming value. The value is specified by the **“typConfigTwilightControl.rDimValueAtTwilight”** switch-on value. When switched off, the **“typConfigTwilightControl.rSunriseLimit”** switching value is passed.

The **“wActuator”** output indicates the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

Sunshade

Sunshade actuator (FbSunshadeActuator)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbSunshadeActuator	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xUp		BOOL	Sunshade UP switch command
xDown		BOOL	Sunshade DOWN switch command
xSafety		BOOL	Signal input: Move to safety position
xLockPosition		BOOL	Lock position of the sunshade
typMaintenanceSunshade		typSunshade	Position command of the maintenance position
typSetSunshade		typSunshade	Position command of the manual override position
typAutomaticSunshade		typSunshade	Position command of the automatic positioning
xSetManualOverride		BOOL	Set the manual override
xResetManualOverride		BOOL	Reset the manual override
typConfigSunshade		typConfigSunshade	Configuration parameter of the sunshade actuator
.tTotalRunningTimeUp		TIME	Total running time UP move command Default setting: t#60s
.tTotalRunningTimeDown		TIME	Total running time DOWN command Default setting: t#60s
.tReverseIdleTime		TIME	Pause period when changing direction of movement Default setting: t#800ms
.tMechanicReverseTime		TIME	Compensation for the mechanical dead time Default setting: t#0s
.tTotalRunningTimeLamella		TIME	Total running time for lamella from 0% to 100% Default setting: t#1500ms
.tShortPushButton		TIME	Maximum time for a brief button press Default setting: t#500ms
.tTimeManualOverride		TIME	Time of manual override Default setting: t#60m
.bLamellaSteps		BYTE	Number of short button commands to control the lamella from 0% to 100% Default setting: 7

.bType	BYTE	Sunshade actuator type Default setting: 1
.xAutoMoveUp	BOOL	Ascend automatically if no position commands active Default setting: FALSE
Input/Output parameters:	Data type:	Comment:
typSunshadePosition	typSunshadePosition	Current position of the sunshade
.rPositionBlind	REAL	Height position of the sunshade [%]
.rPositionLamella	REAL	Slat position of the sunshade [%]
Return value:	Data type:	Comment:
xMoveUp	BOOL	Sunshade UP actuator command
xMoveDown	BOOL	Sunshade DOWN actuator command
xManualOverride	BOOL	Status output of the manual override
bPriority	BYTE	Output of the active priority
Graphical illustration:		
		
Function description:		
<p>The FbSunshadeActuator function block is used to control conventional sunshade motors. The sunshade is controlled based on depending on priority. Commands of higher priority override commands of lower priority. For commands of the same priority, the last command made is executed. The priorities listed in descending order are:</p> <ul style="list-style-type: none"> • 1 – Safety (“xSafety”) • 4 – Maintenance (“typMaintenanceSunshade”, “xLockPosition”) • 5 – Manual (“xUp” / “xDown”, “typSetPosition”) • 6 – Automatic (“typAutomaticSunshade”) 		

The sunshade is controlled manually by two button inputs **“xUp”** and **“xDown”**. An extended button press on one of these inputs causes the sunshade to move to the upper or lower position. A short button press triggers a STOP command or a command to adjust the slats.

The safety position (upper end position) of the sunshade (e.g. on a wind alarm) can be controlled via the **“xSafety”** input. When the sunshade has been moved to the safety position, the sunshade cannot be manually controlled by commands of lower priority until the **“xSafety”** input is set to FALSE.

The **“xLockPosition”** input can be used to interlock the sunshade. Motion commands are not canceled. Only the **“xSafety”** input can override the lock.

If there is a continuous signal TRUE at **“typMaintenanceSunshade.xMove”** input, the sunshade moves to the position specified at the **“typMaintenanceSunshade”** input and is then locked. This enables the sunshade to be moved to a defined cleaning or maintenance position, for example.

A rising edge at the **“typSetSunshade.xMove”** variable initiates a manual motion command to the position specified at the **“typSetSunshade”** input.

The **“typAutomaticSunshade”** input is used to move the sunshade to the automatic sunshade position (automatic sunshade function). As long as the **“xAutomaticPosition”** input signal is TRUE, the value changes from the automatic sunshade function input are updated.

The automatic sunshade function can be overridden. That means that commands are not evaluated via the **“typAutomaticSunshade”** input. The automatic sunshade function is overridden for the configured time **“typConfigSunshade.tTimeManualOverride”** if:

- A command was initiated via one of the **“xUp”** or **“xDown”** inputs.
- A position was approached via the **“xSetSunshade”** input.
- The **“xSetManualOverride”** input with signal TRUE is connected. It should be noted that the override time only elapses if the **“xSetManualOverride”** is switched to FALSE again. Thus, the automatic sunshade function can be overridden longer than the time set.

The reset of the override commands the sunshade to move to the position specified by the sunshade automatic.

The automatic sunshade function can be reset before the override time **“typConfigSunshade.tTimeManualOverride”** has elapsed. A TRUE signal at the **“xSafety”**, **“xLockPosition”**, **“typMaintenanceSunshade.xMove”** or **“xResetManualOverride”** input can trigger a reset.

After resetting the override, the sunshade moves to the last position specified by the automatic sunshade function. If there was never an automatic sunshade command, the sunshade moves up. The sunshade moves up if there is no previous automatic sunshade command and the parameter **“typConfigSunshade.xAutoMoveUp”** is set TRUE.

The **“xManualOverride”** output signals that the automatic sunshade function is overridden by manual commands. The output remains on TRUE for the manual override time.

The **"typConfigSunshade"** input contains configuration parameters of the sunshade actuator:

- **".tTotalRunningTimeUp"** specifies the total runtime for the sunshade UP command.
- **".tTotalRunningTimeDown"** specifies the total runtime for the sunshade DOWN command.
- **".tReverseldleTime"** specifies the pause period for switching directions. This must be set based on the motor.
- **".tMechanicReverseTime"** specifies the time allotted for compensation of the mechanical dead time. Due to tightening of the webbing, a dead time elapses until the slat first responds.
- **".tTotalRunningTimeLamella"** specifies the total runtime of the slat from the 0 to 100% position.
- **".tShortPushButton"** defines the maximum actuation time for a short button press. If the input is pressed longer, the input is interpreted as a long button press.
- **".tTimeManualOverride"** specifies the time for manually overriding the automatic function.
- **".bLamellaSteps"** specifies the number of short button commands required for moving the slat from 0 to 100% (Attention: This number is a function of the program cycle time!).
- **".bType"** defines the sunshade type and displays how the sunshade moves. The sunshade type is identified by the position of the slate per direction of motion:
 - **Type 1:** down closed / up open
- **".xAutoMoveUp"** determines whether the UP motion command should be sent in the case of a falling edge at the **"xManualOverride"** output, or whether the sunshade moves to the last automatic position. The same applies when the **"typAutomaticSunshade.xMove"** input signal switches to FALSE.

The **"xMoveUp"** and **"xMoveDown"** output signals the current direction of motion. The outputs can be connected to the motor control.

The **"bPriority"** output signals the active priority. The coding is as described above.

The **"typSunshadePosition"** input/output variable includes the current positions of the sunshade as feedback:

- **".rPositionBlind"** signals the position of the sunshade.
- **".rPositionLamella"** signals the slat position.

Notes:

- It is absolutely essential that the sunshade motor being operated is equipped with integrated limit switches.
- During voltage recovery, no move command can be triggered. The sunshade actuator maintains its current position.
- The current *typSunshadePosition* position values should be declared as RETAIN PERSISTENT, so that the last position moved to is retained even after a controller reset.
- The precision of positioning depends on the program cycle time. The lower the cycle time, the more precise the specified positions will be moved to.
- A minimum runtime of 2 s for the height position and 30 ms for the slat angle position are taken into account as the positioning hysteresis, meaning that smaller changes in the shadow position do not initiate a move command.
- The “*typConfigSunshade.tReverseIdleTime*” pause time for a reverse motion direction must be configured according to the type of motor used. Too short of a pause time can result in the hardware being destroyed. The minimum pause time is limited to 500 ms.
- After the manual override time “*typConfigSunshade.tTimeManualOverride*” the sunshade actuator moves to the automatic position. If there is no automatic command connected, the actuator moves up.
- The function block currently supports blind type 1 only (down closed/up open).

Safety Commands

Frost protection alarm (FbFrostAlarm)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbFrostAlarm	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xRain		BOOL	Input signal precipitation
rTemperature		REAL	Input signal temperature [°C]
typConfigFrostAlarm		typConfigFrostAlarm	Configuration parameters for the frost alarm
.rFrostTemperature		REAL	Threshold of the frost temperature [°C] Default setting: 0°C
.rDeiceTemperature		REAL	Threshold of the deice temperature [°C] Default setting: 4°C
.tDeicingTime		TIME	Deicing time of the sunshade Default setting: t#15m
Return value:		Data type:	Comment:
xFrostAlarm		BOOL	Signal output frost alarm
tDeiceTime		TIME	Elapsed deicing time
xError		BOOL	Error indicator in the event of incorrect parameterization
Graphical illustration:			
<div><div>FbFrostAlarm</div><div><div>xRain</div><div>rTemperature</div><div>typConfigFrostAlarm</div><div>xFrostAlarm</div><div>tDeiceTime</div><div>xError</div></div></div>			

Function description:

The **FbFrostAlarm** function block is used for weather protection by external sunshades against damage caused by icing. The time-related behavior is shown below as an example.

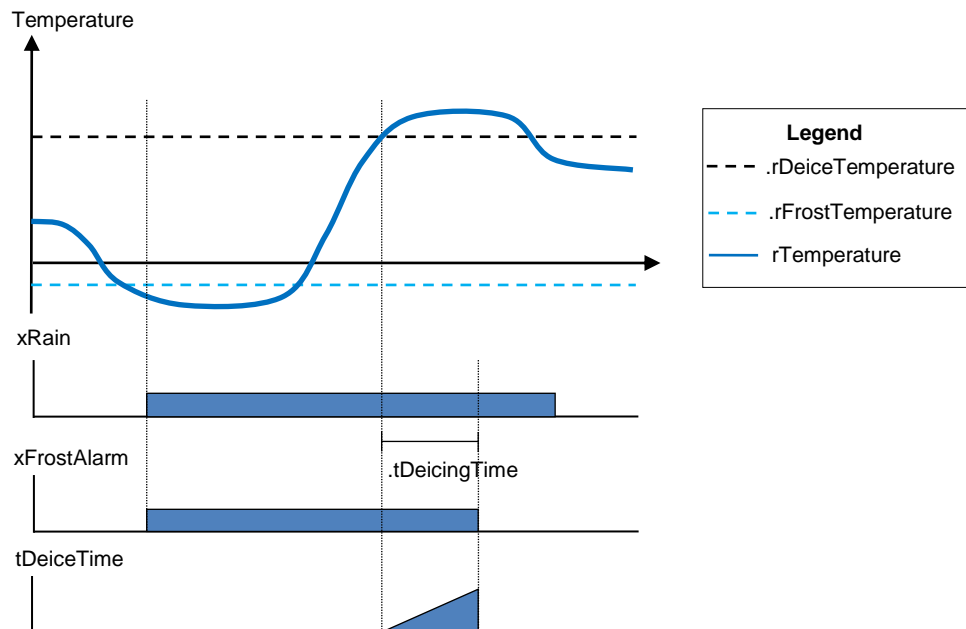
The sensor values received are connected to the **"xRain"** input for precipitation detection and **"rTemperature"** input for outside temperature.

The **"typConfigFrostAlarm"** input contains the configuration parameters for the frost alarm.

- **"rFrostTemperature"** defines the upper limit temperature of the outside air at which there is a risk of icing for the sunshade. If this value is overwritten by **"rTemperature"** and precipitation is displayed at **"xRain"**, the **"xFrostAlarm"** safety function is switched ON.
- **"rDeiceTemperature"** defines the lower limit temperature of the outside air at which the deicing process begins. The **"rDeiceTemperature"** limit temperature set must be greater than or equal to **"rFrostTemperature"**.
- **"tDeicingTime"** specifies the deicing time of the sunshade after exceeding the **"rDeiceTemperature"** limit temperature. During this time, the sunshade is deemed locked, so that the safety function remains active.

The **"xFrostAlarm"** output switches on when it falls below the frost temperature and precipitation is detected. When the deicing temperature is reached, the deicing time starts, which is displayed at the **"tDeiceTime"** output. Once the deicing time has elapsed, the **"xFrostAlarm"** output is reset.

The **"xError"** output signals incorrect parameterization of the function block thresholds.

Chronological sequence:

Wind Alarm (FbWindAlarm)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWindAlarm	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
rWindVelocity		REAL	Input signal wind velocity [m/s]
typConfigWindAlarm		typConfigWindAlarm	Configuration parameters for the wind alarm
.rWindThreshold		REAL	Threshold wind [m/s] Default setting: 5 m/s
.rSquallThreshold		REAL	Threshold squall [m/s] Default setting: 8 m/s
.tDelayOnWind		TIME	Switch-on delay wind alarm Default setting: t#5s
.tDelayOffWind		TIME	Switch-off delay wind alarm Default setting: t#15m
.tTimeOutWind		TIME	Time-out time of the wind sensor for wind standstill Default setting: t#48h Deactivation at t#0ms
.tTimeOutSensor		TIME	Time-out time of the wind sensor for faulty sensor Default setting: t#48h Deactivation at t#0ms
Return value:		Data type:	Comment:
xWindAlarm		BOOL	Signal output wind alarm
tWindThresholdTime		BOOL	Elapsed time before wind alarm drop-out
xError		BOOL	Error indicator in the event of incorrect parameterization
Graphical illustration:			
<div><div>FbWindAlarm</div><div><div>rWindVelocity</div><div>xWindAlarm</div><div>typConfigWindAlarm</div><div>tWindThresholdTime</div><div>xError</div></div></div>			

Function description:

The **FbWindAlarm** function block is used for weather protection by external sunshades against damage caused by wind. The time-related behavior is shown below as an example.

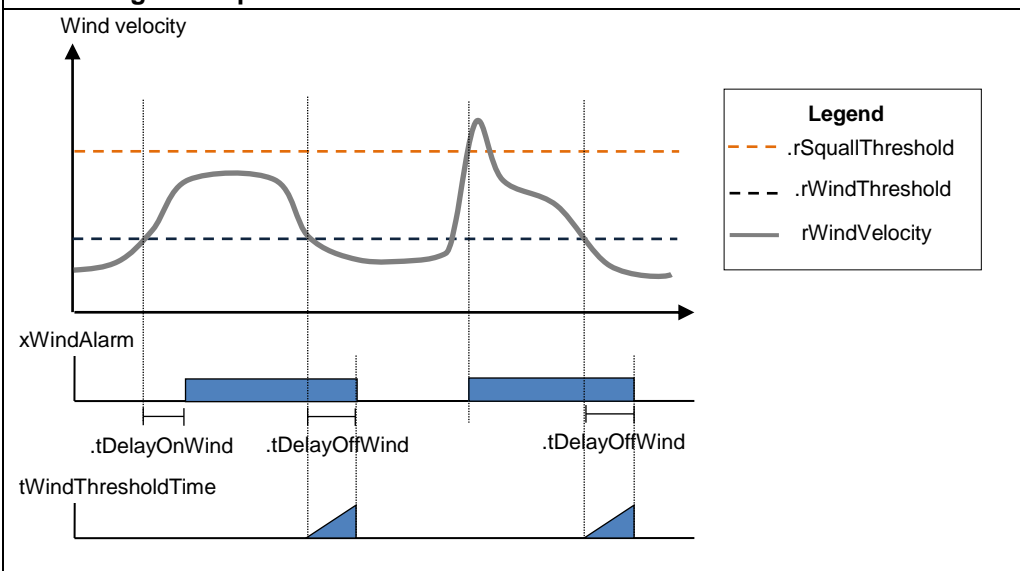
The measured wind velocity is connected to the **"rWindVelocity"** input.

