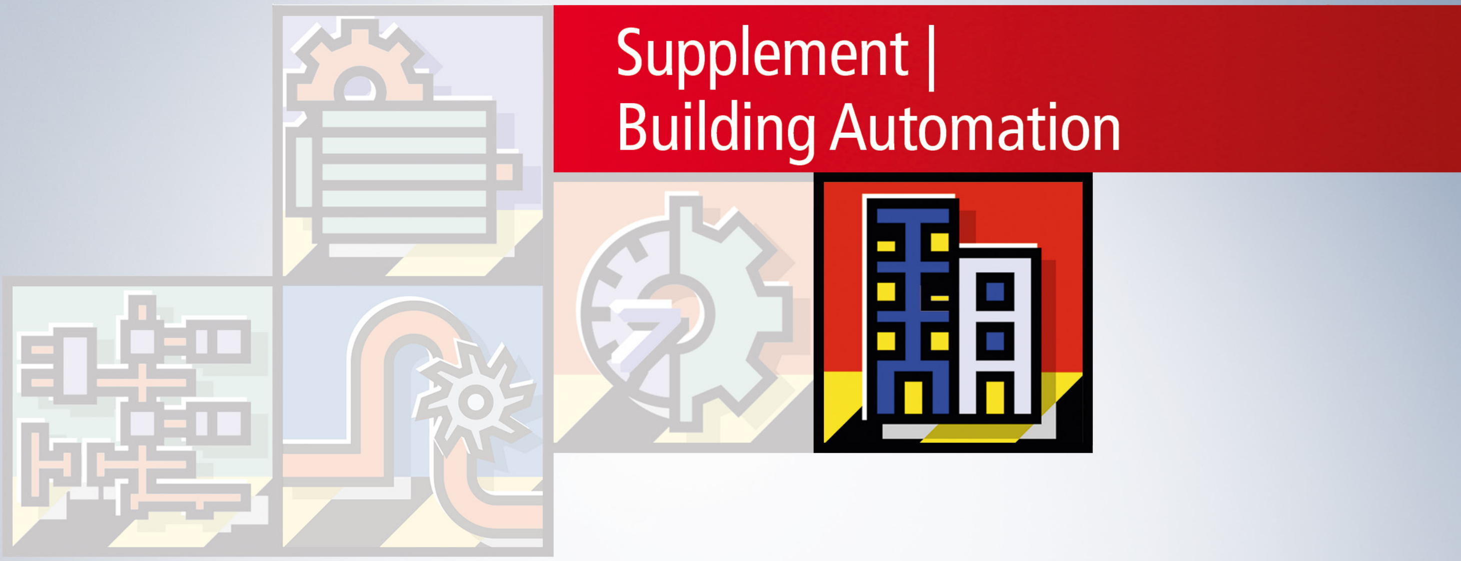
BECKHOFF New Automation Technology

Manual | EN

TS8100

TwinCAT 2 | Building Automation Framework V2.2x



2022-12-20 | Version: 1.1

Table of contents

1 Foreword 5
- 1.1 Notes on the documentation 5
- 1.2 For your safety 5
- 1.3 Notes on information security 7

2 Overview 8

3 Architecture 11

4 Differences to version V2.12 14

5 Setup 15

6 First Steps 16

7 Building Automation Manager 28
- 7.1 Control Elements 28
  - 7.1.1 Main Window 28
  - 7.1.2 Navigation Tree 29
  - 7.1.3 Message Window 30
  - 7.1.4 Link Dialog 31
  - 7.1.5 Edit 32
  - 7.1.6 Project 33
  - 7.1.7 Tools 37
- 7.2 Controller Configuration 54
  - 7.2.1 Controller Settings 54
  - 7.2.2 Application Program 56
  - 7.2.3 I/O Configuration 190
- 7.3 Application Examples 193
  - 7.3.1 Dyn. group allocation for lamps 193
  - 7.3.2 Scene cascading 194
  - 7.3.3 Change of operating states 195
  - 7.3.4 Linking of power supply terminals with diagnostics 197

8 PLC Function Blocks 198
- 8.1 Object Id 200
- 8.2 Constants 202
- 8.3 CFC Sample 204
- 8.4 Sensors 206
  - 8.4.1 Digital Signals 206
  - 8.4.2 Analog Signals 210
- 8.5 Actuators 214
  - 8.5.1 Lamps 214
  - 8.5.2 Blind Drives 220
  - 8.5.3 Window Drives 223
  - 8.5.4 Valve Actuators 224
- 8.6 Sensor Groups 227
  - 8.6.1 Digital Signal Groups 227
  - 8.6.2 Analog Signal Groups 228

8.7 Actuator Groups 230
8.7.1 Lamp Groups 230
8.7.2 Blind Drive Groups 232
8.7.3 Window Drive Groups 233
8.7.4 Valve Actuator Groups 235
8.8 Functional Units 236
8.8.1 Lightings 236
8.8.2 Sunblinds 246
8.8.3 Windows 249
8.8.4 HVAC 250
8.9 Scenes 252
8.9.1 FB_BAFScenesStandard 252
8.10 Time Scheduler Channels 253
8.10.1 FB_BAFSchedulerDaily 253
8.10.2 FB_BAFSchedulerWeeklyIndividual 254
8.10.3 FB_BAFSchedulerWeeklyContinual 254
8.10.4 FB_BAFSchedulerMonthlyNthDay 255
8.10.5 FB_BAFSchedulerMonthlyIndividualWeekday 256
8.11 Weather Stations 257
8.11.1 FB_BAFWeatherStationVirtual 257
8.11.2 FB_BAFWeatherStationElsnerP03 260
8.11.3 FB_BAFWeatherStationThiesClimaSensorD 264
8.11.4 FB_BAFWeatherStationAnalogDigitalSignals 268
8.12 Subsystems 271
8.12.1 FB_BAFPowerMeasurementKL3403 271
8.13 Application Program 273
8.13.1 FB_BAFApplicationProgram 273
8.14 Appendix 274
8.14.1 Data types 274
8.14.2 Error codes 276

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product

ℹ This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found here.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed.

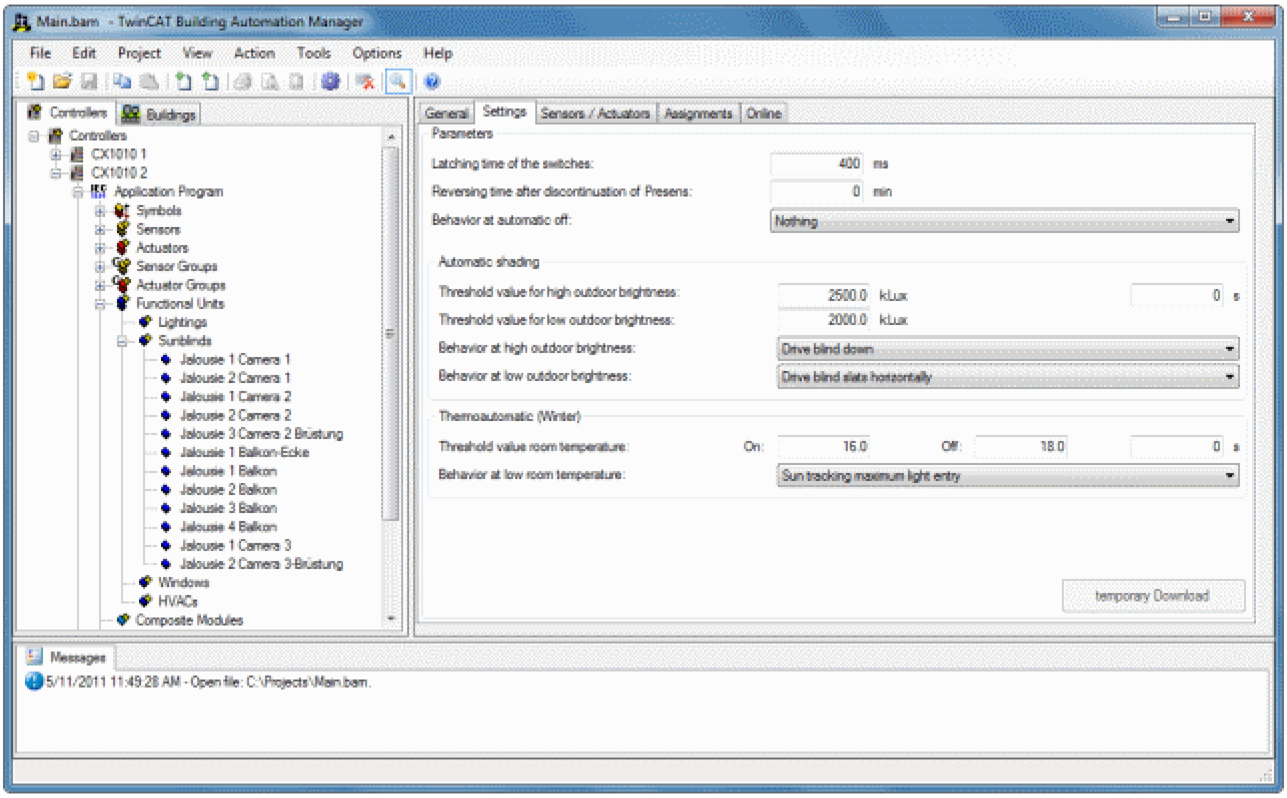
2 Overview

Room automation has gained increasingly in significance in recent years. The term is understood to mean cross-system automation within rooms or, in the case of larger rooms, within zones. Functions from the fields of lighting, air conditioning, sun shading and windows are combined with each other in such a way that an improved energy balance is achieved for the operator. Heating or cooling with open windows or additional lighting when there is sufficient daylight worsen the energy efficiency. Particularly large potential is offered by intelligent façade control, for example. Hence, sun blinds can be retracted on sunny winter days at the weekend in such a way that the rooms are adequately heated by the sun. In the summer the cooling load is reduced by summer night cooling. Unoccupied rooms are completely shaded. Lighting and shading can be attuned to one another in such a way that a constant light intensity is continuously present. The result for the user is increased usage comfort.

Today, buildings such as offices, hotels, schools, museums or salesrooms are expected that to be able to adapt themselves quickly and simply to the changing requirements of different users. This places particular demands on the automation system, so that changes in the division or use of the area can take place without rewiring. These adaptations are possible with the TwinCAT Building Automation Framework even without programming knowledge. A flat system architecture and a uniform tool for all systems permit the user to make changes and extensions simply and quickly.

The system integrators of medium to larger buildings are faced with particular challenges. The large number of data points and the fact that the project is distributed in most cases over several controllers means that powerful tools are required for commissioning. Data points, functions and their relationships, as well as the intercommunication between the controllers must be represented clearly.

This is where the TwinCAT Building Automation Framework is applied. On the one hand, the concept of the parameterisable PLC library offers the possibility to parameterise room automation functions comfortably. Sensors, actuators, groups, controllers, timer functions, weather stations, energy measurement and scenes are contained in a PLC library, which is intuitively parameterised using a configuration program, the TwinCAT Building Automation Manager. On the other hand, the controller function can be extended by the user’s own PLC blocks in IEC 61131-3. Function blocks offer the possibility to access the configured modules within the PLC.



Parameters

Latching time of the switches: 400 ms

Reversing time after discontinuation of Presens: 0 min

Behavior at automatic off: Nothing

Automatic shading

Threshold value for high outdoor brightness: 2500.0 kLux 0 s

Threshold value for low outdoor brightness: 2000.0 kLux

Behavior at high outdoor brightness: Drive blind down

Behavior at low outdoor brightness: Drive blind slats horizontally

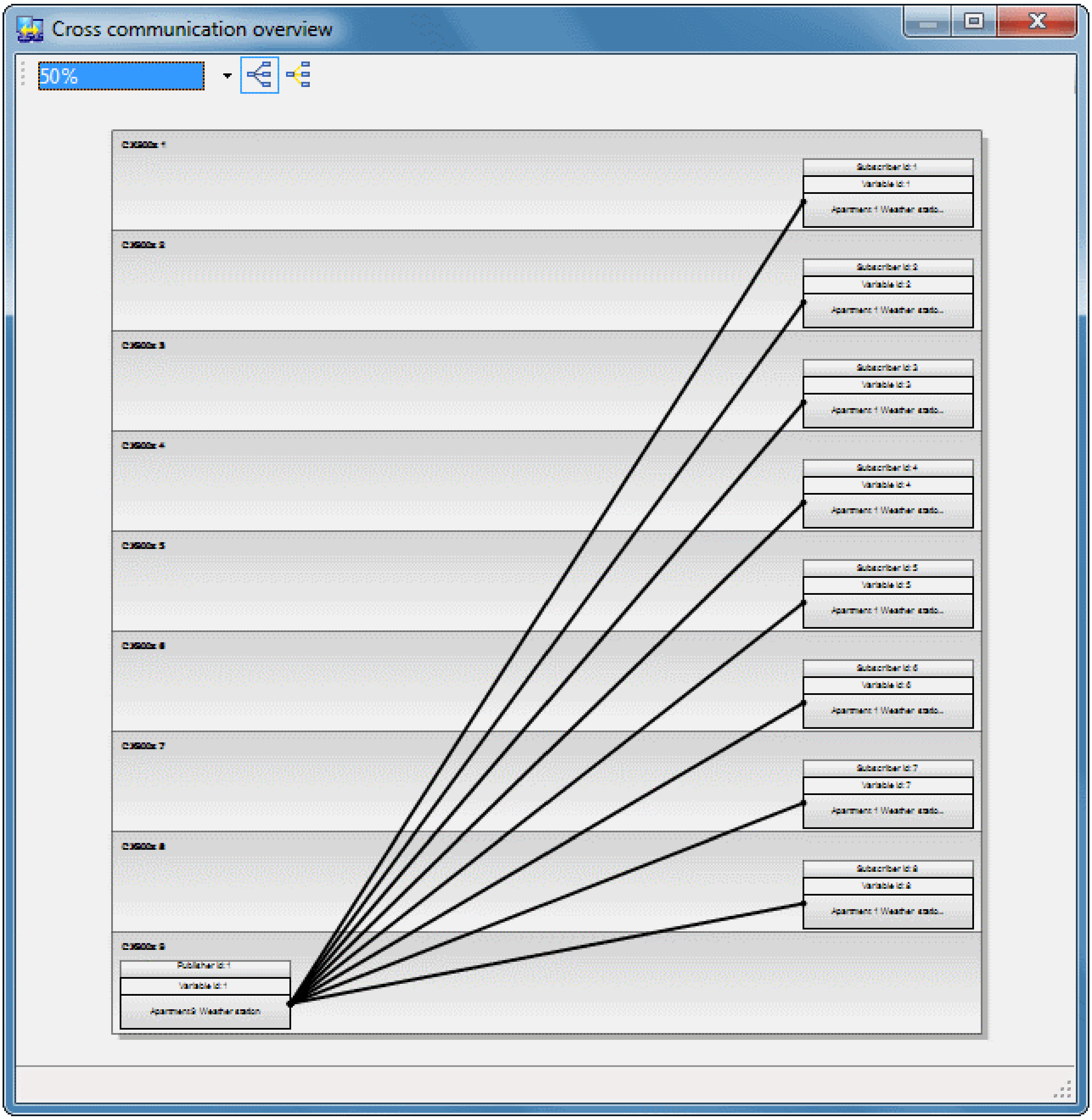
Thermoautomatic (Winter)

Threshold value room temperature: On: 16.0 Off: 18.0 0 s

Behavior at low room temperature: Sun tracking maximum light entry

5/11/2011 11:49:28 AM - Open file: C:\Projects\Main.bam.

The TwinCAT Building Automation Framework offers particular support for the intercommunication between the controllers. Groups that contain actuators or sensors are not limited to the local controllers. Groups can also contain elements of controllers that are connected to one another via a network. Scenes are also not limited to the local controllers. Hence, a scene can activate a further scene. This scene can be on any other controller. The TwinCAT Building Automation Framework provides all resources necessary for the intercommunication between the controllers. The TwinCAT real-time network variables are the basis for this. They ensure virtually delay-free communication.



The Building Automation Framework provides the following functions:

- Digital sensors
- Analog sensors
- Counter
- Lamps
- Blind drives
- Valve actuators
- Window drives
- Groups with sensors or actuators
- Lighting
- Shading
- Heating/cooling

- Room control units
- Scenes
- Timer functions
- Weather stations
- Energy measurement
- Security lighting (by DALI)
- DALI
- EnOcean
- ...

On the basis of VDI 3813 – Part 2, various room automation functions can be implemented with the TwinCAT Building Automation Framework, such as:

- Constant light control
- Brightness-dependent automatic light
- Timer programs for the modes of operation of the air conditioning system and limitation of light intensity
- Window monitoring
- Occupancy switching
- Automatic thermal control of sun blinds
- Free night cooling
- ...

The options of the TwinCAT Building Automation Framework are supplemented by additional modules. Modules are presently offered for the generation of terminal lists, the reading in of Excel lists, the documentation of set parameters and the generation of user interfaces.

Differences between Framework and Library

A library provides functions for certain tasks, in the form of a collection of functions, function blocks and data types of an area of application. A framework represents a layer covering different libraries, thereby simplifying software development. Frameworks already contain functionality that can be integrated in the actual implementation. In contrast to a library, from which the programmer uses individual function blocks as required, the framework provides the structure for the application.

Quote from: Design Patterns - Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson and Vlissides, published by Addison-Wesley):
"A framework is a set of cooperating classes (function blocks) that define a reusable design for a certain application context. The framework determines the architecture of the application."

One benefit of frameworks is that they provide readily designed and implemented software structures that can be reused, so that developers don't have to keep implementing basic functionalities such as management of group objects, for example. Encapsulation of implementation details makes the application modular and enhances the software quality. Another advantage of frameworks is expandability. The framework offers developers fixed points at which they can intervene and configure and expand the framework.

3 Architecture

The TwinCAT Building Automation Framework essentially consists of two components: the TwinCAT Building Automation Manager and a PLC library containing the application program. The purpose of the TwinCAT Building Automation Manager is to parameterise the application program and to generate the I/O mapping. For generation of the I/O mapping, the TwinCAT Building Automation Manager generates and activates a project file for each controller for the TwinCAT System Manager (*.tsm file).

While the TwinCAT Building Automation Manager is needed only for commissioning, the PLC library contains all function blocks required at runtime in the application program. Unlike most other PLC libraries, the Building Automation Framework's PLC library does not contain individual function blocks that still have to be assembled by a programmer into an executable program, but instead contains a complete, executable PLC program. After inserting the PLC library it only remains to call the entry routines. Everything else runs independently in the PLC library. In order to make this possible, all necessary global variables are provided with the PLC library. The purpose of these is to receive the parameters from the TwinCAT Building Automation Manager. Moreover, the global variables are used for internal communication between the individual objects. The assignments of global variables among one another are also defined by the global variables without having to re-compile the PLC program.

For the later application of the TwinCAT Building Automation Framework it is advantageous to understand the relationships of the individual objects (digital signals, lamps, timer, groups etc.) to one another. All objects have a clearly defined task and communicate with one another via permanently defined interfaces. The assignment of the objects among one another takes place over several layers. Each layer has a clear task and also communicates with the other layers via defined interfaces.

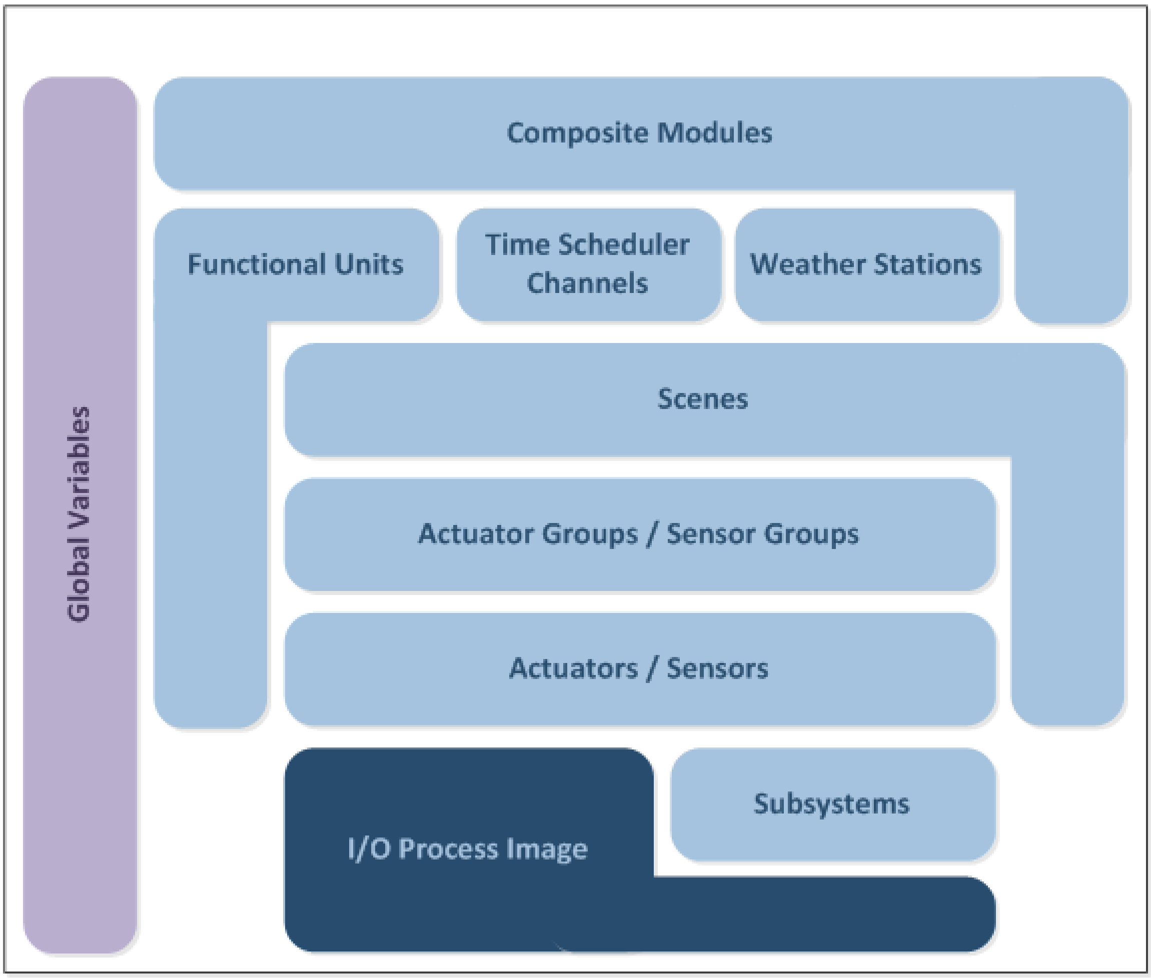


Fig. 1: TcBAFramework_ArchitekturDiagramm01

I/O process image

At the lowest level is the process image, via which the application program communicates with the hardware. The connection between the application program and the process image is defined via the I/O mapping. The user specifies which actuator, sensor or subsystem (DALI line, EnOcean line etc.) is assigned to which fieldbus terminal. The TwinCAT Building Automation Manager automatically generates the I/O mapping during the activation of the configuration. As a matter of principle only sensors, actuators or subsystems can be linked with the process image.

Actuators, sensors and subsystems

Information that is read in via input terminals usually has to be prepared. This includes functions such as scaling, filtering, negation, generation of replacement values, flank evaluation or delay. Thus ‘clean’ input signals are available to all higher-level objects. Threshold switches can be defined at sensor level. It is thus possible to activate scenes when the actual value is exceeded or undercut.

An actuator object represents a lamp, a blind drive, a valve actuator or a window drive. The actuator object receives switching commands, prepares them and generates the process data. The behaviour of the actuator can be specified via the parameters of the actuator object. The limitation of the control value for a lamp or the driving duration of a blind drive are just a few examples of this.

Not every sensor object or actuator object is directly connected to a fieldbus terminal. In some cases linking takes place indirectly via a further fieldbus such as DALI, EnOcean or RS485. A subsystem object abstracts the access to such a subsystem. All services required for operation and configuration are also contained in the subsystem object. For example, there is a subsystem object for DALI lines, which is connected to the appropriate DALI fieldbus terminal. The configuration and addressing of the DALI devices can be accomplished with the help of the DALI object. A DALI object also makes all necessary interfaces available, so that a lamp object can access individual DALI lamps. A lamp object can be assigned selectively to a channel of a fieldbus terminal or a DALI lamp of a DALI subsystem. The interface for controlling a lamp is always the same, irrespective of whether a lamp object for DALI or a lamp object for a dimmer terminal is involved. Access to the higher levels is thus “fieldbus-neutral” and considerably simplifies further use.

Actuator groups and sensor groups

It is very frequently necessary to group sensors or actuators. Hence there can be several blind drives in a room that are always driven at the same time, or several temperature sensors, whose average value is passed on to the controller. For this task there is an equivalent group object for each sensor or actuator object. An actuator group object contains a reference to a list of the respective group elements. If the group object receives a switching command, then this is passed on to all group elements. The states of the individual elements are likewise collected and evaluated. It is thus possible to read from each group object how many elements are assigned to this object or how many of them are signalling an error. Like sensor objects, sensor groups can also call scenes with the aid of threshold switches.

Group objects are not limited to one controller. Elements can be assigned not only by the local controller, but also by any controller located within the same network. The TwinCAT Building Automation Manager generates the necessary infrastructure. TwinCAT network variables are used here so that virtually delay-free communication is possible.

The lamp group object is of particular importance if several DALI lamps are registered. The DALI lamps can originate from the same DALI line, from a further DALI line of the same controller or from a DALI line of another controller. The TwinCAT Building Automation Framework ensures that the DALI devices are assigned to the correct DALI groups and that the necessary TwinCAT network variables are created for communication between the controllers. A group of lamps that is distributed over the entire network can thus be controlled with a group object.

Scenes

Scenes have a central function in the TwinCAT Building Automation Framework. Sensor objects, group objects, timer channel objects and weather station objects can exclusively call scenes. Hence, all event-controlled commands merge in the scenes. The result of triggering a scene is that the stored commands are executed. Commands from any local objects may be contained. Since scenes can also be called via the network, they offer an efficient method of executing global building actions (outside lights, changing the mode of operation of the air conditioning system or the like). The cascading of scene calls is also possible. Hence, one scene can call another. This scene may be located on the same controller or on a different one.

Functional units

The task of sensor objects, actuator objects and group objects is to standardise access to the hardware. The objects also contain fundamental functions that affect the input and output. Switching functions are possible via threshold switches. However, this does not suffice for air conditioning or constant light control. This task is performed by functional units. Functional units represent the link between sensors and actuators. The values of the sensors are evaluated according to the task and the assigned actuators are controlled.

Timer switch channels

Recurring events can be activated by timer switch channels. Objects for daily, week and monthly timer switch channels are available for this. Sunrise and sunset can also be selected as events in addition to the absolute time.

Weather stations

The values from the weather stations can be distributed via weather station objects in the network and individually evaluated on each controller. At least one threshold switch can be parameterised for each measured value. This can call scenes selectively when the threshold is exceeded or undercut. The concept of the virtual weather station supports the distribution of the meteorological data over a network. The data is transmitted virtually delay-free on the basis of the TwinCAT real-time network variables. In order to do this, a virtual weather station object is linked to a ‘real’ weather station of another controller. The TwinCAT Building Automation Manager generates all necessary TwinCAT network variables upon activation of the configuration.

Composite modules

Composite modules represent the highest level. They enable access to different objects without being assigned to a specific system such as lighting or air conditioning. This is used, for example, for room control units which frequently access different functional units and which must be capable at the same time of calling scenes. The TwinCAT Building Automation Framework offers two object types with which lighting, shading and scene management can be implemented with simple digital inputs and outputs. Using buttons, individual lights can be dimmed, individual sun blinds can be driven and scenes can be called. However, the scenes can also be changed and persistently saved via the buttons.

Global variables

The individual objects use global variables in order to exchange information and commands with one another. All necessary variables are provided in the PLC library of the TwinCAT Building Automation Framework, so that it is not necessary to explicitly create them.

4 Differences to version V2.12

Version V2.20 has been fundamentally extended and improved in relation to version V2.12. The entire functional range of room automation is covered by the new functions. In addition to the extensions, however, the existing performance features have been revised. So that projects created with V2.12 can continue to be maintained, versions V2.12 and V2.20 can be installed and used in parallel on the same computer. The project files can easily be distinguished by the file extension. Due to some extensions, however, old projects cannot be directly read into version V2.20. In future there may be a migration tool that will transfer as much information as possible from an old project to a new one. The following table contains a summary of the most important extensions and changes:

Requirements

V2.12	V2.20	Improvement
File extension of the project file is *.bam.	File extension of the project file is *.bamx.	The individual project files can be assigned to the individual versions by the file extension alone.
TwinCAT V2.10 is required.	TwinCAT V2.11 is required.	The CX5000 can thus be supported.
Supports EnOcean via KL6021-0023.	Supports EnOcean via KL6021-0023 and via KL6581.	Several EnOcean receivers can be connected to one terminal.
Only signal groups can call scenes.	Signals and signal groups can call scenes.	In the case of a signal, a group does not always have to be created in order to call a scene.
There are 2 versions for the lighting.	There are 6 versions for the lighting.	Lightings for staircase lighting, twilight switch, emergency lighting or the like are now also available.
-	Objects for timer switch channels, composite modules and weather stations have been added.	Important performance features are thus available with the new version.
The commands are transmitted by ADS/TCP.	The commands are transmitted via TwinCAT network variables.	Network variables can be integrated more easily into the system. It is not necessary to register all devices in the TwinCAT router of the controller. The transmission by TwinCAT network variables is considerably more efficient, since the same command can be sent to several devices with one telegram.
Access to the objects in the PLC takes place via global variables.	Function blocks are available in the PLC for each type of object. Each object can be addressed via these function blocks.	The extension of the PLC program with the user's own PLC functions has become simpler, in particular for the graphic display types.
-	Copy, pasting, importing and exporting are supported by the TwinCAT Building Automation Manager.	Project creation is very much simplified. Once parameterised, elements can be duplicated as often as required.
-	Searching for I/O devices and fieldbus terminals is supported.	This can accelerate the registration of the I/O devices.
The sensors are assigned to the functional units via an event/action assignment on the sensor.	All sensors and all actuators are assigned on the functional unit.	The overview is significantly improved. It is easier to see which sensors and actuators are affecting the functional unit.

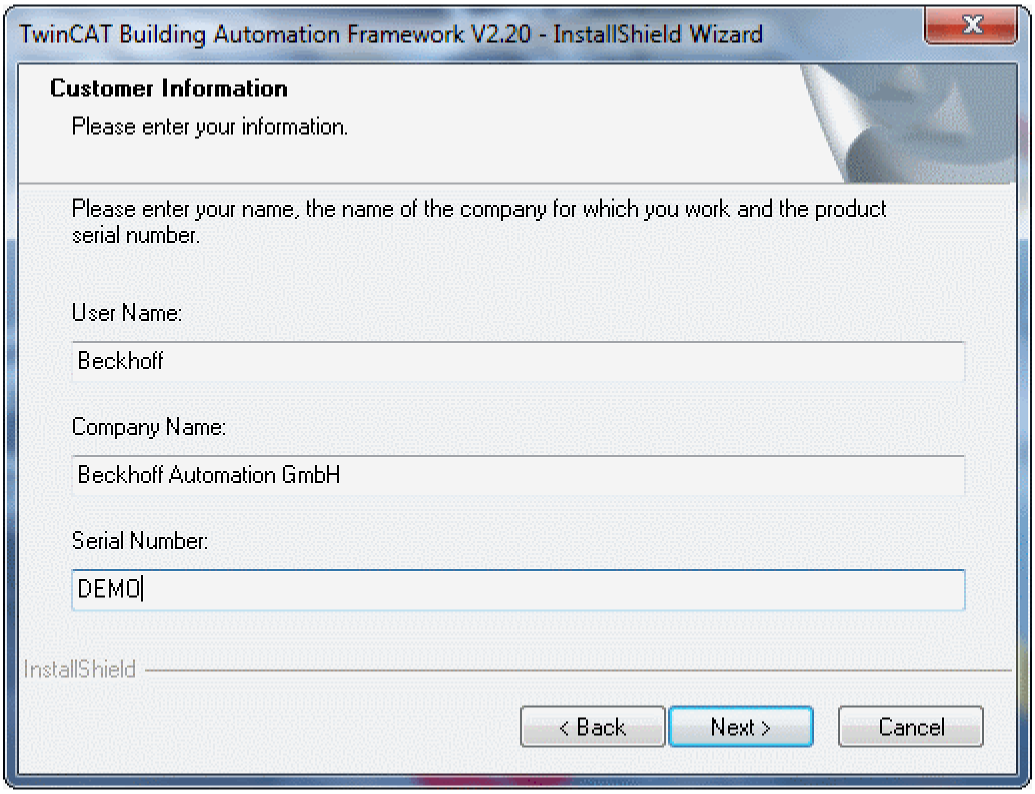
5 Setup

System requirements

- Version of Beckhoff TwinCAT PLC (V2.11 or higher)
- .NET Framework (V4.0 or higher)
- The TcBAFramework.exe - file, located on the Beckhoff Software CD within the TwinCAT Supplement Installations.

Installation

1. Execute the *.exe file of the TwinCAT Supplement.
2. Click *Next*, read the license agreement carefully, accept it and click *Next*.
3. Insert user name, company name and serial number.



TwinCAT Building Automation Framework V2.20 - InstallShield Wizard

Customer Information

Please enter your information.

Please enter your name, the name of the company for which you work and the product serial number.

User Name: Beckhoff

Company Name: Beckhoff Automation GmbH

Serial Number: DEMO

   If you do not have a serial number you can type DEMO to install the evaluation version.
   Accept it by clicking *Next*.

4. Click *Install* to start the installation process.
5. In the end click *Finish* to complete the installation.

After the installation the TwinCAT System folder in the Windows Start-up menu contains a new subfolder named TwinCAT Building Automation Framework V2.20.
This folder includes the TwinCAT Building Automation Manager.

Uninstallation

1. Go to the Control Panel and select *Programs and Features*.
2. Select TwinCAT Building Automation Framework V2.20 and click *Uninstall*.
3. Click *Yes* to start the uninstallation process.
4. After the uninstallation process click *Finish* to complete the removal of the Building Automation Framework V2.20.

6 First Steps

In this sample a project for a room containing a lamp and blinds is created with the help of the Building Automation Manager

Hardware

The following hardware is needed:

- CX1010 (embedded PC)
- KL1104 (4-channel digital input terminal, 24 V DC) → 1 button for the lamp and 2 buttons for the blinds
- KL9260 (power supplier, 230V AC) → power supply for the following two terminals
- KL2751 (1-channel universal dimmer, 230 V AC) → output for the dimmer lamp
- KL2722 (2-channel triac output terminal, 230 V AC) → output for the blinds
- KL9010 (end terminal)

⚠ CAUTION

Operating high voltage

Be careful when working with 230V devices. Wrong usage may easily cause injuries, damages of the device or short-circuits. Let only qualified personnel work with these devices and read the corresponding hardware documentation.

Setting up the PLC

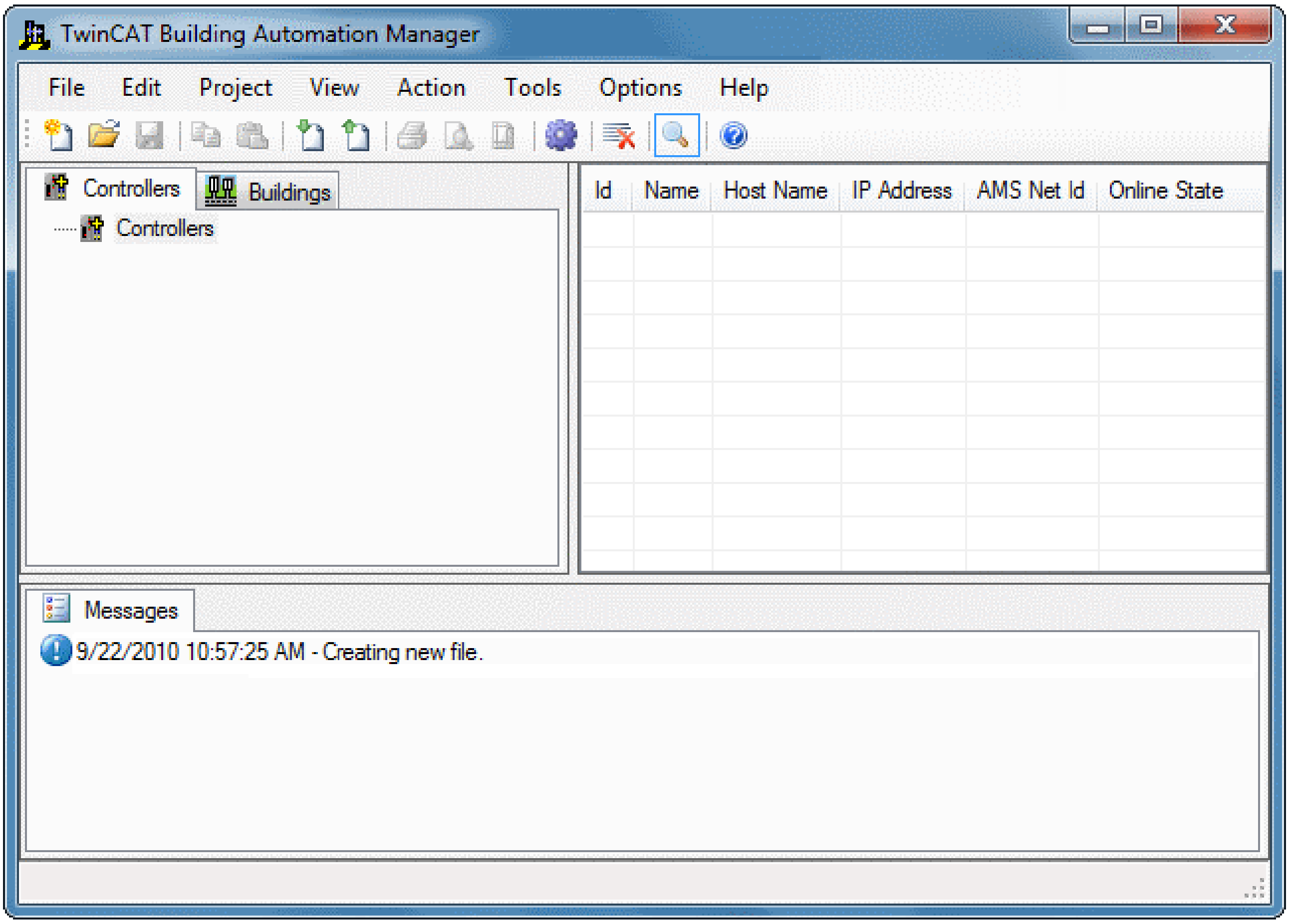
1. Register the CX1010 as a remote computer for TwinCAT.
2. The local TwinCAT program must run in Start or Config mode.
3. Create a folder for your project.
4. Copy the Building Automation PLC project to it (usually located in *C:\TwinCAT\BAFrameworkV2.20\BAManager\Samples\FirstSteps*).
   Make sure the project is NOT read-only.
5. Open the Building Automation PLC project with TwinCAT PLC Control.
6. Select *PC or CX (x86)* as the target system in PLC Configuration
7. Login to the CX1010 (which causes compiling and downloading of the program).
8. Set this project as the boot project for this CX1010.
9. Start the PLC.
10. Save the project. Make sure that the *.tpy file is now in your project folder

NOTICE

Read the documentation of PLC Control if you are not confirmed with its usage. It is recommended to have taken part in a TwinCAT training.

Using the Building Automation Manager

To start the application go to *Start → All Programs → TwinCAT System → TwinCAT Building Automation Framework → TwinCAT Building Automation Manager*.
The Building Automation Manager should look like this:

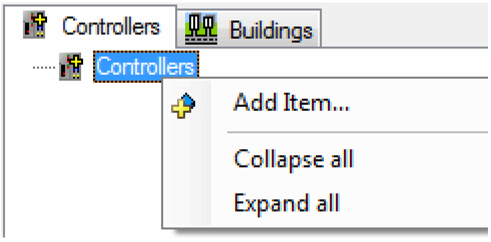


9/22/2010 10:57:25 AM - Creating new file.

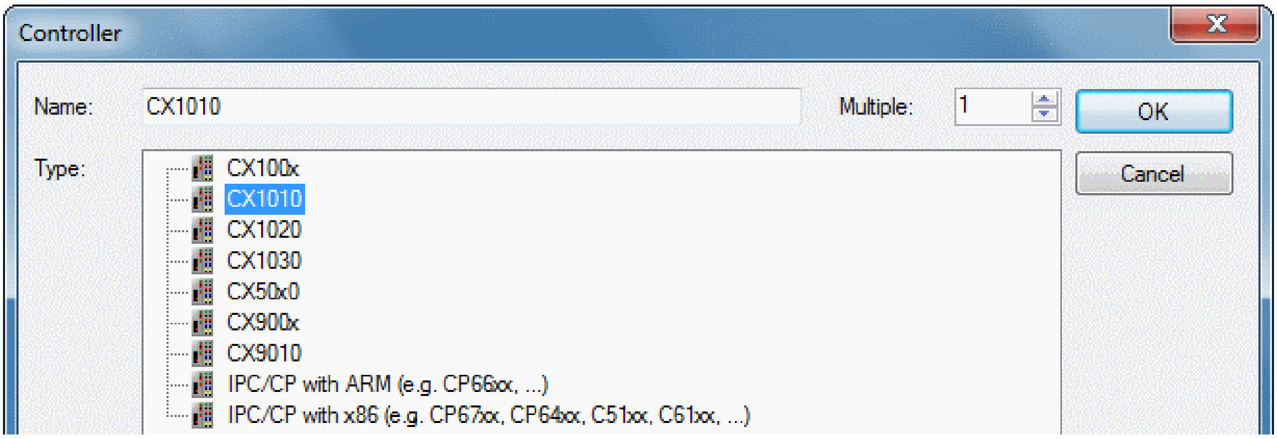
If the BA Manager does not look like this please click on File → New.

Adding the hardware

First add the controller. Right-click on *Controllers* in the *Controllers* tab and select *Add Item...*.



The following dialog will appear:



Controller

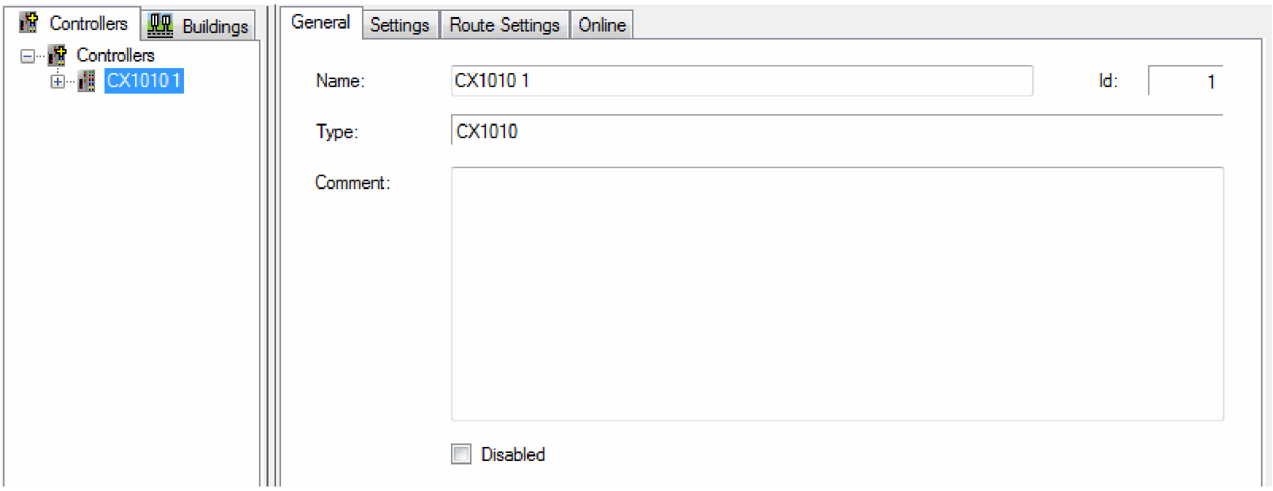
Name: CX1010

Multiple: 1

Type:

- CX100x
- CX1010
- CX1020
- CX1030
- CX50x0
- CX900x
- CX9010
- IPC/CP with ARM (e.g. CP66xx, ...)
- IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...)

This dialog is always used when you want to append elements. It lists up all possible elements you can append at this position.
If you insert no name then a default name is used. The name can be changed afterwards as well.
We now want to add just one CX1010, so you have to choose the *CX1010*.



Name: CX1010 1

Id: 1

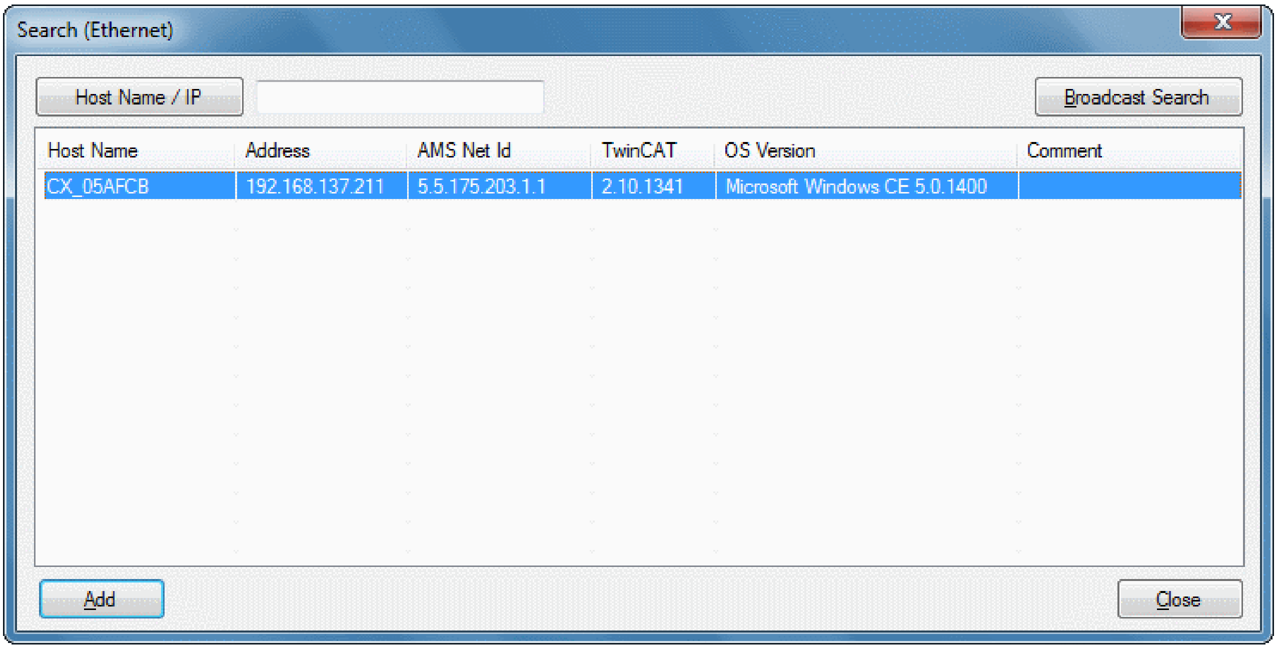
Type: CX1010

Comment:

□ Disabled

Now you can see the general information about the new *CX1010*. On this page you can change the name of the element, see the internal id and the type, add comments or disable the element.

To establish the connection select the *Settings* tab and click on the *Search (Ethernet)...* button.



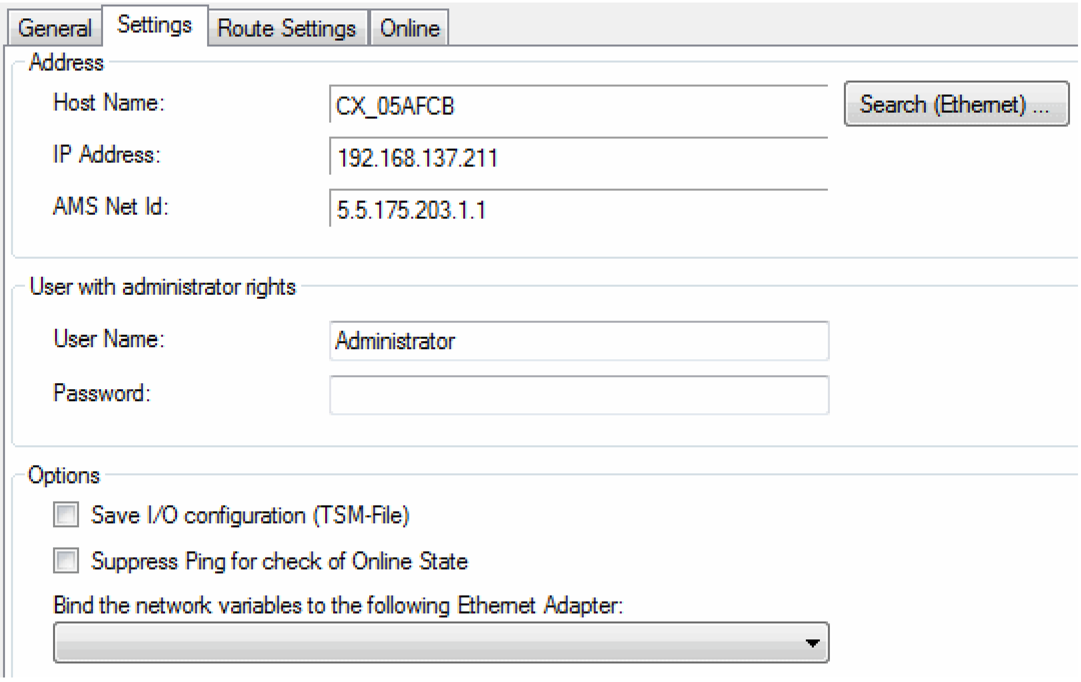
Host Name	Address	AMS Net Id	TwinCAT	OS Version	Comment
CX_05AFCB	192.168.137.211	5.5.175.203.1.1	2.10.1341	Microsoft Windows CE 5.0.1400	

There are now two different ways to find your CX1010 in the network:

1. Enter the Hostname or IP of the CX1010 and click on *Host Name / IP*.
2. Click *Broadcast Search*. Thereby all devices in the network are listed.

Select your device and click the *Add* button.

The information of the device show up in the *Settings* tab:



Address

Host Name: CX_05AFCB

IP Address: 192.168.137.211

AMS Net Id: 5.5.175.203.1.1

User with administrator rights

User Name: Administrator

Password:

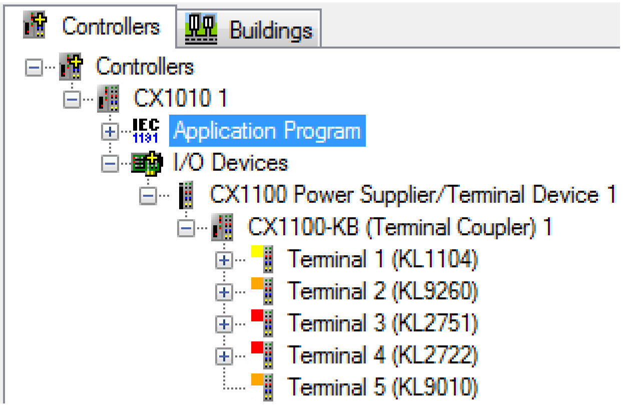
Options

☐ Save I/O configuration (TSM-File)

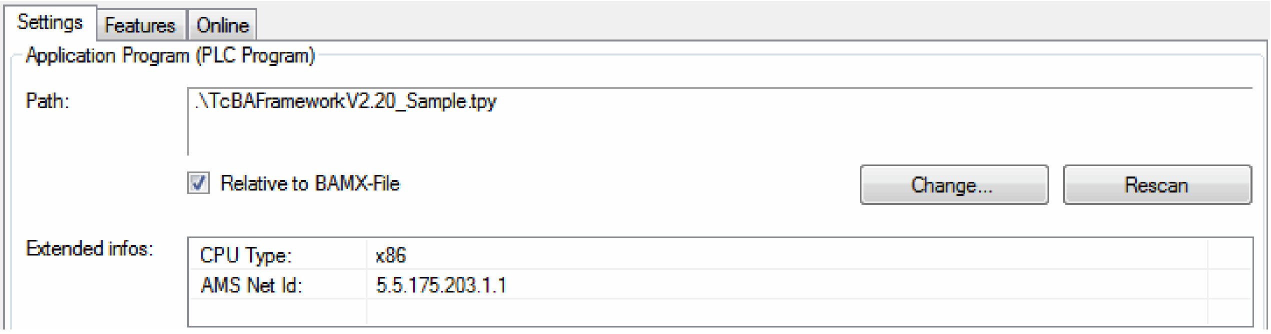
☐ Suppress Ping for check of Online State

Bind the network variables to the following Ethernet Adapter:

Click on the + to the left of the *CX1010 1* node, select *I/O Devices* and append a *CX1100 Power Supplier/ Terminal Device*.
Append a *CX1100-KB (Terminal Coupler)* to *CX1100 Power Supplier/Terminal Device 1*.
Now append all the above mentioned terminals to Coupler *CX1100-KB (Terminal Coupler) 1*. When you're finished it should look like this:



Click on *Application Program* in the tree-view and select the *Settings* tab. The *.tpy file of your Building Automation PLC project has to be entered here. To do so click on *Change...* and open the *.tpy file of your PLC project.



Application Program (PLC Program)

Path: .\TcBAFrameworkV2.20_Sample.tpy

☑ Relative to BAMX-File

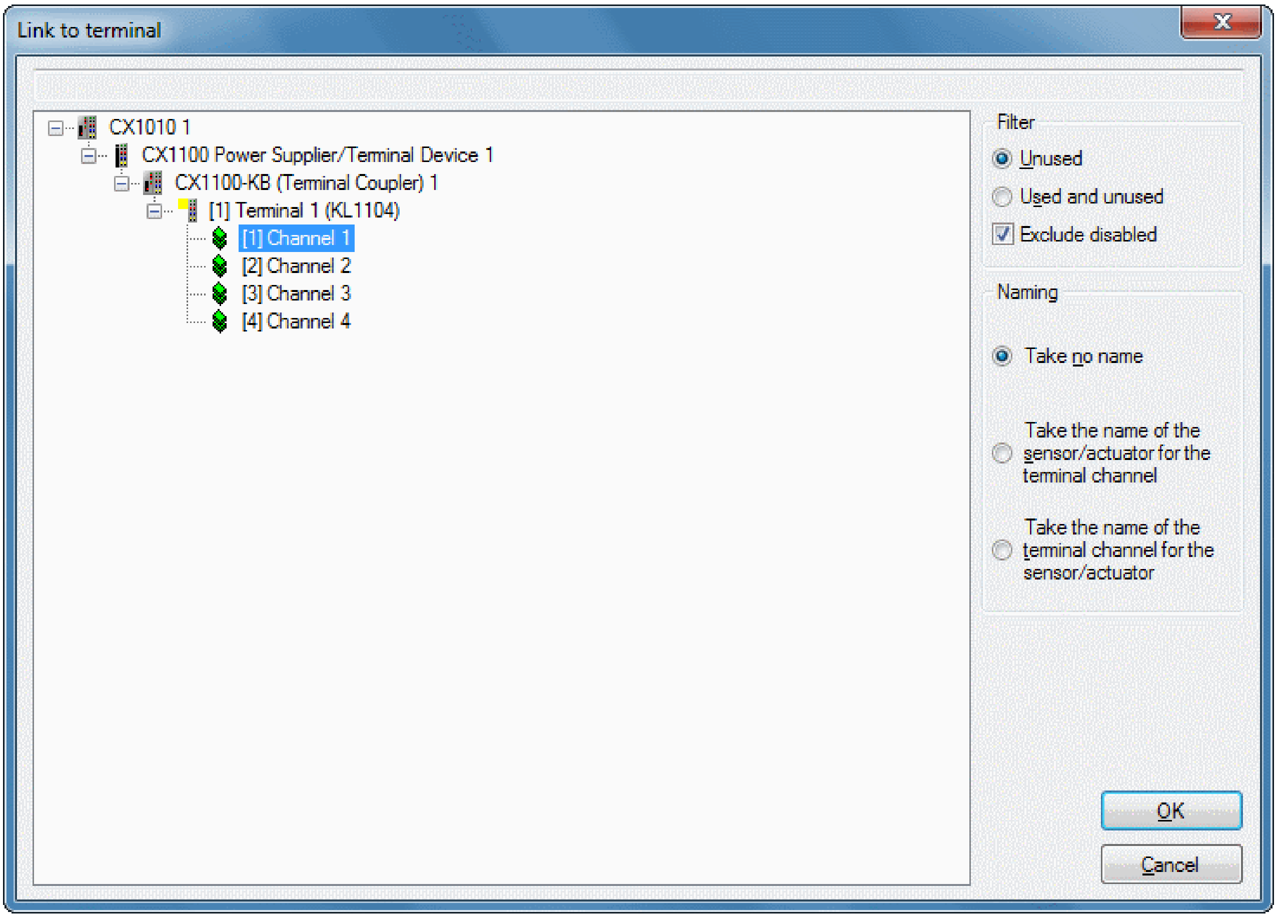
Extended infos:

CPU Type:	x86
AMS Net Id:	5.5.175.203.1.1

It is recommended to have all project-related files in one folder. The Relative to BAMX-File option becomes available, after the project was saved for the first time. It can be done by clicking on File → Save As....

Creating sensors and actuators

Sensors have to be linked to the input-channels and actuators have to be linked to the output-channels of the hardware.
Click on the + left to the *Application Program* node and after that on the + to the left of the *Sensors*. Append 3 *Type 1 - standard Digital Signal* (use the *Multiple* control-box to append them in one step).
Select the *Settings* tab of the first digital signal and click on the *Linked to...* button.



Link to terminal

- CX1010 1
  - CX1100 Power Supplier/Terminal Device 1
    - CX1100-KB (Terminal Coupler) 1
      - [1] Terminal 1 (KL1104)
        - [1] Channel 1
        - [2] Channel 2
        - [3] Channel 3
        - [4] Channel 4

Filter

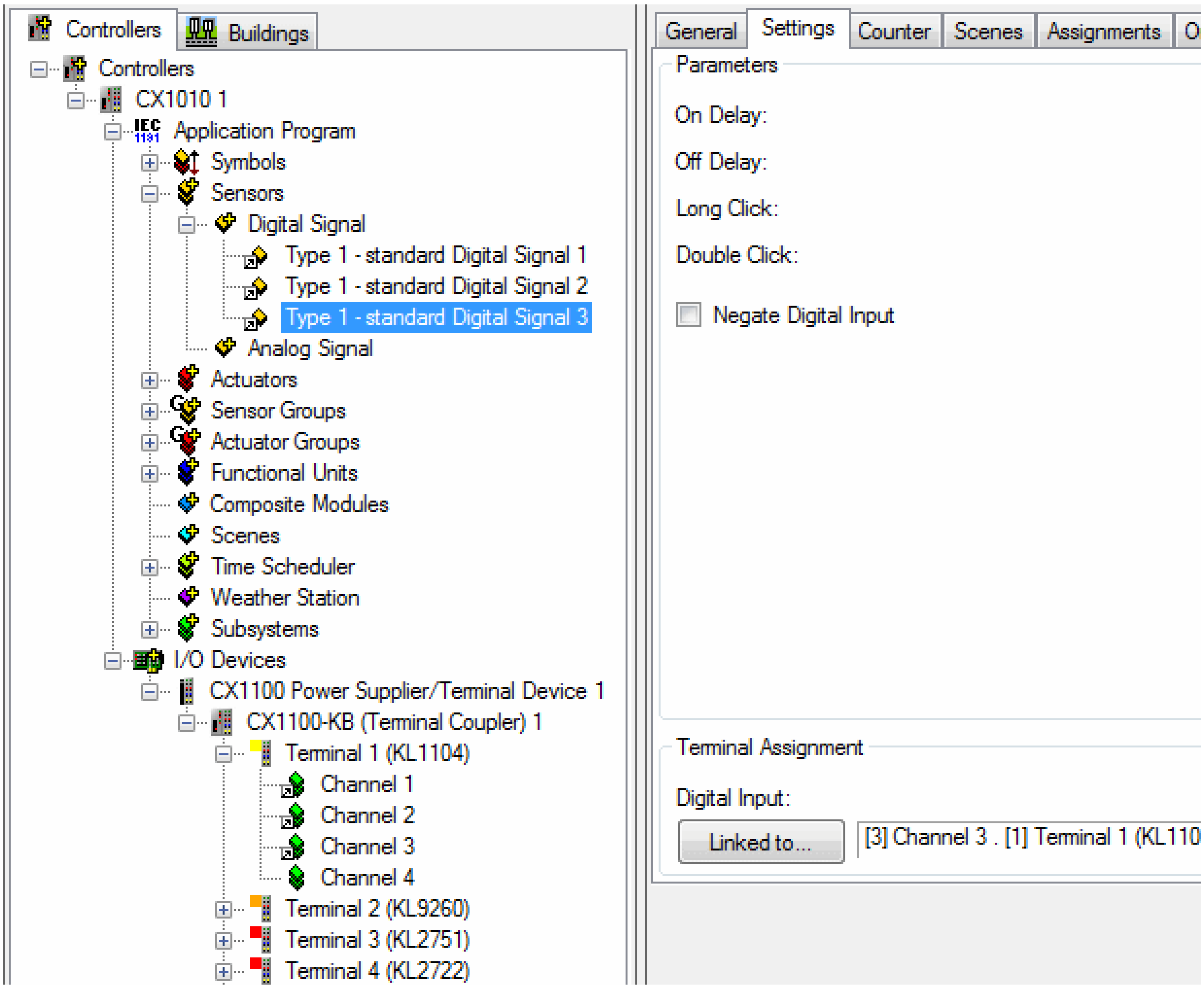
- ● Unused
- ○ Used and unused
- ☑ Exclude disabled

Naming

- ● Take no name
- ○ Take the name of the sensor/actuator for the terminal channel
- ○ Take the name of the terminal channel for the sensor/actuator

Link to the first channel of the KL1104.

Repeat this for the other digital signals and link them to channel 2 and 3 of the KL1104 terminal.



Parameters

On Delay:

Off Delay:

Long Click:

Double Click:

□ Negate Digital Input

Terminal Assignment

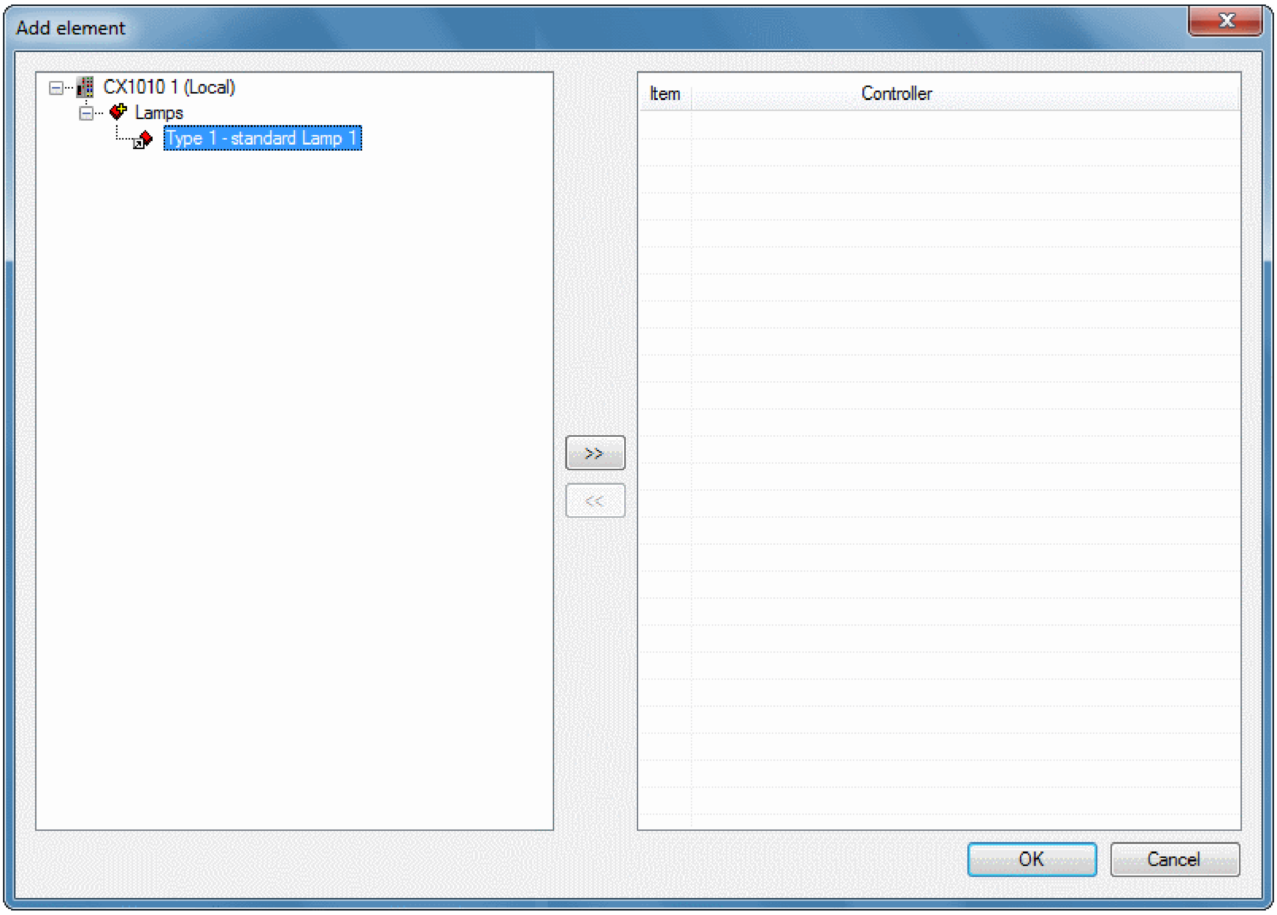
Digital Input: [3] Channel 3 . [1] Terminal 1 (KL110

Expand *Actuators* and append a *Type 1 - standard Lamp*. Link the *Analog Output* of the lamp to channel 1 of the KL2751.Append a *Type 1 - standard Blind Drive* and link *Digital Output Up* to channel 1 and *Digital Output Down* to channel 2 of the KL2722.

Setting up groups and functional units

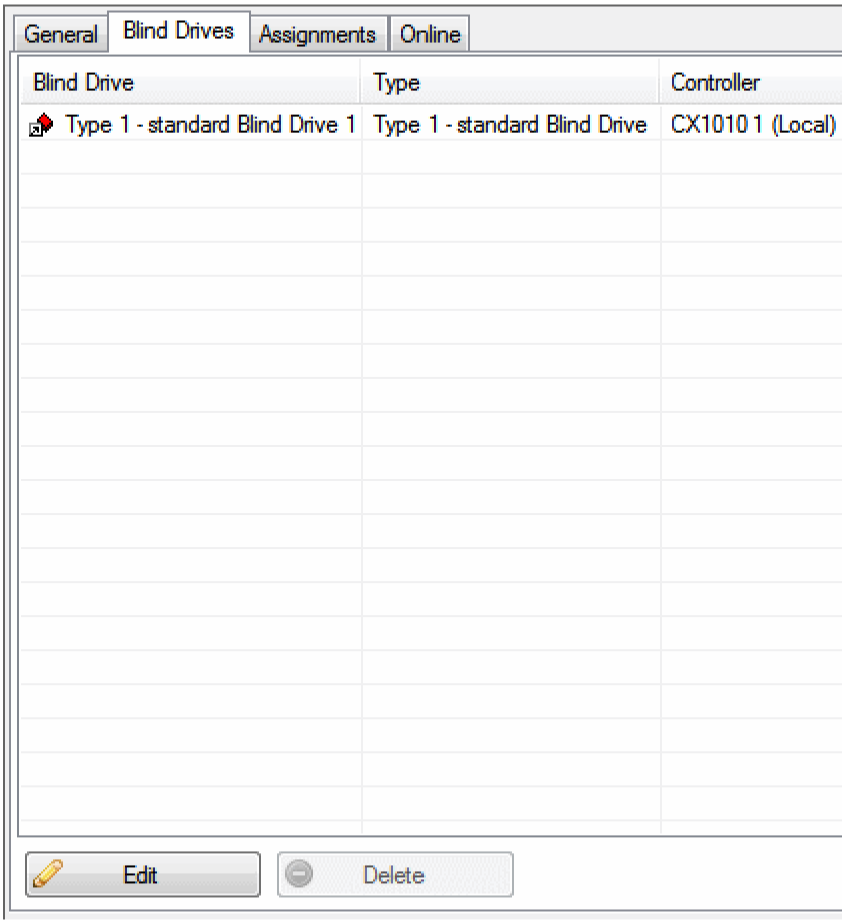
To be able to add functionality, every sensor and actuator has to be placed inside of a group. Actuator groups have to be called up from a functional unit.

Expand *Actuator Groups* and append a *Type 1 - standard Lamp Group*. Select the *Lamps* tab and click on the *Edit* button.
Select *Type 1 - standard Lamp 1* and click >> (or alternatively double-click on *Type 1 - standard Lamp 1*).



Now the element moves to the right side to indicate that it is selected.

Repeat this the same way with the blind drive group.

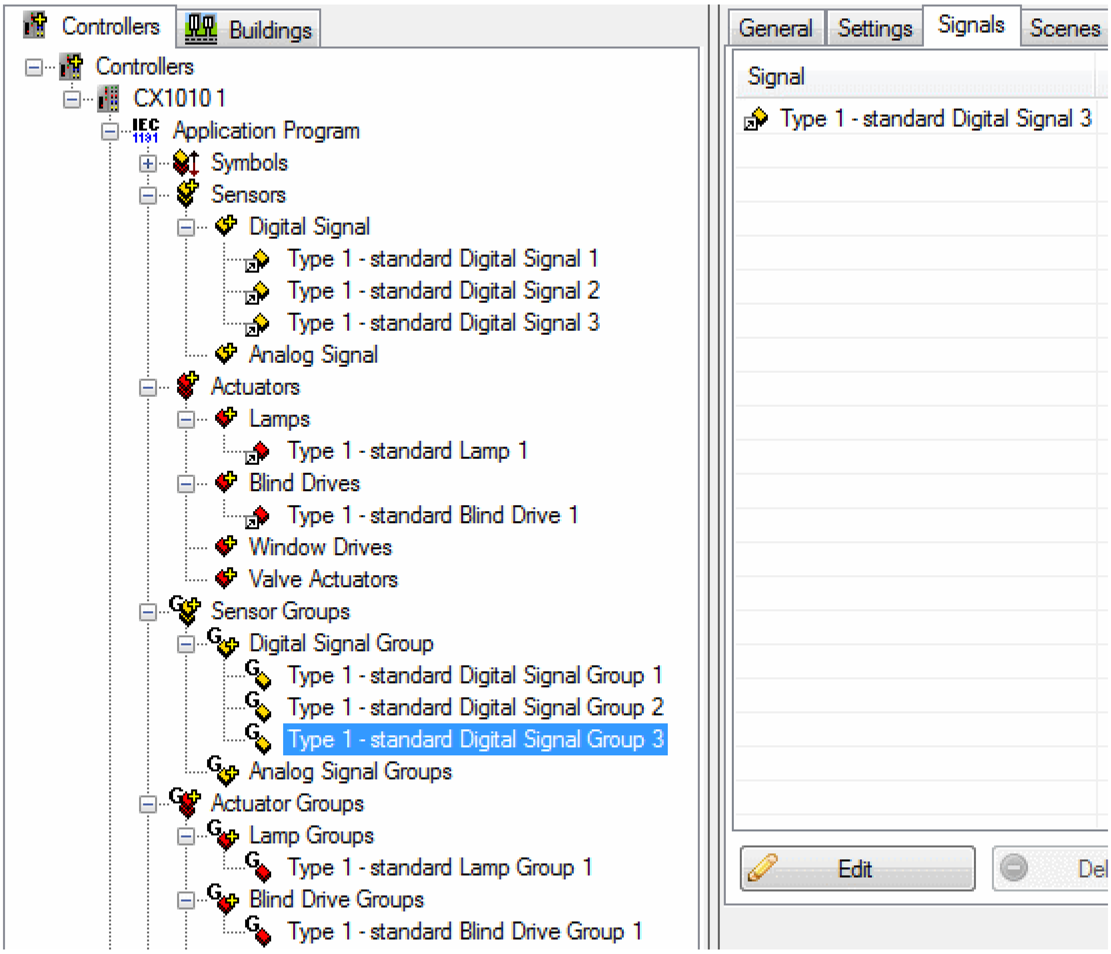


Blind Drive	Type	Controller
Type 1 - standard Blind Drive 1	Type 1 - standard Blind Drive	CX1010 1 (Local)

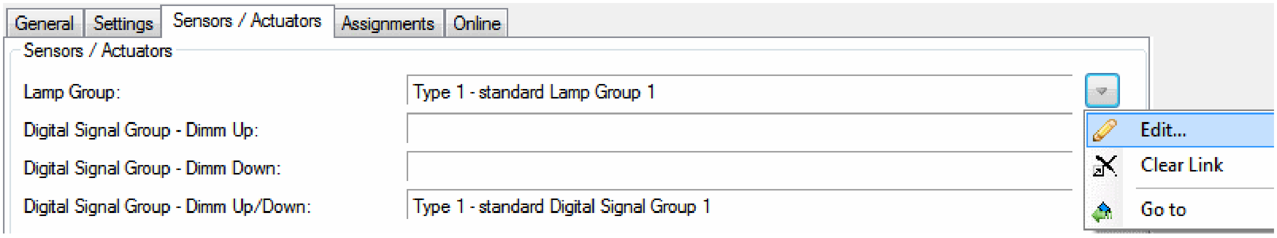
Creating sensor groups

The sensor groups provide the ability to set up and customize functionalities. They contain the information which action of which functional unit is called up when something happens to a sensor.

Expand *Sensor Groups* and append 3 *Type 1 - standard Digital Signal Group*. Select the *Signals* tab of the 1st (2nd, 3rd) group and add the 1st (2nd, 3rd) digital signal.


Controllers
Buildings
Controllers
CX1010 1
Application Program
Symbols
Sensors
Digital Signal
Type 1 - standard Digital Signal 1
Type 1 - standard Digital Signal 2
Type 1 - standard Digital Signal 3
Analog Signal
Actuators
Lamps
Type 1 - standard Lamp 1
Blind Drives
Type 1 - standard Blind Drive 1
Window Drives
Valve Actuators
Sensor Groups
Digital Signal Group
Type 1 - standard Digital Signal Group 1
Type 1 - standard Digital Signal Group 2
Type 1 - standard Digital Signal Group 3
Analog Signal Groups
Actuator Groups
Lamp Groups
Type 1 - standard Lamp Group 1
Blind Drive Groups
Type 1 - standard Blind Drive Group 1
General
Settings
Signals
Scenes
Signal
Type 1 - standard Digital Signal 3
Edit
Del

Expand *Functional Units* and append a *Type 1 - Dimmer Switch* unit. Select the lamp group you created before in the *Sensors / Actuators* tab of this new unit by clicking on *Edit*. Choose the *Type 1 - standard Digital Signal Group 1* for *Dimm Up/Down* by clicking on *Edit*.

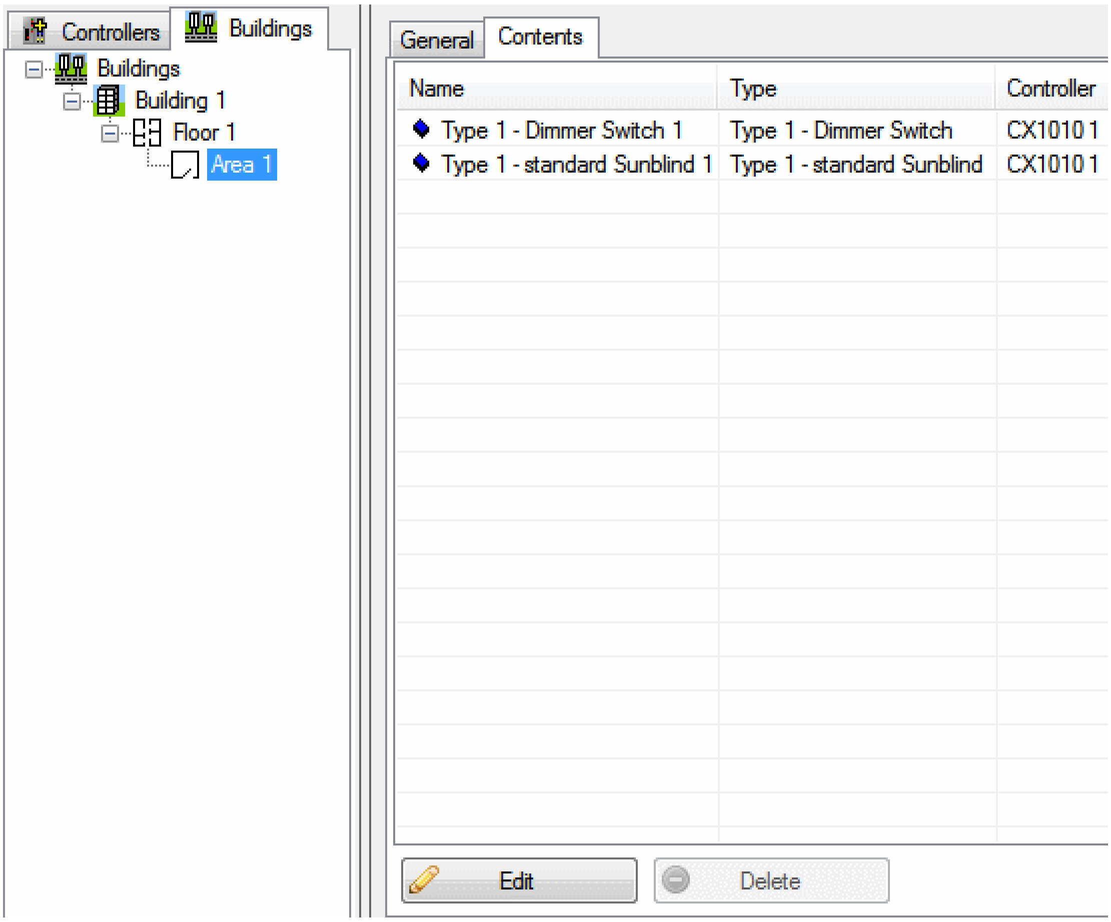


Lamp Group:	Type 1 - standard Lamp Group 1
Digital Signal Group - Dimm Up:	
Digital Signal Group - Dimm Down:	
Digital Signal Group - Dimm Up/Down:	Type 1 - standard Digital Signal Group 1

Repeat this process for the sunblind. Choose *Type 1 - standard Digital Signal Group 2* for drive *Up* and 3 for drive *Down*.

Designing the building

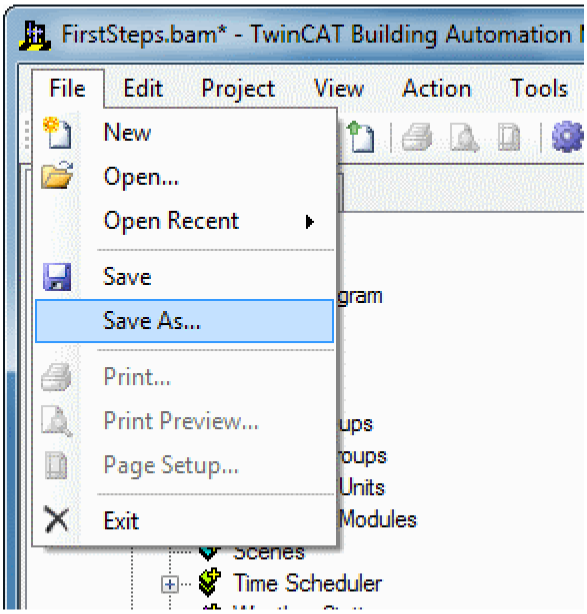
The Building Automation Manager allows you to design the whole structure of your building and add the corresponding functional units to it. Select the *Buildings* tab and append a Building with a floor and an area. Select the *Contents* tab of the *Area 1* and add the *Type 1 - Dimmer Switch 1* and *Type 1 - standard Sunblind 1* functional units.



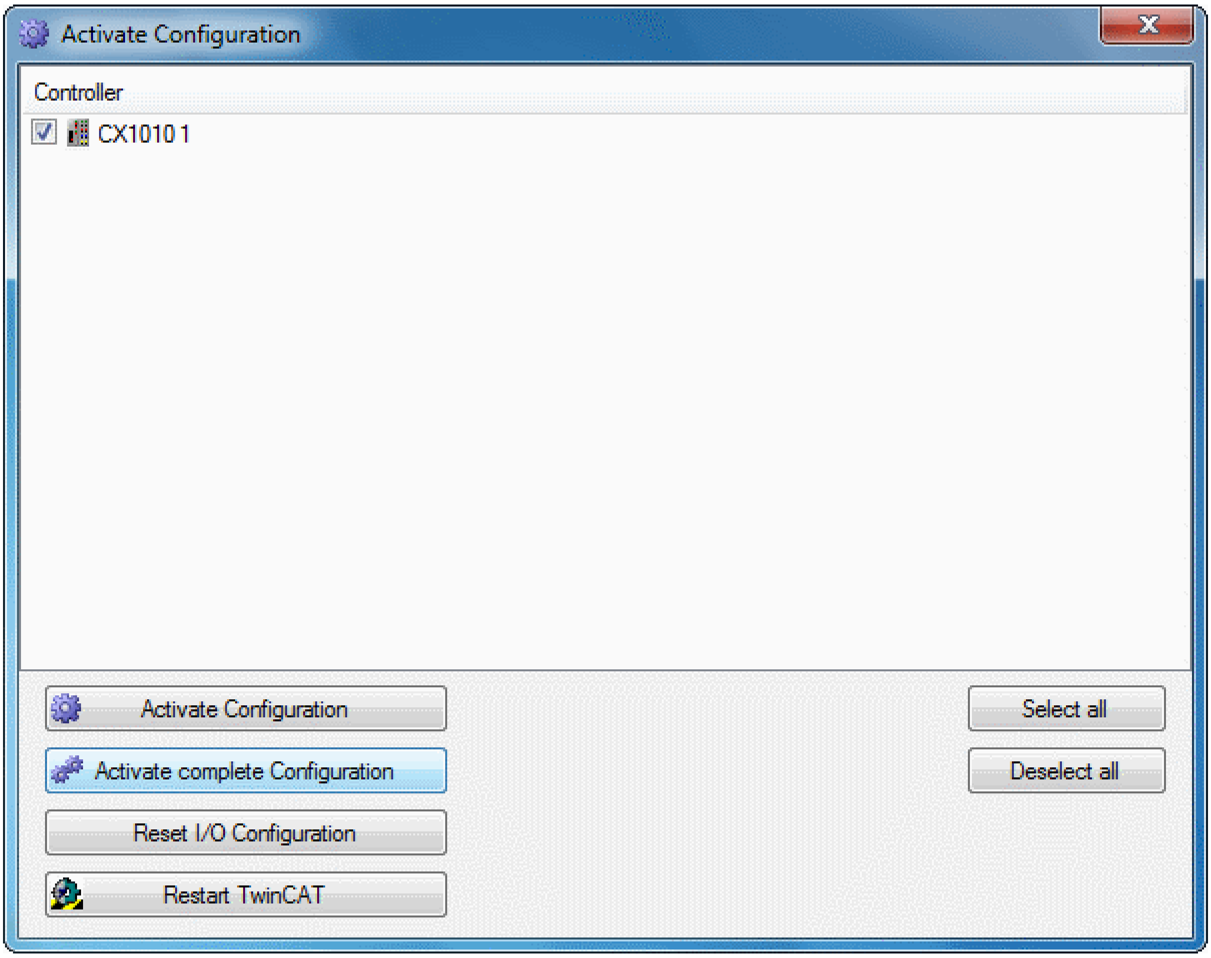
Name	Type	Controller
Type 1 - Dimmer Switch 1	Type 1 - Dimmer Switch	CX1010 1
Type 1 - standard Sunblind 1	Type 1 - standard Sunblind	CX1010 1

Activate the configuration

First, save the project again by clicking on *File* → *Save As...*.



The *Activate Configuration* dialog can be accessed via the menu *Action* → *Activate Configuration*. Select the *CX1010 1* and click on *Activate complete Configuration*.



Now wait until the Building Automation Manager has finished the activation of the configuration. If no warning or error appears in the *Messages* field at the bottom of the BA Manager everything is working perfectly.

Congratulations, you have created your first and fully functional Building Automation Project!

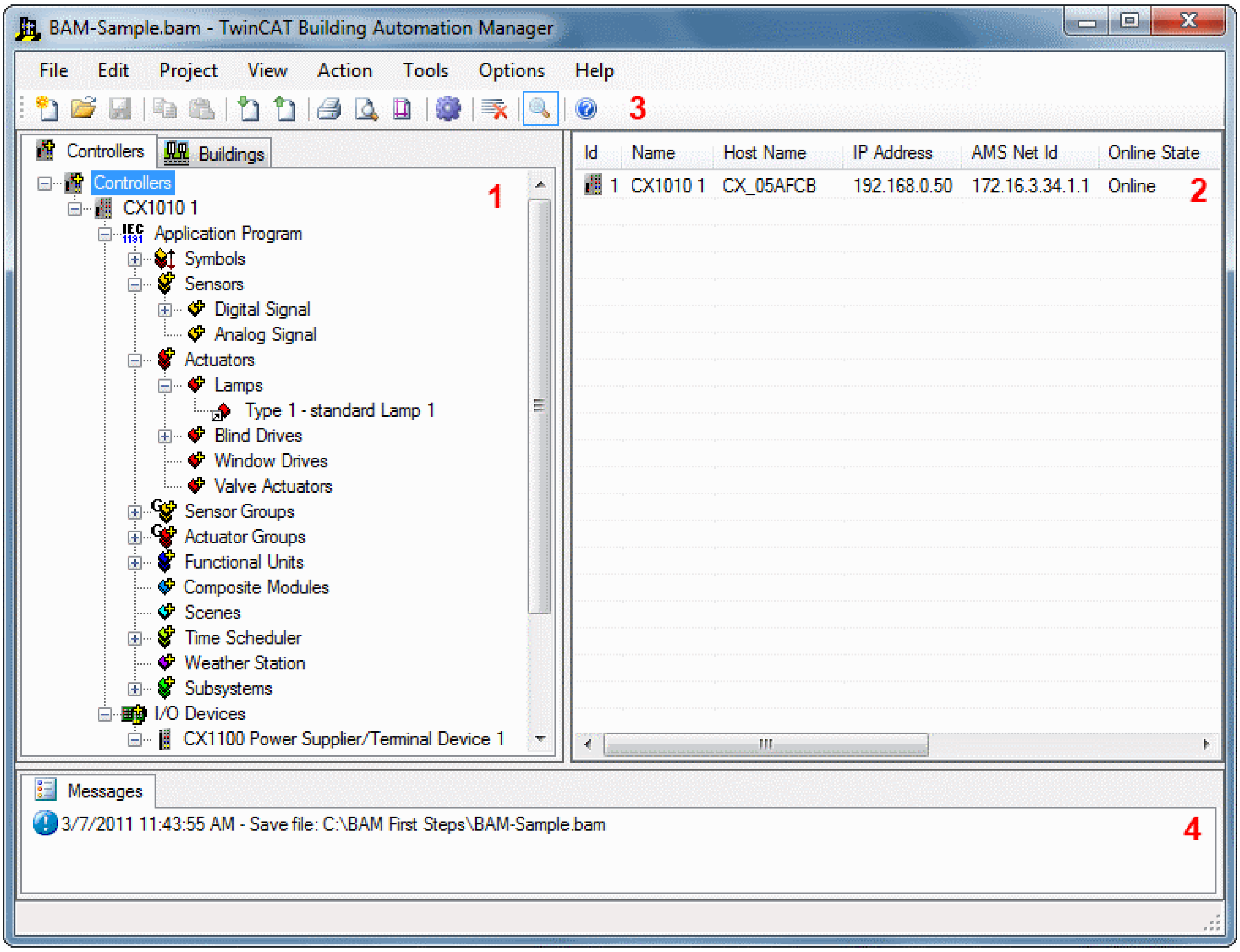
7 Building Automation Manager

The Building Automation Manager is used to design and configure your building automation solutions. In addition the whole structure of buildings can be implemented. All this is done within a very visual and self-explaining user interface.

7.1 Control Elements

7.1.1 Main Window

The main window of the Building Automation Manager is structured into 4 main areas, which are explained below.



Id	Name	Host Name	IP Address	AMS Net Id	Online State
1	CX1010 1	CX_05AFCB	192.168.0.50	172.16.3.34.1.1	Online

3/7/2011 11:43:55 AM - Save file: C:\BAM First Steps\BAM-Sample.bam

1 Navigation Tree

On the left side of the main window the navigation tree can be found.
Click here [▶ 29] to get a detailed description of this element.

2 Information and Settings Panel

The large control on the right side of the main window is used to configure and set up the element selected in the navigation tree.
Information about the individual options of an element can be found in the specific documentation page of this element.

3 Toolbar

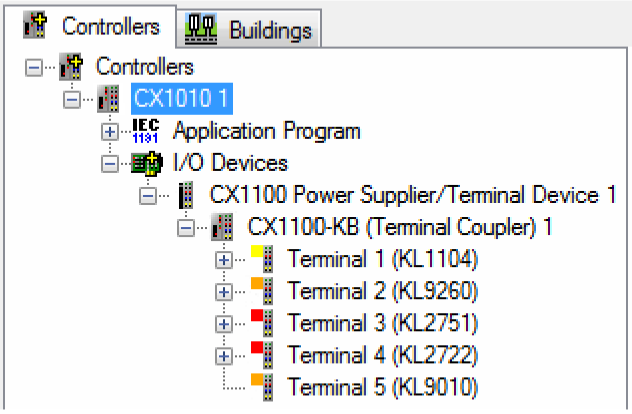
With the toolbar at the top of the main window you can open, save, and create new projects. More functions are available in a specific context.
To access the options window click on *Options* → *Settings*. You can set the language and other behaviors of the Building Automation Manager there.

4 Message Window

At the bottom of the main window the message window is displayed.
A detailed description of this control can be found here [▶ 30]

7.1.2 Navigation Tree

The navigation tree gives access to the whole system configuration and all of its settings.

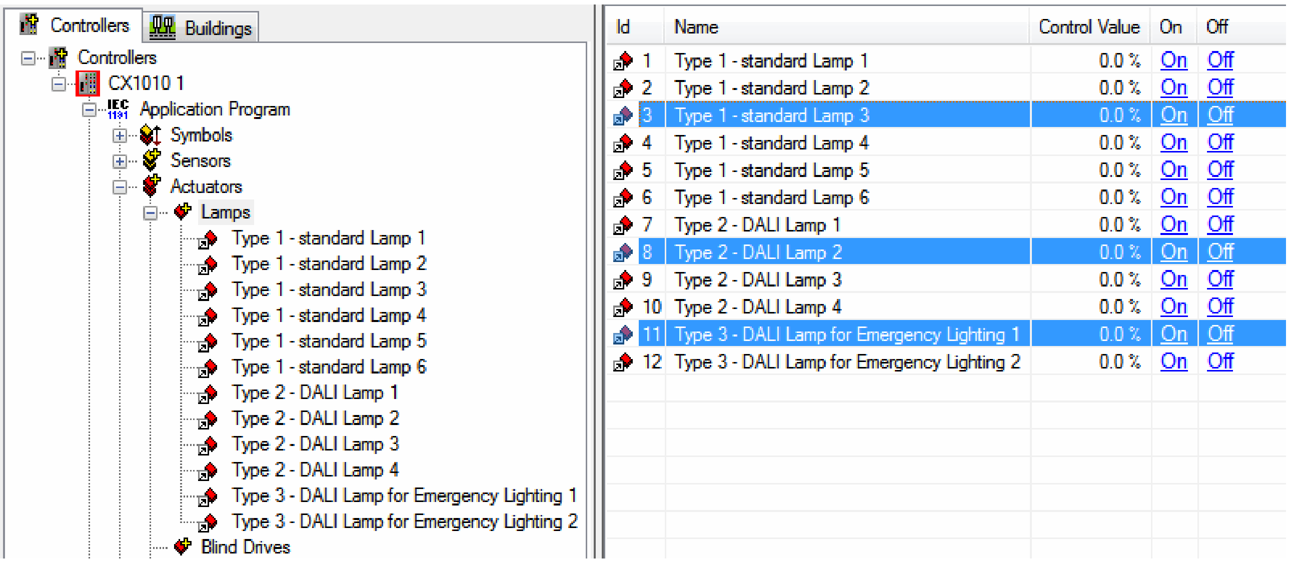


Controllers | Buildings

- Controllers
  - CX1010 1
    - Application Program
    - I/O Devices
      - CX1100 Power Supplier/Terminal Device 1
        - CX1100-KB (Terminal Coupler) 1
          - Terminal 1 (KL1104)
          - Terminal 2 (KL9260)
          - Terminal 3 (KL2751)
          - Terminal 4 (KL2722)
          - Terminal 5 (KL9010)

A hierarchical system (like the file system seen in the Windows Explorer) is used to display this information. If a node is selected, the settings panel of this control will appear on the right side of the main window. Information about the individual options of a node displayed in the settings panel can be found in the specific documentation page of this element. To add a sub node right-click on the parent node and select *Add Item...* in the following dialog the item to add has to be specified. You can configure the name, the type and the quantum. To delete a node right-click on it and select *Delete*. Note that some elements cannot be deleted.

Controllers Tab



Controllers | Buildings

- Controllers
  - CX1010 1
    - Application Program
      - Symbols
      - Sensors
      - Actuators
        - Lamps
          - Type 1 - standard Lamp 1
          - Type 1 - standard Lamp 2
          - Type 1 - standard Lamp 3
          - Type 1 - standard Lamp 4
          - Type 1 - standard Lamp 5
          - Type 1 - standard Lamp 6
          - Type 2 - DALI Lamp 1
          - Type 2 - DALI Lamp 2
          - Type 2 - DALI Lamp 3
          - Type 2 - DALI Lamp 4
          - Type 3 - DALI Lamp for Emergency Lighting 1
          - Type 3 - DALI Lamp for Emergency Lighting 2
        - Blind Drives

Id	Name	Control Value	On	Off
1	Type 1 - standard Lamp 1	0.0 %	On	Off
2	Type 1 - standard Lamp 2	0.0 %	On	Off
3	Type 1 - standard Lamp 3	0.0 %	On	Off
4	Type 1 - standard Lamp 4	0.0 %	On	Off
5	Type 1 - standard Lamp 5	0.0 %	On	Off
6	Type 1 - standard Lamp 6	0.0 %	On	Off
7	Type 2 - DALI Lamp 1	0.0 %	On	Off
8	Type 2 - DALI Lamp 2	0.0 %	On	Off
9	Type 2 - DALI Lamp 3	0.0 %	On	Off
10	Type 2 - DALI Lamp 4	0.0 %	On	Off
11	Type 3 - DALI Lamp for Emergency Lighting 1	0.0 %	On	Off
12	Type 3 - DALI Lamp for Emergency Lighting 2	0.0 %	On	Off

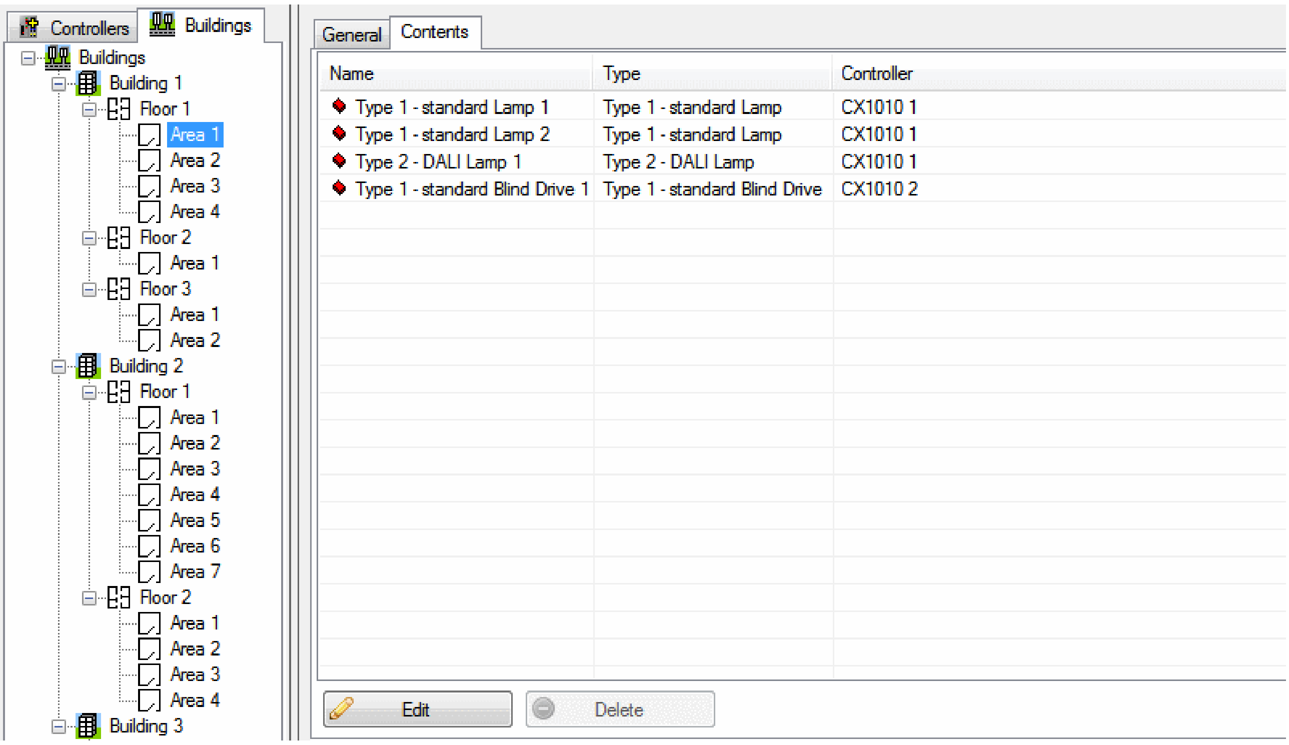
This tab contains the hardware you are using and the functionality the hardware should provide. A multitude of controllers can be inserted. Each controller consists of the *Application Program* and the *I/O Devices*. The application program contains sensors (digital and analogue) and actuators (lamps, blinds and window drives,

valve actuators ...). Sensors and actors must be placed into groups to add functionality. The functionality is defined and configured in the *Functional Units* node.
Subsystems like DALI and EnOcean are supported as well (to be found in node *Subsystems*).