The **"typConfigWindAlarm"** input contains the configuration parameters for the wind alarm.

- **"rWindThreshold"** defines the lower limiting value of the continuous wind velocity that is not permitted. If the **"rWindVelocity"** wind velocity exceeds the limiting value for the duration **"tDelayOnWind"**, the **"xWindAlarm"** safety function is switched ON.
- **"rSquallThreshold"** defines the lower limiting value of the short-term wind velocity (wind gusts) that is not permitted. If the **"rWindVelocity"** wind velocity exceeds the limiting value, the **"xWindAlarm"** safety function is switched ON without delay. The set **"rSquallThreshold"** limiting value must be greater than **"rWindThreshold"**.
- **"tDelayOnWind"** is the switch-on delay for the safety function if the permitted wind velocity is continuously exceeded.
- **"tDelayOffWind"** is the switch-off delay if the safety function has been triggered by wind.
- **"tTimeOutWind"** defines the time-out time of the wind sensor for wind standstill. If the wind sensor times out, the **"xError"** output is switched to TRUE. The function can be disabled by setting **"tTimeOutWind"** to t#0ms.
- **"tTimeOutSensor"** defines the time-out time of the wind sensor for faulty sensor. If the wind sensor times out, the **"xError"** output is switched to TRUE. The function can be disabled by setting **"tTimeOutWind"** to t#0ms.

The **"xWindAlarm"** switches on in the event of strong wind or a storm. If the limiting value is not reached, the **"tWindThresholdTime"** switch-off delay starts. Once the switch-off delay has elapsed, the **"xWindAlarm"** output is reset.

The **"xError"** output signals a time-out of the wind sensor or incorrect parameterization of the function block thresholds.

Chronological sequence of the wind alarm:

Weather protection (FbBasicWeatherProtection)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbBasicWeatherProtection	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
rWindVelocity		REAL	Input signal wind velocity [m/s]
rTemperature		REAL	Input signal temperature [°C]
xRain		BOOL	Input signal precipitation
typConfigWeatherProtection		typConfigWeatherProtection	Configuration parameters for the weather protection
.typConfigFrostAlarm		typConfigFrostAlarm	Configuration parameters for the frost alarm
.rFrostTemperature		REAL	Threshold of the frost temperature [°C] Default setting: 0°C
.rDeiceTemperature		REAL	Threshold of the deice temperature [°C] Default setting: 4°C
.tDeicingTime		TIME	Deicing time of the sunshade Default setting: t#15m
.typConfigWindAlarm		typConfigWindAlarm	Configuration parameters for the wind alarm
.rWindThreshold		REAL	Threshold wind [m/s] Default setting: 5 m/s
.rSquallThreshold		REAL	Threshold squall [m/s] Default setting: 8 m/s
.tDelayOnWind		TIME	Switch-on delay wind alarm Default setting: t#5s
.tDelayOffWind		TIME	Switch-off delay wind alarm Default setting: t#15m
.tTimeOutWind		TIME	Time-out time of the wind sensor Default setting: t#48h Deactivation at t#0ms
Return value:		Data type:	Comment:
xSafety		BOOL	Signal output safety mode
xError		BOOL	Error indicator in the event of incorrect parameterization
Graphical illustration:			
<div><div><div>FbBasicWeatherProtection</div><div><div>rWindVelocity</div><div>rTemperature</div><div>xRain</div><div>typConfigWeatherProtection</div></div><div><div>xSafety</div><div>xError</div></div></div></div>			

Function description:

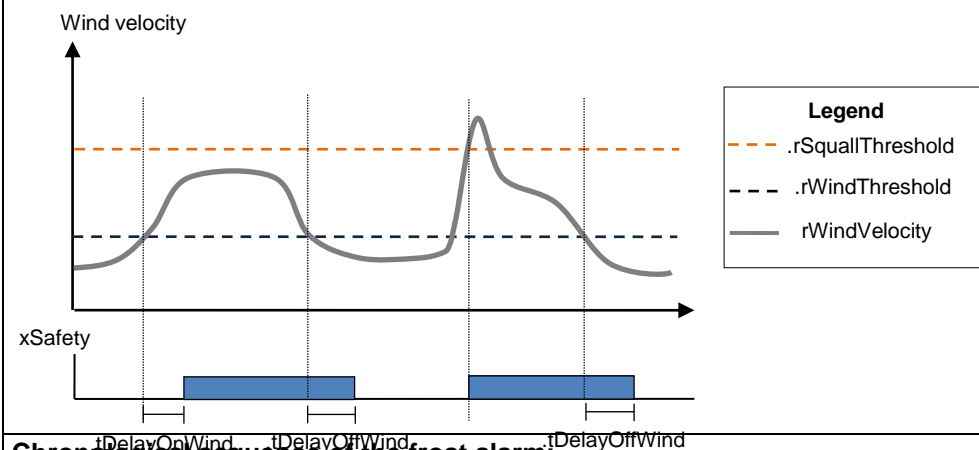
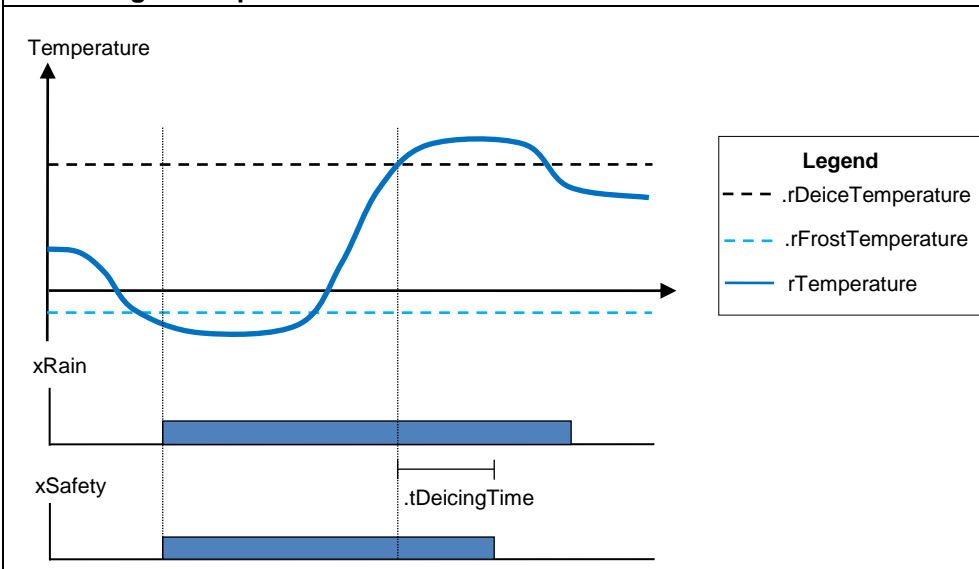
The **FbBasicWeatherProtection** function block is used for weather protection by external sunshades against damage caused by wind, rain or icing. The detected sensor values of wind velocity, outside temperature and precipitation detection are evaluated and trigger the safety function of the sunshade actuator when there is a risk of damage. The time-related behavior for the wind alarm and frost alarm is shown below as an example.

The FbBasicWeatherProtection function block combines the functions of the [FbFrostAlarm](#) and [FbWindAlarm](#). The explanation of the inputs can be found in the above descriptions.

The configuration parameters at the **“typConfigWeatherProtection”** input also come from the FbFrostAlarm and FbWindAlarm function blocks.

The **“xSafety”** output signals when the safety function of the sunshade is switched on. The output can be connected to the signal output for the safety position of the sunshade actuator.

The **“xError”** output signals a time-out of the wind sensor or incorrect parameterization of the function block thresholds.

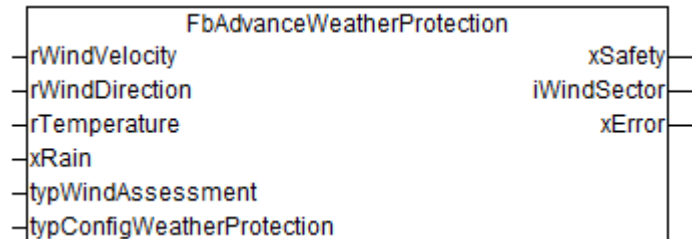
Chronological sequence of the wind alarm:**Chronological sequence of the frost alarm:**

Weather Protection With Weather Assessment (FbAdvanceWeatherProtection)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbAdvanceWeatherProtection	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
rWindVelocity	REAL	Input signal wind velocity [m/s]
rWindDirection	REAL	Input signal wind direction [°]
rTemperature	REAL	Input signal temperature [°C]
xRain	BOOL	Input signal precipitation
typWindAssessment	typWindAssessment	Configuration parameters for the wind direction
.arWindFactor	ARRAY [1..12] OF REAL	Wind factors for the sector Default setting of all sectors: 1
.arWindDirection	ARRAY [1..12] OF REAL	Minimum wind direction for the sector [°] Default setting of the fields: 0°, 30°, 60°, 90°, 120°, 150°, 180°, 210°, 240°, 270°, 300°, 330°
typConfigWeatherProtection	typConfigWeatherProtection	Configuration parameters for the weather protection
.typConfigFrostAlarm	typConfigFrostAlarm	Configuration parameters for the frost alarm
.rFrostTemperature	REAL	Threshold of the frost temperature [°C] Default setting: 0°C
.rDeiceTemperature	REAL	Threshold of the deice temperature [°C] Default setting: 4°C
.tDeicingTime	TIME	Deicing time of the sunshade Default setting: t#15m
.typConfigWindAlarm	typConfigWindAlarm	Configuration parameters for the wind alarm
.rWindThreshold	REAL	Threshold wind [m/s] Default setting: 5 m/s
.rSquallThreshold	REAL	Threshold squall [m/s] Default setting: 8 m/s
.tDelayOnWind	TIME	Switch-on delay wind alarm Default setting: t#5s
.tDelayOffWind	TIME	Switch-off delay wind alarm Default setting: t#15m
.tTimeOutWind	TIME	Time-out time of the wind sensor Default setting: t#48h Deactivation at t#0ms

Return value:	Data type:	Comment:
xSafety	BOOL	Signal output safety mode
iWindSector	INT	Active wind sector
xError	BOOL	Error indicator in the event of incorrect parameterization

Graphical illustration:



Function description:

The **FbAdvanceWeatherProtection** function block is used for weather protection by external sunshades against damage caused by wind, rain or icing. The detected sensor values of wind velocity, outside temperature and precipitation detection are evaluated and trigger the safety function of the sunshade actuator when there is a risk of damage.

The FbAdvanceWeatherProtection function block is based on the [FbBasicWeatherProtection](#) function block. The explanation of the inputs can be found in the above description.

The measured wind direction “**rWindDirection**” is connected to the function block. The wind direction is assigned with a wind sector. The measured wind strength is multiplied by the factor of the wind sector and the calculated wind strength further processed.

The “**typWindAssessment**” input contains the configuration parameters for the wind assessment. The classification of wind sectors is illustrated in the figure below. 12 wind sectors can be defined.

- “**arWindFactor**” defines the factor of the measured wind strength for the individual wind sectors.
- “**arWindDirection**” defines the lower limiting angle of the individual wind sectors.

The measured wind direction “**rWindDirection**” is connected to the function block. The wind direction is assigned with a wind sector. The measured wind strength is multiplied by the factor of the wind sector and the calculated wind strength further processed.

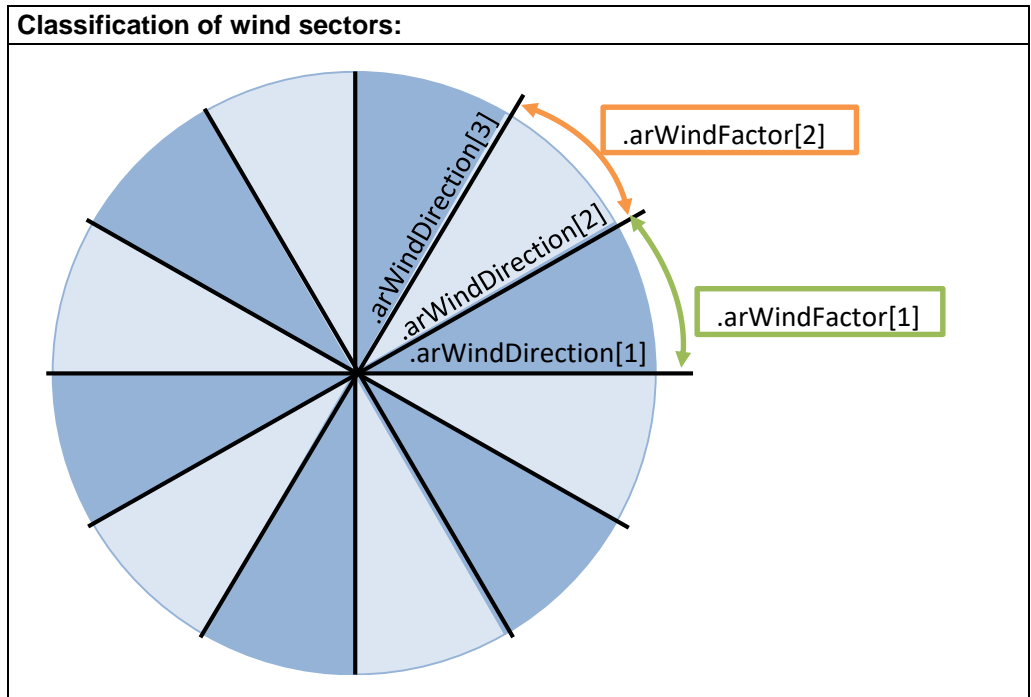
The “**xSafety**” output signals when the safety function of the sunshade is switched on. The output can be connected to the signal output for the safety position of the sunshade actuator.

The “**iWindSector**” output signals the active wind sector.

The “**xError**” output signals a time-out of the wind sensor or incorrect parameterization of the function block thresholds.

Notice:

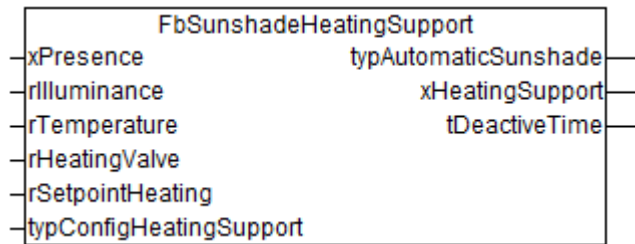
This function block should only be used together with one wind assessment.



Automatic Thermal Control

Sunshade as heating support (FbSunshadeHeatingSupport)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbSunshadeHeatingSupport	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xPresence	BOOL	Presence detection
rIlluminance	REAL	Measured intensity of the daylight [lx]
rTemperature	REAL	Measured room temperature [°C]
rHeatingValve	REAL	Position of the heating valve [%]
rSetpointHeating	REAL	Controller setpoint for heating support [°C] Default setting: 18°C
typConfigHeatingSupport	typConfigHeatingSupport	Configuration of heating support
.rIlluminationLimit	REAL	Brightness limiting value for activation of heating support [lx] Default setting: 20000 lx
.rOpenValve	REAL	Valve opening value for activation of heating support [%] Default setting: 15%
.tTimeHysteresisIllumination	TIME	Time hysteresis of the light intensity Default setting: t#15m
.tDelayRestartHeating	TIME	Time delay for restarting the heating support after its deactivation Default setting: t#1h
.wPositionBlind	WORD	Height position of the sunshade for heating support [%] Default setting: 0%
.wPositionLamella	WORD	Slat position of the sunshade for heating support [%] Default setting: 0%
Return value:	Data type:	Comment:
typAutomaticSunshade	typSunshade	Position command of the automatic position
xHeatingSupport	BOOL	Heating Support active
tDeactiveTime	TIME	Elapsed Time of heating support inactivity

Graphical illustration:**Function description:**

The **FbSunshadeHeatingSupport** function block is used to support heating by the sunshade. Solar thermal energy is allowed specifically in unoccupied rooms to reduce the heat energy expended.

Heating support is activated by a low room temperature or open heating valve. The heating support keeps permanently inactive for a parametrable time after its deactivation.

The **“xPresence”** input is connected to the presence detection. If presence is detected, heating support is disabled. A change in the presence detection leads to undelayed switching behavior.

The measured intensity of the natural light is applied to the **“rIlluminance”** input. The light intensity must exceed the configured limiting value continuously, so that heating support is activated. If the light intensity is low, heating support is disabled.

The measured room temperature is applied to the **“rTemperature”** input.

The **“rSetpointHeating”** defines the threshold of the room temperature at which a position command is transmitted to the sunshade for heating support.

The **“rHeatingValve”** input is assigned the percentage opening of the heating valve. By opening the heating valve, the position command to the sunshade for heating support can be triggered.

The **“typConfigHeatingSupport”** input contains the configuration parameters for heating support.

- **“rIlluminationLimit”** defines the threshold of the light intensity for activation of heating support.
- **“rOpenValve”** defines the threshold from with the valve is opened. If the valve is open, heating support is activated.
- **“tTimeHysteresisIllumination”** specifies the time hysteresis for the measured light intensity. The **“rIlluminationLimit”** threshold must be exceeded or not exceeded for the duration **“tTimeHysteresisIllumination”** to enable or disable heating support.
- **“tDelayRestartHeating”** delays the reactivation of the heating support after deactivation to prevent frequently positioning.
- **“wPositionBlind”** switches the position height of the sunshade for heating support ON.
- **“wPositionLamella”** switches the slat position of the sunshade for heating support ON.

The ***“xHeatingSupport”*** output displays the heating support activity. The ***“typAutomaticSunshade”*** output variables contain the position commands for the sunshade actuator. The ***“tDeactiveTime”*** output displays the elapsed time after deactivation of the heating support. If the ***“tDeactiveTime”*** value reaches the parametrable time ***„typConfigHeatingSupport.tDelayRestartHeating“***, the heating support can be reactivated.

Note:

The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.

Sunshade as Cooling Support (FbSunshadeCoolingSupport)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbSunshadeCoolingSupport	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xPresence		BOOL	Presence detection
rIlluminance		REAL	Measured intensity of the daylight [lx]
rTemperature		REAL	Measured room temperature [°C]
rCoolingValve		REAL	Position of the cooling valve [%]
rSetpointCooling		REAL	Controller setpoint for cooling support [°C] Default setting: 22°C
typConfigCoolingSupport		typConfigCoolingSupport	Configuration of cooling support
.rIlluminationLimit		REAL	Brightness limiting value for activation of cooling support [lx] Default setting: 20000 lx
.rOpenValve		REAL	Valve opening value for activation of cooling support [%] Default setting: 15%
.tTimeHysteresisIllumination		TIME	Time hysteresis of the light intensity Default setting: t#15m
.tDelayRestartHeating		TIME	Time delay for restarting the heating support after its deactivation Default setting: t#1h
.wPositionBlind		WORD	Height position of the sunshade for cooling support [%] Default setting: 100%
.wPositionLamella		WORD	Slat position of the sunshade for cooling support [%] Default setting: 100%
Return value:		Data type:	Comment:
typAutomaticSunshade		typSunshade	Position command of the automatic position
xCoolingSupport		BOOL	Kühlunterstützung aktiv
tDeactiveTime		TIME	Elapsed Time of cooling support inactivity

Graphical illustration:**Function description:**

The **FbSunshadeCoolingSupport** function block is used to support cooling by the sunshade. Solar thermal energy is prevented specifically in unoccupied rooms to reduce the energy expended.

Cooling support is activated by a high room temperature or open cooling valve. The cooling support keeps permanently inactive for a parametrable time after its deactivation.

The **“xPresence”** input is connected to the presence detection. If presence is detected, cooling support is disabled. A change in the presence detection leads to undelayed switching behavior.

The measured intensity of the natural light is applied to the **“rIlluminance”** input. The light intensity must exceed the configured limiting value continuously, so that cooling support is activated. If the light intensity is low, cooling support is disabled.

The measured room temperature is applied to the **“rTemperature”** input.

The **“rSetpointCooling”** defines the threshold of the room temperature at which a position command is transmitted to the sunshade for cooling support.

The **“rCoolingValve”** input is assigned the percentage opening of the cooling valve. By opening the cooling valve, the position command to the sunshade for cooling support can be triggered.

The **“typConfigCoolingSupport”** input contains the configuration parameters for cooling support.

- **“rIlluminationLimit”** defines the threshold of the light intensity for activation of cooling support.
- **“rOpenValve”** defines the threshold from with the valve is opened. If the valve is open, cooling support is activated.
- **“tTimeHysteresisIllumination”** specifies the time hysteresis for the measured light intensity. The **“rIlluminationLimit”** threshold must be exceeded or not exceeded for the duration **“tTimeHysteresisIllumination”** to enable or disable cooling support.
- **“tDelayRestartHeating”** delays the reactivation of the heating support after deactivation to prevent frequently positioning.
- **“wPositionBlind”** switches the position height of the sunshade for cooling support ON.
- **“wPositionLamella”** switches the slat position of the sunshade for cooling support ON.

The “**xCoolingSupport**” output displays the cooling support activity. The “**typAutomaticSunshade**” output variables contain the position commands for the sunshade actuator. The “**tDeactiveTime**” output displays the elapsed time after deactivation of the cooling support. If the “**tDeactiveTime**” value reaches the parametrable time „**typConfigCoolingSupport.tDelayRestartCooling**“, the cooling support can be reactivated.

Note:

The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.

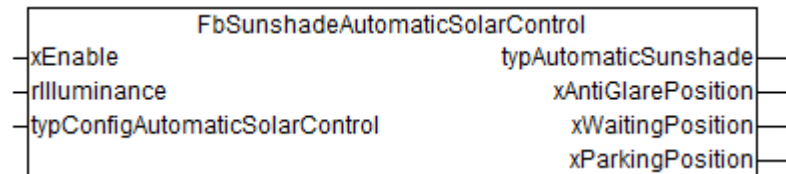
Automatic Commands

Automatic sunlight sensor (FbSunshadeAutomaticSolarControl)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbSunshadeAutomaticSolarControl	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xEnable	BOOL	Activate / deactivate automatic solar control Default setting: TRUE (executing function)
rIlluminance	REAL	Measured intensity of the daylight [lx]
typConfigAutomaticSolarControl	typConfigAutomaticSolarControl	Configuration parameters for automatic solar control
.rAntiGlareLimit	REAL	Brightness threshold for activation of the antiglare position [lx] Default setting: 10000 lx
.rParkingLimit	REAL	Brightness threshold for activation of the parking position [lx] Default setting: 1000 lx
.tDelayOnIlluminance	TIME	Switch-on delay automatic solar control Default setting: t#5s
.tDelayOffIlluminance	TIME	Switch-off delay automatic solar control Default setting: t#5m
.wAntiGlarePositionBlind	WORD	Height position of the antiglare position [%] Default setting: 100%
.wAntiGlarePositionLamella	WORD	Slat position of the antiglare position [%] Default setting: 50%
.wWaitingPositionBlind	WORD	Height position of the waiting position [%] Default setting: 100%
.wWaitingPositionLamella	WORD	Slat position of the waiting position [%] Default setting: 0%
.wParkingPositionBlind	WORD	Height position of the parking position [%] Default setting: 0%
.wParkingPositionLamella	WORD	Slat position of the parking position [%] Default setting: 0%

Return value:	Data type:	Comment:
typAutomaticSunshade	typSunshade	Position command of the automatic position
xAntiGlarePosition	BOOL	Antiglare position active
xWaitingPosition	BOOL	Waiting position active
xParkingPosition	BOOL	Parking position active

Graphical illustration:



Function description:

The **FbSunshadeAutomaticSolarControl** function block maps the function of the automatic solar control. The automatic solar control prevents the impact to the user by incoming sunrays of high intensity by moving the sunshade to a defined antiglare position once the natural light exceeds a defined intensity. With decreasing brightness, a waiting position is taken. If the brightness falls below a defined intensity, a parking position is taken.

The “**xEnable**” input can be used to disable automatic solar control. The information can be come from the time program or a command from the building management system.

The current measured intensity of the natural light is connected the “**rIlluminance**” input.

The parameters are configured using the “**typConfigAutomaticSolarControl**” input:

- “**.rAntiGlareLimit**” defines the brightness threshold for positioning the antiglare position.
- “**.rParkingLimit**” defines the brightness threshold for positioning the parking position.
- “**.tDelayOnIlluminance**” sets a switch-on delay for the anti glare position to prevent a positioning operation after brief brightness variations. The activation criteria must be met without interruption during the entire time interval to trigger a positioning operation.
- “**.tDelayOffIlluminance**” sets a switch-off delay for the waiting and parking position to prevent a positioning operation after brief brightness variations. The deactivation criteria must be met without interruption during the entire time interval to trigger a positioning operation.
- “**.wAntiGlarePositionBlind**” switches the position height of the sunshade for antiglare support ON.
- “**.wAntiGlarePositionLamella**” switches the slat position of the sunshade for antiglare support ON.
- “**.wWaitingPositionBlind**” switches the position height of the sunshade for waiting position ON.
- “**.wWaitingPositionLamella**” switches the slat position of the sunshade for waiting position ON.

- **“*wParkingPositionBlind*”** switches the position height of the sunshade for the parking position ON.
- **“*wParkingPositionLamella*”** switches the slat position of the sunshade for the parking position ON.

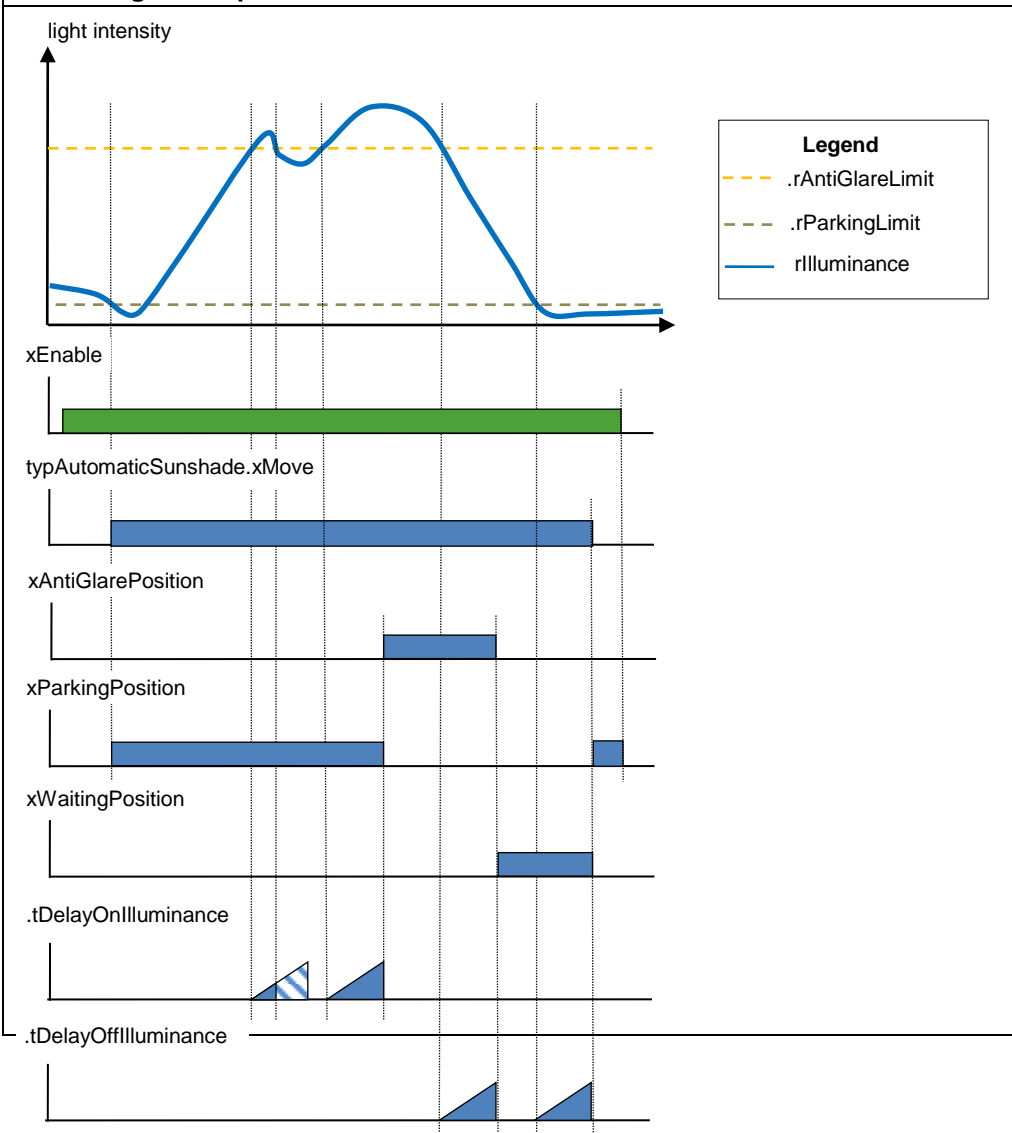
The **“*xAntiGlarePosition*”**, **“*xWaitingPosition*”** and **“*xParkingPosition*”** outputs display the active position. The position commands of the antiglare and waiting position are assigned permanently. The position command of the parking position is assigned temporary with a rising edge.

The **“*typAutomaticSunshade*”** output variable contains the position command for the sunshade actuator.

Note:

The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.

Chronological sequence:



Automatic Twilight Control (FbSunshadeAutomaticTwilightControl)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbSunshadeAutomaticTwilightControl	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xEnable		BOOL	Activate / deactivate automatic solar control Default setting: TRUE (executing function)
rIlluminance		REAL	Measured intensity of the daylight [lx]
typConfigAutomaticTwilightControl		typConfigAutomaticSolarControl	Configuration parameters for the automatic twilight control
.rTwilightLimit		REAL	Brightness threshold for activation of the dimming position [lx] Default setting: 50 lx
.rSunriseLimit		REAL	Brightness threshold for activation of the sunrise position [lx] Default setting: 100 lx
.tTimeHysteresis		TIME	Time hysteresis of the automatic twilight control Default setting: t#5m
.wTwilightPositionBlind		WORD	Height position at twilight [%] Default setting: 100%
.wTwilightPositionLamella		WORD	Slat position at twilight [%] Default setting: 100%
.wSunrisePositionBlind		WORD	Height position at sunrise [%] Default setting: 0%
.wSunrisePositionLamella		WORD	Slat position at sunrise [%] Default: 0 %
Return value:		Data type:	Comment:
typAutomaticSunshade		typSunshade	Position command of the automatic position
xTwilightPosition		BOOL	Twilight position active
xSunrisePosition		BOOL	Sunrise position active
Graphical illustration:			
<div><div>FbSunshadeAutomaticTwilightControl</div><div><div>xEnable</div><div>rIlluminance</div><div>typConfigAutomaticTwilightControl</div><div>typAutomaticSunshade</div><div>xTwilightPosition</div><div>xSunrisePosition</div></div></div>			

Function description:

The **FbSunshadeAutomaticTwilightControl** function block maps the function of the automatic twilight control. The automatic twilight control can be used to position sunshades based on outdoor brightness. The automatic control makes it possible, for example, to close the sunshade during the night to reduce heat loss through the windows or to reduce light emissions.

The **“xEnable”** input can be used to disable automatic twilight control. The information can be come from the time program or a command from the building management system.

The current measured intensity of the natural light is applied to the **“rIlluminance”** input.