In the *I/O Devices* node the used hardware has to be defined. Similar to the TwinCAT System Manager the device with its couplers and its terminals can be configured here. The hardware inputs and outputs then can be linked to the sensors and actuators. Changes that need to be activated, will be indicated by the controller highlighted in red.

By selecting a category, the information window on the right side lists all elements regardless of their folder-affiliation. Elements can be sorted, renamed or deleted. The deletion of elements is multi-select supported. The category also provides printing options for the list available via the right-click menu.

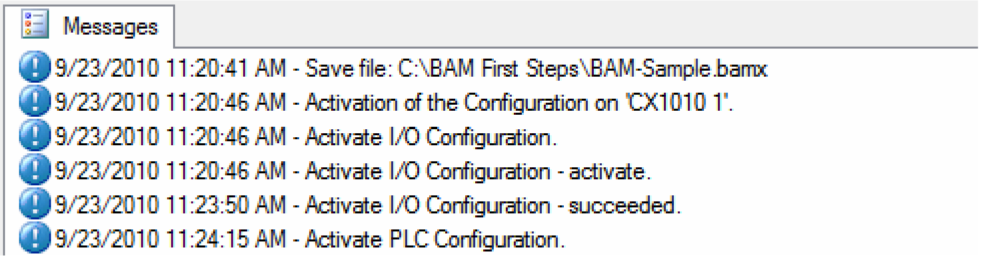
Buildings Tab



Name	Type	Controller
Type 1 - standard Lamp 1	Type 1 - standard Lamp	CX1010 1
Type 1 - standard Lamp 2	Type 1 - standard Lamp	CX1010 1
Type 2 - DALI Lamp 1	Type 2 - DALI Lamp	CX1010 1
Type 1 - standard Blind Drive 1	Type 1 - standard Blind Drive	CX1010 2

The building structure in the Buildingstab offers the option to assign all created elements (e.g. lamps, blinds, lighting units etc.) locally according to building, storey and room. In addition, the contents list of a room permits direct access to the onlinemenu of the respective selected element.
It would also be conceivable to use the information from the building structure for tools, such as for the report function (Tools → Reports...).

7.1.3 Message Window



```
9/23/2010 11:20:41 AM - Save file: C:\BAM First Steps\BAM-Sample.bamx
9/23/2010 11:20:46 AM - Activation of the Configuration on 'CX1010 1'.
9/23/2010 11:20:46 AM - Activate I/O Configuration.
9/23/2010 11:20:46 AM - Activate I/O Configuration - activate.
9/23/2010 11:23:50 AM - Activate I/O Configuration - succeeded.
9/23/2010 11:24:15 AM - Activate PLC Configuration.
```

The *Message Window* gives information about actions the Building Automation Manager is performing. There are 3 types of messages:

- **Notifications:** These messages just provide basic information. If you only see notifications everything is working perfectly.

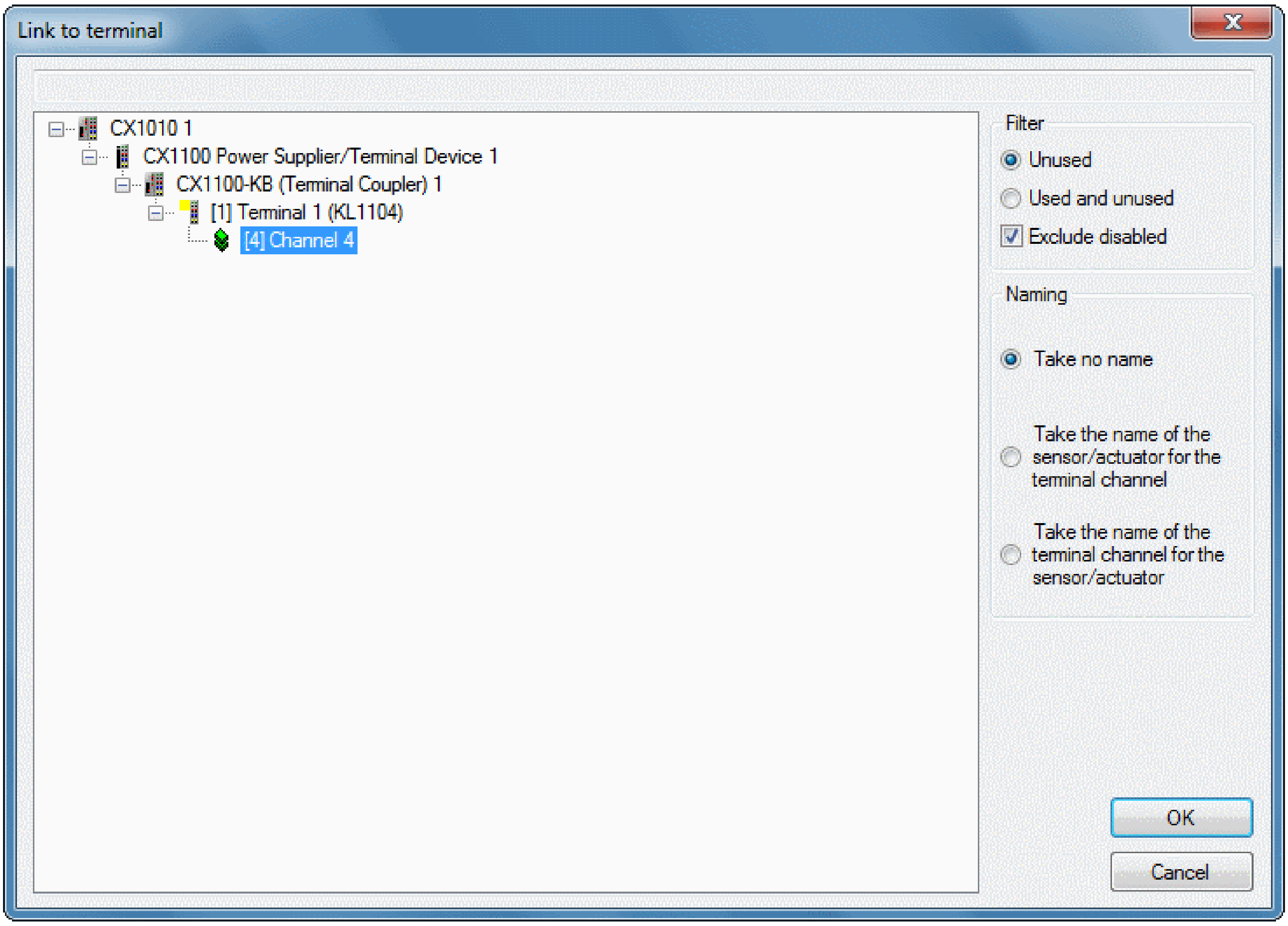
- **Warnings:** Warnings are mostly harmless but may in some cases be harmful. You should always try to avoid them.
- **Errors:** Errors show that a serious problem is present. Your project is not able to work when errors occur.

It also provides a right-click menu with following options:

- **Clear:** Deletes all messages within the *Message Window*.
- **Open:** Replaces existing messages with those from the file.
- **Save:** Saves all messages to a *.xml file.
- **Close:** Hides the *Message Window*.

7.1.4 Link Dialog

The link dialog is called whenever a sensor or actuator is to be linked with a terminal channel or a PLC variable with a terminal variable. This is normally done by clicking on a sensor, actuator or symbol in the navigation tree, selecting the *Settings* tab and clicking *Linked to...* for the respective input or output.



Link to terminal

- CX1010 1
  - CX1100 Power Supplier/Terminal Device 1
    - CX1100-KB (Terminal Coupler) 1
      - [1] Terminal 1 (KL1104)
        - [4] Channel 4

Filter

● Unused
○ Used and unused
☑ Exclude disabled

Naming

● Take no name
○ Take the name of the sensor/actuator for the terminal channel
○ Take the name of the terminal channel for the sensor/actuator

The possible links are shown in a hierarchical tree structure (similar to Windows Explorer).

Only channels that are compatible with the sensor/actuator with which they are to be linked are shown. For symbols only terminal variables with the same bit size are available for selection.

To select a link left-click on it in the tree (this will highlight it in colour). Click *OK* to accept the link. If you do not wish to (re-)link the sensor/actuator/symbol click *Cancel*.

The *Filter* option can be used to select whether all suitable terminal channels are to be displayed or whether all already linked and/or blocked (deactivated) channels should be hidden.

You can also influence the naming of the linked elements. The options are to retain both names, transfer the name of the sensor/actuator to the channel, or transfer the name of the channel to the sensor/actuator. **This option is not available for symbols!**

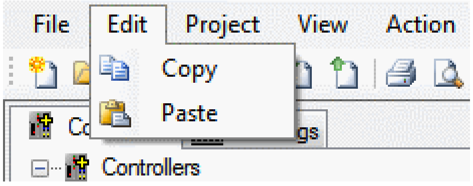
7.1.5 Edit

7.1.5.1 Copy and Paste

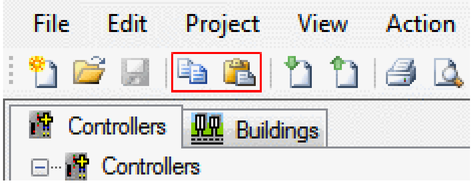
Copy and *Paste* provide the option to copy entire TwinCAT Building Automation Manager elements. First of all the component to be copied must be marked in the navigation tree. The *Copy* and *Paste* functions are then called. A copy of the element can now be found in the navigation tree.

Apart from the well-known Windows shortcuts *Ctrl* + *C* (copy) and *Ctrl* + *V* (paste), these functions can be called by three further paths.

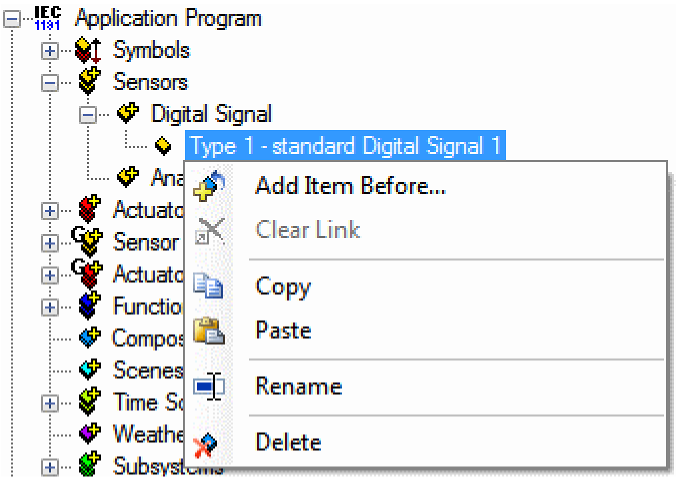
The first possibility is offered by the menu bar in the *Edit* option. Here also, the element is simply marked in the navigation tree, *Edit* -> *Copy* is selected and *Paste* is subsequently called again.



A further possibility is offered by the tool bar using the corresponding symbols. If this is not visible, it can be displayed via *View* → *Tool Bar*.



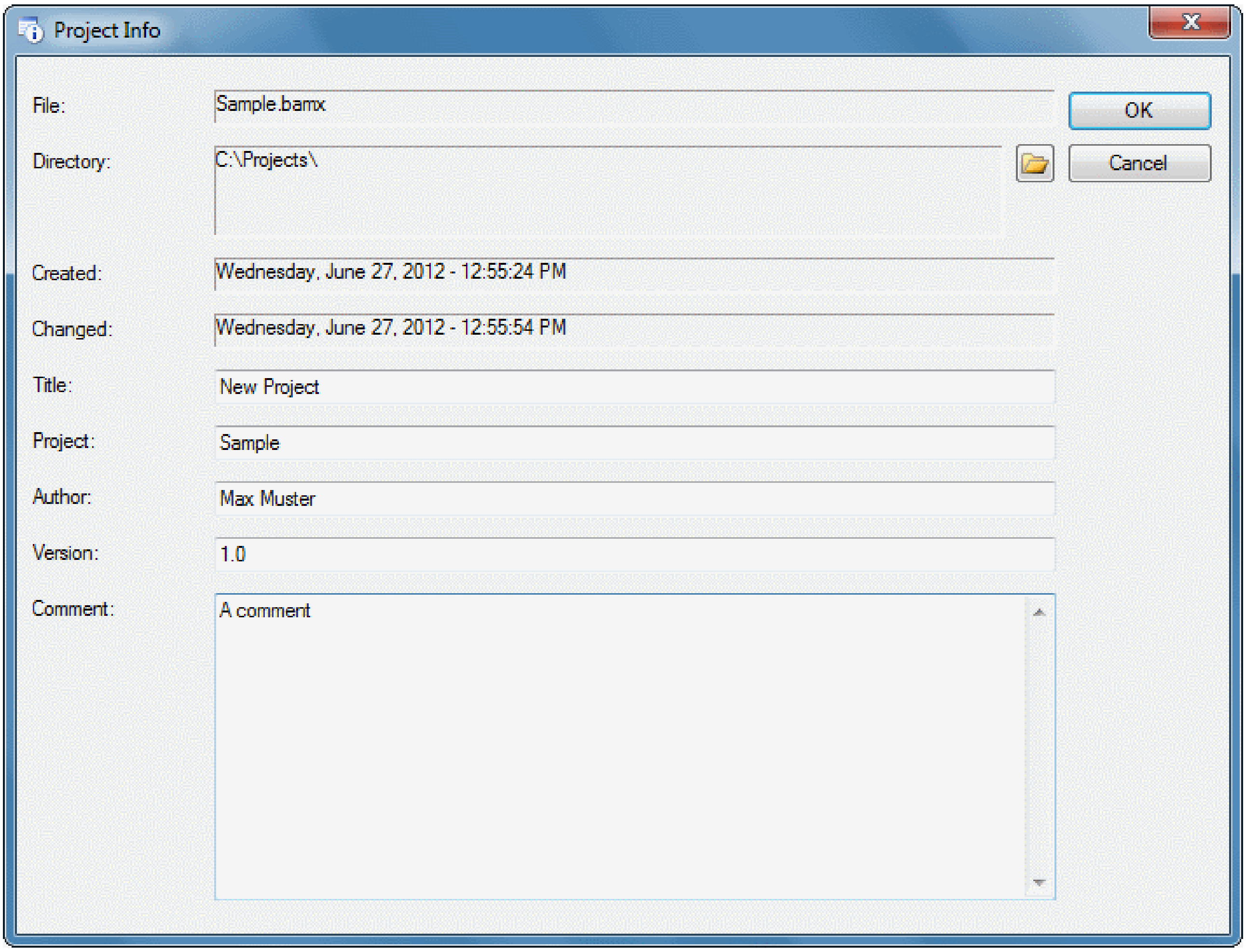
As the final variant, Copy and Paste are available via the context menu of the mouse.



7.1.6 Project

7.1.6.1 Info

The project can be provided with additional information in this dialog. After saving the project, you can open the directory of the *.bamx file by click on the folder symbol.



File: Sample.bamx

Directory: C:\Projects\

Created: Wednesday, June 27, 2012 - 12:55:24 PM

Changed: Wednesday, June 27, 2012 - 12:55:54 PM

Title: New Project

Project: Sample

Author: Max Muster

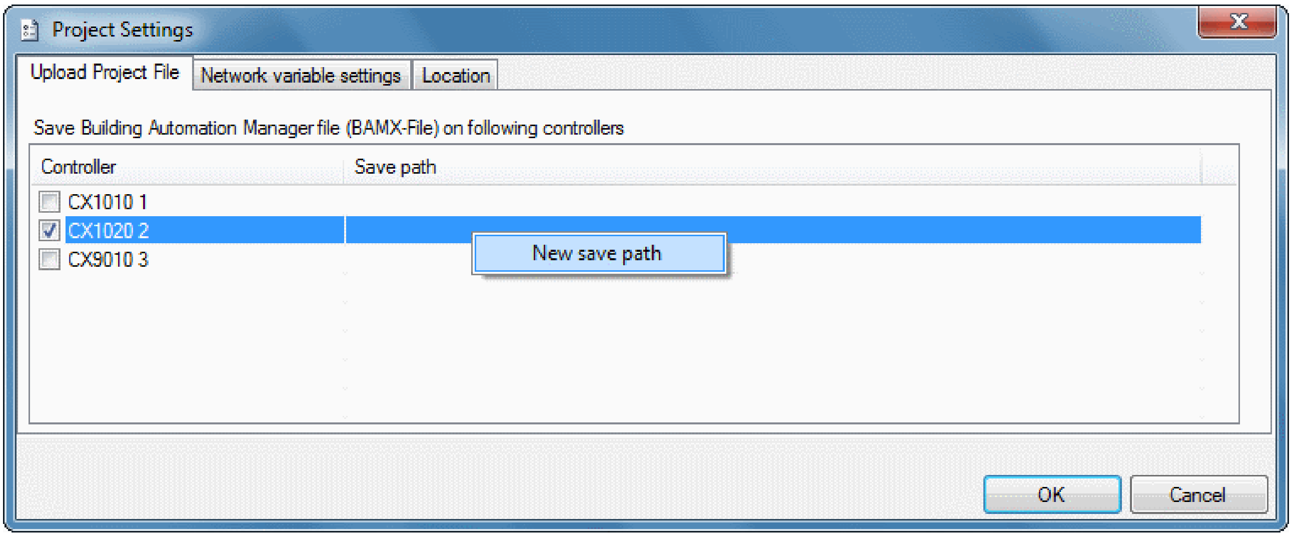
Version: 1.0

Comment: A comment

7.1.6.2 Settings

Further project settings can be made here.

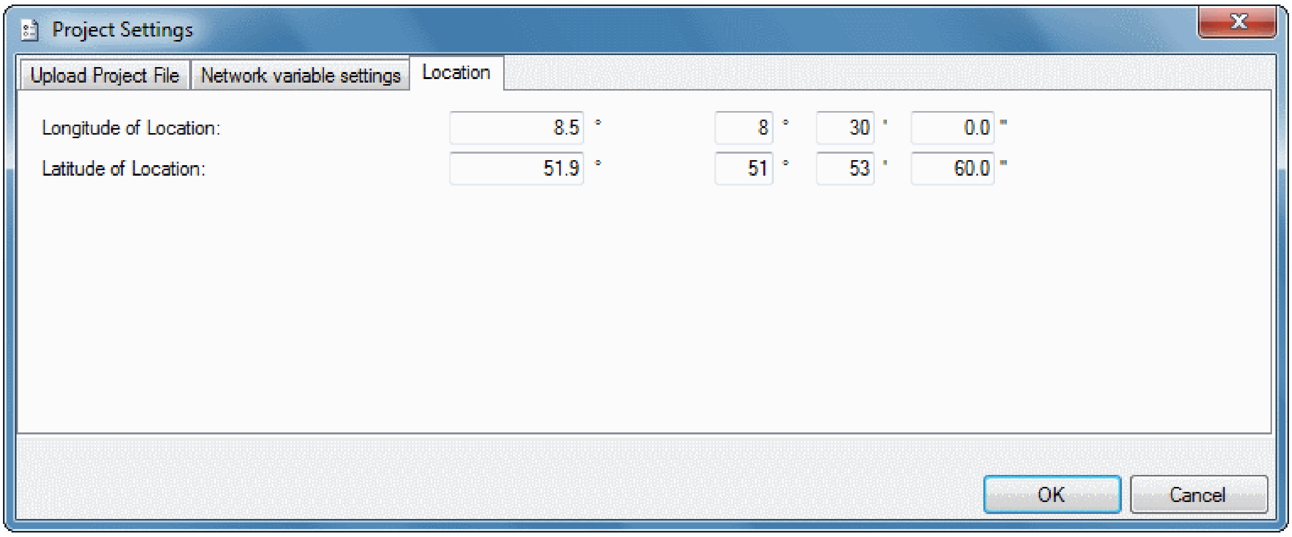
On activation of the configuration, the project file is loaded to all the controllers selected in this dialog. The path for this is specified by right clicking with the mouse in the corresponding line. You can choose between the boot directory and a self-defined path. Make sure that the folder already exists on the target device!



Save Building Automation Manager file (BAMX-File) on following controllers

Controller	Save path
□ CX1010 1	
☑ CX1020 2	
□ CX9010 3	

Geographical data for the project can be entered here.



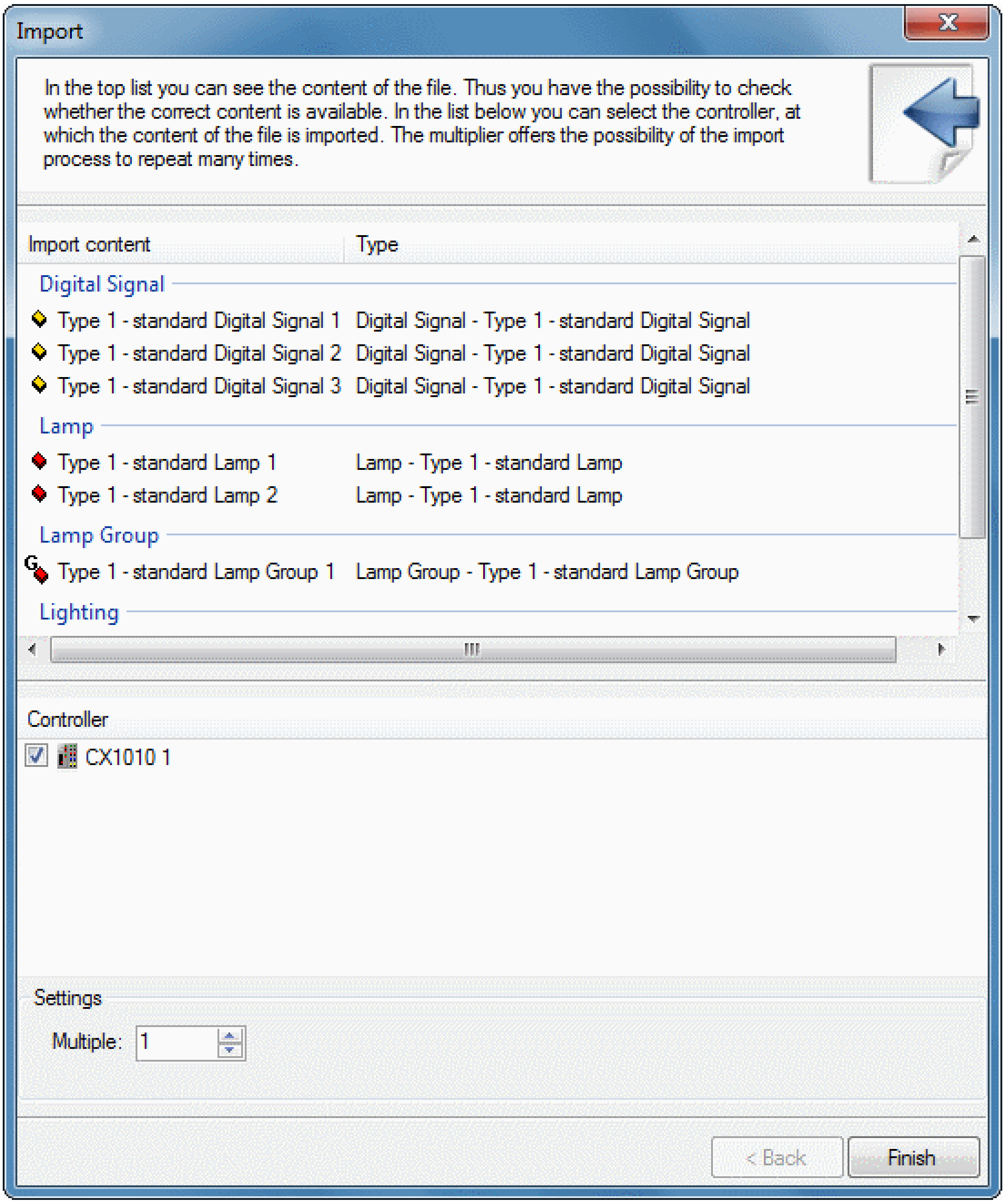
Longitude of Location:	8.5 °	8 °	30 '	0.0 "
Latitude of Location:	51.9 °	51 °	53 '	60.0 "

The latitude is specified as the distance of a place on the surface of the earth from the equator to the north or to the south in degrees. The latitude can assume a value from 0° (at the equator) to ±90° (at the poles). A positive sign thereby indicates a northern direction and a negative sign a southern direction. The longitude is an angle that can assume values up to ±180° starting from the prime meridian 0° (an artificially determined North-South line). A positive sign indicates a longitude in an eastern direction and a negative sign in a western direction.

7.1.6.3 Import

Imports *.bamexp files that were created with the export function [▶ 35] of the TwinCAT Building Automation Manager.

After calling the import function and selecting the export file, you can examine the elements that it contains once again, depending on the content.



In the top list you can see the content of the file. Thus you have the possibility to check whether the correct content is available. In the list below you can select the controller, at which the content of the file is imported. The multiplier offers the possibility of the import process to repeat many times.

Import content	Type
Digital Signal	
Type 1 - standard Digital Signal 1	Digital Signal - Type 1 - standard Digital Signal
Type 1 - standard Digital Signal 2	Digital Signal - Type 1 - standard Digital Signal
Type 1 - standard Digital Signal 3	Digital Signal - Type 1 - standard Digital Signal
Lamp	
Type 1 - standard Lamp 1	Lamp - Type 1 - standard Lamp
Type 1 - standard Lamp 2	Lamp - Type 1 - standard Lamp
Lamp Group	
Type 1 - standard Lamp Group 1	Lamp Group - Type 1 - standard Lamp Group
Lighting	

Controller

☑ CX1010 1

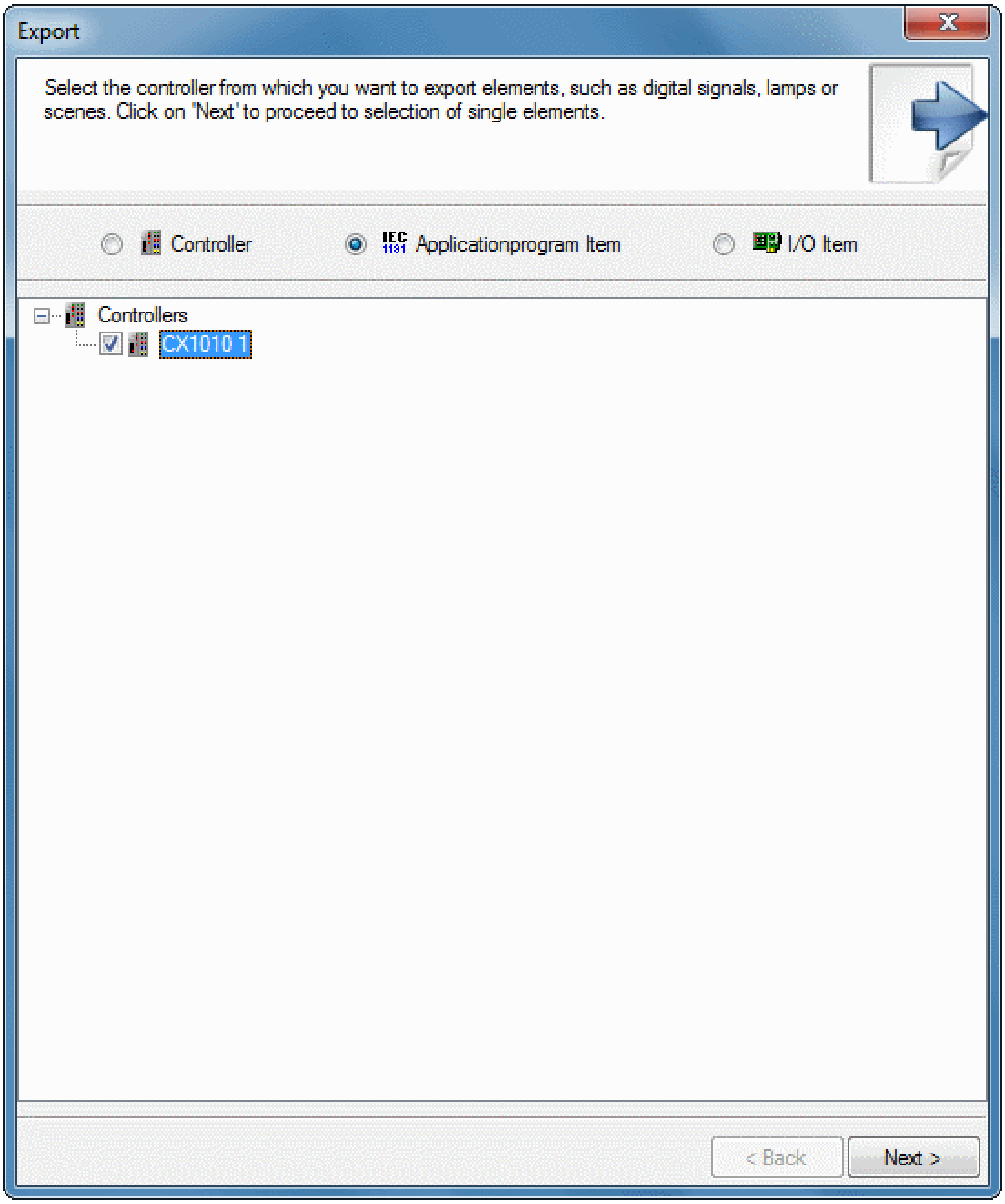
Settings

Multiple: 1

7.1.6.4 Export

Exports a selection of TwinCAT Building Automation Manager elements including parameters, but without references, to a *.bamexp export file. The contents can then be imported [▸ 34] into other TwinCAT Building Automation Manager projects, taking into account the type compatibility.

First of all the type of content is selected. The entire controller, parts of the application program or parts of the I/O devices can be exported.



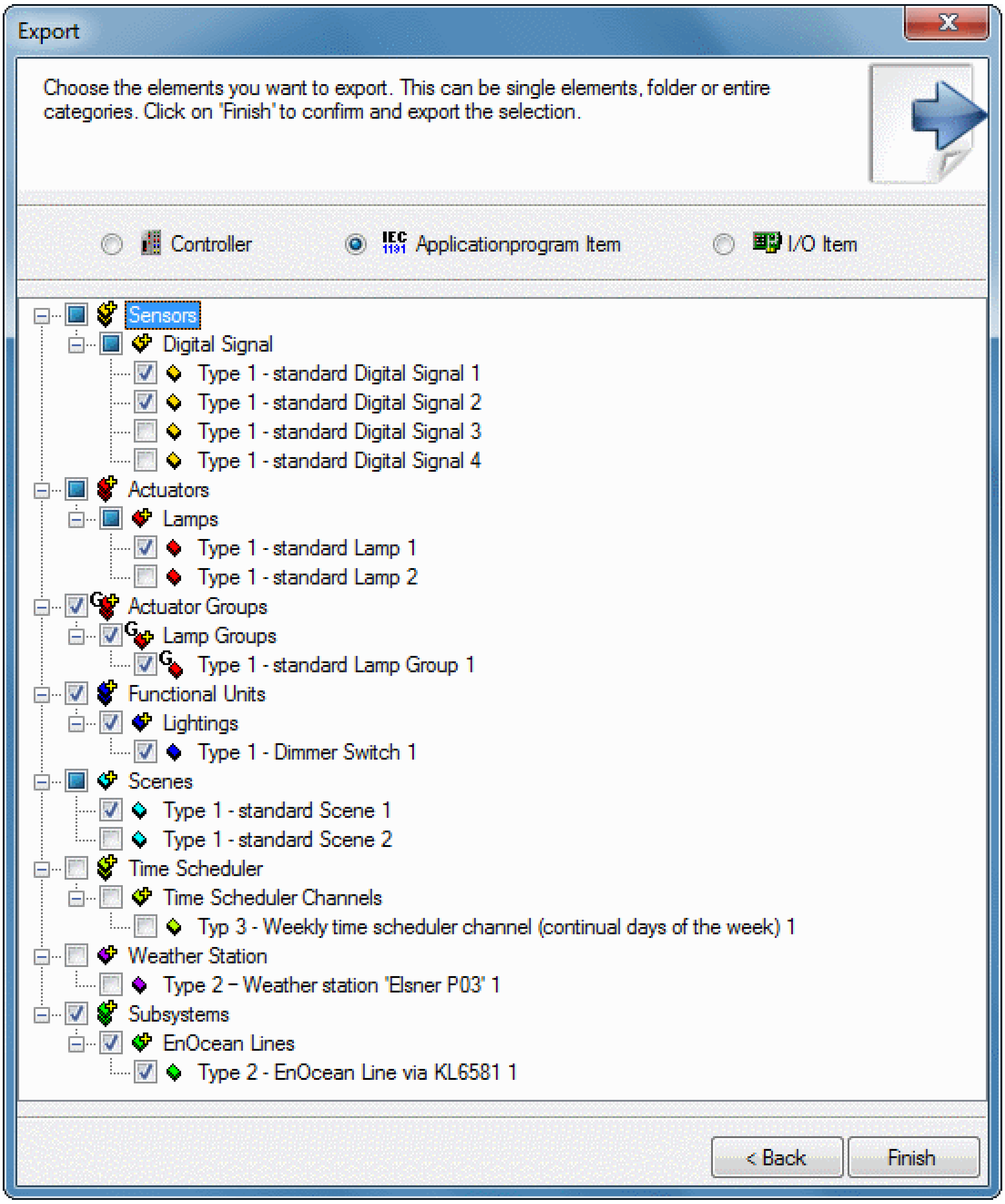
Export

Select the controller from which you want to export elements, such as digital signals, lamps or scenes. Click on 'Next' to proceed to selection of single elements.

○ Controller ● Applicationprogram Item ○ I/O Item

Controllers
☑ CX1010 1

The elements can then be selected in the next step.



Choose the elements you want to export. This can be single elements, folder or entire categories. Click on 'Finish' to confirm and export the selection.

○ Controller ● Applicationprogram Item ○ I/O Item

- ▣ Sensors
  - ▣ Digital Signal
    - ☑ Type 1 - standard Digital Signal 1
    - ☑ Type 1 - standard Digital Signal 2
    - □ Type 1 - standard Digital Signal 3
    - □ Type 1 - standard Digital Signal 4
- ▣ Actuators
  - ▣ Lamps
    - ☑ Type 1 - standard Lamp 1
    - □ Type 1 - standard Lamp 2
- ☑ Actuator Groups
  - ☑ Lamp Groups
    - ☑ Type 1 - standard Lamp Group 1
- ☑ Functional Units
  - ☑ Lightings
    - ☑ Type 1 - Dimmer Switch 1
- ▣ Scenes
  - ☑ Type 1 - standard Scene 1
  - □ Type 1 - standard Scene 2
- □ Time Scheduler
  - □ Time Scheduler Channels
    - □ Typ 3 - Weekly time scheduler channel (continual days of the week) 1
- □ Weather Station
  - □ Type 2 – Weather station 'Elsner P03' 1
- ☑ Subsystems
  - ☑ EnOcean Lines
    - ☑ Type 2 - EnOcean Line via KL6581 1

The export file is saved in the specified directory by pressing the *Finish* button.

7.1.7 Tools

7.1.7.1 Reports

7.1.7.1.1 Terminal Statistics

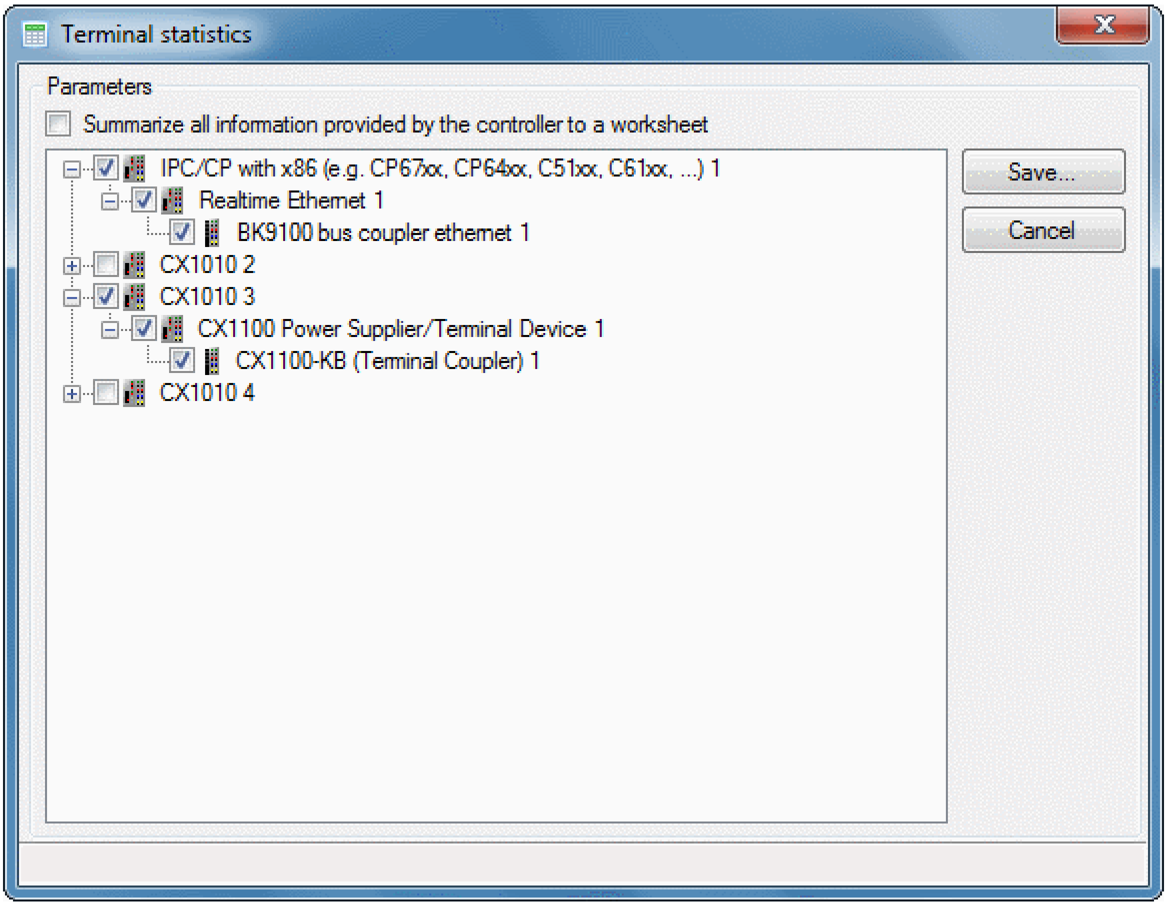
The *Terminal Statistics* tool provides information about the terminals used in the project in the form of an OOXML (Office Open XML) table.

The table contains the following information:

- Terminal type
- Number of terminals

- Number of terminal channels
- Number of used and unused channels
- Unused channels in percent

Under the menu item *Tools → Reports...* the tool *Terminal Statistics* can be called up in the following dialog.



Terminal statistics

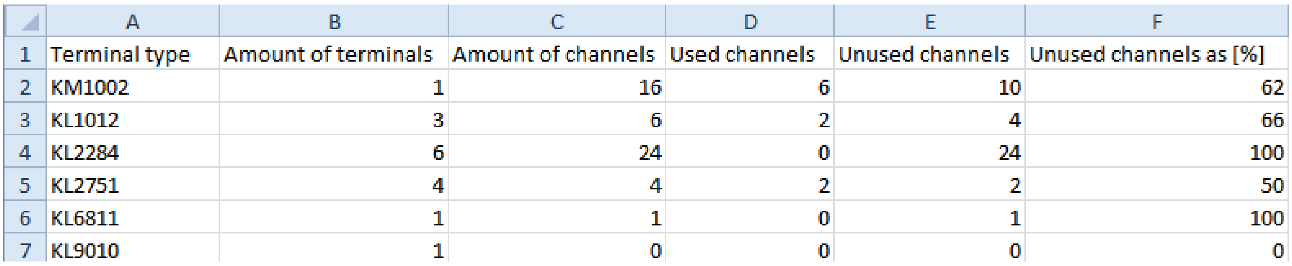
Parameters

□ Summarize all information provided by the controller to a worksheet

- ☑ IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1
  - ☑ Realtime Ethernet 1
    - ☑ BK9100 bus coupler ethernet 1
- □ CX1010 2
- ☑ CX1010 3
  - ☑ CX1100 Power Supplier/Terminal Device 1
    - ☑ CX1100-KB (Terminal Coupler) 1
- □ CX1010 4

The main form initially lists all controllers. Select a controller to include its terminals in the statistics. Each selected controller is assigned its own worksheet within the table. Press the *Save...* button to enter a name and location for the table.

In addition, an overview can be created. To this end select the control box *Consolidate all controller information in a worksheet*. In this case the terminals for the selected controllers are no longer distributed to separate worksheets, but consolidated in a single one.



	A	B	C	D	E	F
1	Terminal type	Amount of terminals	Amount of channels	Used channels	Unused channels	Unused channels as [%]
2	KM1002	1	16	6	10	62
3	KL1012	3	6	2	4	66
4	KL2284	6	24	0	24	100
5	KL2751	4	4	2	2	50
6	KL6811	1	1	0	1	100
7	KL9010	1	0	0	0	0

To view the generated *.xlsx file a spreadsheet with OOXML (Office Open XML) support is required, such as Microsoft Excel 2010, OpenOffice.org 3 or LibreOffice 3.

7.1.7.2 Website Builder

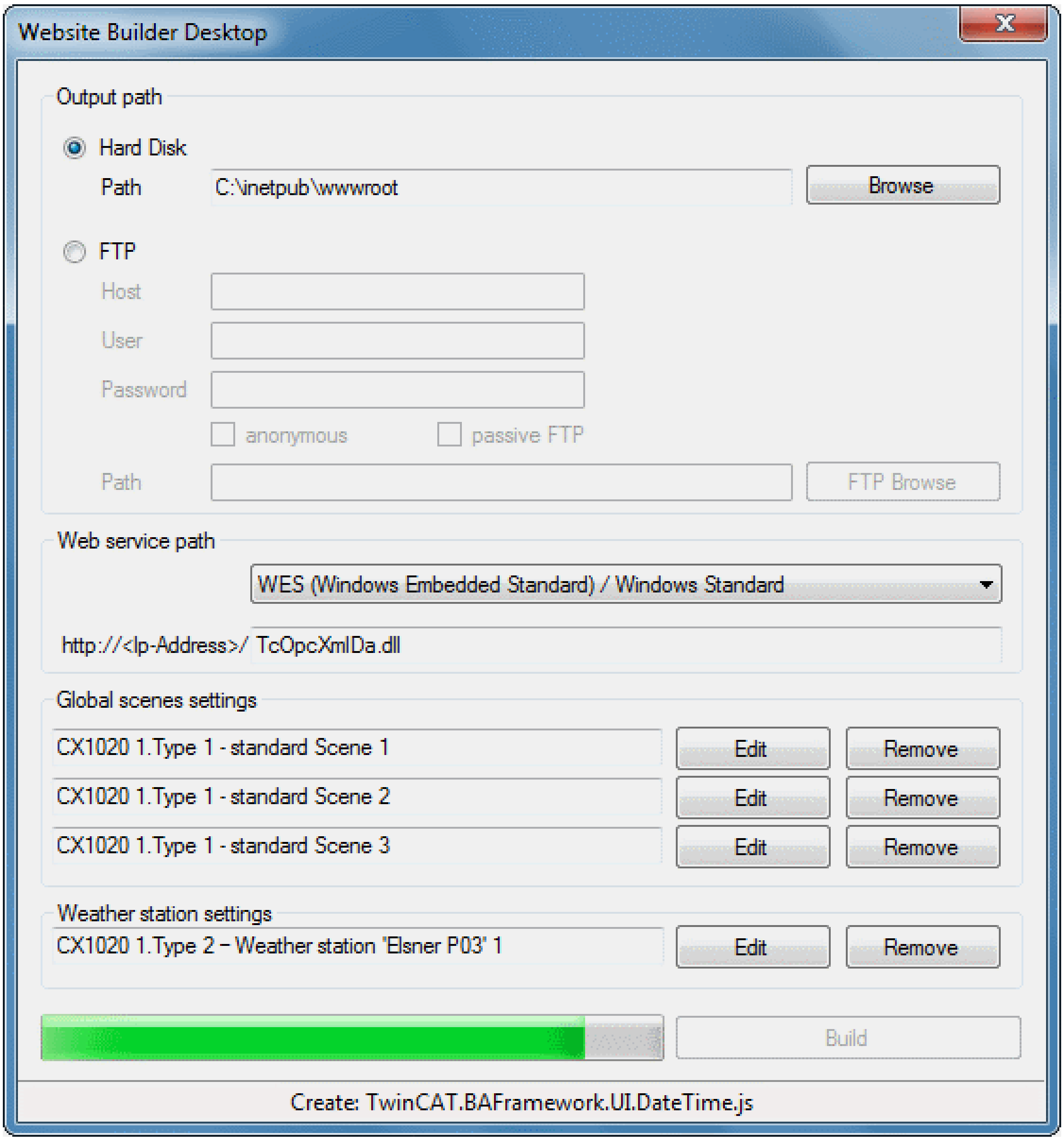
7.1.7.2.1 Website Builder Desktop

This *Website Builder Desktop* generates user interfaces for the performance characteristics of desktop PCs. It is designed for the monitor size of PCs with mouse and keyboard.

The following browser versions have been tested:

- Microsoft Internet Explorer 6 (Windows CE)
- Microsoft Internet Explorer 9
- Mozilla Firefox 5
- Google Chrome 11
- Apple Mobile Safari 3 (IPhone, IPod touch, IPad)
- Apple Safari 5
- Opera 11

Under the menu item *Tools → Website Builder...* the *Website Builder Desktop* can be called up in the following dialog.



Website Builder Desktop

Output path

● Hard Disk

Path C:\inetpub\wwwroot

○ FTP

Host

User

Password

□ anonymous □ passive FTP

Path

Web service path

WES (Windows Embedded Standard) / Windows Standard

http://<Ip-Address>/ TcOpcXmlDa.dll

Global scenes settings

CX1020 1.Type 1 - standard Scene 1

CX1020 1.Type 1 - standard Scene 2

CX1020 1.Type 1 - standard Scene 3

Weather station settings

CX1020 1.Type 2 – Weather station 'Elsner P03' 1

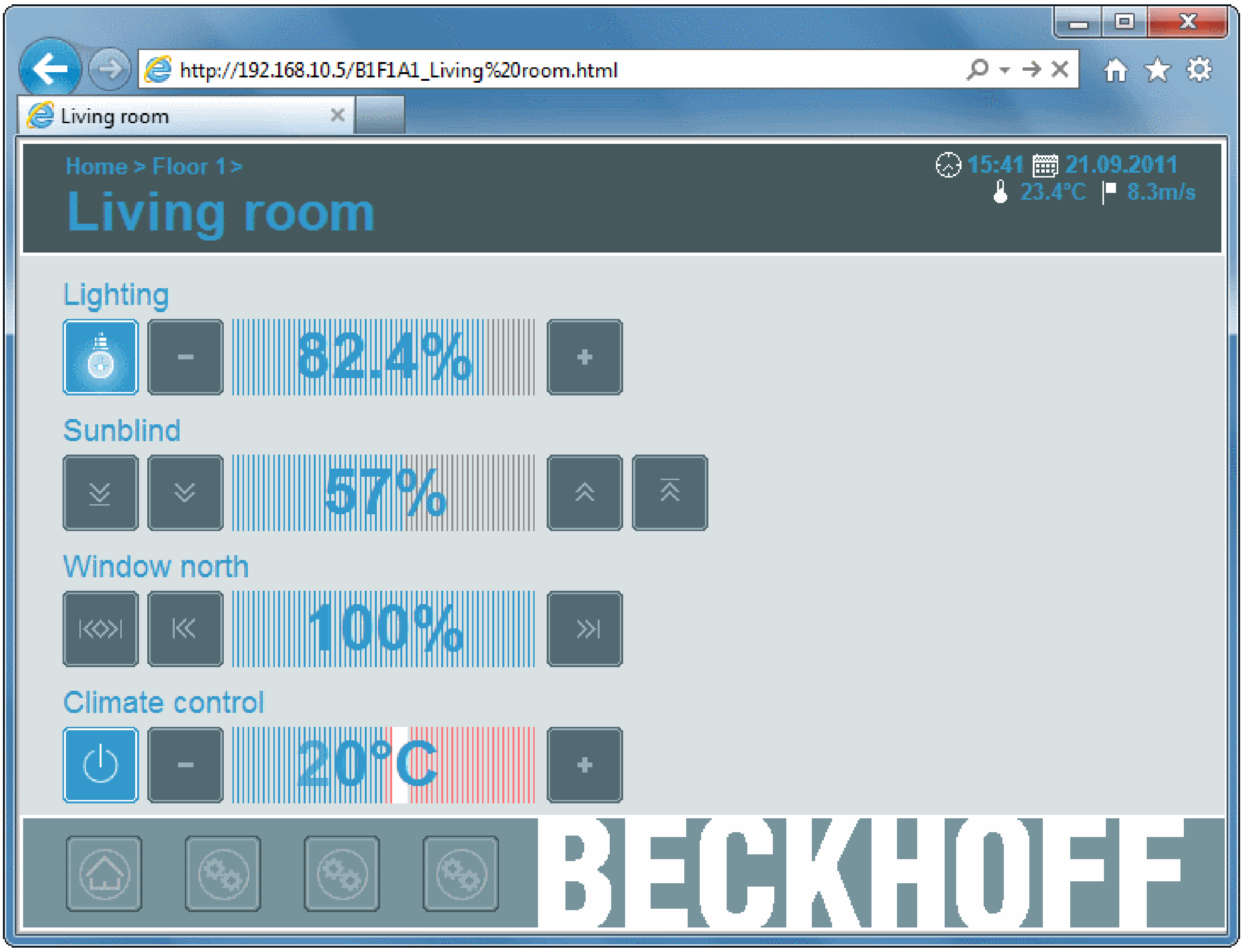
Create: TwinCAT.BAFramework.UI.DateTime.js

The dialog offers the option to specify the storage location for the web pages. If *Hard Disk* is selected, you can select a location on the hard disk via the *Browse* button. If *FTP* is selected the pages are stored directly in an FTP folder.

Dependent on the used image the web service is selected. If it is not in the standard directory, the path can be adapted under the entry "user-defined".

In addition you can assign scenes to the three quick access keys by pressing the *Edit* button. To display weather data you can specify a weather station via the *Edit* button.

Then start the process by pressing the *Build* button.



The main screen appears when the page is called up. From there you can navigate to all subsections, from where all assigned elements can be controlled.

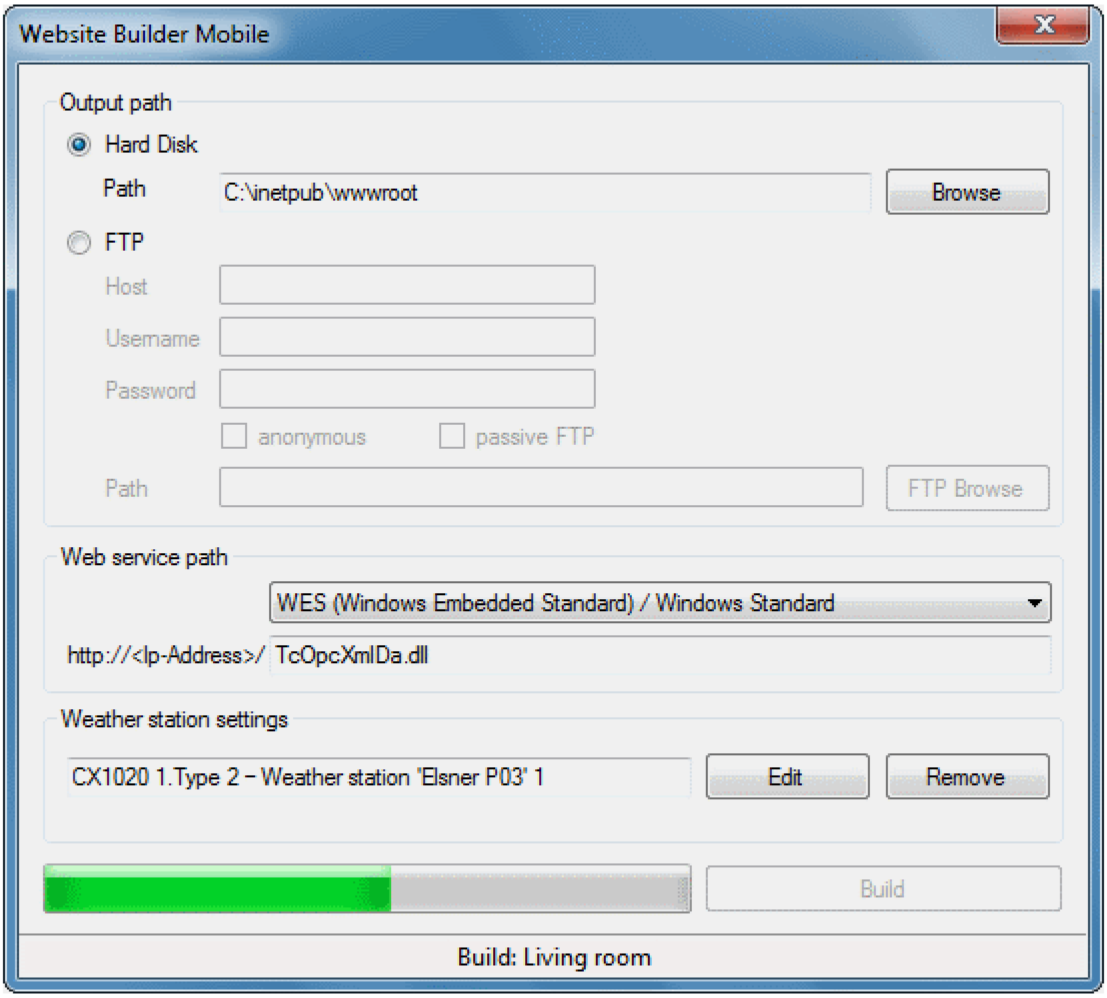
7.1.7.2.2 Website Builder Mobile

The *Website Builder Mobile* generates user interfaces for the performance characteristics of Smartphones. It is designed for the size of Smartphone displays with touch control. The browser must support *jQuery Mobile*.

The following browser versions have been tested:

- Microsoft Internet Explorer 9
- Mozilla Firefox 5
- Google Chrome 11
- Apple Mobile Safari 3 (IPhone, IPod touch, IPad)
- Apple Safari 5
- Opera 11

Under the menu item *Tools* → *Website Builder...* you can call up the *Website Builder Mobile* in the following dialog.



Website Builder Mobile

Output path

● Hard Disk

Path: C:\inetpub\wwwroot

○ FTP

Host:

Username:

Password:

□ anonymous □ passive FTP

Path:

Web service path: WES (Windows Embedded Standard) / Windows Standard

http://<Ip-Address>/ TcOpcXmlDa.dll

Weather station settings: CX1020 1.Type 2 – Weather station 'Elsner P03' 1

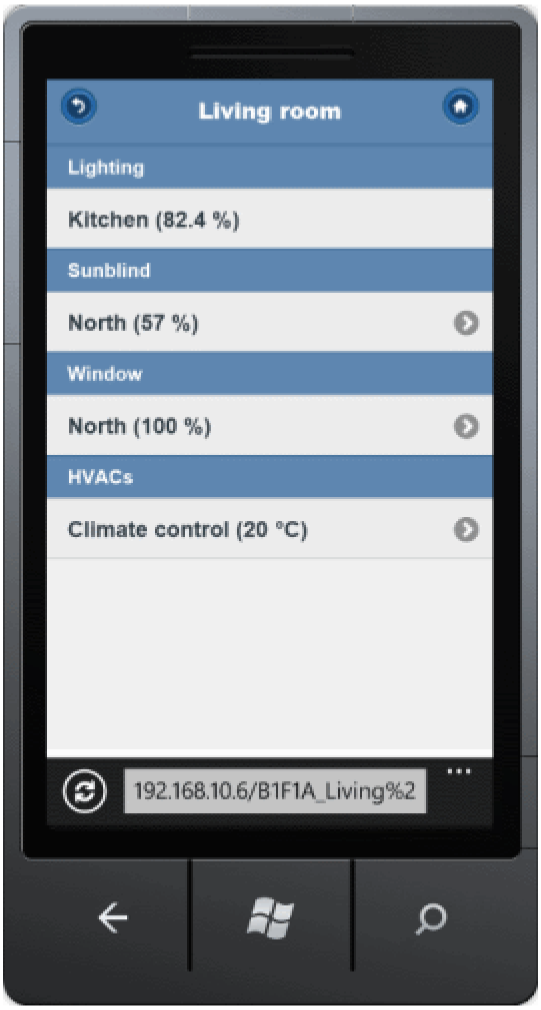
Build: Living room

The dialog offers the option to specify the storage location for the web pages. If *Hard Disk* is selected, you can select a location on the hard disk via the *Browse* button. If *FTP* is selected the page is stored directly in an FTP folder.

Dependent on the used image the web service is selected. If it is not in the standard directory, the path can be adapted under the entry "user-defined".

To display weather data you can specify a weather station via the *Edit* button.

Then start the process by pressing the *Build* button.



Living room

Lighting

Kitchen (82.4 %)

Sunblind

North (57 %)

Window

North (100 %)

HVACs

Climate control (20 °C)

192.168.10.6/B1F1A_Living%2

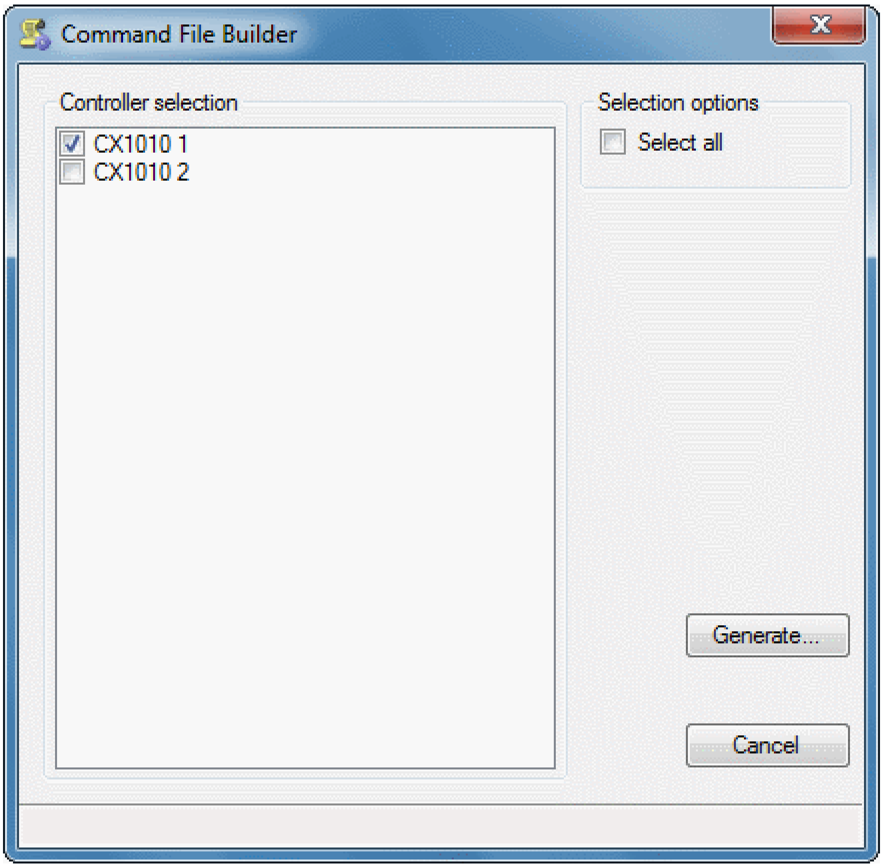
The main screen appears when the page is called up. From there you can navigate to all subsections Arrived at the selected subsection, all assigned elements can be controlled.

7.1.7.3 Extensions

7.1.7.3.1 Command File Builder

In some cases PLC programs for several controllers may have to be updated. The *Command File Builder* supports this task. It generates an executable *.bat file which processes a *.ini instruction list. Apart from starting the *.bat file no further user intervention is required. The controllers are automatically selected in turn, the PLC program is reloaded, a boot project is generated, the runtime is set to *Run* mode, followed by logout.

Under the menu item *Tools → Extensions...* the *Command File Builder* can be called up in the selection dialog.



Command File Builder

Controller selection

☑ CX1010 1
□ CX1010 2

Selection options

□ Select all

In the main dialog all controllers are listed that are valid for this process. That is, a controller with an AmsNetID and an application program with a linked *.tpy file. To add a controller to the update process it has to be selected. For a larger number of controllers the option *Select All* is available. This option selects or deselects all controllers.

When the *Generate* button is pressed the two files are saved in the selected folder.

```
query off ok
out open CommandFileBuilder.log
out clear

file open "TcBAFrameworkV2.20_Sample.pro"
delay 3000
chooseruntime 192.168.10.5.1.1:801
delay 2000
online login
delay 4000
createbootproject
delay 6000
online run
delay 1000
online logout
delay 1000
file close

out close
file quit
```

The content of the *.ini file which is processed once the *.bat file has been started is listed above.

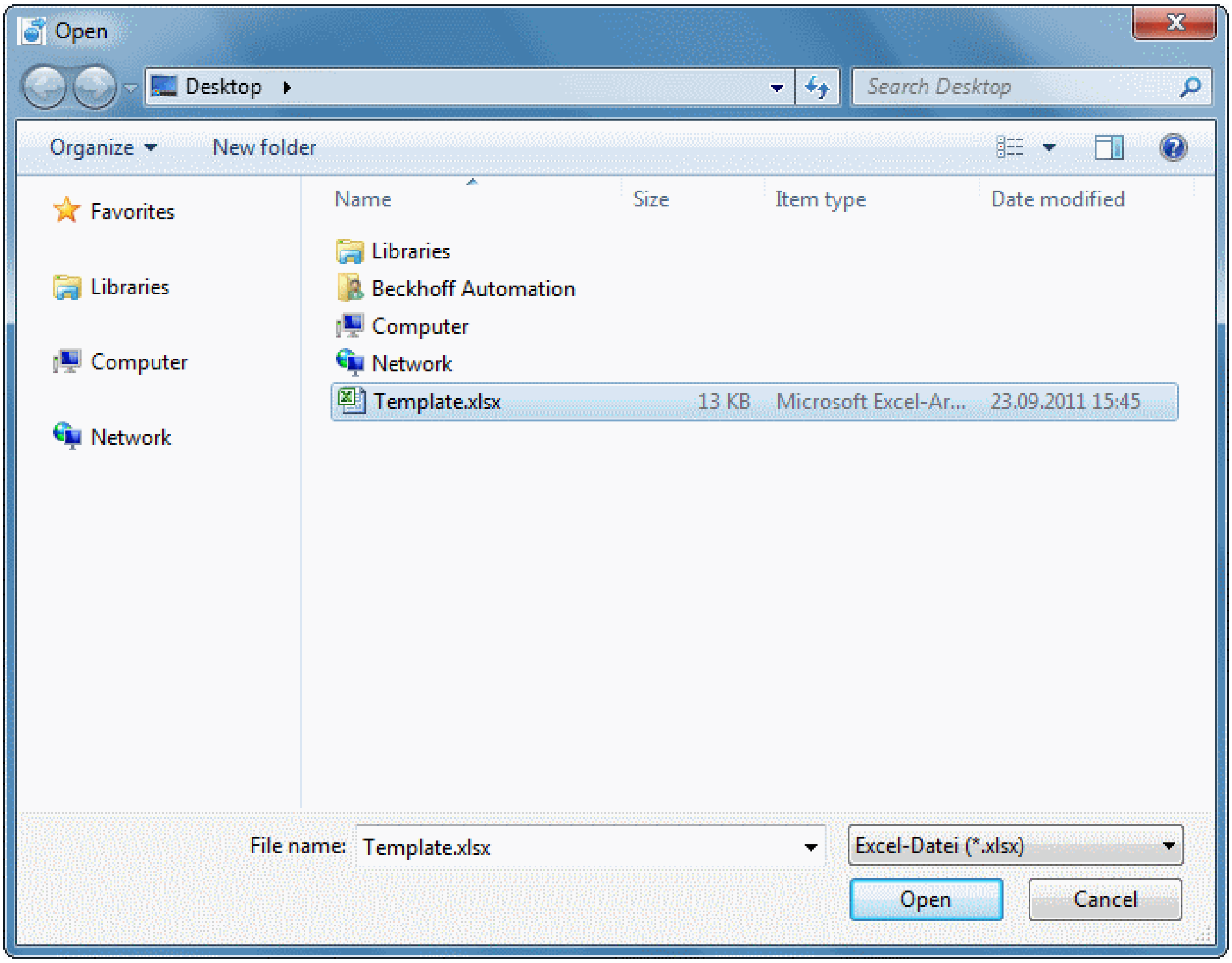
First, the log file *CommandFileBuilder.log* is created to which all log entries from TwinCAT PLC Control are written. Then, the project file is opened, followed by login to the target system. The project is then loaded in the controller, a boot project is generated, and the runtime system is set to *Run* mode. After logout the project is closed again, output to the log file is stopped, and the command window for the *.bat file is terminated.

Further information on command line instructions in TwinCAT PLC Control is available in the Beckhoff Information System under the following path:
Beckhoff Information System → TwinCAT → TwinCAT PLC → TwinCAT PLC Control → Annex → Command line instructions

7.1.7.3.2 Project File Builder

The *Project File Builder* generates a Building Automation Manager project from an OOXML (Office Open XML) table.

Under the menu item *Tools → Extensions...* the *Project File Builder* can be called up in the subsequent selection dialog.



Name	Size	Item type	Date modified
Template.xlsx	13 KB	Microsoft Excel-Ar...	23.09.2011 15:45

File name: Template.xlsx

Excel-Datei (*.xlsx)

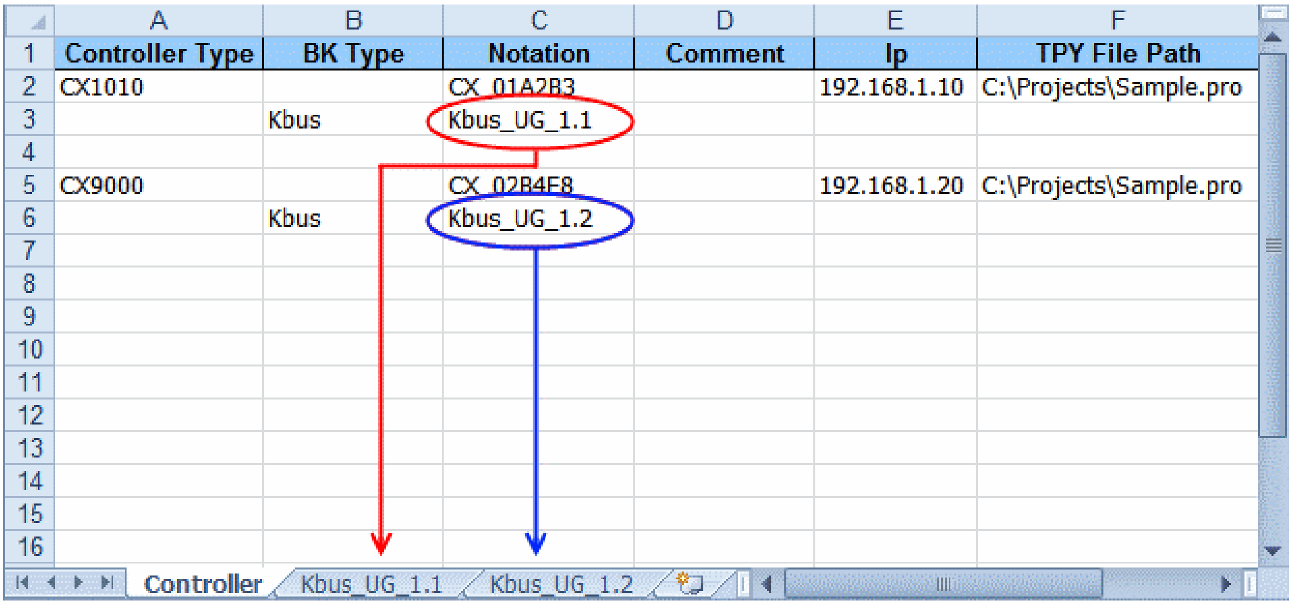
The familiar Windows dialog for opening files appears, in which the required *.xlsx file can be selected. Press the *Open* button to start processing the table. Once the process is complete, all windows can be closed until the main window of the BA Manager is shown again.

Importing an OOXML table results in creation of a new project! An existing project cannot be extended.

Creating the table

Successful reading of the OOXML table requires a valid structure. This includes defined column and worksheet labels and table entries.

Controller worksheet



	A	B	C	D	E	F
1	Controller Type	BK Type	Notation	Comment	Ip	TPY File Path
2	CX1010		CX_01A2B3		192.168.1.10	C:\Projects\Sample.pro
3		Kbus	Kbus_UG_1.1			
4						
5	CX9000		CX_02B4E8		192.168.1.20	C:\Projects\Sample.pro
6		Kbus	Kbus_UG_1.2			
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

The first worksheet defines the controllers used. It must be called *Controller*. As indicated in the diagram, each controller is defined by at least 2 lines.

The first line contains:

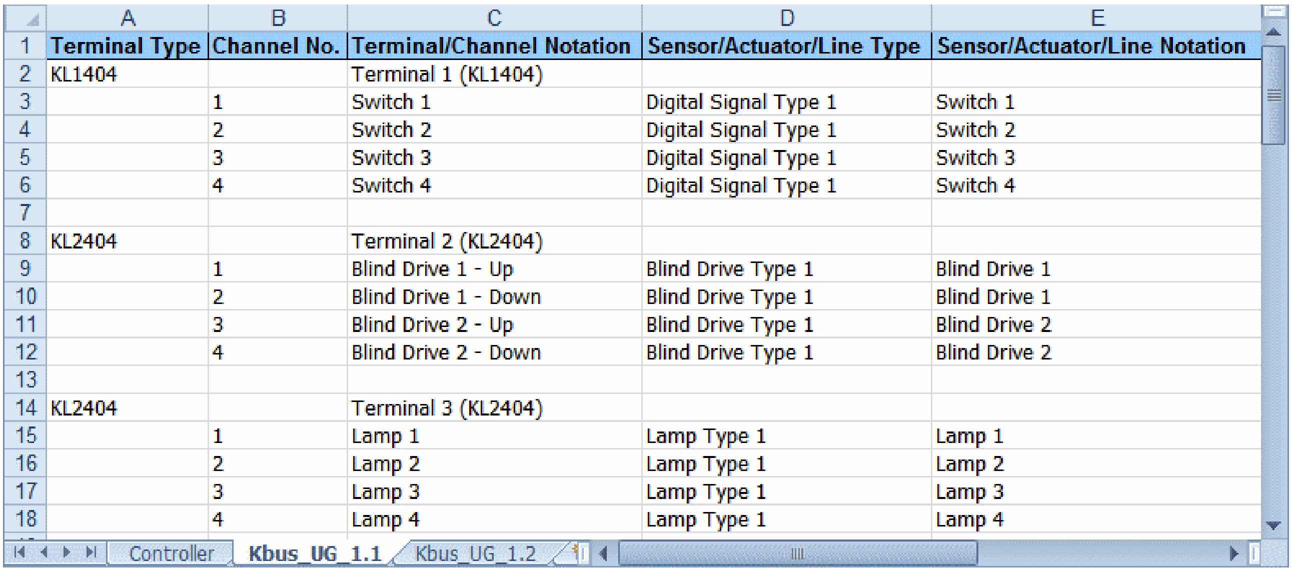
- **Controller Type:** Indicates the controller type.

  Valid controller types:
  - CX1000
  - CX1010
  - CX1020
  - CX1030
  - CX5000
  - CX9000
  - CX9001
  - CX9010
  - CX9020
  - IpcCpX86
  - IpcCpArm
- **Notation:** Name of the controller.
- **Comment:** Comment (optional).
- **Ip:** IP address (valid IPv4) of the controller (not for *KBUS*).
- **TPY File Path:** Path to the PLC project for obtaining the *.tpy file.

The second line contains:

- **BK Type:** Indicates the Bus Coupler type. The permitted value is *KBUS* (only CX devices). It goes without saying that the controller type must support the specified Bus Coupler type. Please refer to the relevant documentation.
- **Notation:** Name of the Bus Coupler and the worksheet. **These names must be identical!**

Bus Coupler worksheet



	A	B	C	D	E
1	Terminal Type	Channel No.	Terminal/Channel Notation	Sensor/Actuator/Line Type	Sensor/Actuator/Line Notation
2	KL1404		Terminal 1 (KL1404)		
3		1	Switch 1	Digital Signal Type 1	Switch 1
4		2	Switch 2	Digital Signal Type 1	Switch 2
5		3	Switch 3	Digital Signal Type 1	Switch 3
6		4	Switch 4	Digital Signal Type 1	Switch 4
7					
8	KL2404		Terminal 2 (KL2404)		
9		1	Blind Drive 1 - Up	Blind Drive Type 1	Blind Drive 1
10		2	Blind Drive 1 - Down	Blind Drive Type 1	Blind Drive 1
11		3	Blind Drive 2 - Up	Blind Drive Type 1	Blind Drive 2
12		4	Blind Drive 2 - Down	Blind Drive Type 1	Blind Drive 2
13					
14	KL2404		Terminal 3 (KL2404)		
15		1	Lamp 1	Lamp Type 1	Lamp 1
16		2	Lamp 2	Lamp Type 1	Lamp 2
17		3	Lamp 3	Lamp Type 1	Lamp 3
18		4	Lamp 4	Lamp Type 1	Lamp 4

These worksheets contain the terminals, sensors, actuators and subsystems connected to the Bus Coupler. Each terminal is defined by a minimum of 2 lines. For channel-less terminals (e.g. end terminal KL9010) only the top line is required.

The first line contains:

- **Terminal Type:** terminal type.
- **Terminal/Channel Notation:** terminal name.

The second line contains:

- **Channel No.:**terminal channel to be linked.
- **Terminal/Channel Notation:** channel name.
- **Sensor/Actuator/Line Type:** type of sensor/actuator/subsystem to be created, with which the channel is subsequently linked (optional).

  Permitted sensor/actuator/subsystem types:

  - DIGITAL SIGNAL TYPE 1 - 3
  - ANALOG SIGNAL TYPE 1 - 3
  - LAMP TYPE 1 - 3
  - BLIND DRIVE TYPE 1 - 2
  - WINDOW DRIVE TYPE 1
  - VALVE ACTUATOR 2 POINT
  - VALVE ACTUATOR 3 POINT
  - VALVE ACTUATOR CONTINUOUS
  - ENOCEAN LINE TYPE 1 - 2
  - DALI LINE TYPE 1
- **Sensor/Actuator/Line Notation:** name of sensor/actuator/subsystem.
- **Function:** actuator parameters. Permitted values for

  LAMP TYPE 1

  - DIGITAL
  - ANALOG

  BLIND DRIVE TYPE 1

  - UP
  - DOWN

  WINDOW DRIVE TYPE 1 and VALVE ACTUATOR 3 POINT

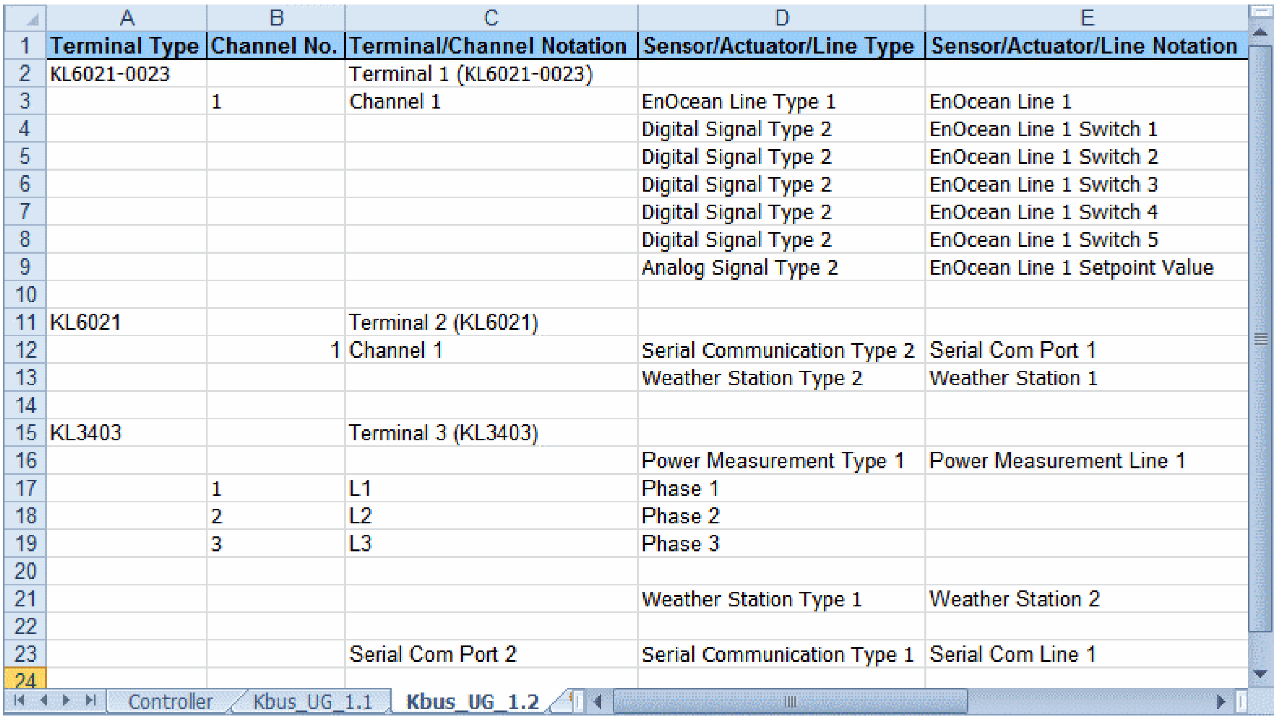
- OPEN
- CLOSE

- **Parameter 1 - 4:** Additional parameters

Ensure that terminal channel, sensor/actuator/subsystem and the function, if applicable, are compatible. Linking a channel of an analog output terminal with a digital signal would not be permitted, for example.

A function must be defined if an element from type *LAMP TYPE 1*, *BLIND DRIVE TYPE 1*, *WINDOW DRIVE TYPE 1* or *VALVE ACTUATOR 3 POINT is to be defined.*

Except for the element names the document is not case sensitive, and hyphenation is not taken into account. Entries such as *EnOceanLineType1* or *cx 9 0 0 0* are therefore also permitted, for example. You only have to list the terminal channels you actually want to use. Ensure that the listed channels exist.



	A	B	C	D	E
1	Terminal Type	Channel No.	Terminal/Channel Notation	Sensor/Actuator/Line Type	Sensor/Actuator/Line Notation
2	KL6021-0023		Terminal 1 (KL6021-0023)		
3		1	Channel 1	EnOcean Line Type 1	EnOcean Line 1
4				Digital Signal Type 2	EnOcean Line 1 Switch 1
5				Digital Signal Type 2	EnOcean Line 1 Switch 2
6				Digital Signal Type 2	EnOcean Line 1 Switch 3
7				Digital Signal Type 2	EnOcean Line 1 Switch 4
8				Digital Signal Type 2	EnOcean Line 1 Switch 5
9				Analog Signal Type 2	EnOcean Line 1 Setpoint Value
10					
11	KL6021		Terminal 2 (KL6021)		
12		1	Channel 1	Serial Communication Type 2	Serial Com Port 1
13				Weather Station Type 2	Weather Station 1
14					
15	KL3403		Terminal 3 (KL3403)		
16				Power Measurement Type 1	Power Measurement Line 1
17		1	L1	Phase 1	
18		2	L2	Phase 2	
19		3	L3	Phase 3	
20					
21				Weather Station Type 1	Weather Station 2
22					
23			Serial Com Port 2	Serial Communication Type 1	Serial Com Line 1
24					

For subsystems the first channel of the respective terminal should be linked with the corresponding subsystem line.

EnOcean

EnOcean modules can be defined at the same time. To this end add the required content in the following columns for each EnOcean module:

- **Sensor/Actuator/Line Type:** *DIGITAL SIGNAL TYPE 1* or *ANALOG SIGNAL TYPE 1*.
- **Sensor/Actuator/Line Notation:** module name.
- **Parameter 1:** terminal ID.
- **Parameter 2:** transmitter ID of the module.
- **Parameter 3:** module type.

  Permitted values:
  - PTM 100 (only for *DIGITAL SIGNAL TYPE 1*)
  - PTM 200 (only for *DIGITAL SIGNAL TYPE 1*)
  - PTM 250 (only for *DIGITAL SIGNAL TYPE 1*)
  - STM 100
  - STM 250

- **Parameter 4:** depends on the selected module type and describes the data byte and bit (for PTM modules therefore the rubber contact).

  Permitted values for *PTM 100*:
  - STATE O / CHANNEL A
  - STATE O / CHANNEL B
  - STATE O / CHANNEL C
  - STATE O / CHANNEL D
  - STATE I / CHANNEL A
  - STATE I / CHANNEL B
  - STATE I / CHANNEL C
  - STATE I / CHANNEL D

  Permitted values for *PTM 200* and *PTM 250*:
  - STATE O / CHANNEL A
  - STATE O / CHANNEL B
  - STATE I / CHANNEL A
  - STATE I / CHANNEL B

  For STM 100 and *STM 250* the data byte (low word) must be set (value range: 0-3). Optionally, it can be followed by the data bit (high word), separated by "." (value range: 0-7).
  The meaning of these bits is described in the EnOcean documentation.

DALI

DALI lamps can also be defined at the same time. To this end add the required content in the following columns for each DALI lamp:

- **Sensor/Actuator/Line Type:** *LAMP TYPE 2*.
- **Sensor/Actuator/Line Notation:** lamp name.
- **Parameter 1:** DALI address of the lamp.

To speed up the process a template is available for download via the following link. This table contains the basic structure and a sample entry for each possible type.

https://infosys.beckhoff.com/content/1033/tcbaframework/Resources/11368455179/.zip

7.1.7.4 Appendix

7.1.7.4.1 Creating custom tools

The TwinCAT Building Automation Manager offers the possibility to create custom tools (add-ins). Add-ins have the possibility to access the internal data management of the TwinCAT Building Automation Manager via an object model. This may be a read or a write access. Hence, add-ins can be created with which documentation on the system can be automatically generated. It is also possible to create add-ins for reading files (e.g. Excel files) from which the project file is generated. Of particular importance is the possibility to generate entire operator interfaces.

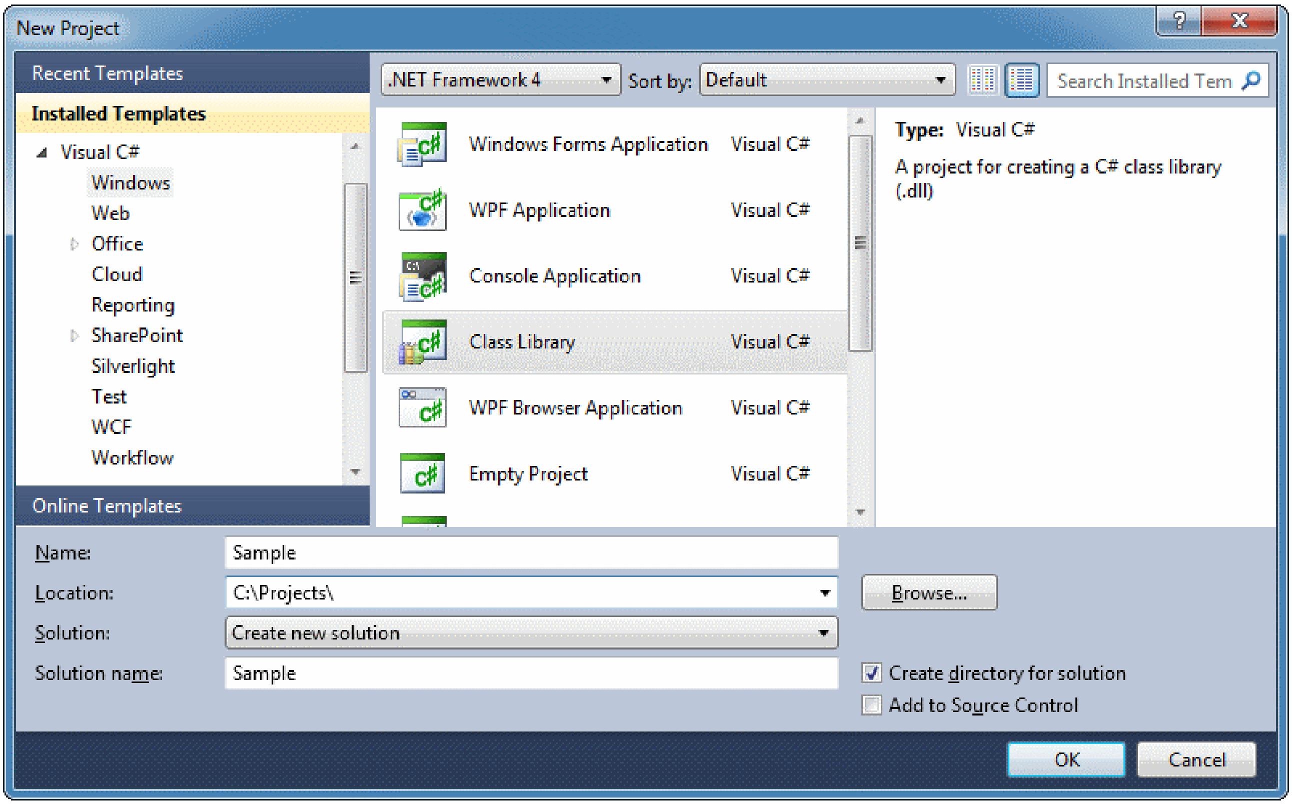
The following executions show all steps necessary to create a basis for the actual add-in development. The complete Microsoft Visual Studio 2010 project is available in the download area [▶ 54] at the foot of the page.

Requirements

- Microsoft Visual Studio 2010 (or higher)
- Microsoft .NET Framework 4.0 (or higher)
- Programming skills in C#

Creating a project

Start Microsoft Visual Studio 2010 and create a new project. The project must be of the type *Class Library*.



Name: Sample

Location: C:\Projects\

Solution: Create new solution

Solution name: Sample

☑ Create directory for solution

□ Add to Source Control

Remove the *Class1.cs* file created as standard in the *Solution Explorer* and add a new Windows form to the project. Mark the newly added *Form1.cs* file and open the code view. The source text which appears should now look as follows:

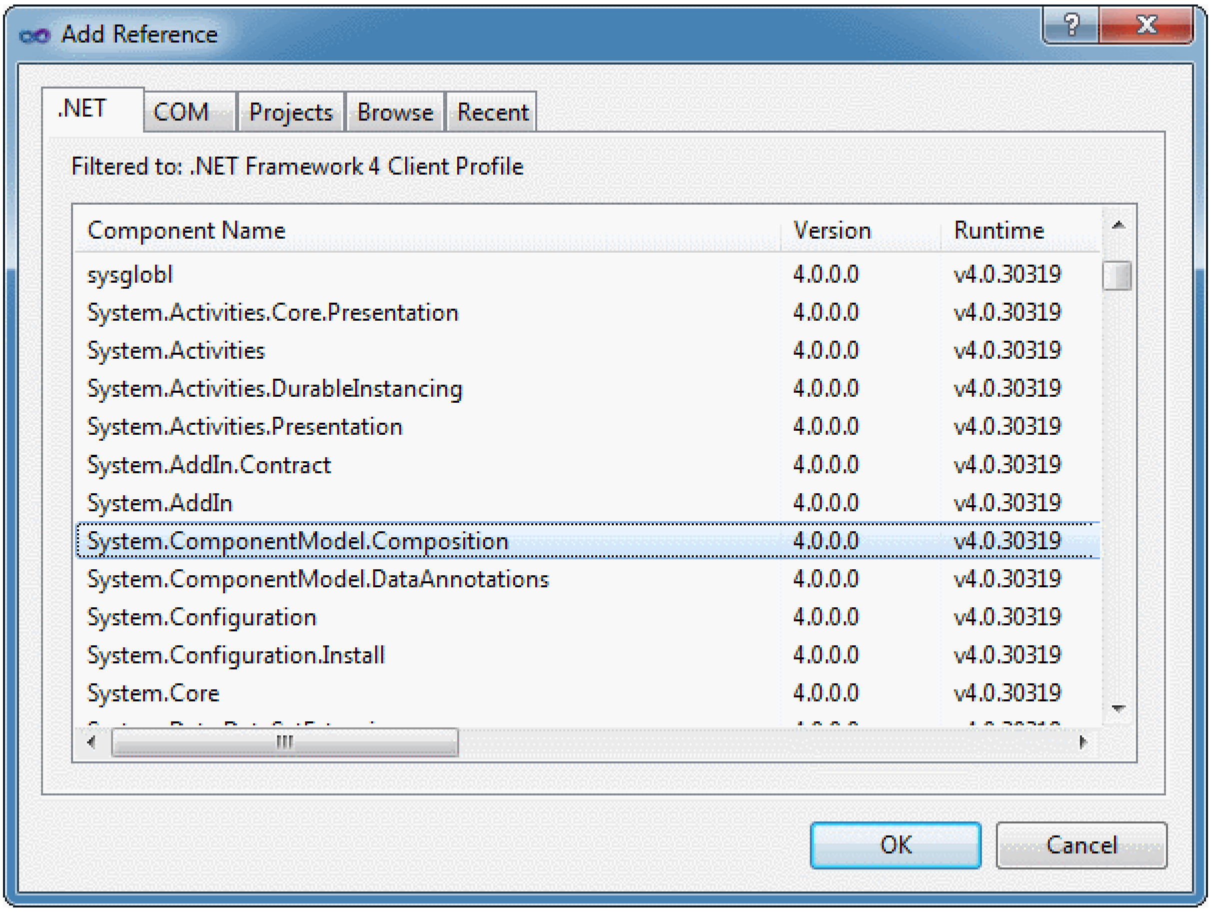
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Sample
{
    public partial class Form1 : Form
    {
    public Form1()
    {
        InitializeComponent();
    }
    }
}
```

Adding references

The next step covers the adding of three references.

- **System.ComponentModel.Composition:** provides the necessary attributes *ExportMetadata* and *Export*.
- **TcBAManager.AddIn:** offers access to classes that are necessary for working with the TwinCAT Building Automation Manager.
- **TcBAManager.AddIn.Contract:** interfaces and enumeration types which the add-in makes available to the TwinCAT Building Automation Manager.

Right-click in the *Solution Explorer* on *References* and select *Add Reference…*. In the window which then opens, select *Assemblies* on the left-hand side and then look for the reference *System.ComponentModel.Composition*. A double-click adds it to the project.



Add Reference

Filtered to: .NET Framework 4 Client Profile

Component Name	Version	Runtime
sysglobl	4.0.0.0	v4.0.30319
System.Activities.Core.Presentation	4.0.0.0	v4.0.30319
System.Activities	4.0.0.0	v4.0.30319
System.Activities.DurableInstancing	4.0.0.0	v4.0.30319
System.Activities.Presentation	4.0.0.0	v4.0.30319
System.AddIn.Contract	4.0.0.0	v4.0.30319
System.AddIn	4.0.0.0	v4.0.30319
System.ComponentModel.Composition	4.0.0.0	v4.0.30319
System.ComponentModel.DataAnnotations	4.0.0.0	v4.0.30319
System.Configuration	4.0.0.0	v4.0.30319
System.Configuration.Install	4.0.0.0	v4.0.30319
System.Core	4.0.0.0	v4.0.30319

The references from the TwinCAT Building Automation Manager follow next. To this end press the *Browse* button and navigate to the TwinCAT Building Automation Manager folder (standard is *C:\TwinCAT\BAFrameworkV2.20\BAManager*). *TcBAManager.AddIn.dll* and *TcBAManager.AddIn.Contract.dll* are located there. Add both to the project. The window can now be closed.

Placing at disposal

So that the add-in is available in the TwinCAT Building Automation Manager and can be correctly displayed, a couple of extensions are necessary in the source text of the *Form1.cs* file.

Begin with the extension of the namespaces. *System.ComponentModel.Composition*, *TwinCAT.BAManager.AddIns.Model* and *TwinCAT.BAManager.AddIns.Contract* are added.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.ComponentModel.Composition;
using TwinCAT.BAManager.AddIns.Model;
using TwinCAT.BAManager.AddIns.Contract;

namespace Sample
{
    public partial class Form1 : Form
    {
    public Form1()
    {
        InitializeComponent();
    }
    }
}
```

Now extend the constructor by the interfaces *IAddInContractV1* and *IDisposable*. The attributes via which the additional information within the TwinCAT Building Automation Manager is made available follow.

These include the display name and the description in the respective language, the version number and the type of add-in. This specifies the place of display of the add-in in the menu item *Tools* within the TwinCAT Building Automation Manager.

The specification next the *Export* attribute enables the TwinCAT Building Automation Manager to identify valid add-ins.

Now the interface *IAddInContractV1* needs to be implemented and the global variable of the class *Project* created and initialised.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.ComponentModel.Composition;
using TwinCAT.BAManager.AddIns.Model;
using TwinCAT.BAManager.AddIns.Contract;

namespace Sample
{
    [ExportMetadata("Name1031", "Beispiel")]
    [ExportMetadata("Name1033", "Sample")]
    [ExportMetadata("Description1031", "Ein Beispielprogramm.")]
    [ExportMetadata("Description1033", "A sample program.")]
    [ExportMetadata("Version", "V1.0.0")]
    [ExportMetadata("Target", AddInTarget.Extension)]
    [Export(typeof(IAddInContractV1))]
    public partial class Form1 : Form, IAddInContractV1, IDisposable
    {

    private Project project;

    public Form1()
    {
        InitializeComponent();
        project = Project.Instance;
    }

    public Form AddInUserControl
    {
        get
        {
        return this;
        }
    }
    }
}
```

The instance of the *Project* class now offers access to all functions and information provided by the TwinCAT Building Automation Manager.

Since this project concerns a class library, you must first define an external program in the project properties with which the add-in is to be started in order to start and debug. To do this, open the *Solution Explorer* and call the *Properties* by right-clicking on the project. In the *Debug* menu item, enter the path to the TwinCAT Building Automation Manager next to *Start external program*.

Select the *Build* menu item and, in the *Output* area, enter the path to the *AddIn* folder of the TwinCAT Building Automation Manager (standard is *C:\TwinCAT\BAFrameworkV2.20\BAManager\AddIns*). The properties can now be closed.

Finally, steps must be taken to prevent the entire TwinCAT Building Automation Manager from being copied to the AddIn folder when compiling. To do this, open the *Solution Explorer* and expand the *References* folder. Right-click to open the properties of the *TcBAManager.AddIn* reference and set *Copy Local* to false. Repeat this procedure for the *TcBAManager.AddIn.Contract* reference.

Information and functions of the interface

The properties are used for querying or setting simple object information, such as name or comment. More complex information, such as links or occupied terminal channels, is available as an XML document and can be queried with the *GetParameters()* method or set with *SetParameters()*. The structure for the setting of parameters thereby corresponds precisely to the delivered structure of *GetParameters()*.

Furthermore, messages can be deposited in the message window of the TwinCAT Building Automation Manager.

The attached project, *AddIns.Test*, in the download area [▸ 54] provides a detailed insight into the functions and structure.

Application snippets

The project from the previous steps serves as the basis for the following snippets.

Snippet 1 - Reading a scene

Using the TwinCAT Building Automation Manager, create a new project and add to it any controller with at least four scenes and lamps. The first scene receives a command that switches on the first lamp. Repeat this procedure for the other scenes.