The parameters are configured using the **“typConfigAutomaticTwilightControl”** input:

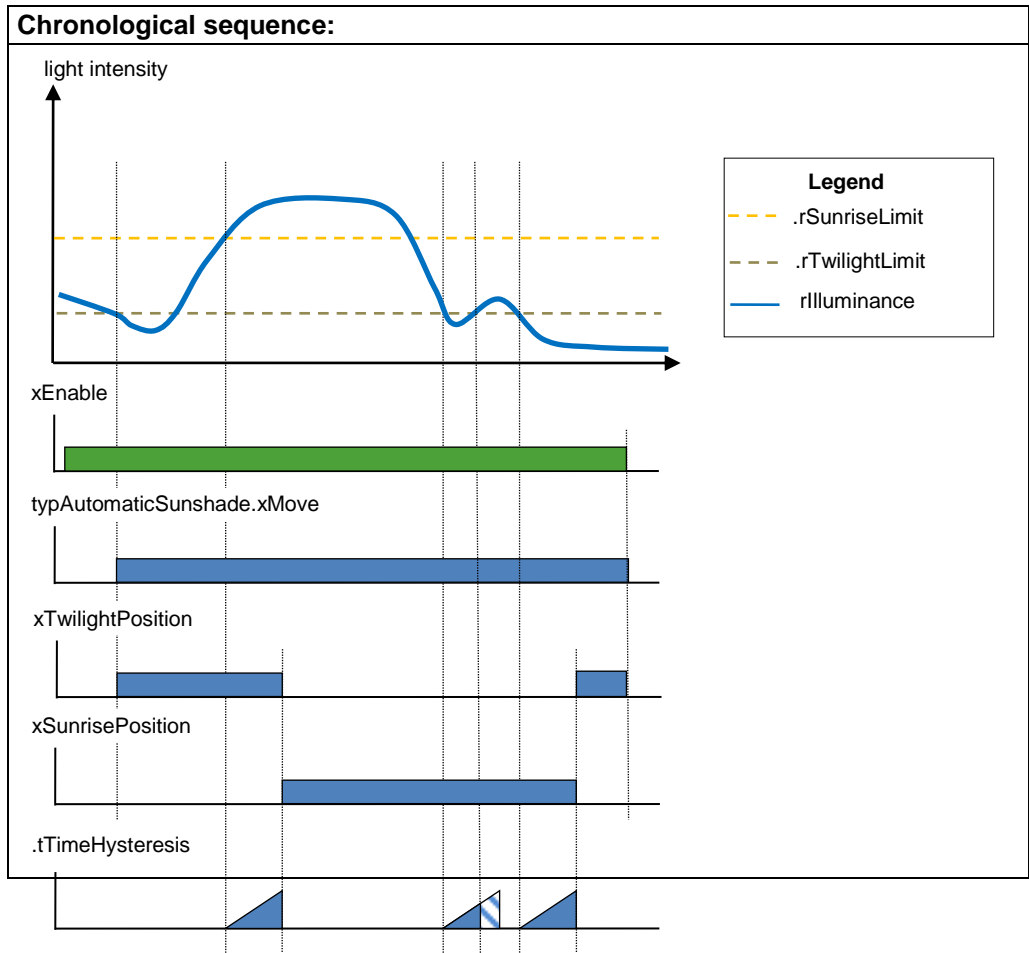
- **“rTwilightLimit”** defines the lighting limiting value for activation of twilight positioning.
- **“rSunriseLimit”** defines the lighting limiting value for activation of sunrise positioning.
- **“tTimeHysteresis”** sets a time hysteresis to prevent an accidental positioning operation after brief brightness variations. The activation and deactivation criteria must be met without interruption during the entire time interval to trigger a positioning operation.
- **“.wTwilightPositionBlind”** switches the position height of the sunshade for the activation position ON.
- **“.wTwilightPositionLamella”** switches the slat position of the sunshade for the activation position ON.
- **“.wSunrisePositionBlind”** switches the position height of the sunshade for the deactivation position ON.
- **“.wSunrisePositionLamella”** switches the slat position of the sunshade for the deactivation position ON.

The **“xTwilightPosition”** and **“xSunrisePosition”** outputs display the active position. The position command of the twilight position is assigned permanently. The position command of the sunrise position is assigned temporary with a rising edge.

The **“typAutomaticSunshade”** output variable contains the position commands for the sunshade actuator.

Note:

The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.



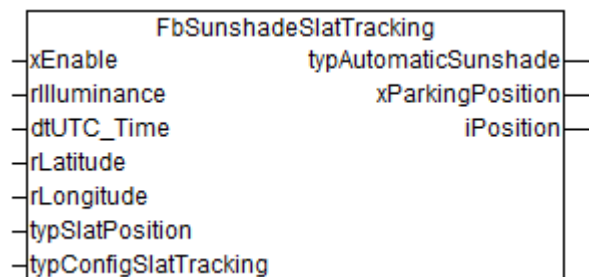
Slat Tracking (FbSunshadeSlatTracking)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbSunshadeSlatTracking	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xEnable		BOOL	Activate / deactivate automatic solar control Default setting: TRUE (executing function)
rIlluminance		REAL	Measured intensity of the daylight [lx]
dtUTC_Time		DT	Coordinated Universal Time UTC
rLatitude		REAL	Latitude Default setting: 52,305
rLongitude		REAL	Longitude Default setting: 8,922
typSlatPosition		typSlatPosition	Configuration parameters of the positioning of the slat tracking
.awPositionBlind		ARRAY [1..7] OF WORD	Height positions of the tracking [%] Default setting: all 100%
.awPositionLamella		ARRAY [1..7] OF WORD	Slat positions of the tracking [%] Default setting: 100%, 85%, 68%, 51%, 34%, 17%, 0%
.arElevation		ARRAY [0..7] OF REAL	Limits of the elevation angle of the sun [°] Default setting: 0°, 13°, 26°, 39°, 52°, 65°, 78°, 90°
.rMinAzimuth		REAL	Minimum azimuth angle of the sun [°] North = 0° Default setting: 90°
.rMaxAzimuth		REAL	Maximum azimuth angle of the sun [°] North = 0° Default setting: 270°
typConfigSlatTracking		typConfigSlatTracking	Configuration parameters of the slat tracking
.rSlatTrackingLimit		REAL	Brightness limiting value for activation of the slat tracking [lx] Default setting: 10000 lx
.rParkingLimit		REAL	Brightness limiting value for deactivation of the slat tracking [lx] Default setting: 1000 lx
.tDelayOnIlluminance		TIME	Switch-on delay of the slat tracking Default setting: t#5s
.tDelayOffIlluminance		TIME	Switch-off delay of the slat tracking Default setting: t#5m

.wParkingPositionBlend	WORD	Height position of the parking position [%] Default setting: 0%
.wParkingPositionLamella	WORD	Slat position of the parking position [%] Default setting: 0%

Return value:	Data type:	Comment:
typAutomaticSunshade	typSunshade	Position command of the automatic position
xParkingPosition	BOOL	Twilight position active
iPosition	INT	Sunrise position active

Graphical illustration:



Function description:

The **FbSunshadeSlatTracking** function block maps the function of the slat tracking. Like the automatic solar control, the slat tracking prevents the impact to the user by incoming sunrays of high intensity by moving to an antiglare position. In contrast to the automatic solar control, however, the position of the slats is adjusted to the current position of the sun when light intensity is high. In this way, every room receives optimal sunlight while preventing direct sunlight. With decreasing brightness or with the sun out of parameterized angle limits, a parking position is taken.

The “**xEnable**” input can be used to disable automatic solar control. The information can be come from the time program or a command from the building management system.

The measured intensity of the natural light is applied to the “**rIlluminance**” input.

The “**dtUTC_Time**”, “**rLatitude**” and “**rLongitude**” inputs are used to calculate the position of the sun. The elevation angle and azimuth angle are calculated and evaluated. The “**rLatitude**” and “**rLongitude**” coordinates are input as following:

Value := value in degrees + (value in minutes / 60) + (value in seconds / 3600)

Example 8° 55' 19" O: “**rLongitude**” := 8 + 55 / 60 + 19 / 3600 = 8.9219

The facade-dependent parameters are configured using the **“typSlatPosition”** input:

- **“.awPositionBlind”** sets the position height of the sunshade for the individual antiglare positions of the slat tracking.
- **“.awPositionLamella”** sets the slat position of the sunshade for the individual antiglare positions of the slat tracking.
- **“.arElevation”** defines the elevation angle for the individual antiglare positions of the slat tracking.
- **“.rMinAzimuth”** defines the minimum azimuth angle of the sun for activation of the slat tracking.
- **“.rMaxAzimuth”** defines the maximum azimuth angle of the sun for activation of the slat tracking.

The general parameters are configured using the **“typConfigSlatTracking”** input:

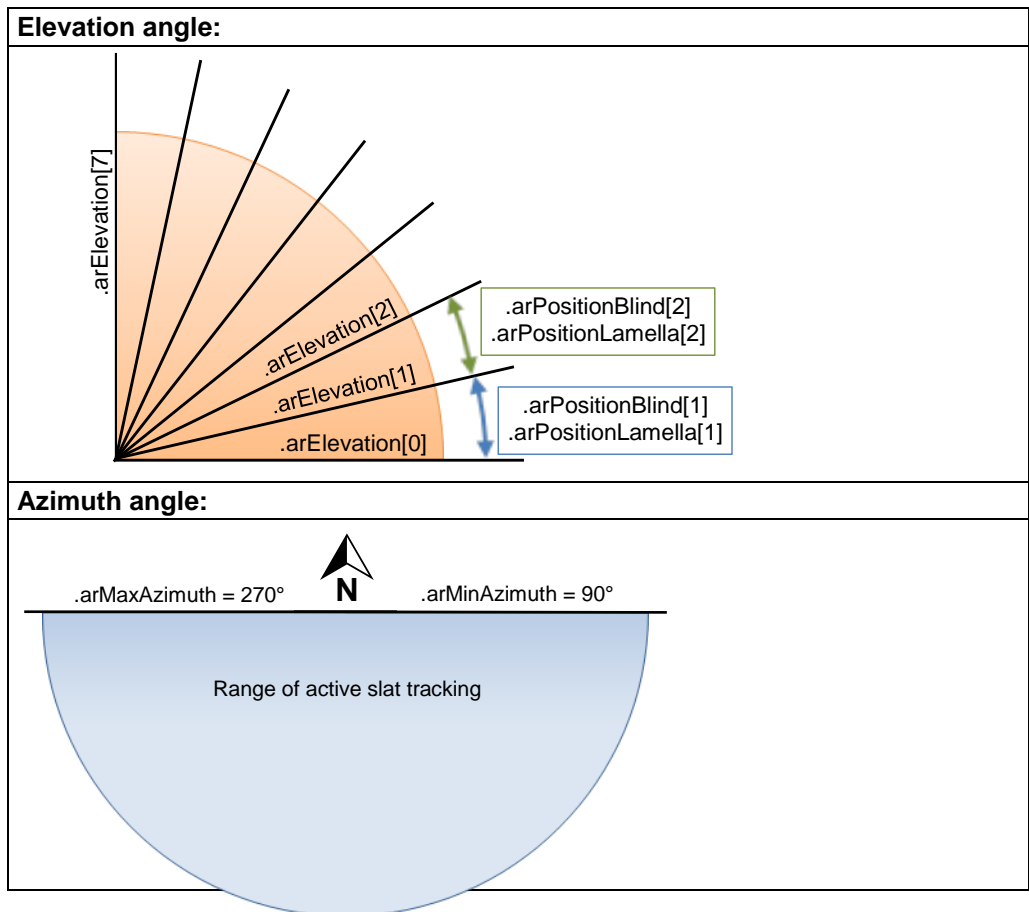
- **“.rSlatTrackingLimit”** defines the minimum light intensity for positioning the antiglare position.
- **“.rParkingLimit”** defines the maximum light intensity for positioning the parking position.
- **“.tDelayOnIlluminance”** sets a switch-on delay for the slat tracking to prevent a positioning operation after brief brightness variations. The activation criteria must be met without interruption during the entire time interval to trigger a positioning operation.
- **“.tDelayOffIlluminance”** sets a switch-off delay for the parking position to prevent a positioning operation after brief brightness variations. The deactivation criteria must be met without interruption during the entire time interval to trigger a positioning operation.
- **“.tTimeHysteresis”** sets a time hysteresis to prevent a positioning operation after brief brightness variations. The activation and deactivation criteria must be met without interruption during the entire time interval to trigger a positioning operation.
- **“.wParkingPositionBlind”** switches the position height of the sunshade for the parking position ON.
- **“.wParkingPositionLamella”** switches the slat position of the sunshade for the parking position ON.

The **“,iPosition”** and **“,xParkingPosition”** outputs display the active position. The position commands of the slat tracking position are assigned permanently. The position command of the parking position is assigned temporary with a rising edge.

The **“typAutomaticSunshade”** output variable contains the position commands for the sunshade actuator.

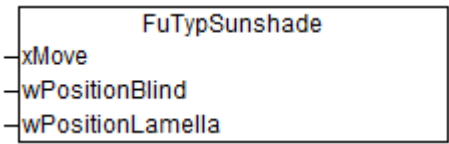
Note:

The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.

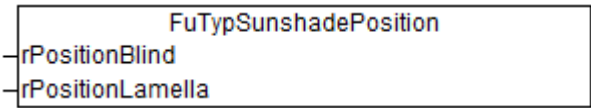


Other Functions

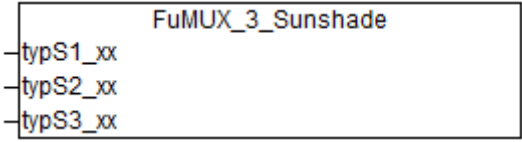
Position command transfer for sunshade (FuTypSunshade)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FuTypSunshade	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xMove	BOOL	Move to position
wPositionBlind	WORD	Height position of the sunshade [%] 0% = Upper end position 100% = Lower end position
wPositionLamella	WORD	Slat position of the sunshade [%] 0% = Slat open 100% = Slat closed
Return value:	Data type:	Comment:
	typSunshade	Position parameter
Graphical illustration:		
		
Function description:		
<p>The FuTypSunshade function is used to transfer the sunshade position command to the sunshade actuator.</p> <p>The “xMove” input is used to activate the positioning command.</p> <p>The “wPositionBlind” input defines the height position of the sunshade to be moved to.</p> <p>The “wPositionLamella” input defines the slat position of the sunshade to be moved to.</p> <p>The function outputs the position command for the sunshade actuator.</p> <p>Note:</p> <p>The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.</p>		

Position transfer for sunshade (FuTypSunshadePosition)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FuTypSunshadePosition	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
rPositionBlind	REAL	Height position of the sunshade [%] 0% = Upper end position 100% = Lower end position
rPositionLamella	REAL	Slat position of the sunshade [%] 0% = Slat open 100% = Slat closed
Return value:	Data type:	Comment:
	typSunshadePosition	Position parameter
.rPositionBlind	REAL	Height position of the sunshade [%]
.rPositionLamella	REAL	Slat position of the sunshade [%]
Graphical illustration:		
		
Function description:		
<p>The FuTypSunshadePosition function is used to transfer the sunshade positions to the scene control for sun protection. The position values can be saved as a scene.</p> <p>The “rPositionBlind” input defines the height position of the sunshade to be moved to.</p> <p>The “rPositionLamella” input defines the slat position of the sunshade to be moved to.</p> <p>The function outputs the position values for connecting to the scene control for sunshade (FbControlSunshadeScene).</p> <p>Note:</p> <p>The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.</p>		

Multiplexer Sunshade (FuMUX_2_Sunshade, FuMUX_3_Sunshade)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FuMUX_2_Sunshade, FuMUX_3_Sunshade	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
typS1_xx	typSunshade	Positioning command input 1 for the sunshade actuator
typS2_xx	typSunshade	Positioning command input 2 for the sunshade actuator
typS3_xx	typSunshade	Positioning command input 3 for the sunshade actuator
Return value:	Data type:	Comment:
	typSunshade	Positioning command output for the sunshade actuator
Graphical illustration:		
		
Function description:		
<p>The FuMUX_X_Sunshade function is used to pass two or three different positioning commands with the same priority on the sunshade actuator.</p> <p>The „typS1_xx“ to „typS3_xx“ inputs are assigned with positioning commands of upstreamed blocks with the same priority.</p> <p>Any active positioning command is send to the output. If more positioning commands are active, the upper command passes.</p>		

Sensor Functions

Presence detection (FbPresenceSensor)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbPresenceSensor	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xSensorSignal		BOOL	Presence signal from the presence sensor
xManualOccupancy		BOOL	Presence signal by manual presence detection
tHoldingTime		TIME	Holding time of the presence status after absence detection Default setting: t#2m
xAND		BOOL	Selection of the link control of both presence signals Default setting: FALSE (OR link)
Return value:		Data type:	Comment:
xPresence		BOOL	Presence status
tElapsedTime		TIME	Elapsed time is deactivated until the presence status
Graphical illustration:			
<div><div>FbPresenceSensor</div><div><div>xSensorSignal</div><div>xManualOccupancy</div><div>tHoldingTime</div><div>xAND</div></div><div><div>xPresence</div><div>tElapsedTime</div></div></div>			
Function description:			
<p>The FbPresenceSensor function block can be used to evaluate occupancy information transmitted by a presence detector and a manual control element.</p> <p>The presence detector is connected to the “xSensorSignal” input. The manual occupancy status is applied at the “xManualOccupancy” input. The presence output immediately responds to switching signals from the “xManualOccupancy” status.</p> <p>With the help of an adjustable holding time “tHoldingTime”, the occupancy status after a falling edge of the presence signal “xSensorSignal” can be held for a certain time.</p>			

With the **“xAND”** input, the logical link control of the **“xSensorSignal”** and **“xOccupancyButton”** inputs for presence detection can be defined. A TRUE signal stands for an AND link, a FALSE signal stands for an OR link.

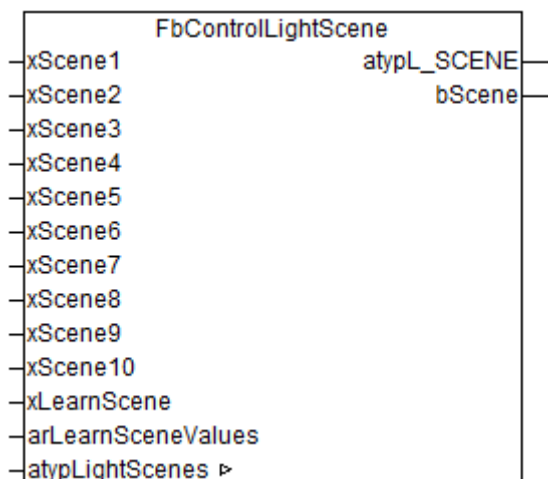
The **“xPresence”** output signals the current presence status. This is the result of the logical combination of the presence inputs.

The **“tElapsedTime”** time displays the elapsed time since the last presence detection. With renewed presence detection, the time is reset. If the **“tHoldingTime”** holding time at the **“tElapsedTime”** output has elapsed, the **“xPresence”** present status is set to FALSE.

Controlling the Room Usage Type

Scene Control for lighting (FbControlLightScene)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FbControlLightScene	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
xScene1	BOOL	Call up of scene 1
xScene2	BOOL	Call up of scene 2
xScene3	BOOL	Call up of scene 3
xScene4	BOOL	Call up of scene 4
xScene5	BOOL	Call up of scene 5
xScene6	BOOL	Call up of scene 6
xScene7	BOOL	Call up of scene 7
xScene8	BOOL	Call up of scene 8
xScene9	BOOL	Call up of scene 9
xScene10	BOOL	Call up of scene 10
xLearnScene	BOOL	Learning the current scene
arLearnSceneValues	ARRAY [1..5] OF REAL	Set value of all groups of a scene Value range: 0 – 100%
Input/Output parameters:	Data type:	Comment:
atypLightScenes	ARRAY [1..10] OF typLightScene	Saved scene values for the groups
.arSceneValues	ARRAY [1..5] OF REAL	Switching values of all groups of the scene
Return value:	Data type:	Comment:
atypL_SCENE	ARRAY [1..5] OF typLight	Current scene parameters of all groups
bScene	BYTE	Active scene

Graphical illustration:**Function description:**

The **FbControlLightScene** function block can be used to choose between several types of room utilization to adjust room conditions. Ten different scenes can be saved. For each scene, five lighting groups can be defined with different brightness values.

The individual scenes are called up via a rising edge at one of the “**xScene1..10**” inputs. The “**bScene**” output displays the scene currently called up.

The function block provides two options for saving scenes.

- With the first option, all scenes and all groups can be stored directly. The scenes are saved by writing the “**atypLightScenes**” input/output variable. The dimming values are entered for all scenes and all groups. This option is suitable for specifications at start-up.
- With the second option, all groups of the current scene can be saved. The dimming values of all lighting groups must be restored at the “**arLearnSceneValues**” input. A rising edge at the “**xLearnScene**” input saves the dimming values from “**arLearnSceneValues**” to the scene “**atypLightScenes[X]**” currently called up. This option is suitable for manually adjust a scene.

The “**atypL_SCENE**” output signals the scene parameters of all groups of the scene currently called up. The scene parameter of a group can be linked to the scene recall of an actuator function block.

Notice:

The selection of individual groups from the “**atypL_SCENE**” output can be realized with the [FuGetLightSceneValue](#) function.

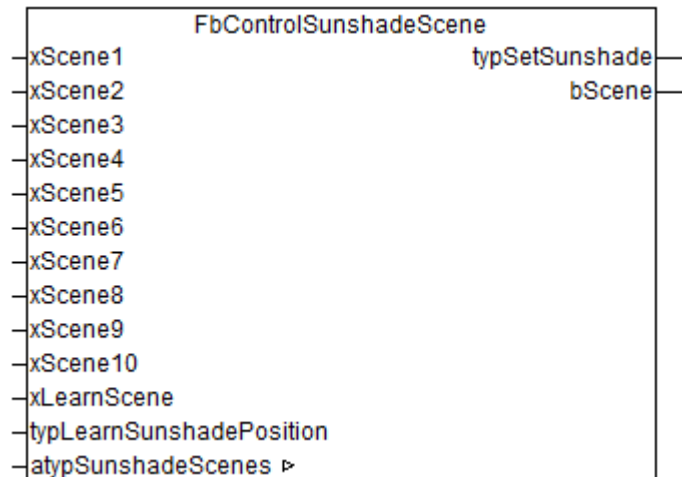
To ensure the saved scene values are retained even after a power failure, the “**atypLightScenes**” input/output variable should be declared as RETAIN PERSISTENT.

Selecting a Group from the Scene Control for Lighting (FuGetLightSceneValue)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FuGetLightSceneValue	
Type:		Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
atypL_SCENE		ARRAY [1.. 5] OF typLightScene	Scene parameters of all groups
bGroup		BYTE	Group selection
Return value:		Data type:	Comment:
		typLight	Scene parameter
Graphical illustration:			
<div><div>FuGetLightSceneValue</div><div><div>atypL_SCENE</div><div>bGroup</div></div></div>			
Function description:			
<p>The FuGetLightSceneValue function is used to transfer scene parameters of a specific group from the scene parameters of all groups.</p> <p>The “atypL_SCENE” input is connected to the output of the same name for the FbControlLightScene scene control.</p> <p>The “bGroup” input is used to select the group.</p> <p>The function outputs the scene parameters for the selected group. The parameters contain the “.rDimValue” scene dimming value and an “.xUpdate” call command. The scene output can be connected to a subsequent lighting function block.</p>			

Scene Control for Sunshade (FbControlSunshadeScene)

WAGO-I/O-PRO Library Elements			
Category:	Building automation		
Name:	FbControlSunshadeScene		
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/>	Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib		
Applicable to:	See Release Note		
Libraries Used:	-		
Input parameter:	Data type:	Comment:	
xScene1	BOOL	Call up of scene 1	
xScene2	BOOL	Call up of scene 2	
xScene3	BOOL	Call up of scene 3	
xScene4	BOOL	Call up of scene 4	
xScene5	BOOL	Call up of scene 5	
xScene6	BOOL	Call up of scene 6	
xScene7	BOOL	Call up of scene 7	
xScene8	BOOL	Call up of scene 8	
xScene9	BOOL	Call up of scene 9	
xScene10	BOOL	Call up of scene 10	
xLearnScene	BOOL	Learning the current scene	
typLearnSunshadePosition	typSunshadePosition	Set value of the respective scenes	
.rPositionBlind	REAL	Height position of the sunshade [%]	
.rPositionLamella	REAL	Slat position of the sunshade [%]	
Input/Output parameters:	Data type:	Comment:	
atypSunshadeScenes	ARRAY [1.. 10] OF typSunshadePosition	Saved scene values	
.rPositionBlind	REAL	Height position of the sunshade [%]	
.rPositionLamella	REAL	Slat position of the sunshade [%]	
Return value:	Data type:	Comment:	
typSetSunshade	typSunshade	Position command of the active scene	
bScene	BYTE	Active scene	

Graphical illustration:**Function description:**

The **FbControlSunshadeScene** function block can be used to choose between several types of room utilization to adjust room conditions. Ten different scenes with position values can be saved.

The individual scenes are called up via a rising edge at one of the “**xScene1..10**” inputs. The “**bScene**” output displays the scene currently called up.

The function block provides two options for saving scenes.

- With the first option, all scenes can be stored directly. The position values for all scenes are entered in the “**atypSunshadeScenes**” input/output variable. This option is suitable for specifications at start-up.
- With the second option, the current scene can be changed. The position values of the sunshade must be restored at the “**typLearnSunshadePosition**” input. A rising edge at the “**xLearnScene**” input saves the position values from “**typLearnSunshadePosition**” to the scene “**atypSunshadeScenes[X]**” currently called up. This option is suitable for manually adjusting a scene.

The “**typSetSunshade**” output variables contain the position commands for the sunshade actuator. The “**typSetSunshade.xMove**” variable is briefly set to TRUE when controlling a scene.

Notice:

To ensure the saved scene values are retained even after a power failure, the “**atypSunshadeScenes**” input/output variable should be declared as RETAIN PERSISTENT.

The set position values must be in the range of 0 – 100%. Otherwise, the sunshade actuator ignores the command.

Segment Control

Segment control for lighting (FbLightControl_X_Segments)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbLightControl_X_Segments	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
atypL_Segment		ARRAY [1..X] OF typLight	Dimming values of the segment Maximum: X=24
axPartition		ARRAY [1..X-1] OF BOOL	Opening state of the partition Default setting of all partitions: TRUE (partition present) Maximum: 23
Return value:		Data type:	Comment:
axActuator		ARRAY [1..X] OF BOOL	Actuator signal ON/OFF Maximum: X=24
arActuator		ARRAY [1..X] OF REAL	Dimming value [%] Value range: 0 – 100% Maximum: X=24
awActuator		ARRAY [1..X] OF WORD	Dimming signal in the area WORD Value range = 0 – 32767 Maximum: X=24
Graphical illustration:			
<div><div>FbLightControl_X_Segments</div><div><div>atypL_Segment</div><div>axPartition</div></div><div><div>axActuator</div><div>arActuator</div><div>awActuator</div></div></div>			

Function description:

The **FbLightControl_X_Segments** function block can be used for segment control or partition wall control of the lighting. Segment control is used to evaluate partition wall information and to transfer set value information to the segments. The **X** variable is the number of total segments ($X=\{2| 3| 24\}$).

The **“axPartition”** input is used to detect if a partition between two segments is open or closed.

If open, the **“axPartition[X]”** input is switched to FALSE. The segments are merged and viewed as one large segment. The merged segments are switched together.

If closed, the **“axPartition[X]”** input is switched to TRUE. The segments are switched independently.

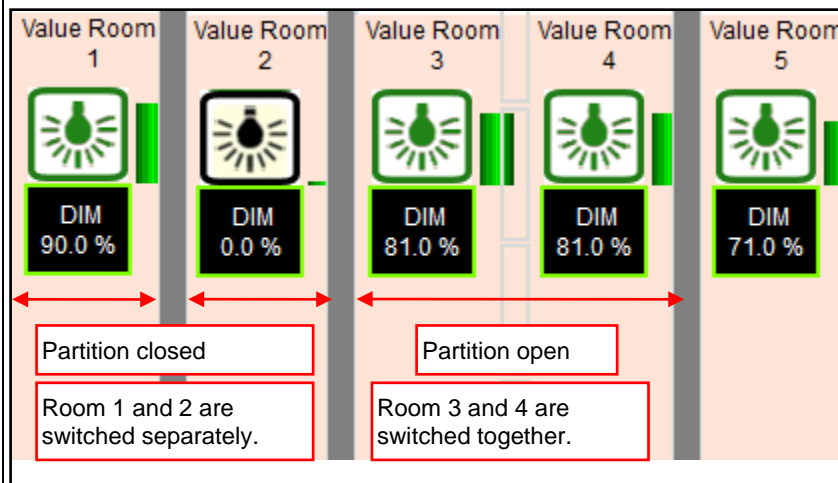
The first partition **“axPartition[1]”** is located between segments one and two.

The **“atypL_Segment”** input is assigned the **“typL_Segment”** outputs of the lighting function blocks. The switching behavior of the segments is specified by the actuator function blocks.

The **“axActuator”** output signals the digital switching states of the connected actuators. If a percentage dimming value is greater than 0, **“axActuator[X]”** switches to TRUE.

The **“arActuator”** output signals the percentage dimming values. The dimming value is specified by the dimming values of the connected actuators.

The **“awActuator”** output signals the dimming value as a signal in a range of 0 to 32767. For example, this output can be used for an analog output module.

Example:

Segment Control for Sunshade (FbSunshadeControl_X_Segments)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbSunshadeControl_X_Segments	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
atypIN_Segment		ARRAY [1..X] OF typSunshadeSegmentControl	Input signals of the segments Maximum X=24
.xUp		BOOL	Sunshade UP switch command
.xDown		BOOL	Sunshade DOWN switch command
.typSetSunshade		typSunshade	Position command of the manual override position
.xSetManualOverride		BOOL	Set the manual override
.xResetManualOverride		BOOL	Reset the manual override
axPartition		ARRAY [1..X-1] OF BOOL	Opening state of the partition Maximum: 23
Return value:		Data type:	Comment:
atypOUT_Segment		ARRAY [1..X] OF typSunshadeSegmentControl	Segment signals for transfer to the sunshade actuators Maximum X=24
.xUp		BOOL	Sunshade UP switch command
.xDown		BOOL	Sunshade DOWN switch command
.typSetSunshade		typSunshade	Position command of the manual override position
.xSetManualOverride		BOOL	Set the manual override
.xResetManualOverride		BOOL	Reset the manual override
Graphical illustration:			
<div><div>FbSunshadeControl_X_Segments</div><div><div>atypIN_Segment</div><div>atypOUT_Segment</div><div>axPartition</div></div></div>			

Function description:

The **FbSunshadeControl_X_Segments** function block can be used for segment control or partition wall control of conventional sunshade motors. Segment control is used to evaluate partition wall information based on priority and to transfer move commands to the segments.

The **X** variable is the number of total segments ($X=\{2| 3| 24\}$).

The **“axPartition”** input is used to detect if a partition between two segments is open or closed.

If open, the **“axPartition[X]”** input is switched to FALSE. The segments are merged and viewed as one large segment. The merged segments receive motion commands together.

If closed, the **“axPartition[X]”** input is switched to TRUE. The segments are moved independently.