```
foreach (Controller controller in project.Controllers)
{
    var scenes = from scene in controller.Scenes
         where (scene.Id % 2) == 0
         select scene;
    Debug.WriteLine("Controller {0}: {1} ", controller.Id, controller.Name);
    foreach (Scene scene in scenes)
    {
    XmlDocument doc = scene.GetParameters();
    Debug.WriteLine("  Scene {0}: {1}", scene.Id, scene.Name);
    }
}
```

After calling the add-in, all scenes of the controller whose ID is divisible by two are transferred to a new list and afterwards output in the debug console.

The *GetParameters( )* method delivers the scene contents in the following XML structure:

```
<?xml version="1.0" encoding="utf-16" ?>
<Scene Id="2">
    <ExternalImplementation>false</ExternalImplementation>
    <SumUpScenes>false</SumUpScenes>
    <SceneItem>
    <EntityControllerId>1</EntityControllerId>
    <EntityType>ActuatorLampBase</EntityType>
    <EntityId>2</EntityId>
    <EntityCommandId>3</EntityCommandId>
    </SceneItem>
</Scene>
```

Snippet 2 - Creating a project

```
project.CreateNew("Sample");

Controller controllerCX1010 = project.Controllers.Add(5);
controllerCX1010.Name = "My CX1010";

FieldBusDevice fieldBusDeviceCX1010 = controllerCX1010.FieldBusDevices.Add(1);
FieldBusCoupler fieldBusCouplerCX1010 = fieldBusDeviceCX1010.FieldBusCouplers.Add(4);

Terminal terminalKL2012 = fieldBusCouplerCX1010.Terminals.Add(project.TerminalDescriptions["2012_0"]
);
terminalKL2012.Name = "My KL2012";

Lamp lampType1 = controllerCX1010.Lamps.Add(1);
lampType1.Name = "My Lamp";

XmlDocument doc = new XmlDocument();
doc.LoadXml(@"
    <Lamp>
    <Link>
        <LinkChannelId>0</LinkChannelId>
        <BusDeviceId>1</BusDeviceId>
        <BusCouplerId>1</BusCouplerId>
        <TerminalId>1</TerminalId>
        <ChannelId>1</ChannelId>
        <ControllerId>1</ControllerId>
    </Link>
    </Lamp>
");

lampType1.SetParameters(doc);
```

Here a new project is created with controller, fieldbus device, fieldbus coupler, terminal and lamp. After creation of the parameters for linking the lamp to the first terminal channel, the XML structure is transferred to the lamp with the *SetParameters( )* method.

The *AddIns.Test* project in the download area [▶ 54] provides a detailed insight into the XML structure of the respective elements.

Close the add-in and the add-in selection dialogue. The newly-created project now appears in the navigation tree.

Download area

Download area

Project template for the development of add-ins.

- https://infosys.beckhoff.com/content/1033/tcbaframework/Resources/11368456587/.zip

Project that demonstrates the functions and structure of the interface.

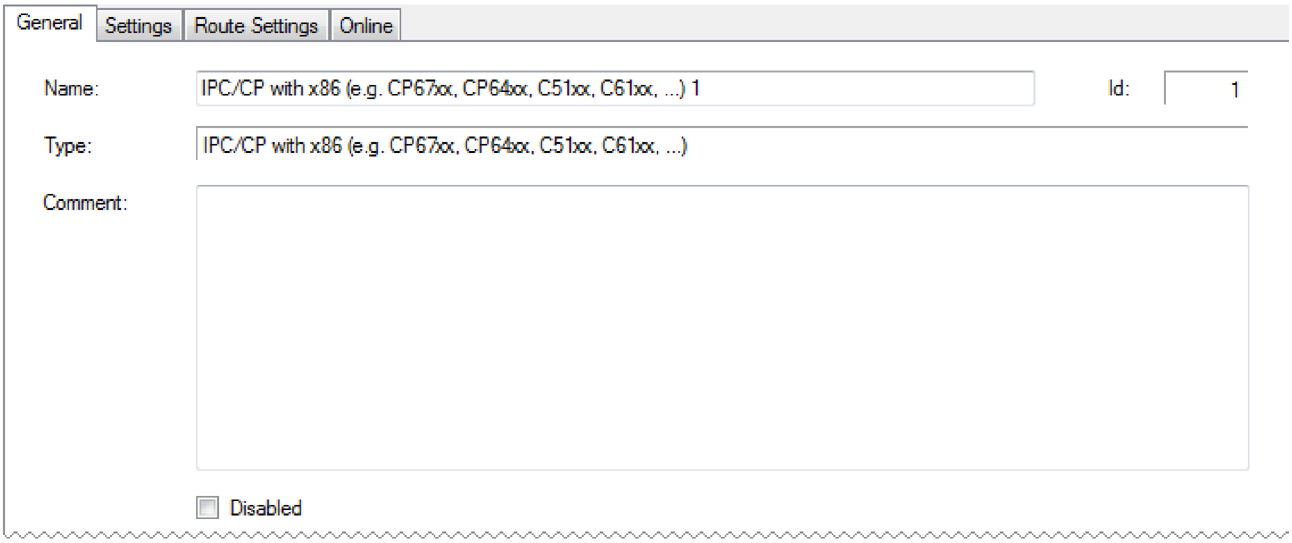
- https://infosys.beckhoff.com/content/1033/tcbaframework/Resources/11368457995/.zip

7.2 Controller Configuration

7.2.1 Controller Settings

By clicking on a controller in the navigation tree the settings of it can be accessed.

General

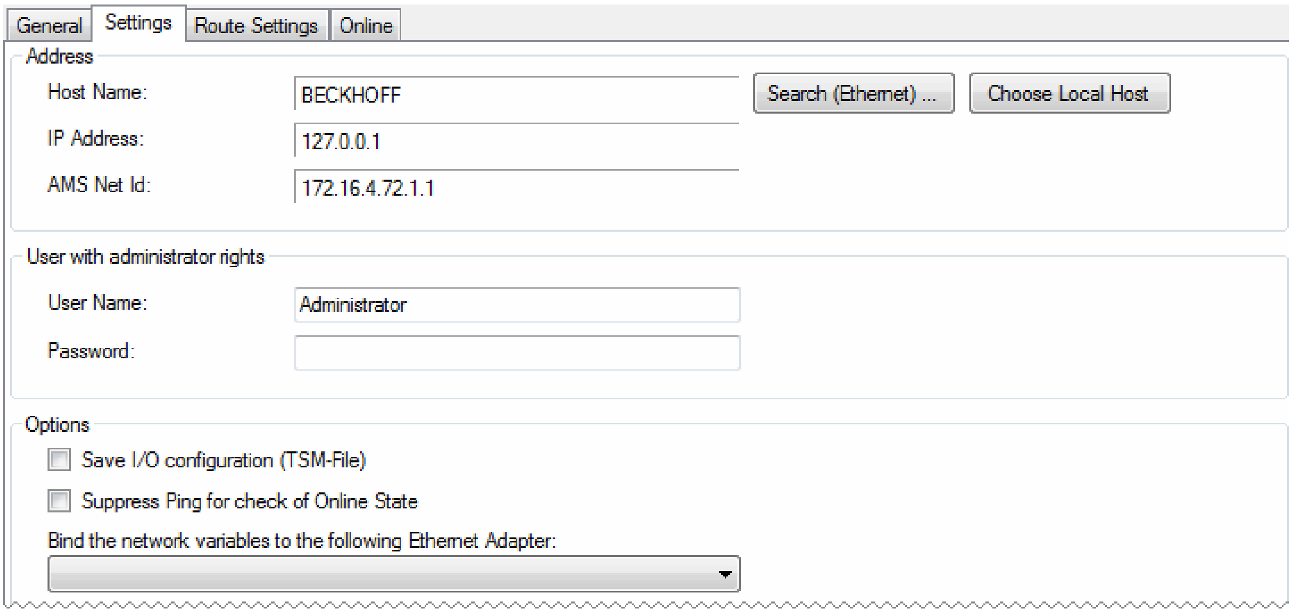


Name: IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1
Id: 1
Type: IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...)
Comment:
□ Disabled

- **Name:** The name of the element can be changed here.
- **Id:** The unique Id of this element.
- **Type:** The type of this element.

- **Comment:** Add comments about this element here.
- **Disabled:** Check this box to disable this element in the current configuration.

Settings



Address

Host Name: BECKHOFF

IP Address: 127.0.0.1

AMS Net Id: 172.16.4.72.1.1

User with administrator rights

User Name: Administrator

Password:

Options

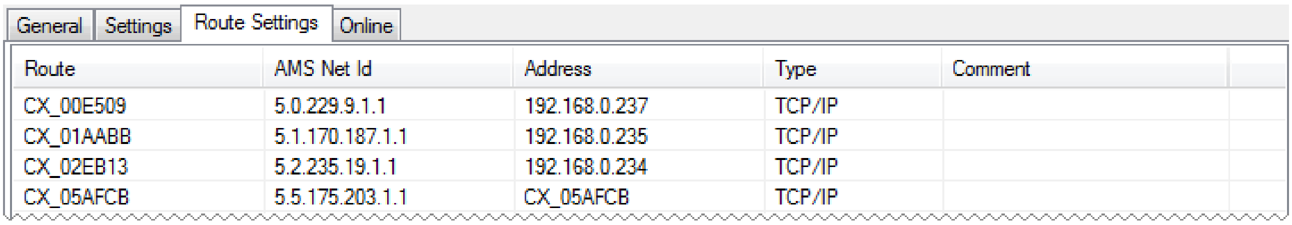
□ Save I/O configuration (TSM-File)

□ Suppress Ping for check of Online State

Bind the network variables to the following Ethernet Adapter:

- **Search (Ethernet)...:** Please enter the hostname or the IP address of the controller into the right text box and click onto the button to test this input.
- **Choose Local Host:** Can be used as an alternative for *Search Ethernet...* and entries the local TwinCAT host as controller.
- **Host Name:** The complete host name of the controller. Is displayed when the input above is successfully.
- **IP Address:** The IP address of the controller. Is displayed when the input above is successfully.
- **AMS Net Id:** The detected AMS Net Id of the controller. Is displayed if the input was successful.
- **User Name:** Name of user with administration rights on the target device.
- **Password:** Password for the user.
- **Save I/O Configuration (TSM-File):** Check this box to create a TwinCAT System Manager compatible *.tsm file if the project is activated. The *.tsm file is stored with the name of the target at the place where the BAM project exists.
- **Suppress Ping for check of Online State:** If the controller has a firewall that is set to prevent ping transfer, error messages may be generated despite the fact that the communication with TwinCAT operates normally. In this case this setting can be used to suppress the ping.
- **Bind the network variables to the following Ethernet Adapter:** Select an Ethernet adapter to which the network variables are to be linked.

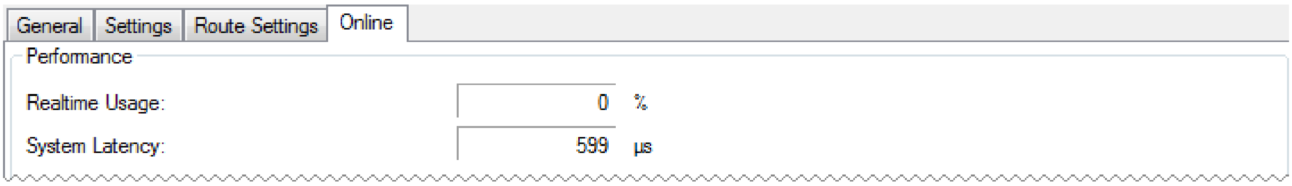
Route Settings



Route	AMS Net Id	Address	Type	Comment
CX_00E509	5.0.229.9.1.1	192.168.0.237	TCP/IP	
CX_01AABB	5.1.170.187.1.1	192.168.0.235	TCP/IP	
CX_02EB13	5.2.235.19.1.1	192.168.0.234	TCP/IP	
CX_05AFCB	5.5.175.203.1.1	CX_05AFCB	TCP/IP	

- Lists all TwinCAT route entries of the target controller.

Online



Realtime Usage: 0 %

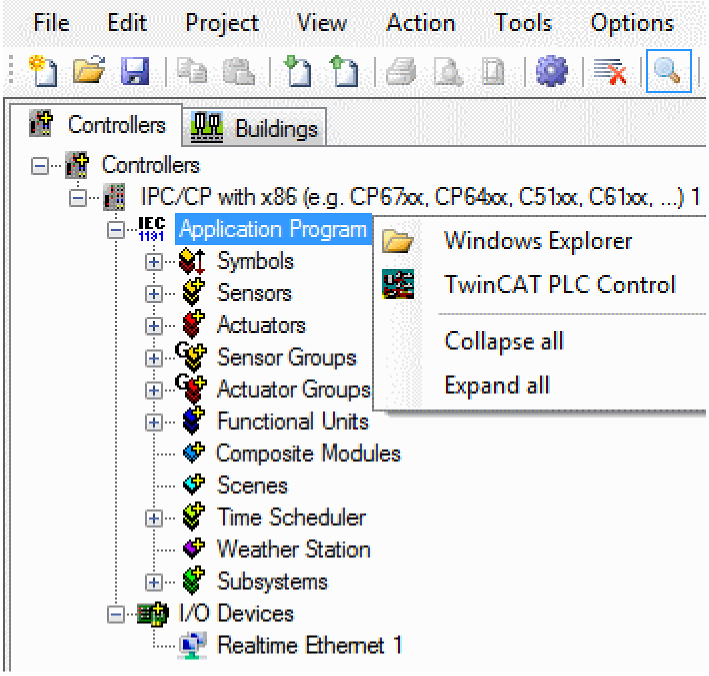
System Latency: 599 µs

- Provides live data of the controller.

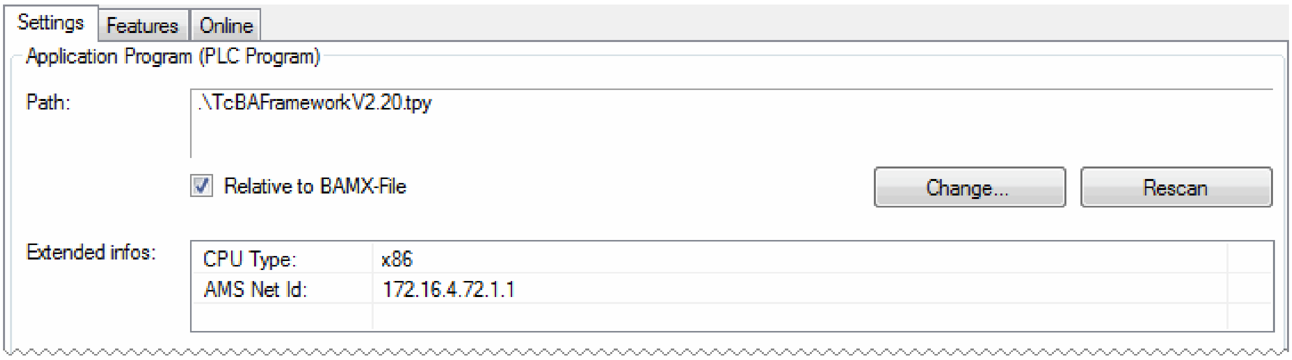
7.2.2 Application Program

7.2.2.1 Application Program Settings

By clicking on *Application Program* in the navigation tree the settings of the PLC-program belonging to the framework can be accessed. Furtheron you can open the directory of the PLC programm with a right click. Is TwinCAT Control istalled, you can open the *.pro file directly. For that the *.tpy and the *.pro file musst have the same name.



Settings



Path: .\TcBAFrameworkV2.20.tpy

☑ Relative to BAMX-File

Extended infos:

CPU Type:	x86
AMS Net Id:	172.16.4.72.1.1

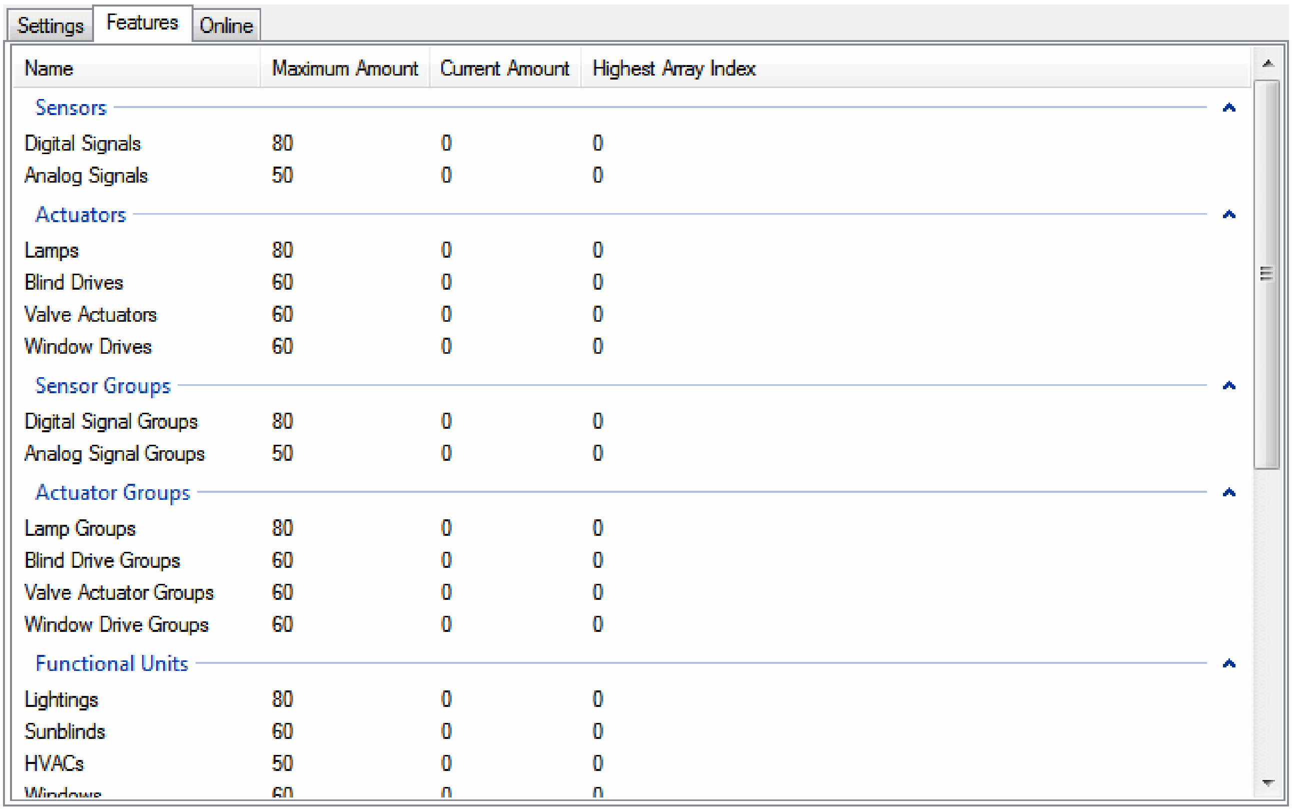
Parameters

- **Path:** If a PLC program is compiled in TwinCAT PLC Control and uploaded to a controller, a *.tpy file is generated in the PLC program directory. The path for the *.tpy file associated with this controller must be entered or selected here.

- **Relative to BAMX-File:** If this box is selected, only the (relative) path from the project file (*.bam file) (e.g. *.\TcBAFrameworkV2.20.tpy*) is stored, rather than the full (absolute) path for the *.tpy file (e.g. *C:\BAM First Steps\TcBAFrameworkV2.20.tpy*).
- **Change... / Rescan:** This option can be used to select or re-read the *.tpy file to be used.

Features

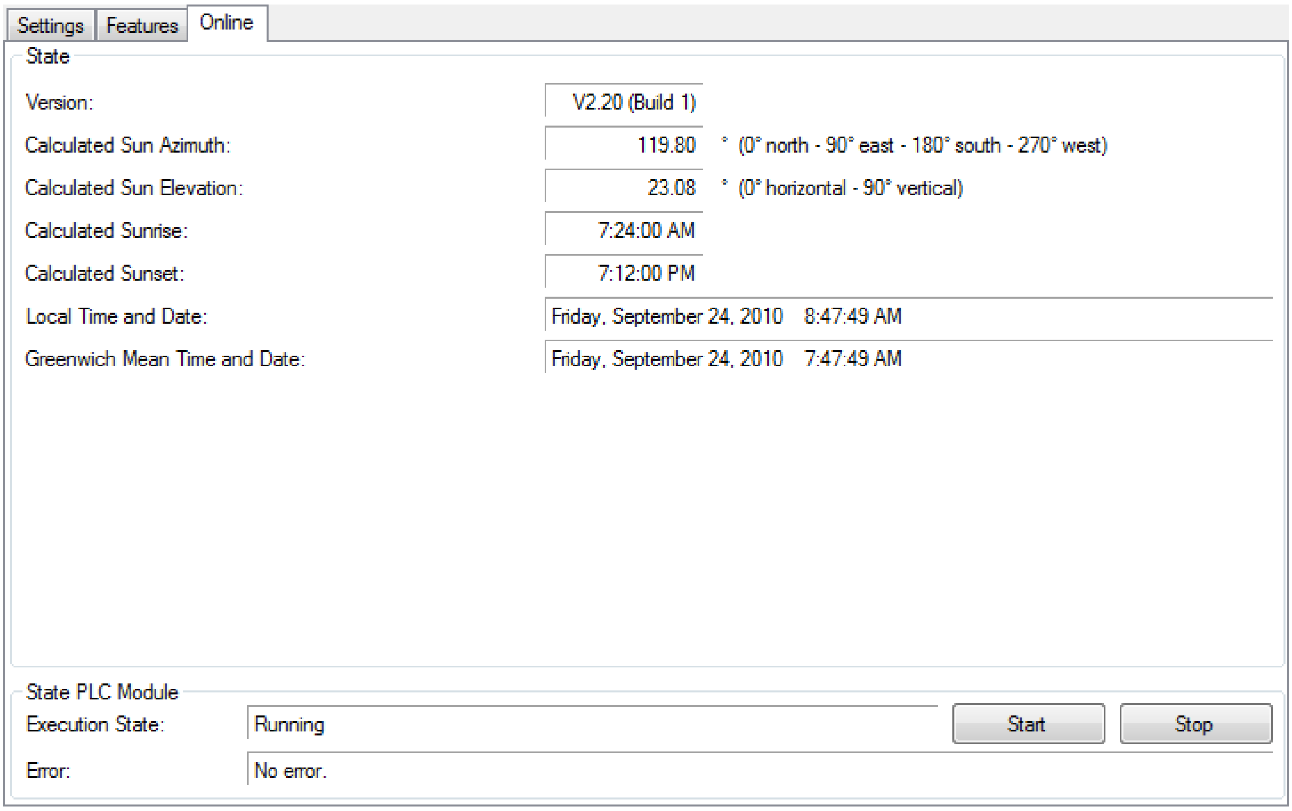
Features



Name	Maximum Amount	Current Amount	Highest Array Index
Sensors			
Digital Signals	80	0	0
Analog Signals	50	0	0
Actuators			
Lamps	80	0	0
Blind Drives	60	0	0
Valve Actuators	60	0	0
Window Drives	60	0	0
Sensor Groups			
Digital Signal Groups	80	0	0
Analog Signal Groups	50	0	0
Actuator Groups			
Lamp Groups	80	0	0
Blind Drive Groups	60	0	0
Valve Actuator Groups	60	0	0
Window Drive Groups	60	0	0
Functional Units			
Lightings	80	0	0
Sunblinds	60	0	0
HVACs	50	0	0
Windows	60	0	0

This dialog lists how many sensors, actuators etc. can be processed by the PLC and how many are in use. Moreover the highest used array indices of all module types are shown. (If the highest used index is greater than the current amount, then a module with a lower index was deleted before.)

Online



State

Version: V2.20 (Build 1)

Calculated Sun Azimuth: 119.80 ° (0° north - 90° east - 180° south - 270° west)

Calculated Sun Elevation: 23.08 ° (0° horizontal - 90° vertical)

Calculated Sunrise: 7:24:00 AM

Calculated Sunset: 7:12:00 PM

Local Time and Date: Friday, September 24, 2010 8:47:49 AM

Greenwich Mean Time and Date: Friday, September 24, 2010 7:47:49 AM

State PLC Module

Execution State: Running

Error: No error.

State

- **Version:** Indicates the current version of the running PLC.
- **Calculated Sun Azimuth:** Result of the azimuth-calculation with the given time and location-coordinates (*Project → Settings → Location*).
- **Calculated Sun Elevation:** Result of the elevation-calculation.
- **Calculated Sunrise:** Estimated sunrise.
- **Calculated Sunset:** Estimated sunset.
- **Local Time and Date:** Local time and date on the computer, where the PLC is running.
- **Greenwich Mean Time and Date:** Greenwich time (GMT). .

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter is displayed in this box.
- **Start/Stop:** Starts/stops the application program.

7.2.2.2 Symbols

Independent input/output PLC-variables are shown in the BA Manager and can be linked with any terminals (variables) with the same bit-size. These variables are listed separately (inputs and outputs) and are available under *Application Program→Symbols* in the navigation tree.

How do I add a user-defined input/output variable?

Add a PLC-variable in the *Application Program* via TwinCAT PLC Control. Arrays and structures are also supported. The position of the variable within the project is irrelevant.

The variable must be in the input or output image.

Example