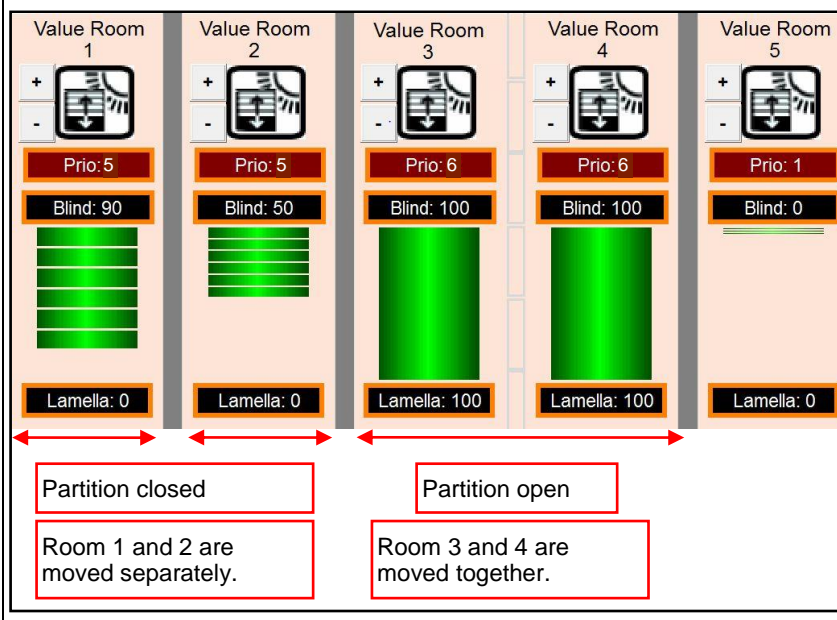
The first partition **“axPartition[1]”** is located between segments one and two.

The **“atypIN_Segment”** input variable is assigned the input signals of the individual segments. Input signals of the same segments are assigned to the same array index.

The **“atypOUT_Segment”** output delivers the input signals for the sunshade actuators. Sunshade actuators of the same segments are assigned to the same array index.

Notice:

If a segment increases in size by removing the partition, the positions of the sunshades must be synchronized in the individual segments. This can be realized by starting together.

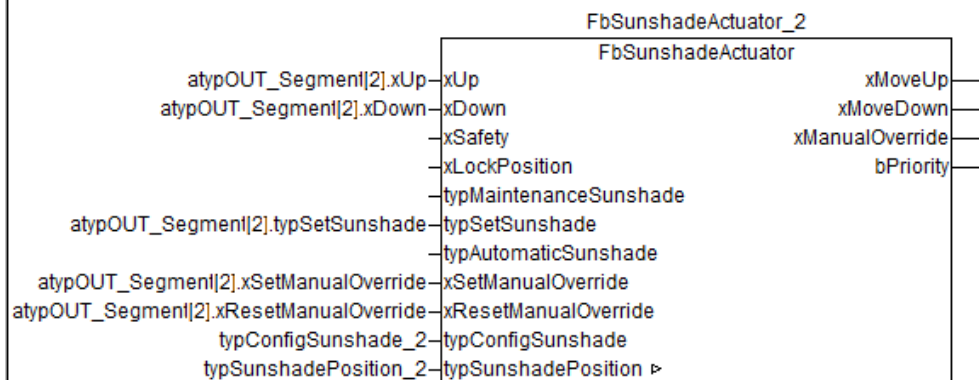
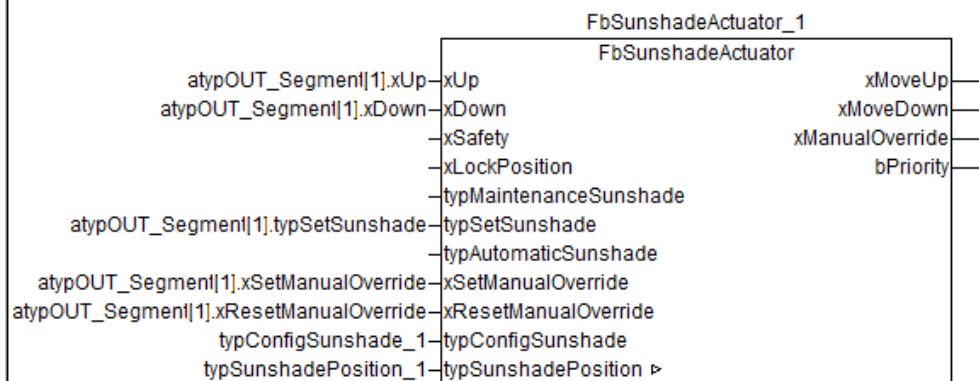
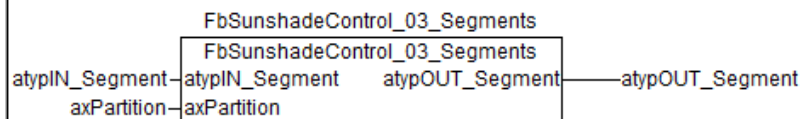
Example:

Program example in FBD:

```

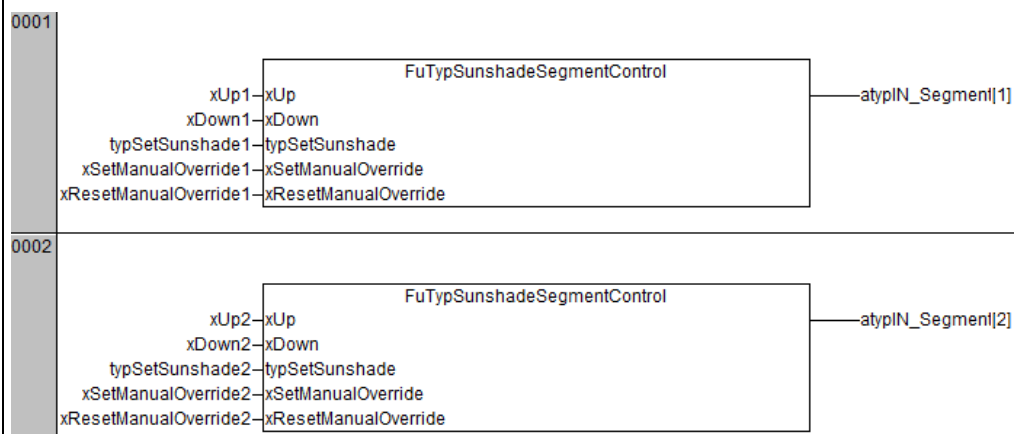
FbSunshadeControl_03_Segments : FbSunshadeControl_03_Segments;
  atypIN_Segment                : ARRAY [1..3] OF typSunshadeSegmentControl;
  axPartition                   : ARRAY [1..2] OF BOOL;
  atypOUT_Segment               : ARRAY [1..3] OF typSunshadeSegmentControl;
FbSunshadeActuator_1           : FbSunshadeActuator;
  typConfigSunshade_1          : typConfigSunshade;
  typSunshadePosition_1        : typSunshadePosition;
FbSunshadeActuator_2           : FbSunshadeActuator;
  typConfigSunshade_2          : typConfigSunshade;
  typSunshadePosition_2        : typSunshadePosition;

```



Signal Transfer to the Segment Control for Sunshade (FuTypSunshadeSegmentControl)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FuTypSunshadeSegmentControl	
Type:		Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xUp		BOOL	Sunshade UP switch command
xDown		BOOL	Sunshade DOWN switch command
typSetSunshade		typSunshade	Position command of the manual override position
xSetManualOverride		BOOL	Set the manual override
xResetManualOverride		BOOL	Reset the manual override
Return value:		Data type:	Comment:
		typSunshadeSegmentControl	Input signals for segment control
.xUp		BOOL	Sunshade UP switch command
.xDown		BOOL	Sunshade DOWN switch command
.typSetSunshade		typSunshade	Position command of the manual override position
.xSetManualOverride		BOOL	Set the manual override
.xResetManualOverride		BOOL	Reset the manual override
Graphical illustration:			
<div><div>FuTypSunshadeSegmentControl</div><div><div>xUp</div><div>xDown</div><div>typSetSunshade</div><div>xSetManualOverride</div><div>xResetManualOverride</div></div></div>			
Function description:			
<p>The FuTypSunshadeSegmentControl function is used to transfer input signals of a segment to the segment control for sunshade.</p> <p>The output of the function is connected to the “<i>atypIN_Segment[X]</i>” input variable of the segment control.</p>			

Example:

Auxiliary Functions

Pushbutton Evaluation

Evaluation of "Multiple" Clicks (FbWB_EvaluateMultipleClick)

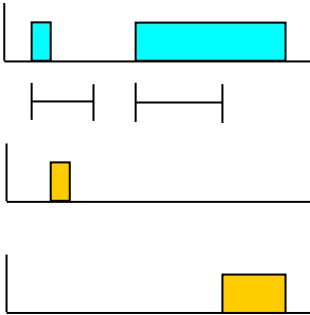
WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_EvaluateMultipleClick	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xButton	BOOL	Touch signal	
bNumberOfClicks	BYTE	Number of pushbutton signals Default setting: 2	
tPeriodToClick	TIME	Monitoring period for multiple clicks Default setting: t#500ms	
Return value:		Data type:	Comment:
xMultipleClick	BOOL	Output shows multiple clicks	
xFewerClick	BOOL	Output shows fewer clicks	
Graphical illustration:			
<div><div>FbWB_EvaluateMultipleClicks</div><div><div>xButton</div><div>bNumberOfClicks</div><div>tPeriodToClick</div><div>xMultipleClick</div><div>xFewerClick</div></div></div>			
Time referenced behavior:			
<div><div><div>xButton</div><div>tPeriodToClick</div><div>bNumberOfClicks = 2</div><div>bNumberOfClicks = 3</div><div>xMultipleClick</div><div>xFewerClick</div></div></div>			

Function description:

The **FbWB_EvaluateMultipleClicks** function block detects if a certain number of pushbutton signals has been made on the **“xButton”** binary input signals. The number of pushbutton signals can be parameterized at the **“bNumberOfClicks”** input.

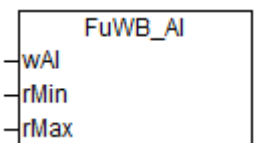
If fewer pushbutton signals occur during the parameterizable time **“tPeriodToClick”**, the **“xFewerClick”** output is set to 1 for the time of one task cycle. If at least **“bNumberOfClicks”** pushbutton signals occur during the period **“tPeriodToClick”**, the **“xMultipleClick”** output signals is set to 1 for the time of one task cycle.

Evaluation of “short”/“long” Key Action (FbWB_EvaluateShortLongPress)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_EvaluateShortLongPress	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xButton		BOOL	Touch signal
tShortPushButton		TIME	Maximum time for a brief button press Default setting: t#500ms
Return value:		Data type:	Comment:
xShort		BOOL	Output signal with a short key actuation
xLong		BOOL	Output signal with a long key actuation
Graphical illustration:			
<div><div>FbWB_EvaluateShortLongPress</div><div><div>xButton</div><div>tShortPushButton</div></div><div><div>xShort</div><div>xLong</div></div></div>			
Time referenced behavior:			
<div><div><div>xButton</div><div>tShortPushButton</div><div>xShort</div><div>xLong</div></div><div></div></div>			
Function description:			
<p>The FbWB_EvaluateShortLongPress function block detects whether the “xButton” input signal is set shorter or longer than the specified “tShortPushButton” time. In this manner, short or long signals can be distinguished from switching sensors.</p> <p>If the input signal is present for longer than the specified time, the “xLong” output signal is TRUE for the remaining time that “tButton” is active. If present for shorter, the input signal is set to TRUE via the “xShort” output signal for the time of one task cycle.</p>			

Analog Signals

Scaling the Input Values 0 to 32767 (FuWB_AI)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FuWB_AI	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Libraries Used:	-	
Input parameter:	Data type:	Comment:
wAI	WORD	Measured value of the analog input module Value range = 0 – 32767
rMin	REAL	Minimum output value for scaling
rMax	REAL	Maximum output value for scaling
Return value:	Data type:	Comment:
FuWB_AI	REAL	Scaled output value
Graphical illustration:		
		
Function description:		
<p>The FuWB_AI function scales the measured value of the analog input modules (0 – 32767) and converts it into REAL.</p> <p>The scale value range is defined via the inputs “rMin” and “rMax”.</p> <p><u>Example:</u></p> <p>Active temperature sensor 0 – 10 V, measurement range –20°C to 60°C</p> <p>Measured temperature = 10°C</p> <p>Measured value of the input module: 16384 (5 V) ,rMin = -20; rMax = 60</p> <p>Scaled measured value (REAL) = 10</p>		

Scaling of Temperature Values In °C (FuWB_AI_Temp)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FuWB_AI_Temp	
Type:		Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
iTemp		INT	Temperature value is tenths of °C
Return value:		Data type:	Comment:
FuWB_AI_Temp		REAL	Scaled temperature [°C]
Graphical illustration:			
<div><div></div><div>FuWB_AI_Temp</div><div>iTemp</div></div>			
Function description:			
<p>The FuWB_AI_Temp function scales the measured value of the resistance modules (tenth of °C) in degree Celsius (°C) and converts it into REAL.</p> <p><u>Example:</u> Measured temperature: 25.5°C Input value of the resistance module: 255 Scaled measured value (REAL) = 25.5</p>			

Scaling of Output Values 0 to 32767 (FuWB_AO)

WAGO-I/O-PRO Library Elements			
Category:	Building automation		
Name:	FuWB_AO		
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/>	Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib		
Applicable to:	See Release Note		
Libraries Used:	-		
Input parameter:	Data type:	Comment:	
rAO	REAL	Set value Value range = 0 – 100	
Return value:	Data type:	Comment:	
FuWB_AO	WORD	Scaled output value Value range = 0 – 32767	
Graphical illustration:			
<div><div></div><div><div></div><div>FuWB_AO</div><div>-rAO</div></div></div>			
Function description:			
<p>The function FuWB_AO scales the set point as a percentage to a set value for the analog output modules (0 – 32767) .</p> <p><u>Example of a 0 – 10 V signal:</u></p> <p>Set value of controller (REAL): 50%</p> <p>Output set value (WORD): 16383</p> <p>Output voltage: 5 V</p>			

Low Pass Filter, 1st Order (FbWB_LowPassFilter)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_LowPassFilter	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
rInput		REAL	Input value
typWB_ConfigLowPassFilter		←	Configuration parameters
.tCycleTime		TIME	Cycle time for the PT1 circuit Default = t#100ms
.tT1		TIME	Time constant for the PT1 circuit Default setting: t#2s
.rOffset		REAL	Measured value compensation for the input Default setting = 0
.rLowLimitAlarm		REAL	Lower limit for alarm Default setting = -32767
.rHighLimitAlarm		REAL	Upper limit for alarm Default setting = 32768
.tAlarm		TIME	Minimum time on the limiting violation until an alarm is issued. Default setting: t#10s
.rDefaultValue		REAL	Defined output value as long as the xAlarm output is set Default setting = 0
.xAutoQuit		BOOL	Automatic acknowledgement of the alarm Default setting = TRUE
xQuit		BOOL	Error acknowledgement
Return value:		Data type:	Comment:
rOutput		REAL	Filtered output value
xAlarm		BOOL	Input signal error
Graphical illustration:			
<div><div>FbWB_LowPassFilter</div><div><div>rInput</div><div>typWB_ConfigLowPassFilter</div><div>xQuit</div></div><div><div>rOutput</div><div>xAlarm</div></div></div>			

Function description:

The **FbWB_LowPassFilter** function is used to smooth noisy input signals. It can also be used to define the upper and lower alarm limits.

Configuration parameters:

The **"typWB_ConfigLowPassFilter"** configuration structure contains the following parameters:

- **"tCycleTime"** defines the cycle time for the PT1 circuit (low pass).
- **"tT1"** defines the time constant for the PT1 circuit.
- **"rOffset"** enables measured value compensation for the input signal.
- **"rLowLimitAlarm"** defines the lower limit for issuing an alarm.
- **"rHighLimitAlarm"** defines the upper limit for issuing an alarm.
- **"tAlarm"** defines the time period for which the input value must have violated the lower or upper limit before an alarm is issued.
- **"rDefaultValue"** defines the output value active while the alarm is being issued.
- **"xAutoQuit"** acknowledges the error message as soon as the input value is again situated within the defined alarm limits.

The **"rInput"** input signal is smoothed via a PT1 circuit and output at the **"rOutput"** output.

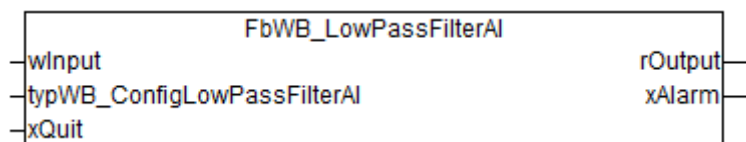
If the input signal violates the defined limits for a defined time, an alarm message is output at the **"xAlarm"** output.

In this case, the **"rOutput"** output assumes the defined default setting.

The alarm can be acknowledged after elimination of the error via a positive edge at the **"xQuit"** input, or by automatic acknowledgement.

Low Pass Filter, 1st Order for AI 0 - 32767 (FbWB_LowPassFilterAI)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_LowPassFilterAI	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
wInput		WORD	Measured value of the analog input module Value range = 0 – 32767
typWB_ConfigLowPassFilterAI		←	Configuration parameters
.tCycleTime		TIME	Cycle time for the PT1 circuit Default = t#100ms
.tT1		TIME	Time constant for the PT1 circuit Default setting: t#2s
.rOffset		REAL	Measured value compensation for the input Default setting = 0
.rMin		REAL	Minimum output value for scaling Default setting = 0
.rMax		REAL	Maximum output value for scaling Default setting = 32767
.rLowLimitAlarm		REAL	Lower limit for alarm Default setting = -32767
.rHighLimitAlarm		REAL	Upper limit for alarm Default setting = 32768
.tAlarm		TIME	Minimum time on the limiting violation until an alarm is issued. Default setting: t#10s
.rDefaultValue		REAL	Defined output value as long as the xAlarm output is set Default setting = 20
.xAutoQuit		BOOL	Automatic acknowledgement of the alarm Default setting = TRUE
xQuit		BOOL	Error acknowledgement
Return value:		Data type:	Comment:
rOutput		REAL	Scaled and filtered output value
xAlarm		BOOL	Analog input signal error

Graphical illustration:**Function description:**

The **FbWB_LowPassFilterAI** function block scales the input value and smoothens noisy input signals. It can also be used to define the upper and lower alarm limits.

Configuration parameters:

The **“typWB_ConfigLowPassFilterAI”** configuration structure contains the following parameters:

- **“.tCycleTime”** defines the cycle time for the PT1 circuit (low pass).
- **“.tT1”** defines the time constant for the PT1 circuit.
- **“.rOffset”** enables measured value compensation for the input signal.
- **“.rMin”** defines the minimum output value for scaling.
- **“.rMax”** defines the maximum output value for scaling.
- **“.rLowLimitAlarm”** defines the lower limit for issuing an alarm.
- **“.rHighLimitAlarm”** defines the upper limit for issuing an alarm.
- **“.tAlarm”** defines the time period for which the input value must have violated the lower or upper limit before an alarm is issued.
- **“.rDefaultValue”** defines the output value active while the alarm is being issued.
- **“.xAutoQuit”** acknowledges the error message as soon as the input value is again situated within the defined alarm limits.

The **“wInput”** input signal is scaled using a 4-point characteristic curve and smoothed via a PT1 circuit. The scaled and smoothed value is output at the **“rOutput”** output.

If the input signal violates the defined limits for a defined time, an alarm message is output at the **“xAlarm”** output.

In this case, the **“rOutput”** output assumes the defined default setting.

The alarm can be acknowledged after elimination of the error via a positive edge at the **“xQuit”** input, or by automatic acknowledgement.

Low Pass Filter, 1st Order for Temperatures (FbWB_LowPassFilterTemp)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_LowPassFilterTemp	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
iInput		INT	Temperature value is tenths of°C
typWB_ConfigLowPassFilterTemp		←	Configuration parameters
.tCycleTime		TIME	Cycle time for the PT1 circuit Default = t#100ms
.tT1		TIME	Time constant for the PT1 circuit Default setting: t#2s
.rOffset		REAL	Measured value compensation for the input Default setting = 0
.rLowLimitAlarm		REAL	Lower limit for alarm Default setting = -32767
.rHighLimitAlarm		REAL	Upper limit for alarm Default setting = 32768
.tAlarm		TIME	Minimum time on limit violation until an alarm is issued. Default setting: t#10s
.rDefaultValue		REAL	Defined output value as long as the xAlarm output is set Default setting = 20
.xAutoQuit		BOOL	Automatic acknowledgement of the alarm Default setting = TRUE
xQuit		BOOL	Error acknowledgement
Return value:		Data type:	Comment:
rOutput		REAL	Scaled and filtered output value
xAlarm		BOOL	Analog input signal error
Graphical illustration:			
<div><div>FbWB_LowPassFilterTemp</div><div><div>iInput</div><div>typWB_ConfigLowPassFilterTemp</div><div>xQuit</div></div><div><div>rOutput</div><div>xAlarm</div></div></div>			

Function description:

The **FbWB_LowPassFilterTemp** function block scales the input value and smoothens noisy input signals. It can also be used to define the upper and lower alarm limits.

Configuration parameters:

The **"typWB_ConfigLowPassFilterTemp"** contains the following parameters:

- **"tCycleTime"** defines the cycle time for the PT1 circuit (low pass).
- **"tT1"** defines the time constant for the PT1 circuit.
- **"rOffset"** enables measured value compensation for the input signal.
- **"rLowLimitAlarm"** defines the lower limit for issuing an alarm.
- **"rHighLimitAlarm"** defines the upper limit for issuing an alarm.
- **"tAlarm"** defines the time period for which the input value must have violated the lower or upper limit before an alarm is issued.
- **"rDefaultValue"** defines the output value active while the alarm is being issued.
- **"xAutoQuit"** acknowledges the error message as soon as the input value is again situated within the defined alarm limits.

The **"iInput"** input signal is divided by ten (°C) and smoothed via a PT1 circuit. The scaled and smoothed value is output at the **"rOutput"** output.

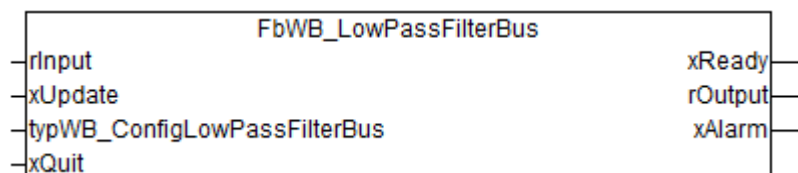
If the input signal violates the defined limits for a defined time, an alarm message is output at the **"xAlarm"** output.

In this case, the **"rOutput"** output assumes the defined default setting.

The alarm can be acknowledged after elimination of the error via a positive edge at the **"xQuit"** input, or by automatic acknowledgement.

Low Pass Filter, 1st Order for Bus Signals (FbWB_LowPassFilterBus)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_LowPassFilterBus	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
rInput	REAL	Input value	
xUpdate	BOOL	A positive edge marks a new measured value from the bus system.	
typWB_ConfigLowPassFilterBus	←	Configuration parameters	
.tCycleTime	TIME	Cycle time for the PT1 circuit Default = t#100ms	
.tT1	TIME	Time constant for the PT1 circuit Default setting: t#2s	
.rOffset	REAL	Measured value compensation for the input Default setting = 0	
.rLowLimitAlarm	REAL	Lower limit for alarm Default setting = -32767	
.rHighLimitAlarm	REAL	Upper limit for alarm Default setting = 32768	
.tAlarm	TIME	Minimum time on the limiting violation until an alarm is issued. Default setting: t#10s	
.rDefaultValue	REAL	Defined output value as long as the xAlarm output is set Default setting = 20	
.xAutoQuit	BOOL	Automatic acknowledgement of the alarm Default setting = TRUE	
xQuit	BOOL	Error acknowledgement	
Return value:		Data type:	Comment:
xReady	BOOL	Indicates that at least one new measured value has been received after a restart.	
rOutput	REAL	Scaled and filtered output value	
xAlarm	BOOL	Analog input signal error	

Graphical illustration:**Function description:**

The **FbWB_LowPassFilterBus** function block scales the input value and smoothens noisy input signals. It can also be used to define the upper and lower alarm limits.

Configuration parameters:

The **"typWB_ConfigLowPassFilterBus"** contains the following parameters:

- **"tCycleTime"** defines the cycle time for the PT1 circuit (low pass).
- **"tT1"** defines the time constant for the PT1 circuit.
- **"rOffset"** enables measured value compensation for the input signal.
- **"rLowLimitAlarm"** defines the lower limit for issuing an alarm.
- **"rHighLimitAlarm"** defines the upper limit for issuing an alarm.
- **"tAlarm"** defines the time period for which the input value must have violated the lower or upper limit before an alarm is issued.
- **"rDefaultValue"** defines the output value active while the alarm is being issued.
- **"xAutoQuit"** acknowledges the error message as soon as the input value is again situated within the defined alarm limits.

The **"iInput"** input signal is divided by ten (°C) and smoothed via a PT1 circuit. The scaled and smoothed value is output at the **"rOutput"** output.

If the input signal violates the defined limits for a defined time, an alarm message is output at the **"xAlarm"** output.

In this case, the **"rOutput"** output assumes the defined default setting.

The alarm can be acknowledged after elimination of the error via a positive edge at the **"xQuit"** input, or by automatic acknowledgement.

Additional Functions

Sun Position Calculation (FbWB_CalculateSunPosition)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_CalculateSunPosition	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
dtUTC_Time		DT	Coordinated Universal Time UTC
rLatitude		REAL	Latitude Default setting: 52,305
rLongitude		REAL	Longitude Default setting: 8,922
Return value:		Data type:	Comment:
rAzimuth		REAL	Azimuth angle of the sun [°]
rElevation		REAL	Elevation angle of the sun [°]
Graphical illustration:			
<div><div>FbWB_CalculateSunPosition</div><div><div>dtUTC_Time</div><div>rAzimuth</div></div><div><div>rLatitude</div><div>rElevation</div></div><div><div>rLongitude</div></div></div>			
Function description:			
<p>The FbWB_CalculateSunPosition function block is used to calculate the current position of the sun by the current time and geographic coordinates.</p> <p>The UTC time “dtUTC_Time” is required to calculate the position of the sun.</p> <p>The actual position is determined via inputs “rLatitude” and “rLongitude”. Latitude “rLatitude” and longitude “rLongitude” can also be calculated as follows:</p> <p>Latitude := North latitude in degrees + (north latitude in minutes / 60) Longitude := East longitude in degrees + (east longitude in minutes / 60)</p> <p>Output “rAzimuth” indicates the actual azimuth and output “rElevation” the elevation in degrees.</p>			

Sunrise and Sunset Calculation (FbWB_CalculateSunriseSunset)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_CalculateSunriseSunset	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
dtUTC_Time		DT	Coordinated Universal Time UTC
rTimeZone		REAL	Time zone
xDST		BOOL	Daylight saving time
rLatitude		REAL	Latitude Default setting: 52,305
rLongitude		REAL	Longitude Default setting: 8,922
Return value:		Data type:	Comment:
tDayLenght		TIME	Day lenght
dtSunrise		DT	Time of sunrise
dtSunset		DT	Time of sunset
Graphical illustration:			
<div><div>FbWB_CalculateSunriseSunset</div><div><div>dtUTC_Time</div><div>rTimeZone</div><div>xDST</div><div>rLatitude</div><div>rLongitude</div></div><div><div>tDayLenght</div><div>dtSunrise</div><div>dtSunset</div></div></div>			
Function description:			
<p>The FbWB_CalculateSunriseSunset function block is used to calculate the sunrise and sunset by the current time and geographic coordinates.</p> <p>The UTC time “dtUTC_Time” is required to calculate the sunrise and sunset. The calculation with local time can be realized by using the “rTimeZone” input. The daylight saving time can be activated by switching the “xDST” input to TRUE.</p> <p>The actual position is determined via inputs “rLatitude” and “rLongitude”. The calculation can be found in the description of FbWB_CalculateSunPosition.</p> <p>The “dtSunrise” and “dtSunset” outputs display the time of the sunrise and sunset. The “tDayLenght” ouput displays the time between sunrise and sunset.</p>			
Note:			
The sunrise and sunset calculation got an accuracy of +/- 3 minutes.			

Hysteresis (FbWB_Hysteresis)

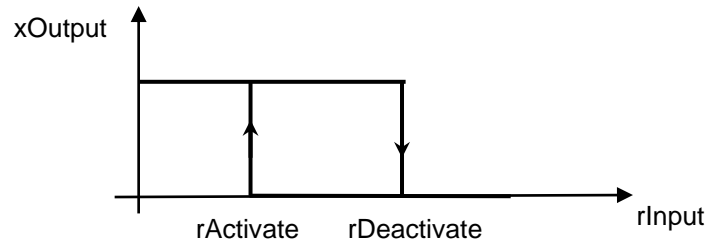
WAGO-I/O-PRO Library Elements			
Category:	Building automation		
Name:	FbWB_Hysteresis		
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/>	Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib		
Applicable to:	See Release Note		
Libraries Used:	-		
Input parameter:	Data type:	Comment:	
rInput	REAL	Input value	
rActivate	REAL	Threshold value at which the output signal is set to TRUE	
rDeactivate	REAL	Threshold value at which the output signal is set to FALSE	
Return value:	Data type:	Comment:	
xOutput	BOOL	Output signal	
Graphical illustration:			
<div><div>FbWB_Hysteresis</div><div><div>rInput</div><div>rActivate</div><div>rDeactivate</div><div>xOutput</div></div></div>			
Function description:			
<p>The FbWB_Hysteresis function block permits a switching function with adjustable hysteresis.</p> <p>Two variations are to be considered during the analysis of the input values:</p> <p>1) rActivate > rDeactivate</p> <p>The output signal “xOutput” is set to TRUE, if the condition “rInput” ≥ “rActivate” is fulfilled.</p> <p>The output signal “xOutput” is set to FALSE, if the condition “rInput” ≤ “rDeactivate” is fulfilled.</p> <p>The output signal does not change as long as the input value moves between the values “rActivate” and “rDeactivate”.</p> <div><div>xOutput</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>			

2) $rActive \leq rDeactivate$

The output signal “*xOutput*” is set to TRUE, if the condition “*rInput*” \leq “*rActivate*” is fulfilled.

The output signal “*xOutput*” is set to FALSE, if the condition “*rInput*” \geq “*rDeactivate*” is fulfilled.

The output signal does not change as long as the input value moves between the values “*rActivate*” and “*rDeactivate*”.



Operating Hours Counter (FbWB_OperatingHours_01)

WAGO-I/O-PRO Library Elements			
Category:		Building automation	
Name:		FbWB_OperatingHours_01	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Name of Library:		WagoBuilding_01.lib	
Applicable to:		See Release Note	
Libraries Used:		-	
Input parameter:		Data type:	Comment:
xEnable		BOOL	Enable operating hours counter
Input/output parameter:		Data type:	Comment:
dwOperatingMinutes		DWORD	Minutes of operation
Return value:		Data type:	Comment:
dwOperatingHours		DWORD	Operating Hours
Graphical illustration:			
<div><div>FbWB_OperatingHours_01</div><div><div>xEnable</div><div>dwOperatingHours</div><div>dwOperatingMinutes ▶</div></div></div>			
Function description:			
<p>The FbWB_OperatingHours_01 function block determines the operating hours express in minutes.</p> <p>When the “xEnable” input is activated, the minutes of operation “dwOperatingMinutes” are counted upward minute by minute.</p> <p>If the counter is to be initialized with values, the variable “dwOperatingMinutes” can be directly overwritten.</p> <p>The operating hours calculated from the minutes of operation are indicated at the “dwOperatingHours” output.</p> <p>Note:</p> <p>The operating minutes function “dwOperatingMinutes” should be defined as RETAIN PERSISTENT so that the set values are retained in the event of a loss of power or after a project upload.</p>			

Characteristics

Two-Point Characteristics (FuWB_2Point)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FuWB_2Point	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Input parameter:	Data type:	Comment:
rInput	REAL	Input value
rX1	REAL	x-coordinate of the first value
rY1	REAL	y-coordinate of the first value
rX2	REAL	x-coordinate of the second value
rY2	REAL	y-coordinate of the second value
Return value:	Data type:	Comment:
FuWB_2Point	REAL	Output value
Graphical illustration:		
<div><div>FuWB_2Point</div><div><div>rInput</div><div>rX1</div><div>rY1</div><div>rX2</div><div>rY2</div></div></div>		
Characteristic:		

Function description:

The **FuWB** function describes a linear equation defined by the two points ("**rX1**", "**rY1**") and ("**rX2**", "**rY2**").