```
VAR
(* will be displayed *)
    Var1 AT %IX5.3 : BOOL;
    Var2 AT %I* : INT;
    Var3 AT %QB17 :WORD;
    Var4 AT %Q* : REAL;
(* will not be displayed *)
    Var5 AT %M* : BOOL;
    Var6 : UINT;
END_VAR
```

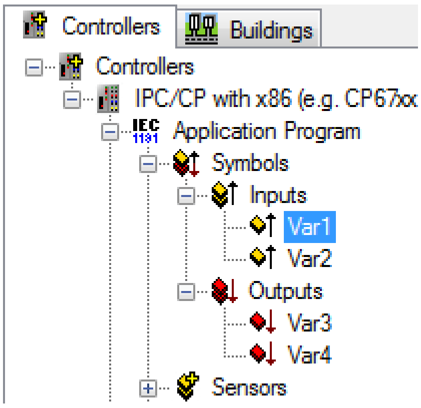
Var1 to *Var4* are displayed in the BA Manager.
Var5 is not displayed, since it is in the memory image.

Var6 is not displayed since the memory location is not in the input or output image. For further information on creating variables and allocating memory locations please refer to the TwinCAT PLC Control documentation.

Recompile the project so that the *.tpy file is recreated.

Re-read the *.tpy file in the BA Manager by selecting *Application Program* for the corresponding controller in the navigation tree and click on *Rescan* in the *Settings*-menu in the *Information and settings panel* on the right.

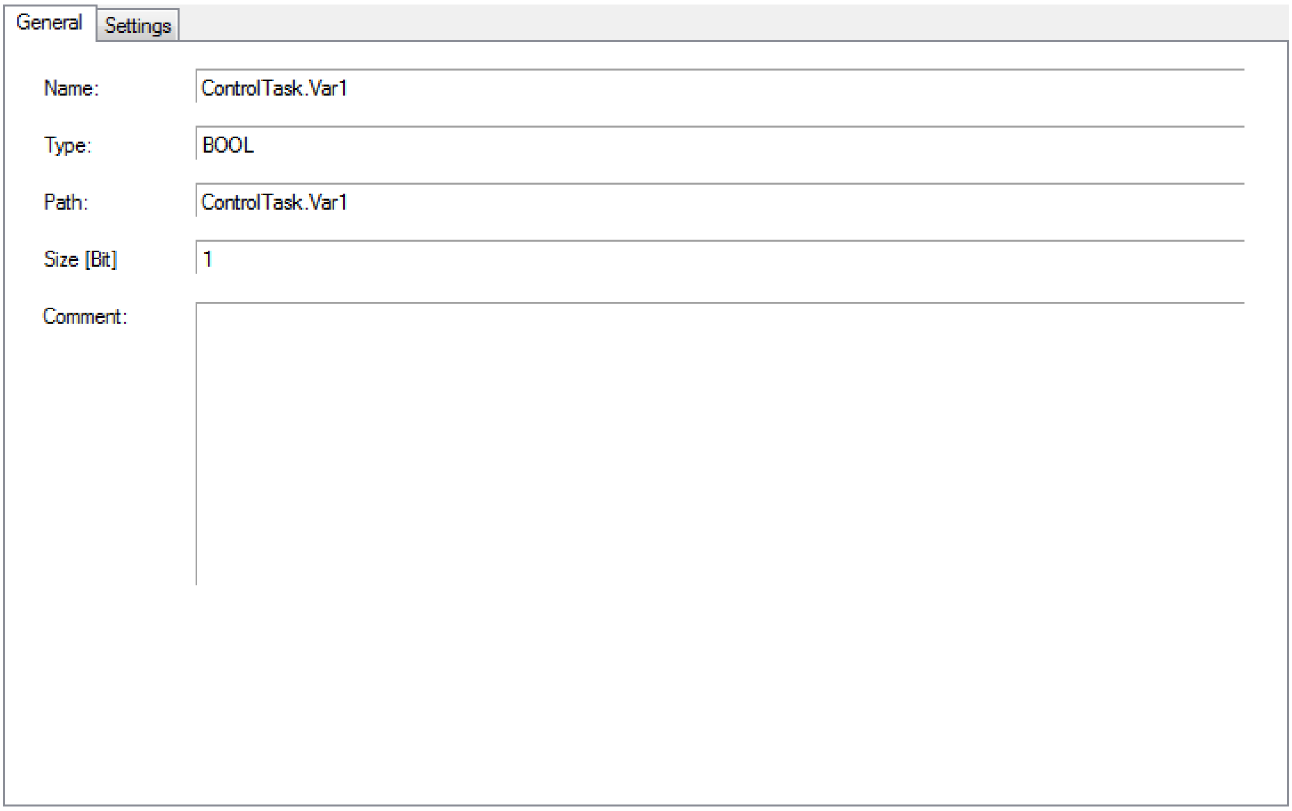
The symbols are now listed in the *Navigation tree*.



How do I configure a user-defined symbol?

- Left-click on a symbol.
- Click the *Settings*-menu in the *Information and settings panel* on the right.
- The *Terminal Assignment* section appears at the bottom of the *Settings*-menu.
- Click on the *Linked to...*-button. The Link Dialog [▶ 31] appears.

7.2.2.2.1 Symbol Settings



Name: ControlTask.Var1

Type: BOOL

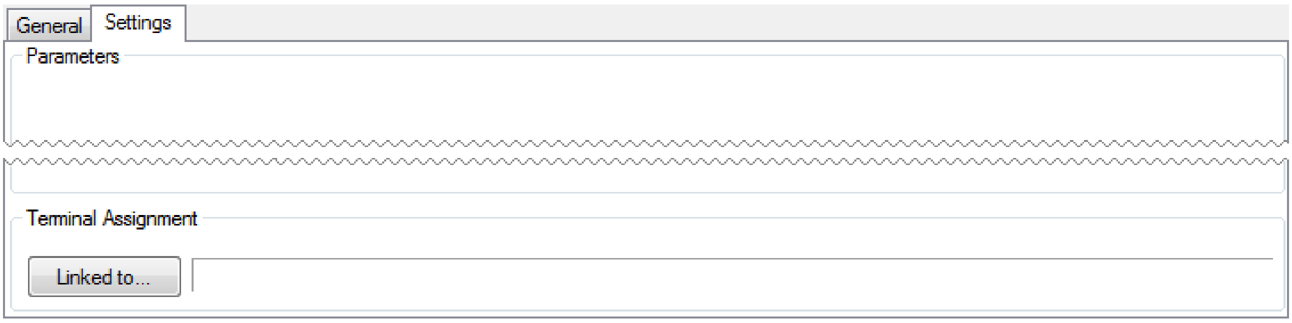
Path: ControlTask.Var1

Size [Bit] 1

Comment:

General

- **Name:** Name of the symbol in the PLC program.
- **Type:** Variable type of this symbol.
- **Path:** Path of the symbol in the PLC program.
- **Comment:** Comment for the symbol in the PLC program.



Settings

- **Terminal Assignment:** Click on *Linked to...* to link a terminal variable with the same bit size with this symbol.

7.2.2.3 Sensors

7.2.2.3.1 Digital Signals

The following types of Digital Signals are available:

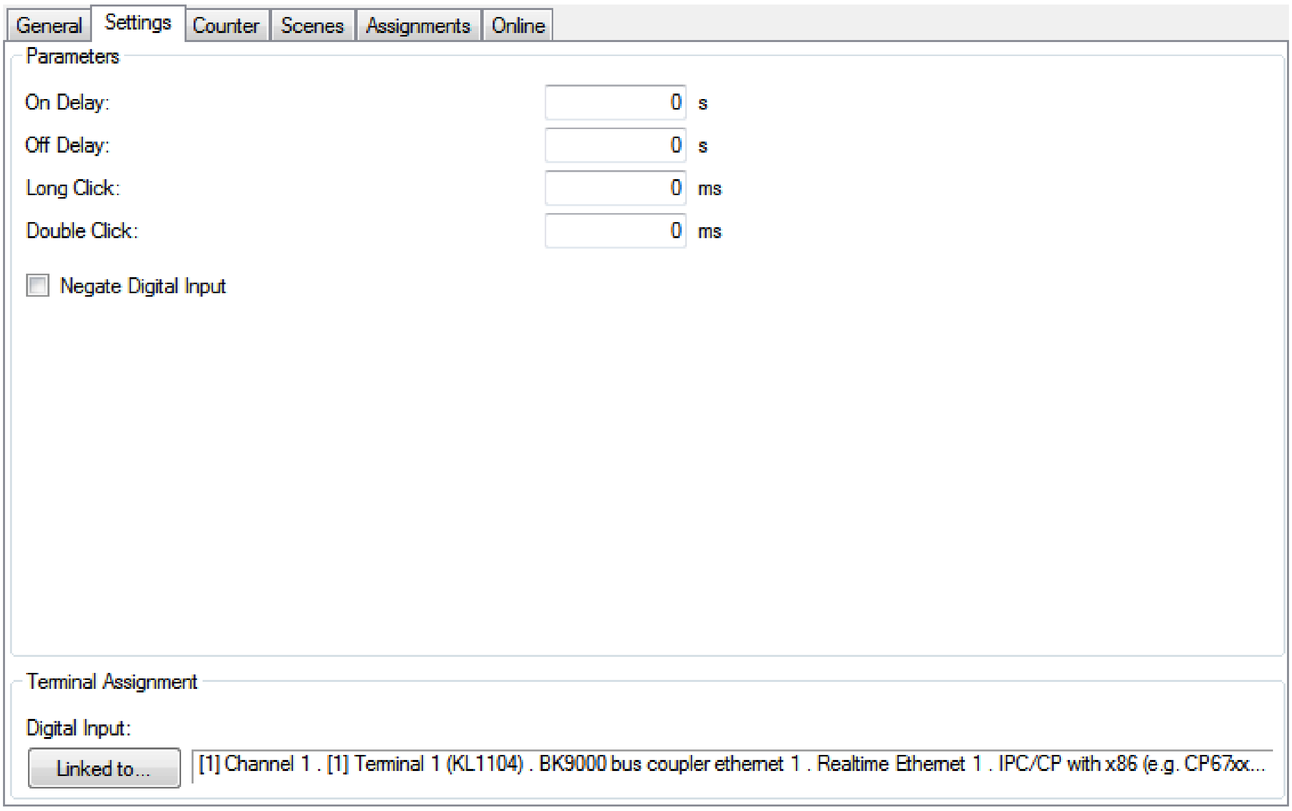
- Standard Digital Signal [▶ 61]
- EnOcean Digital Signal [▶ 63]
- DALI smartSPOT Digital Signal [▶ 66]

- Up/Down counter via KL1512 Digital Signal [▶ 69]

7.2.2.3.1.1 Standard Digital Signal

Standard Digital Signal reads and analyses digital signals assigned to standard digital input-terminals.

Settings



Parameters

On Delay: 0 s

Off Delay: 0 s

Long Click: 0 ms

Double Click: 0 ms

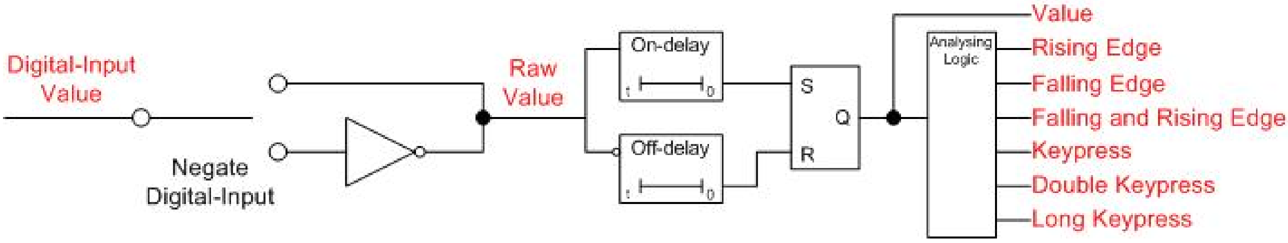
□ Negate Digital Input

Terminal Assignment

Digital Input: [1] Channel 1 . [1] Terminal 1 (KL1104) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

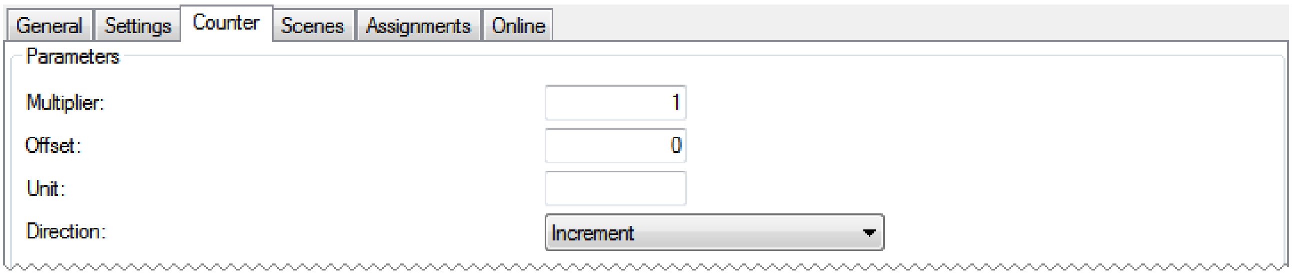
Parameters

- **On Delay:** Time delay on.
- **Off Delay:** Time delay off.
- **Long / Double Click:** Time to recognise both long- and double-click, which are used to activate scenes, see below. If the button is pressed longer than this time, it is interpreted as a long-click. Pressing the button twice within this time span will be recognised as a double-click.
- **Negate Digital Input:** Negates the digital input.



Pressing the *Linked to...*-button opens a dialog, where the designated input-terminal can be assigned to this module.

Counter



Parameters

Multiplier: 1

Offset: 0

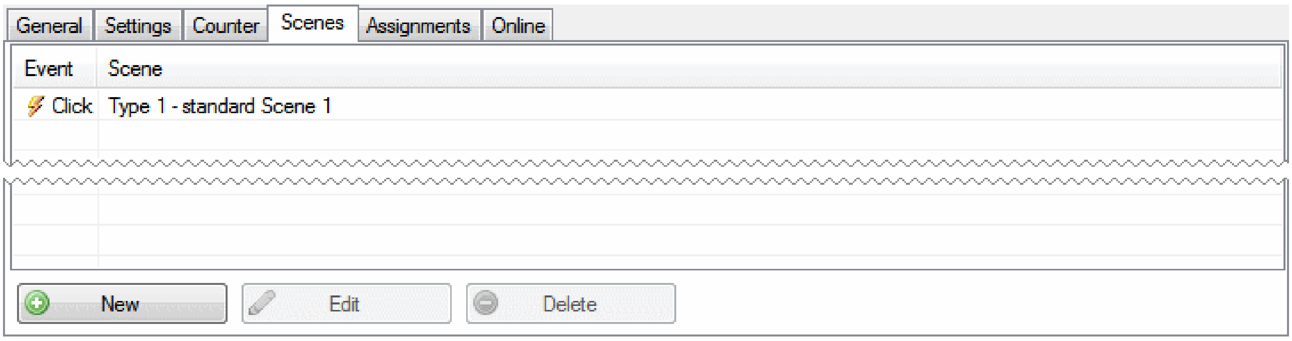
Unit:

Direction: Increment

Parameters

- **Multiplier:** Multiplier for the counter.
- **Offset:** Offset to add to the counter.
- **Unit:** The unit of the scaled value. For display-purposes in the Building Automation Framework only.
- **Direction:** Counting direction.

Scenes



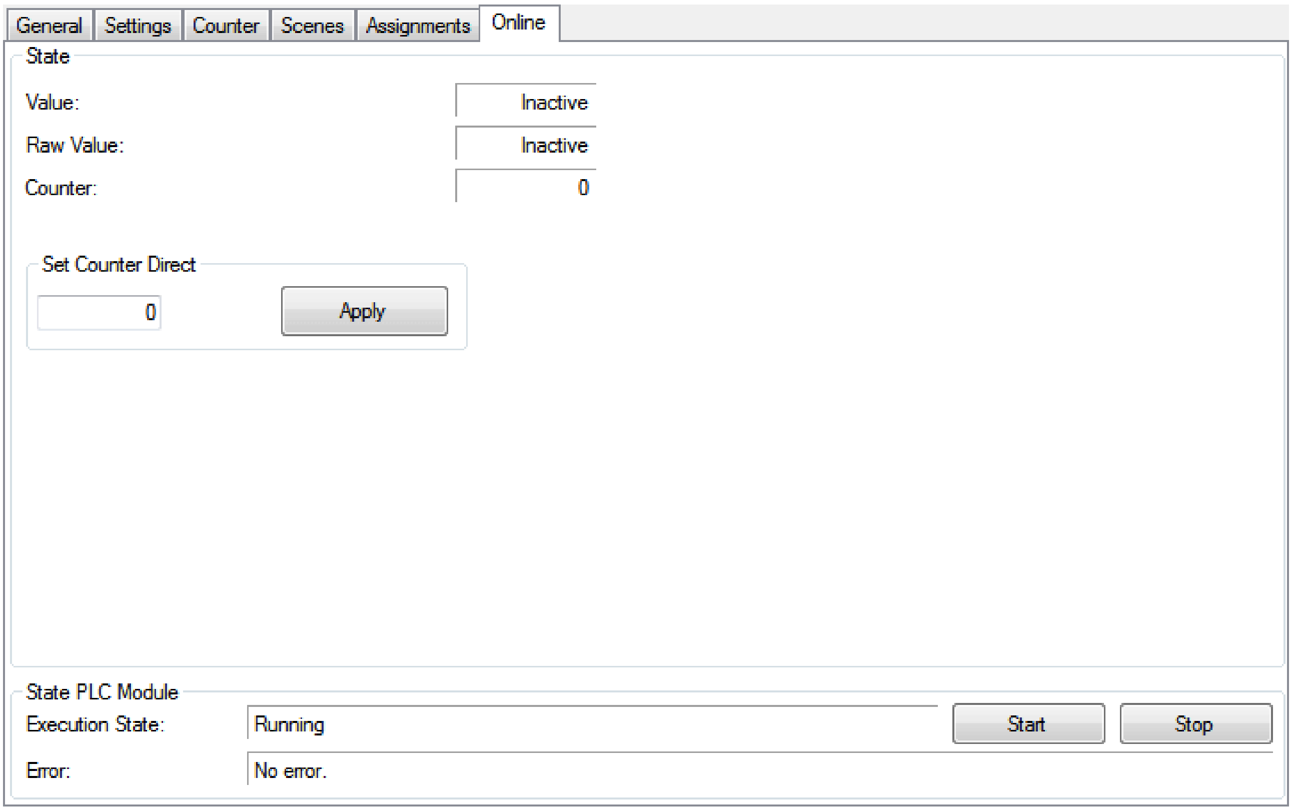
Event	Scene
Click	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the analysed events

- Rising edge
- Falling edge
- Edge changed
- Click
- Double click
- Long click

(See logical diagram above) can be assigned to the execution of a scene. The scene however must be defined previously.

Online



State

Value: Inactive

Raw Value: Inactive

Counter: 0

Set Counter Direct: 0

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Resulting value (see logical diagram above).
- **Raw Value:** Raw value received from the digital input terminal including activated / deactivated negation (see logical diagram above).
- **Counter:** Current value of the counter.
- **Set Counter Direct:** Value to set direct to the counter.

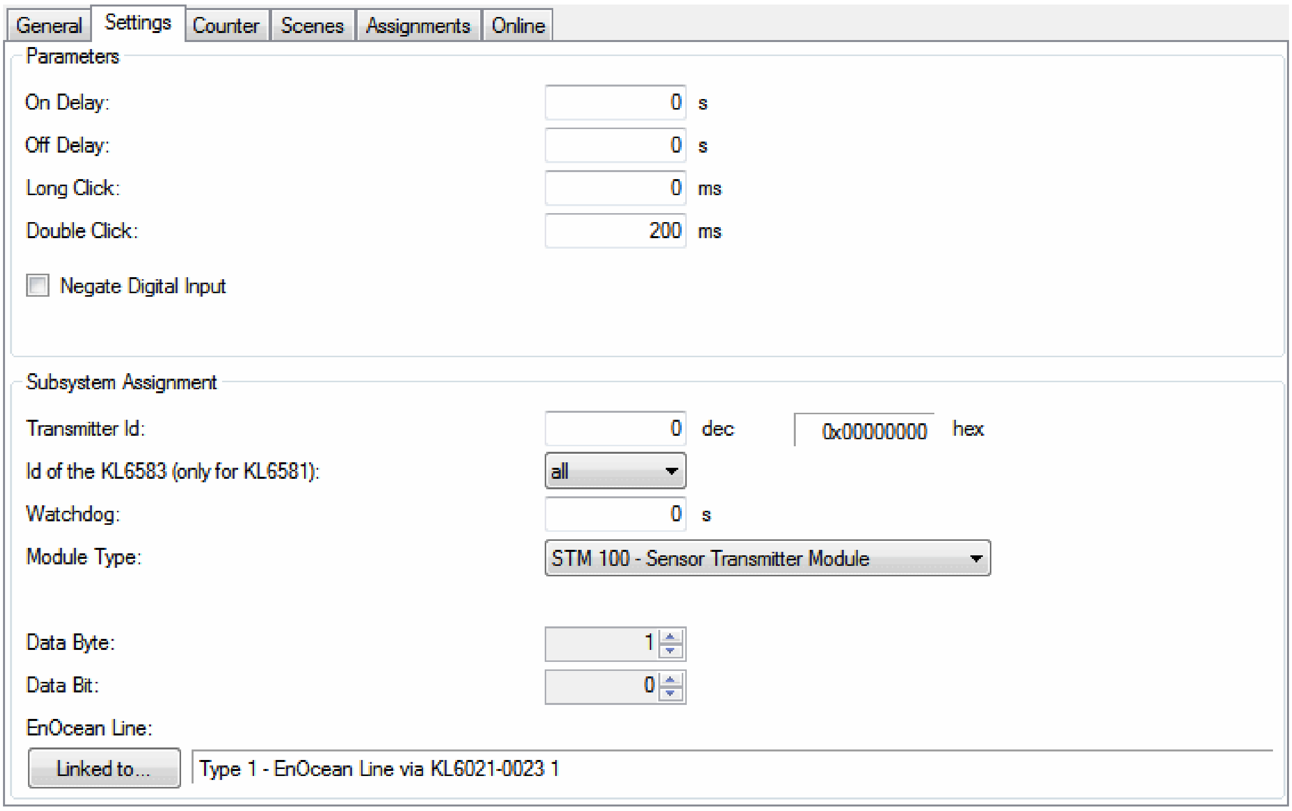
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.3.1.2 EnOcean Digital Signal

EnOcean Digital Signal analyses the telegram of a digital EnOcean transmitter.

Settings



Parameters

On Delay: 0 s

Off Delay: 0 s

Long Click: 0 ms

Double Click: 200 ms

☐ Negate Digital Input

Subsystem Assignment

Transmitter Id: 0 dec 0x00000000 hex

Id of the KL6583 (only for KL6581): all

Watchdog: 0 s

Module Type: STM 100 - Sensor Transmitter Module

Data Byte: 1

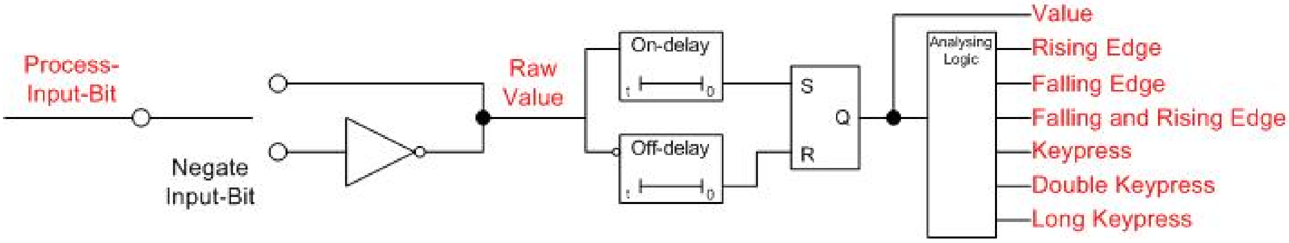
Data Bit: 0

EnOcean Line:

Linked to... Type 1 - EnOcean Line via KL6021-0023 1

Parameters

- **On Delay:** Time delay on.
- **Off Delay:** Time delay off.
- **Long / Double Click:** Time to recognize both long- and double-click, which are used to activate scenes, see below. If the button is pressed longer than this time, it is interpreted as a long-click. Pressing the button twice within this time span will be recognised as a double-click.
- **Negate Digital Input:** Negates the digital input.

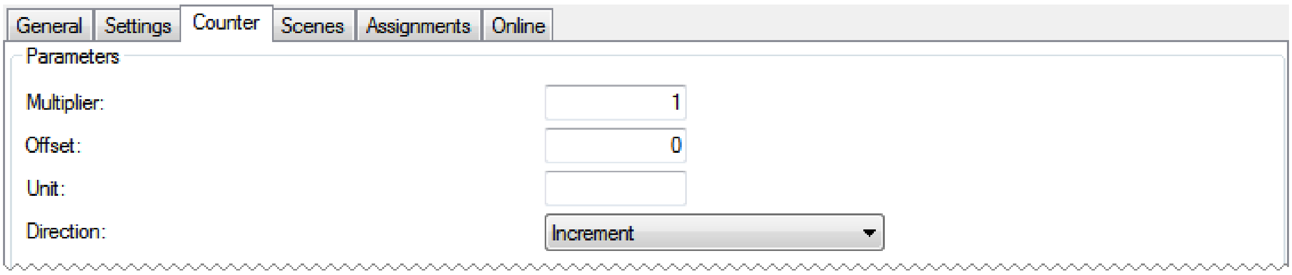


Subsystem Assignment

- **Transmitter Id:** Id of the EnOcean-transmitter.
- **Id of the KL6583 (only for 6581):** Id of the terminal.
- **Watchdog:** EnOcean-modules sending frequent telegrams, like presence-sensors, can be observed with this function. For switch-modules, however, a watchdog does not make sense and must be deactivated by entering 0s.
- **Module Type:** The correct transmitter type, as described in the manual / manufacturer-reference, must be selected here.
- **Data Byte / Data Bit:** The proper Data byte / Data Bit of the desired value have to be entered.

Pressing the *Linked to...*-button opens a dialog, where the designated EnOcean-line can be assigned to this module.

Counter



Multiplier: 1

Offset: 0

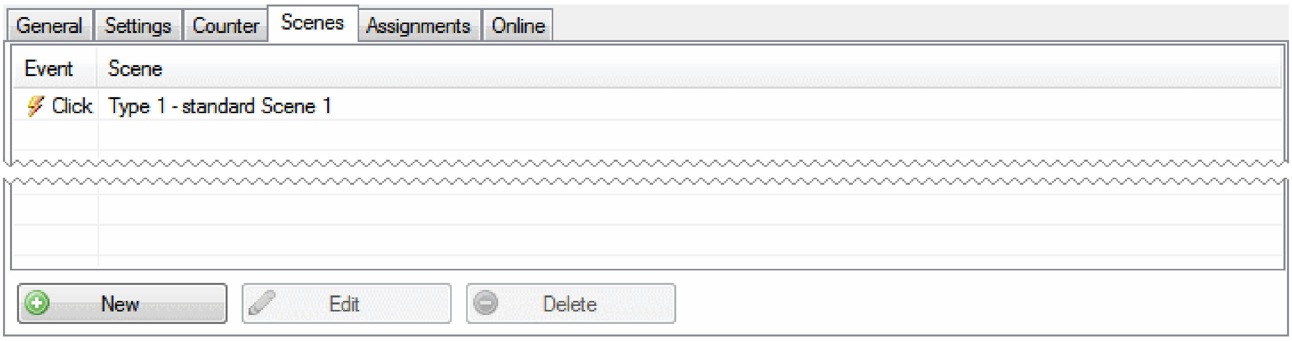
Unit:

Direction: Increment

Parameters

- **Multiplier:** Multiplier for the counter.
- **Offset:** Offset to add to the counter.
- **Unit:** The unit of the scaled value. For display-purposes in the Building Automation Framework only.
- **Direction:** Counting direction.

Scenes



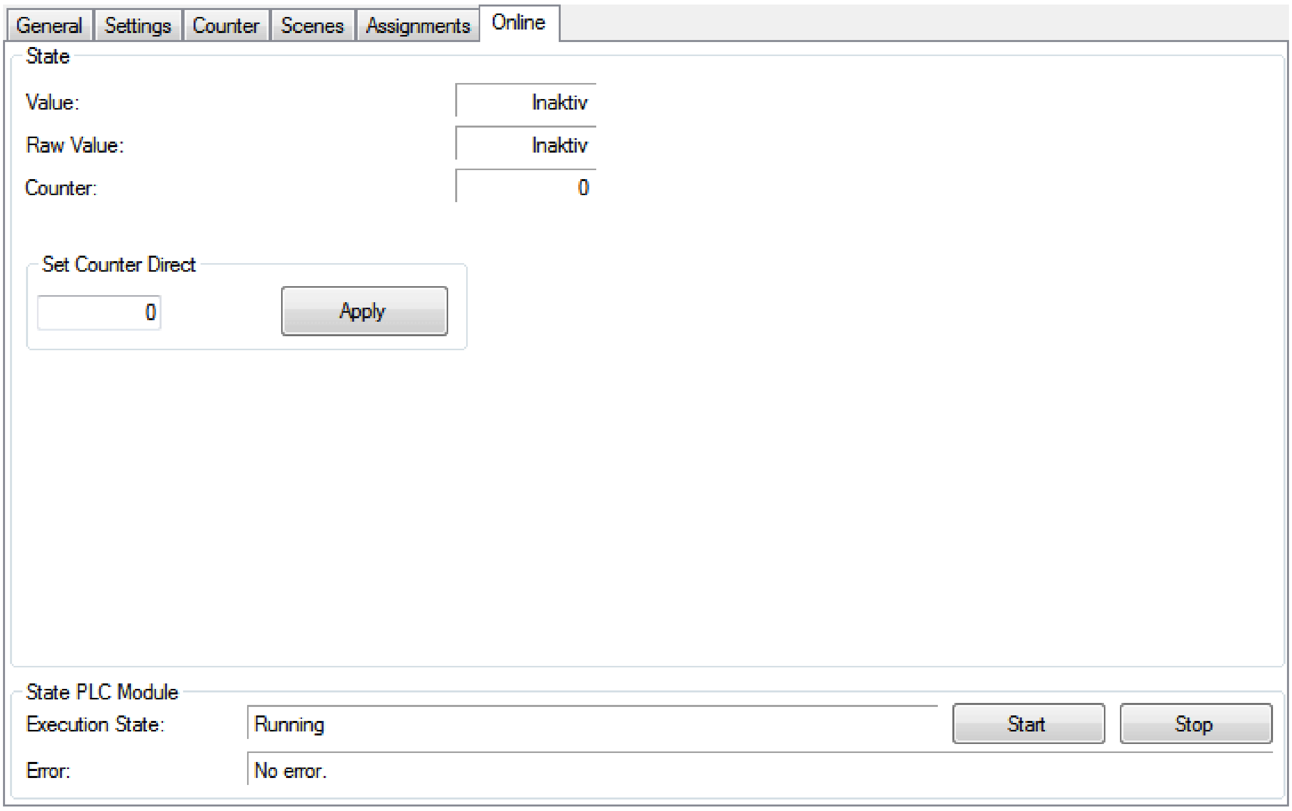
Event	Scene
Click	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the analysed events

- Rising edge
- Falling edge
- Edge changed
- Click
- Double click
- Long click

(See logical diagram above) can be assigned to the execution of a scene. The scene however must be defined previously.

Online



State

Value: Inaktiv

Raw Value: Inaktiv

Counter: 0

Set Counter Direct

0

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Resulting value, see logical diagram above.
- **Raw Value:** Raw value received from the digital input terminal including activated / deactivated negation (see logical diagram above).
- **Counter:** Current value of the counter.
- **Set Counter Direct:** Value to set direct to the counter.

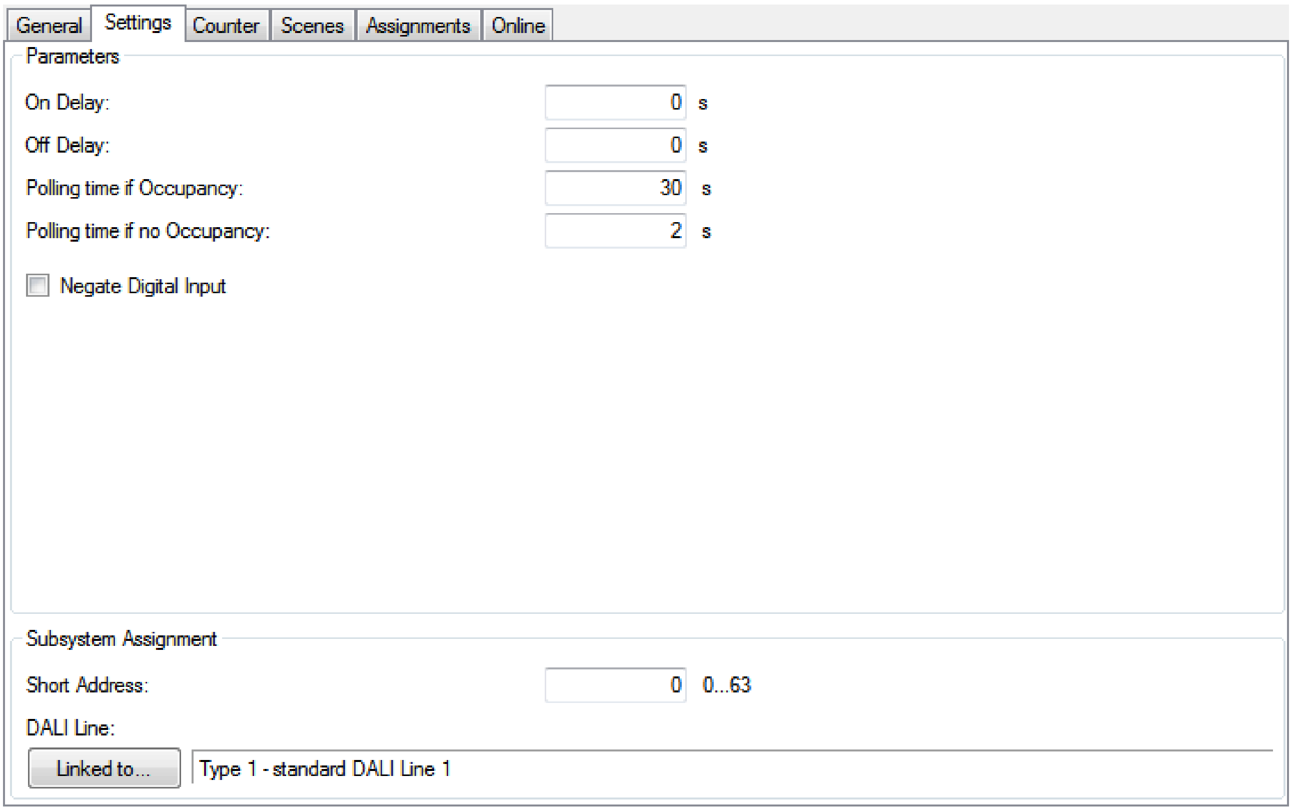
State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.3.1.3 DALI smartSPOT Digital Signal

The digital signal DALI smartSpot evaluates the presence indication of a DALI SmartSpot.

Settings



Parameters

On Delay: 0 s

Off Delay: 0 s

Polling time if Occupancy: 30 s

Polling time if no Occupancy: 2 s

□ Negate Digital Input

Subsystem Assignment

Short Address: 0 0...63

DALI Line:

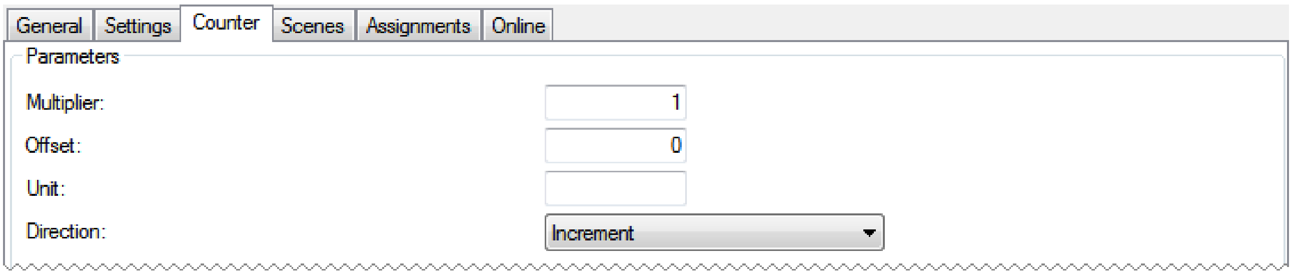
Linked to... Type 1 - standard DALI Line 1

Parameters

- **On Delay:** Time delay on.
- **Off Delay:** Time delay off.
- **Polling time if Occupancy:** Time if occupancy was detected.
- **Polling time if no Occupancy:** Time if no occupancy was detected.
- **Negate Digital Input:** Negates the digital input.

Pressing the *Linked to...*-button opens a dialog, where the designated DALI-line can be assigned to this module.

Counter



Parameters

Multiplier: 1

Offset: 0

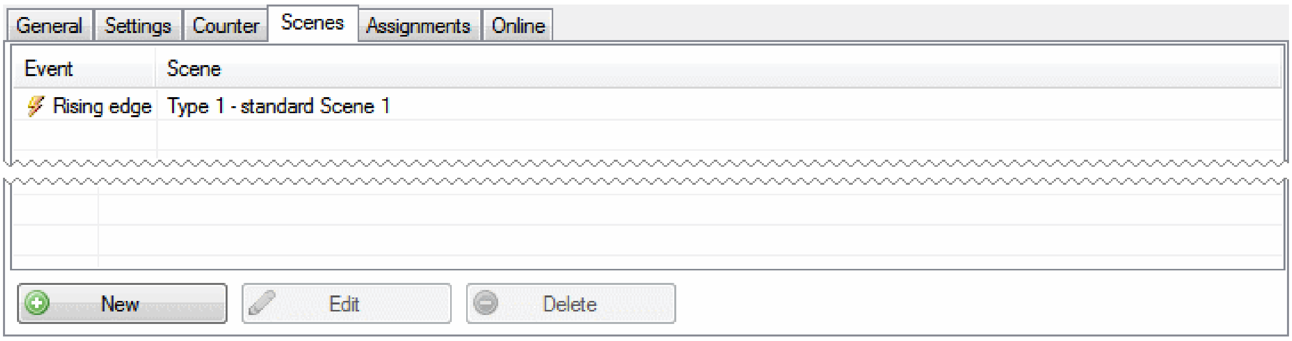
Unit:

Direction: Increment

Parameters

- **Multiplier:** Multiplier for the counter.
- **Offset:** Offset to add to the counter.
- **Unit:** The unit of the scaled value. For display-purposes in the Building Automation Framework only.
- **Direction:** Counting direction.

Scenes



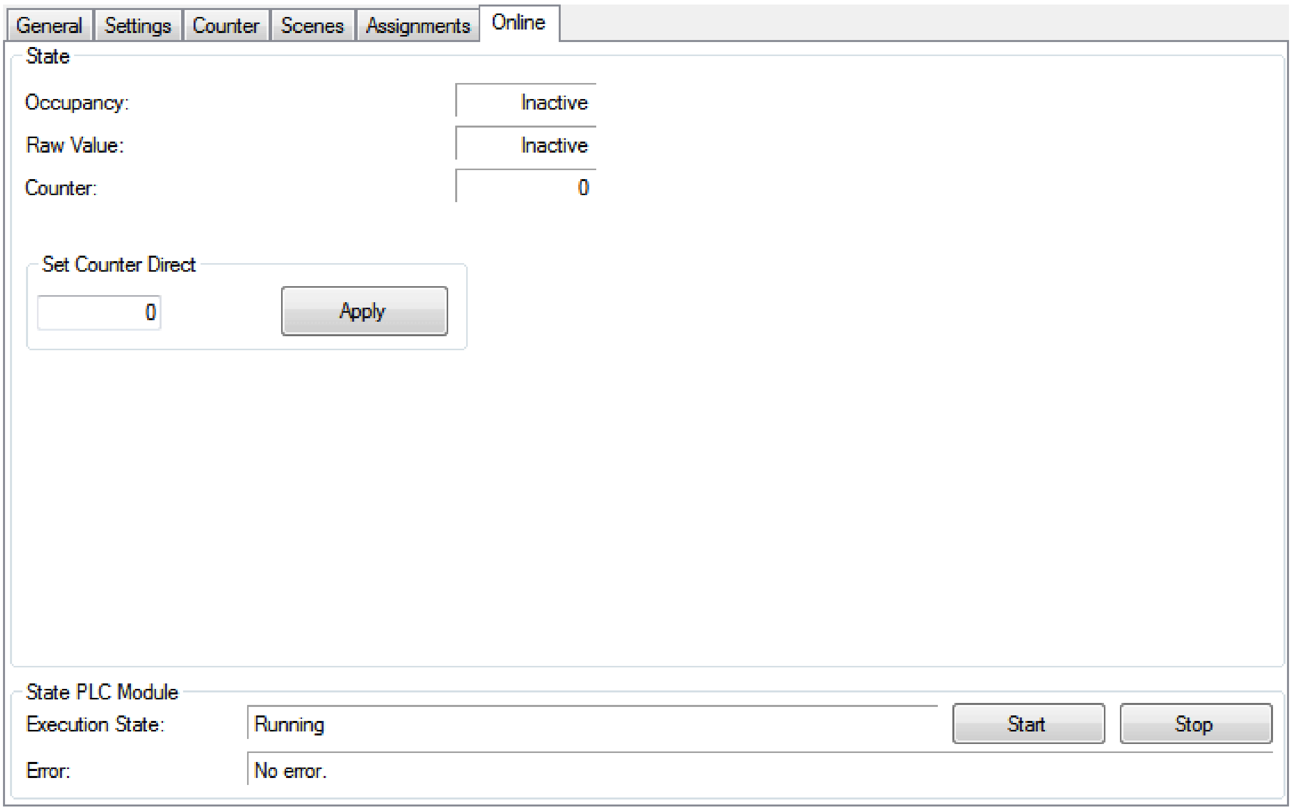
Event	Scene
Rising edge	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the analysed events

- Rising edge
- Falling edge
- Edge changed

can be assigned to the execution of a scene. The scene however has to be defined previously.

Online



Occupancy: Inactive

Raw Value: Inactive

Counter: 0

Set Counter Direct: 0

Execution State: Running

Error: No error.

State

- **Occupancy:** Occupancy detection.
- **Raw Value:** Raw value received from the digital input terminal including activated / deactivated negation.
- **Counter:** Current value of the counter.
- **Set Counter Direct:** Value to set direct to the counter.

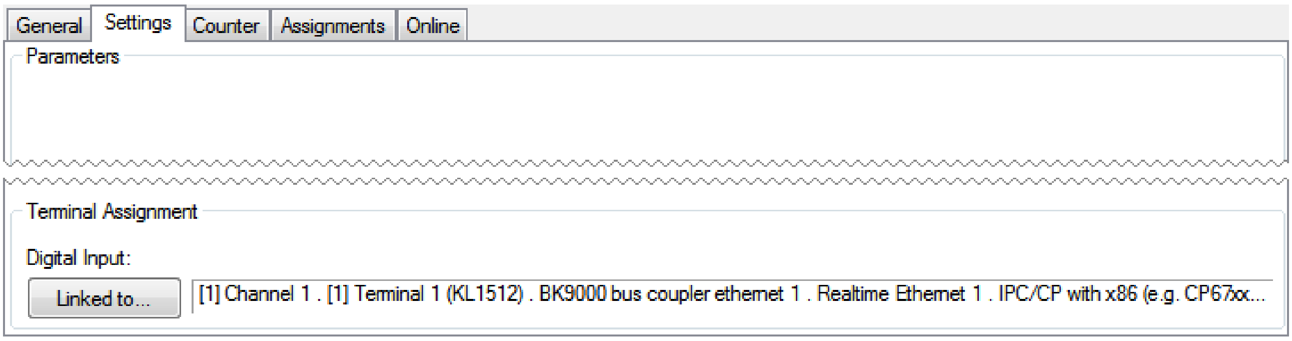
State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.3.1.4 Up/Down counter via KL1512 Digital Signal

The digital signal Up/Down counter is connected to the counter terminal KL1512 and provides its current count.

Settings



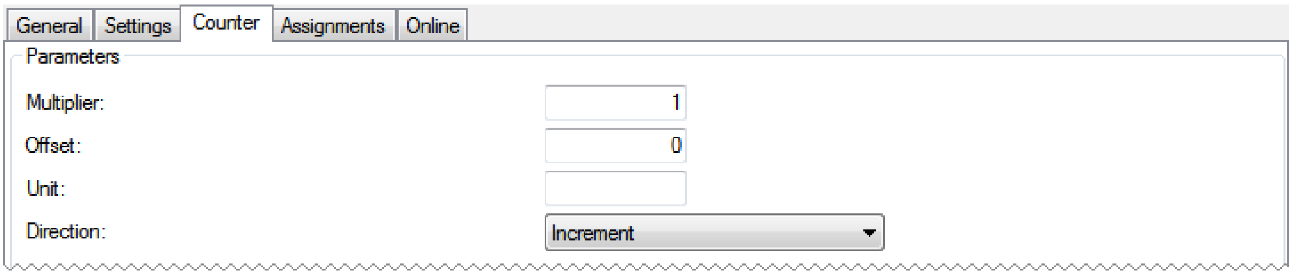
Parameters

Terminal Assignment

Digital Input: [1] Channel 1 . [1] Terminal 1 (KL1512) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Pressing the *Linked to...*-button opens a dialog, where the designated counter-terminal can be assigned to this module.

Counter



Parameters

Multiplier: 1

Offset: 0

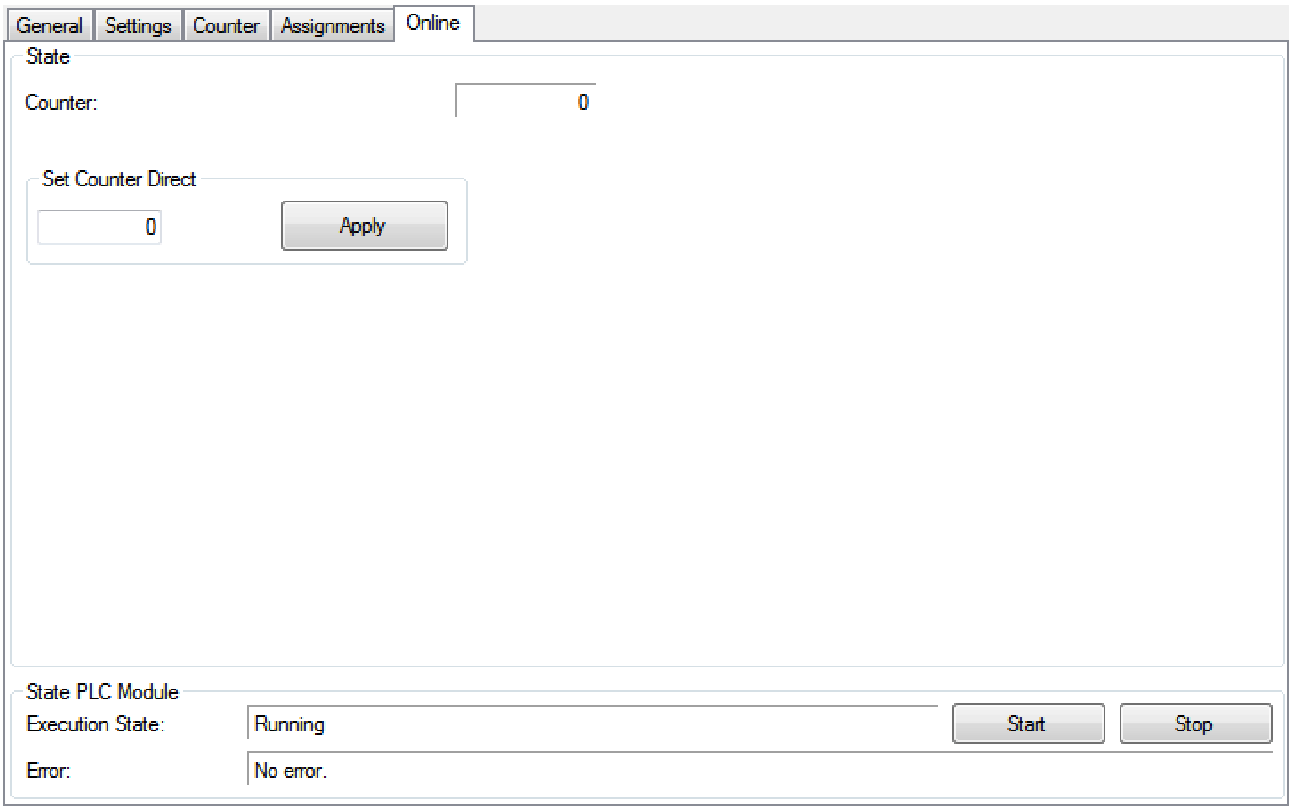
Unit:

Direction: Increment

Parameters

- **Multiplier:** Multiplier for the counter.
- **Offset:** Offset to add to the counter.
- **Unit:** The unit of the scaled value. For display-purposes in the Building Automation Framework only.
- **Direction:** Counting direction.

Online



Counter: 0

Set Counter Direct: 0

Execution State: Running

Error: No error.

State

- **Counter:** Current value of the counter.
- **Set Counter Direct:** Value to set direct to the counter.

State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.3.2 Analog Signals

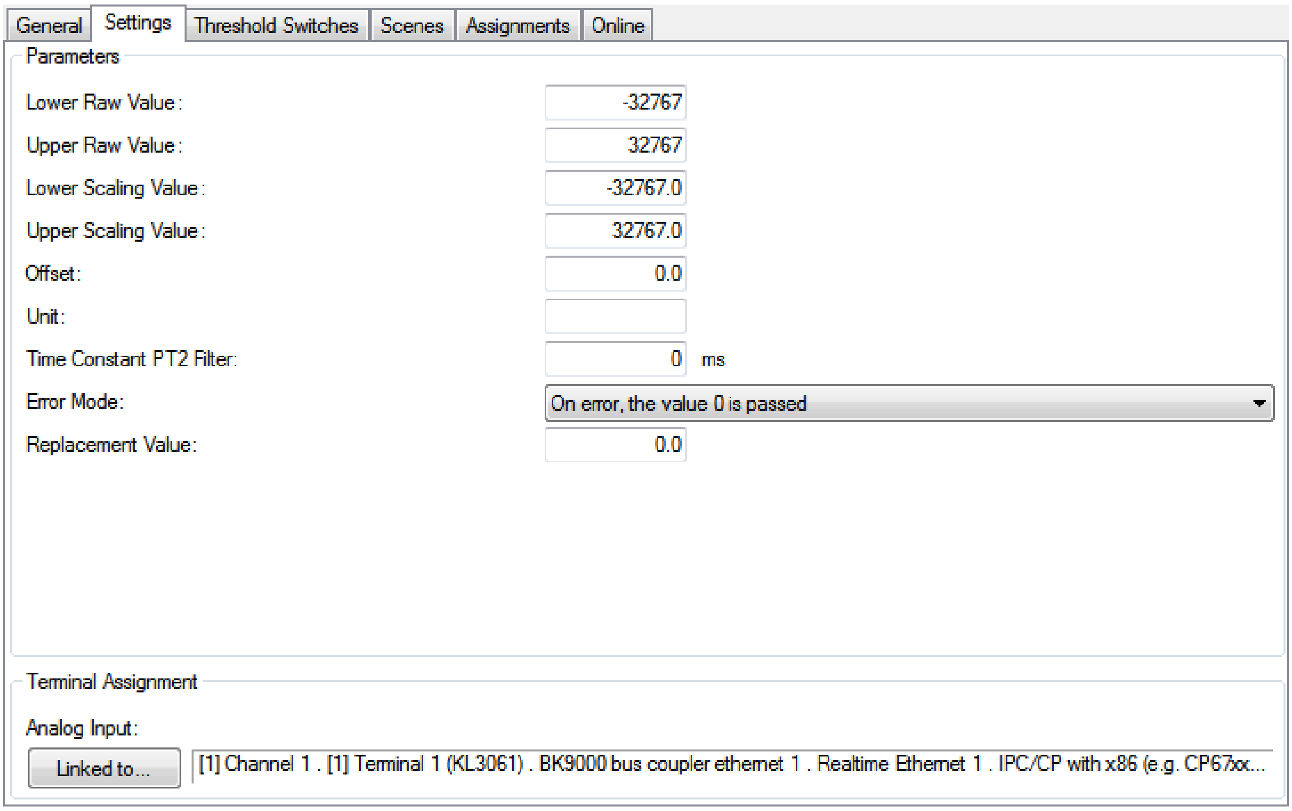
The following types of Analog Signals are available:

- Standard Analog Signal [▶ 70]
- EnOcean Analog Signal [▶ 73]
- DALI smartSPOT Analog Signal [▶ 76]

7.2.2.3.2.1 Standard Analog Signal

The Standard Analog Signal analyses the value of an analog input terminal.

Settings



Parameters

Lower Raw Value: -32767

Upper Raw Value: 32767

Lower Scaling Value: -32767.0

Upper Scaling Value: 32767.0

Offset: 0.0

Unit:

Time Constant PT2 Filter: 0 ms

Error Mode: On error, the value 0 is passed

Replacement Value: 0.0

Terminal Assignment

Analog Input:

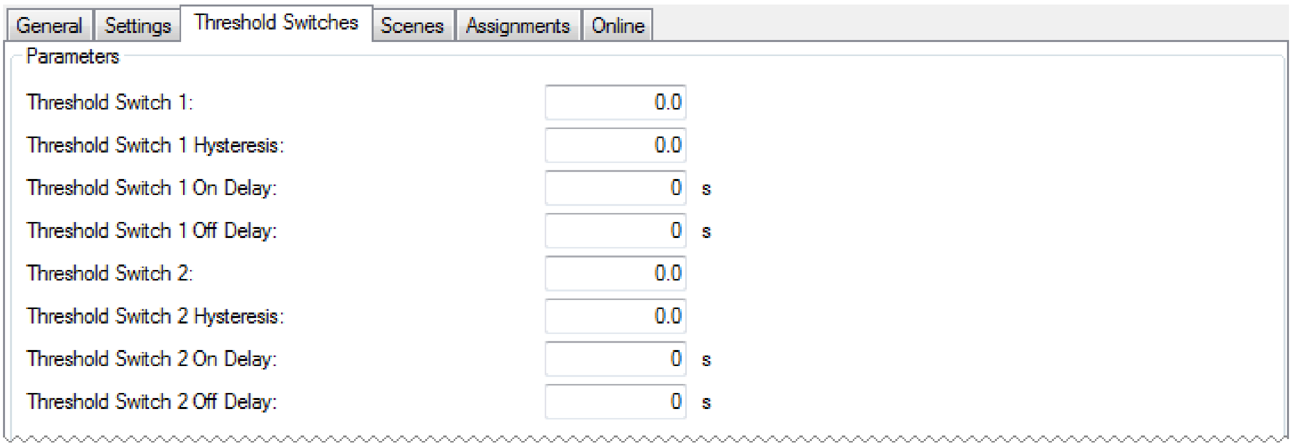
[1] Channel 1 . [1] Terminal 1 (KL3061) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Parameters

- **Lower Raw Value,**
  Upper Raw Value,
  Lower Scaled Value,
  Upper Scaled Value: These parameters build up the linear equation for the conversion raw value ↔ scaled value.
  The *Lower Raw Value* must always be smaller than the *Upper Raw Value*.
- **Offset:** Basically, the relationship between raw value and scaled value is clearly defined by the linear equation, described above. This offset will be added to the calculated scaled value for fine-tuning.
- **Unit:** The unit of the scaled value. For display-purposes in the Building Automation Framework only.
- **Time Constant PT2 Filter:** If necessary, the calculated scaled value can be damped by a PT2-Filter. If this feature is not used, the time has to be set to 0ms.
- **Error Mode:** Determine how errors are handled.
- **Replacement Value:** In case of an error, this value is used instead.

Pressing the *Linked to...*-button opens a dialog, where the designated input-terminal can be assigned to this module.

Threshold Switches



Threshold Switch 1: 0.0

Threshold Switch 1 Hysteresis: 0.0

Threshold Switch 1 On Delay: 0 s

Threshold Switch 1 Off Delay: 0 s

Threshold Switch 2: 0.0

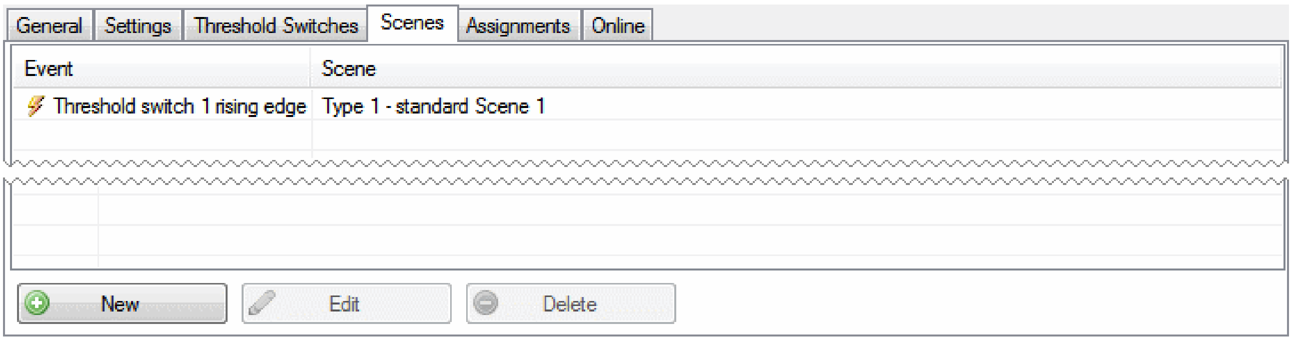
Threshold Switch 2 Hysteresis: 0.0

Threshold Switch 2 On Delay: 0 s

Threshold Switch 2 Off Delay: 0 s

Two threshold-switches can be defined for the scaled value. The rising and falling edges of these switches can be used to execute scenes (see *Scenes*-menu below). Additionally a hysteresis and delay can be defined to dampen the behaviour of the scaled value.

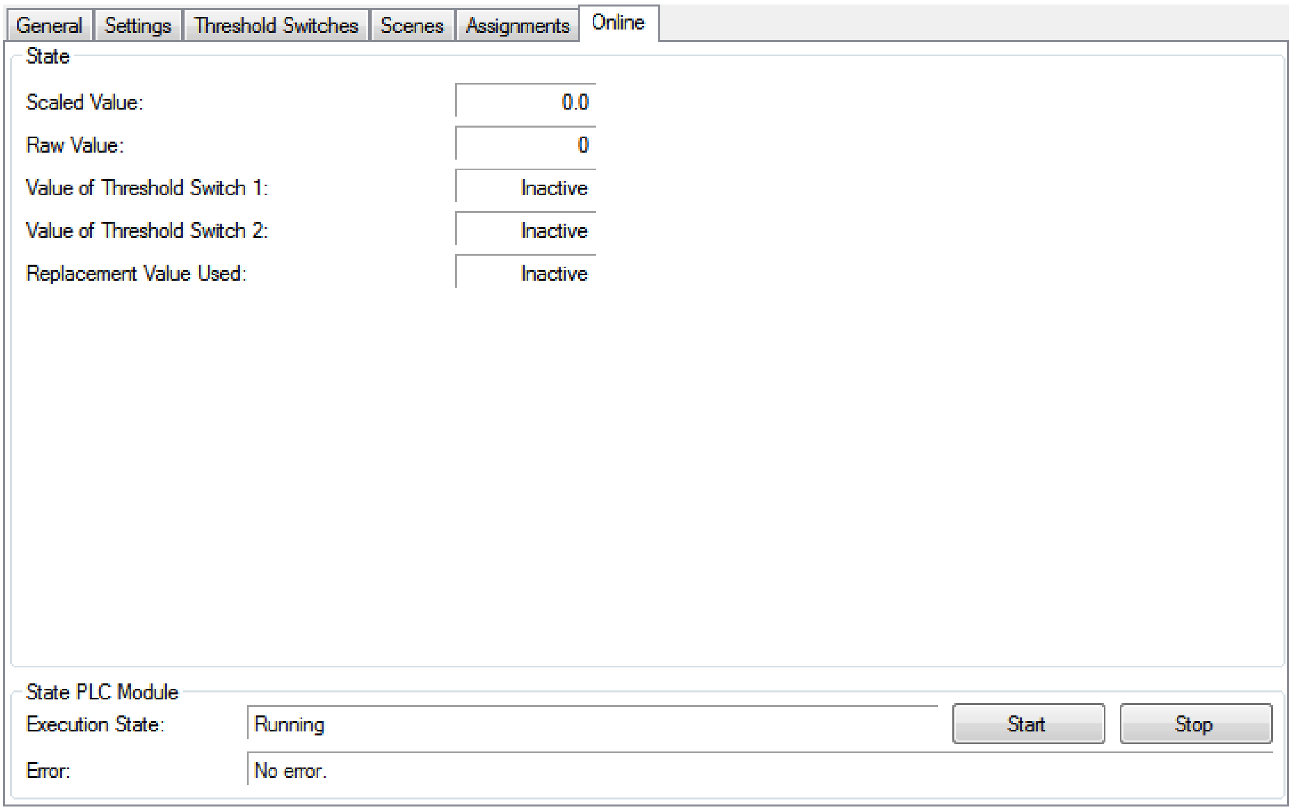
Scenes



Event	Scene
⚡ Threshold switch 1 rising edge	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the rising and the falling edge events of the threshold-switches can be assigned to the execution of a scene. The scene however has to be defined previously.

Online



State

Scaled Value: 0.0

Raw Value: 0

Value of Threshold Switch 1: Inactive

Value of Threshold Switch 2: Inactive

Replacement Value Used: Inactive

State PLC Module

Execution State: Running

Error: No error.

State

- **Scaled Value:** Calculated scaled value as described in the settings menu (raw value → linear equation → additional offset → PT2-filter → scaled value).
- **Raw Value:** Raw value received from the analog input terminal.
- **Value of Threshold Switch 1/2:** Active if the scaled value exceeds the threshold, otherwise inactive.
- **Replacement Value Used:** Indicates whether the *Replacement Value* is used.

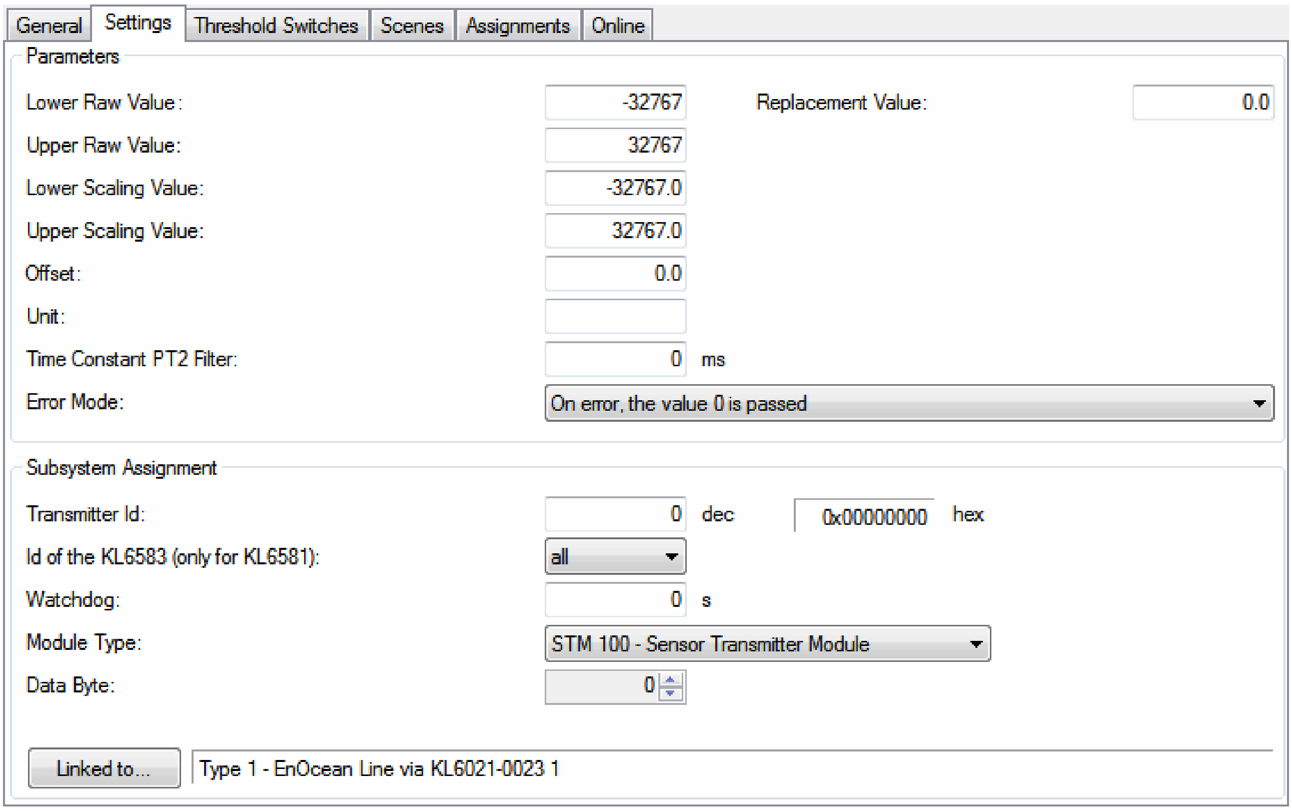
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.3.2.2 EnOcean Analog Signal

The EnOcean Analog Signal analyses the analog output data of an EnOcean transmitter.

Settings



Parameters

Lower Raw Value: -32767

Replacement Value: 0.0

Upper Raw Value: 32767

Lower Scaling Value: -32767.0

Upper Scaling Value: 32767.0

Offset: 0.0

Unit:

Time Constant PT2 Filter: 0 ms

Error Mode: On error, the value 0 is passed

Subsystem Assignment

Transmitter Id: 0 dec 0x00000000 hex

Id of the KL6583 (only for KL6581): all

Watchdog: 0 s

Module Type: STM 100 - Sensor Transmitter Module

Data Byte: 0

Type 1 - EnOcean Line via KL6021-0023 1

Parameters

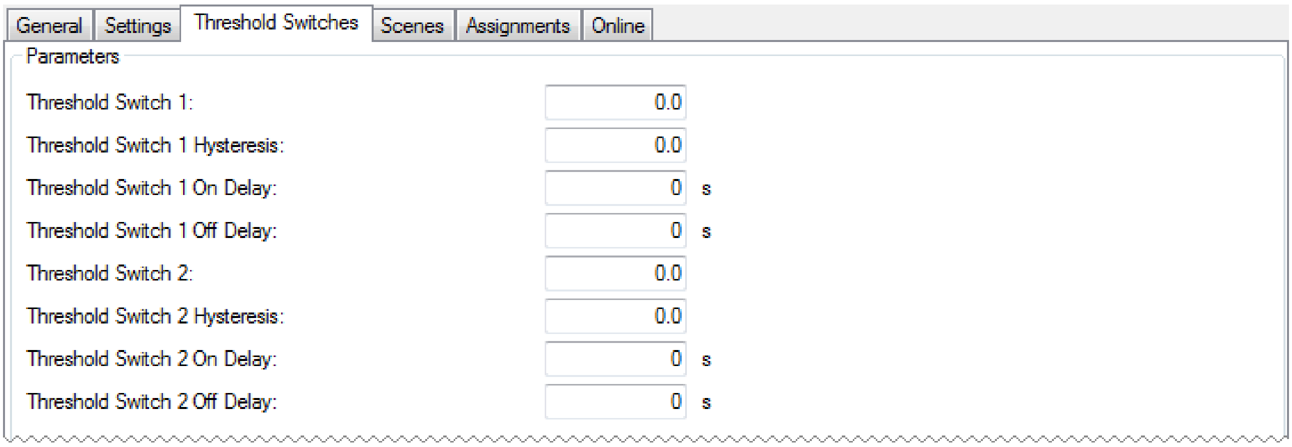
- **Lower Raw Value,**
  Upper Raw Value,
  Lower Scaled Value,
  Upper Scaled Value: These parameters build up the linear equation for the conversion raw value ↔ scaled value.
  The *Lower Raw Value* must always be smaller than the *Upper Raw Value*.
- **Offset:** Basically, the relationship between raw value and scaled value is clearly defined by the linear equation, described above. This offset will be added to the calculated scaled value for fine-tuning.
- **Unit:** The unit of the scaled value. For display-purposes in the Building Automation Framework only.
- **Time Constant PT2 Filter:** If necessary, the calculated scaled value can be damped by a PT2-Filter. If this feature is not used, the time has to be set to 0ms.
- **Error Mode:** Determine how errors are handled.
- **Replacement Value:** In case of an error, this value is used instead.

Subsystem Assignment

- **Transmitter Id:** Id of the EnOcean transmitter.
- **Id of the KL6583 (only for KL6581):** Id of the terminal.
- **Watchdog:** EnOcean-modules sending frequent telegrams, like temperature-sensors, can be observed with this function. For switch-modules, however, a watchdog does not make sense and has to be deactivated by entering 0s.
- **Module Type:** The correct transmitter type, as described in the manual / manufacturer-reference, has to be selected here.
- **Data Byte:** The proper Data byte of the desired value has to be entered.

Pressing the *Linked to...*-button opens a dialog, where the designated EnOcean-line can be assigned to this module.

Threshold Switches

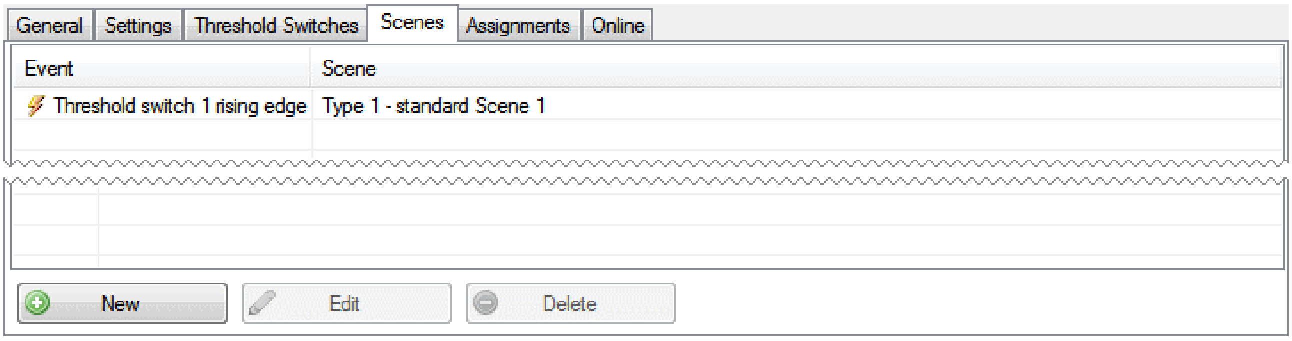


Parameters

Threshold Switch 1:	0.0	
Threshold Switch 1 Hysteresis:	0.0	
Threshold Switch 1 On Delay:	0	s
Threshold Switch 1 Off Delay:	0	s
Threshold Switch 2:	0.0	
Threshold Switch 2 Hysteresis:	0.0	
Threshold Switch 2 On Delay:	0	s
Threshold Switch 2 Off Delay:	0	s

Two threshold-switches can be defined for the scaled value. The rising and falling edges of these switches can be used to execute scenes (see *Scenes*-menu below).
Additionally a hysteresis and delay can be defined to dampen the behaviour of the scaled value.

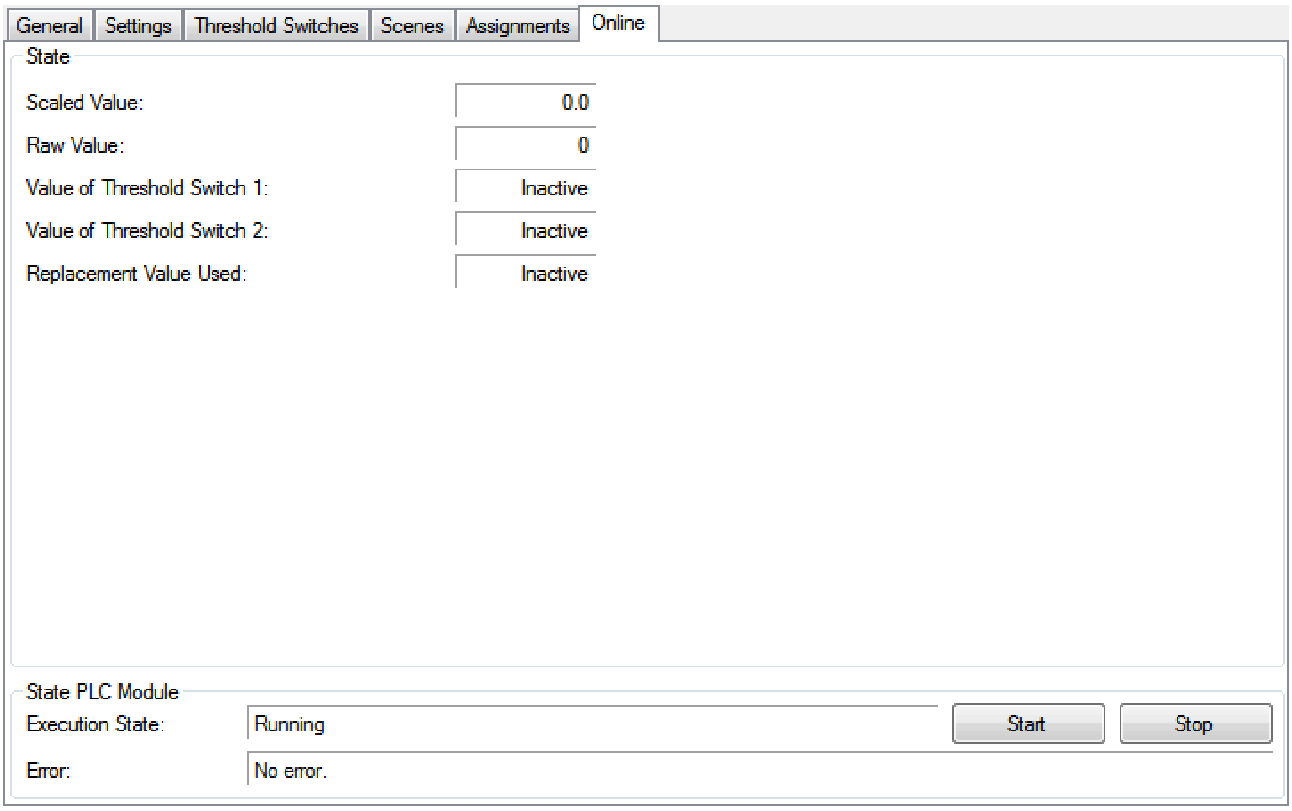
Scenes



Event	Scene
⚡ Threshold switch 1 rising edge	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the rising and the falling edge events of the threshold-switches can be assigned to the execution of a scene. The scene however must be defined previously.

Online



State

Scaled Value: 0.0

Raw Value: 0

Value of Threshold Switch 1: Inactive

Value of Threshold Switch 2: Inactive

Replacement Value Used: Inactive

State PLC Module

Execution State: Running

Error: No error.

State

- **Scaled Value:** Calculated scaled value as described in the settings-menu. (raw value → linear equation → additional offset → PT2-filter → scaled value)
- **Raw Value:** Raw value received from the analog input-terminal.
- **Value of Threshold Switch 1/2:** Active if the scaled value exceeds the threshold, otherwise inactive.
- **Replacement Value Used:** Indicates whether the *Replacement Value* is used.

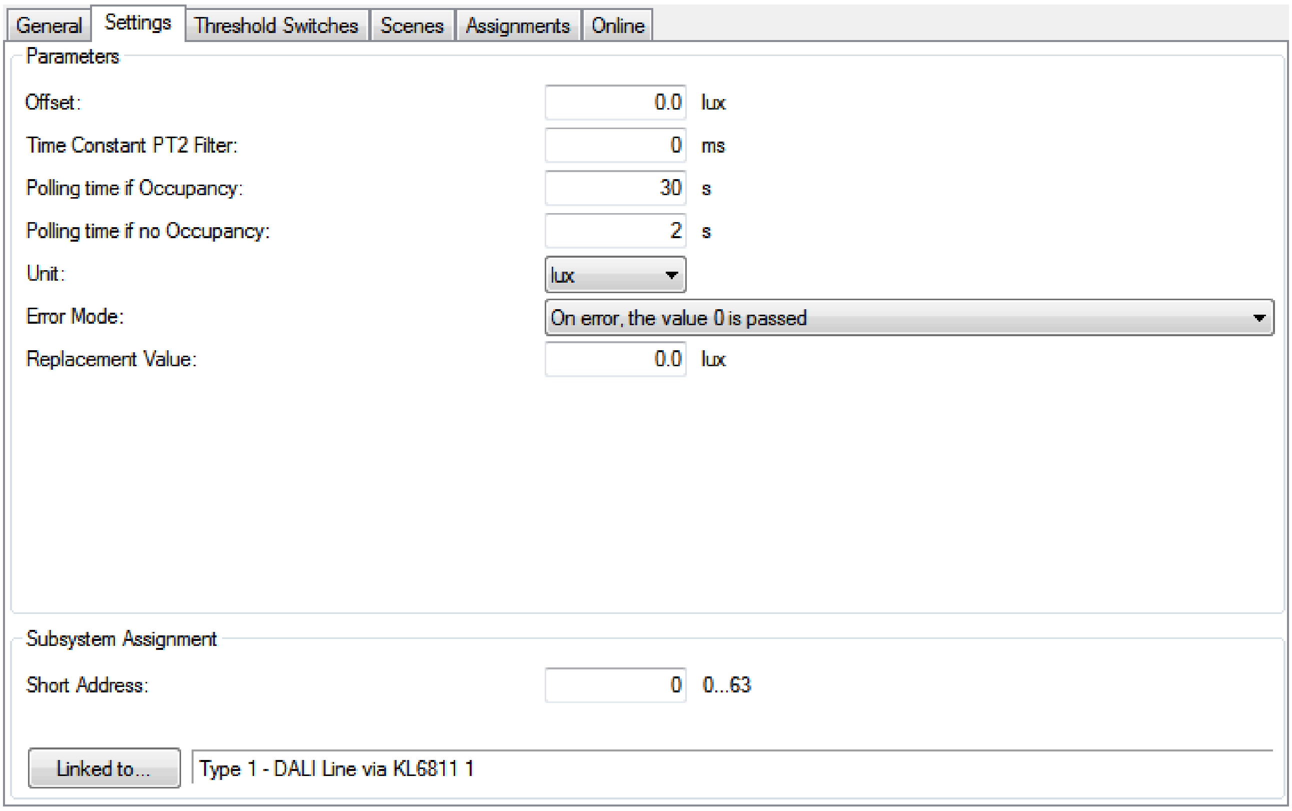
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.3.2.3 DALI smartSPOT Analog Signal

The analog signal DALI smartSPOT evaluates the brightness of an DALI smartSPOT.

Settings



Parameters

Offset: 0.0 lux

Time Constant PT2 Filter: 0 ms

Polling time if Occupancy: 30 s

Polling time if no Occupancy: 2 s

Unit: lux

Error Mode: On error, the value 0 is passed

Replacement Value: 0.0 lux

Subsystem Assignment

Short Address: 0 0...63

Type 1 - DALI Line via KL6811 1

Parameters

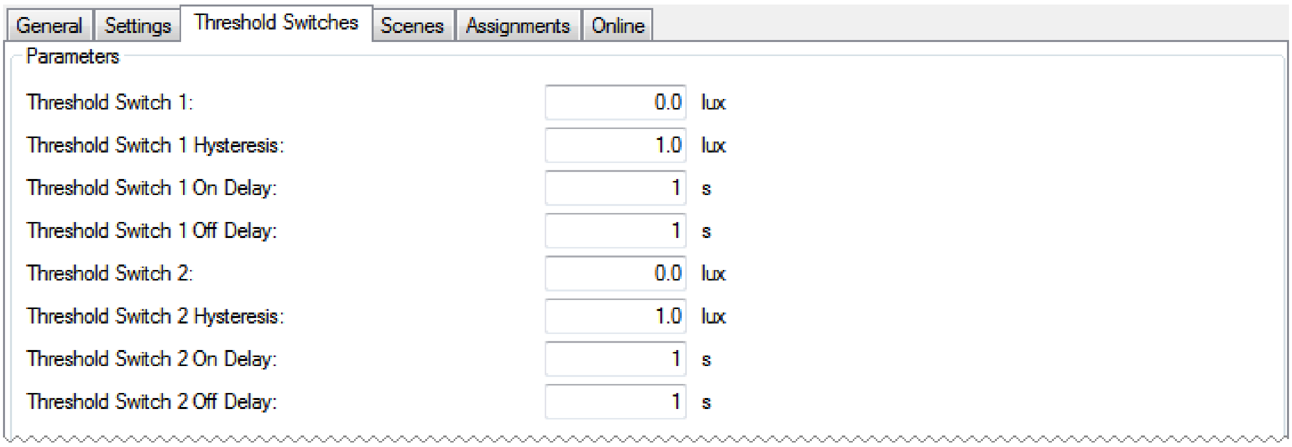
- **Offset:** This offset will be added to the received value for fine-tuning.
- **Time Constant PT2 Filter:** If necessary, the received value can be damped by a PT2-filter. If this feature is not used, the time has to be set to 0ms,
- **Polling time if Occupancy / Polling time if no Occupancy:** Like all other DALI-devices, the smartSPOT does not send its information by its own. A polling-routine is required to get the actual information.
  Since it does make sense to poll more frequently, when the observed room is not occupied, two different polling-times depending on the occupancy of the room can be entered. The occupancy is recognized by the smartSPOT itself.
- **Unit:** The unit of the scaled value. For display-purposes in the Building Automation Framework only.
- **Error Mode:** Determine how errors are handled.
- **Replacement Value:** In case of an error, this value is used instead.

Subsystem Assignment

- **Short Address:** DALI-Short Address of the smartSPOT.

Pressing the *Linked to...*-button opens a dialog, where the designated DALI-line can be assigned to this module.

Threshold Switches

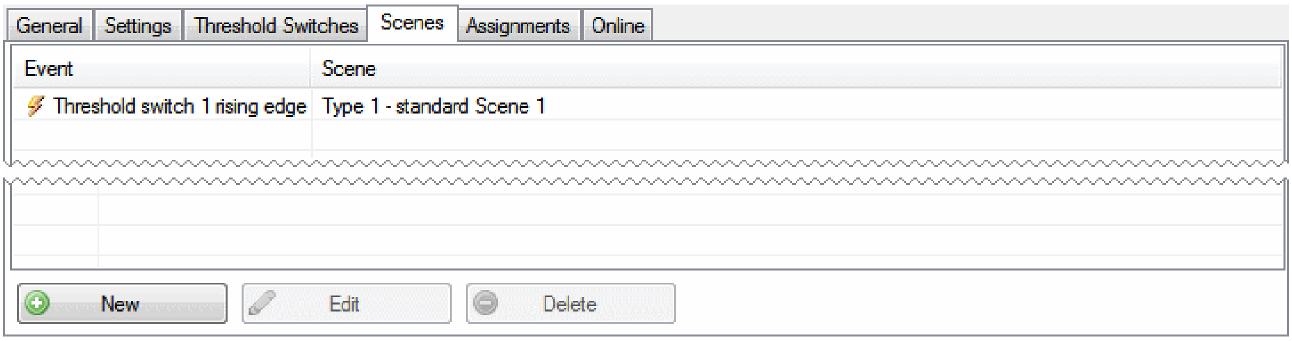


Parameters

Threshold Switch 1:	0.0	lux
Threshold Switch 1 Hysteresis:	1.0	lux
Threshold Switch 1 On Delay:	1	s
Threshold Switch 1 Off Delay:	1	s
Threshold Switch 2:	0.0	lux
Threshold Switch 2 Hysteresis:	1.0	lux
Threshold Switch 2 On Delay:	1	s
Threshold Switch 2 Off Delay:	1	s

Two threshold-switches can be defined for the scaled value. The rising and falling edges of these switches can be used to execute scenes (see *Scenes*-menu below).
Additionally a hysteresis and delay can be defined to dampen the behaviour of the scaled value.

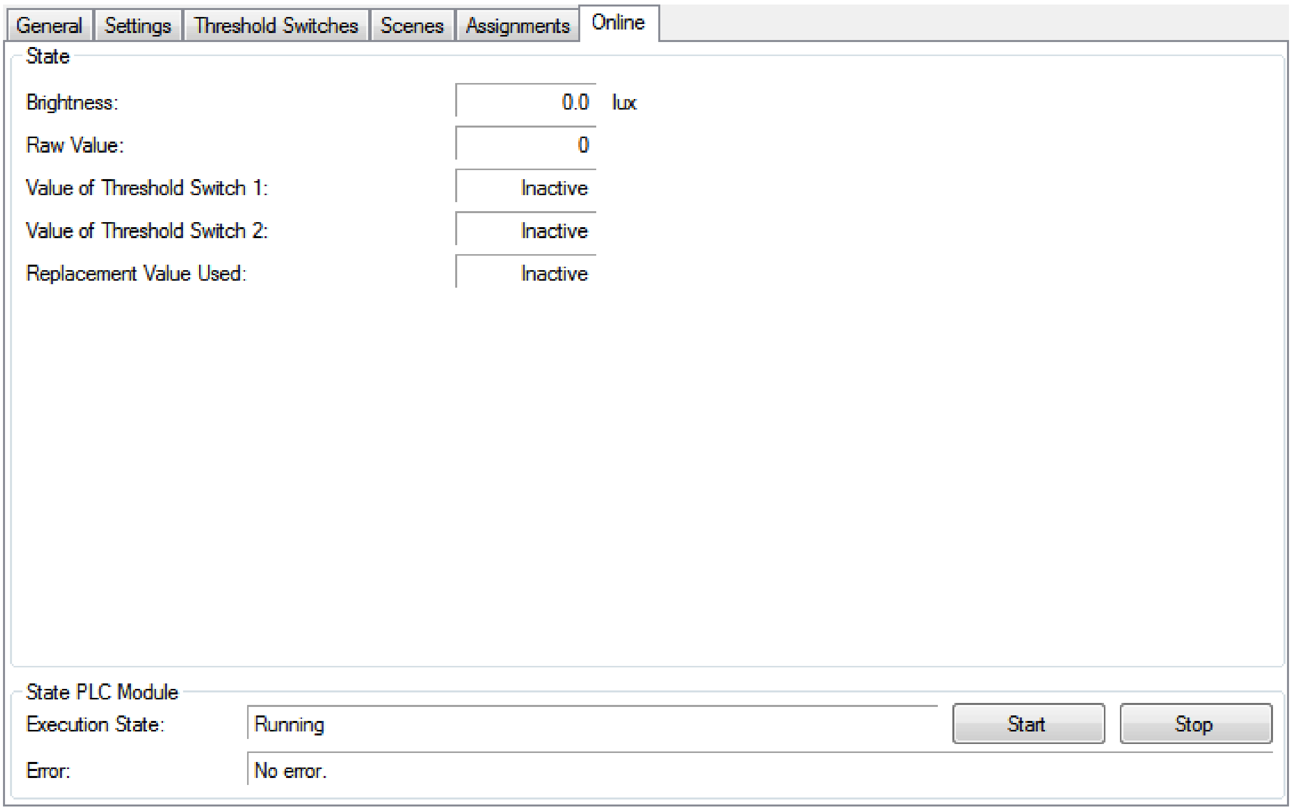
Scenes



Event	Scene
Threshold switch 1 rising edge	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the rising and the falling edge events of the threshold-switches can be assigned to the execution of a scene. The scene however has to be defined previously.

Online



State

Brightness: 0.0 lux

Raw Value: 0

Value of Threshold Switch 1: Inactive

Value of Threshold Switch 2: Inactive

Replacement Value Used: Inactive

State PLC Module

Execution State: Running

Error: No error.

State

- **Brightness:** The latest polled and calculated brightness value of the smartSPOT.
- **Raw Value:** The latest polled raw value of the smartSPOT.
- **Value of Threshold Switch 1/2:** Active if the scaled value exceeds the threshold, otherwise inactive.
- **Replacement Value Used:** Indicates whether the *Replacement Value* is used.

State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4 Actuators

7.2.2.4.1 Lamps

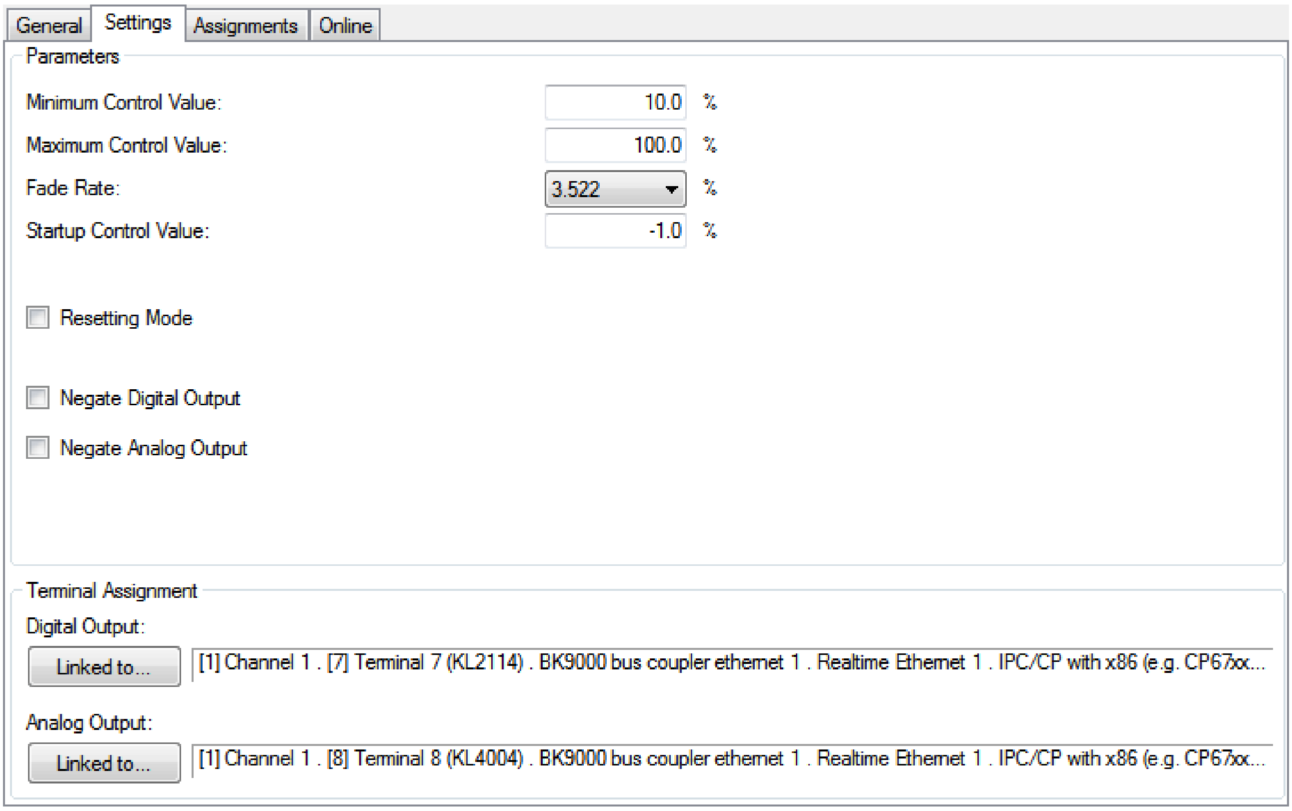
The following types of Lamps are available:

- Standard Lamp [▶ 79]
- DALI Lamp [▶ 82]
- DALI Lamp for Emergency Lighting [▶ 84]

7.2.2.4.1.1 Standard Lamp

This modul is used to control lamps over a digital and/or analog ouput. In order to build groups with lamps of other sorts, e.g. DALI, the original range of 0..32767 for an analog lamp at a dimmer-terminal is displayed in percentage. For digital lamps a value of 0% means that the lamp is off and any other value on. The values however are taken over for further group-calculation, if needed.

Settings



Minimum Control Value: 10.0 %

Maximum Control Value: 100.0 %

Fade Rate: 3.522 %

Startup Control Value: -1.0 %

□ Resetting Mode

□ Negate Digital Output

□ Negate Analog Output

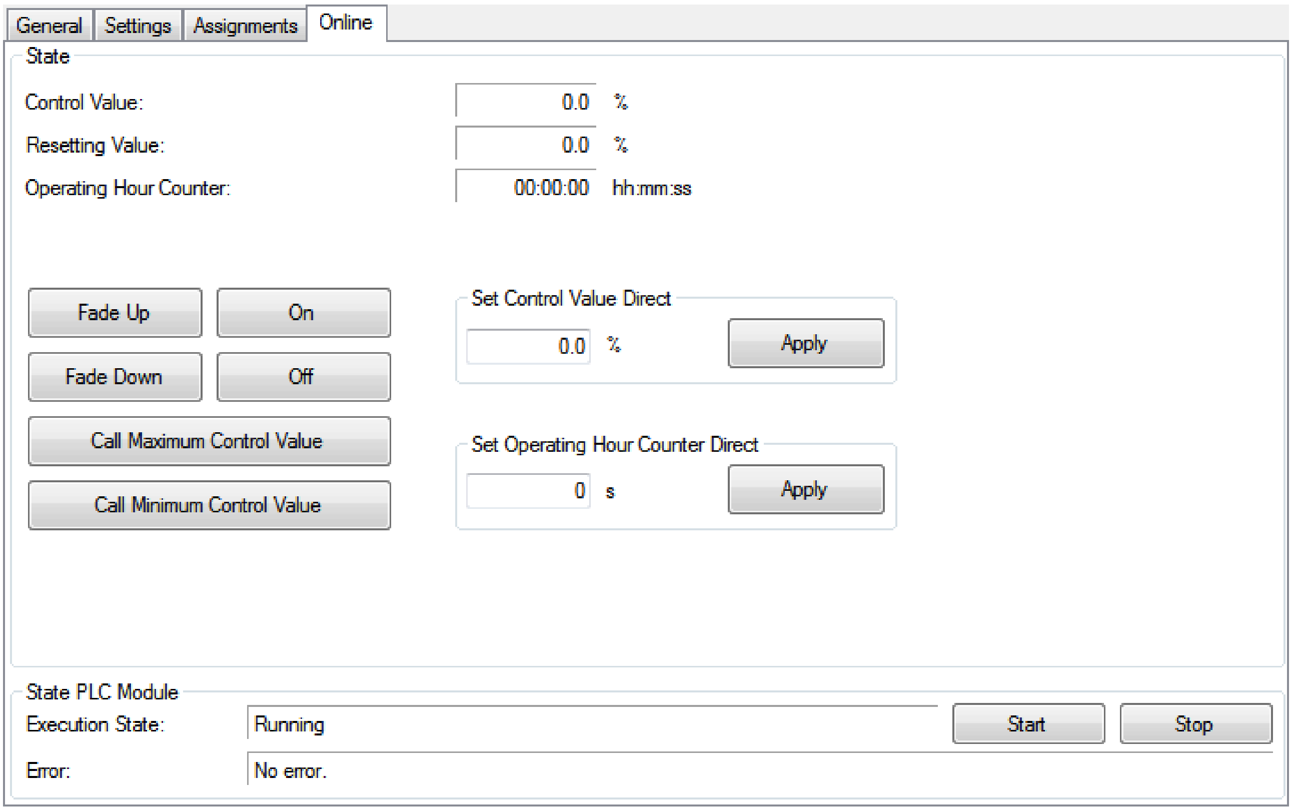
Digital Output: [1] Channel 1 . [7] Terminal 7 (KL2114) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Analog Output: [1] Channel 1 . [8] Terminal 8 (KL4004) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Parameters

- **Minimum Control Value:** The minimum value, to which the lamp can be dimmed, if analog.
- **Maximum Control Value:** The maximum value, to which the lamp can be dimmed.
- **Fade Rate:** The amount by which the lamps value is increased / decreased with a *Fade Up* / *Fade Down*-command.
- **Startup Control Value:** Initial value of the lamp, when the PLC-application is started. If the lamp is explicitly set to a value lower than its minimum or greater than its maximum value, it will be automatically set to the appropriate limit. The exception is 0%, which turns the lamp off.
- **Resetting Mode:** When selected, the last value, before turning the lamp off will be stored. If the lamp is turned on again by the *On*-command, it will be set back to this value.
- **Negate Digital Output / Negate Analog Output:** 0% means completely on and 100% completely off. The limiting rules mentioned above however, are still the same: values lower than the *Minimum Control Value* will be set to the minimum and values greater than the *Maximum Control Value* to the maximum. 0% is allowed but in this case the lamp will be turned on.

Online



State

Control Value: 0.0 %

Resetting Value: 0.0 %

Operating Hour Counter: 00:00:00 hh:mm:ss

Set Control Value Direct: 0.0 %

Set Operating Hour Counter Direct: 0 s

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light value by the entered fade-rate. The command will only work when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the lamp on either to the maximum or the *Resetting Value*, which depends on the fact, if the *Resetting Mode* is activated or not.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off-state will be stored.
- **Call Maximum Control Value:** The lamp will be turned to its maximum level.
- **Call Minimum Control Value:** The lamp will be turned to its minimum level.
- **Set Control Value Direct:** The entered value will be taken over, when the *Apply*-button is pressed. If the lamp is explicitly set to a value lower than its minimum or greater than its maximum value, it will be automatically set to the appropriate limit. The exception is 0%, which turns the lamp off.
- **Set Operating Hour Counter Direct:** The entered value will be taken over, when the *Apply*-button is pressed.

State

- **Control Value:** Actual output value.
- **Resetting Value:** When *Resetting Mode* is selected and the lamp is turned off, the last value, before turning off will be stored. If the lamp is turned on again by the *On*-command, it will be set back to this value.
- **Operating Hour Counter:** Service life of the lamp.

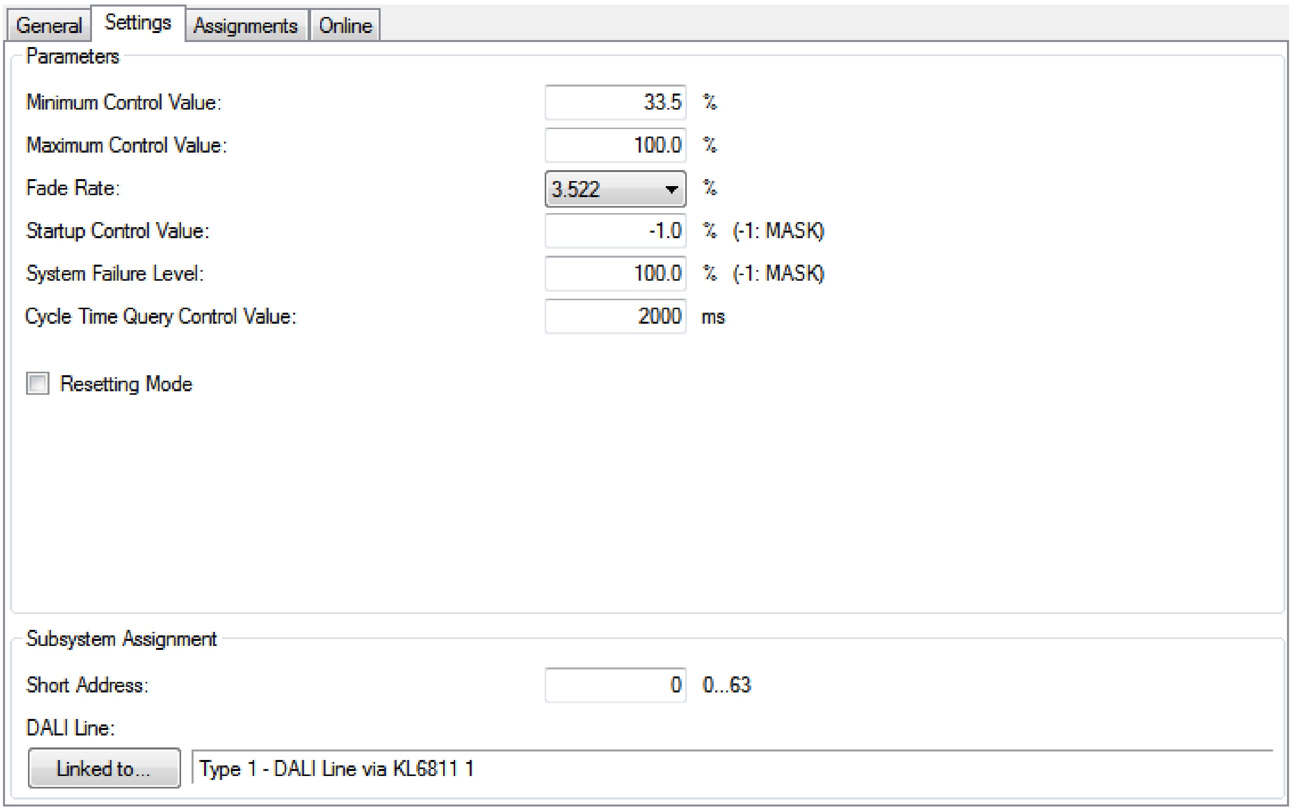
State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.1.2 DALI Lamp

This module is used to control DALI ballasts. In order to build groups with standard lamps, the original range of 0..254 for an DALI-lamp displayed in percentage. The calculation is linear: 0% means 0 (DALI) , 50% means 127 (DALI) and 100% means 254 (DALI).

Settings



Parameters

Minimum Control Value: 33.5 %

Maximum Control Value: 100.0 %

Fade Rate: 3.522 %

Startup Control Value: -1.0 % (-1: MASK)

System Failure Level: 100.0 % (-1: MASK)

Cycle Time Query Control Value: 2000 ms

□ Resetting Mode

Subsystem Assignment

Short Address: 0 0...63

DALI Line:

Linked to... Type 1 - DALI Line via KL6811 1

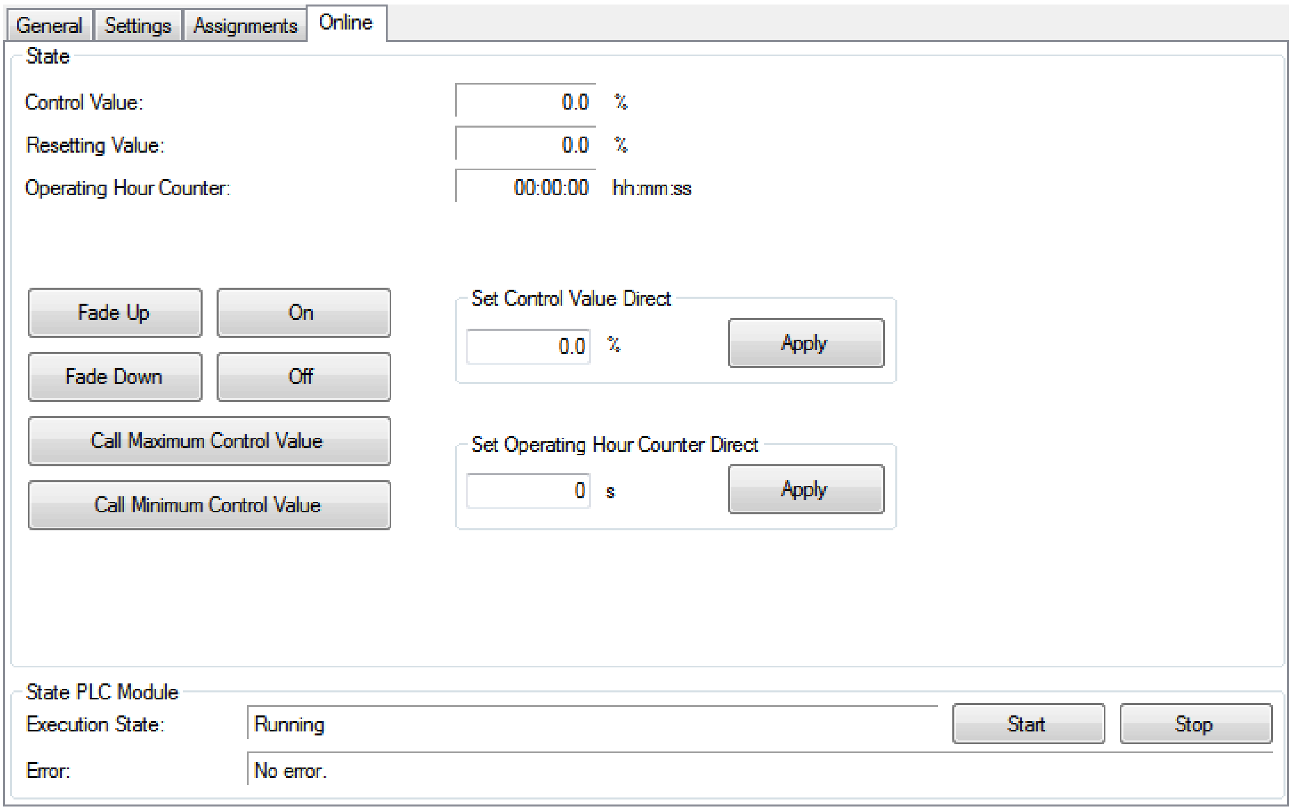
Parameters

- **Minimum Control Value:** The minimum value, to which the lamp can be dimmed.
- **Maximum Control Value:** The maximum value, to which the lamp can be dimmed. Changing these values does not affect the ballast internal min and max values, which will always prevail.
- **Fade Rate:** The amount by which the lamps value is increased / decreased with a *Fade Up* / *Fade Down*-command.
- **Startup Control Value:** Initial value of the lamp, when the PLC-application is started. If the lamp is explicitly set to a value lower than its minimum or greater than its maximum value, it will be automatically set to the appropriate limit. The exception is 0%, which turns the lamp off.
- **System Failure Level:** The system failure level of the DALI-lamp. This is the level of the lamp in case of a disconnection from the DALI-terminal.
- **Cycle Time Query Control Value:** Since not all DALI-commands change the ballasts to an absolute value, it is necessary to query the actual value from time to time. This entry defines the interval.
- **Resetting Mode:** When selected, the last value, before turning the lamp off will be stored. If the lamp is turned on again by the *On*-command, it will be set back to this value.

Subsystem Assignment

- **Short Address:** The short address of the assigned DALI-device.
- **Linked to...:** The DALI-line, at which the ballast is installed, has to be entered here. Once the button is activated, a window will open, where the DALI-terminal can be chosen.

Online



General | Settings | Assignments | Online

State

Control Value: 0.0 %

Resetting Value: 0.0 %

Operating Hour Counter: 00:00:00 hh:mm:ss

Set Control Value Direct: 0.0 %

Set Operating Hour Counter Direct: 0 s

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light value by the entered fade-rate. The command will only work, when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the lamp either to the maximum or the *Resetting Value*, which depends on the fact, if the *Resetting Mode* is activated or not.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off-state will be stored.
- **Call Maximum Control Value:** The lamp will be turned to its maximum level.
- **Call Minimum Control Value:** The lamp will be turned to its minimum level.
- **Set Control Value Direct:** The entered value will be taken over, when the *Apply*-button is pressed. If the lamp is explicitly set to a value lower than its minimum or greater than its maximum value, it will be automatically set to the appropriate limit. The exception is 0%, which turns the lamp off.
- **Set Operating Hour Counter Direct:** The entered value will be taken over, when the *Apply*-button is pressed.

State

- **Control Value:** Actual output value. If the lamp is set to an new value, it will be shown in the *Control Value*-field first, even if the lamp does not have reached this level, yet, due to a high device internal fade-time. The lamp may slowly dim to its end-level and a short *Cycle Time Query Control Value* will make the dimming visible. In the end, the lamp will be set closest possible percentage value.
- **Resetting Value:** When *Resetting Mode* is selected and the lamp is turned off, the last value, before turning off will be stored. If the lamp is turned on again by the *On*-command, it will be set back to this value.
- **Operating Hour Counter:** Service life of the lamp.

State PLC Module

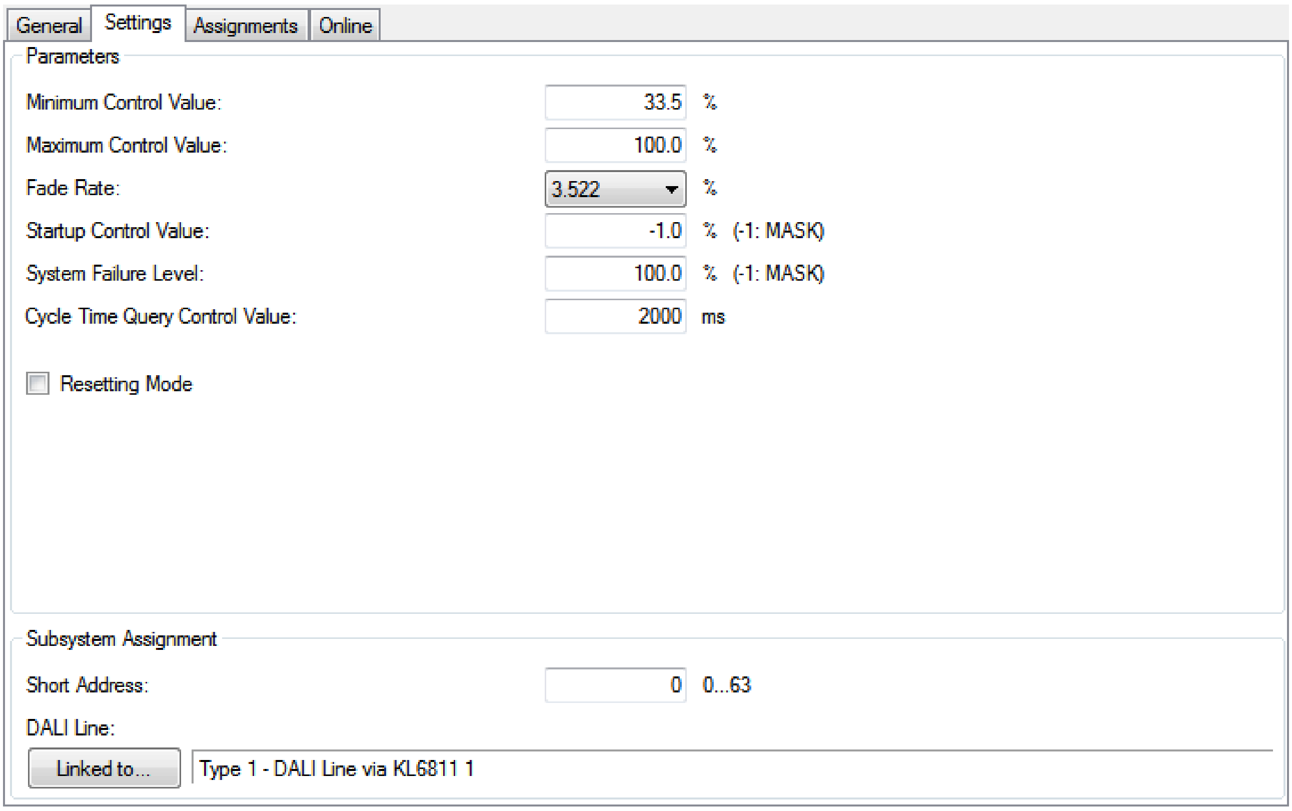
- **Execution State:** Shows if the module is running or stopped.

- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.1.3 DALI Lamp for Emergency Lighting

This modul is used to control DALI ballasts for emergency lighting. In order to build groups with standard lamps, the original range of 0..254 for an DALI-lamp displayed in percentage. The calculation is linear: 0% means 0 (DALI), 50% means 127 (DALI) and 100% means 254 (DALI).

Settings



Minimum Control Value: 33.5 %
Maximum Control Value: 100.0 %
Fade Rate: 3.522 %
Startup Control Value: -1.0 % (-1: MASK)
System Failure Level: 100.0 % (-1: MASK)
Cycle Time Query Control Value: 2000 ms

□ Resetting Mode

Short Address: 0 0...63
DALI Line:
Type 1 - DALI Line via KL6811 1

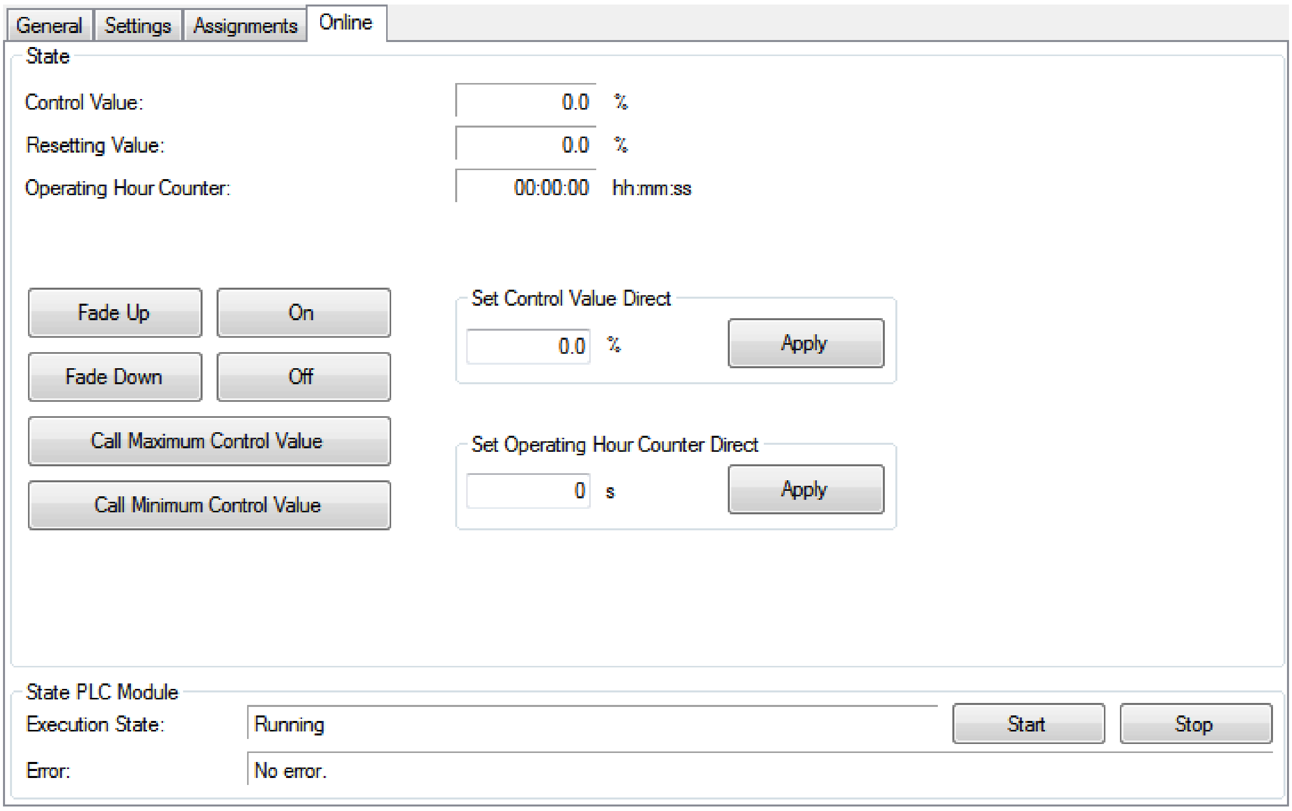
Parameters

- **Minimum Control Value:** The minimum value, to which the lamp can be dimmed.
- **Maximum Control Value:** The maximum value, to which the lamp can be dimmed. Changing these values does not affect the ballast internal min and max values, which will always prevail.
- **Fade Rate:** The amount by which the lamps value is increased / decreased with a *Fade Up* / *Fade Down*-command.
- **Startup Control Value:** Initial value of the lamp when the PLC-application is started. If the lamp is explicitly set to a value lower than its minimum or greater than its maximum value, it will be automatically set to the appropriate limit. The exception is 0%, which turns the lamp off.
- **System Failure Level:** The system failure level of the DALI-lamp. This is the level of the lamp in case of a disconnection from the DALI-terminal.
- **Cycle Time Query Control Value:** Since not all DALI-commands change the ballasts to an absolute value, it is necessary to query the actual value from time to time. This entry defines the interval.
- **Resetting Mode:** When selected, the last value, before turning the lamp off will be stored. If the lamp is turned on again by the *On*-command, it will be set back to this value.

Subsystem Assignment

- **Short Address:** The short address of the assigned DALI-device.
- **Linked to...:** The DALI-line, at which the ballast is installed, must be entered here. Once the button is activated, a window will open, where the DALI-terminal can be chosen.

Online



State

Control Value: 0.0 %

Resetting Value: 0.0 %

Operating Hour Counter: 00:00:00 hh:mm:ss

Set Control Value Direct: 0.0 %

Set Operating Hour Counter Direct: 0 s

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light-value by the entered fade-rate. The command will only work when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the lamp either to the maximum or the *Resetting Value*, which depends on the fact, if the *Resetting Mode* is activated or not.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off-state will be stored.
- **Call Maximum Control Value:** The lamp will be turned to its maximum level.
- **Call Minimum Control Value:** The lamp will be turned to its minimum level.
- **Set Control Value Direct:** The entered value will be taken over, when the *Apply*-button is pressed. If the lamp is explicitly set to a value lower than its minimum or greater than its maximum value, it will be automatically set to the appropriate limit. The exception is 0%, which turns the lamp off.
- **Set Operating Hour Counter Direct:** The entered value will be taken over, when the *Apply*-button is pressed.

State

- **Control Value:** Actual output value. If the lamp is set to a new value, it will be shown in the *Control Value*-field first, even if the lamp does not have reached this level, yet, due to a high device internal fade-time. The lamp may slowly dim to its end-level and a short *Cycle Time Query Control Value* will make the dimming visible. In the end, the lamp will be set closest possible percentage-value.
- **Resetting Value:** When *Resetting Mode* is selected and the lamp is turned off, the last value, before turning off will be stored. If the lamp is turned on again by the *ON*-command, it will be set back to this value.
- **Operating Hour Counter:** Service life of the lamp.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.

- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.2 Blind Drives

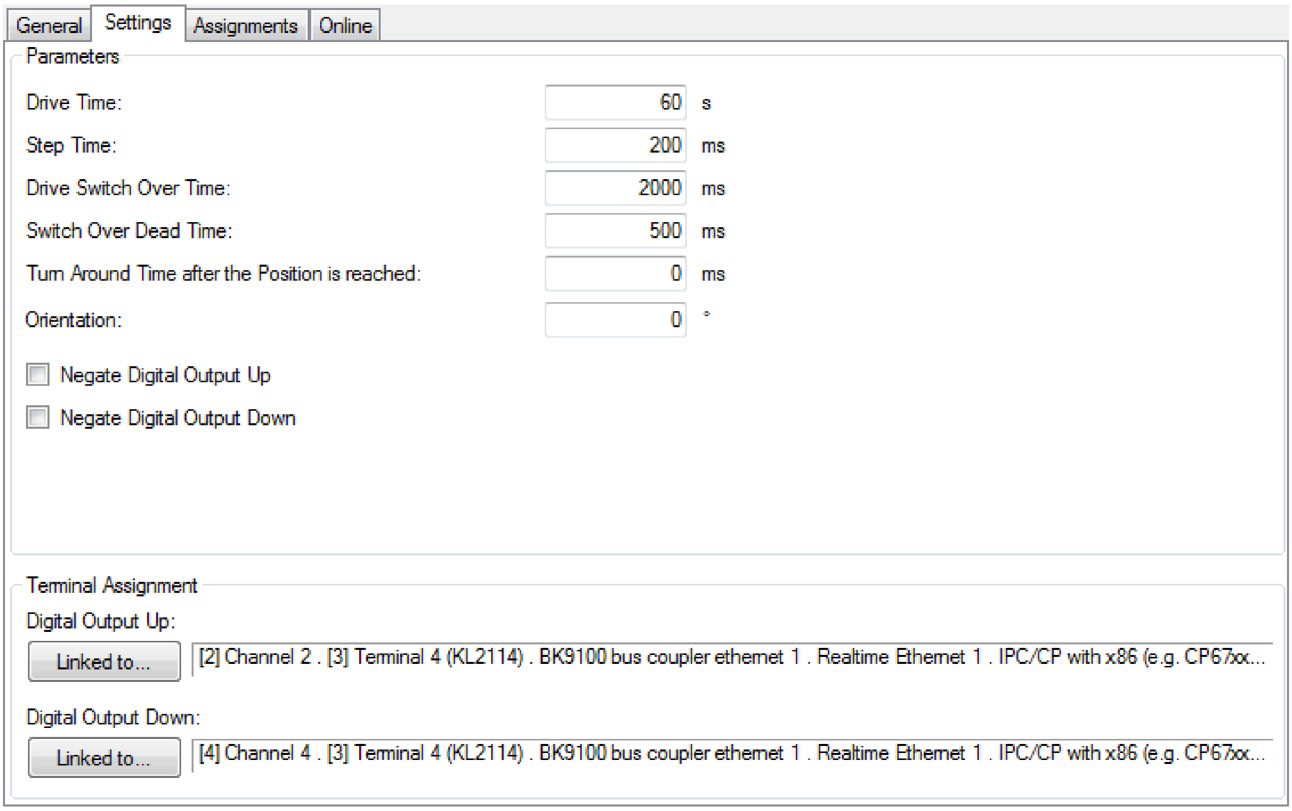
The following types of Blind Drives are available:

- Standard Blind Drive [▸ 86]
- Blind Drive connected with KL2532/KL2552 [▸ 88]

7.2.2.4.2.1 Standard Blind Drive

This module is used to control blind drives over two digital outputs.

Settings



Drive Time: 60 s

Step Time: 200 ms

Drive Switch Over Time: 2000 ms

Switch Over Dead Time: 500 ms

Turn Around Time after the Position is reached: 0 ms

Orientation: 0 °

□ Negate Digital Output Up

□ Negate Digital Output Down

Terminal Assignment

Digital Output Up:

[2] Channel 2 . [3] Terminal 4 (KL2114) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

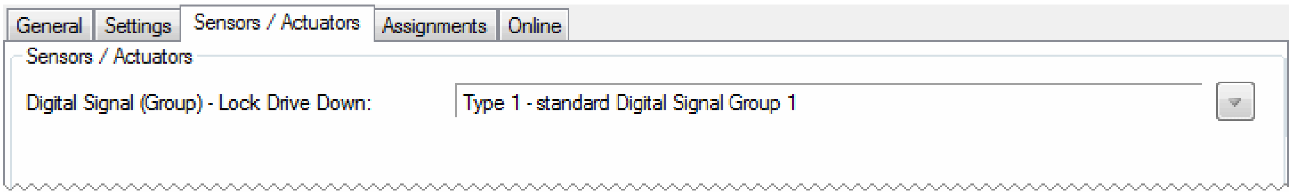
Digital Output Down:

[4] Channel 4 . [3] Terminal 4 (KL2114) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Parameters

- **Drive Time:** Total driving time of the blind.
- **Step Time:** Driving time for one step.
- **Drive Switch Over Time:** Time to turn the lamellas from one limiting position to the other (translucence from below / translucence from above).
- **Switch Over Dead Time:** Waiting time when the driving direction is changed.
- **Turn Around Time after the Position is reached:** If the blind is driven downwards to a specific position except 100% (completely closed) it may be necessary to turn the lamellas in order to prevent downward translucence (e.g., for bed or bathrooms). This time defines the turning of the lamellas.
- **Orientation:** Orientation of the blind. 0° = North, 90° = East, 180° = South, 270° = West.
- **Negate Digital Output Up/Down:** Inverts the control of the digital outputs.

Sensors / Actuators

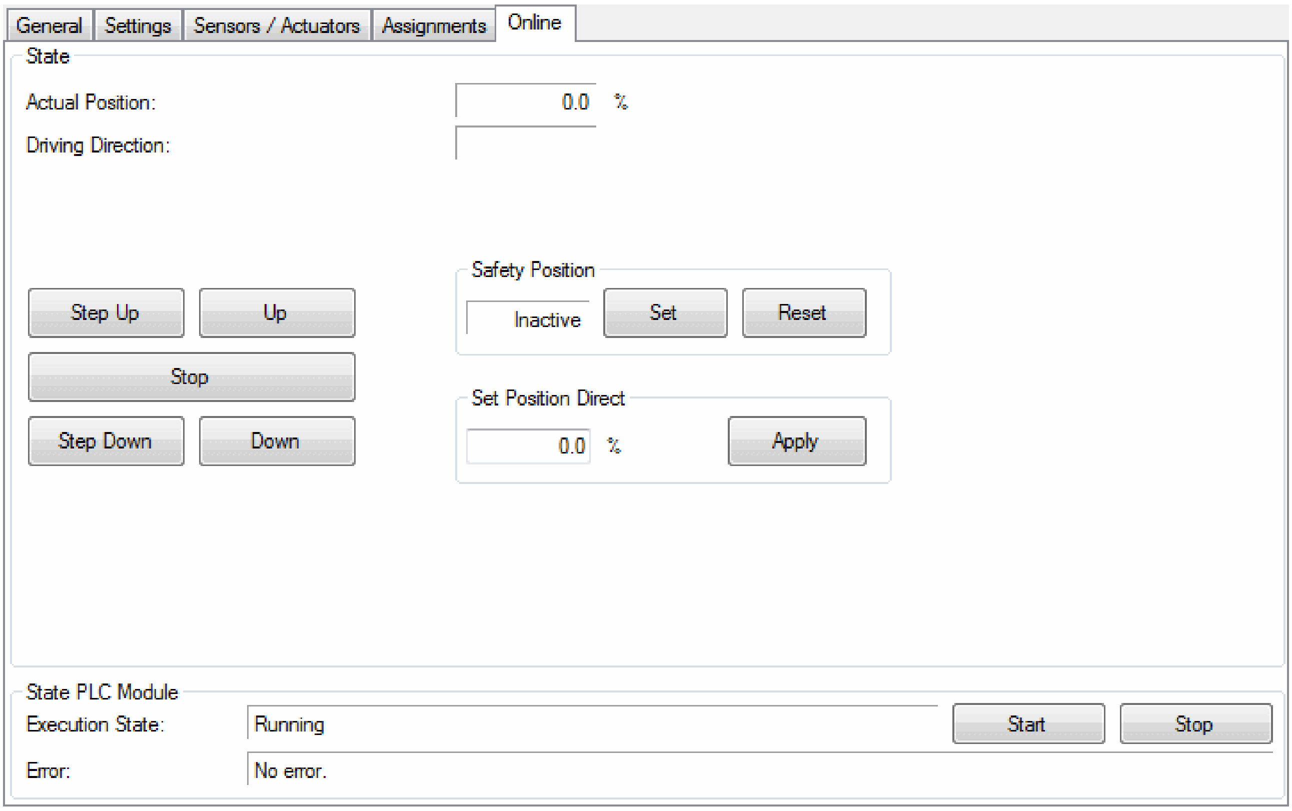


Sensors / Actuators

Digital Signal (Group) - Lock Drive Down: Type 1 - standard Digital Signal Group 1

In this menu you can enter a digital signal group (e.g. window contacts) that inhibits a blind moving down. In this way, automatic blind moving down in the case of open patio doors can be prevented.

Online



State

Actual Position: 0.0 %

Driving Direction:

Safety Position: Inactive

Set Position Direct: 0.0 %

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Step Up/Step Down:** Drives the blind for one step up/down. See parameter *Step Time*.
- **Up/Down:** Drives the blind up/down.
- **Stop:** Stops the blind.
- **Safety Position Set:** Driving into safety-position. For this purpose the blind drive will be moved up. No other commands except *Safety Position Reset* are accepted then.
- **Safety Position Reset:** Releases the lock of the safety-position.
- **Set Position Direct:** The blind drive will be driven to a specific position. 100% means completely closed, 0% completely opened.

State

- **Actual Position:** Actual position in %. 100% means completely closed, 0% completely opened.
- **Driving Direction:** The driving direction will show here, if the blind moves.

State PLC Module

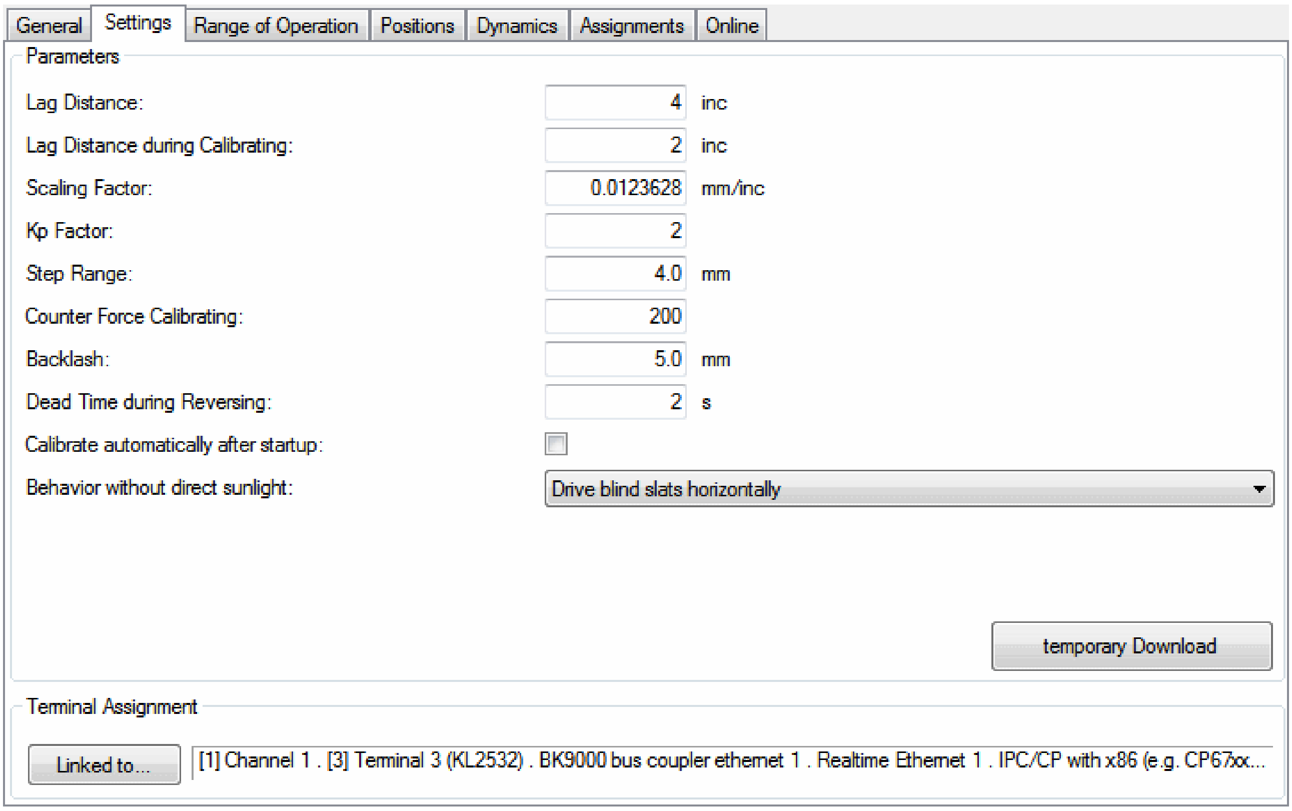
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.2.2 Blind Drive connected with KL2532

The Blind Drive controls a blind drive connected to a KL2532.

Settings

Settings



Lag Distance:	4	inc
Lag Distance during Calibrating:	2	inc
Scaling Factor:	0.0123628	mm/inc
Kp Factor:	2	
Step Range:	4.0	mm
Counter Force Calibrating:	200	
Backlash:	5.0	mm
Dead Time during Reversing:	2	s
Calibrate automatically after startup:	☐	
Behavior without direct sunlight:	Drive blind slats horizontally	

[1] Channel 1 . [3] Terminal 3 (KL2532) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Commands

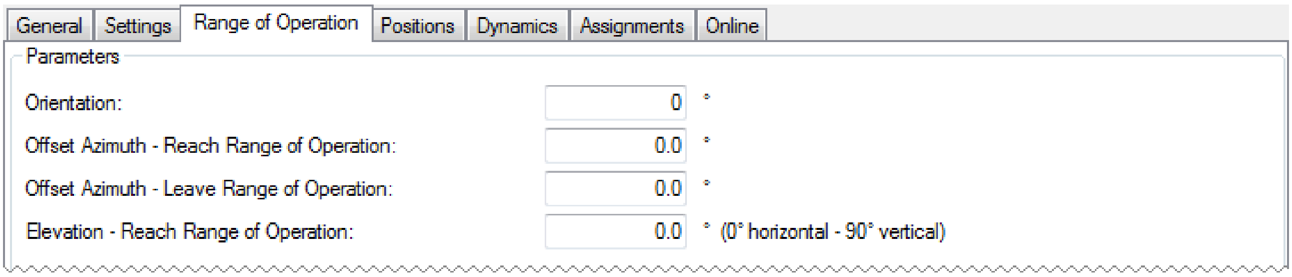
- **temporary Download:** If you want to change parameters to test the blind, then a normal download of parameters (activate configuration) would take a relative long time. To make work easier and save time, therefore exists the possibility of a temporary download. Here, it is not necessary to stop the PLC modules to download the parameters. The parameters of the blind will remain temporary and not persistent. These changes will discard after a PLC restart or activating the configuration.

Parameters

- **Lag Distance:** Maximum allowed control deviation (calculated setpoint value of the controller ↔ actual value of the encoder/resolver). The motor will be stopped if this value is exceeded.
- **Lag Distance during Calibrating:** Ditto but during calibration [▸ 93].
- **Scaling Factor:** Ratio traverse path of the blind to accumulated increments of the encoder.
- **Kp Factor:** Acceleration ramp of the motor.
- Step Range**:** Distance, the blind covers when driven by one step-command.
- Counter Force Calibrating: Despite using an encoder, small inaccuracies, depending on the weight of the blind, its largeness and the driving velocity, may accumulate. Thus it is necessary to calibrate [▸ 93] the blind from time to time. When the *Counter Force Calibrating* is exceeded by the number of upward-drives, the blind will be automatically re-calibrated [▸ 93].
- **Backlash:**The position-error, which comes up, when the driving-direction is changed.
- **Dead Time during Reversing:** Pause time when the direction is changed during reversing-mode.
- **Calibrate automatically after startup:** In case of power failure the actual position of the blind gets lost and the drive has to be calibrated [▸ 93] again. Activating this checkbox lets the PLC re-calibrated [▸ 93] the drive automatically after start-up. Otherwise, the blind will be calibrated [▸ 93], when it receives its first driving-command.

- **Behaviour without direct sunlight:** This defines the behavior of the blind, when it is not exposed to the sunlight due to its orientation (North, East, South or West).
  - Drive blind slats horizontally
  - Drive blind up
  - No change

Range of Operation



Parameters

Orientation: 0 °

Offset Azimuth - Reach Range of Operation: 0.0 °

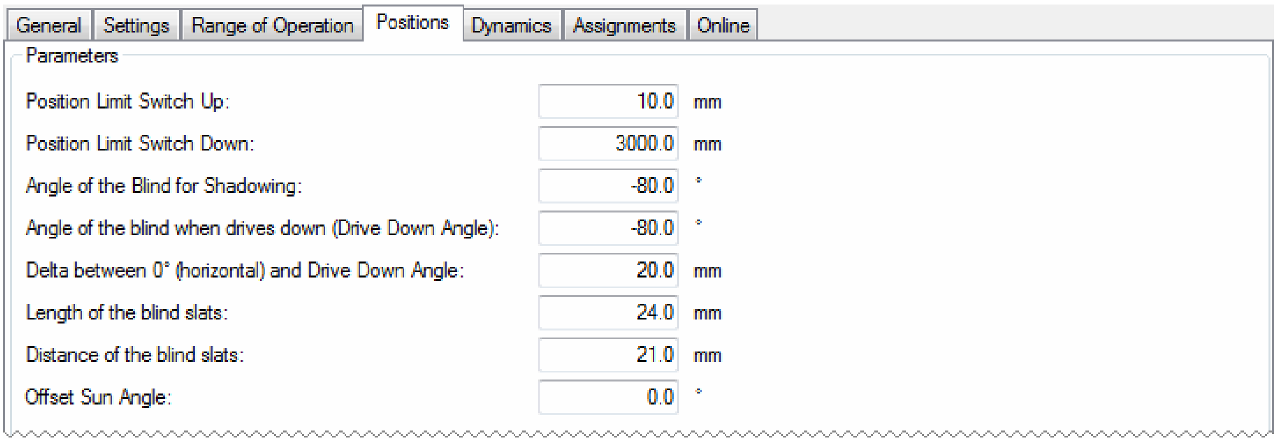
Offset Azimuth - Leave Range of Operation: 0.0 °

Elevation - Reach Range of Operation: 0.0 ° (0° horizontal - 90° vertical)

Parameters

- **Orientation:** Orientation of the blind (0°: North, 90°: East, 180°: South, 270°: West 0), see Range of Operation [▶ 92])
- **Offset Azimuth - Reach Range of Operation:** See Range of Operation [▶ 92]
- **Offset Azimuth - Leave Range of Operation:** See Range of Operation [▶ 92]
- **Elevation - Reach Range of Operation:** See Range of Operation [▶ 92]

Positions



Parameters

Position Limit Switch Up: 10.0 mm

Position Limit Switch Down: 3000.0 mm

Angle of the Blind for Shadowing: -80.0 °

Angle of the blind when drives down (Drive Down Angle): -80.0 °

Delta between 0° (horizontal) and Drive Down Angle: 20.0 mm

Length of the blind slats: 24.0 mm

Distance of the blind slats: 21.0 mm

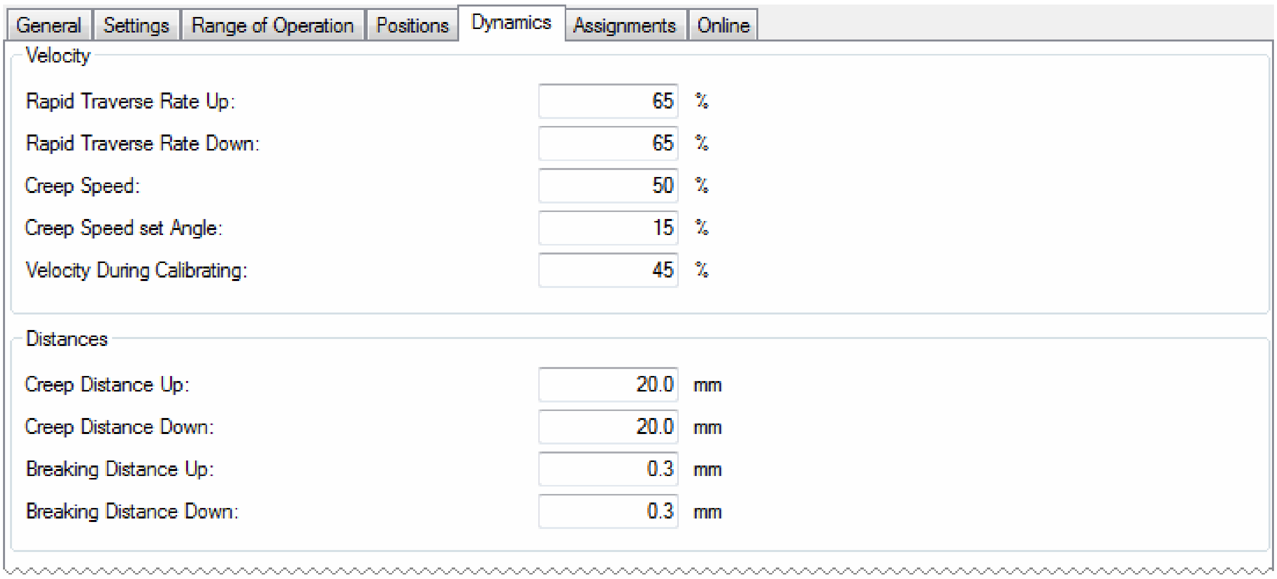
Offset Sun Angle: 0.0 °

Parameters

- **Position Limit Switch Up:** Position of the upper limit switch in mm.
- **Position Limit Switch Down:** Position of the lower limit switch in mm.
- **Angle of the Blind for Shadowing:** Angle the blind must move to, when the blind is driving down (favorite position).
- **Angle of the Blind when driving down (Drive Down Angle):** Angle of the lamellas when the blind is completely shut.
- **Difference between 0° (horizontal) and Drive Down Angle:** The traverse path between the bottom position (about 80°) and the horizontal position (about 0°). This distance is required, to perform the angle calculation.
- **Length of the blind slats:** Width of the lamellas (used to calculate the Cut off [▶ 94] position).
- **Distance of the blind slats:** Distance between the lamellas (used to calculate the Cut off [▶ 94] position).

- **Offset Sun Angle:** This offset will be added to the calculated angle of the lamellas and can be used to correct the angle-positioning of the lamellas.

Dynamics



Velocity

Rapid Traverse Rate Up:	65	%
Rapid Traverse Rate Down:	65	%
Creep Speed:	50	%
Creep Speed set Angle:	15	%
Velocity During Calibrating:	45	%

Distances

Creep Distance Up:	20.0	mm
Creep Distance Down:	20.0	mm
Breaking Distance Up:	0.3	mm
Breaking Distance Down:	0.3	mm

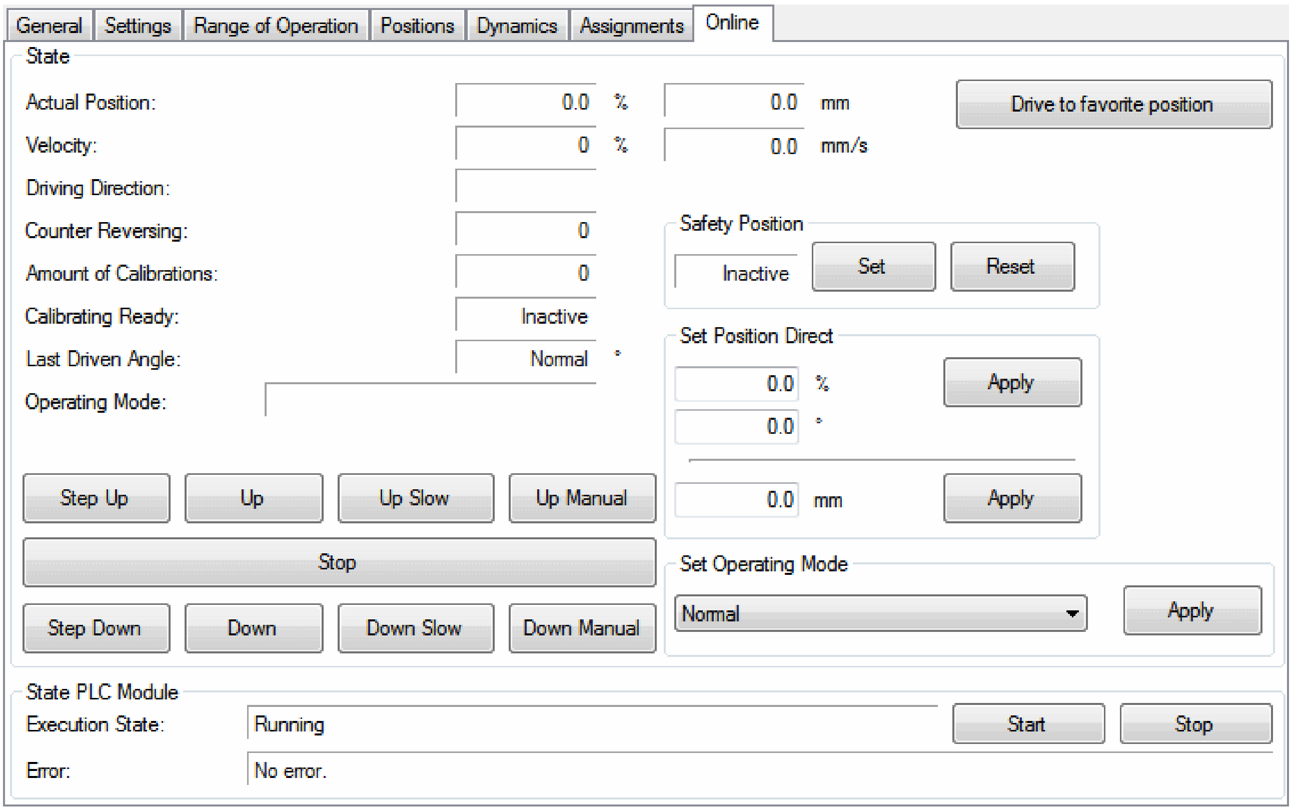
Velocity

- Rapid Traverse Rate Up**:** Rapid velocity, when driving the blind up.
- Rapid Traverse Rate Down**:** Rapid velocity, when driving the blind down.
- Creep Speed**:** Creep velocity short before the blind is driving to its position.
- **Creep Speed set Angle:** Velocity when setting a specific angle in automatic-mode.
- Velocity During Calibrating**:** Velocity during calibration [▶ 93].

Distances

- **Creep Distance Up:** Distance to destination when the blind drive is switched from rapid to creep velocity (driving the blind up).
- **Creep Distance Down:** Ditto, driving the blind down.
- **Breaking Distance Up:** Distance to destination when the blind drive is stopped. (Driving the blind up)
- **Breaking Distance Down:** Ditto, driving the blind down.

Online



Actual Position:	0.0	%	0.0	mm
Velocity:	0	%	0.0	mm/s
Driving Direction:				
Counter Reversing:	0			
Amount of Calibrations:	0			
Calibrating Ready:	Inactive			
Last Driven Angle:	Normal	°		
Operating Mode:				

Safety Position: Inactive

Set Position Direct: 0.0 % / 0.0 ° / 0.0 mm

Set Operating Mode: Normal

Execution State: Running

Error: No error.

All commands sent by groups or functional units have a higher priority. This means, that if the automatic-mode of a functional unit, for example, is activated, the blind will be controlled by this module. All other commands sent to the blind directly will be overwritten. The only exception is driving to safety-position.

Commands

- **Step Up:** Stepping the blind up - the step size can be entered in the settings section (see Step Range [▶ 88]).
- **Up:** Driving upwards with rapid velocity. The rapid velocity is to be entered in the dynamics section (see Rapid Traverse Rate Up [▶ 90]).
- **Up Slow:** Driving upwards with creep velocity. The creep velocity is to be entered in the dynamics section (see Creep Speed [▶ 90]).
- **Up Manual:** The blind is driven up if this button is pressed. Opposed to *Step Up* the blind is driven without encoder evaluation. This function is mainly for commissioning, if still it is not clear whether the blind is properly connected. When driving without encoder evaluation, the blind may take damage caused by careless use.
- **Stop:** Stops the blind.
- **Step Down:** Stepping the blind down - the step size can be entered in the settings section (see Step Range [▶ 88]).
- **Down:** Driving downwards with rapid velocity. The rapid velocity is to be entered in the dynamics section (see Rapid Traverse Rate Down [▶ 90]).
- **Down Slow:** Driving downwards with creep velocity. The creep velocity is to be entered in the dynamics section (see Creep Speed [▶ 90]).
- **Down Manual:** The blind is driven down if this button is pressed. Opposed to *Step Down* the blind is driven without encoder evaluation. This function is mainly for commissioning, if still it is not clear whether the blind is properly connected. When driving without encoder evaluation, the blind may take damage caused by careless use.
- **Safety Position Set:** The blind will be driven to its upper limit. No further commands will be accepted, except *Safety Position Reset*. If the blind is not calibrated, it will be done first.

- **Safety Position Reset:** This command resets the internal safety position latch and driving commands will be accepted again.
- **Set Position Direct:** The blind will be driven to a specific position. This position can either be given by shadowing-percentage and lamella-angle or in millimeters. The selected entry must be activated by pressing the corresponding *Apply*-button.
- Set Operating Mode: This function can be used for test-purposes
  - **Normal**: Non-test-mode, the drive can be normally accessed.
  - **Calibrating:** Starting calibration [▸ 93].
  - **Reversing:** The blind will be continuously driven between its upper and lower limit. The drives will be counted and shown in the box Counter Reversing [▸ 91].
  - **Angle for maximum Shading:** The blind will be driven to its Cut off [▸ 94] position. This however is not possible, if the sun is not within the blinds Range of Operation [▸ 92].
  - **Angle for maximum Illumination:** The lamellas will be set to an angle letting pass the maximum of the sunlight. This is not possible, if the sun is not within the blinds Range of Operation [▸ 92].
  - **Completely closed:** The blind will be completely closed.
  - **Venetian blind horizontal:** The blind will be driven do its downward limit first. Afterwards the lamellas will be set to a horizontal position - 0°.

State

- **Actual Position:** Shows the actual position in percentage and in millimeters.
- **Velocity:** Shows the actual velocity in percentage and in millimeters per second.
- **Driving Direction:** Actual driving direction.
- Counter Reversing**:** Number of drives in reversing mode.
- **Amount of Calibrations:** Counter of calibrations, which will only be incremented, when the parameter Counter Force Calibrating [▸ 88] (see Settings [▸ 88]) is greater than 0.
- **Calibrating Ready:** Indicates if the blind was calibrated [▸ 93] (**Active:** calibration is completed, **Inactive:** calibration is not completed).
- **Last Driven Angle:** Displays the driven angle.
- **Operating Mode:** Displays the chosen operation mode (see Set Operating Mode [▸ 91]).

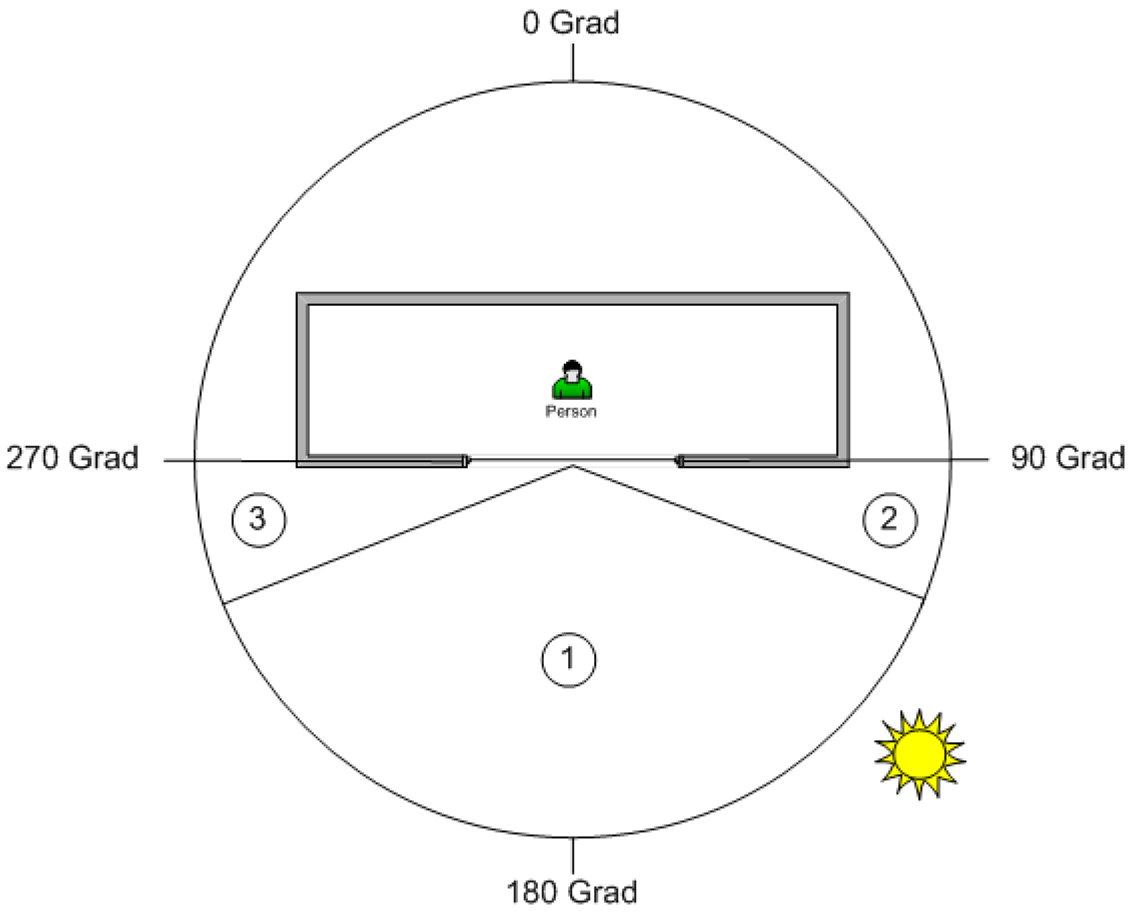
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.2.2.1 Range of Operation

The Range of Operation is the area, where the sun shines on the blind. Only when the blind is exposed to direct sunlight, an automatic positioning is needed and possible. The range of operation can be modified by certain parameters.

Azimuth:



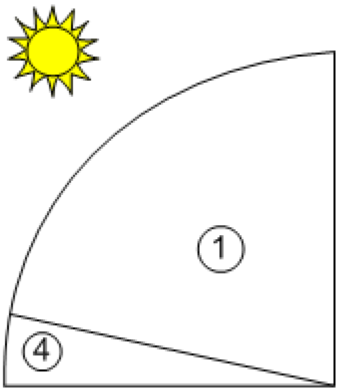
1.Range of Operation

2.Offset Azimuth - Reach Range of Operation

3.Offset Azimuth - Leave Range of Operation

The blinds orientation is South = 180°.

Elevation :



1.Range of Operation

4.Elevation - Reach Range of Operation

7.2.2.4.2.2.2 Calibration

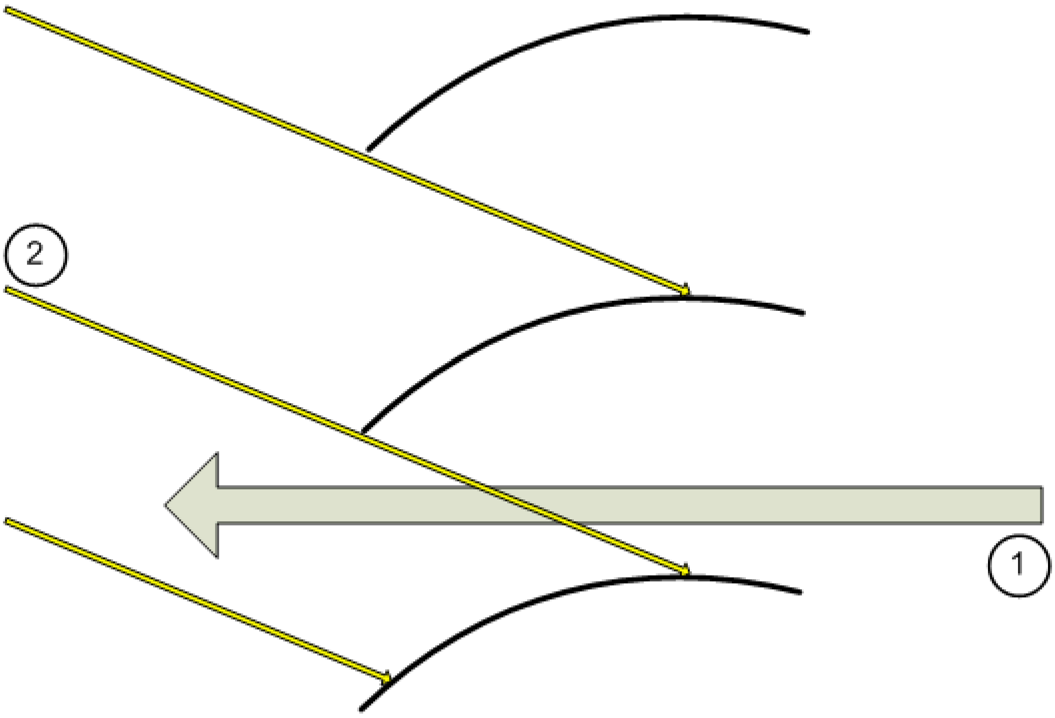
To move a blind to a specific position, the actual absolute position has to be known at any time. Calibration means, that the blind is moved to a point, where the absolute position is known - the upmost position. The blind will be driven shortly downwards and then upwards with Velocity during Calibrating [▶ 90] until the motor is stopped due to overload-detection. At this point the position is set to 0. Finally the blind is driven to

the upper limit switch.
After a restart of the PLC however, the absolute position gets lost and the blind has to be calibrated again. The calibration will be automatically executed at either the system start-up or before the first positioning command. This can be selected, see Blind Drive connected with KL2532 - Settings.

If the blind gets stuck during calibration, the drive will notice an overload and will wrongly interpret this as reaching the upmost position. In this case the calibration goes wrong.

7.2.2.4.2.2.3 Cut off

In Cut off position the sun is not directly visible, but it is still possible to see through the blind and enough daylight can pass.



1. Possibility to see through the blind.

2. Sunbeams, not able to pass.

7.2.2.4.3 Valve Actuators

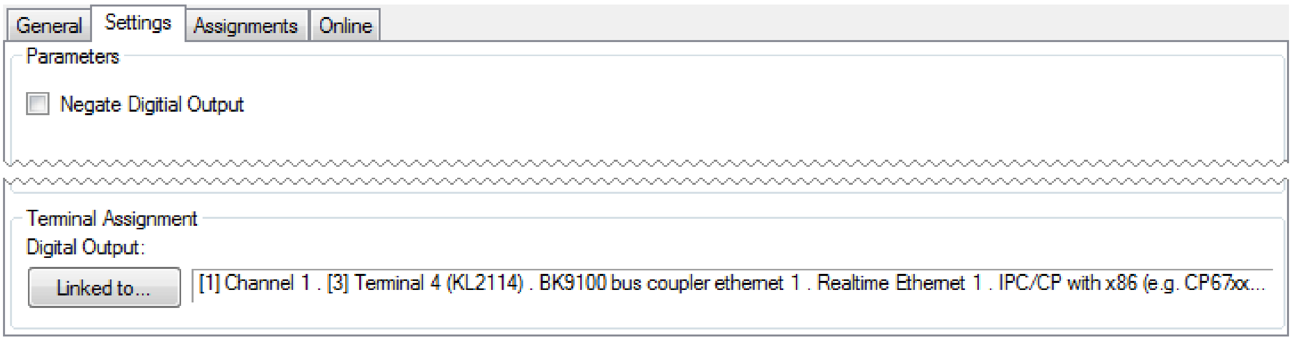
The following types of Valve Actuators are available:

- 2-point Valve Actuator [▸ 94]
- 3-point Valve Actuator [▸ 95]
- Continuous Valve Actuator [▸ 97]

7.2.2.4.3.1 2-point Valve Actuator

This module controls a 2-point Valve Actuator.

Settings



Parameters

□ Negate Digitial Output

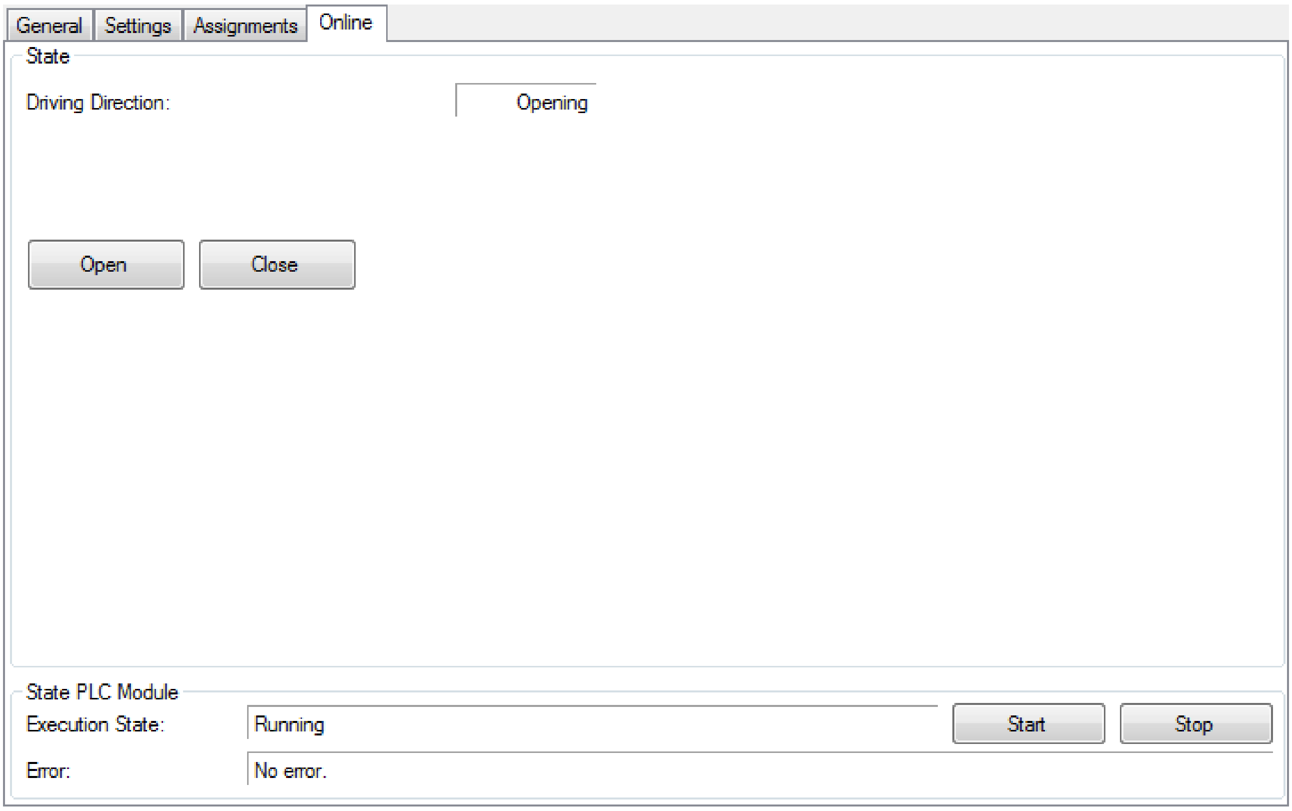
Terminal Assignment

Digital Output: [1] Channel 1 . [3] Terminal 4 (KL2114) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Parameters

- **Negate Digital Output:** Inverts the output controlling the valve.

Online



State

Driving Direction: Opening

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Open:** Opens the valve. The output will be active, if not negated in the *settings*-menu.
- **Close:** Closes the valve. The output will be inactive, if not negated in the *settings*-menu.

State

- **Driving Direction:** Shows the actual state of the valve.

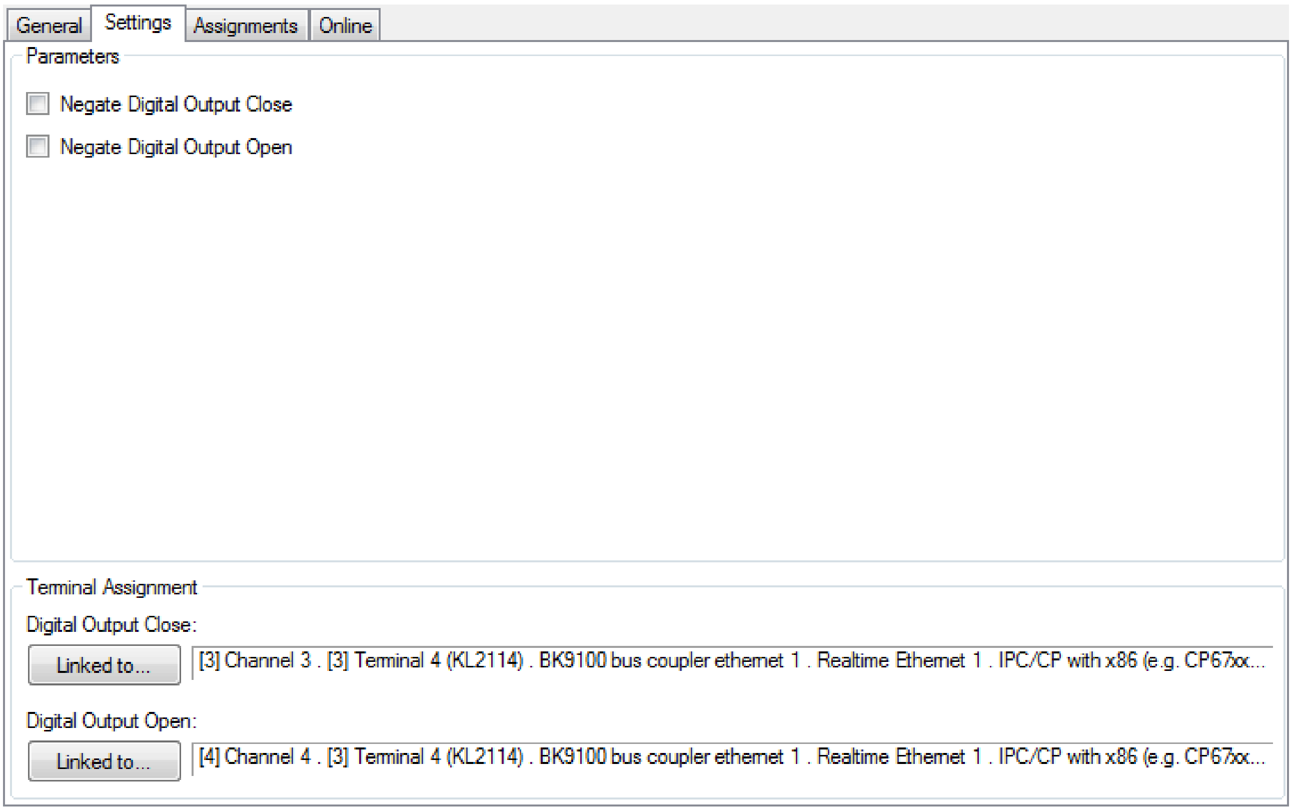
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.3.2 3-point Valve Actuator

This module controls a 3-point Valve Actuator.

Settings



Parameters

□ Negate Digital Output Close

□ Negate Digital Output Open

Terminal Assignment

Digital Output Close:

Linked to... [3] Channel 3 . [3] Terminal 4 (KL2114) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

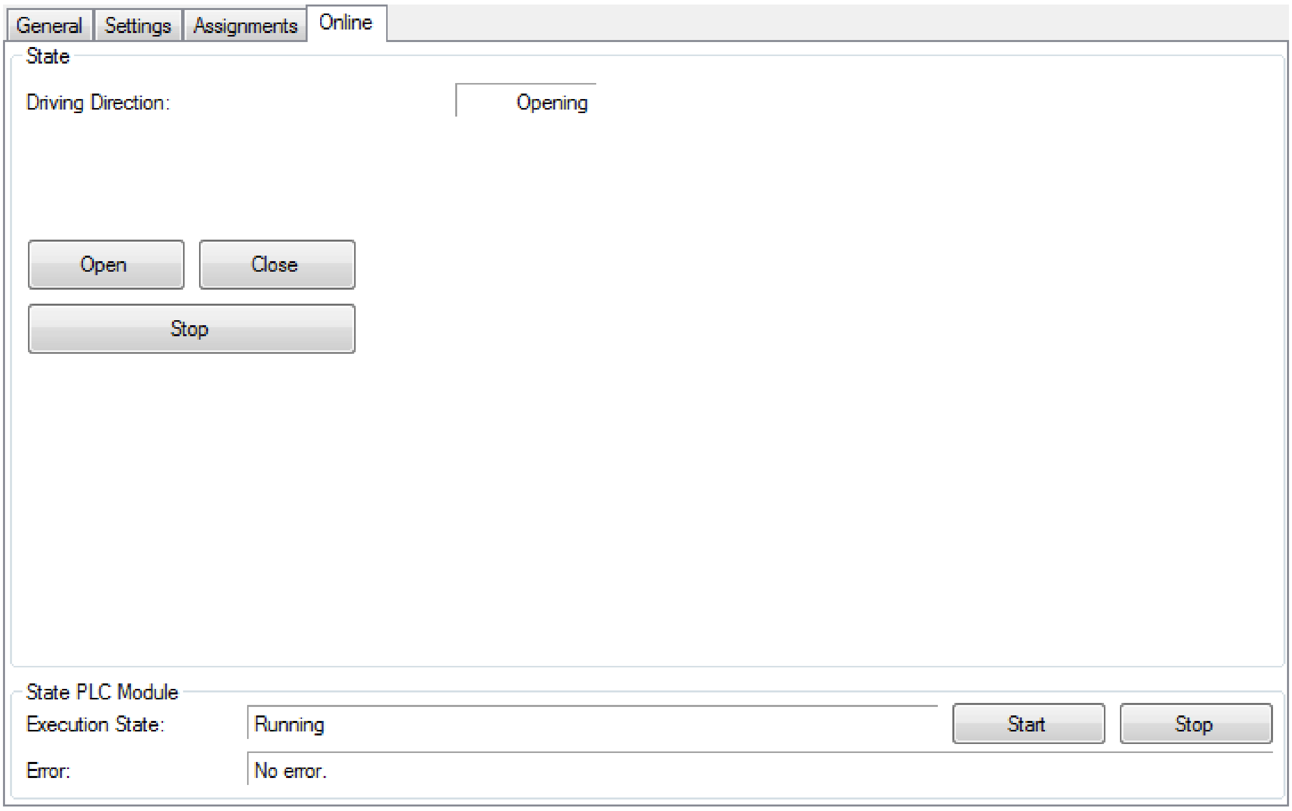
Digital Output Open:

Linked to... [4] Channel 4 . [3] Terminal 4 (KL2114) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Parameters

- **Negate Digital Output Close:** Inverts the control of the closing coil.
- **Negate Digital Output Open:** Inverts the control of the opening coil.

Online



State

Driving Direction: Opening

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Open:** Opens the valve. The open-output will be active, if not negated in the *settings*-menu.
- **Close:** Closes the valve. The close-output will be active, if not negated in the *settings*-menu.
- **Stop:** Both outputs will be deactivated, if they are not negated.

State

- **Driving Direction:** Shows the actual state of the valve.

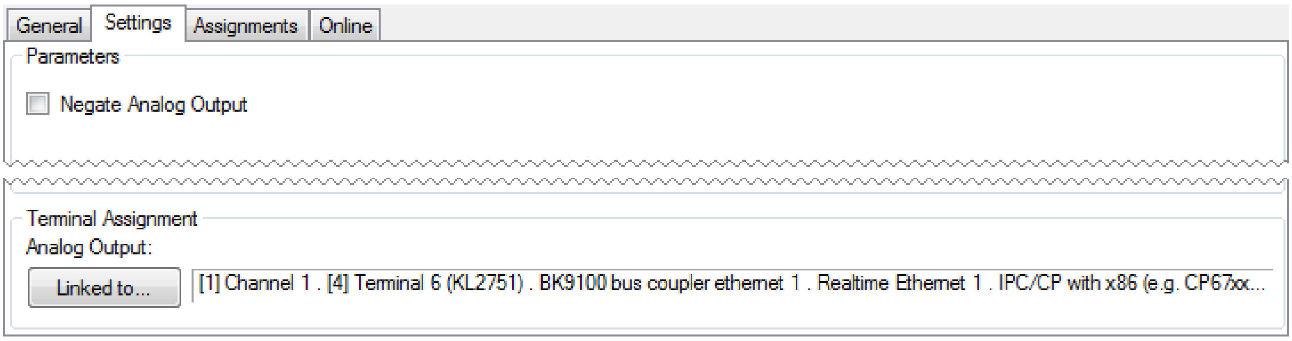
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.3.3 Continuous Valve Actuator

This module controls a continuous valve.

Settings



□ Negate Analog Output

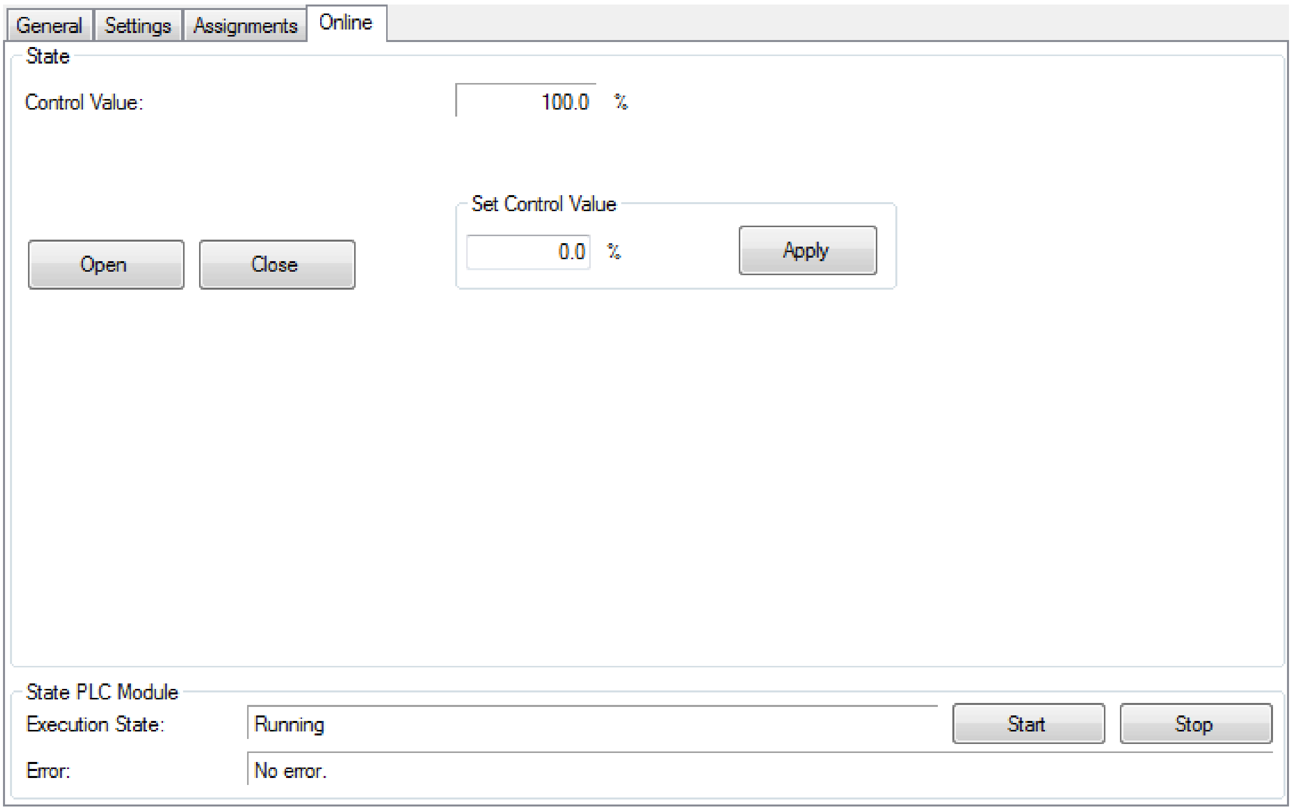
Analog Output: [1] Channel 1 . [4] Terminal 6 (KL2751) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67xx...

Parameters

- **Negate Analog Output:** Within the Building Automation Framework, the control value is counted in percentage. For the analog output terminal however the percentage value is converted to an integer range from 0 to 32767. When negation is activated, the percentage value will be converted to the complementary range from 32767 down to 0.
  Examples:

  Control ValueAnalog output valueAnalog output value (negated)0%03276720%65532621440%131071966060%196601310780%262146553100%32767 0

Online



Control Value: 100.0 %

Set Control Value: 0.0 %

Execution State: Running

Error: No error.

Commands

- **Open:** Opens the valve completely. The control-value will be set to 100%.
- **Close:** Closes the valve completely. The control-value will be set to 0%.
- **Set Control Value:** A value between 0% and 100% can be entered manually and will be taken over to the *Control Value* by pressing *Apply*.

State

- **Control Value:** Actual control value.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.4.4 Window Drives

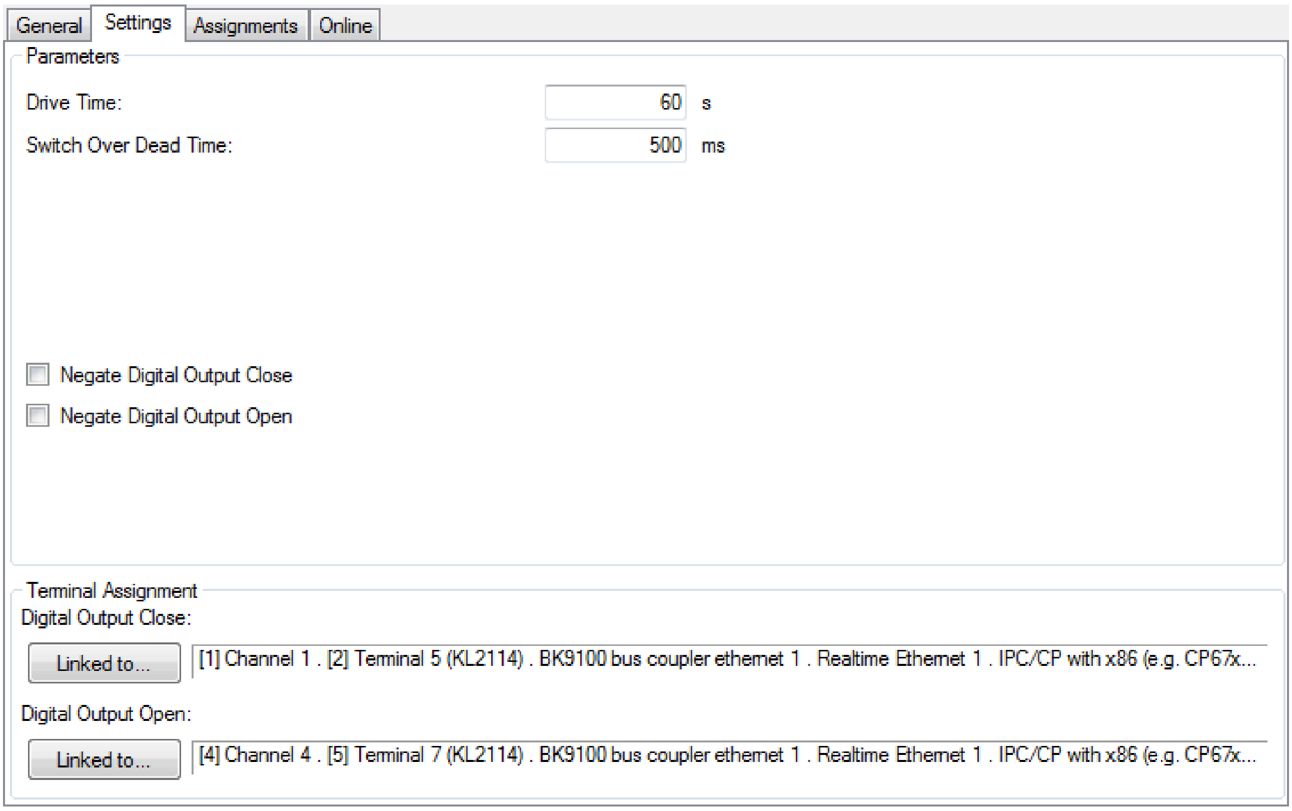
The following types of Window Drives are available:

- Standard Window Drive [▶ 98]

7.2.2.4.4.1 Standard Window Drive

This module is used to control window drives.

Settings



Drive Time: 60 s

Switch Over Dead Time: 500 ms

□ Negate Digital Output Close

□ Negate Digital Output Open

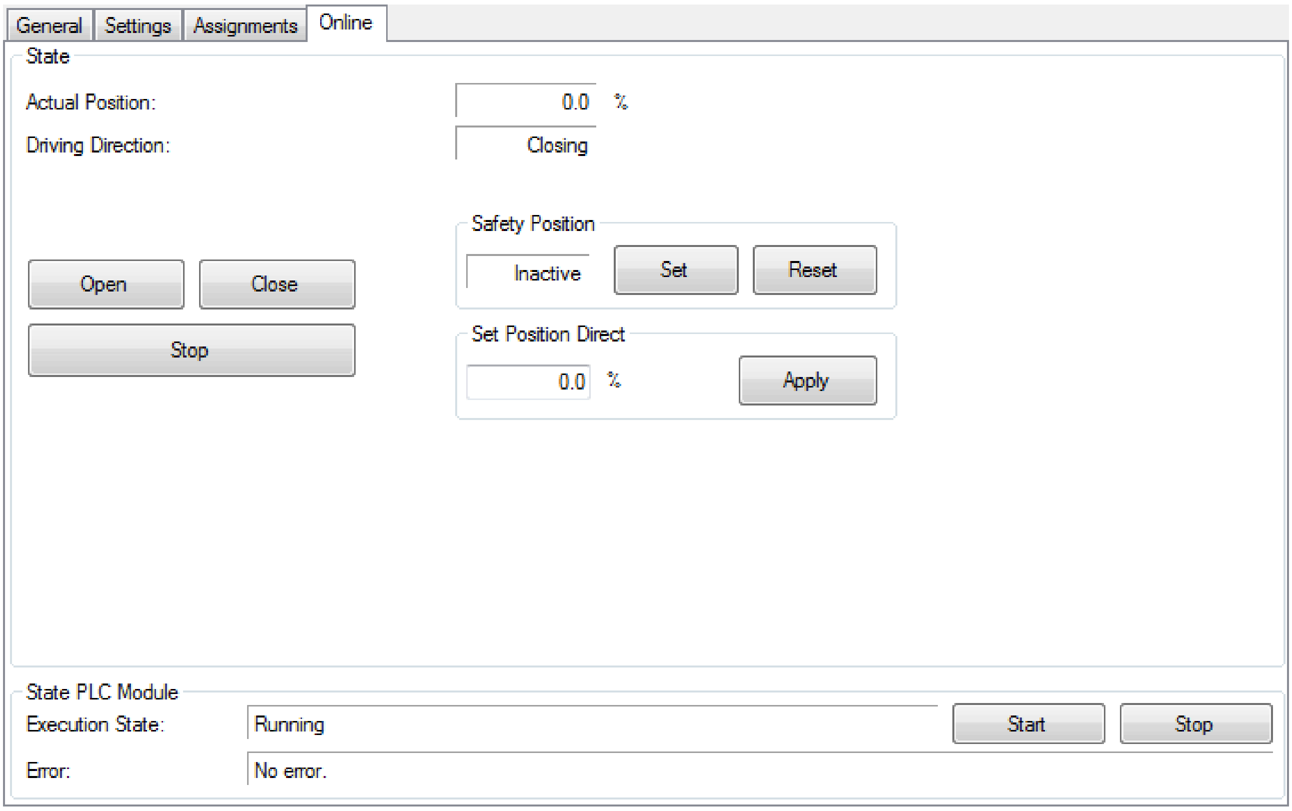
Digital Output Close: [1] Channel 1 . [2] Terminal 5 (KL2114) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67x...

Digital Output Open: [4] Channel 4 . [5] Terminal 7 (KL2114) . BK9100 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP67x...

Parameters

- **Drive Time:** Total driving time of the window.
- **Switch Over Dead Time:** Waiting time when the driving direction is changed abruptly.
- **Negate Digital Output Close/Open:** Inverts the control outputs, so a low-signal at the output-module is driving the window.

Online



Actual Position: 0.0 %

Driving Direction: Closing

Safety Position: Inactive

Set Position Direct: 0.0 %

Execution State: Running

Error: No error.

Commands

- **Open:** Opens the window.
- **Close:** Closes the window.
- **Stop:** Driving the window will be stopped.
- **Safety Position Set:** Driving to safety-position is activated. For this purpose the window will be driven in close-direction for the entered drive-time + 10%. No other commands except *Safety Position Reset* are accepted then.
- **Safety Position Reset:** Releases the safety-position lock. All other commands are accessible again.
- **Set Position Direct:** The window will be driven to the entered value, when the *Apply*-button is pressed. 0% means completely closed, 100% completely opened.

State

- **Actual Position:** Actual position in %. 0% means completely closed, 100% completely opened.
- **Driving Direction:** Actual movement.

State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.5 Sensor Groups

7.2.2.5.1 Digital Signal Groups

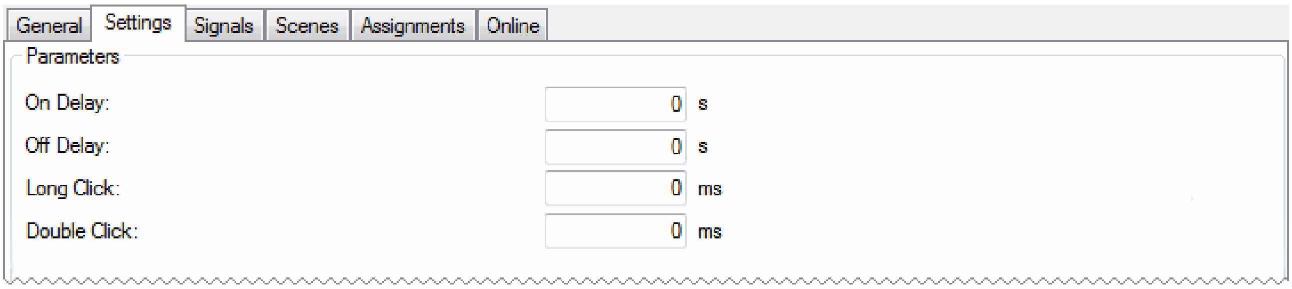
The following types of Digital Signal Groups are available:

- Standard Digital Signal Group [▶ 100]

7.2.2.5.1.1 Standard Digital Signal Group

The standard Digital Signal Group combines multiple digital signals to one group.

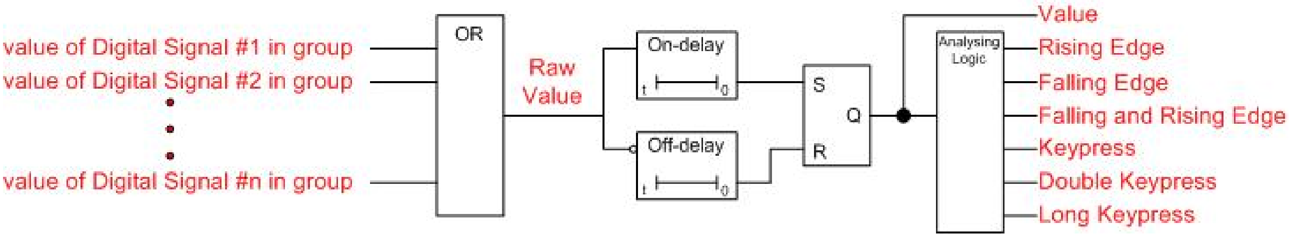
Settings



On Delay: 0 s
Off Delay: 0 s
Long Click: 0 ms
Double Click: 0 ms

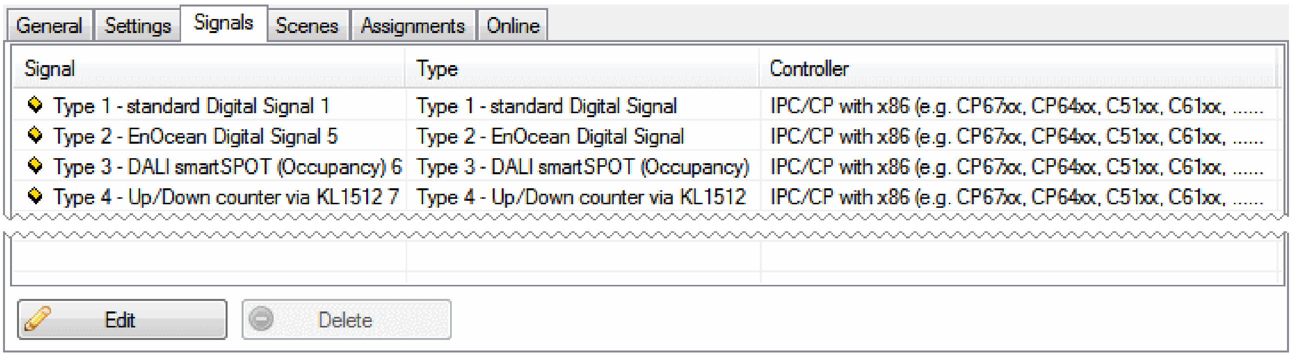
Parameters

- **On Delay:** Time delay on.
- **Off Delay:** Time delay off.
- **Long / Double Click:** Time to recognize both long and double-click, which are used to activate scenes, see below. If the button is pressed longer than this time, it is interpreted as a long click. Pressing the button twice within this time span will be recognized as a double-click.



The resulting value of the group is calculated with the resulting values of the single digital signals - not their raw values.

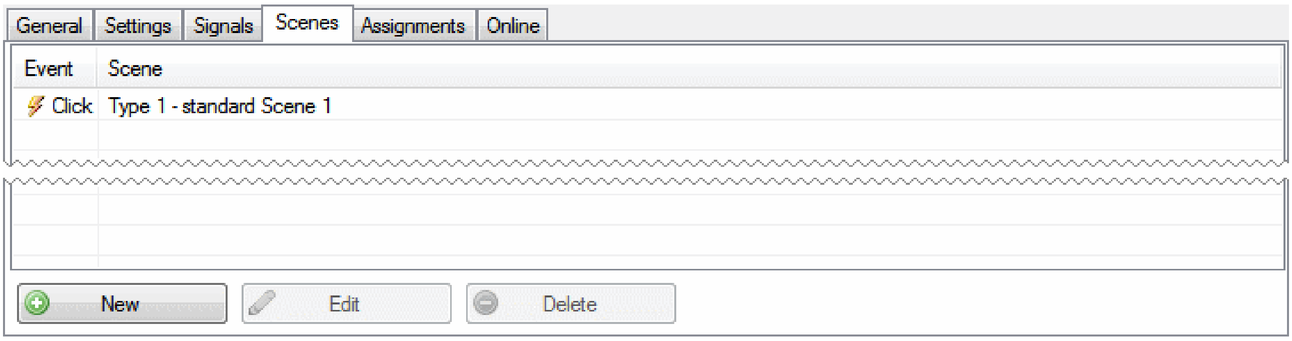
Signals



Signal	Type	Controller
Type 1 - standard Digital Signal 1	Type 1 - standard Digital Signal	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...
Type 2 - EnOcean Digital Signal 5	Type 2 - EnOcean Digital Signal	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...
Type 3 - DALI smartSPOT (Occupancy) 6	Type 3 - DALI smartSPOT (Occupancy)	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...
Type 4 - Up/Down counter via KL1512 7	Type 4 - Up/Down counter via KL1512	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...

All the devices, which shall be combined to one digital signal group, must be chosen in this menu. Pressing the *Edit*-button will open a form, in which all defined digital signals can be chosen. By marking a digital signal and pressing *Delete* the selected digital signal will be removed from the group. The digital signal will only be taken from the group, not from the whole application.

Scenes



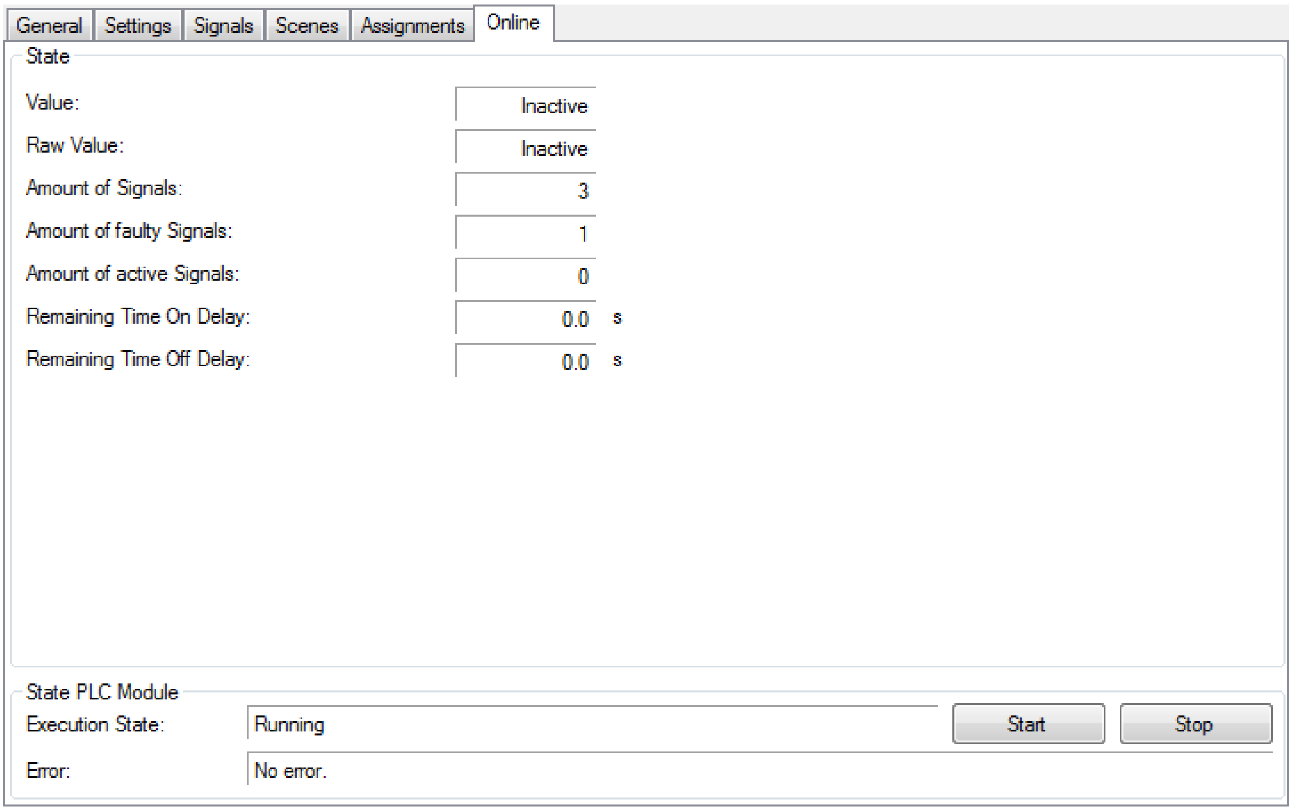
Event	Scene
Click	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the analyzed events

- Rising edge
- Falling edge
- Edge changed
- Click
- Double click
- Long click

(see logical diagram above) can be assigned to the execution of a scene. The scene however has to be defined previously.

Online



State

Value: Inactive

Raw Value: Inactive

Amount of Signals: 3

Amount of faulty Signals: 1

Amount of active Signals: 0

Remaining Time On Delay: 0.0 s

Remaining Time Off Delay: 0.0 s

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Logical OR-interconnection of all non-erroneous digital signal values followed by on and off-delay (see logical diagram above).
- **Raw Value:** Logical OR-interconnection of all non-erroneous digital signal values only.
- **Number of Signals:** Number of digital signals assigned to this group.
- **Number of faulty Signals:** Number of digital signals assigned to this group which are in an error-state.
- **Number of active Signals:** Number of digital signals assigned to this group which are in an active state.
- **Remaining Time On Delay / Remaining Time Off Delay:** Shows the remaining time of the delay-timers.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.5.2 Analog Signal Groups

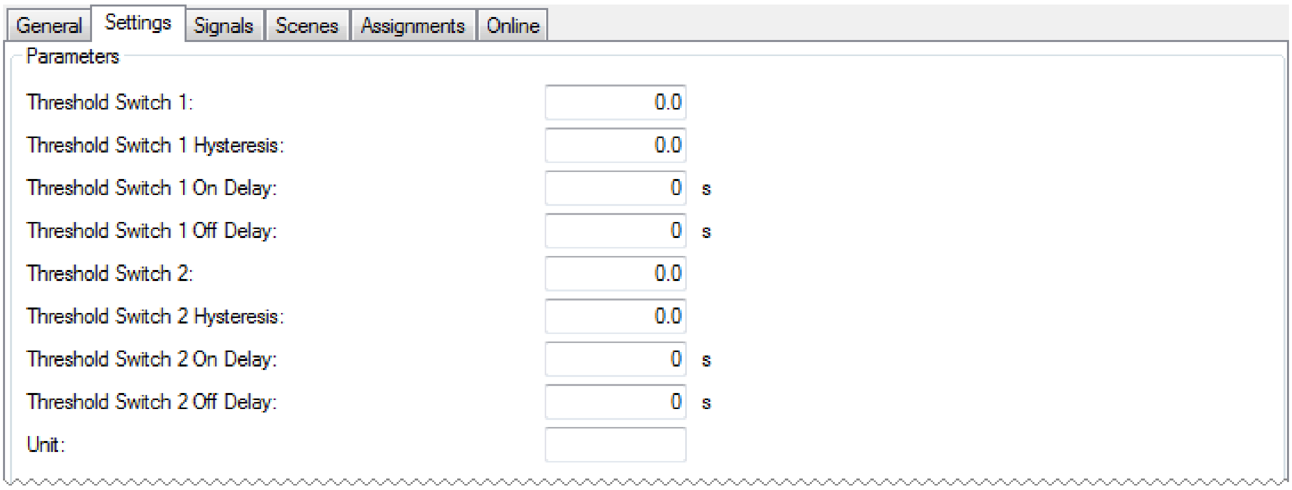
The following types of Analog Signal Groups are available:

- Standard Analog Signal Group [▶ 102]

7.2.2.5.2.1 Standard Analog Signal Group

The standard Analog Signal Group combines multiple analog signals to one group. Like in a single analog module, two threshold-switches can be defined in the group as well. The groups analog value is the averaged value of all assigned modules in this group.

Settings

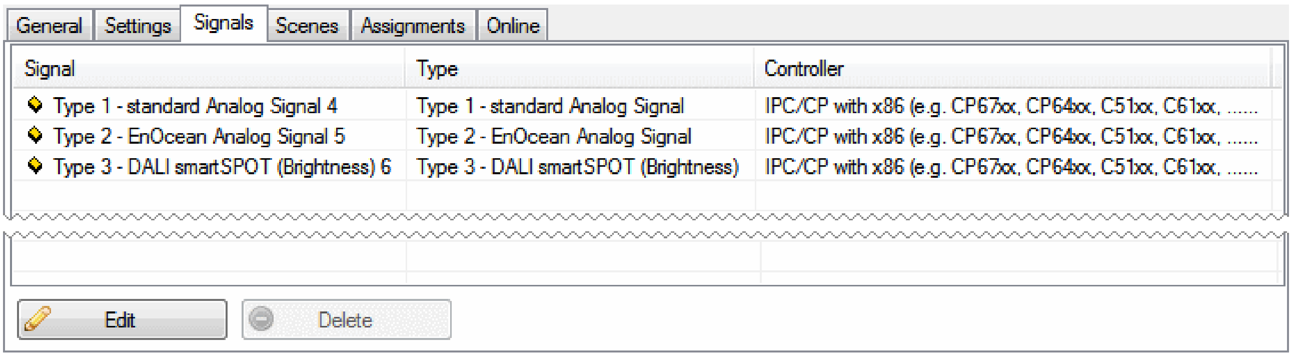


Parameters

Threshold Switch 1:	0.0	
Threshold Switch 1 Hysteresis:	0.0	
Threshold Switch 1 On Delay:	0	s
Threshold Switch 1 Off Delay:	0	s
Threshold Switch 2:	0.0	
Threshold Switch 2 Hysteresis:	0.0	
Threshold Switch 2 On Delay:	0	s
Threshold Switch 2 Off Delay:	0	s
Unit:		

Two threshold-switches can be defined for the average value. The rising and falling edges of these switches can be used to execute scenes (see *Scenes*-menu below). Additionally, a hysteresis and a delay can be defined to dampen the behavior of the average value of the group. The entered unit is used for display-purposes in the Building Automation Framework only.

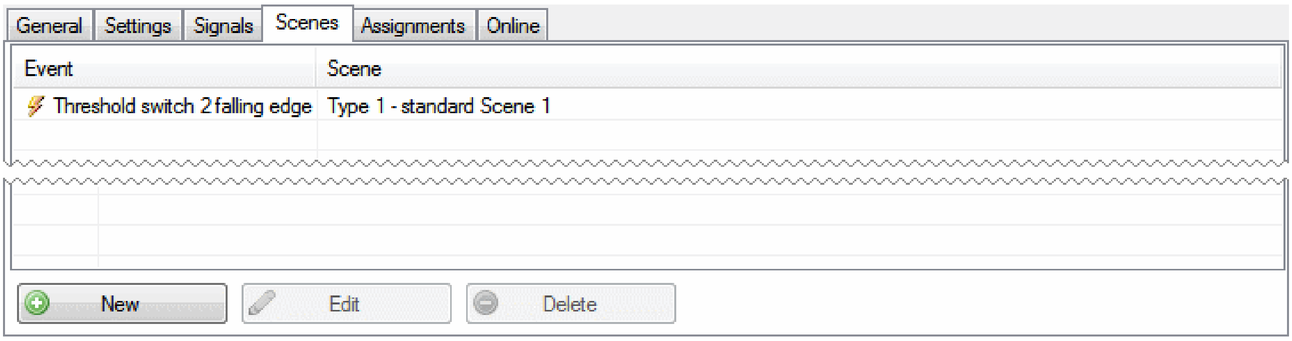
Signals



Signal	Type	Controller
Type 1 - standard Analog Signal 4	Type 1 - standard Analog Signal	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...
Type 2 - EnOcean Analog Signal 5	Type 2 - EnOcean Analog Signal	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...
Type 3 - DALI smartSPOT (Brightness) 6	Type 3 - DALI smartSPOT (Brightness)	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...

All the devices, which shall be combined to one analog signal group, must be chosen in this menu. Pressing the *Edit*-button will open a form, in which all defined analog signals can be chosen. By marking an analog signal and pressing *Delete* the selected analog signal will be removed from the group. The analog signal will only be taken from the group, not from the whole application.

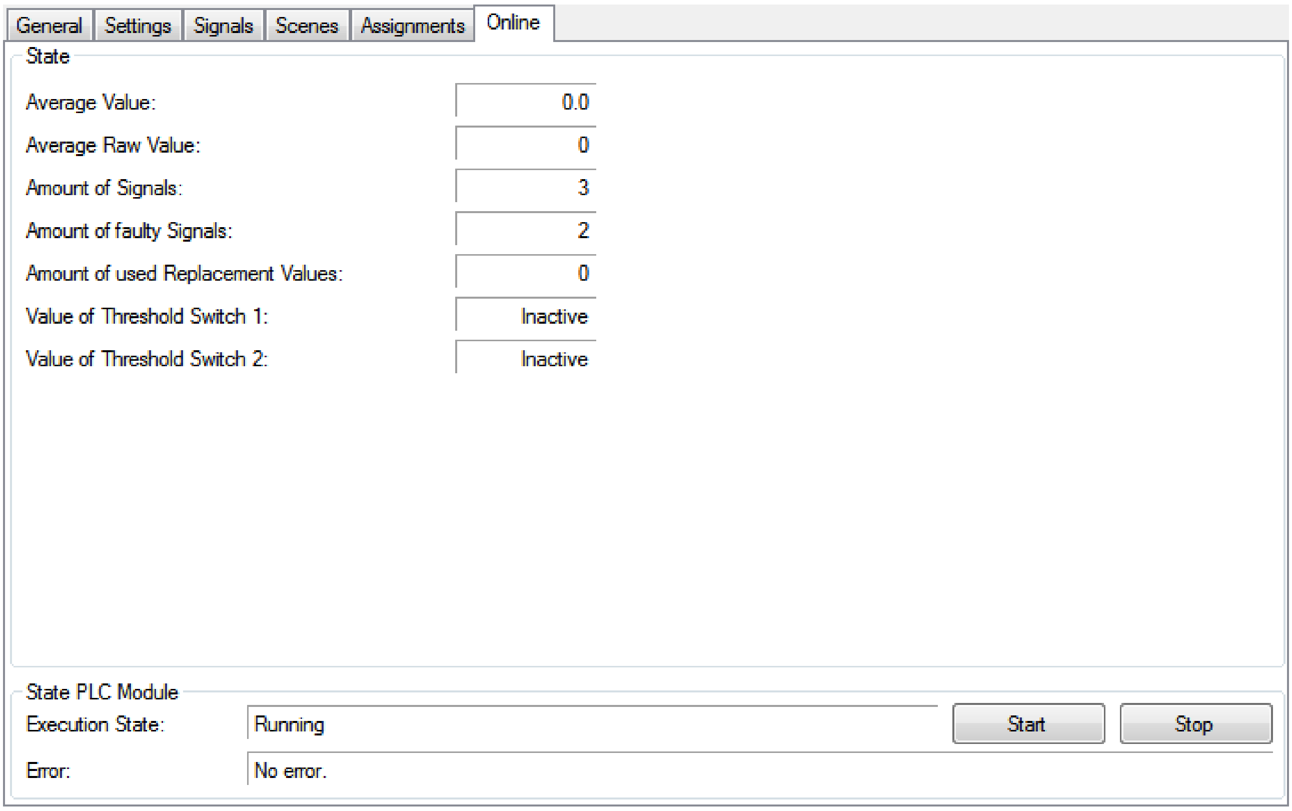
Scenes



Event	Scene
Threshold switch 2 falling edge	Type 1 - standard Scene 1

Pressing the *New*-button within the *Scenes*-menu opens a dialog, where the rising and the falling edge events of the threshold-switches can be assigned to the execution of a scene. The scene however must be defined previously.

Online



Average Value: 0.0
Average Raw Value: 0
Amount of Signals: 3
Amount of faulty Signals: 2
Amount of used Replacement Values: 0
Value of Threshold Switch 1: Inactive
Value of Threshold Switch 2: Inactive

Execution State: Running
Error: No error.

State

- **Average Value:** Calculated average value of all non-erroneous analog signals scaled values.
- **Average Raw Value:** Calculated average raw value of all non-erroneous analog signals raw values.
- **Number of Signals:** Number of analog signals assigned to this group.
- **Number of faulty Signals:** Number of analog signals assigned to this group which are in an error-state.
- **Amount of used Replacement Values:** Indicates how many signals use their *Replacement Value*.
- **Value of Threshold Switch 1/2:** Active if the average value exceeds the threshold, otherwise inactive.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.6 Actuator Groups

7.2.2.6.1 Lamp Groups

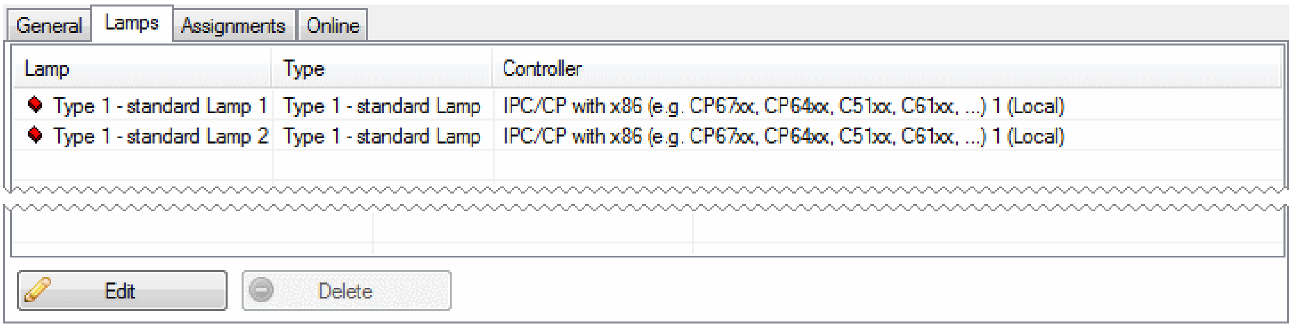
The following types of Lamp Groups are available:

- Standard Lamp Group [▶ 104]

7.2.2.6.1.1 Standard Lamp Group

The standard Lamp Group combines multiple lamps to one group.

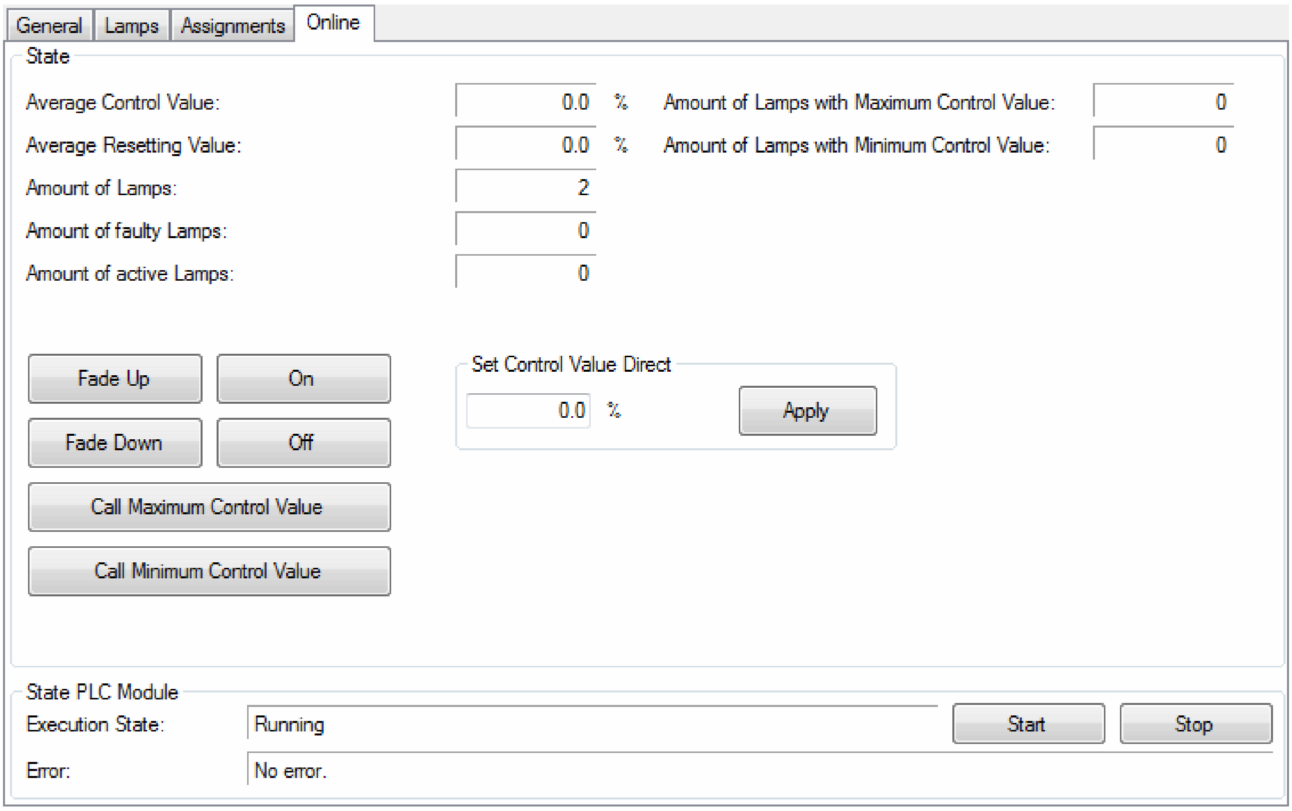
Lamps



Lamp	Type	Controller
Type 1 - standard Lamp 1	Type 1 - standard Lamp	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)
Type 1 - standard Lamp 2	Type 1 - standard Lamp	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)

All the devices, which shall be combined to one lamp group, must be chosen in this menu. Pressing the *Edit*-button will open a form, in which all defined lamps can be chosen. By marking a lamp and pressing *Delete* the selected lamp will be removed from the group. The lamp will only be taken from the group, not from the whole application.

Online



State

Average Control Value: 0.0 %

Amount of Lamps with Maximum Control Value: 0

Average Resetting Value: 0.0 %

Amount of Lamps with Minimum Control Value: 0

Amount of Lamps: 2

Amount of faulty Lamps: 0

Amount of active Lamps: 0

Set Control Value Direct

0.0 %

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light value of all assigned lamps by their entered fade-rate. The command will only work when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the assigned lamps either to the maximum or the resetting-value, which depends on the fact, if the *Resetting Mode* is activated or not. The *Resetting Mode* can be activated for each lamp of a group separately.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off-state will be stored.
- **Call Maximum Control Value:** Each lamp of the assigned group will be turned to its maximum level.
- **Call Minimum Control Value:** Each lamp of the assigned group will be turned to its minimum level.

- **Set Control Value Direct:** The entered value will be taken over to the lamps, when the *Apply*-button is pressed. If the entered value is greater than the maximum value of one of the lamps, the lamp will be set to its maximum level. If it is lower than its minimum level, but not 0, the lamp will be set to its minimum level. Entering and applying a 0 will turn the lamps off.

State

- **Average Control Value:** Average value of all active (non-erroneous) lamps.
- **Average Resetting Value:** This value shows the average *Resetting Value* of all lamps in the group.
- **Number of Lamps:** Number of lamps in this group.
- **Number of faulty Lamps:** Number of lamps in this group in an erroneous state.
- **Number of active Lamps:** Number of lamps in this group, which are turned on (light-level not 0%).

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.6.2 Blind Drive Groups

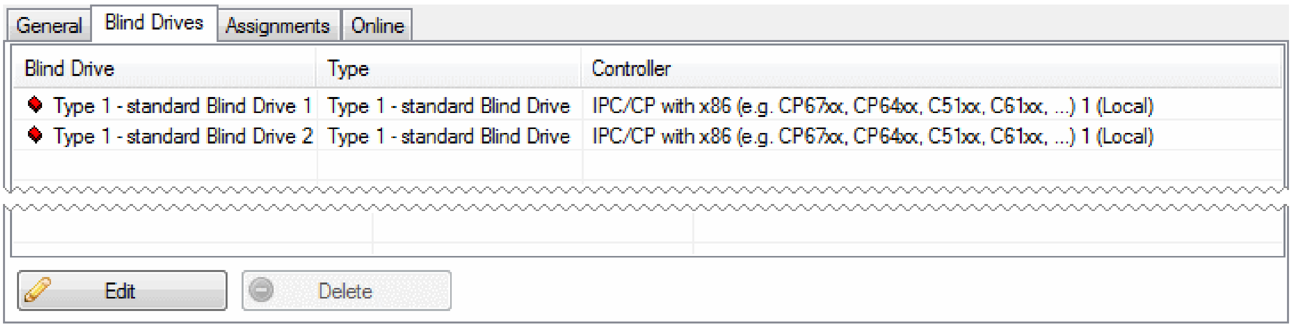
The following types of Blind Drive Groups are available:

- Standard Blind Drive Group [▶ 106]

7.2.2.6.2.1 Standard Blind Drive Group

The standard Blind Drive Group combines multiple blind-drives to one group.

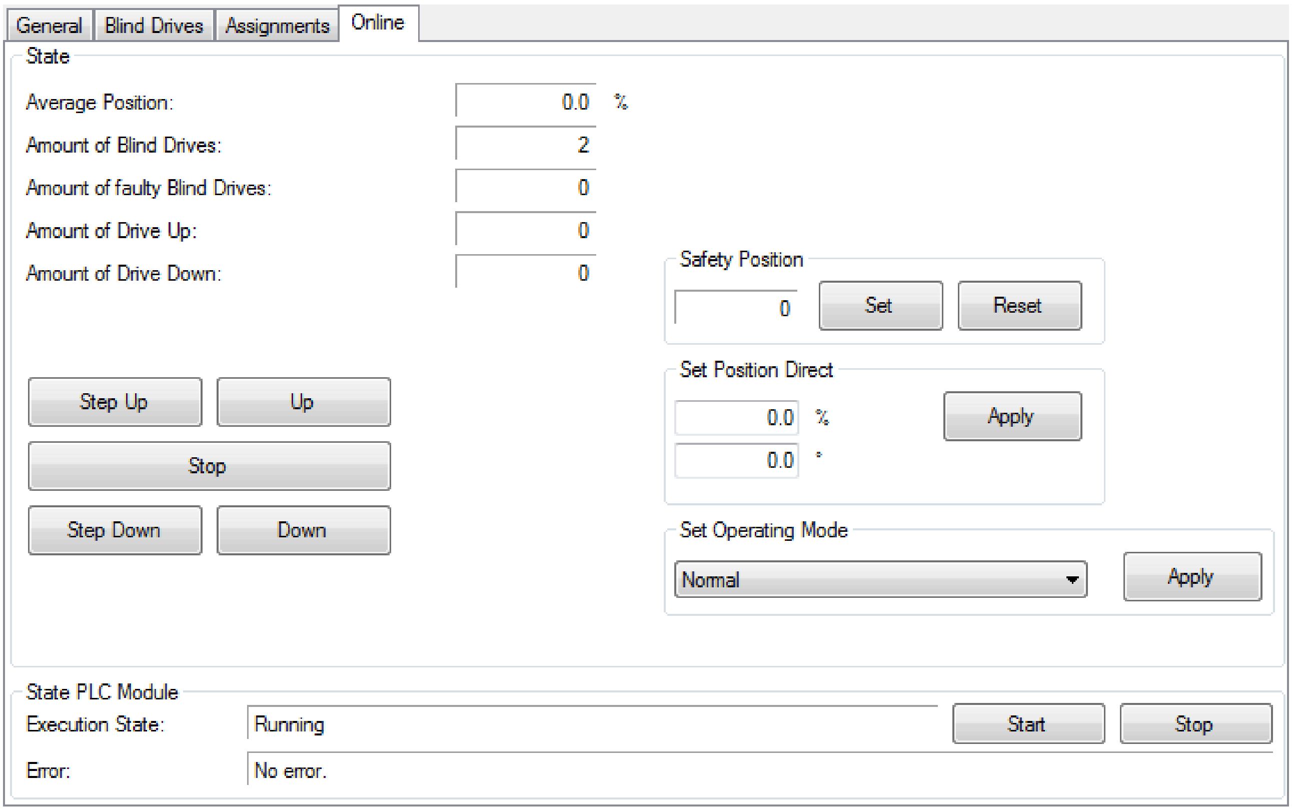
Blind Drives



Blind Drive	Type	Controller
◆ Type 1 - standard Blind Drive 1	Type 1 - standard Blind Drive	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)
◆ Type 1 - standard Blind Drive 2	Type 1 - standard Blind Drive	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)

Pressing the *Edit*-button within the *Blind Drives*-menu opens a window, where the single blind drives can be easily added to the group.

Online



State

Average Position: 0.0 %

Amount of Blind Drives: 2

Amount of faulty Blind Drives: 0

Amount of Drive Up: 0

Amount of Drive Down: 0

Safety Position: 0

Set Position Direct: 0.0 % 0.0 °

Set Operating Mode: Normal

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Step Up:** This command will lift all the blinds in the group for their individually entered *Step Time*.
- **Step Down:** This command will lower all the blinds in the group for their individually entered *Step Time*.
- **Up:** Let's all blinds in the group drive up.
- **Down:** Let's all blinds in the group drive down.
- **Stop:** All blinds will be stopped.
- **Safety Position Set:** Driving to safety-position if activated. For this purpose, all the assigned blind drives will be moved up for their entered drive-time + 10%. No other commands except *Safety Position Reset* are accepted then.
- **Safety Position Reset:** Releases the safety-position lock.
- **Set Position Direct:** The blinds will be driven to a specific position. This position can either be given by shadowing-percentage and lamella-angle or in millimeters. The selected entry must be activated by pressing the corresponding *Apply*-button.
  The angle is evaluated only by blind drives that support this feature.
- **Set Operating Mode:** 6 different modes can be selected and activated by the *Apply*-button:

  Normal: The drives can be accessed by all commands, described above. All the drives will be referenced. The blinds will be alternatingly moved up and down. The parameters for this mode are defined individually in the parameter-range of every single blind drive. The blind-lamellas will be turned to an angle, which is best for darkening the room. The blind-lamellas will be turned to an angle, which lets most of the sunlight pass. The blinds will be totally shut.

State

- **Average Position:** Average position of all active (non-erroneous) blind drives. 100% means completely closed, 0% completely opened.
- **Number of Blind Drives:** Number of drives assigned to this group.
- **Number of faulty Blind Drives:** Number of drives which are in an error-state.
- **Number of Drive Up:** Number of drives driving up.
- **Number of Drive Down:** Number of drives driving down.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.6.3 Valve Actuator Groups

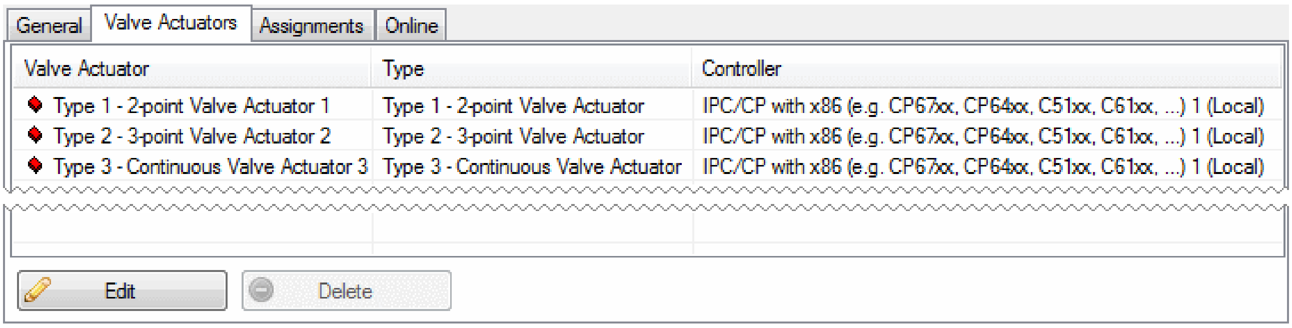
The following types of Valve Actuator Groups are available:

- Standard Valve Actuator Group [▶ 108]

7.2.2.6.3.1 Standard Valve Actuator Group

The standard Valve Actuator Group combines multiple valve actuators to one group.

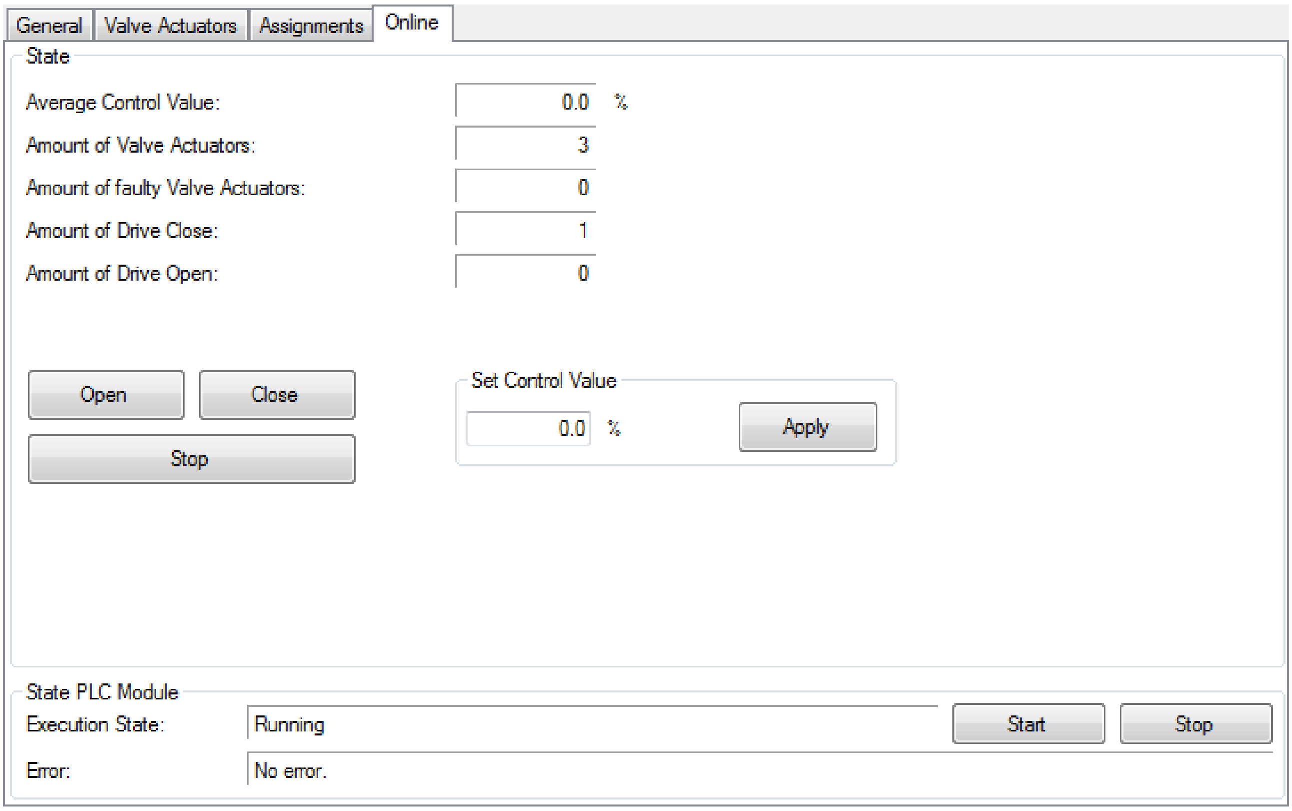
Valve Actuators



Valve Actuator	Type	Controller
◆ Type 1 - 2-point Valve Actuator 1	Type 1 - 2-point Valve Actuator	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)
◆ Type 2 - 3-point Valve Actuator 2	Type 2 - 3-point Valve Actuator	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)
◆ Type 3 - Continuous Valve Actuator 3	Type 3 - Continuous Valve Actuator	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)

All the devices, which shall be combined to one valve group, have to be chosen in this menu. Pressing the *Edit*-button will open a form, in which all defined valves can be chosen. By marking a valve and pressing *Delete* the selected valve will be removed from the group. The valve will only be taken from the group, not from the whole application.

Online



State

Average Control Value: 0.0 %

Amount of Valve Actuators: 3

Amount of faulty Valve Actuators: 0

Amount of Drive Close: 1

Amount of Drive Open: 0

Set Control Value

0.0 %

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Open:** The *Open*-command will be put through to all assigned valve drives.
- **Close:** The *Close*-command will be put through to all assigned valve drives.
- **Stop:** The *Stop*-command of all valve drives in this group will be executed.
- **Set Control Value:** The entered value will be taken over, when the *Apply*-button is pressed. The control value is entered in percentage. 0% means completely closed, 100% completely opened.

State

- **Average Control Value:** This value shows the average control value of all valves in the group.
- **Number of Valve Actuators:** Number of valve actuators in this group.
- **Number of faulty Valve Actuators:** Number of valve actuators in this group in an erroneous state.
- **Number of Drive Close:** Number of valve actuators in this group, which are currently in a closing state.
- **Number of Drive Open:** Number of valve actuators in this group, which are currently in an opening state.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.6.4 Window Drive Groups

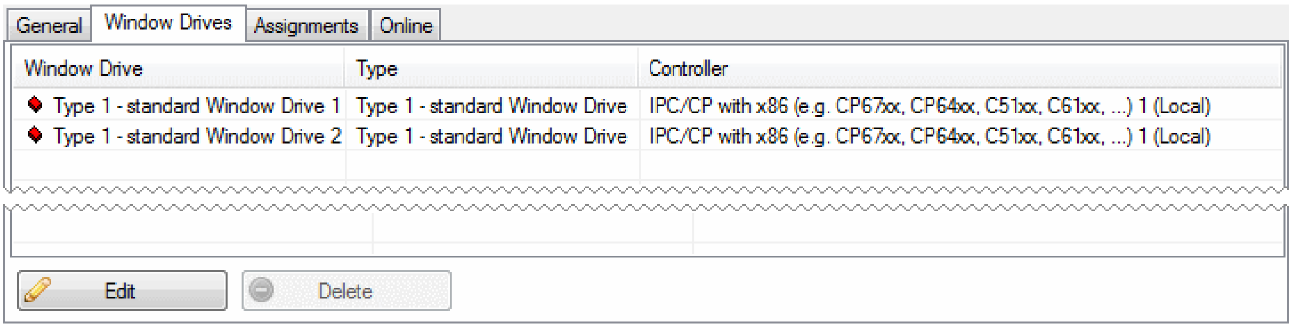
The following types of Window Drive Groups are available:

- Standard Window Drive Group [▶ 109]

7.2.2.6.4.1 Standard Window Drive Group

The standard Window Drive Group combines multiple window drives to one group.

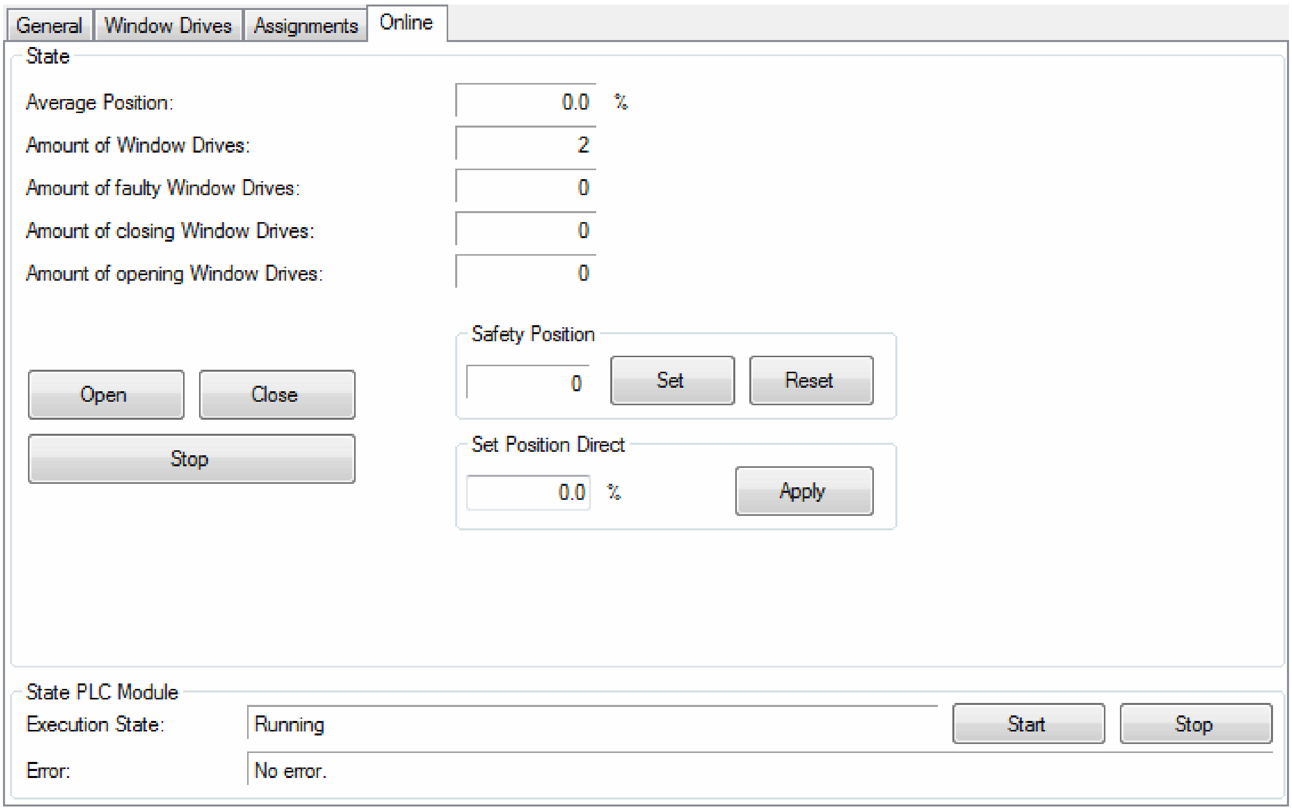
Window Drives



Window Drive	Type	Controller
Type 1 - standard Window Drive 1	Type 1 - standard Window Drive	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)
Type 1 - standard Window Drive 2	Type 1 - standard Window Drive	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) 1 (Local)

All the devices, which shall be combined to one window drive group, have to be chosen in this menu. Pressing the *Edit*-button will open a form, in which all defined window drives can be chosen. By marking a window drive and pressing *Delete* the selected window drive will be removed from the group. The window drive will only be taken from the group, not from the whole application.

Online



State

Average Position: 0.0 %

Amount of Window Drives: 2

Amount of faulty Window Drives: 0

Amount of closing Window Drives: 0

Amount of opening Window Drives: 0

Safety Position: 0

Set Position Direct: 0.0 %

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Open:** Opens all windows in the group.
- **Close:** Closes all windows in the group.
- **Stop:** All window-drives will be stopped.