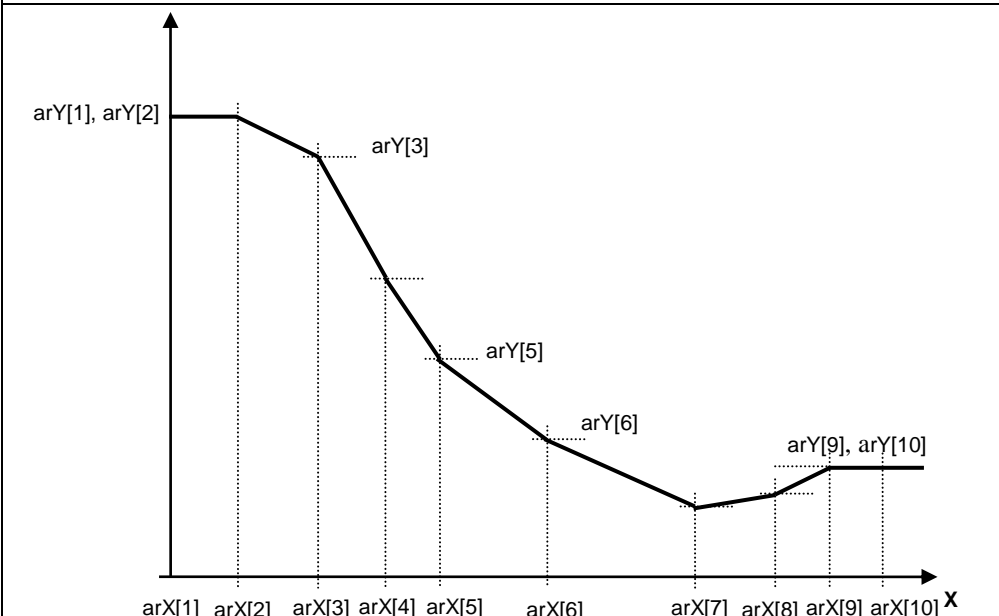
The input value "**rInput**" is converted in accordance with the linear equation and output at the function output.

If "**rX1**" and "**rY2**" are identical (vertical characteristic), the output is set to zero. If "**rY1**" and "**rY2**" are identical, the output is set to "**rY1**".

Note:

The reference points X must always be entered in an ascending order ($rX1 < rX2$).

Ten-Point Characteristics (FuWB_10Point)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FuWB_10Point	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Input parameter:	Data type:	Comment:
rInput	REAL	Input value
arX	ARRAY [1..10] OF REAL	Reference values for points 1 to 10
arY	ARRAY [1..10] OF REAL	Output values for points 1 to 10
Return value:	Data type:	Comment:
FuWB_10Point	REAL	Output value
Graphical illustration:		
<div><div>FuWB_10Point</div><div><div>rInput</div><div>arX</div><div>arY</div></div></div>		
Characteristic:		
		

Function description:

The **FuWB_10Point** function defines straight segments by the points ("**arX[1]**", "**arY[1]**") to ("**arX[10]**", "**arY[10]**").

the "**rInput_X**" input value is divided, linearized and output using these signals.

The points entered thus determine the value of the output signal $Y = f(x)$.

In order to limit the output value outside of the defined segments, it is recommended that you define the last Y points equally high.

Note:

The reference points X must always be entered in an ascending order ($rX1 < rX2$). If two X points that follow one another are identical, the output value is set to 0.

Four-Point Characteristic (FuWB_4Point)

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	FuWB_4Point	
Type:	Function <input checked="" type="checkbox"/>	Function block <input type="checkbox"/> Program <input type="checkbox"/>
Name of Library:	WagoBuilding_01.lib	
Applicable to:	See Release Note	
Input parameter:	Data type:	Comment:
rInput	REAL	Input value
rX1	REAL	x-coordinate of the first value
rY1	REAL	y-coordinate of the first value
rX2	REAL	x-coordinate of the second value
rY2	REAL	y-coordinate of the second value
Return value:	Data type:	Comment:
FuWB_4Point	REAL	Output value
Graphical illustration:		
<div><div>FuWB_4Point</div><div><div>rInput</div><div>rX1</div><div>rY1</div><div>rX2</div><div>rY2</div></div></div>		
Characteristic:		
<div><div>Output</div><div><div><div>Y2</div><div>Y1</div></div><div><div>X1</div><div>X2</div></div><div>Input</div></div><div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div></div>		
Function description:		
<p>The FuWB_4Point function describes a linear equation with lower and upper delimitation. It is described by the two inflection pairs ("rX1", "rY1") and ("rX2", "rY2").</p> <p>The output value is limited to the minimum value Y1 or to the maximum value Y2 in the case of input values "rInput" smaller than X1 and larger than X2. The output value changes according to a linear equation between these two values.</p> <p>Note:</p> <p>The reference points X must always be entered in an ascending order (rX1 < rX2).</p>		

Appendix

Structures

typLight

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	typLight	
Type:	Data type <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Applicable to:	See Release Note	
Declaration:		
TYPE typLight: STRUCT rDimValue : REAL; (* [%], 0 – 100, 0% = OFF *) xUpdate : BOOL; (* Pulse Update *) END_STRUCT END_TYPE		

typSunshade

WAGO-I/O-PRO Library Elements		
Category:	Building automation	
Name:	typSunshade	
Type:	Data type <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Applicable to:	See Release Note	
Declaration:		
TYPE typSunshade:		
STRUCT		
wPositionBlind	: WORD;	(* [%], 0 – 100, 0% = upper end position *)
wPositionLamella	: WORD;	(* [%], 0 – 100, 0% = Lamella open *)
xMove	: BOOL;	(* TRUE = move to position *)
END_STRUCT		
END TYPE		

Initial Values for Variables

The following section briefly explains how variables are declared with initial values. This allows the user to specify parameter values of function blocks at program start.

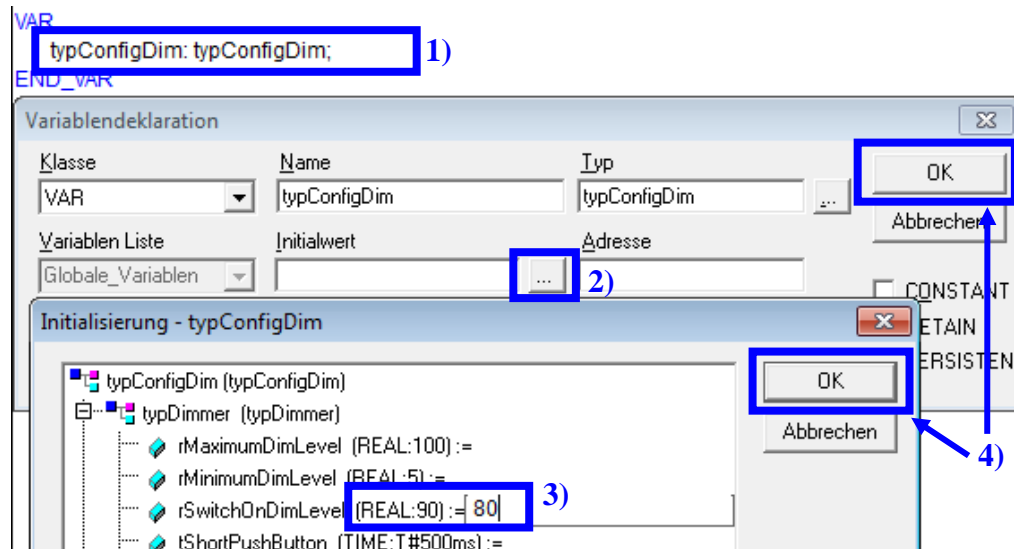


Figure 1: Example - Initial Value Switch-On Value of the “typConfigDim” Data Type

- 1) Select the variable to be initialized. Press the [Shift] + [F2] keys at the same time. You can also right-click to display the [Variable Declaration...] context menu. The dialog for variable declaration automatically opens. Make sure that the correct data type is entered in the “Type” area.
- 2) In the Initial Value areas, click [...]. The “Initialization” window automatically opens.
- 3) Select the respective variables and enter the initial values after “=”.
- 4) Click [OK] in the “Initialization” and “Variable Declaration” windows to confirm the entries. The initial values set appear in the declaration area.

The described process is independent of the programming language used and the variable declaration view.

Calibration of the Brightness Measurement

The following section describes how the brightness measurement of a light sensor is calibrated. It is used to adjust the measured light intensity compared to the light intensity at the workstation. The calibration is carried out using the variables of the “*typBrightnessMeasurement*” structure.

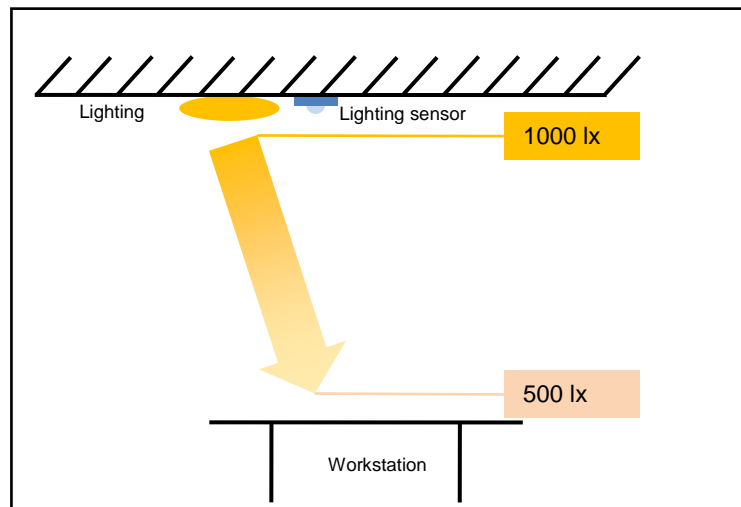


Figure 2: Measured Light Intensity Compared to the Light Intensity at the Workstation

Two measurements from the light sensor are taken for the calibration. For both measurements, the luxmeter is placed on the work surface where the specified light intensity must be reached.

The following preparations must be carried out for calibration:

- The source of light to be measured must be switched on about 20 minutes before measuring, so that the lamps can operate at their full potential.
- The specified light intensity level must be measured on the work surface. A luxmeter that can adapt well to the $V(\lambda)$ curve is required for this.
- The calibration cannot be performed until the room has been completely furnished since the measured values of the light sensor depend on the reflection properties of the room.
- Start value “*rGain*” = 3
- Start value “*rGainAdaptation*” = 20

The first measurement is performed in a darkened room using pure artificial light. The calibration value is determined as follows:

- If the light intensity at the workstation is higher than the light intensity setpoint, the “*rGain*” calibration value must be increased until the specified light intensity is reached.
- If the light intensity at the workstation is lower than the light intensity setpoint, the “*rGain*” calibration value must be decreased until the specified light intensity is reached.

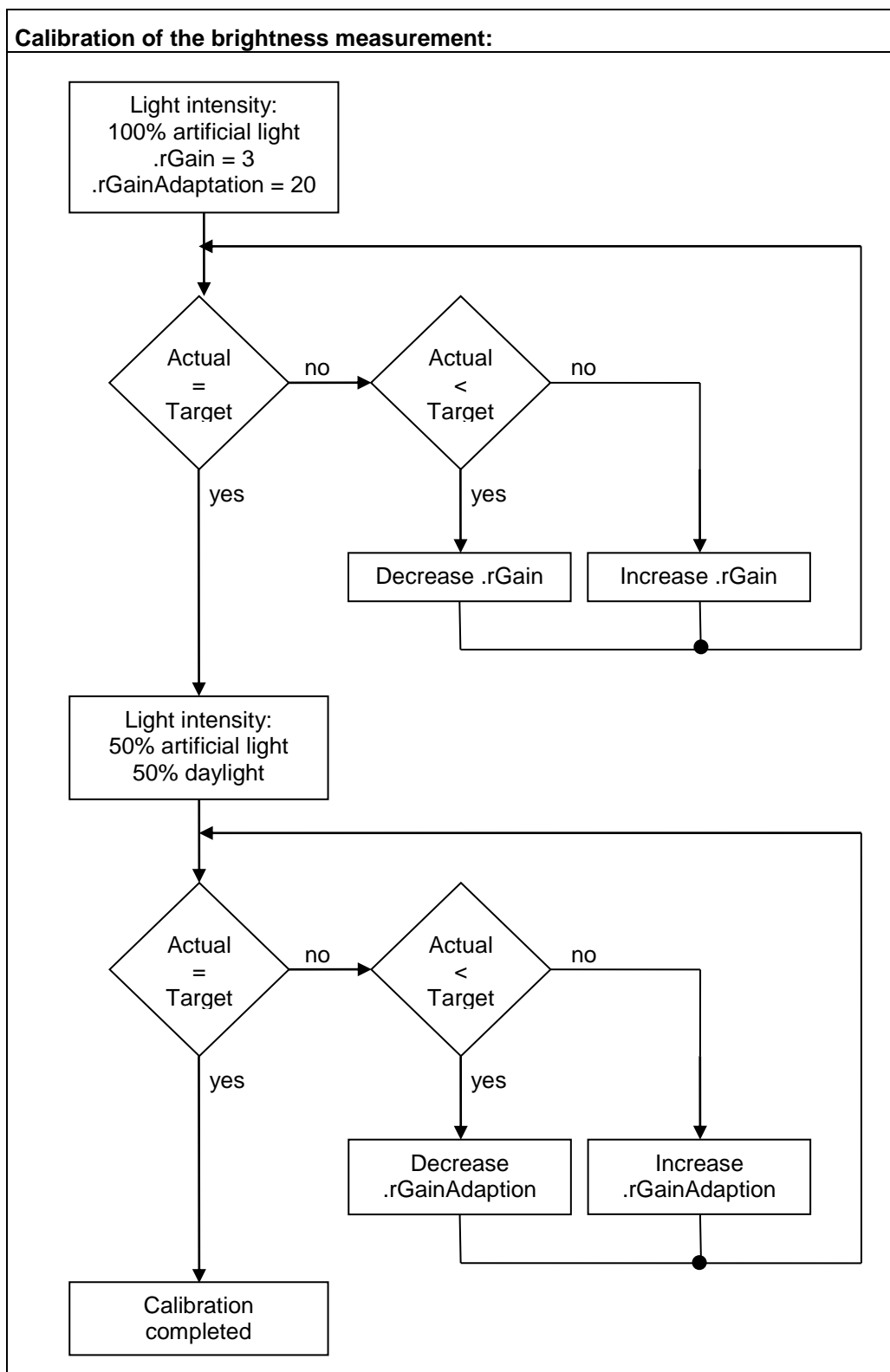
For safety reasons, the light intensity measured by the luxmeter should be about 10% higher than the specified light intensity setpoint.

The second calibration measurement is required in order to determine the percentage adaptation of the calibration value. This measurement is performed in a semi-darkened room with residual artificial light.

The second measurement is performed as follows:

- If the light intensity at the workstation is lower than the light intensity setpoint, the percentage of the adaptation "*rGainAdaptation*" must be increased until the specified light intensity is reached.
- If the light intensity at the workstation is higher than the light intensity setpoint, the percentage of the adaptation "*rGainAdaptation*" must be decreased until the specified light intensity is reached.

If the percentage adaptation of the calibration value is performed in a semi-darkened room, the lowest possible offset is achieved depending on the part of daylight or artificial light. The actual value of the light intensity can still be lower than the light intensity setpoint.



WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49 (0)5 71/8 87 – 0
Fax: +49 (0)5 71/8 87 – 1 69
Email: info@wago.com

Internet: <http://www.wago.com>