- **Safety Position Set:** Driving to safety-position if activated and all the assigned windows will close. No other commands except *Safety Position Reset* are accepted then.
- **Safety Position Reset:** Releases the safety-position lock.
- **Set Position Direct:** The entered value will be taken over to the windows, when the *Apply*-button is pressed. The position is entered in percentage. 0% means completely closed, 100% completely opened.

State

- **Average Position:** Average position of all active (non-erroneous) window-drives. 0% means completely closed, 100% completely opened.
- **Amount of Window Drives:** Number of drives assigned to this group.
- **Amount of faulty Window Drives:** Number of drives which are in an error-state.
- **Amount of closing Window Drives:** Number of drives actually closing.
- **Amount of opening Window Drives:** Number of drives actually opening.

State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.7 Functional Units

7.2.2.7.1 Lightings

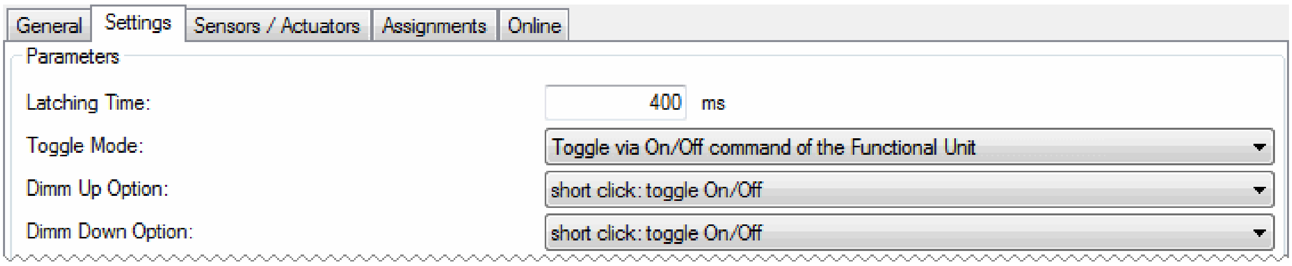
The following types of Lightings are available:

- Dimmer Switch [▶ 111]
- Stairwell Dimmer [▶ 113]
- Twighlight Switch [▶ 115]
- Daylight Lighting Control [▶ 119]
- Constant Light Regulation [▶ 124]

7.2.2.7.1.1 Dimmer Switch

The functional unit Dimmer is the standard solution for switching on / of or dimming a lamp / lamp group.

Settings



Latching Time: 400 ms

Toggle Mode: Toggle via On/Off command of the Functional Unit

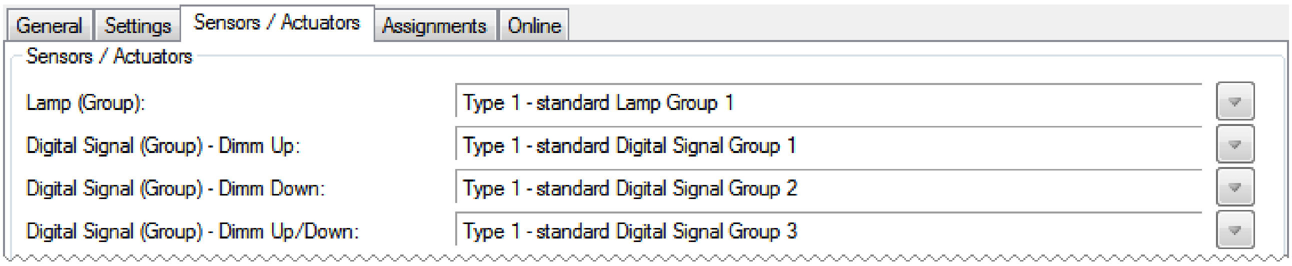
Dimm Up Option: short click: toggle On/Off

Dimm Down Option: short click: toggle On/Off

Parameters

- **Latching Time:** Time distinguishing between short and long-click of a pushbutton.
- **Toggle Mode:** If you use the Toggle Mode via "On / Off command of the functional unit", the lamp / lamp group toggles between the resetting value and "off".
- **Dimm Up Option:** This describes what the assigned lamp (group) will do, if a short-click of *Digital Signal (Group) - Dimm Up* was detected.
- **Dimm Down Option:** This describes what the assigned lamp (group) will do, if a short-click of *Digital Signal (Group) - Dimm Down* was detected.

Sensors / Actuators



Lamp (Group): Type 1 - standard Lamp Group 1

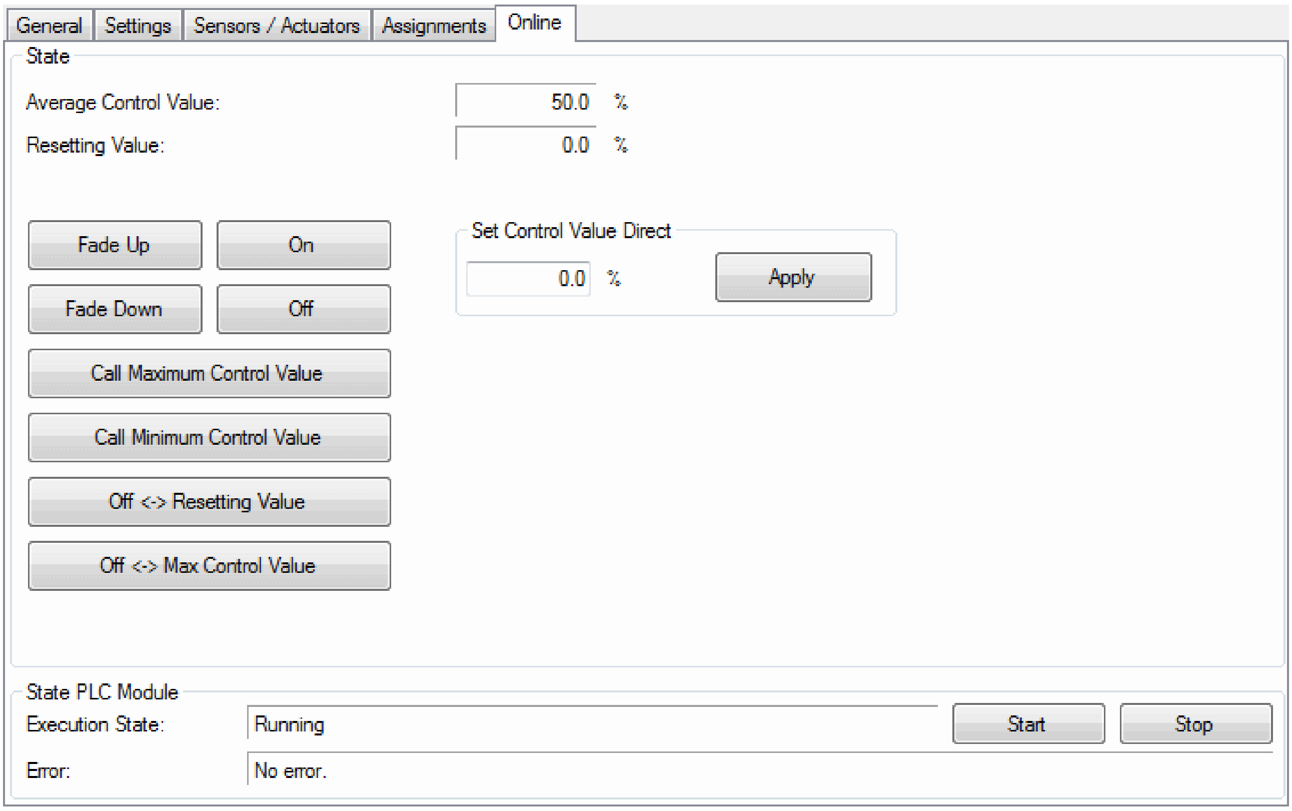
Digital Signal (Group) - Dimm Up: Type 1 - standard Digital Signal Group 1

Digital Signal (Group) - Dimm Down: Type 1 - standard Digital Signal Group 2

Digital Signal (Group) - Dimm Up/Down: Type 1 - standard Digital Signal Group 3

In this menu you enter the lamp / lamp group and the digital signals for switching the correspondent lamp / lamp group.
In general there are two different procedures to control the lamp / lamp group: with two pushbuttons for dimming up and down or with one pushbutton which alternately dims up and down. A combination of both procedures is possible.

Online



Average Control Value: 50.0 %

Resetting Value: 0.0 %

Set Control Value Direct: 0.0 %

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light value of all assigned lamps by their entered fade-rate. The command will only work when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the assigned lamps either to the maximum or the *Resetting Value*, which depends on the fact, if the *Resetting Mode* is activated or not. The *Resetting Mode* can be activated for each lamp separately.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off state will be stored.
- **Call Maximum Control Value:** Each lamp will be turned to its maximum level.
- **Call Minimum Control Value:** Each lamp will be turned to its minimum level.
- **Off ↔ Resetting Value:** The switching direction depends on the average value of the lamp (group). If the value is greater than 0 all lamps will be turned off. When the value is equal to 0 all lamps will be either turned to their *Resetting Value* or to their maximum value, when the *Resetting Mode* is not activated at the lamp.
- **Off ↔ Max Control Value:** The same as *Off ↔ Resetting Value* but all lamps will be toggled between off and their maximum value.
- **Set Control Value Direct:** The entered value will be taken over to the lamps, when the *Apply*-button is pressed. If the entered value is greater than the maximum value of one of the lamps, the lamp will be set to its maximum level. If it is lower than its minimum level, but not 0, the lamp will be set to its minimum level. Entering and applying 0 will turn the lamps off.

State

- **Average Control Value:** Average value of all active (non-erroneous) lamps.
- **Resetting Value:** When the functional unit is turned off, the previous light level will be stored in this variable. If you have selected the entry "Toggle via On/Off command of the Functional Unit" as "Toggle Mode", the lamps in the functional unit are set to this value, when they are switched on with the command *Off ↔ Resetting Value* and the Resetting Mode of the lamp is deactivated.

State PLC Module

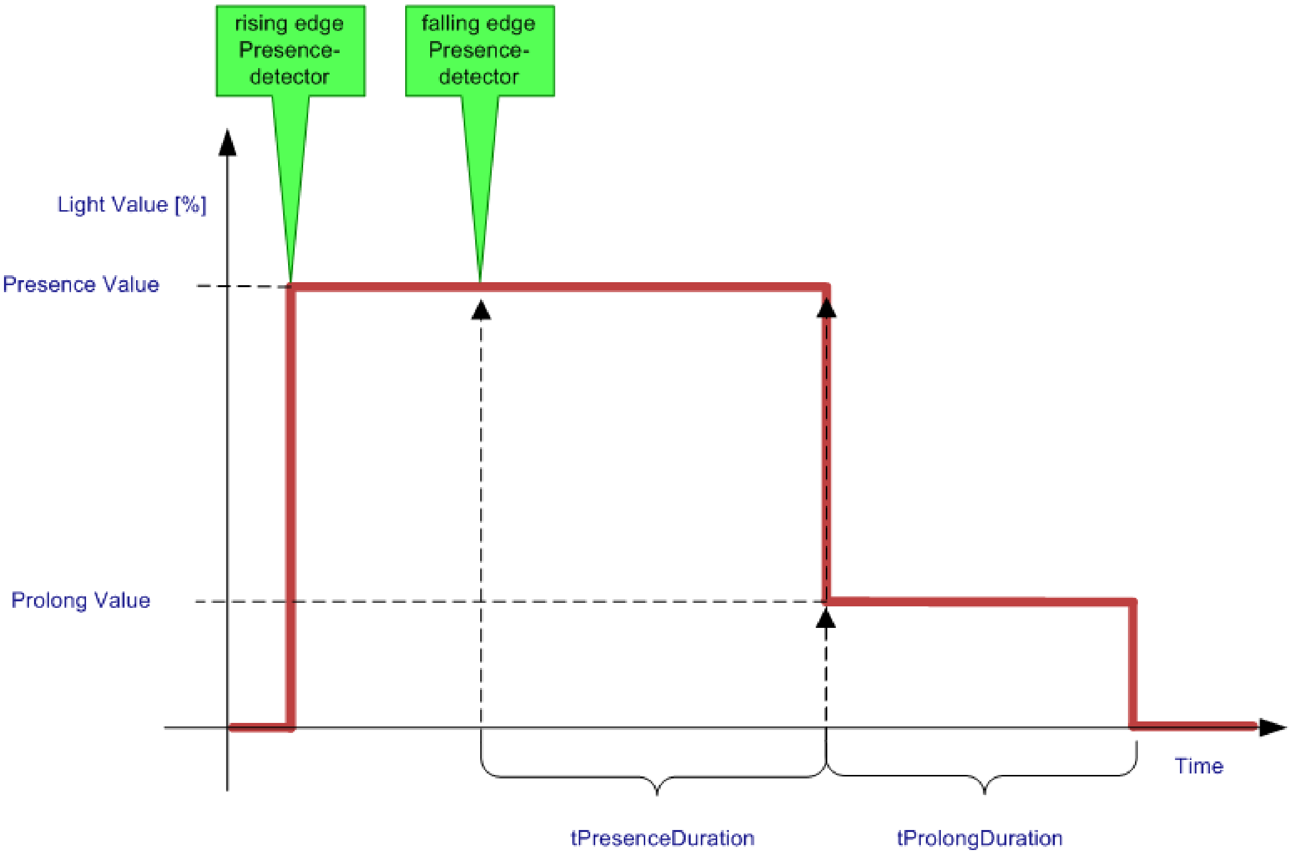
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.7.1.2 Stairwell Dimmer

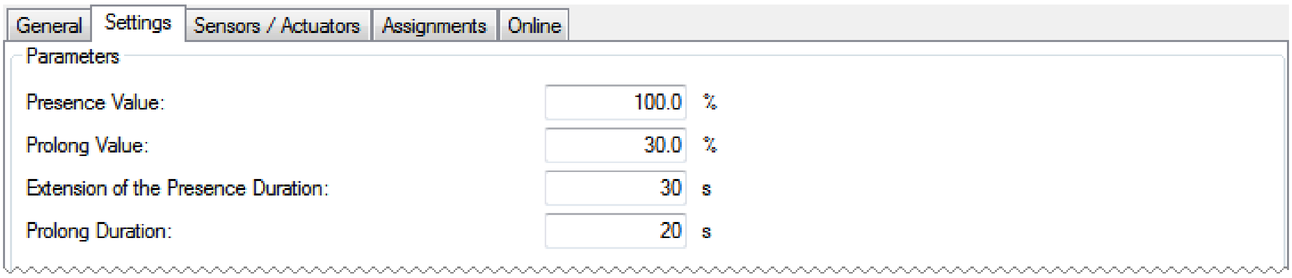
The Stairwell dimmer contains a constant sequence. After a rising trigger at the motion-detector is recognized, the light is turned on to the *Presence Value* and the sequence is started:

- Waiting for a falling trigger at the motion-detector
- Waiting time on the *Presence Value*
- Switching to the *Prolong Value*
- Waiting time on the *Prolong Value*
- Off-switching

Another rising edge at the motion-detector restarts the sequence anytime.



Settings



Presence Value: 100.0 %

Prolong Value: 30.0 %

Extension of the Presence Duration: 30 s

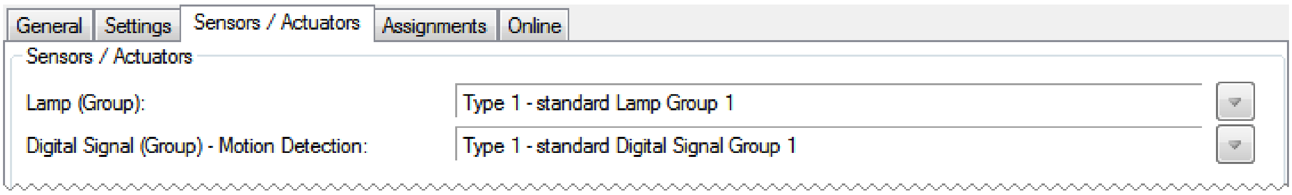
Prolong Duration: 20 s

Parameters

- **Presence Value:** Light value when the sequence is just activated.

- **Prolong Value:** Light value to signal, that the sequence will be terminated soon.
- **Extension of the Presence Duration:** Time, the light value stays on the *Presence Value* after a falling trigger on the motion detector was detected.
- **Prolong Duration:** Time, the light value stays on the *Prolong Value* before the light is turned off.

Sensors / Actuators

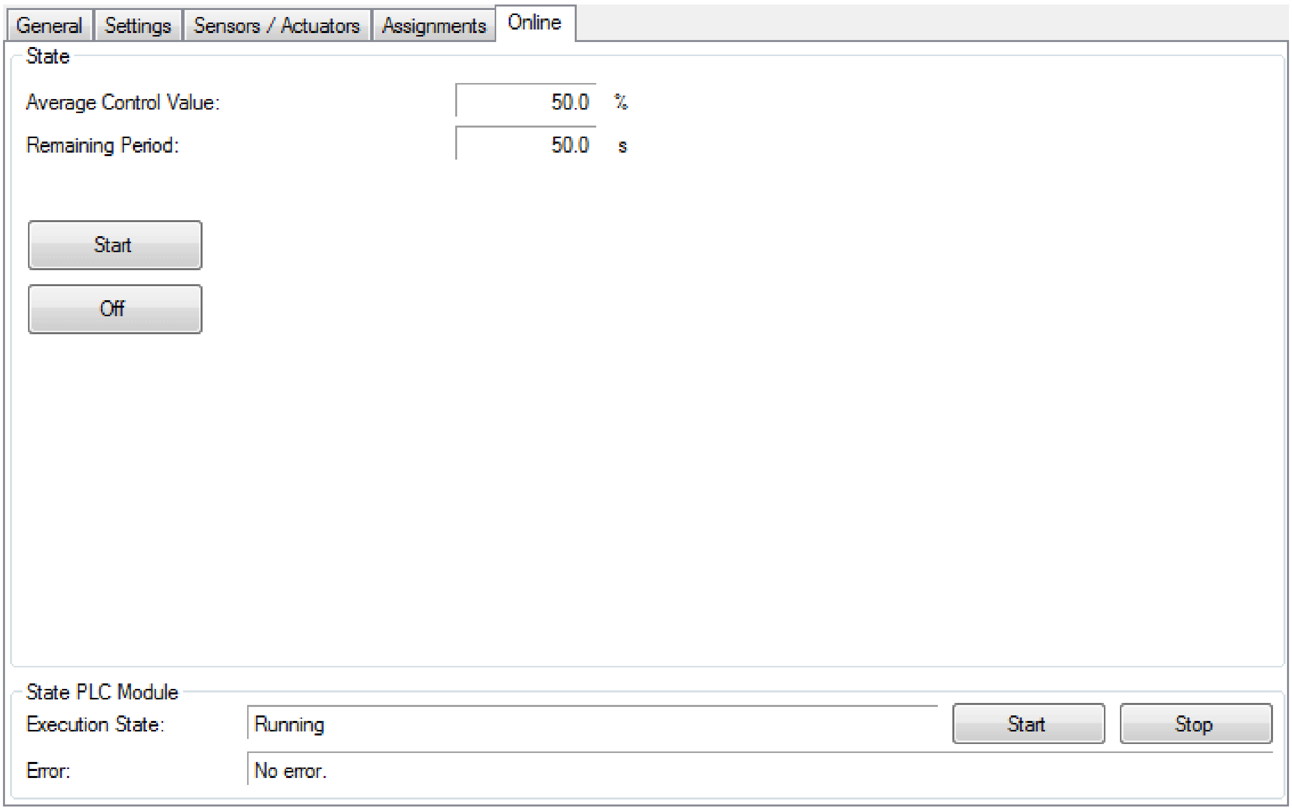


Lamp (Group): Type 1 - standard Lamp Group 1

Digital Signal (Group) - Motion Detection: Type 1 - standard Digital Signal Group 1

In this menu the lamp / lamp group has to be entered together with the correspondent digital signal / digital signal group and the motion detection which switches them.

Online



Average Control Value: 50.0 %

Remaining Period: 50.0 s

Execution State: Running

Error: No error.

Commands

- **Start:** With this button it is possible, to start the dim-sequence manually.
- **Off:** Turns the assigned lamps off and ends the sequence immediately.

State

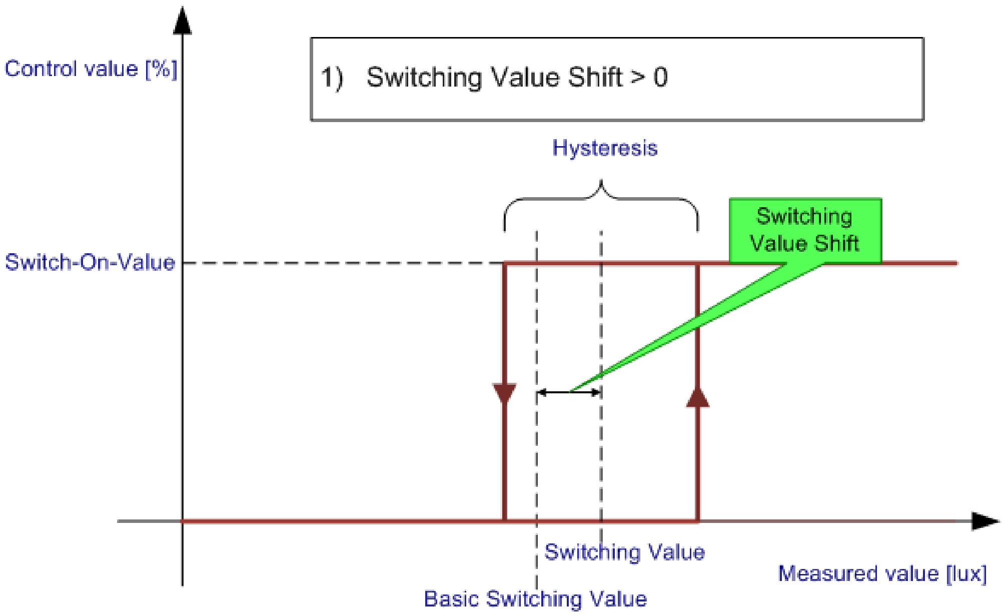
- **Average Control Value:** Average value of all active (non-erroneous) lamps.
- **Remaining Period:** Remaining time until the light will be turned off.

State PLC Module

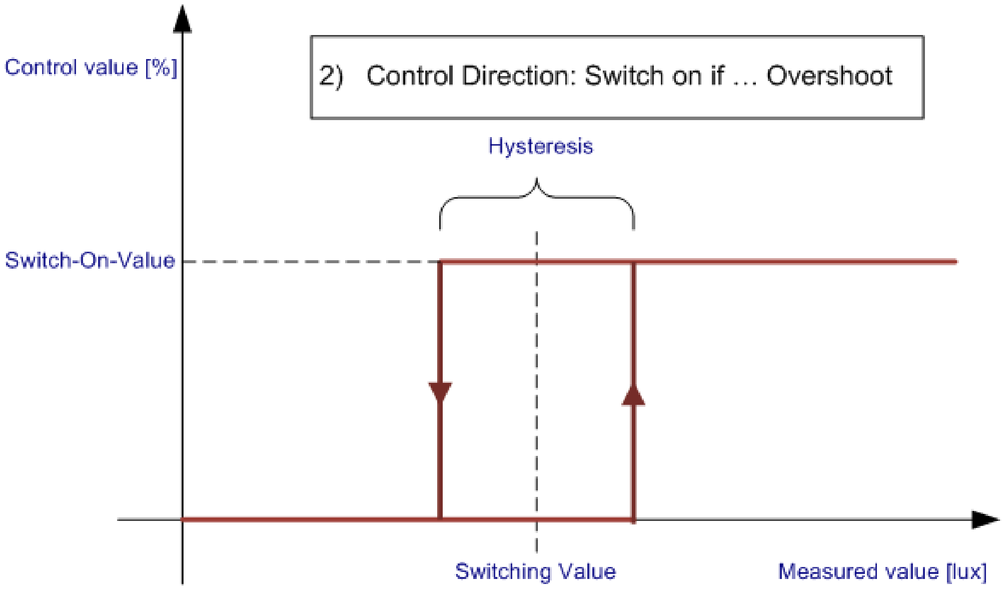
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.7.1.3 Twilight Switch

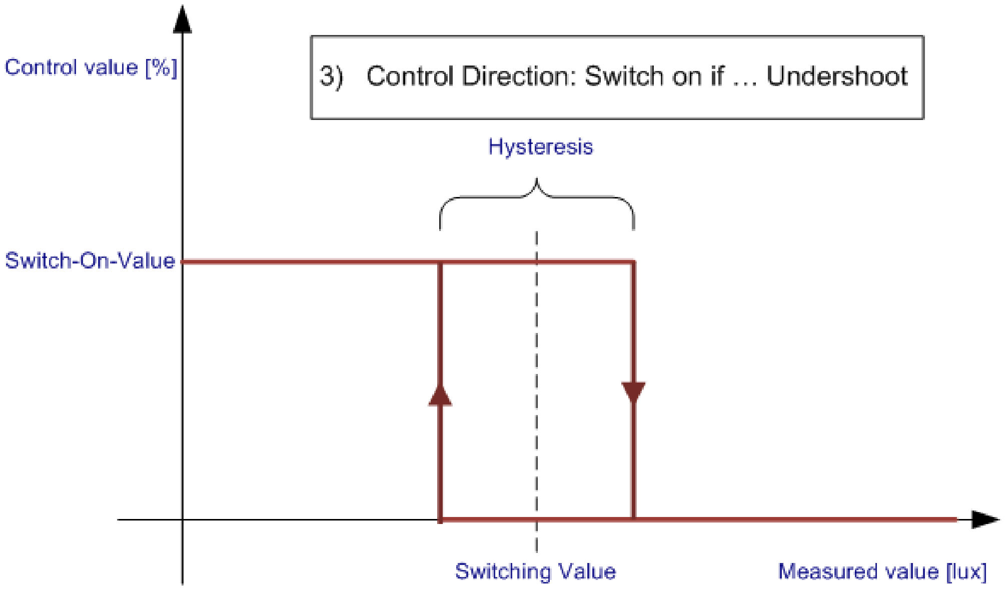
The Twilight Switch basically a 2-point-controller, which is defined by a setpoint value and a hysteresis, as well as the control-direction. The switching values are located half a hysteresis above and below the *Basic Switching Value*, which can be adjusted by an offset: the switching value-shift. The *Control direction (Switch on if...)* lets the user decide, if the controller is turning its output to true exceeding the upper switch point, or reaching below the lower setpoint:



1) Switching Value Shift > 0

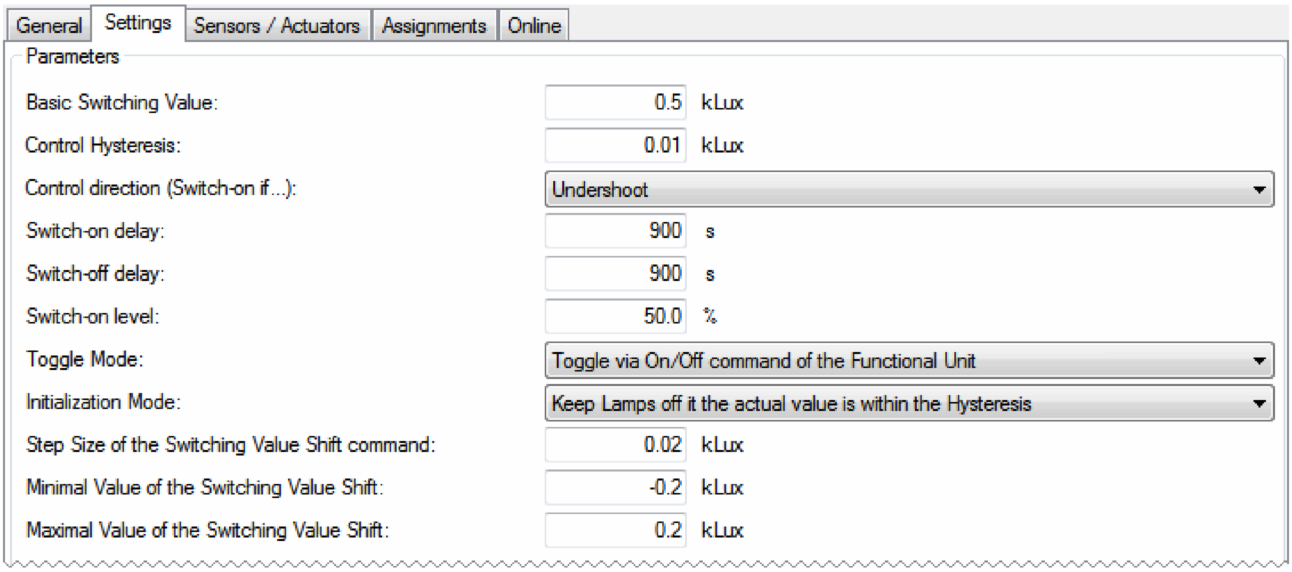


2) Control Direction: Switch on if … Overshoot



3) Control Direction: Switch on if … Undershoot

Settings



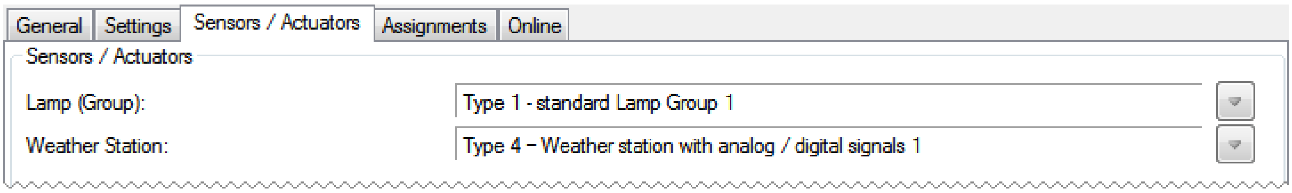
Parameters

Basic Switching Value:	0.5	kLux
Control Hysteresis:	0.01	kLux
Control direction (Switch-on if...):	Undershoot	
Switch-on delay:	900	s
Switch-off delay:	900	s
Switch-on level:	50.0	%
Toggle Mode:	Toggle via On/Off command of the Functional Unit	
Initialization Mode:	Keep Lamps off it the actual value is within the Hysteresis	
Step Size of the Switching Value Shift command:	0.02	kLux
Minimal Value of the Switching Value Shift:	-0.2	kLux
Maximal Value of the Switching Value Shift:	0.2	kLux

Parameters

- **Basic Switching Value:** Setpoint value of the twilight-switch. Can be adjusted by an offset, the *Switching Value Shift*, see *Online*-menu.
- **Control Hysteresis:** Switching-hysteresis around the *Basic Switching Value*, see diagrams above.
- **Control direction:** Behavior of the twilight-switch:
  - Switch-on if *Undershoot*: the lamps are turned on when the measured value is below the lower switching-value.
  - Switch-on if *Overshoot*: the lamps are turned on when the measured value exceeds the upper switching-value.
- **Switch-on delay / Switch-off delay:** An additional time-hysteresis can be defined by these parameters.
- **Switch-on level:** Light-level, when the twilight-switch turns the assigned lamps on.
- **Toggle Mode:** If you use the Toggle Mode via "On / Off command of the functional unit", the lamp / lamp group toggles between the resetting value of the functional unit and "off".
  If you use the Toggle Mode via "On / Off command of the local value", the lamp / lamp group toggles between the resetting value(s) of the lamp(s) and "off".
- **Initialization Mode:** This option lets the user decide, what the assigned lamps do, when the twilight-switch is started initially, and the measured value is within the hysteresis.
- **Step Size of the Switching Value Shift command:** Refers to the *Online*-menu, where the *Switching Value Shift* can be increased/decreased by step *Up/Down*-buttons. This defines the step size.
- **Minimal / Maximal value of the Switching Value Shift:** The max/min value of *Switching Value Shift*.

Sensors / Actuators

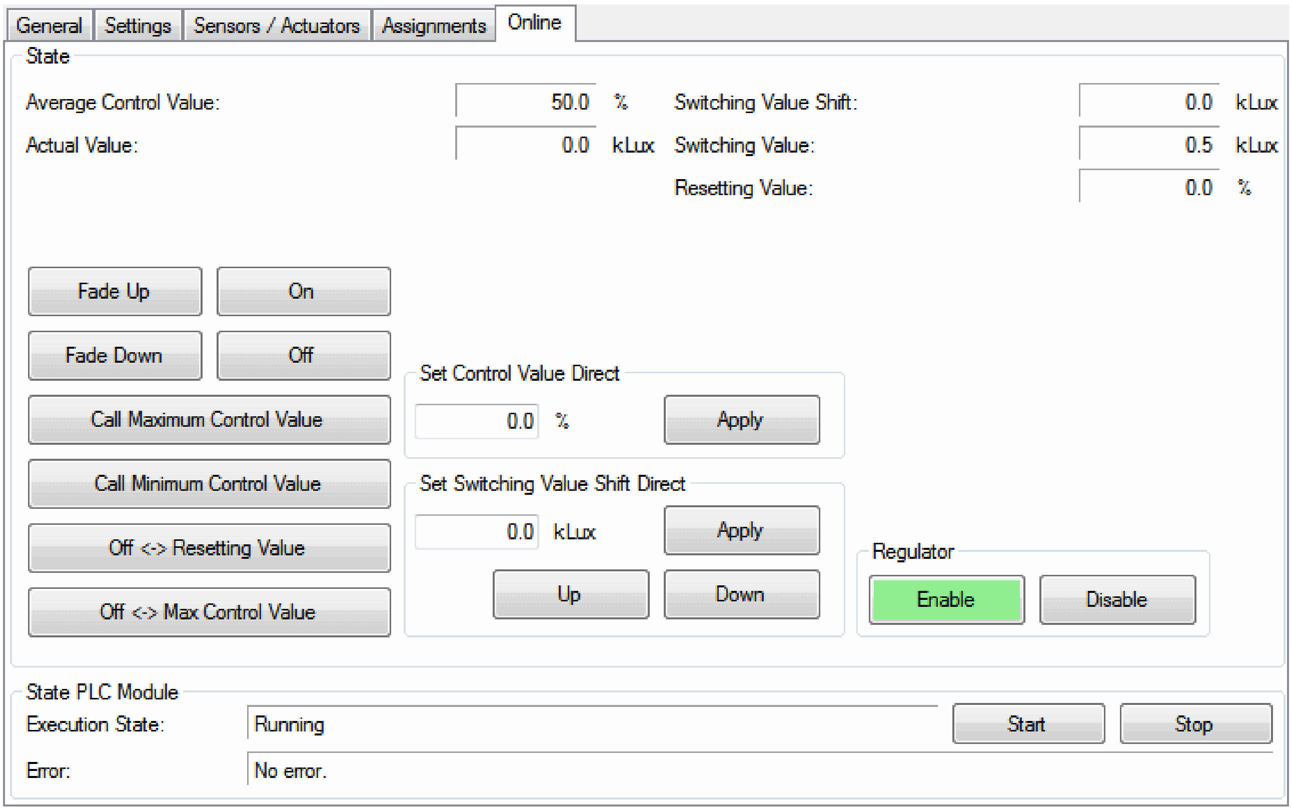


Sensors / Actuators

Lamp (Group):	Type 1 - standard Lamp Group 1
Weather Station:	Type 4 – Weather station with analog / digital signals 1

In this menu you can enter the lamp / lamp group and the weather station with the light sensors.

Online



Average Control Value: 50.0 %

Actual Value: 0.0 kLux

Switching Value Shift: 0.0 kLux

Switching Value: 0.5 kLux

Resetting Value: 0.0 %

Set Control Value Direct: 0.0 %

Set Switching Value Shift Direct: 0.0 kLux

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light value of all assigned lamps by their entered fade-rate. The command will only work when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the assigned lamps either to the maximum or the *Resetting Value*, which depends on the fact, if the *Resetting Mode* is activated or not. The *Resetting Mode* can be activated for each lamp separately.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off state will be stored.
- **Call Maximum Control Value:** Each lamp will be turned to its maximum level.
- **Call Minimum Control Value:** Each lamp will be turned to its minimum level.
- **Off ↔ Resetting Value:** The switching direction depends on the average value of the lamp (group). If the value is greater than 0 all lamps will be turned off. When the value is equal to 0 all lamps will be either turned to their *Resetting Value* or to their maximum value, when the *Resetting Mode* is not activated at the lamp.
- **Off ↔ Max Control Value:** The same as *Off ↔ Resetting Value* but all lamps will be toggled between off and their maximum value.
- **Set Control Value Direct:** The entered value will be taken over to the lamps, when the *Apply*-button is pressed. If the entered value is greater than the maximum value of one of the lamps, the lamp will be set to its maximum level. If it is lower than its minimum level, but not 0, the lamp will be set to its minimum level. Entering and applying a 0 will turn the lamps off.
- **Set Switching Value Shift Direct:** Field to enter the *Switching Value Shift* either directly in a textbox or stepping up and down with the entered step size (see *Settings*-menu). The value is, however, delimited by the maximum and minimum value in the *settings*-menu.
- **Regulator Enable/Disable:** Sets the twilight-switch in automatic/manual mode. **When the PLC is started, the regulator is enabled.**

All Input functions are only possible, when the regulator is disabled, except changing the *Switching Value Shift* and of course the control *Enable/Disable*-functions.

State

- **Average Control Value:** Average value of all active (non-erroneous) lamps.
- **Actual Value:** Measured light level.
- **Switching Value Shift:** Actual *Switching Value Shift*.
- **Switching Value:** Actual *Switching Value* resulting from the *Basic Switching Value* and the *Switching Value Shift*.
- **Resetting Value:** When the functional unit is turned off, the previous light level will be stored in this variable. If you have selected the entry "Toggle via On/Off command of the Functional Unit" as "Toggle Mode", the lamps in the functional unit are set to this value, when they are switched on with the command Off ↔ Resetting Value and the Resetting Mode of the lamp is deactivated.

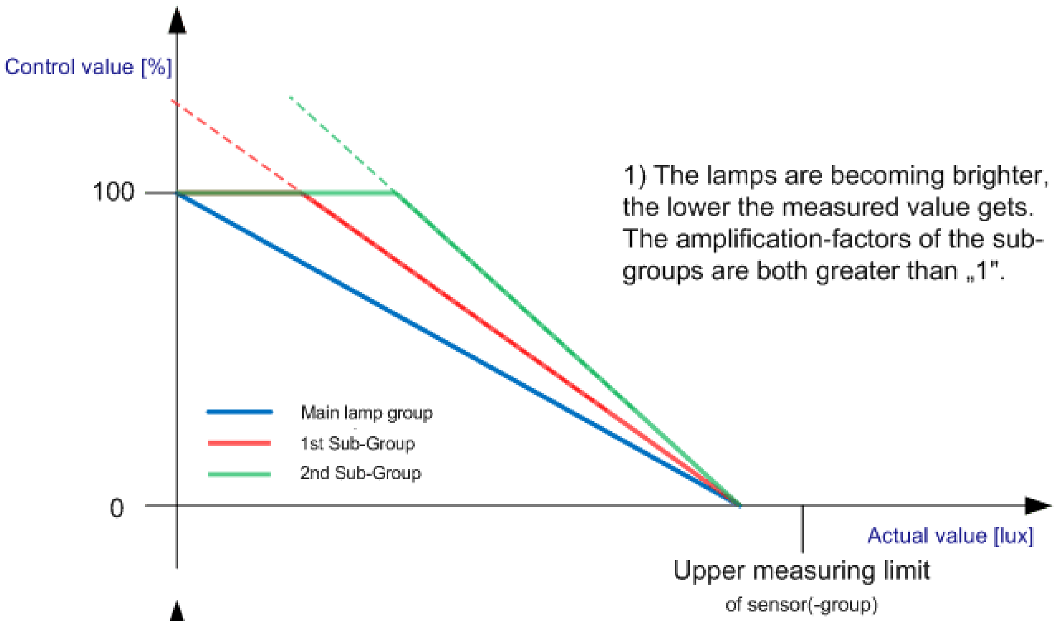
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

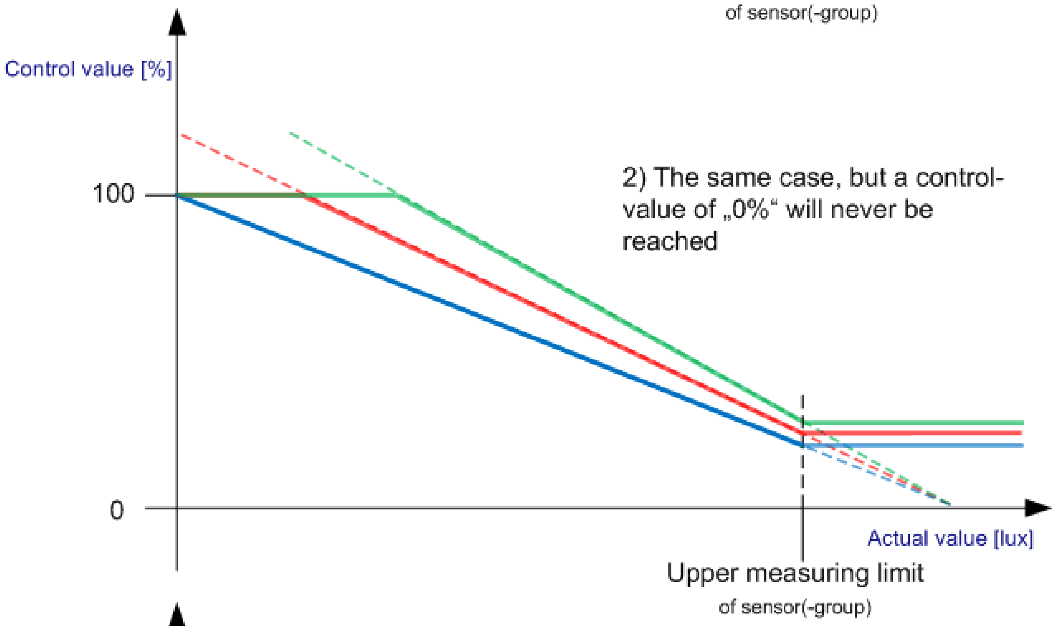
7.2.2.7.1.4 Daylight Lighting Control

The Daylight Lighting Control is a linear open-loop controller mainly consisting of an analog sensor (group) measuring the daylight, a main lamp (group) and two sub lamps/lamp groups. The linear equation for the main lamp (group) has to be entered in 2-point form: the *Actual Value* for a *Control Value* of 0% and the *Actual Value* for a *Control Value* of 100%. The sub lamp groups´ light level is calculated by the level of the main group, multiplied by amplification-factors. Because of these conditions, the three lines will always have the same zero-point. The linear equation however must be calculated. Wrong nodes and/or amplification-factors may cause the light values to run into its limits.

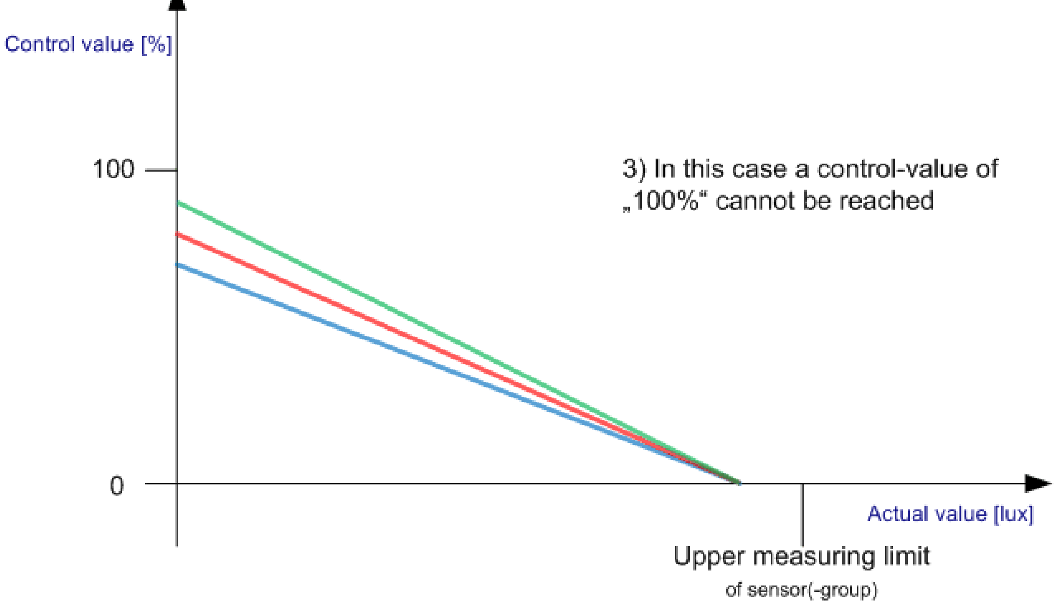
Here are some examples given by diagrams:



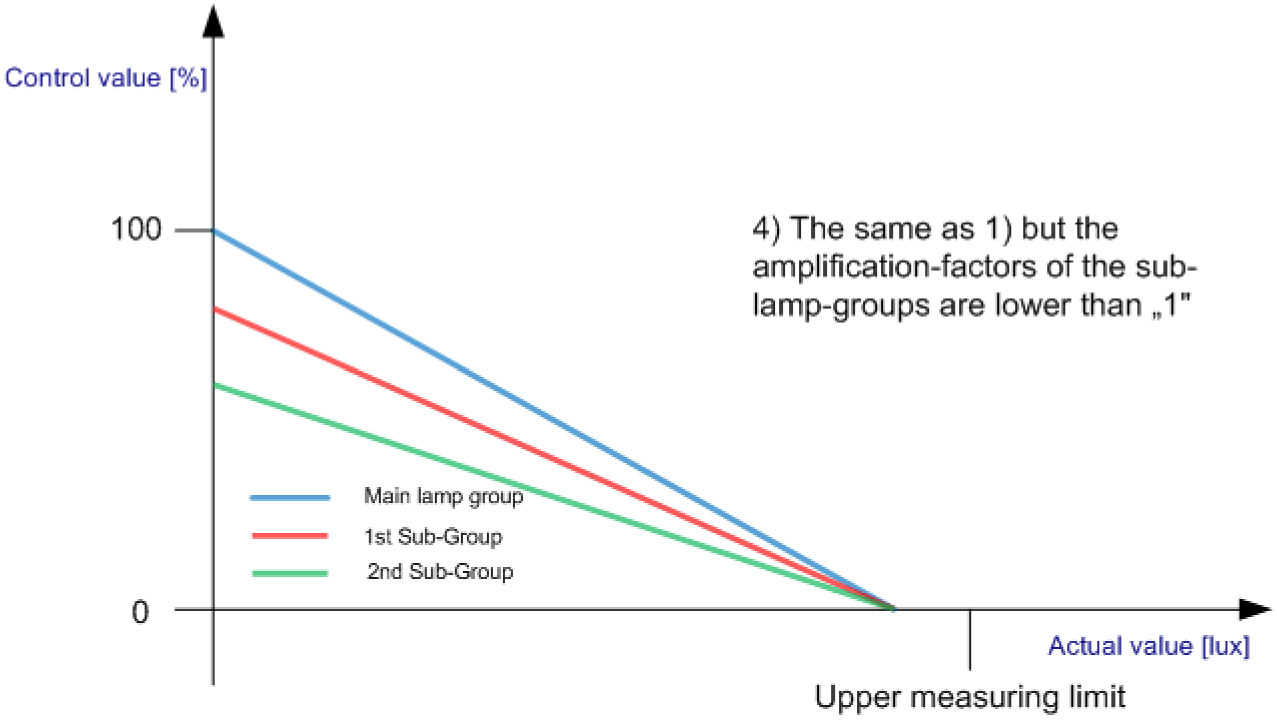
1) The lamps are becoming brighter, the lower the measured value gets. The amplification-factors of the sub-groups are both greater than „1".



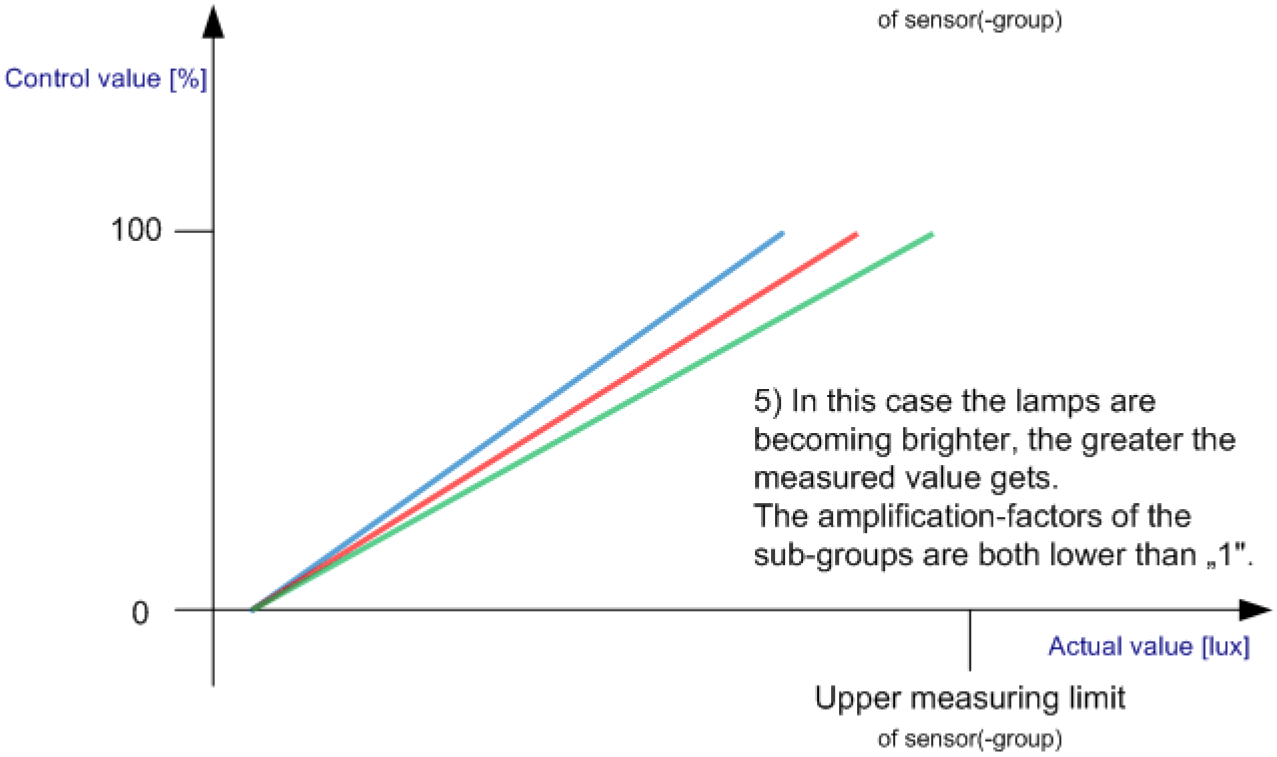
2) The same case, but a control-value of „0%“ will never be reached



3) In this case a control-value of „100%“ cannot be reached

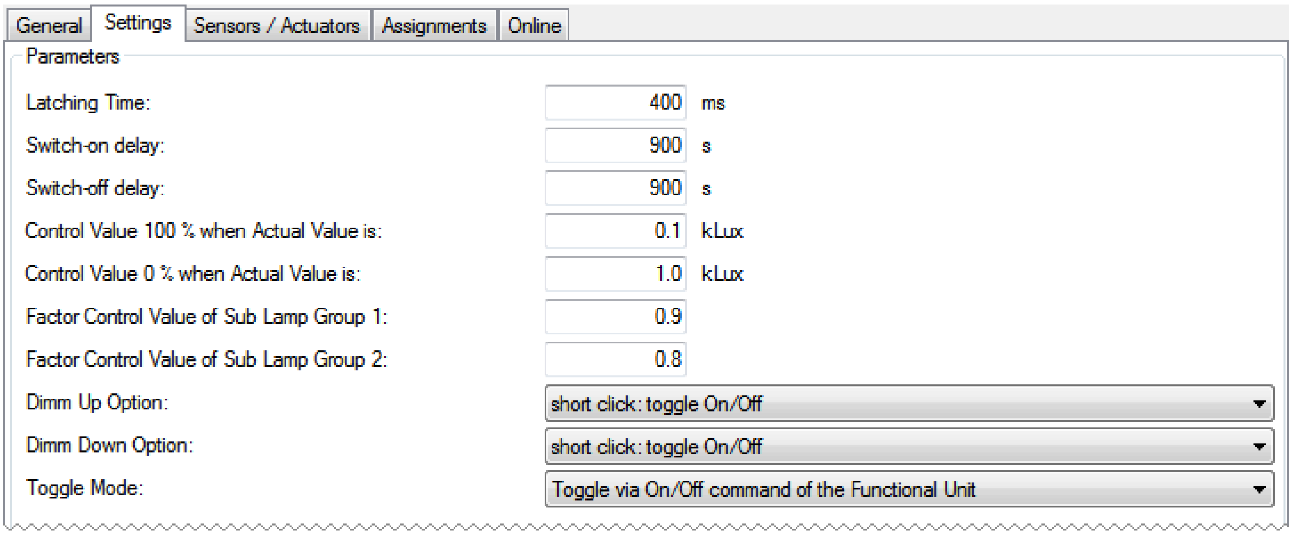


4) The same as 1) but the amplification-factors of the sub-lamp-groups are lower than „1"



5) In this case the lamps are becoming brighter, the greater the measured value gets.
The amplification-factors of the sub-groups are both lower than „1".

Settings



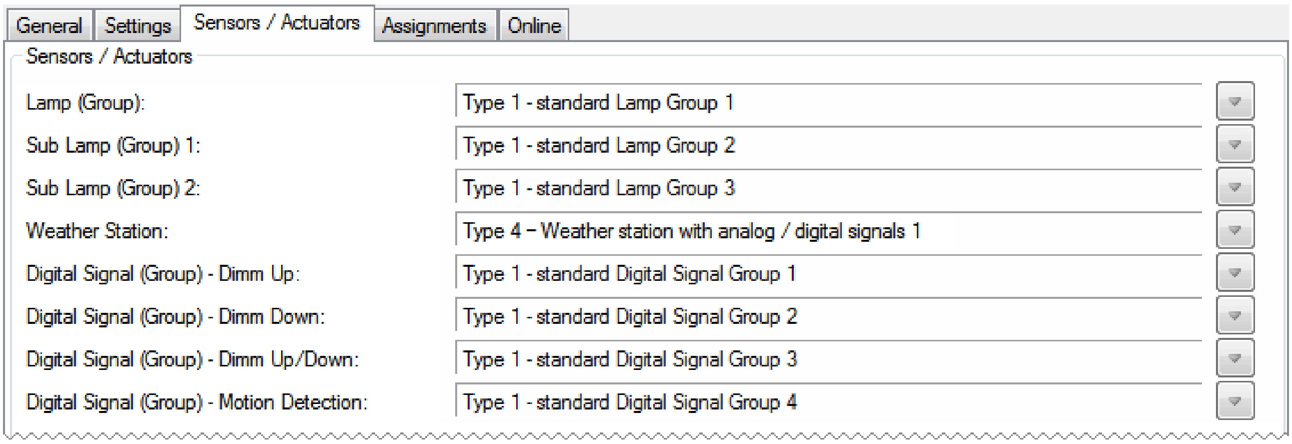
Parameters

Latching Time:	400	ms
Switch-on delay:	900	s
Switch-off delay:	900	s
Control Value 100 % when Actual Value is:	0.1	kLux
Control Value 0 % when Actual Value is:	1.0	kLux
Factor Control Value of Sub Lamp Group 1:	0.9	
Factor Control Value of Sub Lamp Group 2:	0.8	
Dimm Up Option:	short click: toggle On/Off	
Dimm Down Option:	short click: toggle On/Off	
Toggle Mode:	Toggle via On/Off command of the Functional Unit	

Parameters

- **Latching time:** Time distinguishing between short and long-click of a pushbutton.
- **Switch-on delay / Switch-off delay:** An additional delay switching the lamp on or off (0 ↔ min Value) can be defined by these parameters. This can be used to preserve the lamps from damage due to many switching-actions when the supposed control value moves around 0%.
- **Control Value 100% when Actual Value is:** First node of the linear equation.
- **Control Value 0% when Actual Value is:** Second node of the linear equation.
- **Factor Control Value of Sub Lamp (Group) 1 / Factor Control Value of Sub Lamp (Group) 2:** Amplification-factors of the sub lamps/lamp groups relating to the main lamp (group).
- **Dimm Up Option:** This describes what the assigned lamp (group) will do, if a short-click of *Digital Signal (Group) - Dimm Up* was detected.
- **Dimm Down Option:** This describes what the assigned lamp (group) will do, if a short-click of *Digital Signal (Group) - Dimm Down* was detected.
- **Toggle Mode:** If you use the Toggle Mode via "On / Off command of the functional unit", the lamp / lamp group toggles between the resetting value of the functional unit and "off". If you use the Toggle Mode via "On / Off command of the local value", the lamp / lamp group toggles between the resetting value(s) of the lamp(s) and "off".

Sensors / Actuators



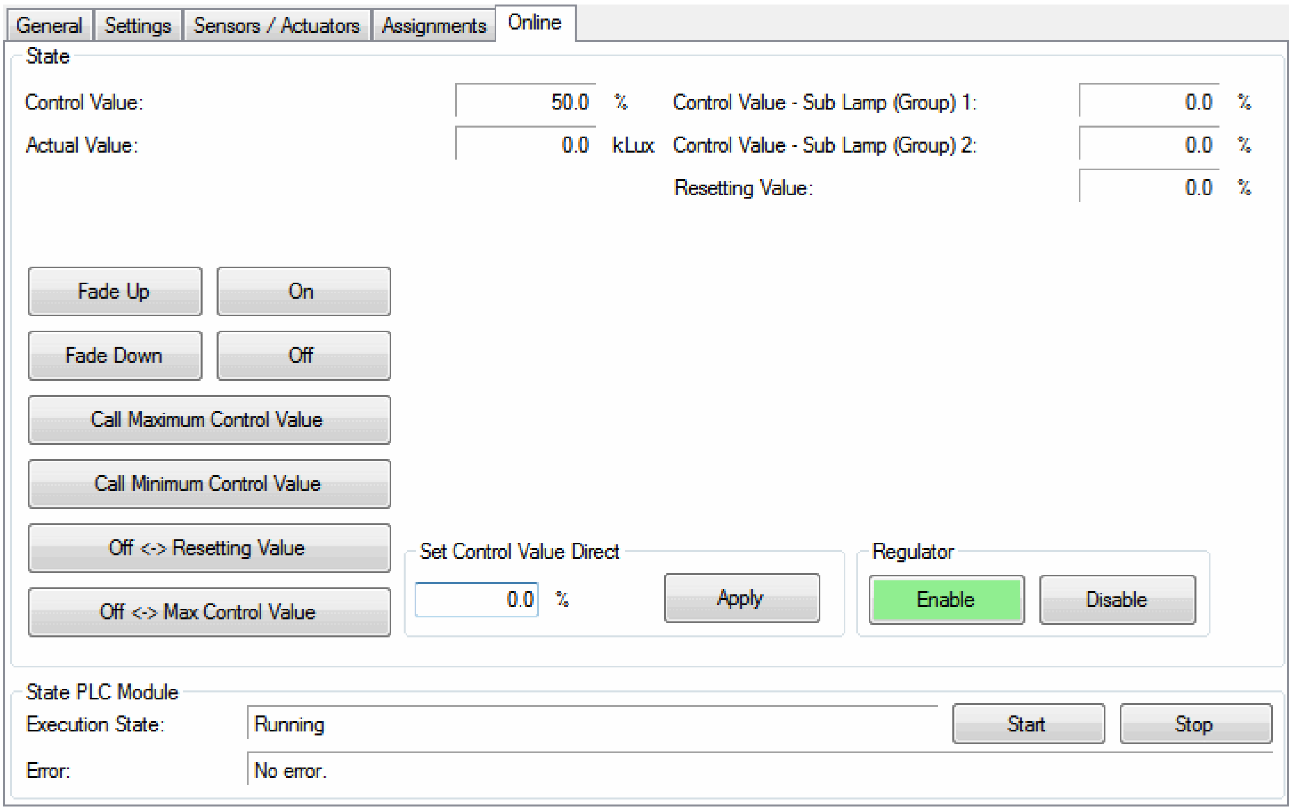
Sensors / Actuators

Lamp (Group):	Type 1 - standard Lamp Group 1
Sub Lamp (Group) 1:	Type 1 - standard Lamp Group 2
Sub Lamp (Group) 2:	Type 1 - standard Lamp Group 3
Weather Station:	Type 4 – Weather station with analog / digital signals 1
Digital Signal (Group) - Dimm Up:	Type 1 - standard Digital Signal Group 1
Digital Signal (Group) - Dimm Down:	Type 1 - standard Digital Signal Group 2
Digital Signal (Group) - Dimm Up/Down:	Type 1 - standard Digital Signal Group 3
Digital Signal (Group) - Motion Detection:	Type 1 - standard Digital Signal Group 4

In this menu you enter the lamp / lamp group and the digital signals for switching the correspondent lamp / lamp group.
In general there are two different procedures to control the lamp / lamp group: with two pushbuttons for dimming up and down or with one pushbutton which alternately dims up and down. A combination of both procedures is possible.

Additionally a motion detector can be used to activate the light control: a rising edge at *Digital Signal (Group) - Motion Detection* activates the light control, a falling edge deactivates the light control and switches it off.

Online



Control Value: 50.0 %

Actual Value: 0.0 kLux

Control Value - Sub Lamp (Group) 1: 0.0 %

Control Value - Sub Lamp (Group) 2: 0.0 %

Resetting Value: 0.0 %

Set Control Value Direct: 0.0 %

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light value of all assigned lamps by their entered fade-rate. The command will only work, when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the assigned lamps either to the maximum or the *Resetting Value*, which depends on the fact, if the *Resetting Mode* is activated or not. The *Resetting Mode* can be activated for each lamp separately.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off state will be stored.
- **Call Maximum Control Value:** Each lamp will be turned to its maximum level.
- **Call Minimum Control Value:** Each lamp will be turned to its minimum level.
- **Off ↔ Resetting Value:** The switching direction depends on the average value of the lamp (group). If the value is greater than 0 all lamps will be turned off. When the value is equal to 0 all lamps will be either turned to their *Resetting Value* or to their maximum value, when the *Resetting Mode* is not activated at the lamp.
- **Off ↔ Max Control Value:** The same as *Off ↔ Resetting Value* but all lamps will be toggled between off and their maximum value.
- **Set Control Value Direct:** The entered value will be taken over to the lamps, when the *Apply*-button is pressed. If the entered value is greater than the maximum value of one of the lamps, the lamp will be set to its maximum level. If it is lower than its minimum level, but not 0, the lamp will be set to its minimum level. Entering and applying a 0 will turn the lamps off.
- **Regulator Enable/Disable:** Sets the twilight-switch in automatic/manual mode. **When the PLC is started, the regulator is disabled.**

All Input functions are only possible, when the regulator is disabled, except the control *Enable/Disable*-functions. In addition, it is possible to disable the active control and turn off the lamps with the *Off*-button. This functionality can be used for e.g., presentations, when a conference room must be dark. The next rising-edge of the motion detector, however, will reactivate the controller, so the delay off-time of the digital signal (group)(motion detection), has to be big enough to ensure secure presence detection over a long time.

State

- **Control Value:** Average value of the active (non-erroneous) lamp/lamps of of the main group.
- **Control Value - Sub Lamp (Group) 1:** Average control value of sub lamp (group) 1 in percentage.
- **Control Value - Sub Lamp (Group) 2:** Average control value of sub lamp (group) 2 in percentage.
- **Actual Value:** Measured light level.
- **Resetting Value:** When the functional unit is turned off, the previous light level will be stored in this variable. If you have selected the entry "Toggle via On/Off command of the Functional Unit" as "Toggle Mode", the lamps in the functional unit are set to this value, when they are switched on with the command *Off* ↔ *Resetting Value*.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.7.1.5 Constant Light Regulation

The Constant Light Regulation is a closed-loop controller regulating the light value of three lamps/lamp groups: a main lamp (group) and two sub lamps/lamp groups. The actual value for calculating the setpoint-deviation is given by an analog sensor (group). An additional motion detector activates/deactivates the control-system automatically. As fast changing light levels are not desired for a light control, the system is dampened by a control delay (dead-time), which becomes shorter, the higher the deviation is. Moreover, a control hysteresis is implemented to prevent the system from swinging around the setpoint value. The value of the hysteresis can be entered in the *settings*-menu, as well as the *Maximum Control Delay*. The following two diagrams show the behavior of the regulation. Please note, that the actual light level is influenced by the (outside-) daylight, as well as by the own inside-light. Without being influenced by the own light, the whole regulation would be a simple open-loop controller. However, this is the reason, why the diagrams show the actual level following the control value. The step size is constant and set to 0,39% which is equal to 1 DALI-Step.

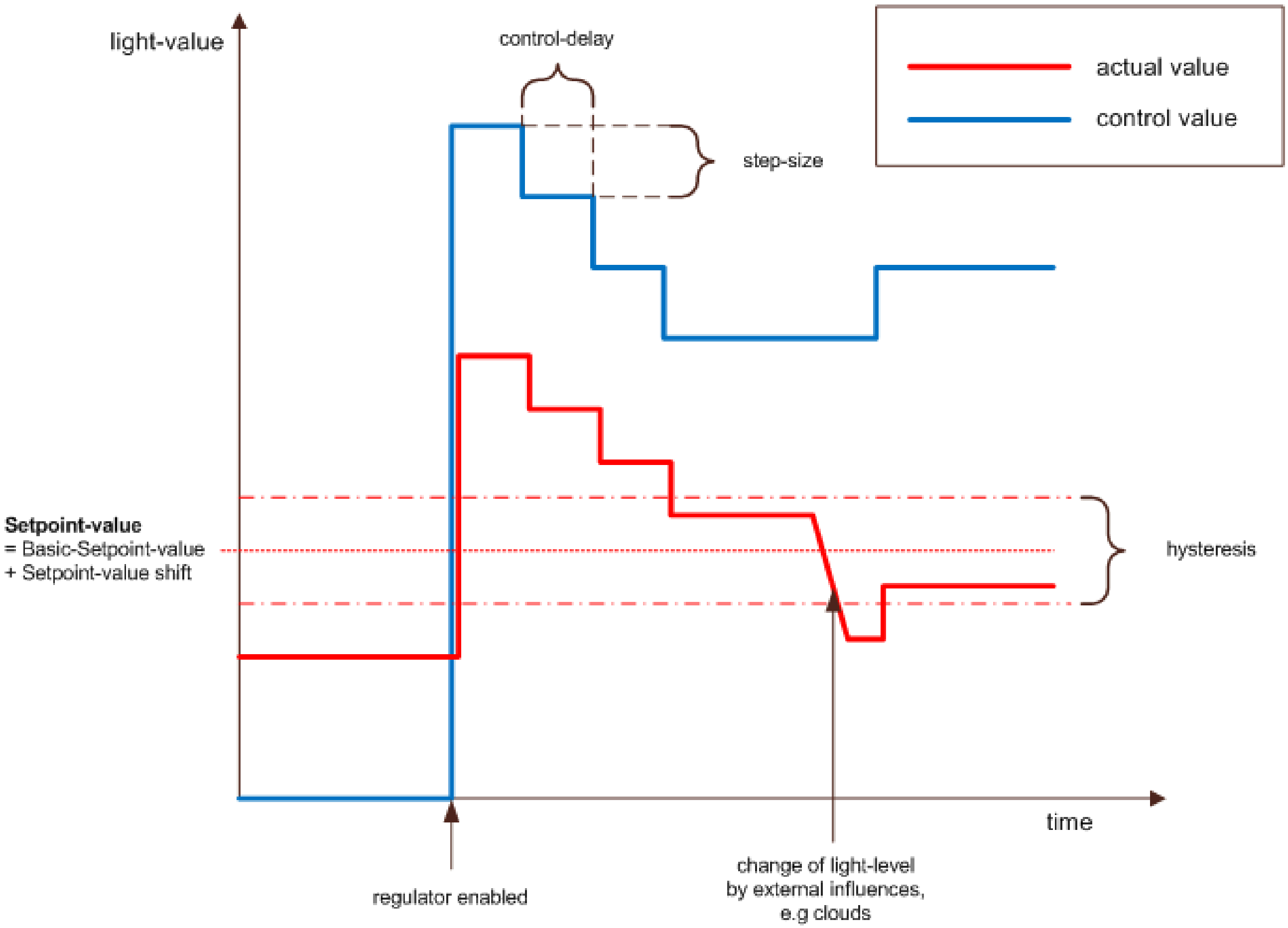


Fig. 1: Hysteresis is big enough to prevent the control-system from oscillating.

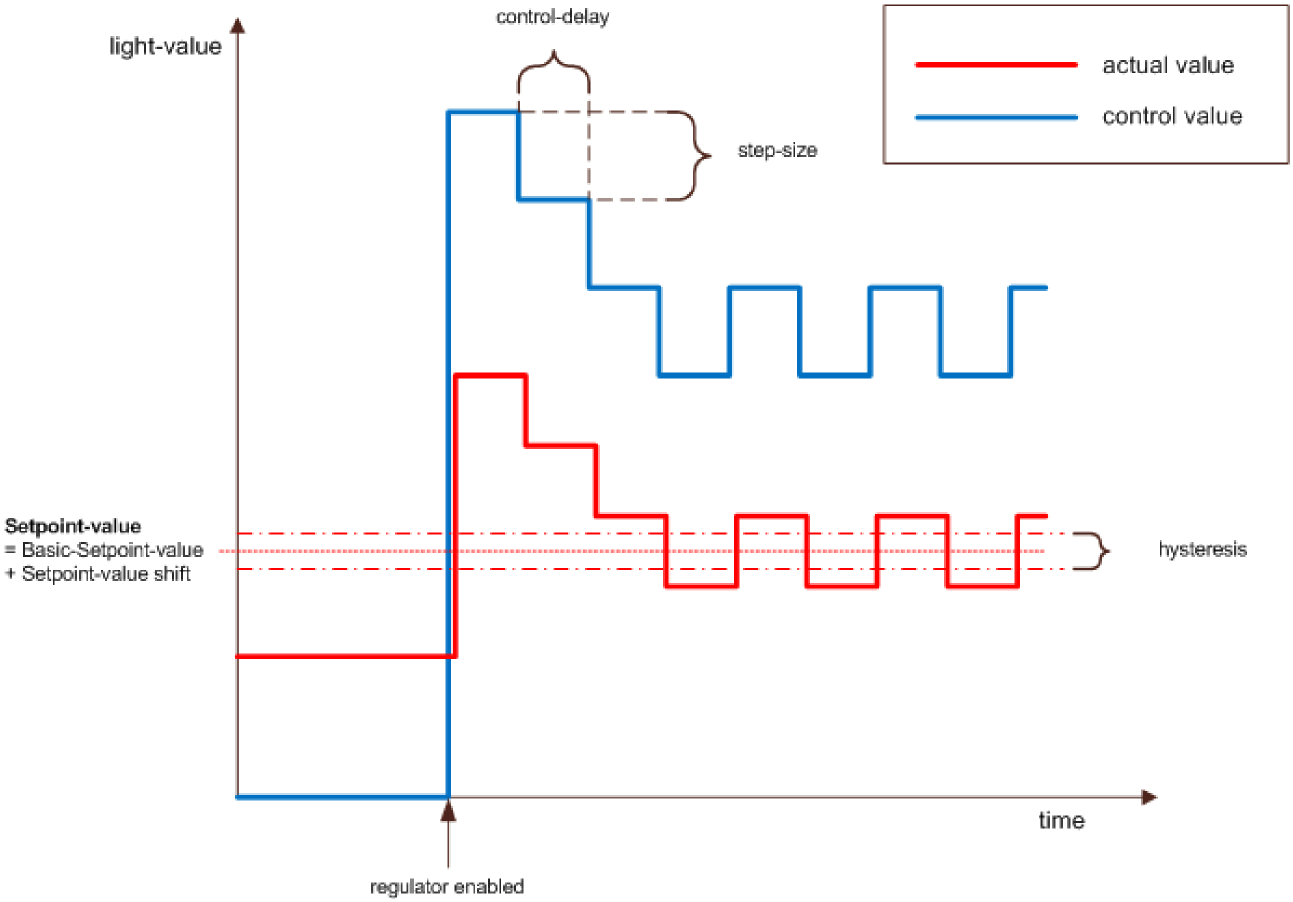
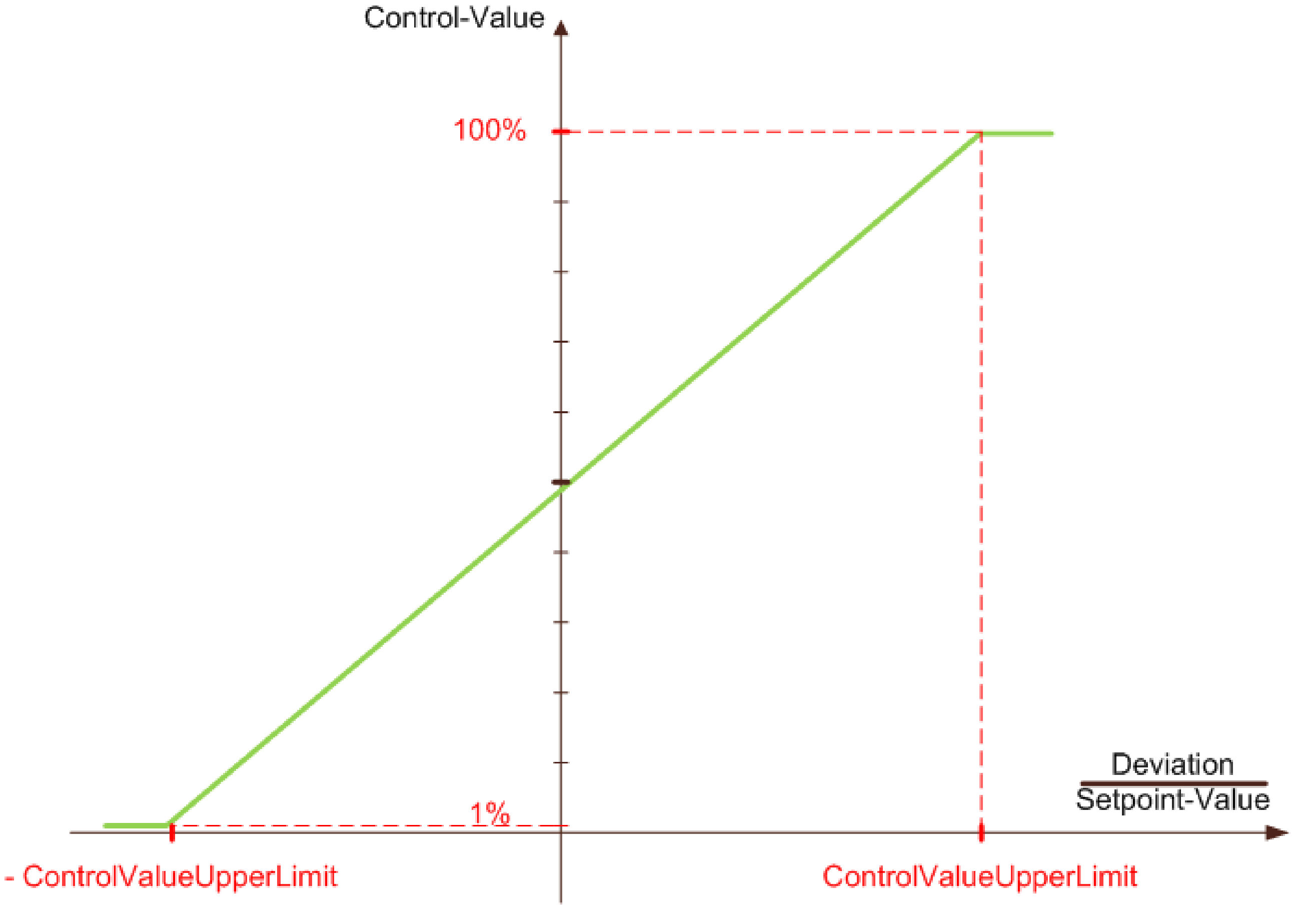


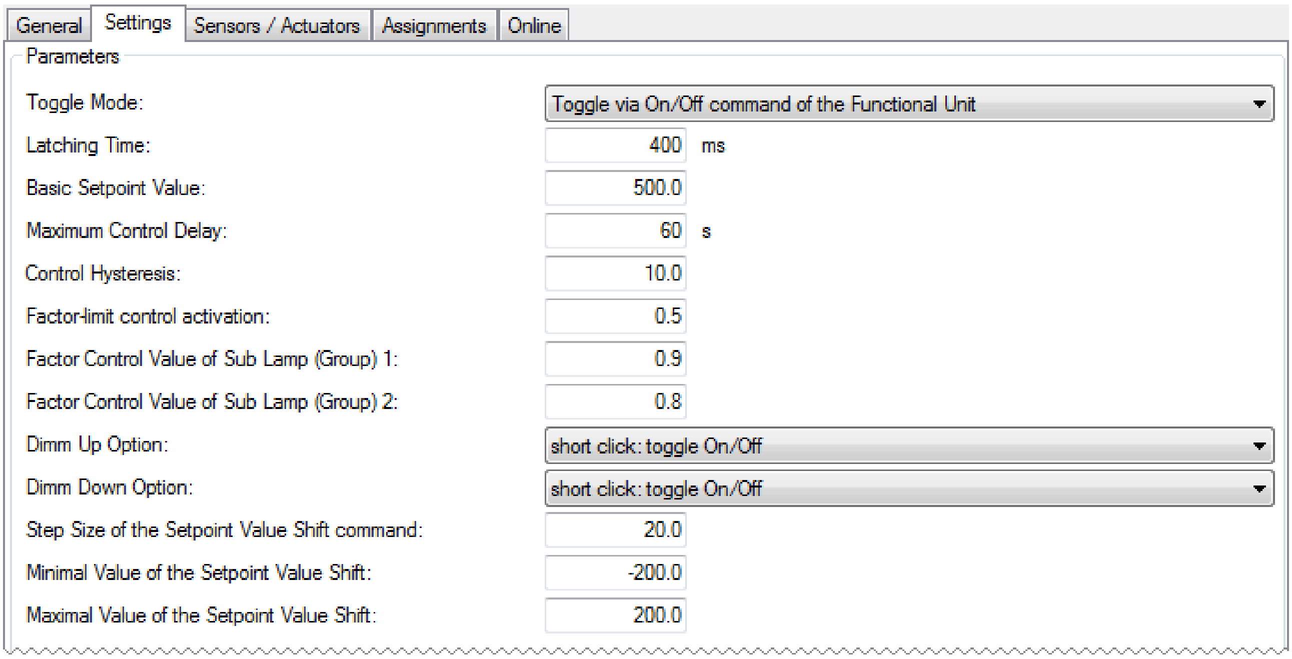
Fig. 2: Hysteresis is too small. The system begins to swing around the setpoint value.

Initial Value

Since the functional unit is a closed control loop, the resulting manipulated variable is continuously recalculated and cannot be determined in only one PLC cycle. It is therefore necessary to start with a value close to the resulting output variable.
The decisive factor for this calculation is the ratio of system deviation to setpoint when the control is activated. If this value is greater than or equal to the parameter Factor limit activation control, then the manipulated variable is set to 100%. If the value is less than or equal to the parameter Factor limit activation control, then the value for the manipulated variable is calculated from the resulting linear equation:



Settings

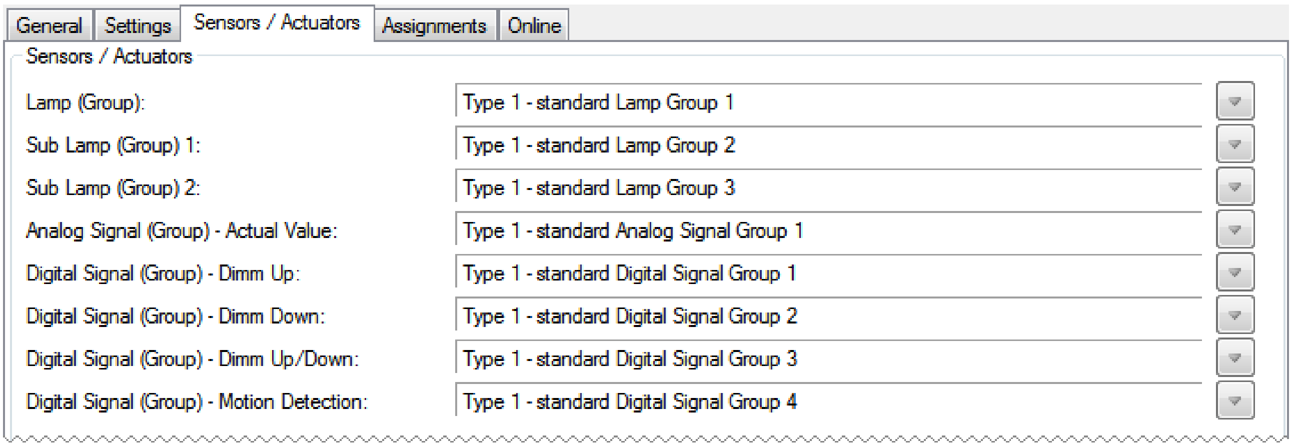


Toggle Mode:	Toggle via On/Off command of the Functional Unit
Latching Time:	400 ms
Basic Setpoint Value:	500.0
Maximum Control Delay:	60 s
Control Hysteresis:	10.0
Factor-limit control activation:	0.5
Factor Control Value of Sub Lamp (Group) 1:	0.9
Factor Control Value of Sub Lamp (Group) 2:	0.8
Dimm Up Option:	short click: toggle On/Off
Dimm Down Option:	short click: toggle On/Off
Step Size of the Setpoint Value Shift command:	20.0
Minimal Value of the Setpoint Value Shift:	-200.0
Maximal Value of the Setpoint Value Shift:	200.0

Parameters

- **Toggle Mode:** If you use the Toggle Mode via "On / Off command of the functional unit", the lamp / lamp group toggles between the resetting value of the functional unit and "off".
  If you use the Toggle Mode via "On / Off command of the local value", the lamp / lamp group toggles between the resetting value(s) of the lamp(s) and "off".
- **Latching Time:** Time distinguishing between short and long-click of a pushbutton.
- **Basic Setpoint Value:** Setpoint value of the *Constant Light Regulation*. Can be adjusted by an offset, the *Setpoint Value Shift*, see *Online*-menu.
- **Maximum Control Delay:** Maximum delay time before a new control value is taken over. See introduction.
- **Control Hysteresis:** Hysteresis-band around the setpoint (1/2 hysteresis above and 1/2 hysteresis below). An actual value outside this band will cause the regulation to calculate new control values.
- **Control On Upper Limit:** See introduction.
- **Factor Control Value of Sub Lamp (Group) 1 / Factor Control Value of Sub Lamp (Group) 2:** Amplification-factors of the sub lamps/lamp groups relating to the main lamp (group).
- **Dimm Up Option:** This describes what the assigned lamp (group) will do, if a short-click of *Digital Signal (Group) - Dimm Up* was detected.
- **Dimm Down Option:** This describes what the assigned lamp (group) will do, if a short-click of *Digital Signal (Group) - Dimm Down* was detected.
- **Step Size of the Switching Value Shift command:** Refers to the *Online*-menu, where the *Switching Value Shift* can be increased/decreased by step *Up*/*Down*-buttons. This defines the step size.
- **Minimal / Maximal Value of the Setpoint Value Shift:** The max/min value of *Setpoint Value Shift*.

Sensors / Actuators



Sensors / Actuators

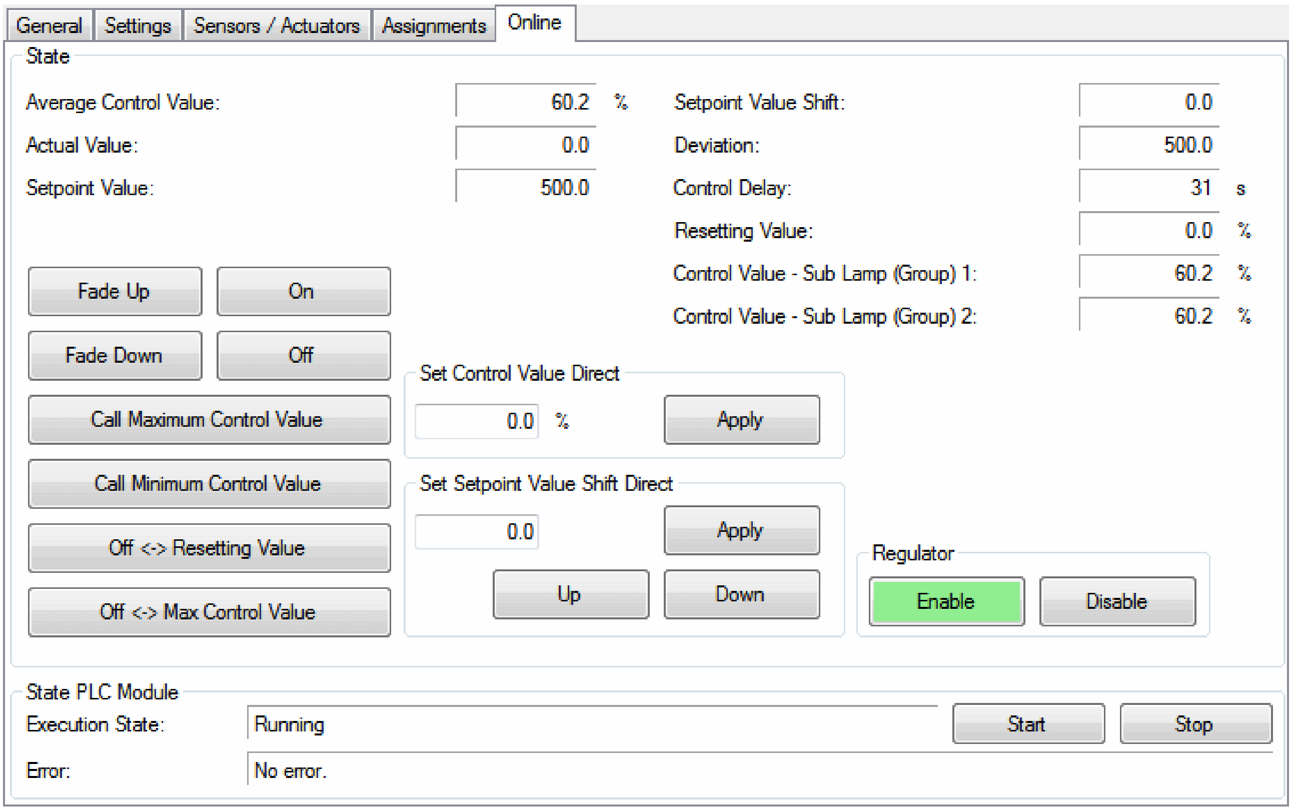
Lamp (Group):	Type 1 - standard Lamp Group 1
Sub Lamp (Group) 1:	Type 1 - standard Lamp Group 2
Sub Lamp (Group) 2:	Type 1 - standard Lamp Group 3
Analog Signal (Group) - Actual Value:	Type 1 - standard Analog Signal Group 1
Digital Signal (Group) - Dimm Up:	Type 1 - standard Digital Signal Group 1
Digital Signal (Group) - Dimm Down:	Type 1 - standard Digital Signal Group 2
Digital Signal (Group) - Dimm Up/Down:	Type 1 - standard Digital Signal Group 3
Digital Signal (Group) - Motion Detection:	Type 1 - standard Digital Signal Group 4

In this menu you enter the lamp / lamp group and the digital signals for switching the correspondent lamp / lamp group.
In general there are two different procedures to control the lamp / lamp group: with two pushbuttons for dimming up and down or with one pushbutton which alternately dims up and down. A combination of both procedures is possible.

Additionally, a Motion Detection can be used to enable the light control: a rising edge at *digital signal (group) - Motion Detection* enables the Controller, a falling edge disables and switches off the controller.

Dimming up and down during an active control has a special effect: The light value increases or decreases and after 5 seconds without activating a dimming function, the current light value is set as the resulting setpoint (basis setpoint + setpoint shift) so that the control deviation is then 0. Before this new setpoint value is accepted, a small correction to the light value can be noticed. By dimming with a dimming command, it is possible that all lamps/lamp groups are changed by the same value and that a correction for the gain factors must be carried out by the main lamp/lamp group ↔ underlamp/lamp group.

Online



State

Average Control Value:	60.2	%	Setpoint Value Shift:	0.0	
Actual Value:	0.0		Deviation:	500.0	
Setpoint Value:	500.0		Control Delay:	31	s
			Resetting Value:	0.0	%
			Control Value - Sub Lamp (Group) 1:	60.2	%
			Control Value - Sub Lamp (Group) 2:	60.2	%

Set Control Value Direct: 0.0 %

Set Setpoint Value Shift Direct: 0.0

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Fade Up:** This command will increase the light value of all assigned lamps by their entered fade-rate. The command will only work when the lamp is already turned on. Increasing the value above the maximum level is not possible.
- **Fade Down:** This command decreases the light value by the entered fade-rate but will not let a lamp turn off completely.
- **On:** Turns the assigned lamps either to the maximum or the *Resetting Value*, which depends on the fact, if the *Resetting Mode* is activated or not. The *Resetting Mode* can be activated for each lamp separately.
- **Off:** Turns the assigned lamps off. When the *Resetting Mode* is activated the light value prior to the off state will be stored.
- **Call Maximum Control Value:** Each lamp will be turned to its maximum level.
- **Call Minimum Control Value:** Each lamp will be turned to its minimum level.
- **Off ↔ Resetting Value:** The switching direction depends on the average value of the lamp (group). If the value is greater than 0 all lamps will be turned off. When the value is equal to 0 all lamps will be either turned to their *Resetting Value* or to their maximum value, when the *Resetting Mode* is not activated at the lamp.
- **Off ↔ Max Control Value:** The same as Off ↔ Resetting Value but all lamps will be toggled between off and their maximum value.
- **Set Control Value Direct:** The entered value will be taken over to the lamps, when the *Apply*-button is pressed. If the entered value is greater than the maximum value of one of the lamps, the lamp will be set to its maximum level. If it is lower than its minimum level, but not 0, the lamp will be set to its minimum level. Entering and applying a 0 will turn the lamps off.
- **Set Setpoint Value Shift Direct:** Field to enter the *Setpoint Value Shift* either directly in a textbox or stepping up and down with the entered step size (see *Settings*-menu). The value is, however, delimited by the maximum and minimum value in the *Settings*-menu.
- **Regulator Enable/Disable:** Sets the twilight-switch in automatic/manual mode. **When the PLC is started, the regulator is disabled.**

All Input functions are only possible, when the regulator is disabled, except the control *Enable/Disable*-functions. In addition, it is possible to disable the active control and turn off the lamps with the *Off*-button. This functionality can be used for e.g., presentations, when a conference room must be dark. The next rising-edge of the motion detector, however, will reactivate the controller, so the delay off-time of the digital signal (group)(motion detection), has to be big enough to ensure secure presence detection over a long time.

State

- **Control Value:** Average value of the active (non-erroneous) lamp/lamps of of the main group.
- **Actual Value:** Measured light level.
- **Setpoint Value:** Actual setpoint value resulting from the *Basic Setpoint Value* and the *Setpoint Value Shift*.
- **Setpoint Value Shift:** Actual setpoint value shift.
- **Deviation:** The deviation in seconds.
- **Control Delay:** Displays the control delay in seconds.
- **Resetting Value:** When the functional unit is turned off, the previous light level will be stored in this variable. If you have selected the entry "Toggle via On/Off command of the Functional Unit" as "Toggle Mode", the lamps in the functional unit are set to this value, when they are switched on with the command Off ↔ Resetting Value and the Resetting Mode of the lamp is deactivated.
- **Control Value - Sub Lamp (Group) 1:** Average control value of sub lamp (group) 1 in percentage.
- **Control Value - Sub Lamp (Group) 2:** Average control value of sub lamp (group) 2 in percentage.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.7.2 Sunblinds

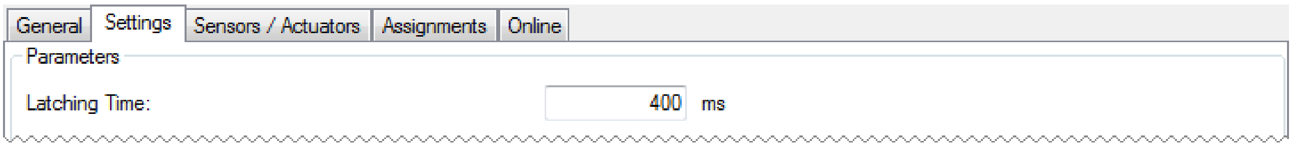
The following types of Sunblinds are available:

- Standard Sunblind [▶ 130]
- Shading as a function of the position of the sun [▶ 131]

7.2.2.7.2.1 Standard Sunblind

The Standard Sunblind - controls a blind-drive/blind-drive group.

Settings



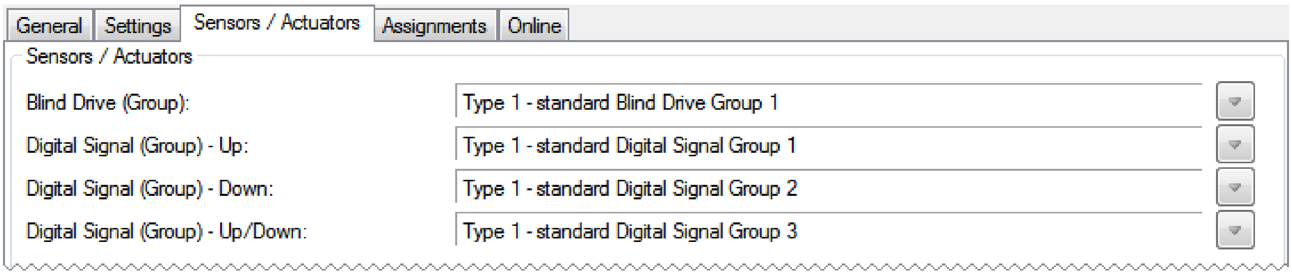
Parameters

Latching Time: 400 ms

Parameters

- **Latching Time:** Time until the commands given by the sensors/sensor groups (see *Sensors/Actuators*) are maintained.

Sensors / Actuators



Sensors / Actuators

Blind Drive (Group): Type 1 - standard Blind Drive Group 1

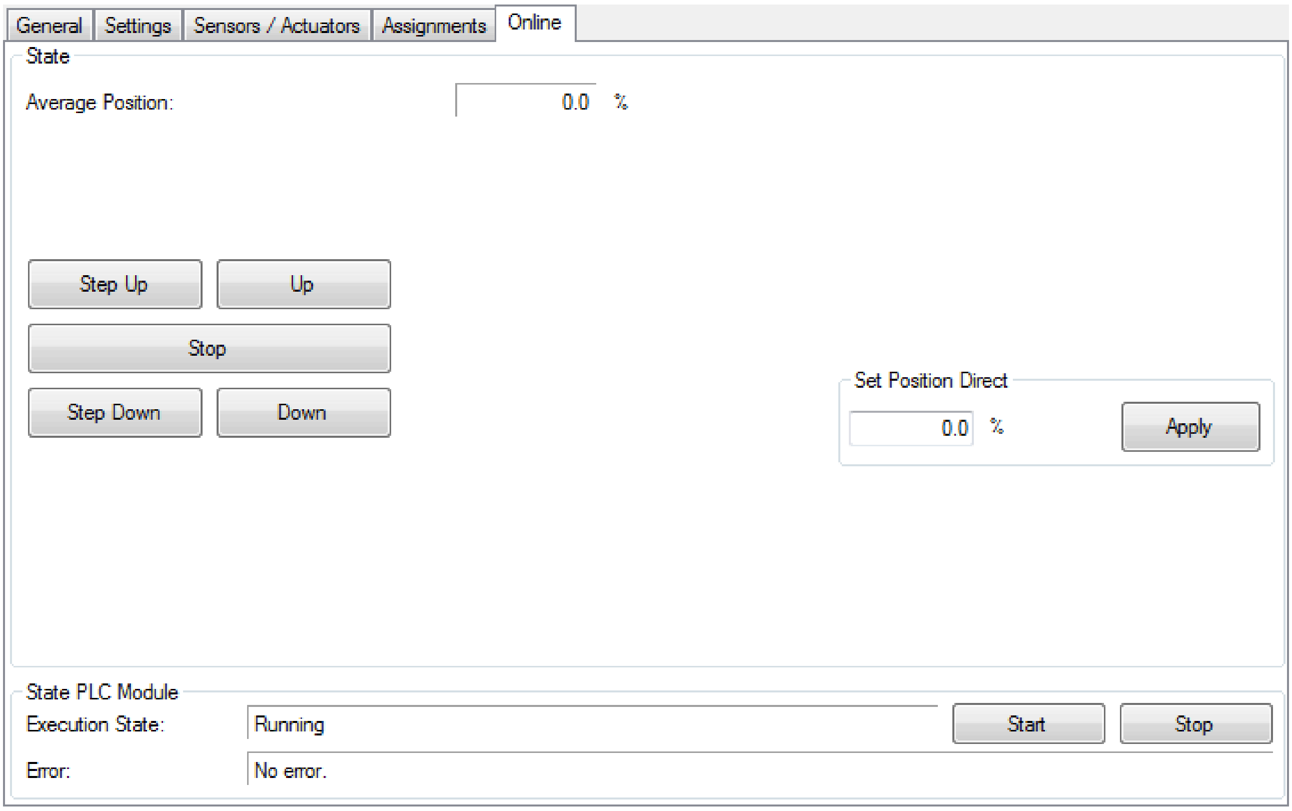
Digital Signal (Group) - Up: Type 1 - standard Digital Signal Group 1

Digital Signal (Group) - Down: Type 1 - standard Digital Signal Group 2

Digital Signal (Group) - Up/Down: Type 1 - standard Digital Signal Group 3

In this menu you enter the Blind Drive / Blind Drive Group and the Digital Signal / Digital Signal Group with the pushbuttons to move up and down.
The usually push button-controlled digital signal moves the blind(s) up and down. The exit is immediately locked. When you work with only one pushbutton for up and down, the Digital Signal (Group) - Up/Down moves the blind in the set direction, with the next push it stops the sunblind, with the next it moves it in the opposite direction, with the next it stops it, and so on.

Online



State

Average Position: 0.0 %

Set Position Direct

0.0 %

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Step Up/Step Down:** Stepping the assigned blinds up/down. Standard blinds (type 1) will be stepped by an adjustable time. Blinds connected with KL2532 will be stepped by the entered *Step Range*.
- **Up/Down:** All assigned drives will be driven up/down. Blinds connected with KL2532 will be moved with rapid velocity. The rapid velocity is to be entered in the *Dynamics*-menu of the blind-drive.
- **Stop:** Stops the blind.
- **Set Position Direct:** The assigned blinds will be driven to a specific position. This position must be given in shadowing-percentage and lamella-angle.

State

- **Average Position:** Average position of the blind-drive(s). 100% means completely closed, 0% completely opened.

State PLC Module

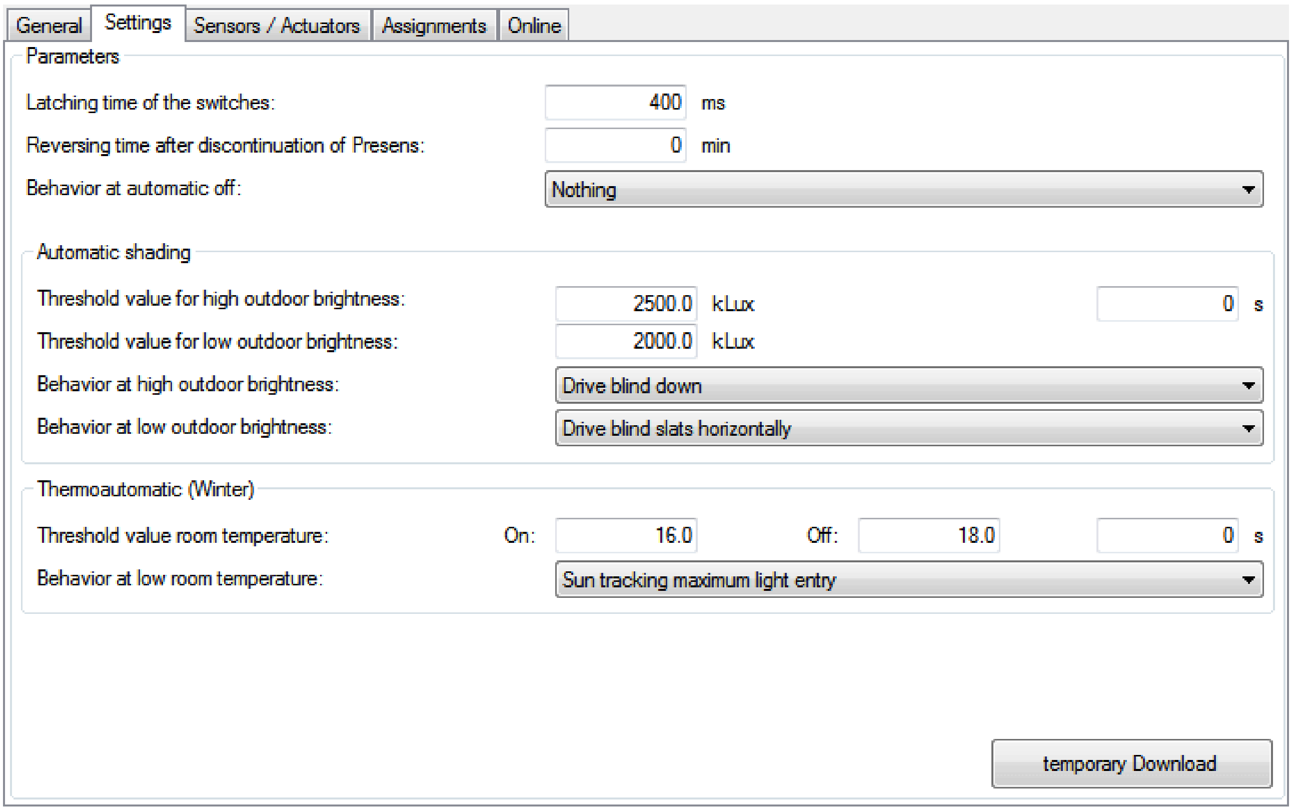
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.7.2.2 Shading as a function of the position of the sun

The blinds are interactively regulated depending on the incidence of light and the position of the sun. This is calculated such that use-optimized room lighting is always achieved. Distinction is made between summer and winter operation based on the outside temperature. The consistent use of solar power reduces heating energy in winter. In summer the cooling load is reduced by optimal shading without completely darkening the room. Optionally the energy balance can be further improved by a motion detector and by a temperature sensor in the room.

Special blind drives (Blind Drive connected with KL2532 [▶ 88]) are required in order to utilise this function unit.

Settings



Parameters

Latching time of the switches: 400 ms

Reversing time after discontinuation of Presens: 0 min

Behavior at automatic off: Nothing

Automatic shading

Threshold value for high outdoor brightness: 2500.0 kLux 0 s

Threshold value for low outdoor brightness: 2000.0 kLux

Behavior at high outdoor brightness: Drive blind down

Behavior at low outdoor brightness: Drive blind slats horizontally

Thermoautomatic (Winter)

Threshold value room temperature: On: 16.0 Off: 18.0 0 s

Behavior at low room temperature: Sun tracking maximum light entry

Commands

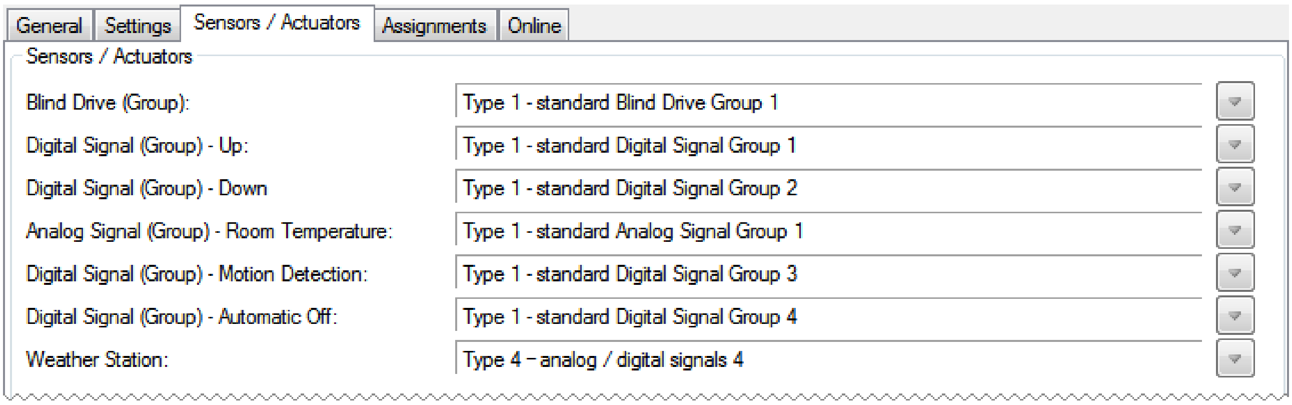
- **temporary Download:** If you want to change parameters to test the blind, then a normal download of parameters (activate configuration) would take a relativ long time. To make work easier and save time, therefore exists the possibility of a temporary download. Here, it is not necessary to stop the PLC modules to download the parameters. The parameters of the blind will remain temporary and not persistent. These changes will discard after a PLC restart or activating the configuration.

Parameters

- **Latching time of the switches:** Time until the commands given by the sensors/sensor groups (see *Sensors/Actuators*) are maintained.
- **Reversing time after discontinuation of Presens:** This parameter is only valid, if a motion detector is connected. If a pushbutton is pressed, the automatic mode of the blind is stopped. Is the motion detector inactive (the room is empty), a reversing time with the here entered duration starts. When the time has expired, the automatic mode is switched on again.
- **Behavior at automatic off:** Function, which shall be executed, when automatic-mode is turned off.
  - **Nothing**
  - **Drive blind slats horizontally**
  - **Drive blind up**
  - **Close blind drive**
- **Threshold value for high outdoor brightness:** If this threshold is exceeded, the action defined under *Behavior at high outdoor brightness* (see below) will be executed.
- **Threshold value for low outdoor brightness:** If this threshold is undershot, the action defined under *Behavior at low outdoor brightness* (see below) will be executed.
- **Behavior at high outdoor brightness:** The following actions can be selected to be executed at high brightness:
  - **Drive blind down**
  - **Sun tracking maximum shading** (Cut off [▸ 94])
- **Behavior at low outdoor brightness:** One of these functions can be chosen for low brightness:
  - **Drive blind slats horizontally**

  - **Drive blind up**
  - **Sun tracking maximum light entry:** The lamellas will be set to an angle letting pass the maximum of the sunlight.
  - **Sun tracking maximum shading** (Cut off [▶ 94])
- **Threshold value for room temperature:**
  - **On :** Exceeding this temperature the action selected under *Behavior at low room temperature* (see below) will be deactivated.
  - **Off :** The action selected under *Behavior at low room temperature* (see below) will be executed by undershooting the entered temperature.
- **Behavior at low room temperature:** Here the function can be selected, which shall be carried out, when the room temperature falls below the above-mentioned threshold (only in winter operation mode).
  - **Drive blind slats horizontally**
  - **Drive blind up**
  - **Sun tracking maximum light entry:** The lamellas will be set to an angle letting pass the maximum of the sunlight.
  - **Sun tracking maximum shading (Cut off [▶ 94]).**

Sensors / Actuators

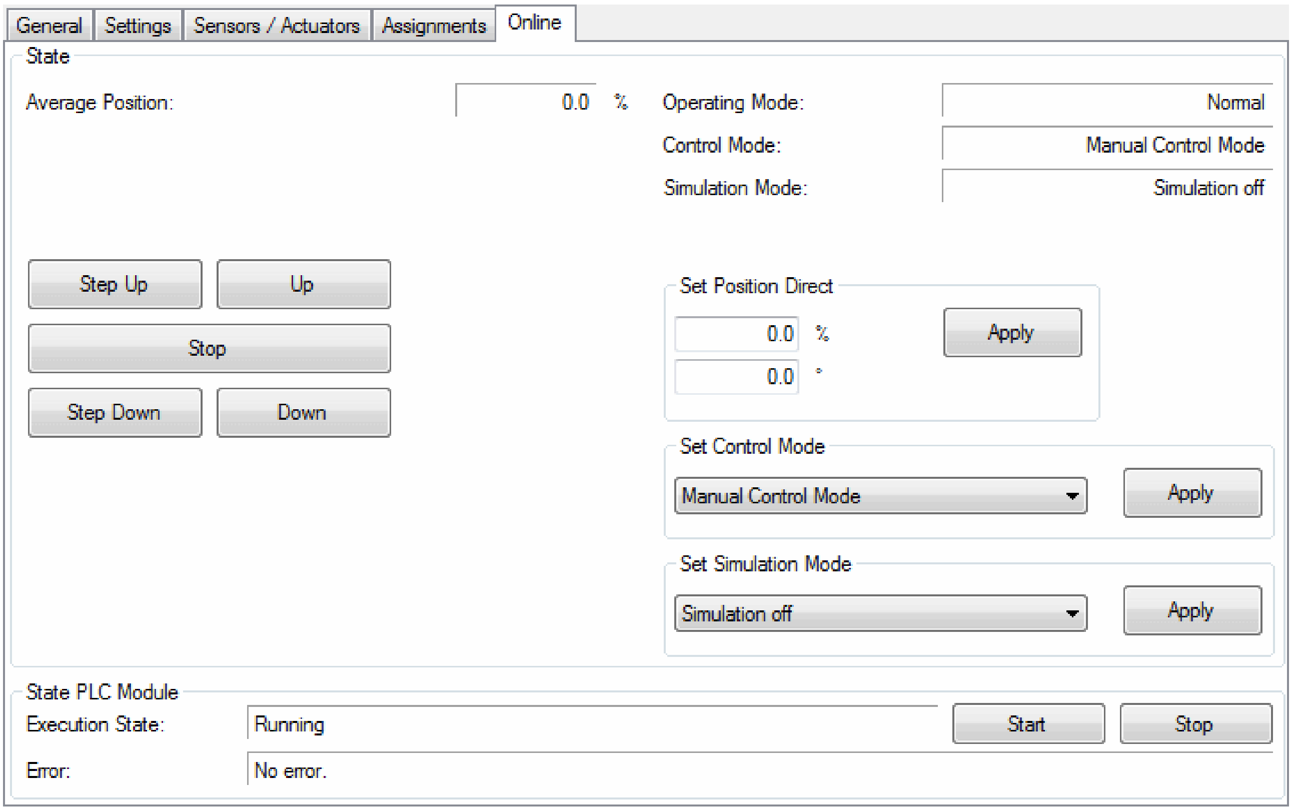


Sensors / Actuators

Blind Drive (Group):	Type 1 - standard Blind Drive Group 1
Digital Signal (Group) - Up:	Type 1 - standard Digital Signal Group 1
Digital Signal (Group) - Down	Type 1 - standard Digital Signal Group 2
Analog Signal (Group) - Room Temperature:	Type 1 - standard Analog Signal Group 1
Digital Signal (Group) - Motion Detection:	Type 1 - standard Digital Signal Group 3
Digital Signal (Group) - Automatic Off:	Type 1 - standard Digital Signal Group 4
Weather Station:	Type 4 – analog / digital signals 4

In this section the actuators and sensors for the functional unit must be entered. To make changes, the *Edit*-button has to be pressed.

Online



Average Position: 0.0 %
Operating Mode: Normal
Control Mode: Manual Control Mode
Simulation Mode: Simulation off
Set Position Direct: 0.0 % 0.0 °
Set Control Mode: Manual Control Mode
Set Simulation Mode: Simulation off
Execution State: Running
Error: No error.

Commands

- **Step Up/Step Down:** Stepping the assigned blinds up/down. Standard Blinds (type 1) will be stepped by an adjustable time. Blinds connected with KL2532 will be stepped by the entered *Step Range*.
- **Up/Down:** All assigned drives will be driven up/down. Blinds connected with KL2532 will be moved with rapid velocity. The rapid velocity is to be entered in the *Dynamics*-menu of the blind-drive.
- **Stop:** Stops the blind.
- **Set Position Direct:** The assigned blinds will be driven to a specific position. This position must be given in shadowing-percentage and lamella-angle.
- **Set Control Mode:** Automatic or manual mode can be chosen here. The selection has to be activated by pressing the *Apply*-button.
- **Set Simulation Mode:** Summer or winter-simulation can be activated, or the simulation mode can be turned off.

State

- **Average Position:** Average position of all assigned blind drives. 100% means completely closed, 0% completely opened.
- **Operating Mode:** Displays the chosen operation mode.
- **Control Mode:** Displays the actual control mode.
- **Simulation Mode:** Shows, which simulation mode is active.
  - **Simulation off**
  - **Simulation Summer operation**
  - **Simulation Winter operation**

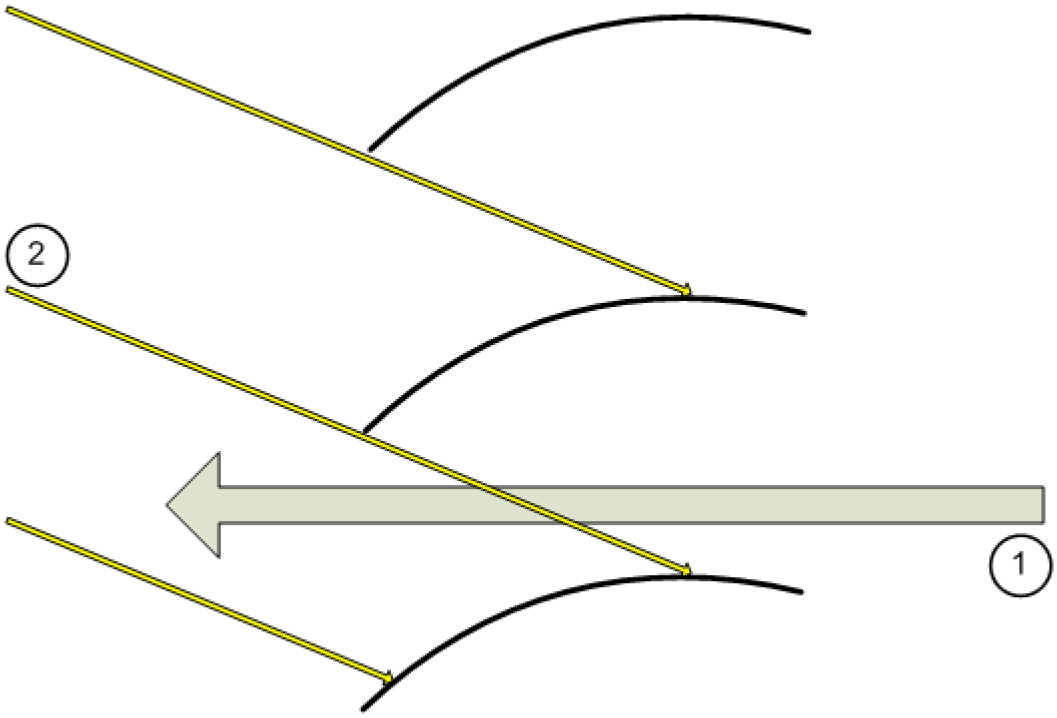
State PLC Module

- **Execution State:** Shows, if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.7.2.2.1 Cut off

Cut off

In Cut off position the sun is not directly visible, but it is still possible to see through the blind and enough daylight can pass.



1. Possibility to see through the blind.

2. Sunbeams, not able to pass.

7.2.2.7.3 HVACs

The following types of HVAC-units (heating, ventilating, and air conditioning) are available:

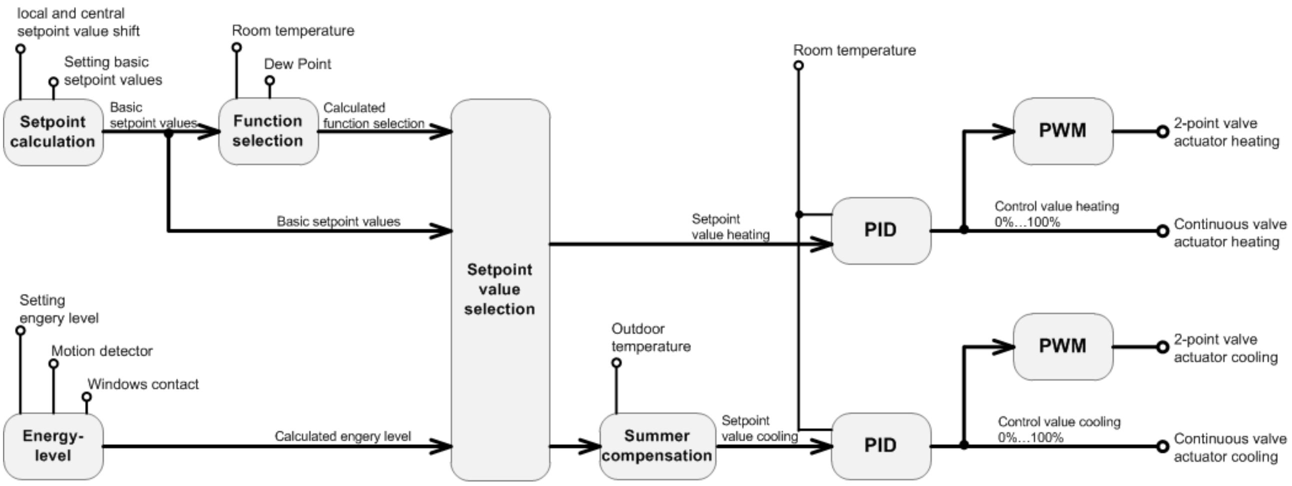
- Heating and Cooling [▶ 135]

7.2.2.7.3.1 Heating and Cooling

The temperature controller can be used to control heating and cooling valves. Both thermal and constant drives are supported.

Principle of operation

The following diagram illustrates the structure of the temperature controller.



Energy level

The energy output can be adapted by the energy level to the use of the building. The longer a building or room is not used, the further the energy level can be lowered.

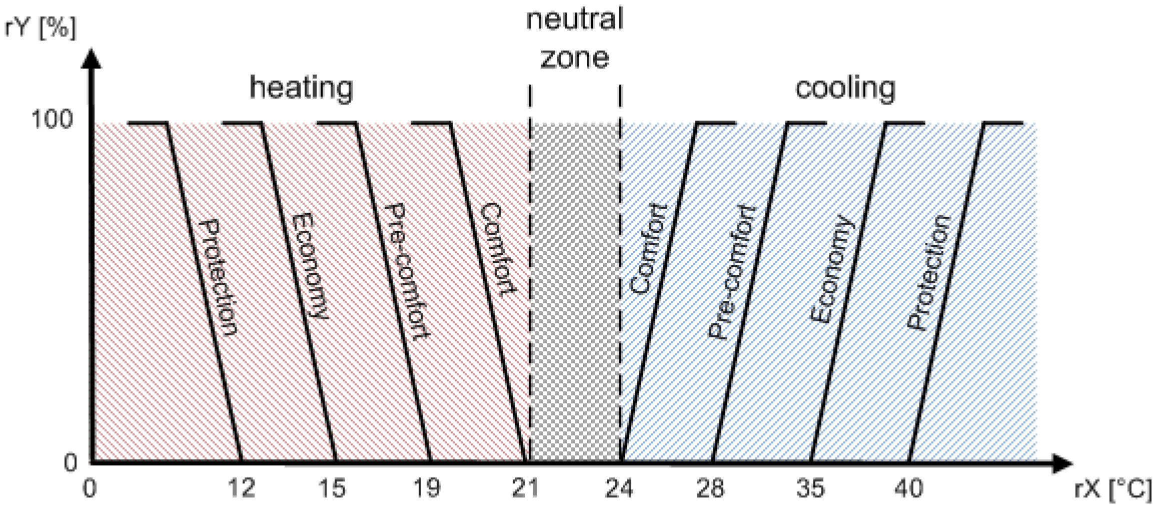
Energy Level	Description
Comfort	If the room is occupied, it is in Comfort mode. Comfort mode can be activated by a timer switch channel or directly by presence recognition.
Pre-comfort	The *Pre-comfort* energy level is for an unused room which, however, can be occupied again shortly.
Economy	The *Economy* energy level is for an unused room that will only be occupied again in a few hours. Economy mode is activated, for example, at night by a timer switch channel.
Protection	This operating mode is activated in the case of long absence times e.g. during works holidays or also when a window is open. The energy level is very low and serves only to protect the building from damage caused by frost or overheating.

The change between the different energy levels is usually affected under time control by means of timer switch channels. Window contacts or motion sensors, which are linked by digital signals to the heating controller, can overwrite this specification. Hence, a window contact always sets the energy level to *Protection*. If a motion sensor is connected to the heating controller, it must be active in order that the *Comfort* energy level can be activated.

The calculated energy level is indicated in the *Online* dialog.

Setpoint value calculation

Each energy level has a base setpoint value for the heating mode and another for the cooling mode. Hence, a total of 8 base setpoint values are specified. The appropriate base setpoint value is considered depending on the energy level (Comfort, Pre-comfort, Economy or Protection) and the control function (heating or cooling).



In addition to the base setpoint values, a central setpoint value shift can be specified as a parameter for both the heating and cooling modes. Furthermore, a local setpoint value shift (user setpoint value shift) can be set as a command. The local setpoint value shift can be changed, for example, by means of an analog signal (potentiometer of a room control unit).

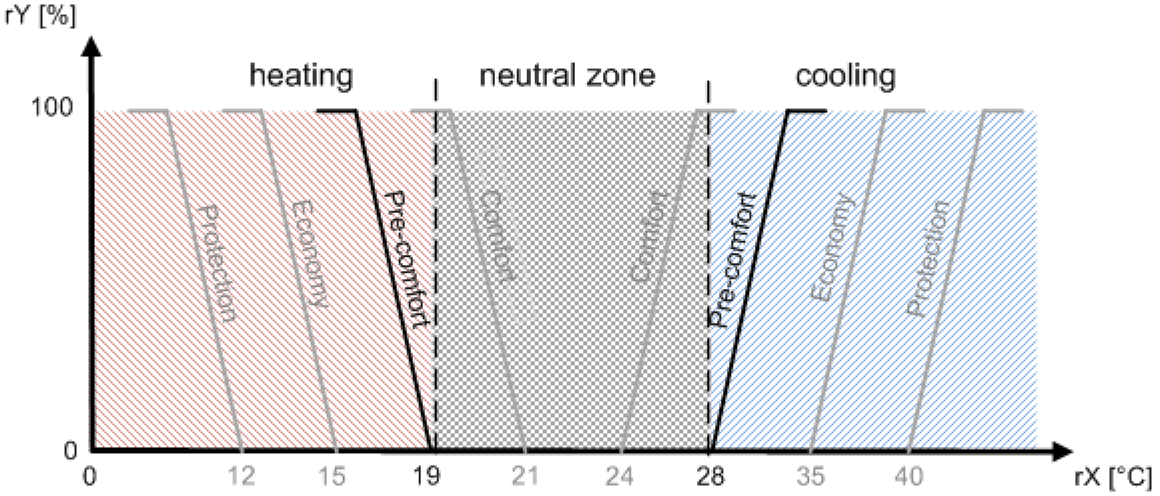
The resulting setpoint value is composed of:

- one of the 8 base setpoint values
- the local setpoint value shift (user setpoint value shift). No effect in the case of the *Protection* energy level
- the central setpoint value shift. No effect in the case of the *Protection* energy level
- the setpoint value shift due to summer compensation.

Function selection (heating/cooling switchover)

The change between heating and cooling mode (control function) takes place automatically based on the difference between the actual and setpoint values (control deviation). The switchover can be delayed with the aid of a run-on time. If dew point sensors are linked via a digital signal group, then the cooling mode is disabled as soon as the digital signal group with the dew point sensors is active. If the actual value lies within the neutral zone, then the controller retains the current control function.

Depending on the energy level the setpoint values for heating and cooling are shifted. The size of the neutral zone also changes as a result. The following diagram shows the characteristic curves with the *Pre-comfort* energy level active:



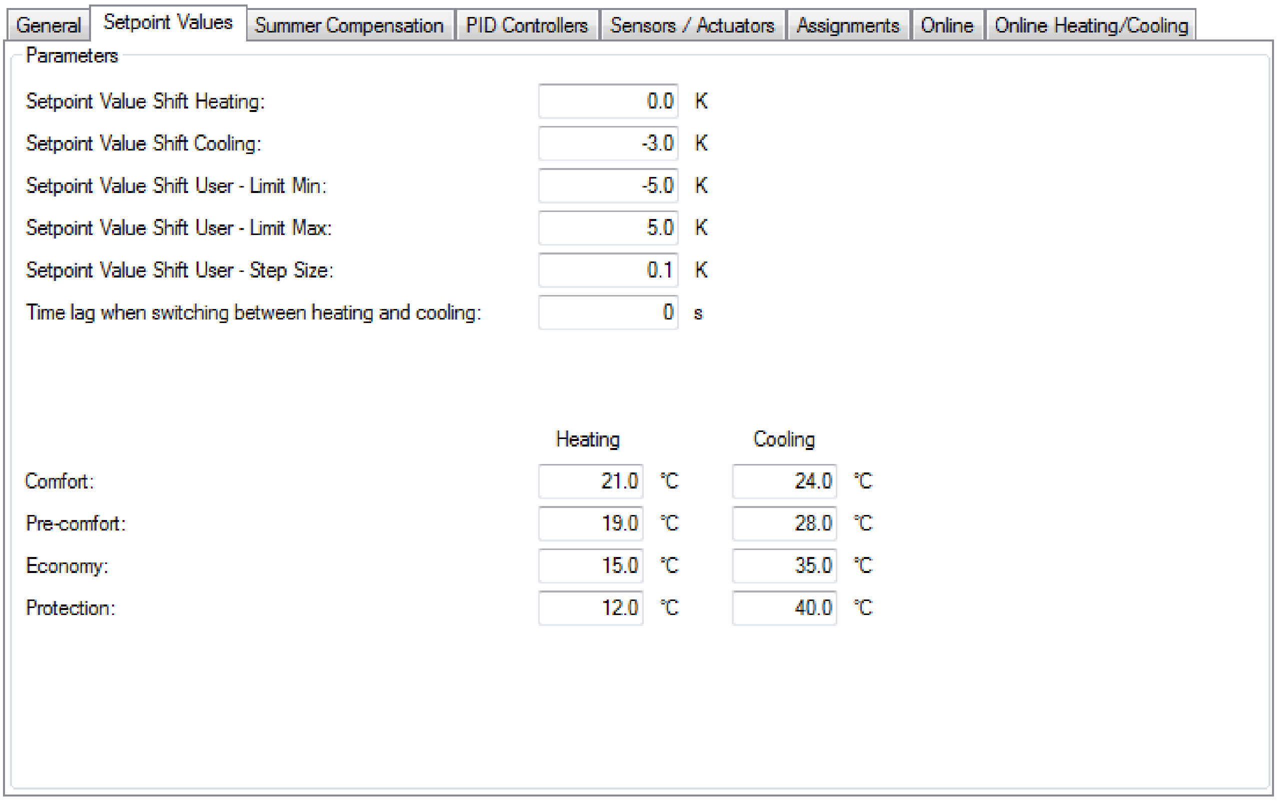
In this example the change from heating mode to cooling mode only takes place at an actual value of 28 °C or higher. If the energy level were to be set to *Comfort*, then the change would already take place at 24 °C. Due to this property the power requirement is reduced whilst at the same time reducing the energy level.

The calculated control function is indicated in the *Online* dialog.

Summer compensation

With the aid of this summer compensation, overly large temperature differences between the outside temperature and the room temperature are avoided in summer. This helps to save energy whilst at the same time increasing comfort. If the outside temperature exceeds an adjustable limit value, then the setpoint value is raised. The factor by which the setpoint value is raised can be set from 0.0 to 10.0.

Settings



Parameters

Setpoint Value Shift Heating:	0.0	K
Setpoint Value Shift Cooling:	-3.0	K
Setpoint Value Shift User - Limit Min:	-5.0	K
Setpoint Value Shift User - Limit Max:	5.0	K
Setpoint Value Shift User - Step Size:	0.1	K
Time lag when switching between heating and cooling:	0	s

	Heating		Cooling	
Comfort:	21.0	°C	24.0	°C
Pre-comfort:	19.0	°C	28.0	°C
Economy:	15.0	°C	35.0	°C
Protection:	12.0	°C	40.0	°C

Parameters

- **Setpoint Value Shift Heating:** central setpoint value shift during heating.
  If the *setpoint value shift for heating* lies in the positive range, then this affects the following setpoint values:

  HeatingCoolingEnergy Level Protection Economy Pre-comfort Comfort Comfort Pre-comfort Economy ProtectionBase Setpoint Value [°C]1215192124283540Setpoint Value Shift [K]---+3+3+3--Setpoint Value [°C]1215192427313540 If the *setpoint value shift for heating* lies in the negative range, then this affects the following setpoint values:

  HeatingCoolingEnergy Level Protection Economy Pre-comfort Comfort Comfort Pre-comfort Economy ProtectionBase Setpoint Value [°C]1215192124283540Setpoint Value Shift [K]---3-3----Setpoint Value [°C]1215161824283540
- **Setpoint Value Shift Cooling:** central setpoint value shift during cooling.
  If the *setpoint value shift for cooling* lies in the positive range, then this affects the following setpoint values:

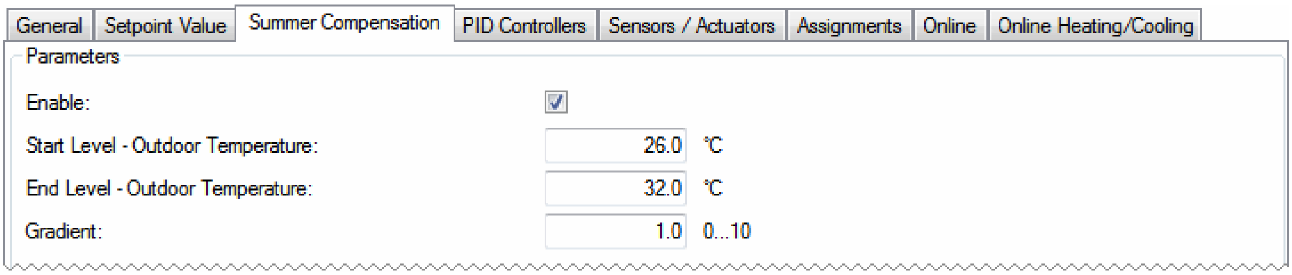
  HeatingCoolingEnergy Level Protection Economy Pre-comfort Comfort Comfort Pre-comfort Economy ProtectionBase Setpoint Value [°C]1215192124283540Setpoint Value Shift [K]----+3+3--Setpoint Value [°C]1215192127313540 If the *setpoint value shift for cooling* lies in the negative range, then this affects the following setpoint values:

  HeatingCoolingEnergy Level Protection Economy Pre-comfort Comfort Comfort Pre-comfort Economy ProtectionBase Setpoint Value [°C]1215192124283540Setpoint Value Shift [K]-----3---Setpoint Value [°C]1215192121283540 If the setpoint value for *Pre-comfort* is shifted beyond *Economy*, then the setpoint value for *Economy* is shifted with it.

HeatingCoolingEnergy Level Protection Economy Pre-comfort Comfort Comfort Pre-comfort Economy ProtectionBase Setpoint Value [°C]1215192124283540Setpoint Value Shift [K]----+8+8--Setpoint Value [°C]1215192124283540

- **Setpoint Value Shift User - Limit Min:** lower limit for the local setpoint value shift.
- **Setpoint Value Shift User - Limit Max:** upper limit for the local setpoint value shift.
- **Setpoint Value Shift User - Step Size:** step size for the local setpoint value shift.
- **Time lag when switching between heating and cooling:** time delay before switching between heating and cooling (control function).
- C**omfort:** If the *Comfort* energy level is active, then the corresponding setpoint value is used as the base setpoint value.
- **Pre-comfort:** If the *Pre-comfort* energy level is active, then the corresponding setpoint value is used as the base setpoint value.
- **Economy:** If the *Economy* energy level is active, then the corresponding setpoint value is used as the base setpoint value.
- **Protection:** If the *Protection* energy level is active, then the corresponding setpoint value is used as the base setpoint value.

Summer compensation



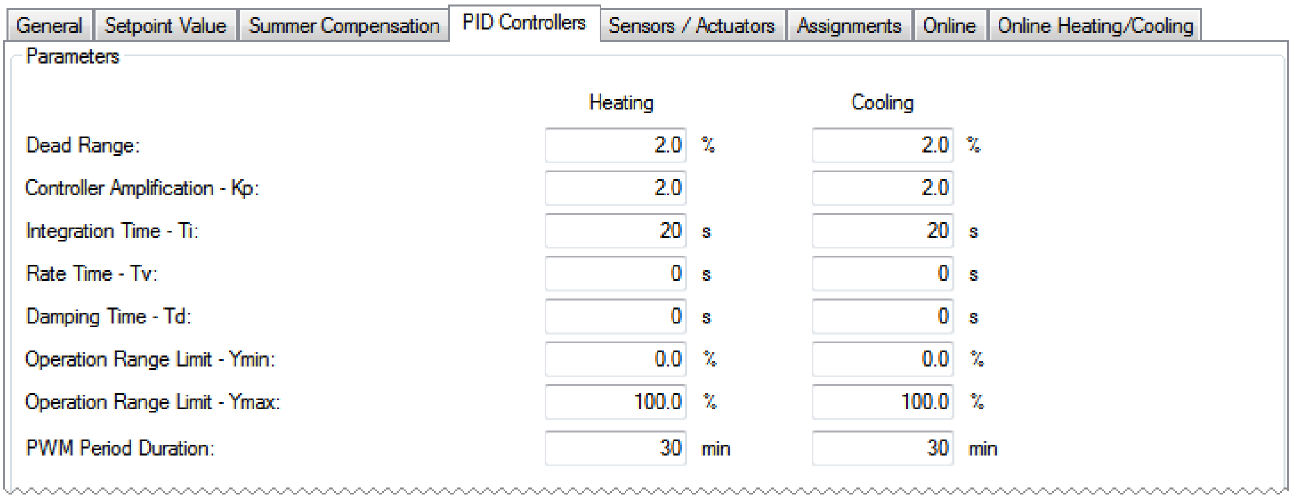
Enable:	☑	
Start Level - Outdoor Temperature:	26.0	°C
End Level - Outdoor Temperature:	32.0	°C
Gradient:	1.0	0...10

Parameters

- **Enable:** enable/disable the summer compensation.
- **Start Level - Outdoor Temperature:** outdoor temperature at which the adaptation of the setpoint value by the specified *factor* begins.
- **End Level - Outdoor Temperature:** outdoor temperature at which the adaptation of the setpoint value by the specified *factor* ends.
- **Gradient:** Factor by which the setpoint value is adapted. The value may not be less than 0 or greater than 10.

PID controller

Separate PID controllers are available for the heating and cooling control functions. The controller parameters can be set in this dialog.



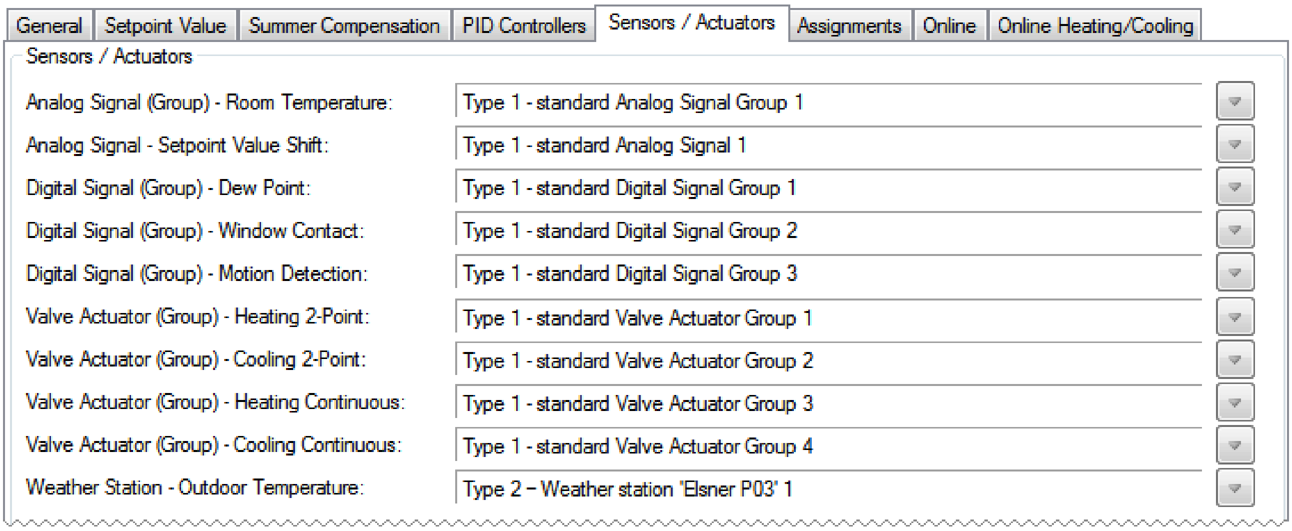
	Heating		Cooling	
Dead Range:	2.0	%	2.0	%
Controller Amplification - Kp:	2.0		2.0	
Integration Time - Ti:	20	s	20	s
Rate Time - Tv:	0	s	0	s
Damping Time - Td:	0	s	0	s
Operation Range Limit - Ymin:	0.0	%	0.0	%
Operation Range Limit - Ymax:	100.0	%	100.0	%
PWM Period Duration:	30	min	30	min

Parameter

- **Dead Range:** To prevent unnecessary movement and thus premature wear of the valves, a dead range can be set for the output signal.
- **Controller Amplification - Kp:** amplification of the controller.
- **Integration Time - Ti:** integration time of the controller.
- **Rate Time - Tv:** rate time of the controller.
- **Damping Time - Td:** damping time of the controller.
- **Operation Range Limit - Ymin:** lower limit of the operating range.
- **Operation Range Limit - Ymax:** upper limit of the operating range.
- **PWM Period Duration:** period of the PWM signal. The PWM signal controls the thermal drives.

Sensors / Actuators

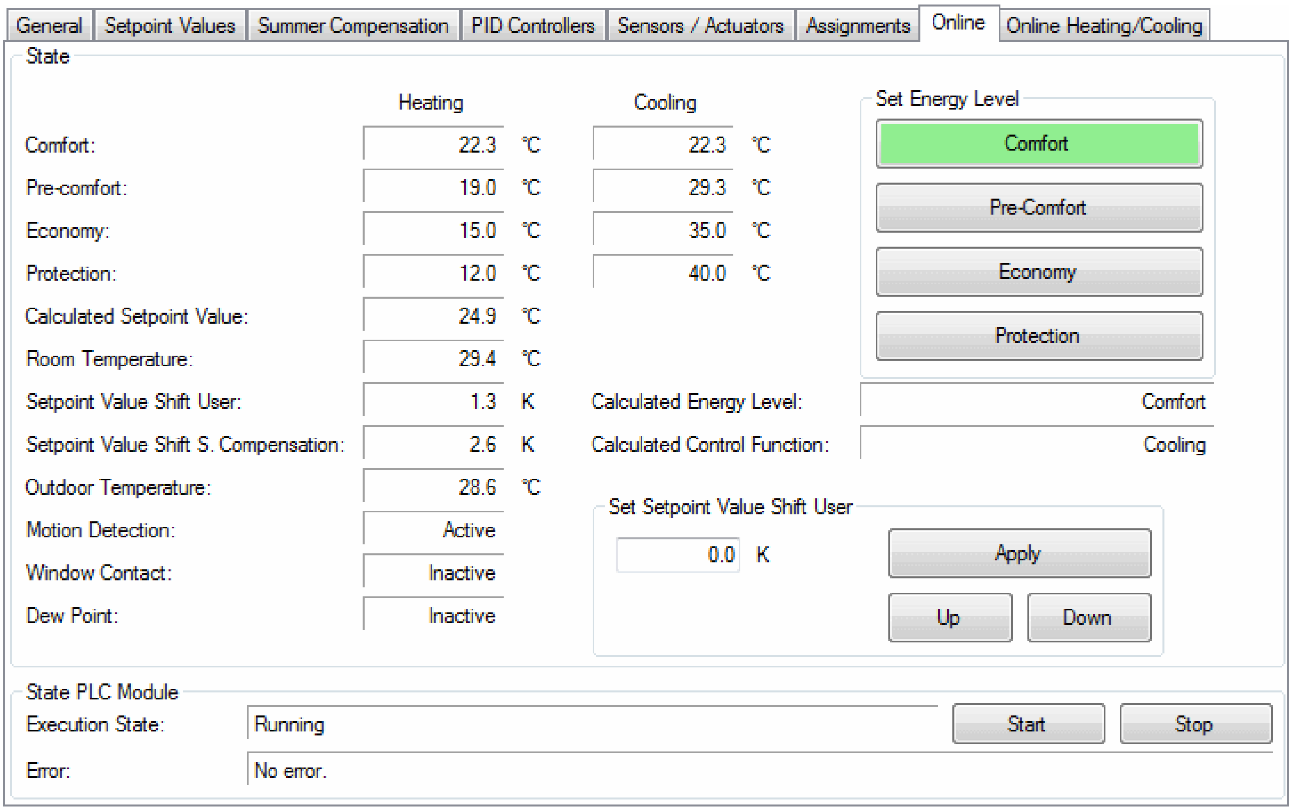
The sensors and actuators that are necessary for control are entered in this dialog.



Sensors / Actuators

Analog Signal (Group) - Room Temperature:	Type 1 - standard Analog Signal Group 1
Analog Signal - Setpoint Value Shift:	Type 1 - standard Analog Signal 1
Digital Signal (Group) - Dew Point:	Type 1 - standard Digital Signal Group 1
Digital Signal (Group) - Window Contact:	Type 1 - standard Digital Signal Group 2
Digital Signal (Group) - Motion Detection:	Type 1 - standard Digital Signal Group 3
Valve Actuator (Group) - Heating 2-Point:	Type 1 - standard Valve Actuator Group 1
Valve Actuator (Group) - Cooling 2-Point:	Type 1 - standard Valve Actuator Group 2
Valve Actuator (Group) - Heating Continuous:	Type 1 - standard Valve Actuator Group 3
Valve Actuator (Group) - Cooling Continuous:	Type 1 - standard Valve Actuator Group 4
Weather Station - Outdoor Temperature:	Type 2 – Weather station 'Elsner P03' 1

Online



State

	Heating		Cooling	
Comfort:	22.3	°C	22.3	°C
Pre-comfort:	19.0	°C	29.3	°C
Economy:	15.0	°C	35.0	°C
Protection:	12.0	°C	40.0	°C
Calculated Setpoint Value:	24.9	°C		
Room Temperature:	29.4	°C		
Setpoint Value Shift User:	1.3	K		
Setpoint Value Shift S. Compensation:	2.6	K		
Outdoor Temperature:	28.6	°C		
Motion Detection:	Active			
Window Contact:	Inactive			
Dew Point:	Inactive			

Calculated Energy Level: Comfort

Calculated Control Function: Cooling

Set Setpoint Value Shift User: 0.0 K

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Set Energy Level:** The desired energy level can be selected using these four buttons. The value can be overwritten by window contacts or motion sensors.
- **Set Setpoint Value Shift User:** the local setpoint value shift (setpoint value shift user) can be changed in this field. The *Up* and *Down* buttons respectively increase or decrease the local setpoint value shift. The *Setpoint Value Shift User – Step Size* parameter (see *Settings* dialog) specifies the step size by which the local setpoint value shift should change. This field is not displayed if an analog signal has been selected for the setpoint value shift in the *Sensors/Actuators* dialog.

  If the *setpoint value shift* lies in the positive range, then this affects the following setpoint values:

  HeatingCoolingEnergy Level Protection Economy Pre-comfort Comfort Comfort Pre-comfort Economy ProtectionBase Setpoint Value [°C]1215192124283540Setpoint Value Shift [K]---+3+3+3--Setpoint Value [°C]1215192427313540
  If the *setpoint value shift* lies in the negative range, then this affects the following setpoint values:

  HeatingCoolingEnergy Level Protection Economy Pre-comfort Comfort Comfort Pre-comfort Economy ProtectionBase Setpoint Value [°C]1215192124283540Setpoint Value Shift [K]---3-3-3---Setpoint Value [°C]1215161821313540

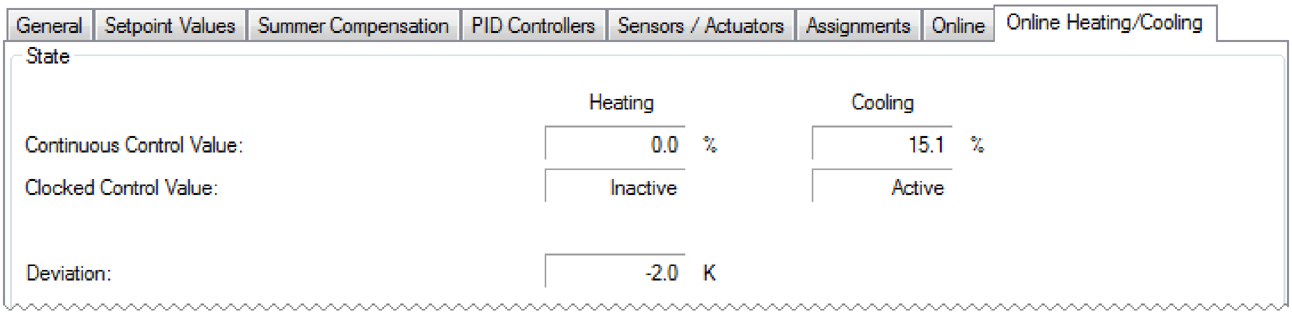
State

- **Comfort, Pre-comfort, Economy and Protection:** calculated setpoint values for the individual energy levels and control functions.
- **Calculated Setpoint Value:** Setpoint value to be referred to for further control. The calculated energy level and the calculated control function determine which of the 8 setpoint values is used.
- **Room Temperature:** current room temperature (actual value).
- **Setpoint Value Shift User:** indicates the local setpoint value shift (user setpoint value shift).
- **Setpoint Value Shift S. Compensation:** indicates the setpoint value shift of the summer compensation.
- **Outdoor Temperature:** indicates the outdoor temperature if a weather station has been connected to the air conditioning controller.
- **Motion Detection:** indicates the state of the motion sensors, if motion sensors have been connected to the air conditioning controller.
- **Window Contact:** indicates the state of the window contacts, if window contacts have been connected to the air conditioning controller.
- **Dew Point:** indicates the state of the dew point sensor, if dew point sensors have been connected to the air conditioning controller.
- **Calculated Energy Level:** the actual energy level. This results from the specified energy level and optionally from the states of the motion sensors and window contacts.
- **Calculated Control Function:** the actual control function. This results from the control deviation and optionally from the states of the dew point sensors.

State PLC Module

- **Execution State:** shows, if the module is running or stopped.
- **Error:** in case of an error, the error code and the error parameter is displayed in this box.
- **Start/Stop:** starts/stops the module.

Online Heating/Cooling



	Heating		Cooling	
Continuous Control Value:	0.0	%	15.1	%
Clocked Control Value:	Inactive		Active	
Deviation:	-2.0	K		

State

- **Continuous Control Value:** control value for controlling the constant valve actuators.
- **Clocked Control Value:** control value for controlling the thermal valve actuators.
- **Deviation:** difference between setpoint value and actual value (setpoint value - actual value).

7.2.2.7.4 Windows

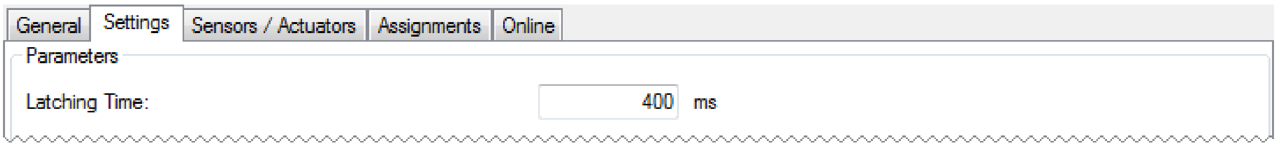
The following types of Windows are available:

- Standard Window [▶ 142]

7.2.2.7.4.1 Standard Window

The functional unit Standard Window controls a window-drive (group).

Settings

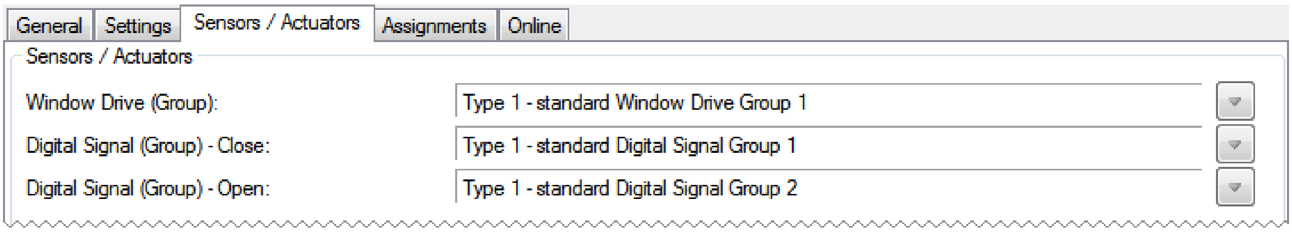


Latching Time: 400 ms

Parameters

- **Latching Time:** Time until the commands given by the sensors/sensor groups (see *Sensors / Actuators*) are maintained.

Sensors / Actuators



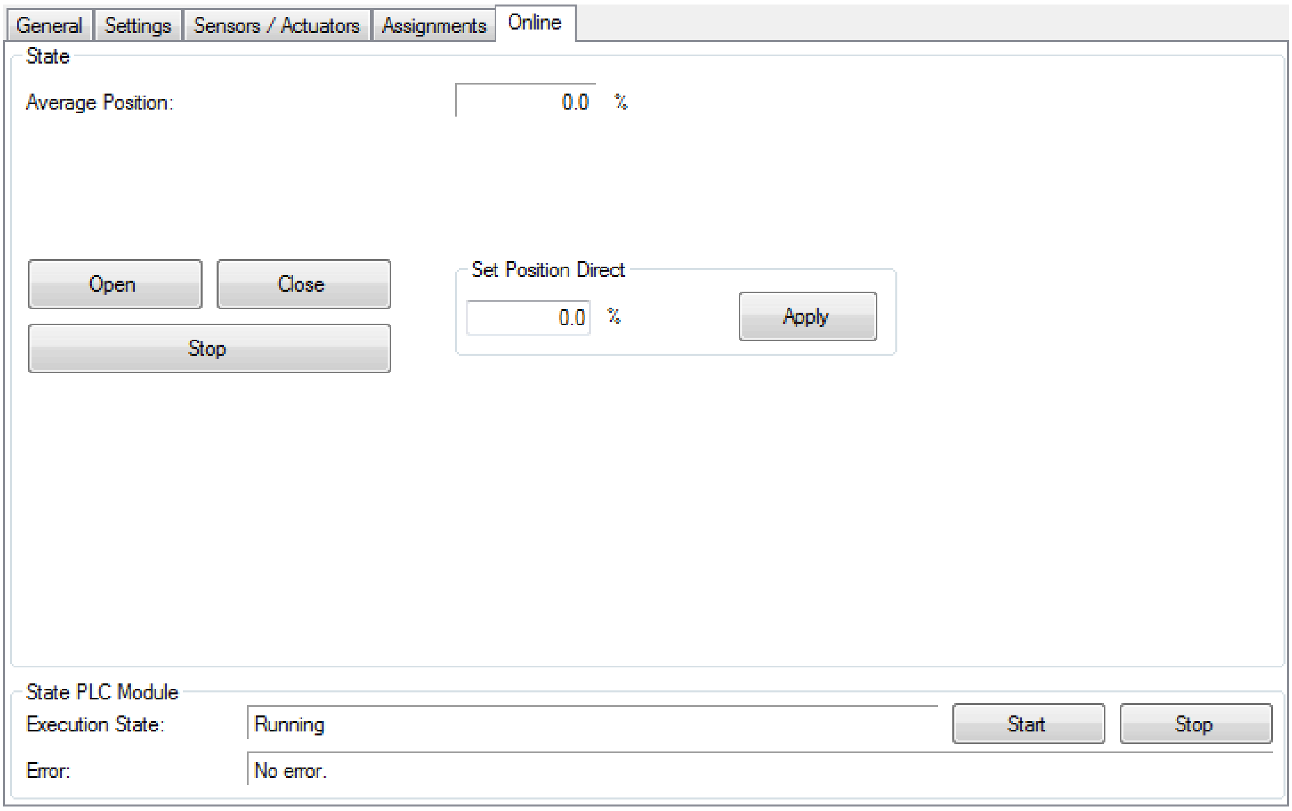
Window Drive (Group): Type 1 - standard Window Drive Group 1

Digital Signal (Group) - Close: Type 1 - standard Digital Signal Group 1

Digital Signal (Group) - Open: Type 1 - standard Digital Signal Group 2

In this menu the Window Drive / Window Drive Group has to be entered together with the Digital Signal / Digital Signal Group with their pushbuttons for opening and shutting.

Online



Average Position: 0.0 %

Set Position Direct: 0.0 %

Execution State: Running

Error: No error.

Commands

- **Open:** Opens all assigned windows.
- **Close:** Closes all assigned windows.
- **Stop:** Driving windows will be stopped.
- **Set Position Direct:** The entered value will be taken over to the window-drive(s), when the *Apply*-button is pressed.

State

- **Average Position:** Average value of all active (non-erroneous) window drives. 100% means completely opened, 0% completely closed.

State PLC Module

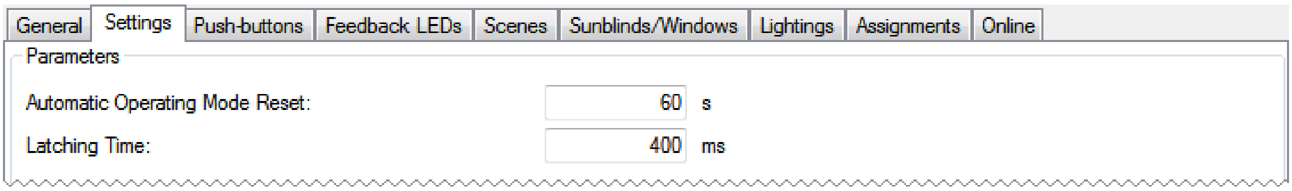
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.8 Composite Modules

7.2.2.8.1 8-gang + 2 Push-button

A push-button module is a special functionality of the Building Automation Framework to realize a simple, but nevertheless comprehensive room-control via a keypad. By an effective multiple configurations of the buttons, up to 3 blinds, 6 dimmable lights and 8 scenes can be operated with 10 individual buttons. Moreover, it is possible to change the lighting of the 6 scenes via the keypad and store them persistently in the controller. The modules are implemented in the BA-Manager as a *Composite Module* and are fully functional as soon as the settings were implemented and activated in the BA-Manager.
Two scene-hotkeys can be allocated for calling up scenes independent of the current mode. They are designed to enable persons who are not familiar with the push-button modules to utilise room automation (through standard scenes such as base-lighting or room-off).

Settings



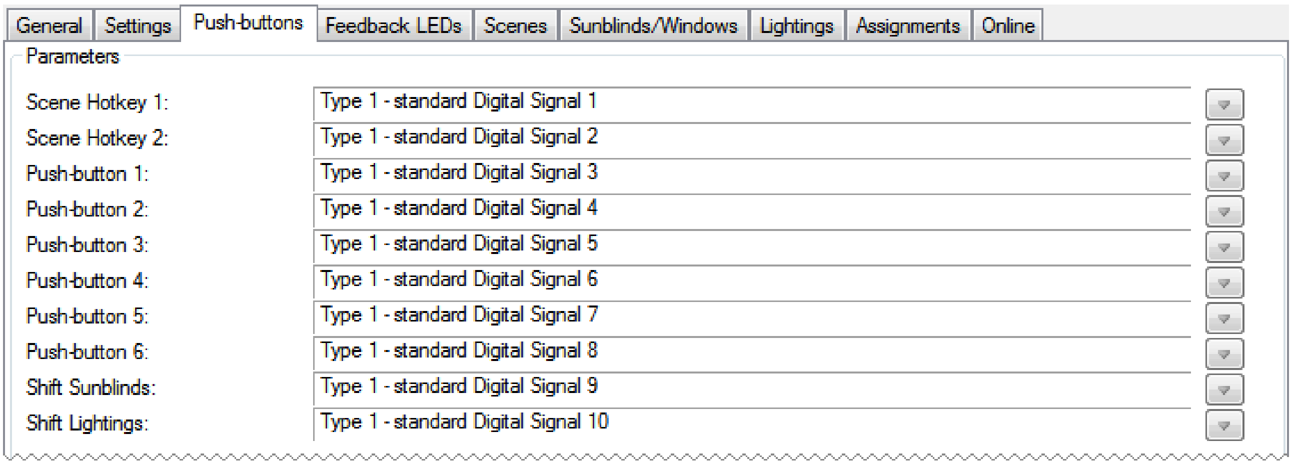
Parameters

Automatic Operating Mode Reset:	60	s
Latching Time:	400	ms

Parameters

- **Automatic Operating Mode Reset:** The time when the system change back automatically to the scene-mode from sunblinds and lighting-modes.
- **Latching Time:** Time after which a keystroke is interpreted as holding, rather than clicking.

Push-buttons

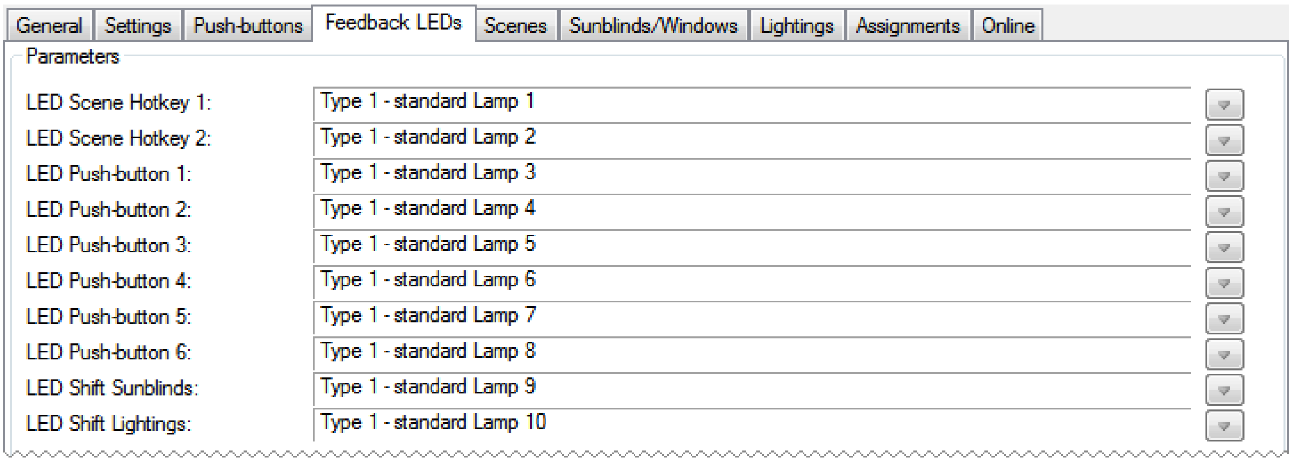


Parameters

Scene Hotkey 1:	Type 1 - standard Digital Signal 1
Scene Hotkey 2:	Type 1 - standard Digital Signal 2
Push-button 1:	Type 1 - standard Digital Signal 3
Push-button 2:	Type 1 - standard Digital Signal 4
Push-button 3:	Type 1 - standard Digital Signal 5
Push-button 4:	Type 1 - standard Digital Signal 6
Push-button 5:	Type 1 - standard Digital Signal 7
Push-button 6:	Type 1 - standard Digital Signal 8
Shift Sunblinds:	Type 1 - standard Digital Signal 9
Shift Lightings:	Type 1 - standard Digital Signal 10

Use this menu to link the button functions with the corresponding digital signals.

Feedback LEDs

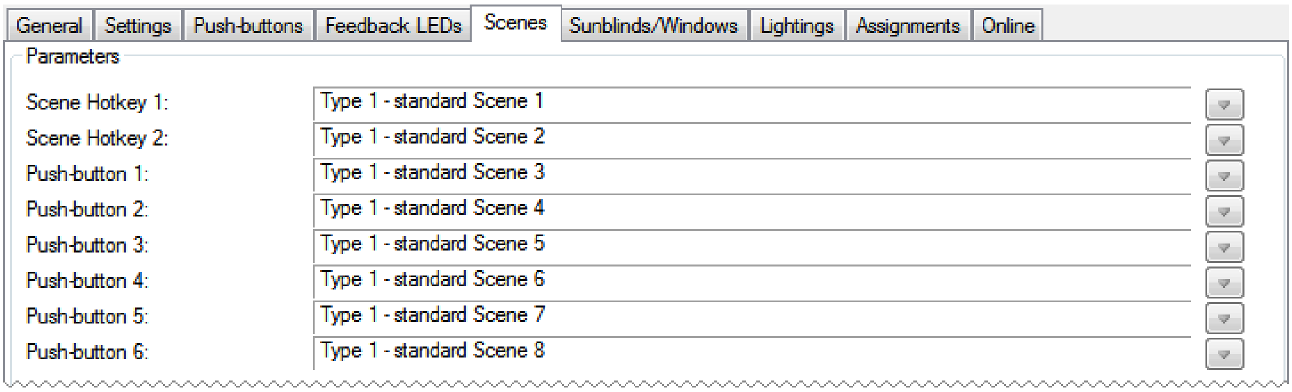


Parameters

LED Scene Hotkey 1:	Type 1 - standard Lamp 1
LED Scene Hotkey 2:	Type 1 - standard Lamp 2
LED Push-button 1:	Type 1 - standard Lamp 3
LED Push-button 2:	Type 1 - standard Lamp 4
LED Push-button 3:	Type 1 - standard Lamp 5
LED Push-button 4:	Type 1 - standard Lamp 6
LED Push-button 5:	Type 1 - standard Lamp 7
LED Push-button 6:	Type 1 - standard Lamp 8
LED Shift Sunblinds:	Type 1 - standard Lamp 9
LED Shift Lightings:	Type 1 - standard Lamp 10

Use this menu to select lamps that show the current state of a button.

Scenes

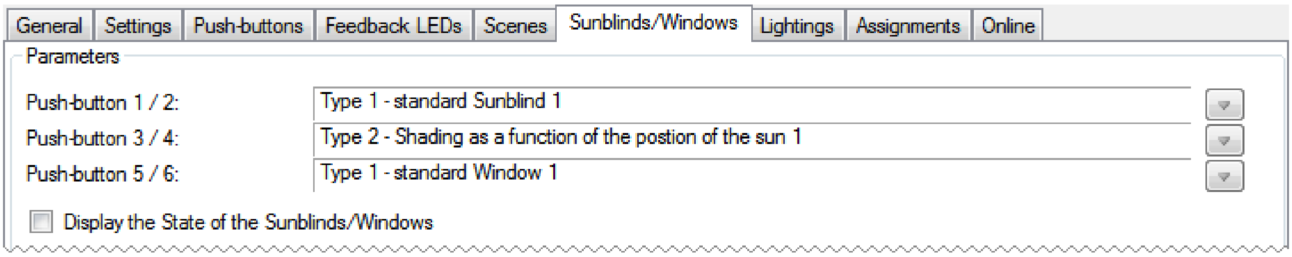


Parameters

Scene Hotkey 1:	Type 1 - standard Scene 1
Scene Hotkey 2:	Type 1 - standard Scene 2
Push-button 1:	Type 1 - standard Scene 3
Push-button 2:	Type 1 - standard Scene 4
Push-button 3:	Type 1 - standard Scene 5
Push-button 4:	Type 1 - standard Scene 6
Push-button 5:	Type 1 - standard Scene 7
Push-button 6:	Type 1 - standard Scene 8

Allocate a scene to the buttons.

Sunblinds/Windows



Parameters

Push-button 1 / 2:	Type 1 - standard Sunblind 1
Push-button 3 / 4:	Type 2 - Shading as a function of the postion of the sun 1
Push-button 5 / 6:	Type 1 - standard Window 1

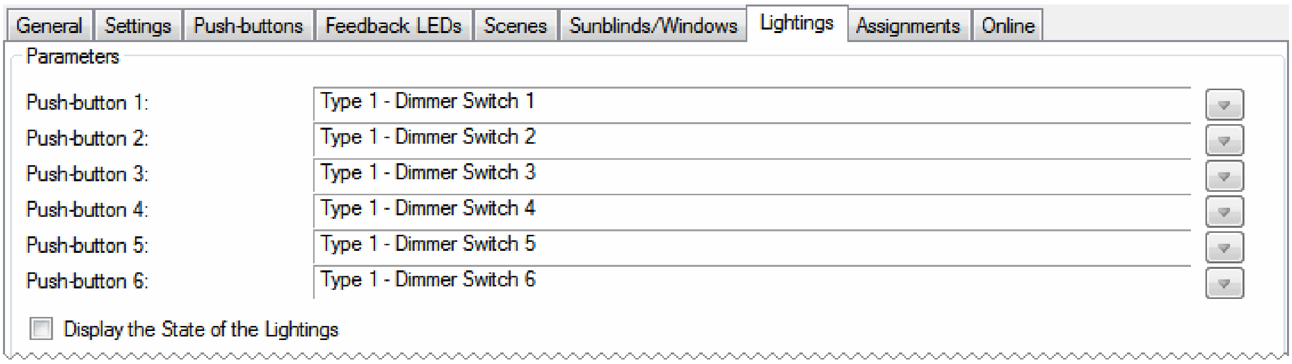
☐ Display the State of the Sunblinds/Windows

Select a shading configuration and/or window to be controlled via the respective buttons.

Parameters

- **Display the State of the Sunblinds/Windows:** Check to determine whether the LEDs should reflect the state of the shading devices and windows.

Lightings



Parameters

Push-button 1:	Type 1 - Dimmer Switch 1
Push-button 2:	Type 1 - Dimmer Switch 2
Push-button 3:	Type 1 - Dimmer Switch 3
Push-button 4:	Type 1 - Dimmer Switch 4
Push-button 5:	Type 1 - Dimmer Switch 5
Push-button 6:	Type 1 - Dimmer Switch 6

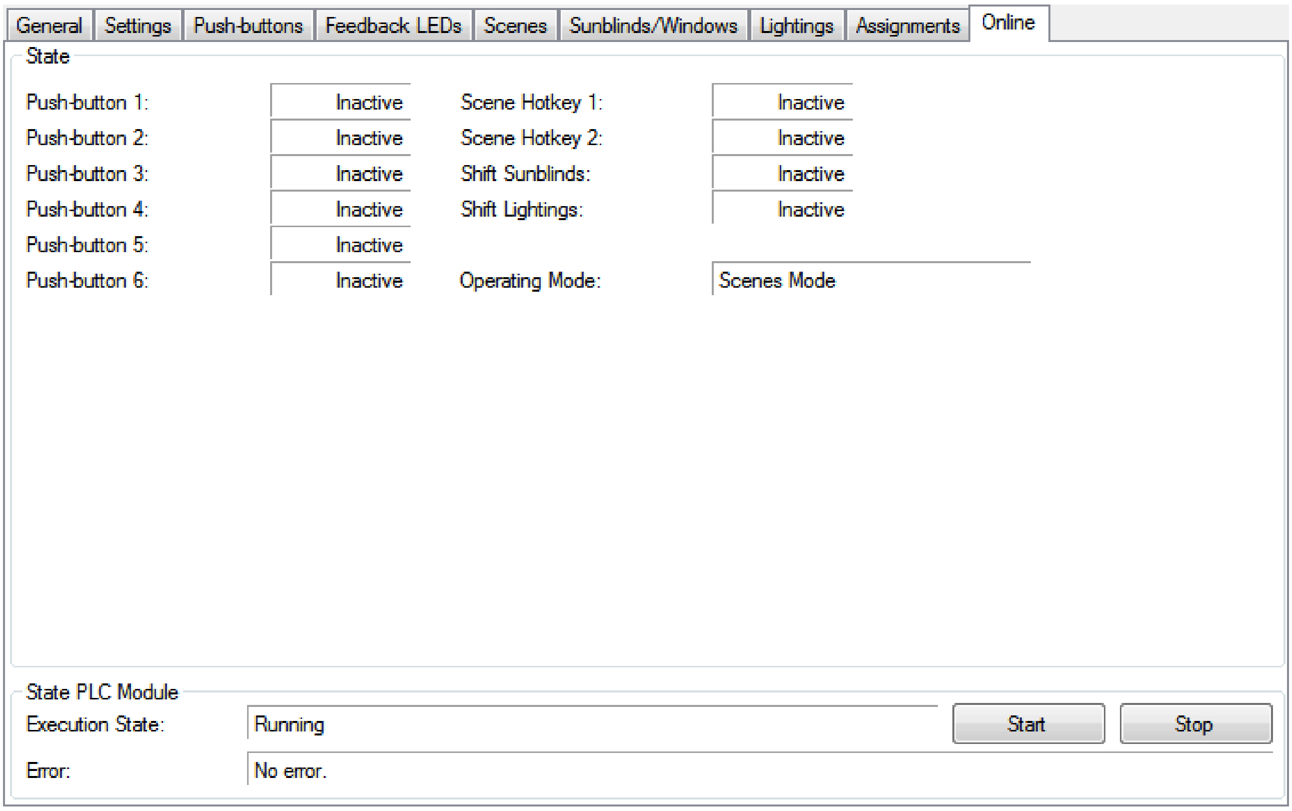
☐ Display the State of the Lightings

Select a lighting configuration to be controlled via the respective button.

Parameters

- **Display the State of the Lightings:** Check to determine whether the LEDs should reflect the state of the lightings.

Online



State

Push-button 1:	Inactive	Scene Hotkey 1:	Inactive
Push-button 2:	Inactive	Scene Hotkey 2:	Inactive
Push-button 3:	Inactive	Shift Sunblinds:	Inactive
Push-button 4:	Inactive	Shift Lightings:	Inactive
Push-button 5:	Inactive		
Push-button 6:	Inactive	Operating Mode:	Scenes Mode

State PLC Module

Execution State: Running

Error: No error.

State

- **Push-button 1...6,**
  Scene Hotkey 1...2,
  Shift Sunblinds,
  Shift Lightings: Displays the state of the buttons.
- **Operating Mode:** Displays the current *Operating Mode*.

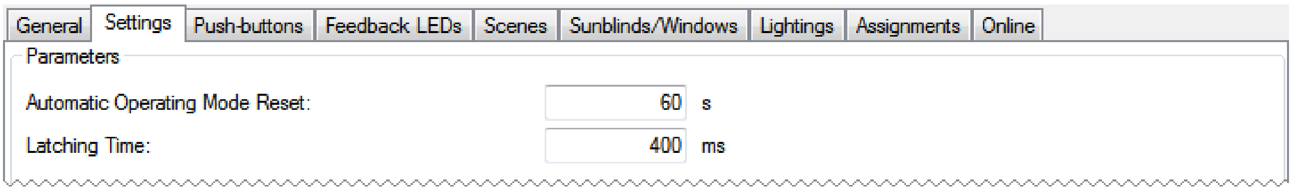
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.8.2 16-gang + 2 Push-button

A push-button module is a special functionality of the Building Automation Framework to realize a simple, but nevertheless comprehensive room-control via a keypad. By an effective multiple configurations of the buttons, up to 7 blinds, 14 dimmable lights and 16 scenes can be operated with 18 individual buttons. Moreover, it is possible to change the lighting of the 16 scenes via the keypad and store them persistently in the controller. The modules are implemented in the BA-Manager as a *Composite Module* and are fully functional as soon as the settings were implemented and activated in the BA-Manager.
Two scene-hotkeys can be allocated for calling up scenes independent of the current mode. They are designed to enable persons who are not familiar with the push-button modules to utilize room automation (through standard scenes such as base-lighting or room-off).

Settings



Parameters

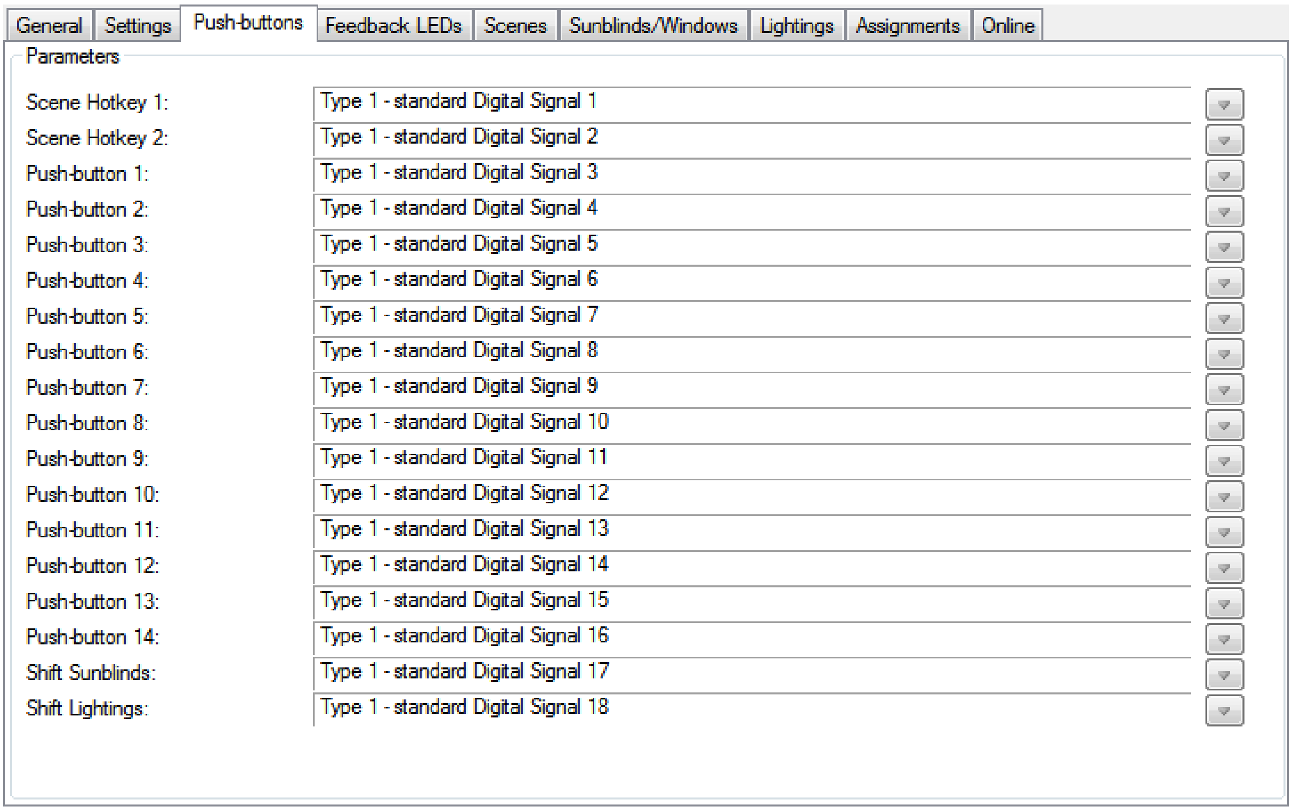
Automatic Operating Mode Reset: 60 s

Latching Time: 400 ms

Parameters

- **Automatic Operating Mode Reset:** The time when the system changes back automatically to the scene-mode from sunblinds and lighting-modes.
- **Latching Time:** Time after which a keystroke is interpreted as holding, rather than clicking.

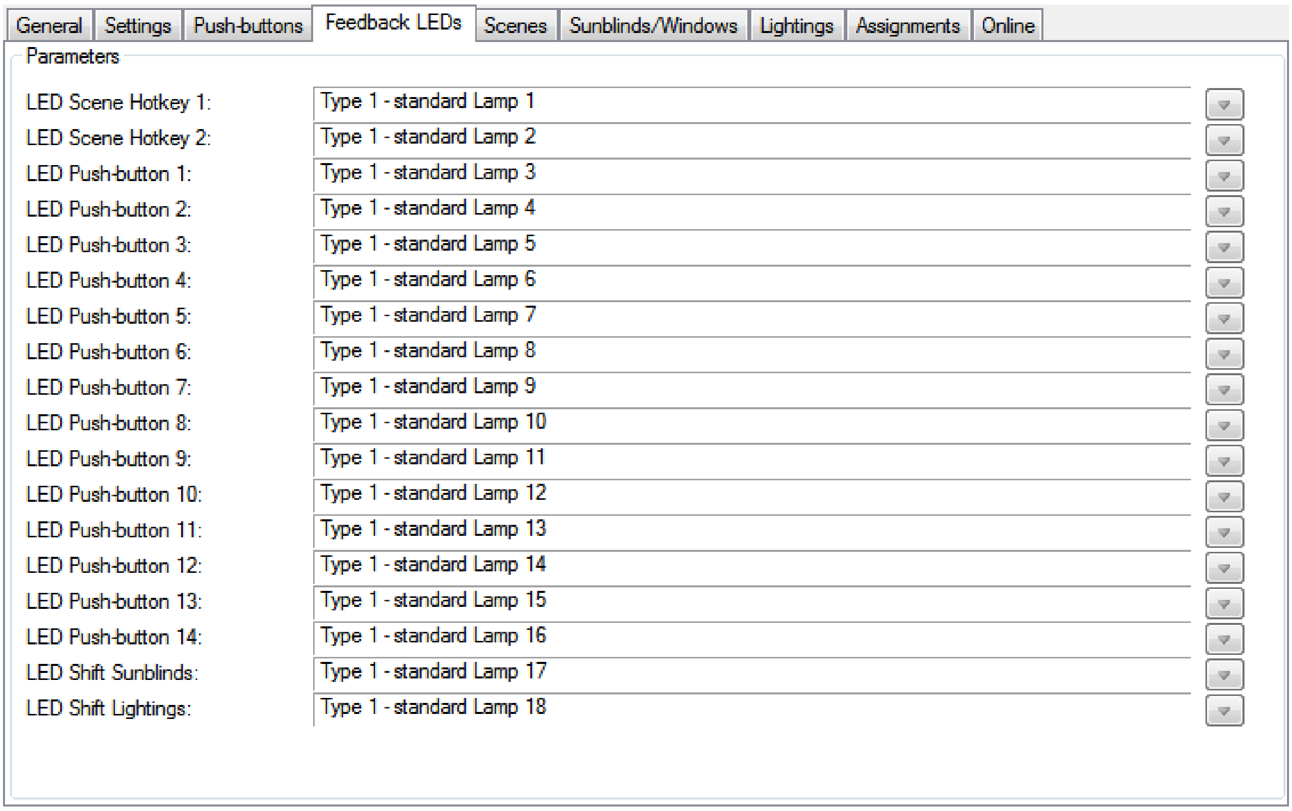
Push-buttons



Scene Hotkey 1:	Type 1 - standard Digital Signal 1
Scene Hotkey 2:	Type 1 - standard Digital Signal 2
Push-button 1:	Type 1 - standard Digital Signal 3
Push-button 2:	Type 1 - standard Digital Signal 4
Push-button 3:	Type 1 - standard Digital Signal 5
Push-button 4:	Type 1 - standard Digital Signal 6
Push-button 5:	Type 1 - standard Digital Signal 7
Push-button 6:	Type 1 - standard Digital Signal 8
Push-button 7:	Type 1 - standard Digital Signal 9
Push-button 8:	Type 1 - standard Digital Signal 10
Push-button 9:	Type 1 - standard Digital Signal 11
Push-button 10:	Type 1 - standard Digital Signal 12
Push-button 11:	Type 1 - standard Digital Signal 13
Push-button 12:	Type 1 - standard Digital Signal 14
Push-button 13:	Type 1 - standard Digital Signal 15
Push-button 14:	Type 1 - standard Digital Signal 16
Shift Sunblinds:	Type 1 - standard Digital Signal 17
Shift Lightings:	Type 1 - standard Digital Signal 18

Use this menu to link the button functions with the corresponding digital signals.

Feedback LEDs

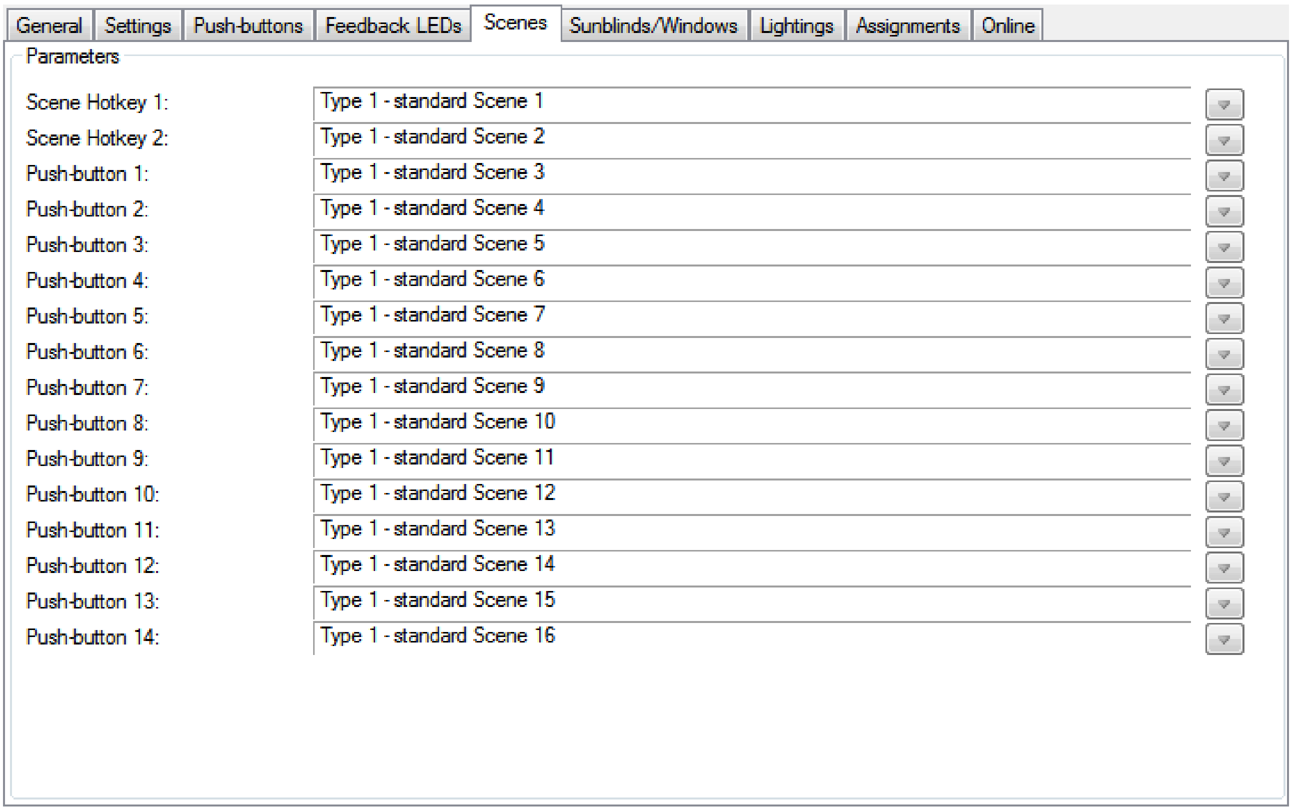


Parameters

LED Scene Hotkey 1:	Type 1 - standard Lamp 1
LED Scene Hotkey 2:	Type 1 - standard Lamp 2
LED Push-button 1:	Type 1 - standard Lamp 3
LED Push-button 2:	Type 1 - standard Lamp 4
LED Push-button 3:	Type 1 - standard Lamp 5
LED Push-button 4:	Type 1 - standard Lamp 6
LED Push-button 5:	Type 1 - standard Lamp 7
LED Push-button 6:	Type 1 - standard Lamp 8
LED Push-button 7:	Type 1 - standard Lamp 9
LED Push-button 8:	Type 1 - standard Lamp 10
LED Push-button 9:	Type 1 - standard Lamp 11
LED Push-button 10:	Type 1 - standard Lamp 12
LED Push-button 11:	Type 1 - standard Lamp 13
LED Push-button 12:	Type 1 - standard Lamp 14
LED Push-button 13:	Type 1 - standard Lamp 15
LED Push-button 14:	Type 1 - standard Lamp 16
LED Shift Sunblinds:	Type 1 - standard Lamp 17
LED Shift Lightings:	Type 1 - standard Lamp 18

Use this menu to select lamps that show the current state of a button.

Scenes

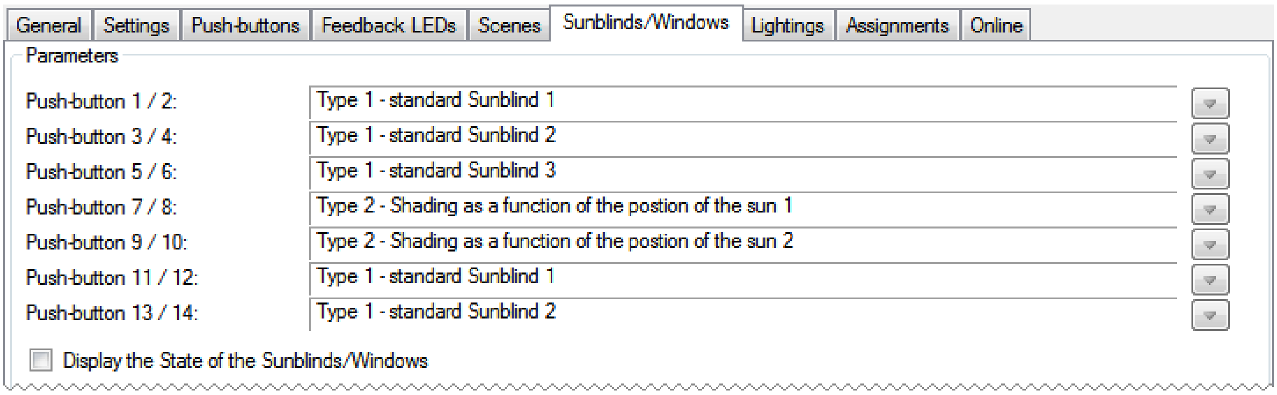


Parameters

Scene Hotkey 1:	Type 1 - standard Scene 1
Scene Hotkey 2:	Type 1 - standard Scene 2
Push-button 1:	Type 1 - standard Scene 3
Push-button 2:	Type 1 - standard Scene 4
Push-button 3:	Type 1 - standard Scene 5
Push-button 4:	Type 1 - standard Scene 6
Push-button 5:	Type 1 - standard Scene 7
Push-button 6:	Type 1 - standard Scene 8
Push-button 7:	Type 1 - standard Scene 9
Push-button 8:	Type 1 - standard Scene 10
Push-button 9:	Type 1 - standard Scene 11
Push-button 10:	Type 1 - standard Scene 12
Push-button 11:	Type 1 - standard Scene 13
Push-button 12:	Type 1 - standard Scene 14
Push-button 13:	Type 1 - standard Scene 15
Push-button 14:	Type 1 - standard Scene 16

Allocate a scene to the buttons.

Sunblinds/Windows



Parameters

Push-button 1 / 2:	Type 1 - standard Sunblind 1
Push-button 3 / 4:	Type 1 - standard Sunblind 2
Push-button 5 / 6:	Type 1 - standard Sunblind 3
Push-button 7 / 8:	Type 2 - Shading as a function of the postion of the sun 1
Push-button 9 / 10:	Type 2 - Shading as a function of the postion of the sun 2
Push-button 11 / 12:	Type 1 - standard Sunblind 1
Push-button 13 / 14:	Type 1 - standard Sunblind 2

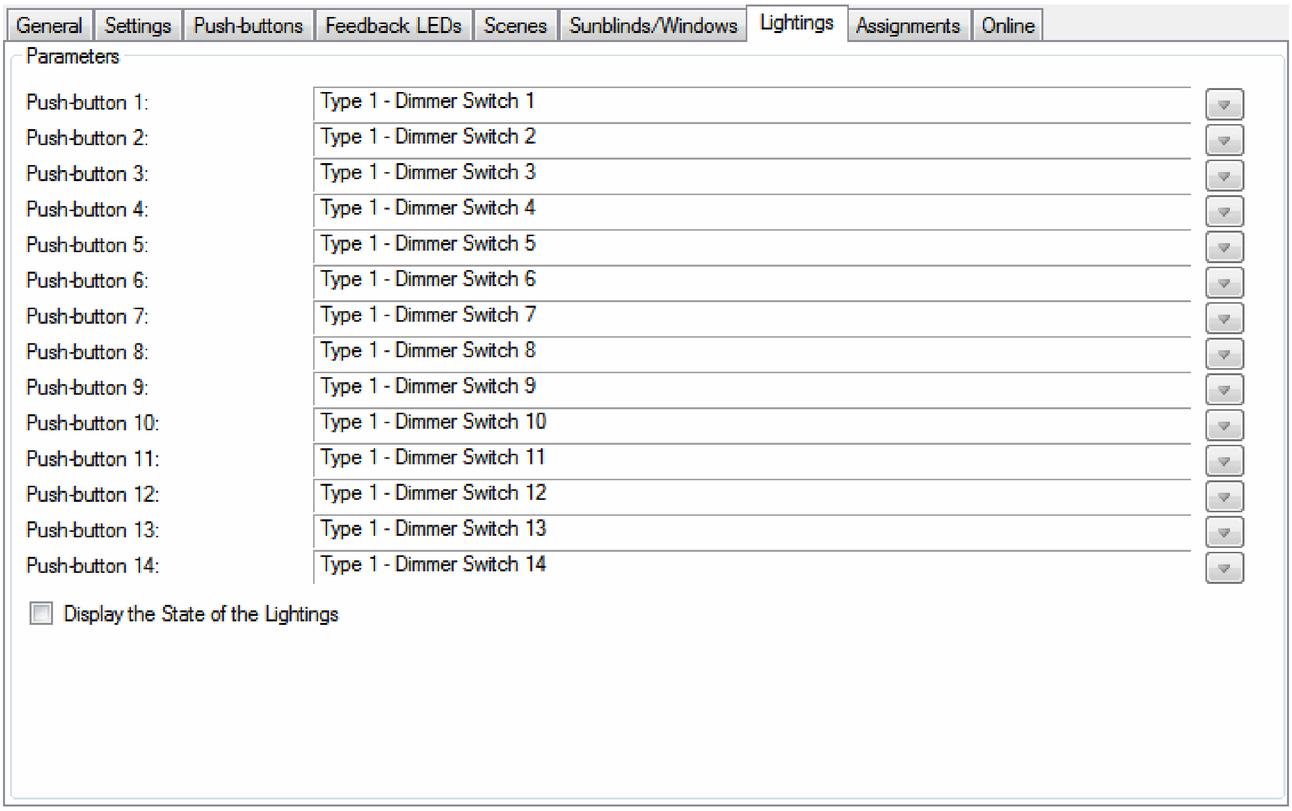
☐ Display the State of the Sunblinds/Windows

Select a shading configuration and/or window to be controlled via the respective buttons.

Parameters

- **Display the State of the Sunblinds/Windows:** Check to determine whether the LEDs should reflect the state of the shading devices and windows.

Lightings



Parameters

Push-button 1:	Type 1 - Dimmer Switch 1
Push-button 2:	Type 1 - Dimmer Switch 2
Push-button 3:	Type 1 - Dimmer Switch 3
Push-button 4:	Type 1 - Dimmer Switch 4
Push-button 5:	Type 1 - Dimmer Switch 5
Push-button 6:	Type 1 - Dimmer Switch 6
Push-button 7:	Type 1 - Dimmer Switch 7
Push-button 8:	Type 1 - Dimmer Switch 8
Push-button 9:	Type 1 - Dimmer Switch 9
Push-button 10:	Type 1 - Dimmer Switch 10
Push-button 11:	Type 1 - Dimmer Switch 11
Push-button 12:	Type 1 - Dimmer Switch 12
Push-button 13:	Type 1 - Dimmer Switch 13
Push-button 14:	Type 1 - Dimmer Switch 14

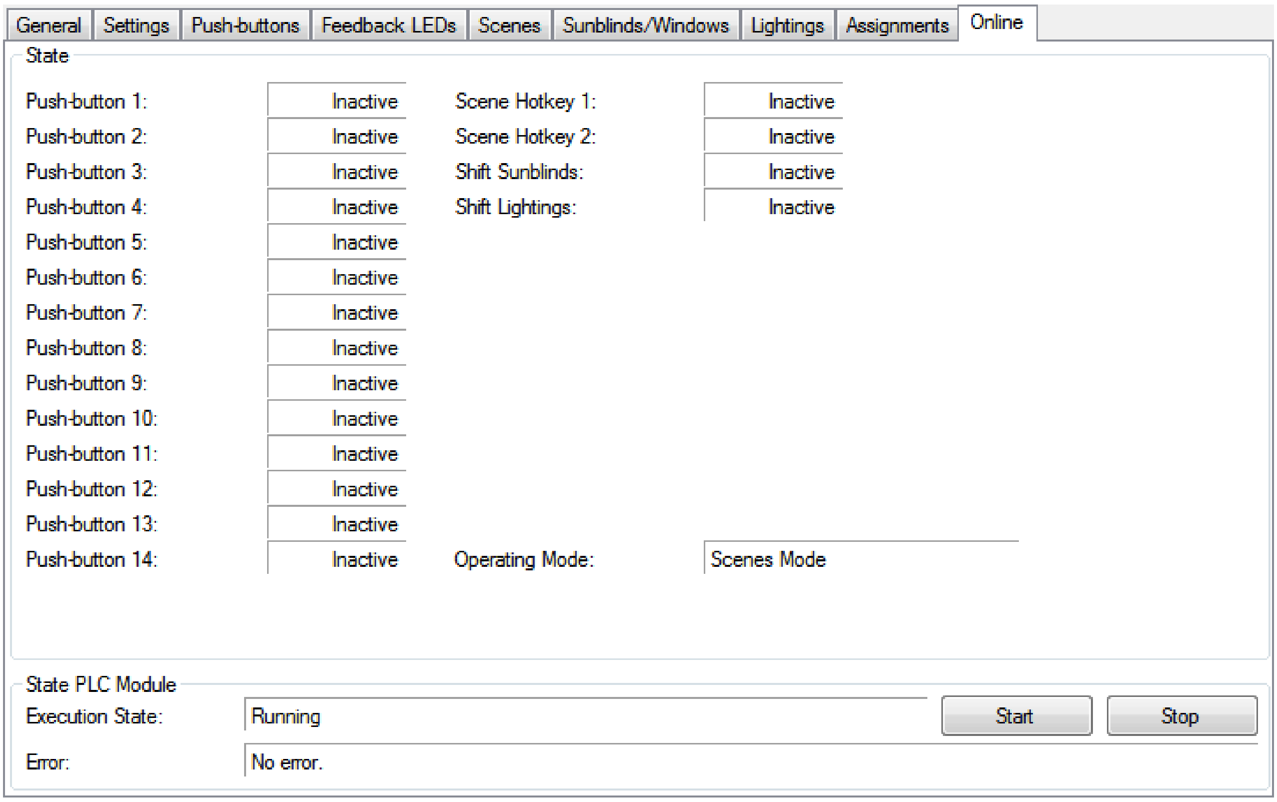
☐ Display the State of the Lightings

Select a lighting configuration to be controlled via the respective button.

Parameters

- **Display the State of the Lightings:** Check to determine whether the LEDs should reflect the state of the lightings.

Online



State

Push-button 1:	Inactive	Scene Hotkey 1:	Inactive
Push-button 2:	Inactive	Scene Hotkey 2:	Inactive
Push-button 3:	Inactive	Shift Sunblinds:	Inactive
Push-button 4:	Inactive	Shift Lightings:	Inactive
Push-button 5:	Inactive		
Push-button 6:	Inactive		
Push-button 7:	Inactive		
Push-button 8:	Inactive		
Push-button 9:	Inactive		
Push-button 10:	Inactive		
Push-button 11:	Inactive		
Push-button 12:	Inactive		
Push-button 13:	Inactive		
Push-button 14:	Inactive	Operating Mode:	Scenes Mode

State PLC Module

Execution State: Running

Error: No error.

State

- **Push-button 1...14,**
  Scene Hotkey 1...2,
  Shift Sunblinds,
  Shift Lightings: Displays the state of the buttons.
- **Operating Mode:** Displays the current *Operating Mode*.

State PLC Module

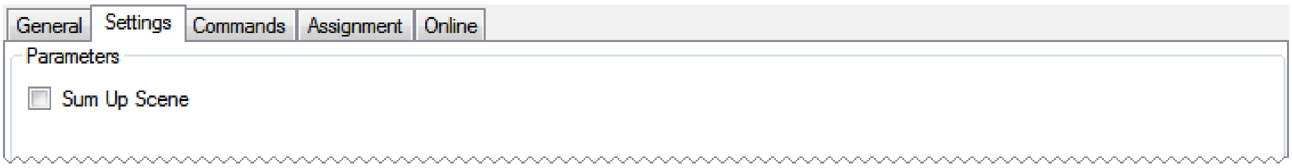
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.9 Scenes

7.2.2.9.1 Standard scene

A scene can send different commands to actuators, groups of actuators, functional units, sensors, groups of sensors and to other scenes. In this way it is possible to adapt a room, a floor or a whole building to certain circumstances with just one call.

Settings



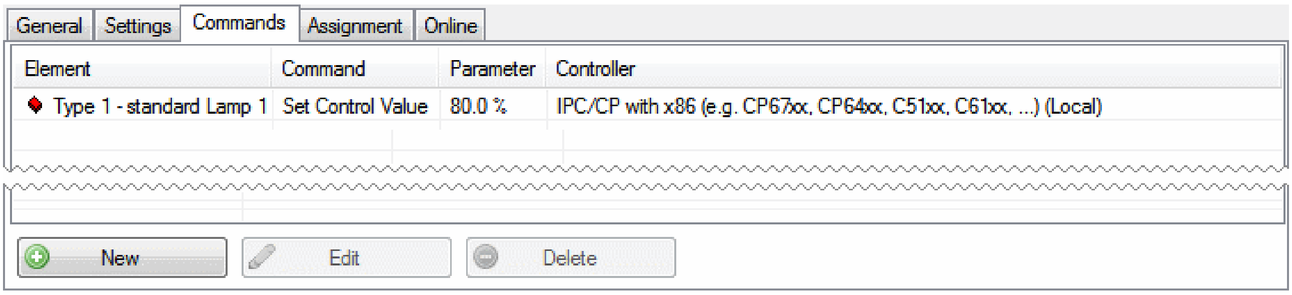
Parameters

□ Sum Up Scene

Parameter

- **Summing of scenes:** if activated, all light commands that set the value directly to 0 are ignored. This prevents the lighting being switching off when this scene is called.

Commands



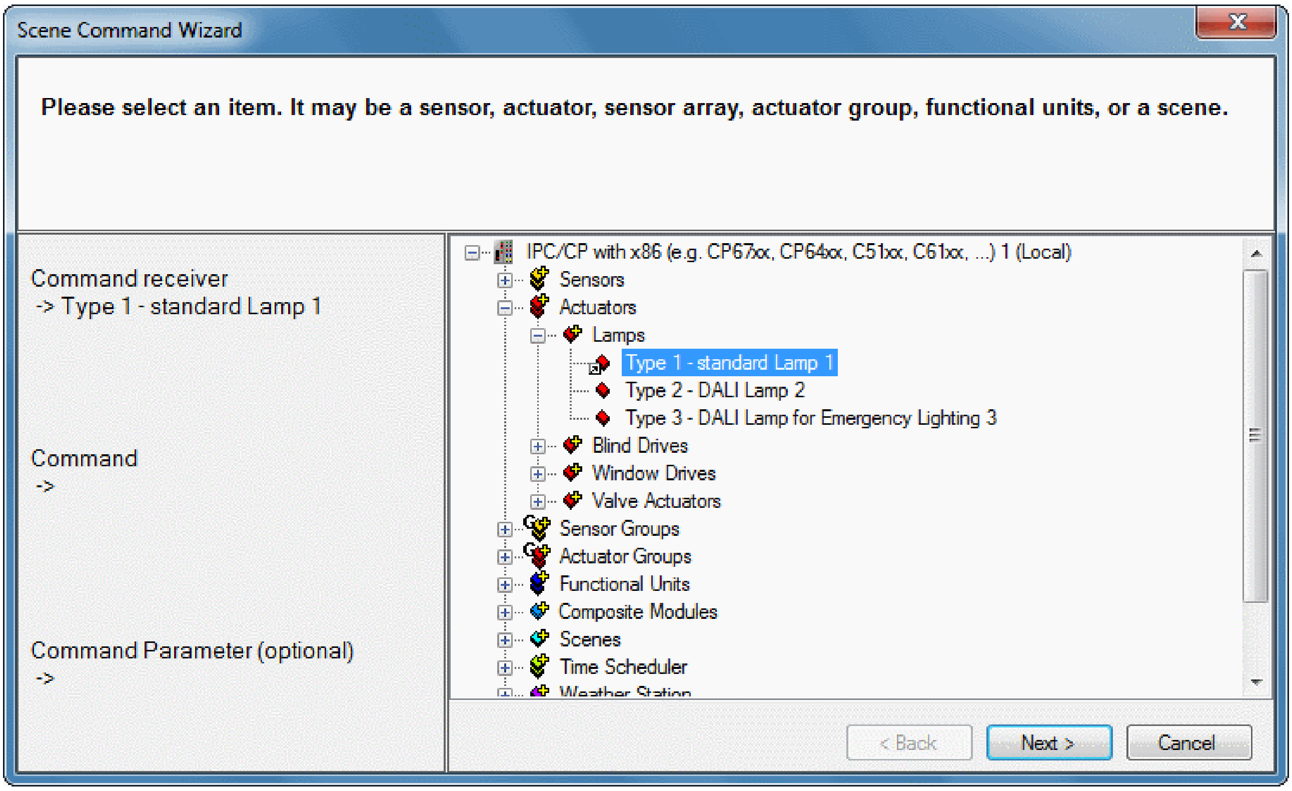
Element	Command	Parameter	Controller
Type 1 - standard Lamp 1	Set Control Value	80.0 %	IPC/CP with x86 (e.g. CP67xx, CP64xx, C51xx, C61xx, ...) (Local)

Pressing the *New* button opens the *Scene assistant*.

Scene assistant

The *scene assistant* guides you through the three steps required to configure a scene.

Step 1 - Command recipient



Please select an item. It may be a sensor, actuator, sensor array, actuator group, functional units, or a scene.

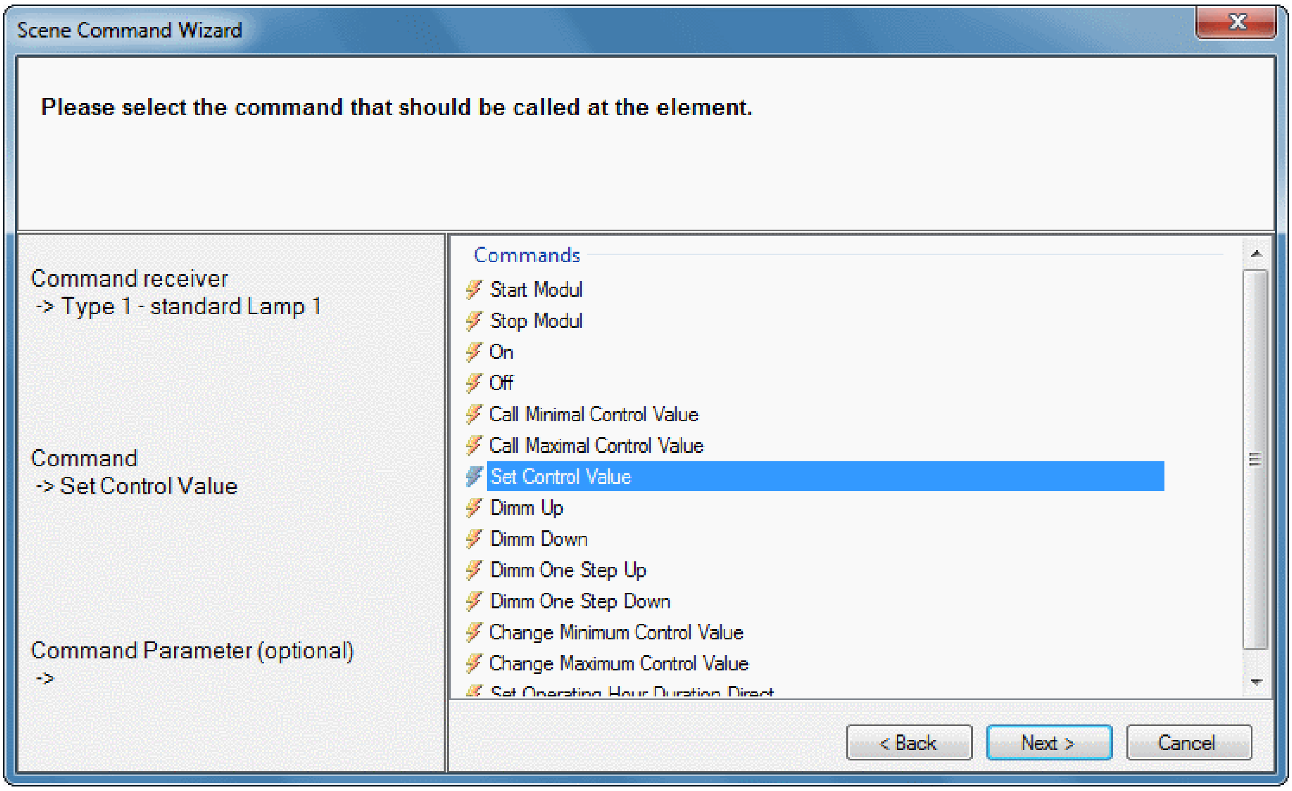
Command receiver
-> Type 1 - standard Lamp 1

Command
->

Command Parameter (optional)
->

Select the command recipient that is to be activated when the scene is called.

Step 2 - Command



Scene Command Wizard

Please select the command that should be called at the element.

Command receiver
-> Type 1 - standard Lamp 1

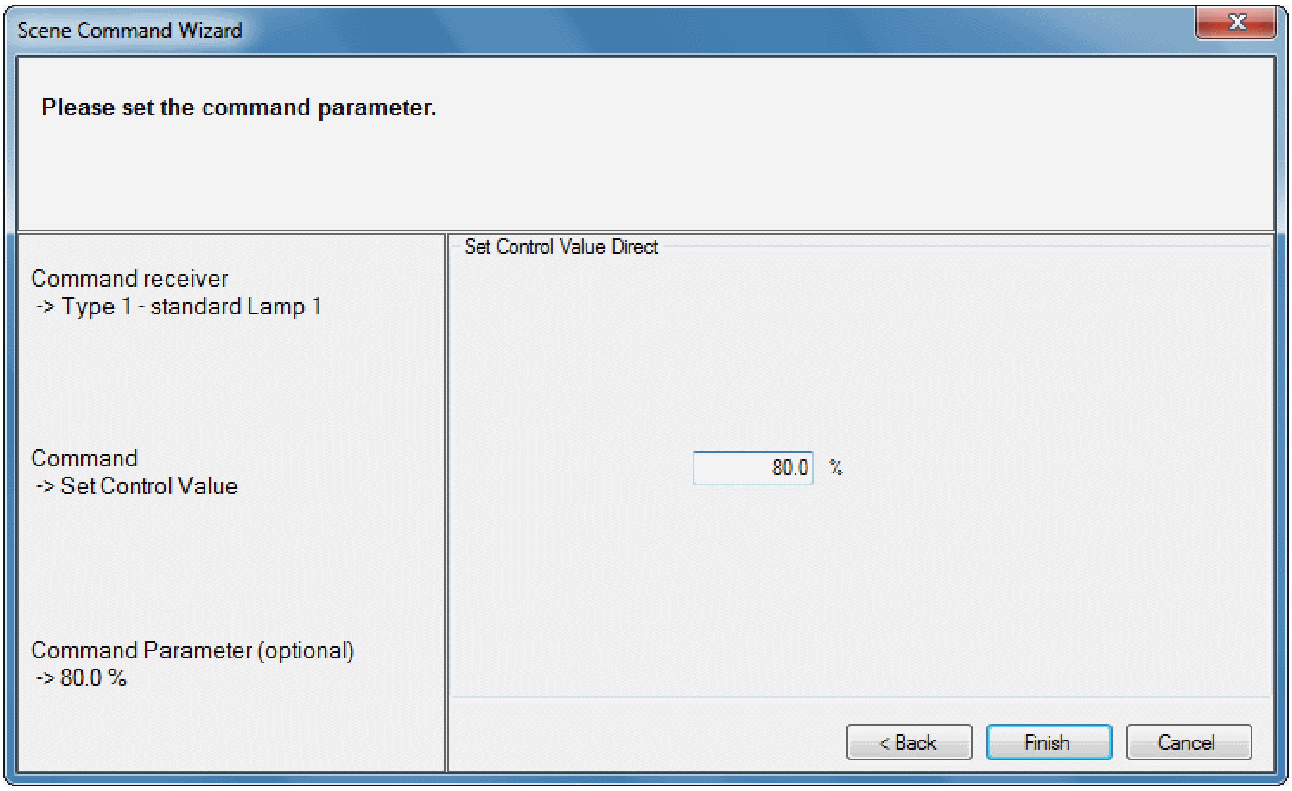
Command
-> Set Control Value

Command Parameter (optional)
->

Commands
Start Modul
Stop Modul
On
Off
Call Minimal Control Value
Call Maximal Control Value
Set Control Value
Dimm Up
Dimm Down
Dimm One Step Up
Dimm One Step Down
Change Minimum Control Value
Change Maximum Control Value
Set Operating Hour Duration Direct
< Back
Next >
Cancel

Selection of the command which the command recipient is to execute.

Step 3 - Command parameters (optional)



Scene Command Wizard

Please set the command parameter.

Command receiver
-> Type 1 - standard Lamp 1

Command
-> Set Control Value

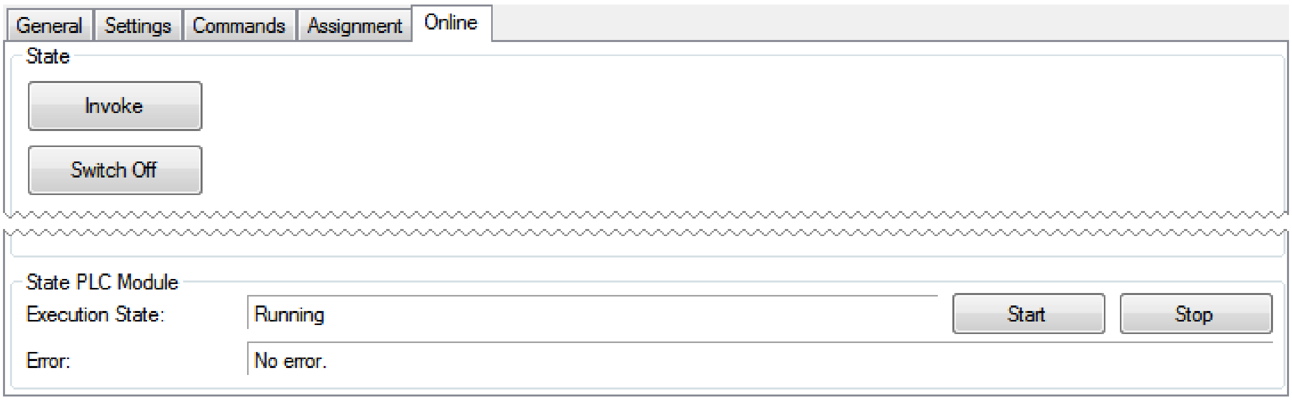
Command Parameter (optional)
-> 80.0 %

Set Control Value Direct

80.0 %

If the selected command can accept further parameters, they can be entered here.

Online



Execution State: Running

Error: No error.

Commands

- **Call:** the scene is called.
- **Switch off:** all elements contained in this scene receive the command *Off* or *Stop*. If an element does not support either of the two commands, it is ignored.

PLC module status

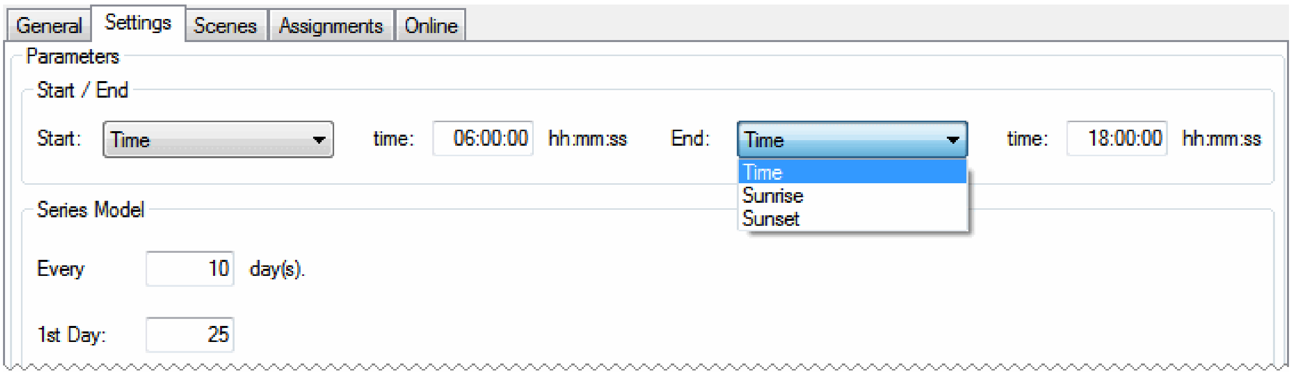
- **Execution status:** indicates whether the module is running or has stopped.
- **Error:** in the event of an error the error code and the error parameter can be seen here.
- **Start/stop:** starts/stops the module.

7.2.2.10 Time Scheduler Channels

7.2.2.10.1 Daily Time Scheduler Channel

The Daily Time Scheduler Channel presents a simple time-switch with an adjustable day-periodicity.

Settings



Start: Time time: 06:00:00 hh:mm:ss End: Time time: 18:00:00 hh:mm:ss

Every 10 day(s).

1st Day: 25

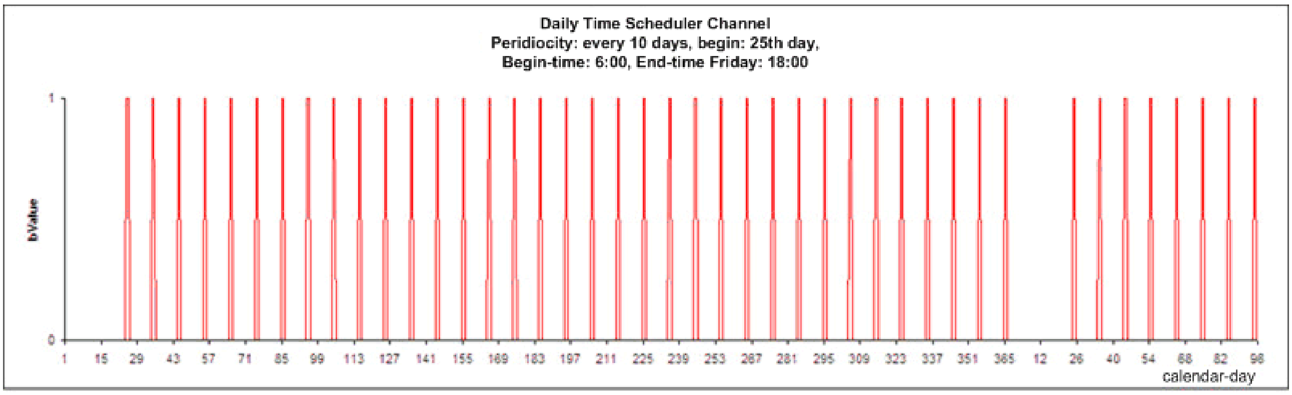
Parameters

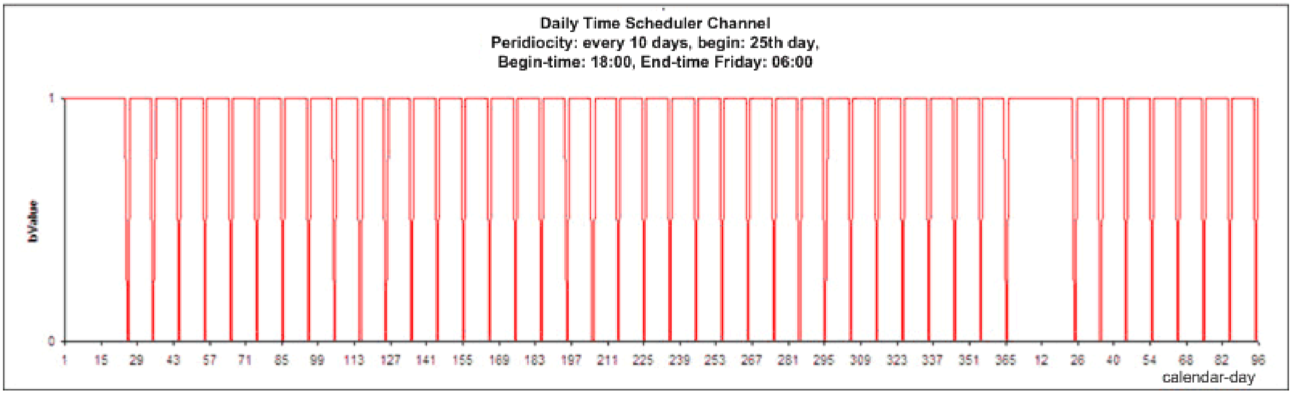
- **Start: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  Start: Sunrise + Offset: The scheduler-channel will be active beginning with the actual sunrise + offset (in minutes) on a switching-day.
  Start: Sunset + Offset: The scheduler-channel will be active beginning with the actual sunset + offset (in minutes) on a switching-day.
- **End: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  End: Sunrise + Offset: The scheduler-channel will be inactive beginning with the actual sunrise + offset (in minutes) on a switching-day.
  End: Sunset + Offset: The scheduler-channel will be inactive beginning with the actual sunset + offset (in minutes) on a switching-day.

Series Model

- **Every n day(s):** Periodicity.
- **1st Day:** From this day on all switching-days are calculated by the periodicity.

Behavior

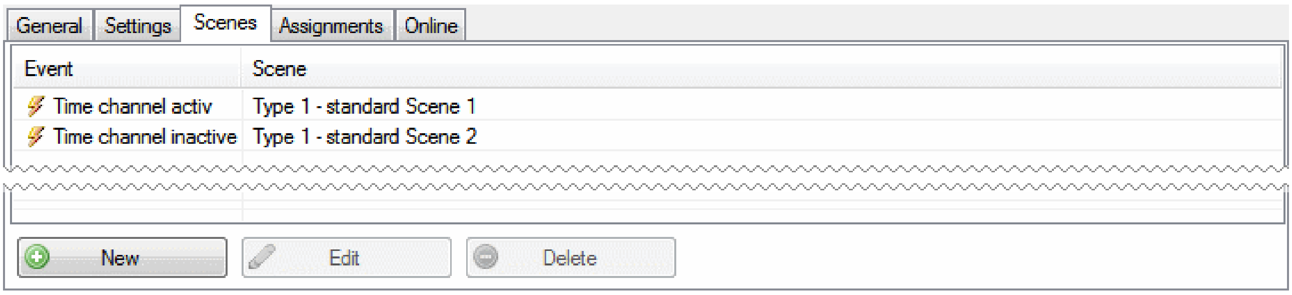

Daily Time Scheduler Channel
Peridiocity: every 10 days, begin: 25th day,
Begin-time: 6:00, End-time Friday: 18:00


Daily Time Scheduler Channel
Peridiocity: every 10 days, begin: 25th day,
Begin-time: 18:00, End-time Friday: 06:00

The periodicity is calculated from the beginning of the day, the PLC is initialized. If the start is earlier than the end, the scheduler channel will only be active from the start-time until the end-time of the switching-day. If the end is earlier than the start, the scheduler-channel will just work inversely. Entering the same time for both start and end would result in an undefined state and will thus be regarded as an input-error.

It is also possible to use the sunrise/sunset function which will calculate both times for the switching-day together with an offset, which can be either positive or negative. These resulting start and end-time however are automatically delimited to the beginning (00:00:00:000) or the end of the day (23:59:59:999).

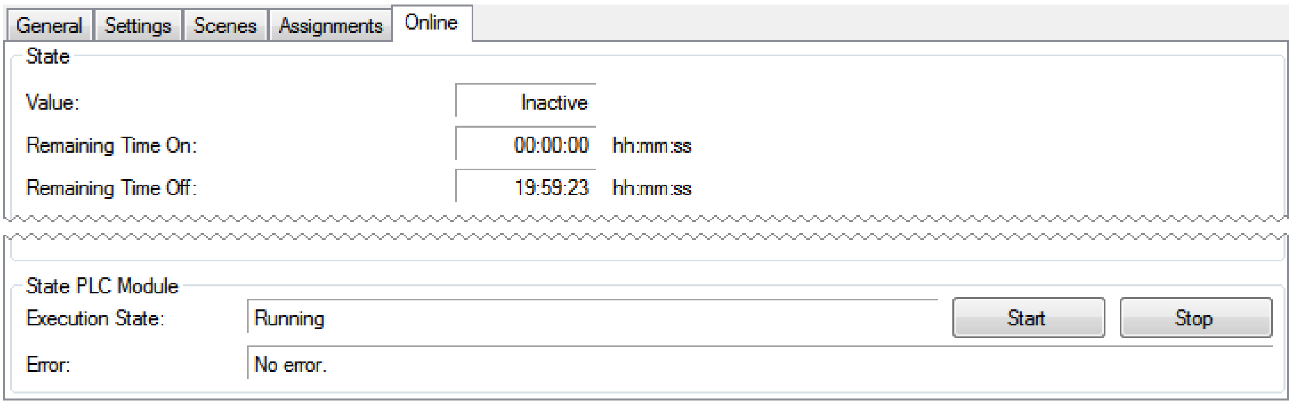
Scenes



Event	Scene
Time channel activ	Type 1 - standard Scene 1
Time channel inactive	Type 1 - standard Scene 2

On this screen the scenes can be entered, which shall be executed, when the scheduler-channel turns to either its active or inactive state.

Online



State

Value: Inactive

Remaining Time On: 00:00:00 hh:mm:ss

Remaining Time Off: 19:59:23 hh:mm:ss

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Displays the current state of the scheduler-channel.
- **Remaining Time On:** Remaining time in which the scheduler-channel is active.
- **Remaining Time Off:** Remaining time in which the scheduler-channel is inactive.

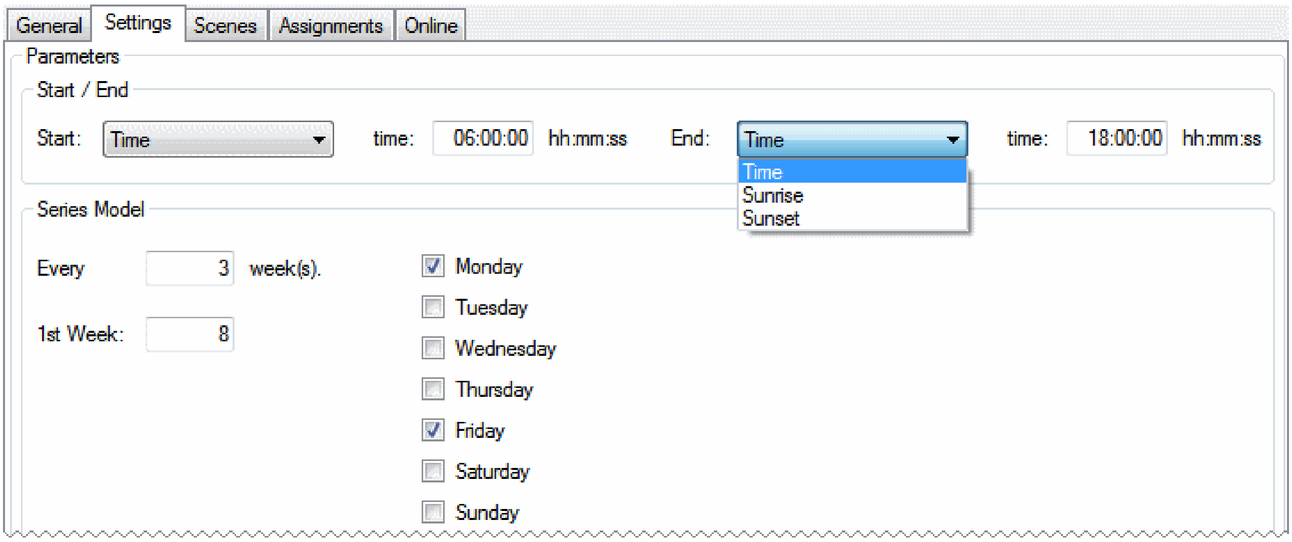
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.10.2 Weekly Time Scheduler Channel (individual days of the week)

The Weekly Time Scheduler Channel (individual days of the week) does the switching on special weekdays with an adjustable week-periodicity.

Settings



Parameters

Start / End

Start: Time time: 06:00:00 hh:mm:ss End: Time time: 18:00:00 hh:mm:ss

Series Model

Every 3 week(s).

1st Week: 8

☑ Monday

□ Tuesday

□ Wednesday

□ Thursday

☑ Friday

□ Saturday

□ Sunday

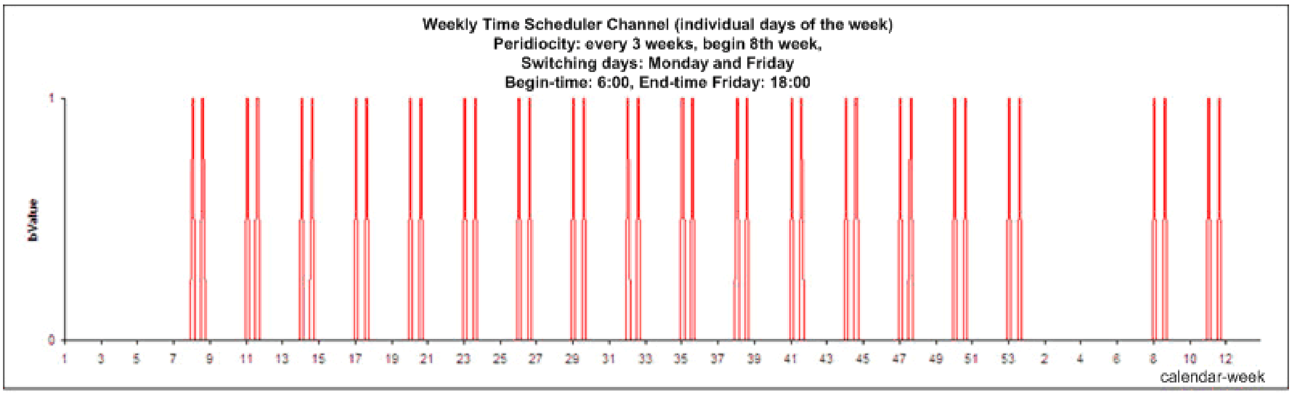
- **Start: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  Start: Sunrise + Offset: The scheduler-channel will be active beginning with the actual sunrise + offset (in minutes) on a switching-day.
  Start: Sunset + Offset: The scheduler-channel will be active beginning with the actual sunset + offset (in minutes) on a switching-day.
- **End: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  End: Sunrise + Offset: The scheduler-channel will be inactive beginning with the actual sunrise +

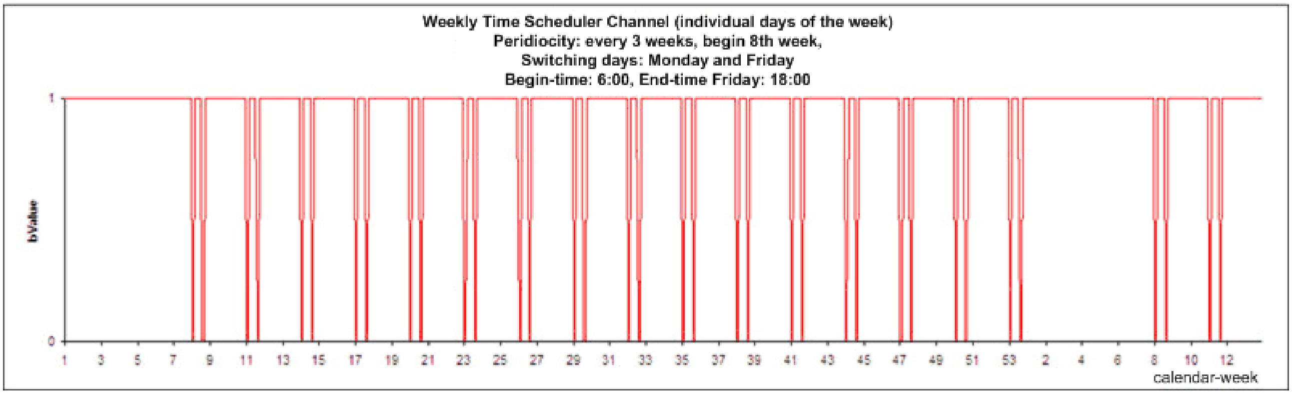
offset (in minutes) on a switching-day.
End: Sunset + Offset: The scheduler-channel will be inactive beginning with the actual sunset + offset (in minutes) on a switching-day.

Series Model

- **Every n week(s):** Periodicity.
- **1st week:** From this week on all other switching-weeks are calculated by the periodicity.
- **Monday...Sunday:** Selectable switching-days.

Behaviour



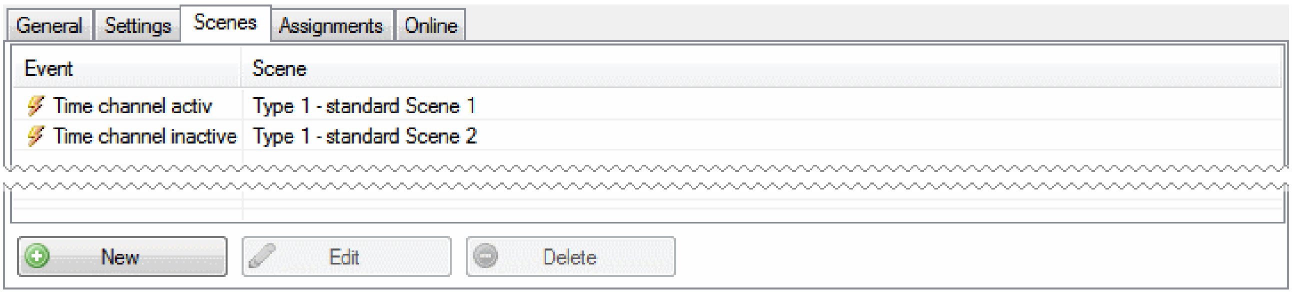


The scheduler is defined by an initial calendar-week and a periodicity. The calendar-weeks before the initial week do not contain any switch event. Afterwards the switching will happen on every selected weekday in every nth week.

If the daytime of the start is lower than the end, the scheduler channel will only be active from the start-time until the end-time of the switching-day. If the daytime of the end is lower than the start, the scheduler-channel will just work inversely. Entering the same time for both start and end would result in an undefined state and will thus be regarded as an input-error.

It is also possible to use the sunrise/sunset function which will calculate both times for the switching-day together with an offset, which can be either positive or negative. These resulting start and end-time however are automatically delimited to the beginning (00:00:00:000) or the end of the day (23:59:59:999).

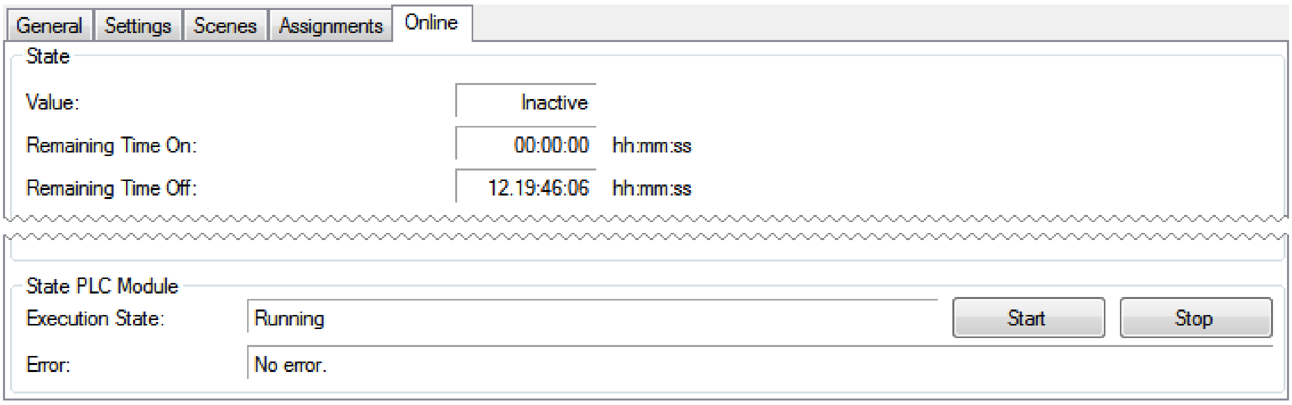
Scenes



Event	Scene
Time channel activ	Type 1 - standard Scene 1
Time channel inactive	Type 1 - standard Scene 2

On this screen the scenes can be entered, which shall be executed, when the scheduler-channel turns to either its active or inactive state.

Online



State

Value: Inactive

Remaining Time On: 00:00:00 hh:mm:ss

Remaining Time Off: 12.19:46:06 hh:mm:ss

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Displays the current state of the scheduler-channel.
- **Remaining Time On:** Remaining time in which the scheduler-channel is active.
- **Remaining Time Off:** Remaining time in which the scheduler-channel is inactive.

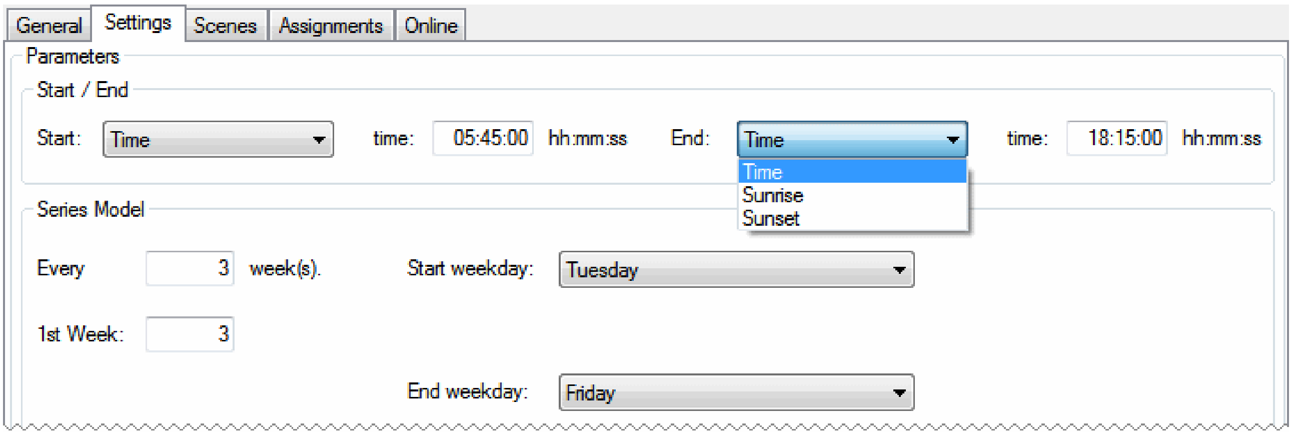
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.10.3 Weekly Time Scheduler Channel (continual days of the week)

The Weekly Time Scheduler Channel (continual days of the week) can be used for switching from one weekday until another.

Settings



Parameters

Start / End

Start: Time time: 05:45:00 hh:mm:ss End: Time time: 18:15:00 hh:mm:ss

Series Model

Every 3 week(s). Start weekday: Tuesday

1st Week: 3

End weekday: Friday

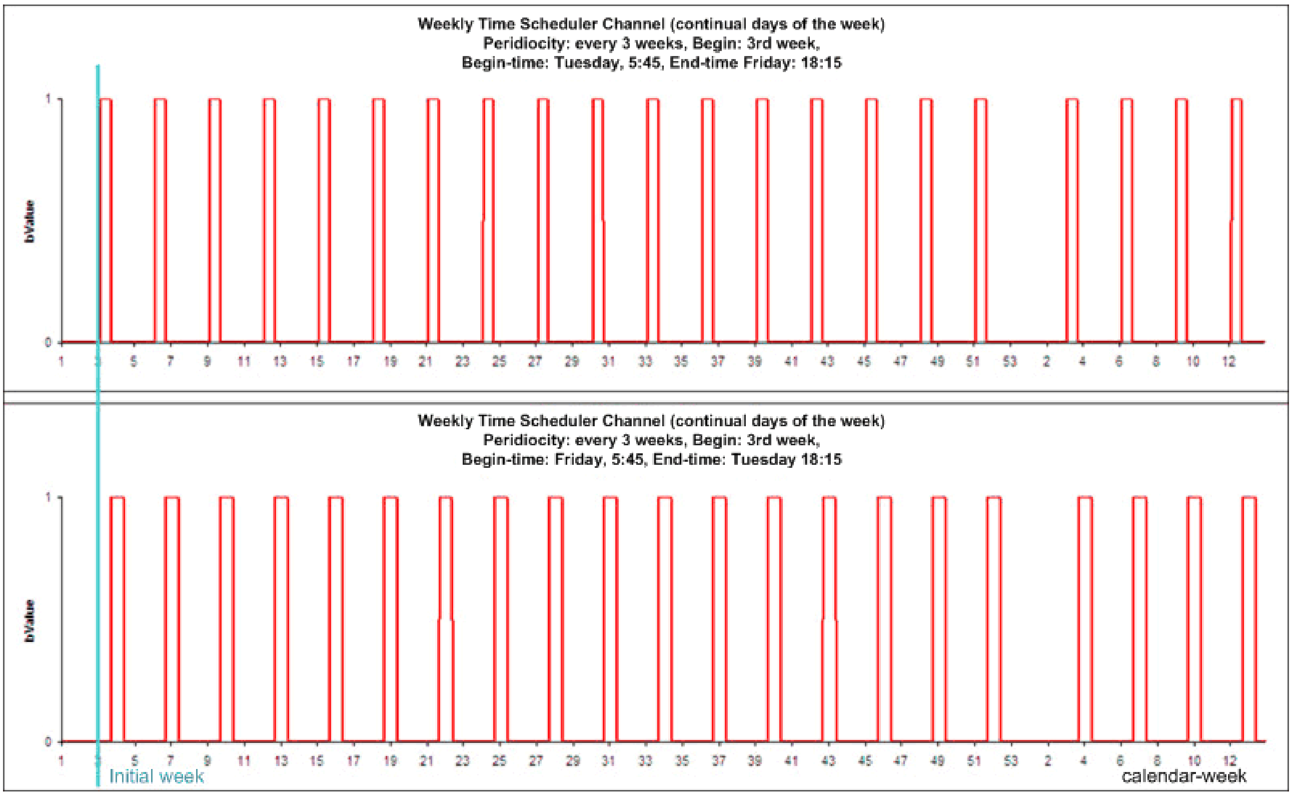
- **Start: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  Start: Sunrise + Offset: The scheduler-channel will be active beginning with the actual sunrise + offset (in minutes) on a switching-day.
  Start: Sunset + Offset: The scheduler-channel will be active beginning with the actual sunset + offset (in minutes) on a switching-day.
- **End: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  End: Sunrise + Offset: The scheduler-channel will be inactive beginning with the actual sunrise +

offset (in minutes) on a switching-day.
End: Sunset + Offset: The scheduler-channel will be inactive beginning with the actual sunset + offset (in minutes) on a switching-day.

Series Model

- **Every n weeks(s):** The week (beginning from Monday) in which the PLC is initialised, is always a switching-week. From this week on all other switching-weeks are calculated by this periodicity.
- **Monday...Sunday:** Selectable switching-days.

Behavior

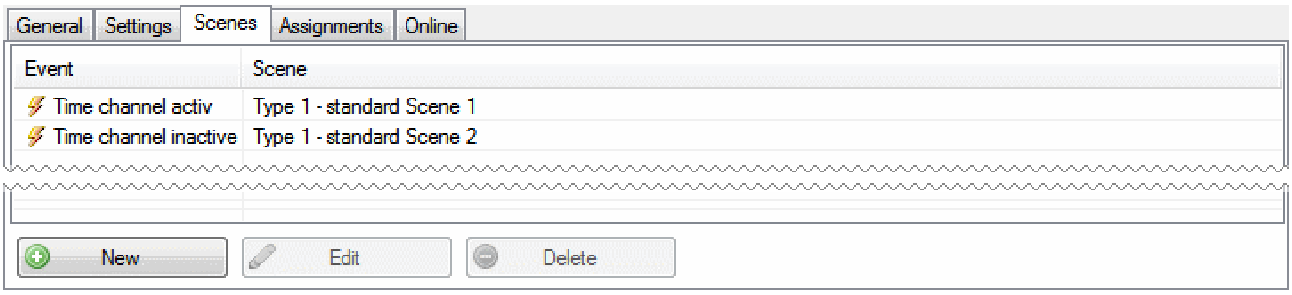


The periodicity is calculated from the beginning of the initial calendar-week (beginning with Monday, according to ISO 8601).

If the beginning-time is lower than the end-time (regarding the week), the time-channel will be activated and deactivated within one week. If the beginning-time is greater than the end-time the deactivation of the channel happens the week after activation. An equal beginning and end will result in a one-week activation of the channel.

It is also possible to use the sunrise/sunset function which will calculate both times for the switching-day together with an offset, which can be either positive or negative. These resulting start and end-time however are automatically delimited to the beginning (00:00:00:000) or the end of the day (23:59:59:999).

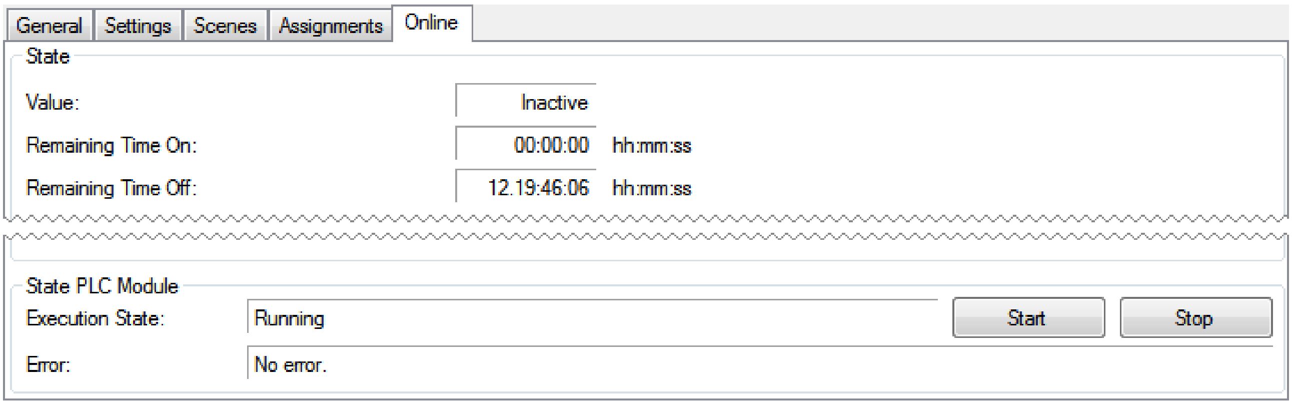
Scenes



Event	Scene
Time channel activ	Type 1 - standard Scene 1
Time channel inactive	Type 1 - standard Scene 2

On this screen the scenes can be entered, which shall be executed, when the scheduler-channel turns to either its active or inactive state.

Online



State

Value: Inactive

Remaining Time On: 00:00:00 hh:mm:ss

Remaining Time Off: 12.19:46:06 hh:mm:ss

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Displays the current state of the scheduler-channel.
- **Remaining Time On:** Remaining time in which the scheduler-channel is active.
- **Remaining Time Off:** Remaining time in which the scheduler-channel is inactive.

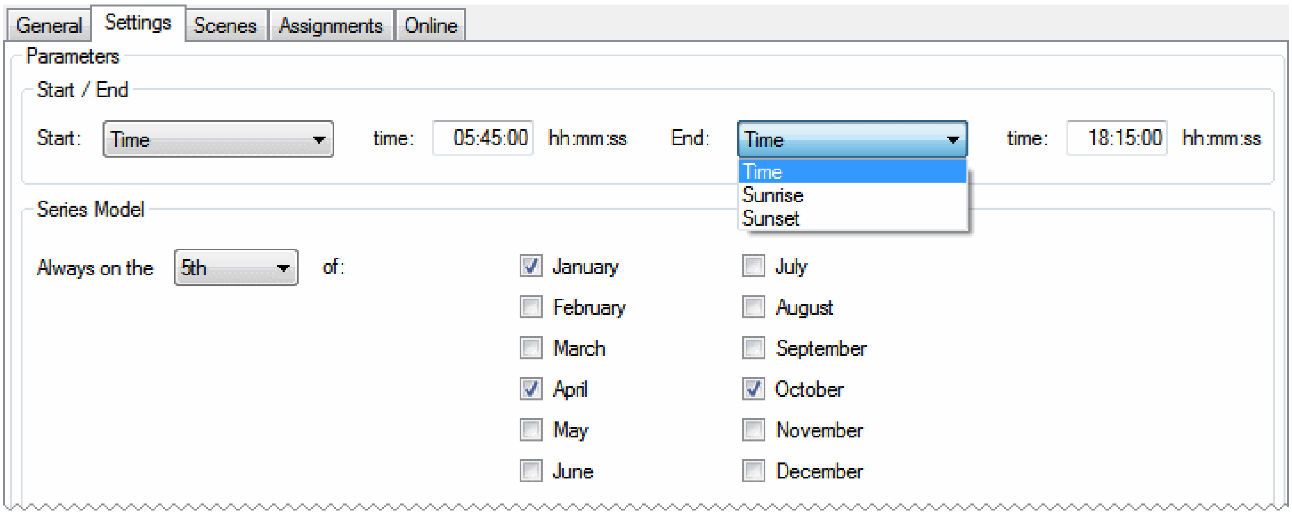
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.10.4 Monthly Time Scheduler Channel (nth day of the month)

With the Monthly Time Scheduler Channel (nth day of the month) it is possible to define a switching-event on a single day of the month in multiple months of the year.

Settings



Parameters

Start / End

Start: Time time: 05:45:00 hh:mm:ss End: Time time: 18:15:00 hh:mm:ss

Series Model

Always on the 5th of:

☑ January
□ February
□ March
☑ April
□ May
□ June
□ July
□ August
□ September
☑ October
□ November
□ December

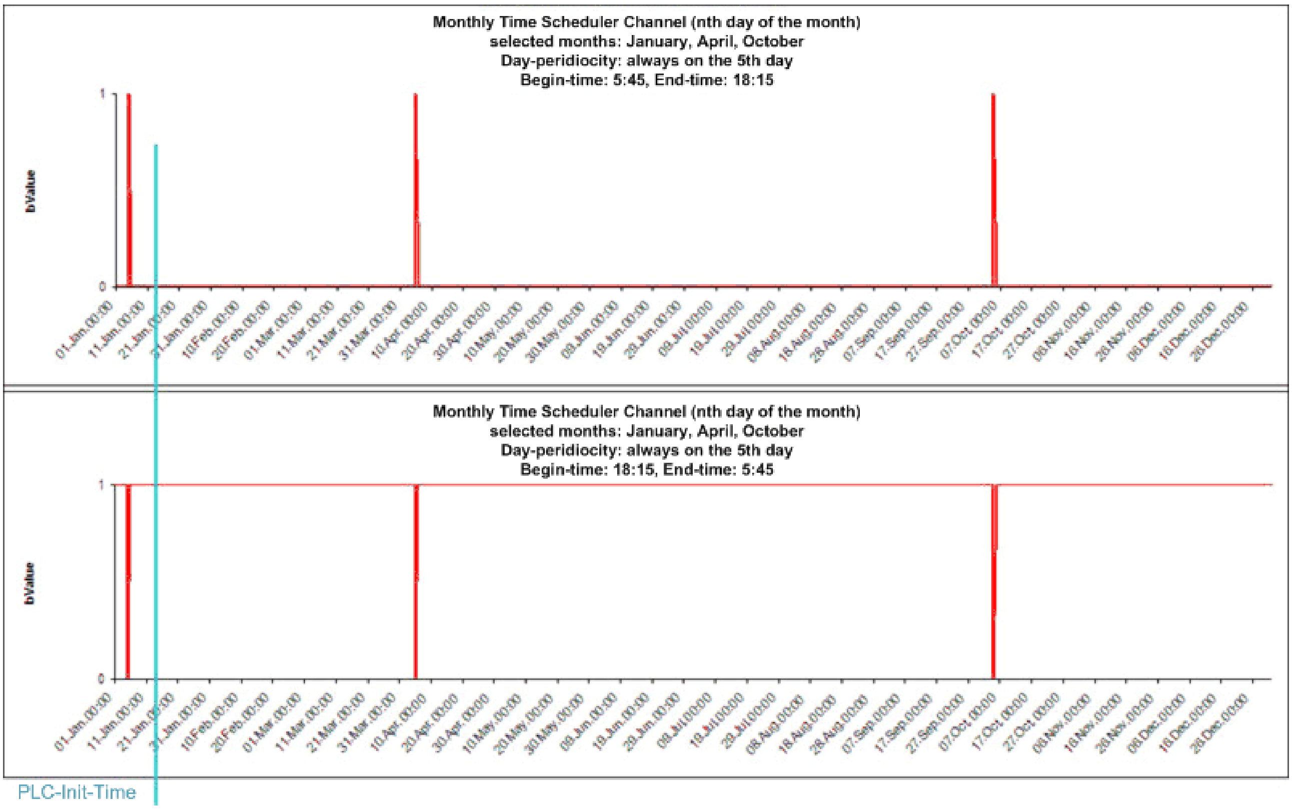
- **Start: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
Start: Sunrise + Offset: The scheduler-channel will be active beginning with the actual sunrise + offset (in minutes) on a switching-day.
Start: Sunset + Offset: The scheduler-channel will be active beginning with the actual sunset + offset (in minutes) on a switching-day.
- **End: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
End: Sunrise + Offset: The scheduler-channel will be inactive beginning with the actual sunrise +

offset (in minutes) on a switching-day.
End: Sunset + Offset: The scheduler-channel will be inactive beginning with the actual sunset + offset (in minutes) on a switching-day.

Series Model

- **Always on the nth of:** The explicitly selectable switching day (1st...31st).
- **January...December:** Selectable switching-months.

Behavior



Monthly Time Scheduler Channel (nth day of the month)
selected months: January, April, October
Day-peridiocity: always on the 5th day
Begin-time: 5:45, End-time: 18:15

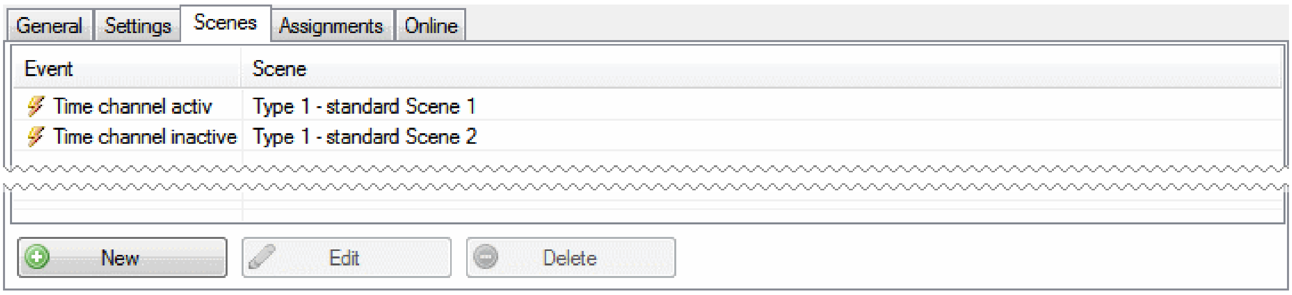
Monthly Time Scheduler Channel (nth day of the month)
selected months: January, April, October
Day-peridiocity: always on the 5th day
Begin-time: 18:15, End-time: 5:45

In contrary to the daily and weekly scheduler-functions, the switching days are explicitly named. There is no periodicity to be calculated, which depends on the PLC-init-time.

If the daytime of the start is lower than the end, the scheduler channel will only be active from the start-time until the end-time of the switching-day. If the daytime of the end is lower than the start, the scheduler-channel will just work inversely. Entering the same time for both start and end would result in an undefined-state and will thus be regarded as an input-error.

It is also possible to use the sunrise/sunset function which will calculate both times for the switching-day together with an offset, which can be either positive or negative. These resulting start and end-time however are automatically delimited to the beginning (00:00:00:000) or the end of the day (23:59:59:999).

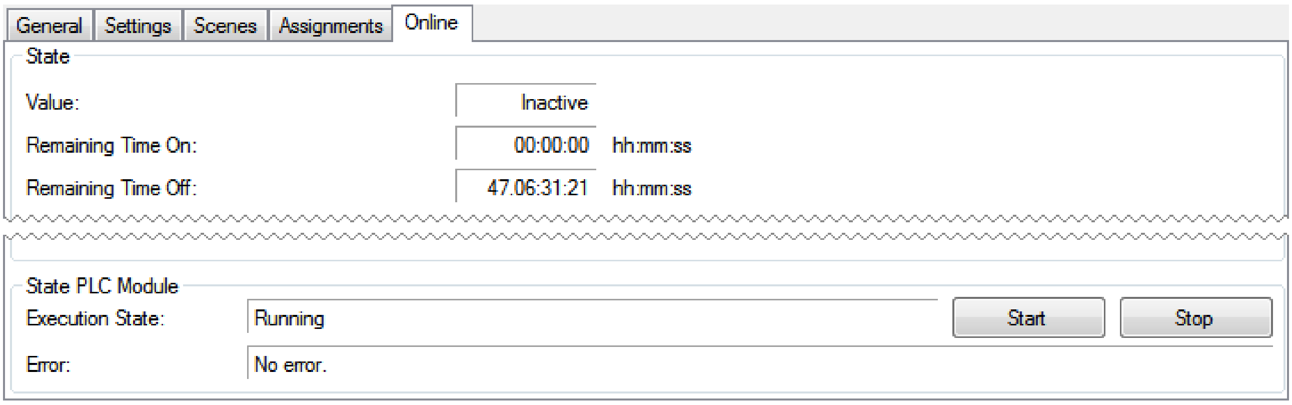
Scenes



Event	Scene
⚡ Time channel activ	Type 1 - standard Scene 1
⚡ Time channel inactive	Type 1 - standard Scene 2

On this screen the scenes can be entered, which shall be executed, when the scheduler-channel turns to either its active or inactive state.

Online



State

Value: Inactive

Remaining Time On: 00:00:00 hh:mm:ss

Remaining Time Off: 47.06:31:21 hh:mm:ss

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Displays the current state of the scheduler-channel.
- **Remaining Time On:** Remaining time in which the scheduler-channel is active.
- **Remaining Time Off:** Remaining time in which the scheduler-channel is inactive.

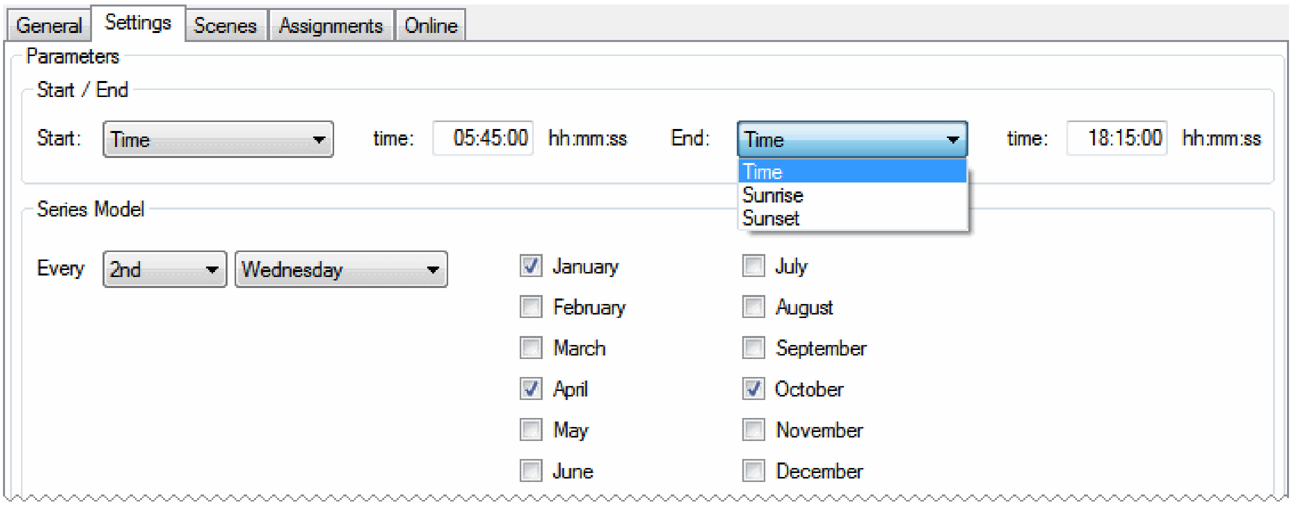
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.10.5 Monthly Time Scheduler Channel (a individual weekday of the month)

With the Monthly Time Scheduler Channel (a individual weekday of the month) it is possible to define a switching-event on a single weekday in multiple months of the year.

Settings



Parameters

Start / End

Start: Time time: 05:45:00 hh:mm:ss End: Time time: 18:15:00 hh:mm:ss

Series Model

Every 2nd Wednesday

☑ January
☐ February
☐ March
☑ April
☐ May
☐ June
☐ July
☐ August
☐ September
☑ October
☐ November
☐ December

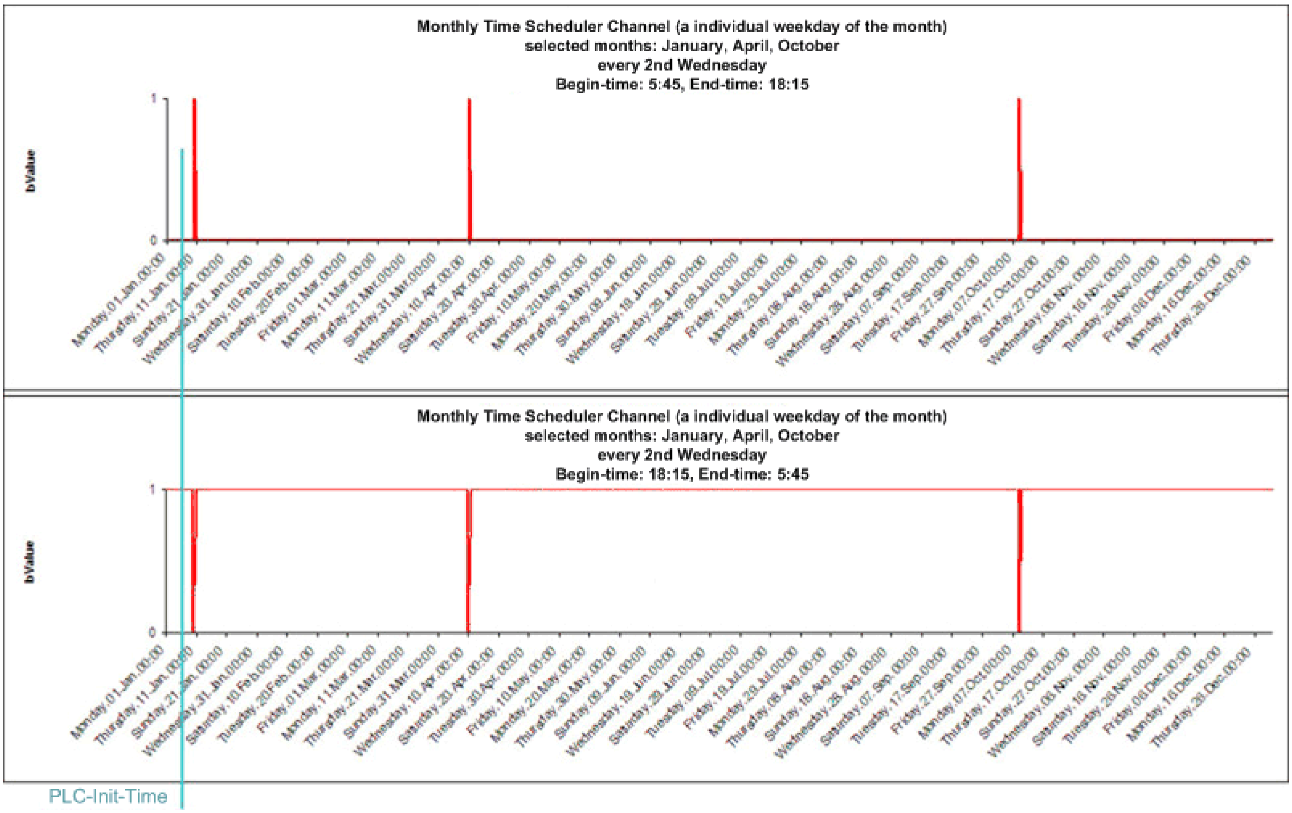
- **Start: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  Start: Sunrise + Offset: The scheduler-channel will be active beginning with the actual sunrise + offset (in minutes) on a switching-day.
  Start: Sunset + Offset: The scheduler-channel will be active beginning with the actual sunset + offset (in minutes) on a switching-day.
- **End: Time:** Absolute start-time on a switching-day. The scheduler-channel will be active beginning with this time.
  End: Sunrise + Offset: The scheduler-channel will be inactive beginning with the actual sunrise +

offset (in minutes) on a switching-day.
End: Sunset + Offset: The scheduler-channel will be inactive beginning with the actual sunset + offset (in minutes) on a switching-day.

Series Model

- **Every nth Weekday:** The explicitly selectable switching day (1st...4th Monday...Sunday).
- **January...December:** Selectable switching-months.

Behavior

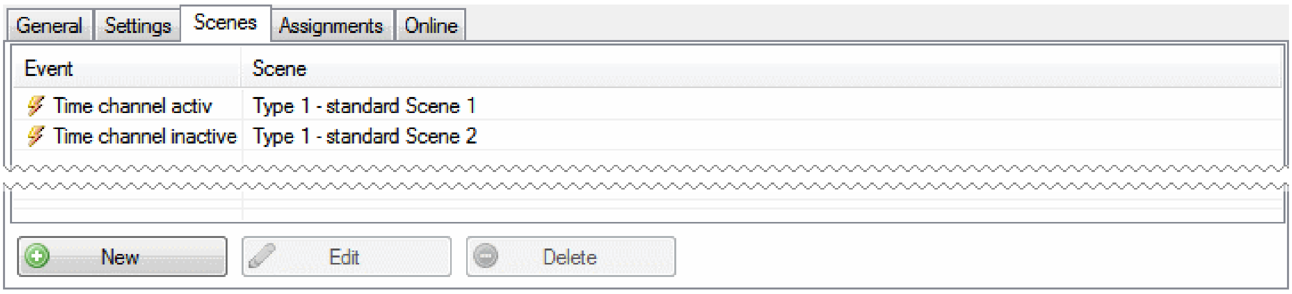


In contrary to the daily and weekly scheduler-functions, the switching days are explicitly named. There is no periodicity to be calculated, which depends on the PLC-init-time.

If the daytime of the start is lower than the end, the scheduler channel will only be active from the start-time until the end-time of the switching-day. If the daytime of the end is lower than the start, the scheduler-channel will just work inversely. Entering the same time for both start and end would result in an undefined-state and will thus be regarded as an input-error.

It is also possible to use the sunrise/sunset function which will calculate both times for the switching-day together with an offset, which can be either positive or negative. These resulting start and end-time however are automatically delimited to the beginning (00:00:00:000) or the end of the day (23:59:59:999).

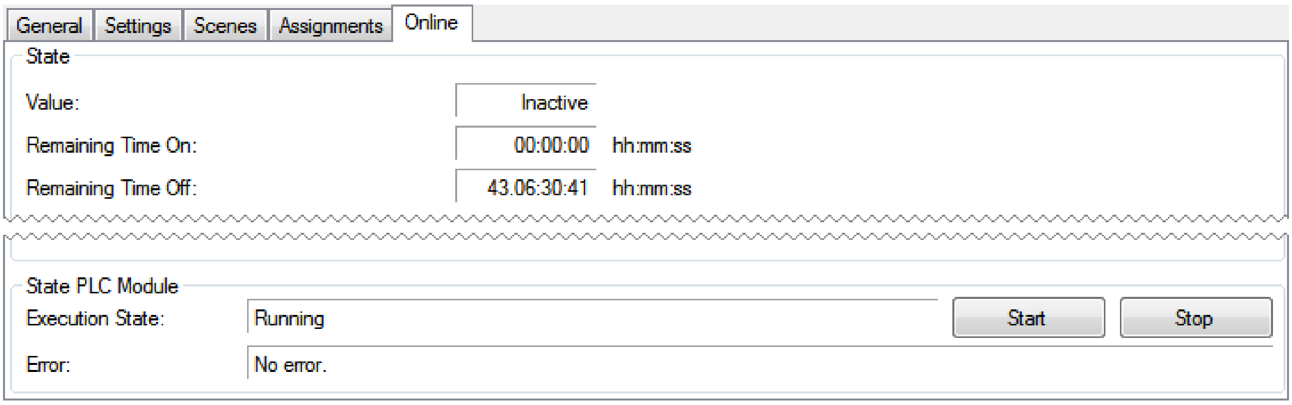
Scenes



Event	Scene
Time channel activ	Type 1 - standard Scene 1
Time channel inactive	Type 1 - standard Scene 2

On this screen the scenes can be entered, which shall be executed, when the scheduler-channel turns to either its active or inactive state.

Online



State

Value: Inactive

Remaining Time On: 00:00:00 hh:mm:ss

Remaining Time Off: 43.06:30:41 hh:mm:ss

State PLC Module

Execution State: Running

Error: No error.

State

- **Value:** Displays the current state of the scheduler-channel.
- **Remaining Time On:** Remaining time in which the scheduler-channel is active.
- **Remaining Time Off:** Remaining time in which the scheduler-channel is inactive.

State PLC Module

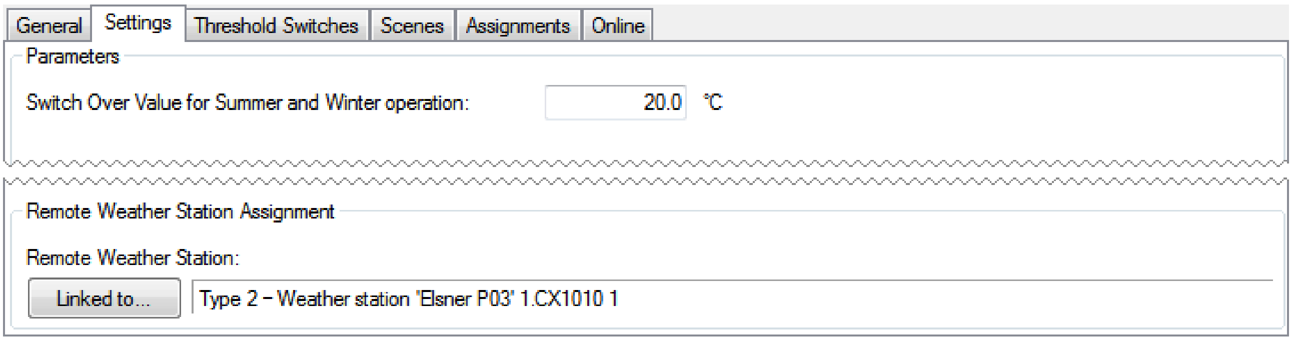
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.11 Weather Stations

7.2.2.11.1 Virtual weather station

The virtual weather station permits access to the measured values from a remotely connected weather station.

Settings



Parameters

Switch Over Value for Summer and Winter operation: 20.0 °C

Remote Weather Station Assignment

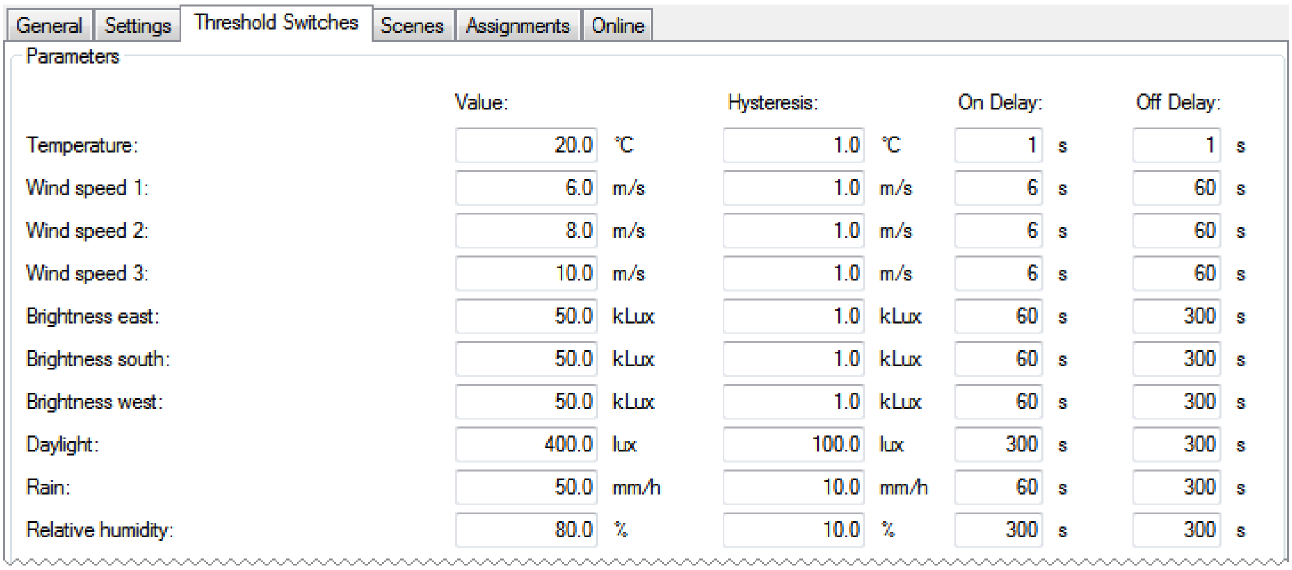
Remote Weather Station: Type 2 – Weather station 'Elsner P03' 1.CX1010 1

Parameter

- **Summer mode/winter mode switchover value:** threshold value for the recognition of a change of season.

Pressing the *Linked with…* button opens a dialogue in which the remote weather station can be linked with this module.

Threshold switch



Parameters

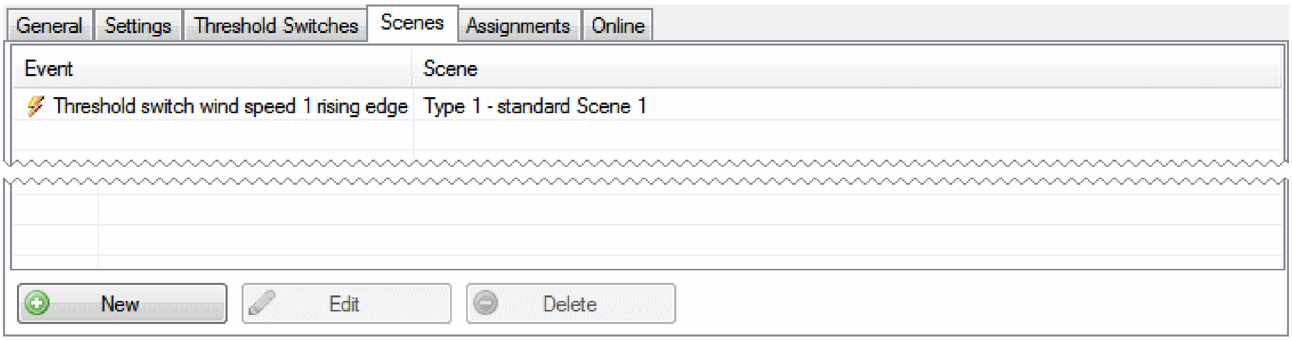
	Value:		Hysteresis:		On Delay:		Off Delay:	
Temperature:	20.0	°C	1.0	°C	1	s	1	s
Wind speed 1:	6.0	m/s	1.0	m/s	6	s	60	s
Wind speed 2:	8.0	m/s	1.0	m/s	6	s	60	s
Wind speed 3:	10.0	m/s	1.0	m/s	6	s	60	s
Brightness east:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness south:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness west:	50.0	kLux	1.0	kLux	60	s	300	s
Daylight:	400.0	lux	100.0	lux	300	s	300	s
Rain:	50.0	mm/h	10.0	mm/h	60	s	300	s
Relative humidity:	80.0	%	10.0	%	300	s	300	s

Threshold switches can be defined in relation to the individual values. There is an option in the *Scene* dialogue (see below) to call scenes via the threshold switches. Hence, for example, a scene can be created that drives all blinds upwards. The 1st threshold switch for the wind velocity switches on at 6.0 m/s (hysteresis 1 m/s). The switch-on delay of 6 seconds ensures that short storm gusts are suppressed. An event is created in the Scenes dialogue that calls the appropriate scene when the threshold switch is tripped and drives the shutters upwards as a result.

Parameter

- **Temperature:** threshold value for the temperature.
- **Wind speed 1,**
  Wind speed 2,
  Wind speed 3: threshold values for the wind velocity.
- **Brightness east,**
  Brightness south,
  Brightness west: threshold values for the brightness.
- **Twilight:** threshold value for the twilight.
- **Precipitation:** threshold value for rainfall.
- **Relative air humidity:** threshold value for the relative humidity.

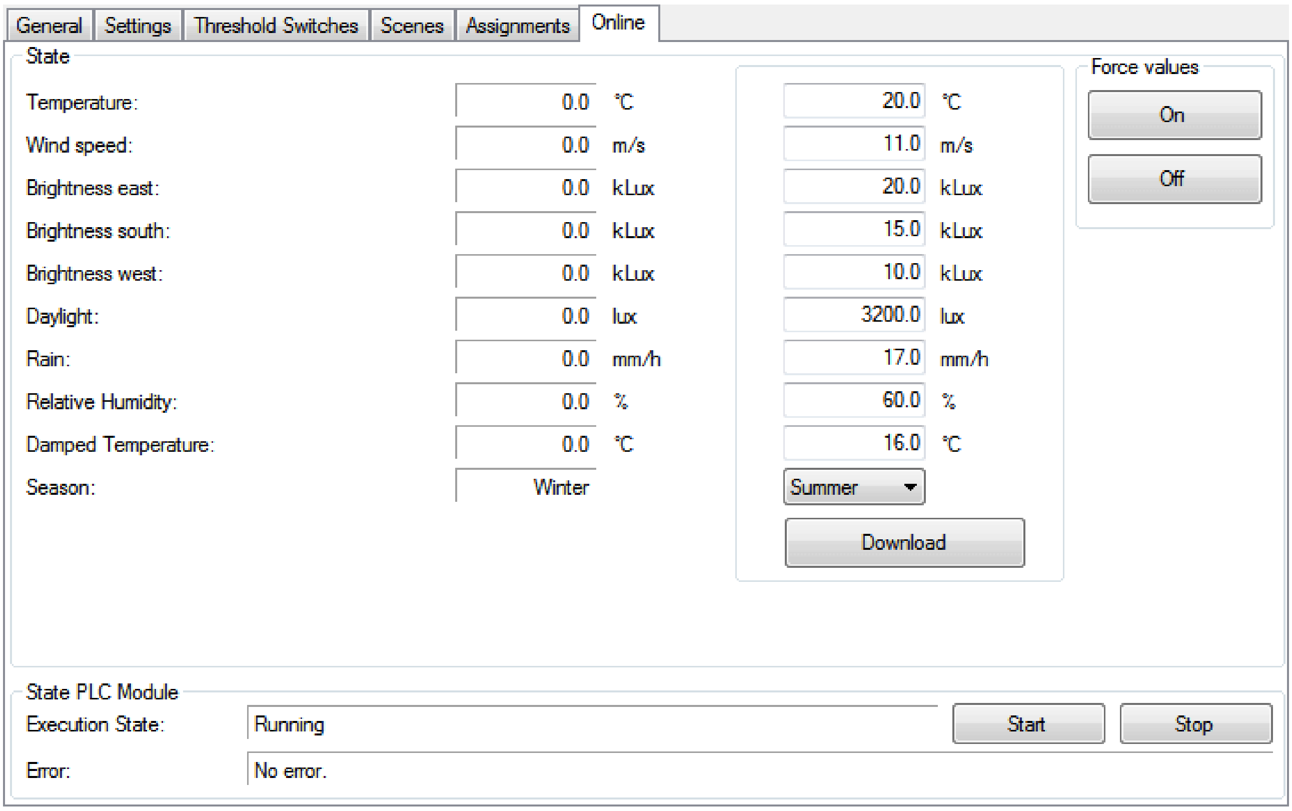
Scenes



Event	Scene
⚡ Threshold switch wind speed 1 rising edge	Type 1 - standard Scene 1

Pressing the *New* button in the *Scenes* menu opens a dialogue in which the rising and falling edges of the threshold switches can be used to call a scene. To do this the scene must have been created beforehand.

Online



State				
Temperature:	0.0	°C	20.0	°C
Wind speed:	0.0	m/s	11.0	m/s
Brightness east:	0.0	kLux	20.0	kLux
Brightness south:	0.0	kLux	15.0	kLux
Brightness west:	0.0	kLux	10.0	kLux
Daylight:	0.0	lux	3200.0	lux
Rain:	0.0	mm/h	17.0	mm/h
Relative Humidity:	0.0	%	60.0	%
Damped Temperature:	0.0	°C	16.0	°C
Season:	Winter		Summer	

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Force values on/off:** activates or deactivates the forcing of the values. The value entered can be written to the PLC by pressing the *Write down* button.

State

- **Temperature:** current outside temperature.
- **Wind speed:** momentary wind velocity.
- **Brightness east:** brightness on the east side.
- **Brightness south:** brightness on the south side.
- **Brightness west:** brightness on the west side.
- **Twilight:** indicates the daylight level.
- **Precipitation:** Indicates rainfall.
- **Relative air humidity:** measured relative air humidity.
- **Damped temperature:** average temperatures measured over the course of the day.
- **Season:** indicates the season in relation to the *damped temperature*.

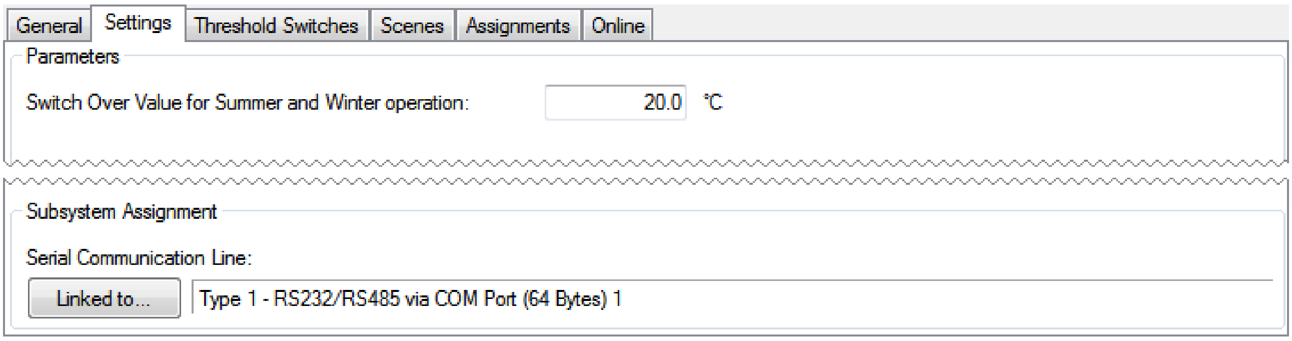
State PLC Module

- **Execution State:** shows if the module is running or stopped.
- **Error:** in case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** starts/stops the module.

7.2.2.11.2 Weather station type ‘Elsner P03’

Acquisition of different meteorological data with the Elsner P03, such as temperature, wind speed or brightness.

Settings



Parameters

Switch Over Value for Summer and Winter operation: 20.0 °C

Subsystem Assignment

Serial Communication Line:

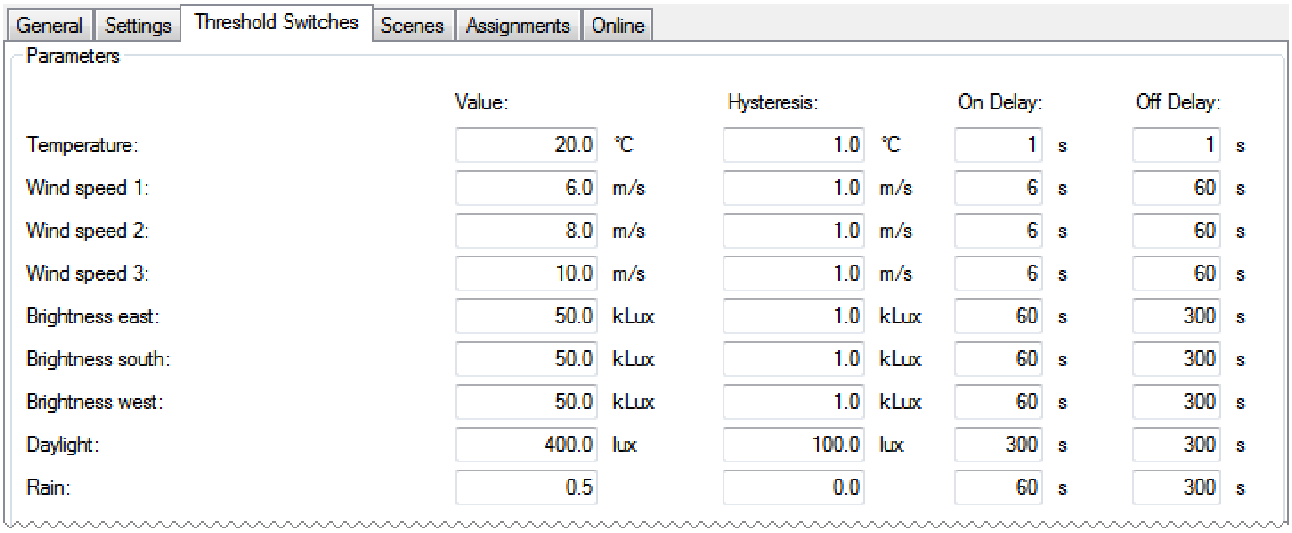
Type 1 - RS232/RS485 via COM Port (64 Bytes) 1

Parameter

- **Summer mode/winter mode switchover value:** Threshold value for the recognition of a change of season.

Pressing the *Linked with…* button opens a dialogue in which the serial communication line can be linked to this module.

Threshold switch



Parameters

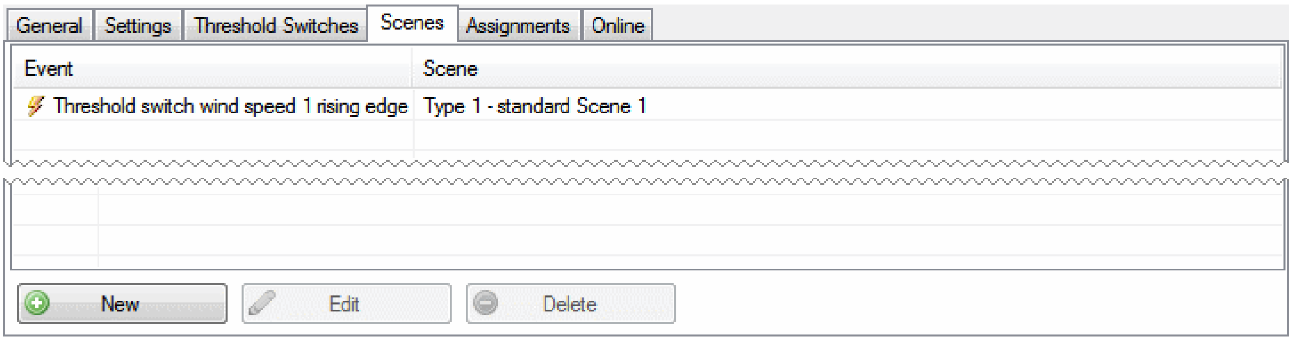
	Value:		Hysteresis:		On Delay:		Off Delay:	
Temperature:	20.0	°C	1.0	°C	1	s	1	s
Wind speed 1:	6.0	m/s	1.0	m/s	6	s	60	s
Wind speed 2:	8.0	m/s	1.0	m/s	6	s	60	s
Wind speed 3:	10.0	m/s	1.0	m/s	6	s	60	s
Brightness east:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness south:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness west:	50.0	kLux	1.0	kLux	60	s	300	s
Daylight:	400.0	lux	100.0	lux	300	s	300	s
Rain:	0.5		0.0		60	s	300	s

Threshold switches can be defined in relation to the individual values. There is an option in the *Scene* dialogue (see below) to call scenes via the threshold switches. Hence, for example, a scene can be created that drives all blinds upwards. The 1st threshold switch for the wind velocity switches on at 6.0 m/s (hysteresis 1 m/s). The switch-on delay of 6 seconds ensures that short storm gusts are suppressed. An event is created in the Scenes dialogue that calls the appropriate scene when the threshold switch is tripped and drives the shutters upwards as a result.

Parameter

- **Temperature:** threshold value for the temperature.
- **Wind speed 1,**
  Wind speed 2,
  Wind speed 3: threshold values for the wind velocity.
- **Brightness east,**
  Brightness south,
  Brightness west: threshold values for the brightness.
- **Twilight:** threshold value for the twilight.
- **Precipitation:** threshold value for rainfall. Please note that with this weather station the precipitation can only adopt the values 0.0 (no rainfall) or 1.0 (rainfall).

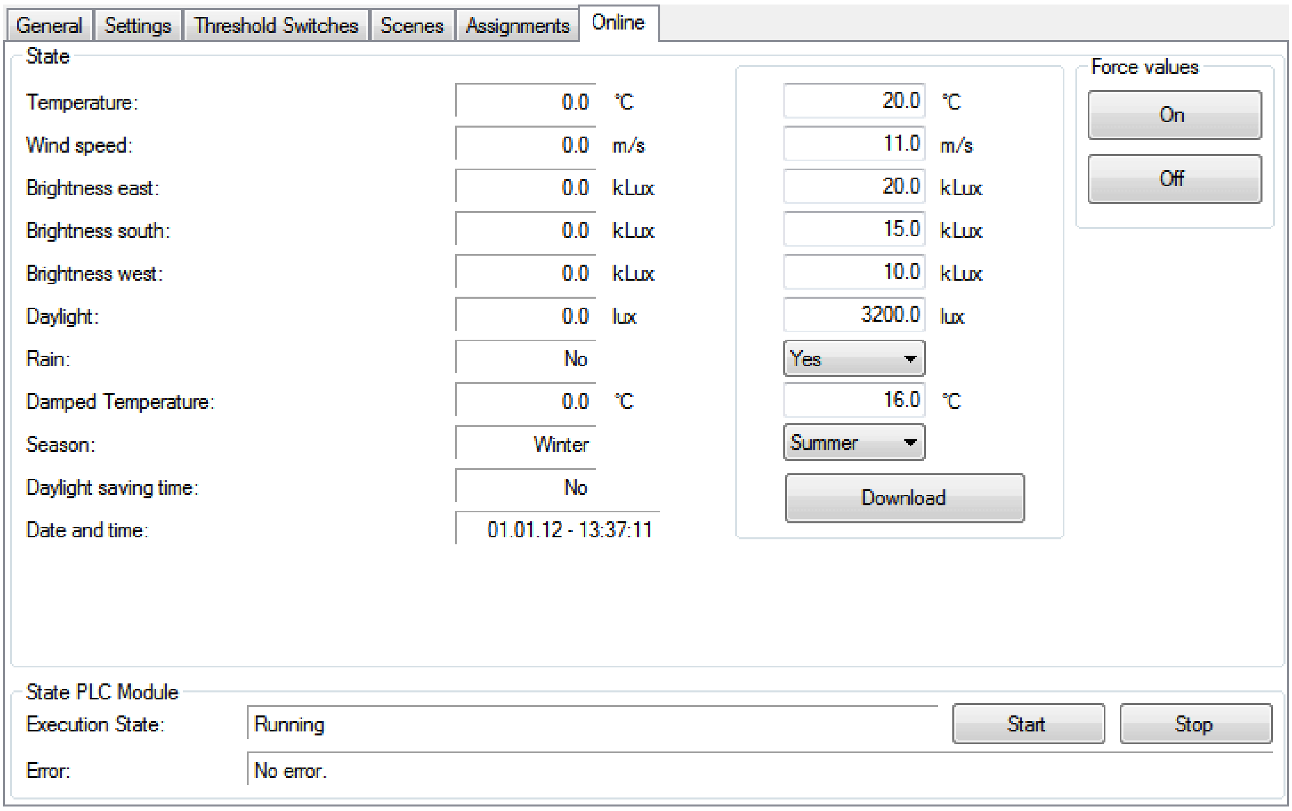
Scenes



Event	Scene
Threshold switch wind speed 1 rising edge	Type 1 - standard Scene 1

Pressing the *New* button in the *Scenes* menu opens a further dialogue in which the rising and falling edges of the threshold switches can be used to call a scene. To do this the scene must have been created beforehand.

Online



State

Temperature:	0.0	°C	20.0	°C
Wind speed:	0.0	m/s	11.0	m/s
Brightness east:	0.0	kLux	20.0	kLux
Brightness south:	0.0	kLux	15.0	kLux
Brightness west:	0.0	kLux	10.0	kLux
Daylight:	0.0	lux	3200.0	lux
Rain:	No		Yes	
Damped Temperature:	0.0	°C	16.0	°C
Season:	Winter		Summer	
Daylight saving time:	No			
Date and time:	01.01.12 - 13:37:11			

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Force values on/off:** activates or deactivates the forcing of the values. The value entered can be written to the PLC by pressing the *Write down* button.

State

- **Temperature:** current outside temperature.
- **Wind speed:** momentary wind velocity.
- **Brightness east:** brightness on the east side.
- **Brightness south:** brightness on the south side.
- **Brightness west:** brightness on the west side.
- **Twilight:** indicates the daylight level.
- **Precipitation:** indicates rainfall.
- **Damped temperature:** average temperatures measured over the course of the day.
- **Season:** indicates the season in relation to the *damped temperature*.

- **Daylight saving time:** indicates whether daylight saving time is active.
- **Date and time:** Date and time received by the time signal transmitter DCF77.

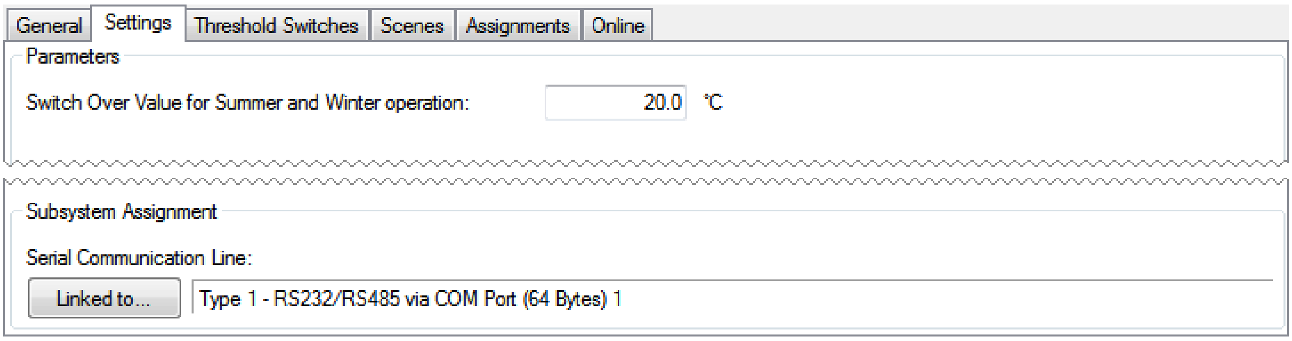
State PLC Module

- **Execution State:** shows if the module is running or stopped.
- **Error:** in case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** starts/stops the module.

7.2.2.11.3 Weather station type ‘Thies Clima Sensor D’

Acquisition of different meteorological data with the Thies Clima Sensor D, such as temperature, wind speed or brightness.

Settings



Parameters

Switch Over Value for Summer and Winter operation: 20.0 °C

Subsystem Assignment

Serial Communication Line:

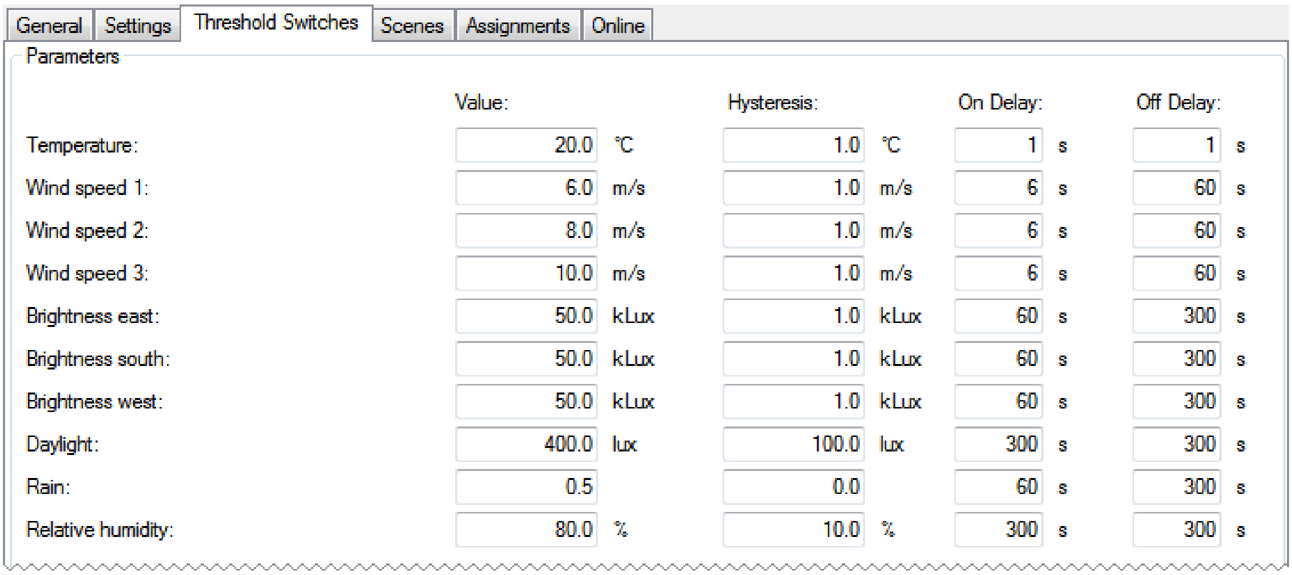
Type 1 - RS232/RS485 via COM Port (64 Bytes) 1

Parameter

- **Summer mode/winter mode switchover value:** threshold value for the recognition of a change of season.

Pressing the *Linked with…* button opens a dialogue in which the serial communication line can be linked to this module.

Threshold switch



Parameters

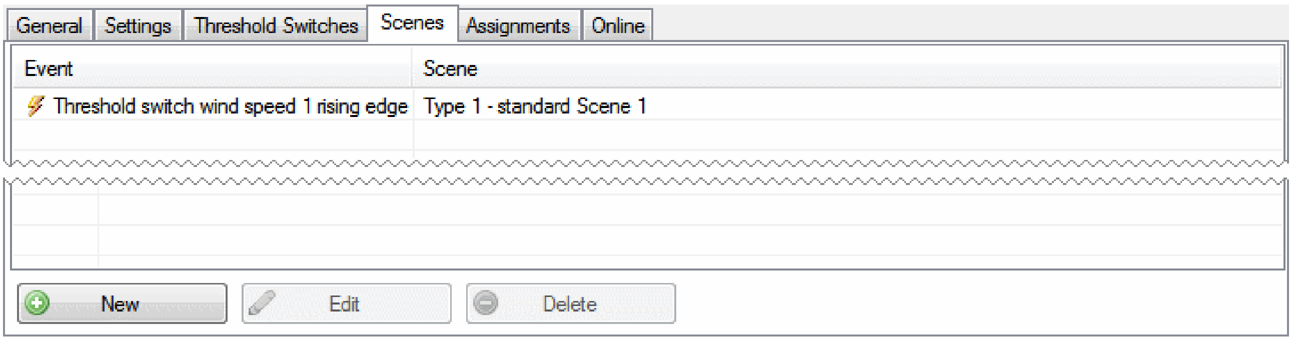
	Value:		Hysteresis:		On Delay:		Off Delay:	
Temperature:	20.0	°C	1.0	°C	1	s	1	s
Wind speed 1:	6.0	m/s	1.0	m/s	6	s	60	s
Wind speed 2:	8.0	m/s	1.0	m/s	6	s	60	s
Wind speed 3:	10.0	m/s	1.0	m/s	6	s	60	s
Brightness east:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness south:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness west:	50.0	kLux	1.0	kLux	60	s	300	s
Daylight:	400.0	lux	100.0	lux	300	s	300	s
Rain:	0.5		0.0		60	s	300	s
Relative humidity:	80.0	%	10.0	%	300	s	300	s

Threshold switches can be defined in relation to the individual values. There is an option in the *Scene* dialogue (see below) to call scenes via the threshold switches. Hence, for example, a scene can be created that drives all blinds upwards. The 1st threshold switch for the wind velocity switches on at 6.0 m/s (hysteresis 1 m/s). The switch-on delay of 6 seconds ensures that short storm gusts are suppressed. An event is created in the Scenes dialogue that calls the appropriate scene when the threshold switch is tripped and drives the shutters upwards as a result.

Parameter

- **Temperature:** threshold value for the temperature.
- **Wind speed 1,**
  Wind speed 2,
  Wind speed 3: threshold values for the wind velocity.
- **Brightness east,**
  Brightness south,
  Brightness west: threshold values for the brightness.
- **Twilight:** threshold value for the twilight.
- **Precipitation:** threshold value for rainfall. Please note that with this weather station the precipitation can only adopt the values 0.0 (no rainfall) or 1.0 (rainfall).
- **Relative air humidity:** threshold value for the relative air humidity.

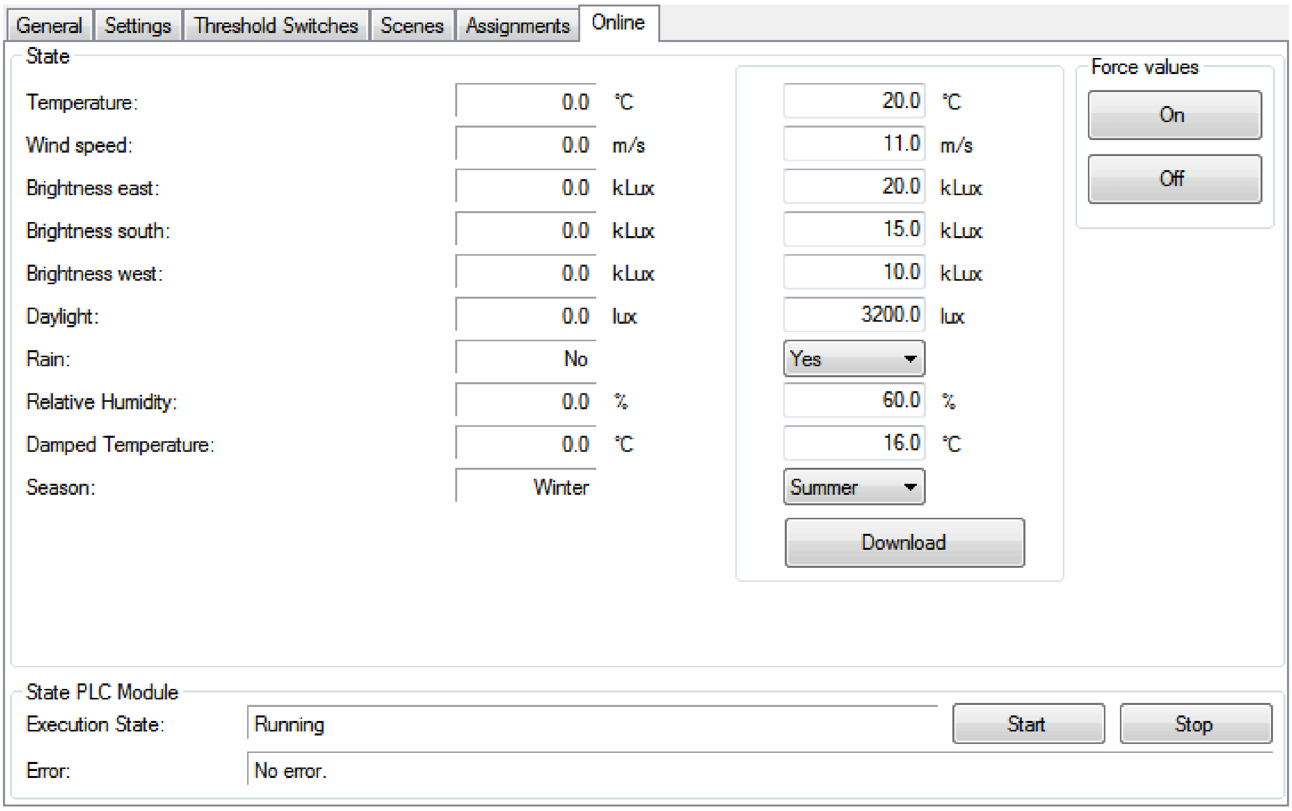
Scenes



Event	Scene
Threshold switch wind speed 1 rising edge	Type 1 - standard Scene 1

Pressing the *New* button in the *Scenes* menu opens a dialogue in which the rising and falling edges of the threshold switches can be used to call a scene. To do this the scene must have been created beforehand.

Online



Temperature:	0.0	°C	20.0	°C
Wind speed:	0.0	m/s	11.0	m/s
Brightness east:	0.0	kLux	20.0	kLux
Brightness south:	0.0	kLux	15.0	kLux
Brightness west:	0.0	kLux	10.0	kLux
Daylight:	0.0	lux	3200.0	lux
Rain:	No		Yes	
Relative Humidity:	0.0	%	60.0	%
Damped Temperature:	0.0	°C	16.0	°C
Season:	Winter		Summer	

Execution State: Running

Error: No error.

Commands

- **Force values on/off:** activates or deactivates the forcing of the values. The value entered can be written to the PLC by pressing the *Write down* button.

State

- **Temperature:** current outside temperature.
- **Wind speed:** momentary wind velocity.
- **Brightness east:** brightness on the east side.
- **Brightness south:** brightness on the south side.
- **Brightness west:** brightness on the west side.
- **Twilight:** indicates the daylight level.
- **Precipitation:** Indicates rainfall.
- **Relative air humidity:** measured relative air humidity.
- **Damped temperature:** average temperatures measured over the course of the day.
- **Season:** indicates the season in relation to the *damped temperature*.

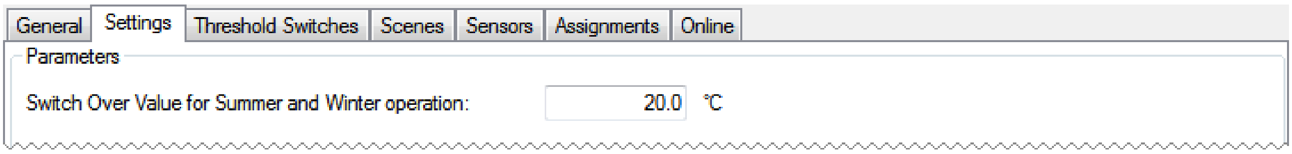
State PLC Module

- **Execution State:** shows if the module is running or stopped.
- **Error:** in case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** starts/stops the module.

7.2.2.11.4 Weather station with analog/digital signals

Acquisition of different meteorological data with analog and digital signals such as temperature, wind speed or brightness.

Settings

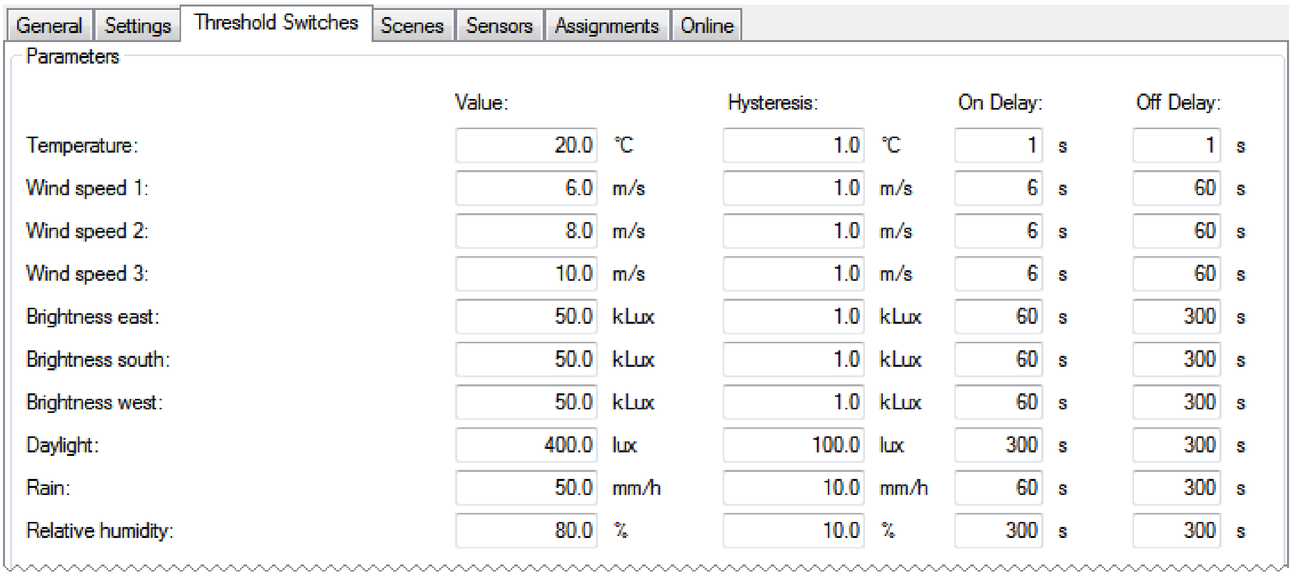


Switch Over Value for Summer and Winter operation: 20.0 °C

Parameter

- **Summer mode/winter mode switchover value:** threshold value for the recognition of a change of season.

Threshold switch



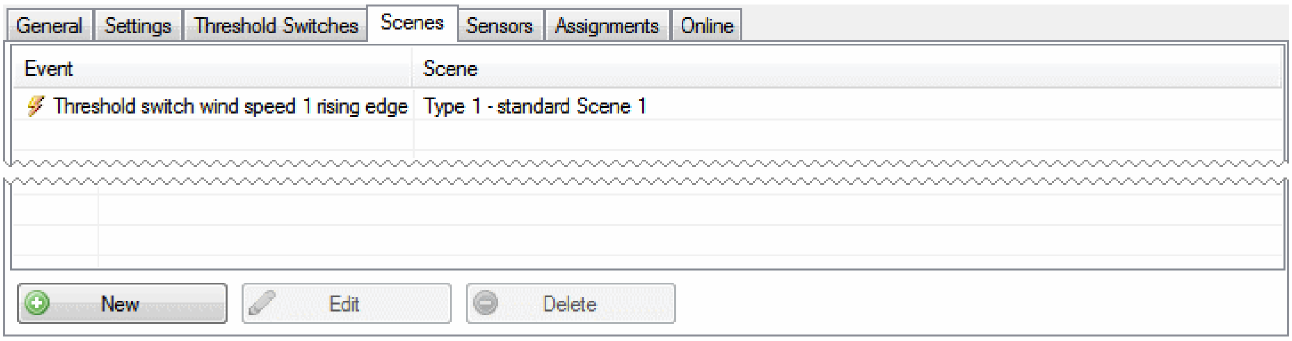
	Value:		Hysteresis:		On Delay:		Off Delay:	
Temperature:	20.0	°C	1.0	°C	1	s	1	s
Wind speed 1:	6.0	m/s	1.0	m/s	6	s	60	s
Wind speed 2:	8.0	m/s	1.0	m/s	6	s	60	s
Wind speed 3:	10.0	m/s	1.0	m/s	6	s	60	s
Brightness east:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness south:	50.0	kLux	1.0	kLux	60	s	300	s
Brightness west:	50.0	kLux	1.0	kLux	60	s	300	s
Daylight:	400.0	lux	100.0	lux	300	s	300	s
Rain:	50.0	mm/h	10.0	mm/h	60	s	300	s
Relative humidity:	80.0	%	10.0	%	300	s	300	s

Threshold switches can be defined in relation to the individual values. There is an option in the *Scene* dialogue (see below) to call scenes via the threshold switches. Hence, for example, a scene can be created that drives all blinds upwards. The 1st threshold switch for the wind velocity switches on at 6.0 m/s (hysteresis 1 m/s). The switch-on delay of 6 seconds ensures that short storm gusts are suppressed. An event is created in the Scenes dialogue that calls the appropriate scene when the threshold switch is tripped and drives the shutters upwards as a result.

Parameter

- **Temperature:** threshold value for the temperature.
- **Wind speed 1,**
  Wind speed 2,
  Wind speed 3: threshold values for the wind velocity.
- **Brightness east,**
  Brightness south,
  Brightness west: threshold values for the brightness.
- **Twilight:** threshold value for the twilight.
- **Precipitation:** threshold value for rainfall.
- **Relative air humidity:** threshold value for the relative air humidity.

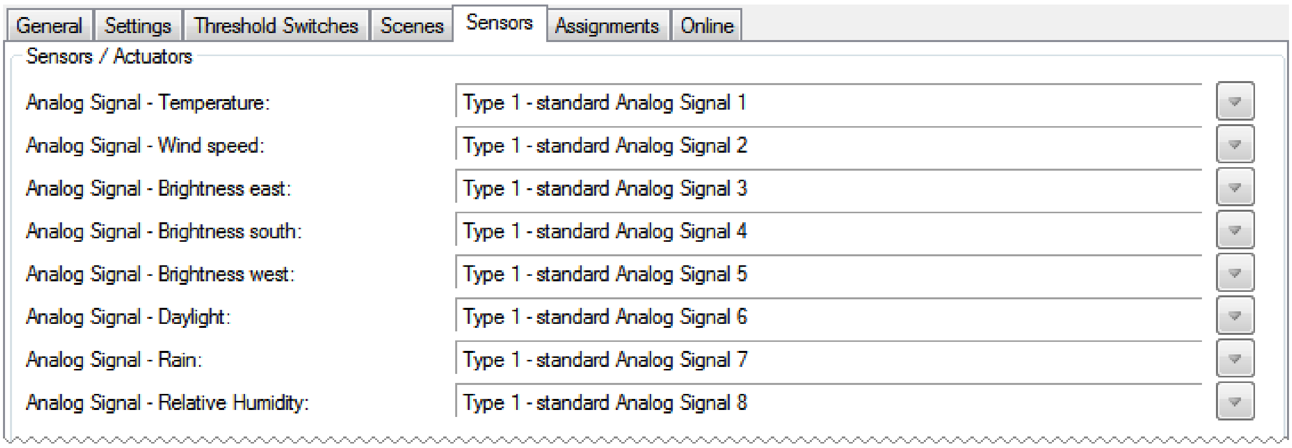
Scenes



Event	Scene
Threshold switch wind speed 1 rising edge	Type 1 - standard Scene 1

Pressing the *New* button in the *Scenes* menu opens a dialogue in which the rising and falling edges of the threshold switches can be used to call a scene. To do this the scene must have been created beforehand.

Sensors

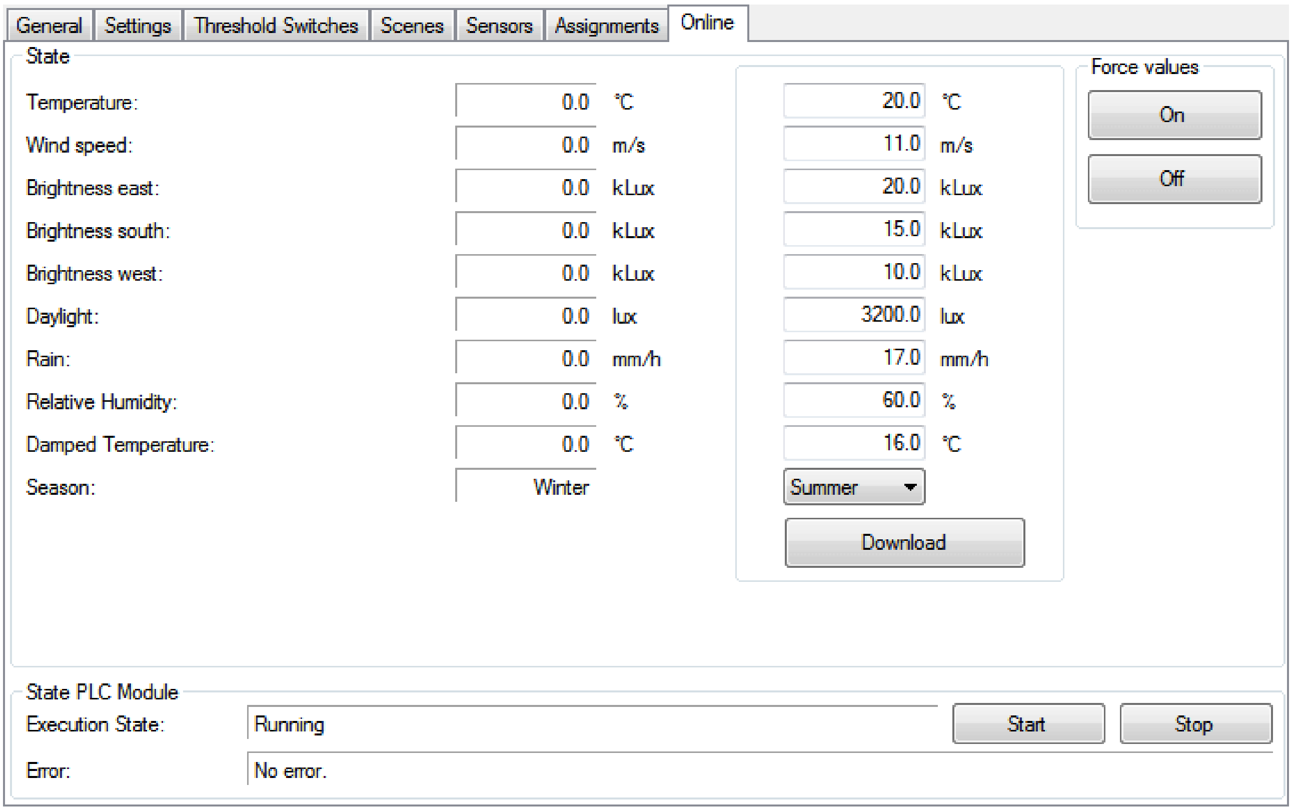


Sensors / Actuators

Analog Signal - Temperature:	Type 1 - standard Analog Signal 1
Analog Signal - Wind speed:	Type 1 - standard Analog Signal 2
Analog Signal - Brightness east:	Type 1 - standard Analog Signal 3
Analog Signal - Brightness south:	Type 1 - standard Analog Signal 4
Analog Signal - Brightness west:	Type 1 - standard Analog Signal 5
Analog Signal - Daylight:	Type 1 - standard Analog Signal 6
Analog Signal - Rain:	Type 1 - standard Analog Signal 7
Analog Signal - Relative Humidity:	Type 1 - standard Analog Signal 8

All signals to be measured must be entered in this menu.

Online



State

Temperature:	0.0	°C	20.0	°C
Wind speed:	0.0	m/s	11.0	m/s
Brightness east:	0.0	kLux	20.0	kLux
Brightness south:	0.0	kLux	15.0	kLux
Brightness west:	0.0	kLux	10.0	kLux
Daylight:	0.0	lux	3200.0	lux
Rain:	0.0	mm/h	17.0	mm/h
Relative Humidity:	0.0	%	60.0	%
Damped Temperature:	0.0	°C	16.0	°C
Season:	Winter		Summer	

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Force values on/off:** activates or deactivates the forcing of the values. The value entered can be written to the PLC by pressing the *Write down* button.

State

- **Temperature:** current outside temperature.
- **Wind speed:** momentary wind velocity.
- **Brightness east:** brightness on the east side.
- **Brightness south:** brightness on the south side.
- **Brightness west:** brightness on the west side.
- **Twilight:** indicates the daylight level.
- **Precipitation:** indicates rainfall.
- **Relative air humidity:** measured relative air humidity.
- **Damped temperature:** average temperatures measured over the course of the day.
- **Season:** indicates the season in relation to the *damped temperature*.

State PLC Module

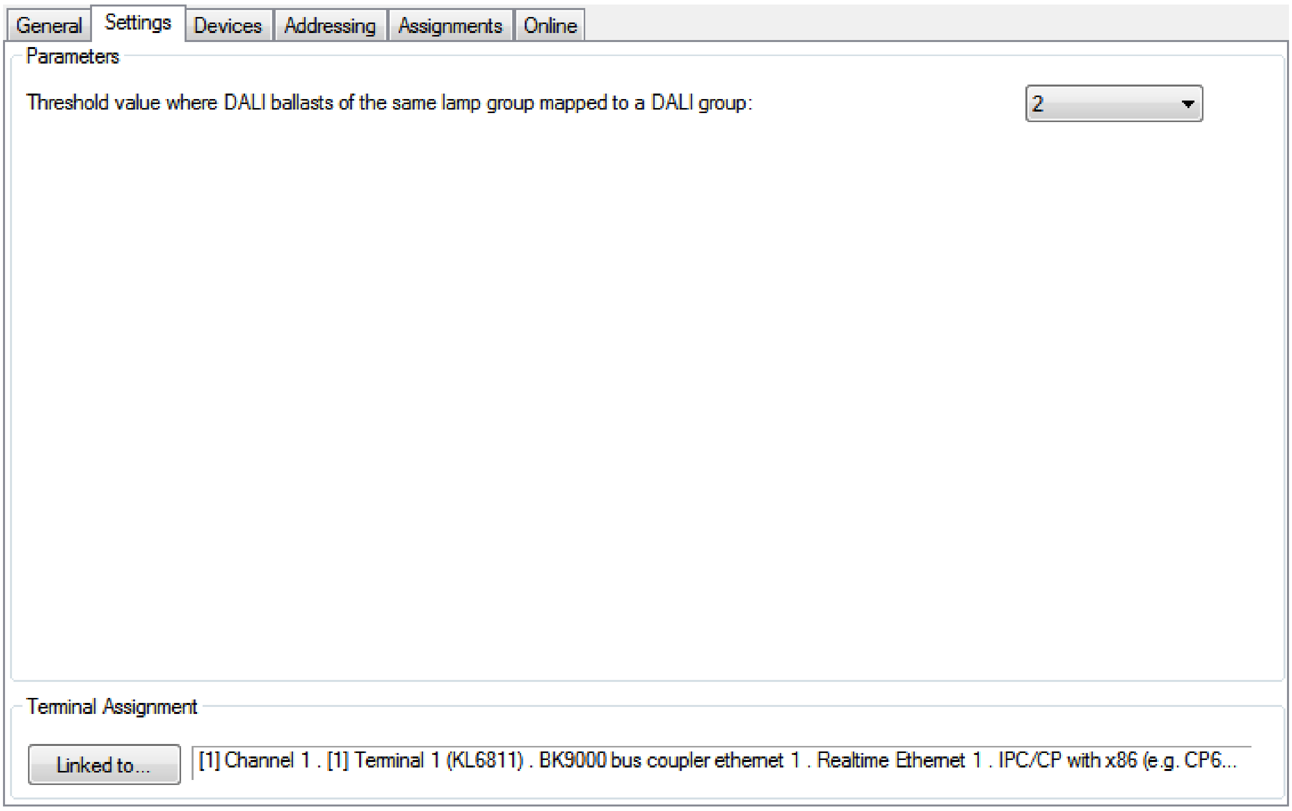
- **Execution State:** shows if the module is running or stopped.
- **Error:** in case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** starts/stops the module.

7.2.2.12 Subsystems

7.2.2.12.1 DALI

The Digital Addressable Lighting Interface (DALI) is a simple bus system for building automation that transmits the signals for the digital operation of lighting devices. The DALI line via KL6811 offers the option to additionally integrate DALI devices in order to configure and work with them.

Settings



Parameters

Threshold value where DALI ballasts of the same lamp group mapped to a DALI group: 2

Terminal Assignment

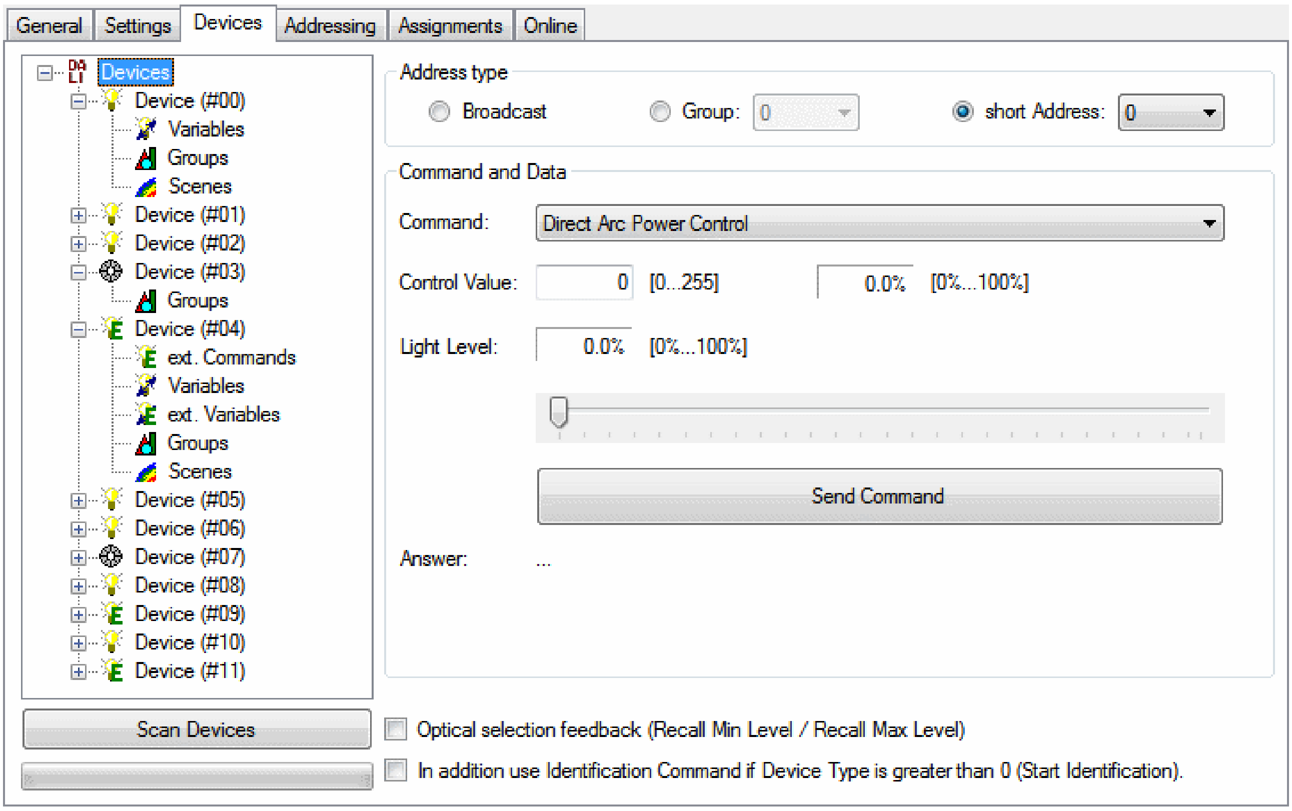
[1] Channel 1 . [1] Terminal 1 (KL6811) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. CP6...

Parameters

- **Threshold value where DALI ballasts of the same lamp group mapped to a DALI group:** Input field for the threshold value, where DALI-ballasts of the same lamp group are mapped to a DALI-group.

Pressing the *Linked to...*-button opens a dialog, where the designated input-terminal can be assigned to this module.

Devices



Address type

○ Broadcast ○ Group: 0 ● short Address: 0

Command and Data

Command: Direct Arc Power Control

Control Value: 0 [0...255] 0.0% [0%...100%]

Light Level: 0.0% [0%...100%]

Answer: ...

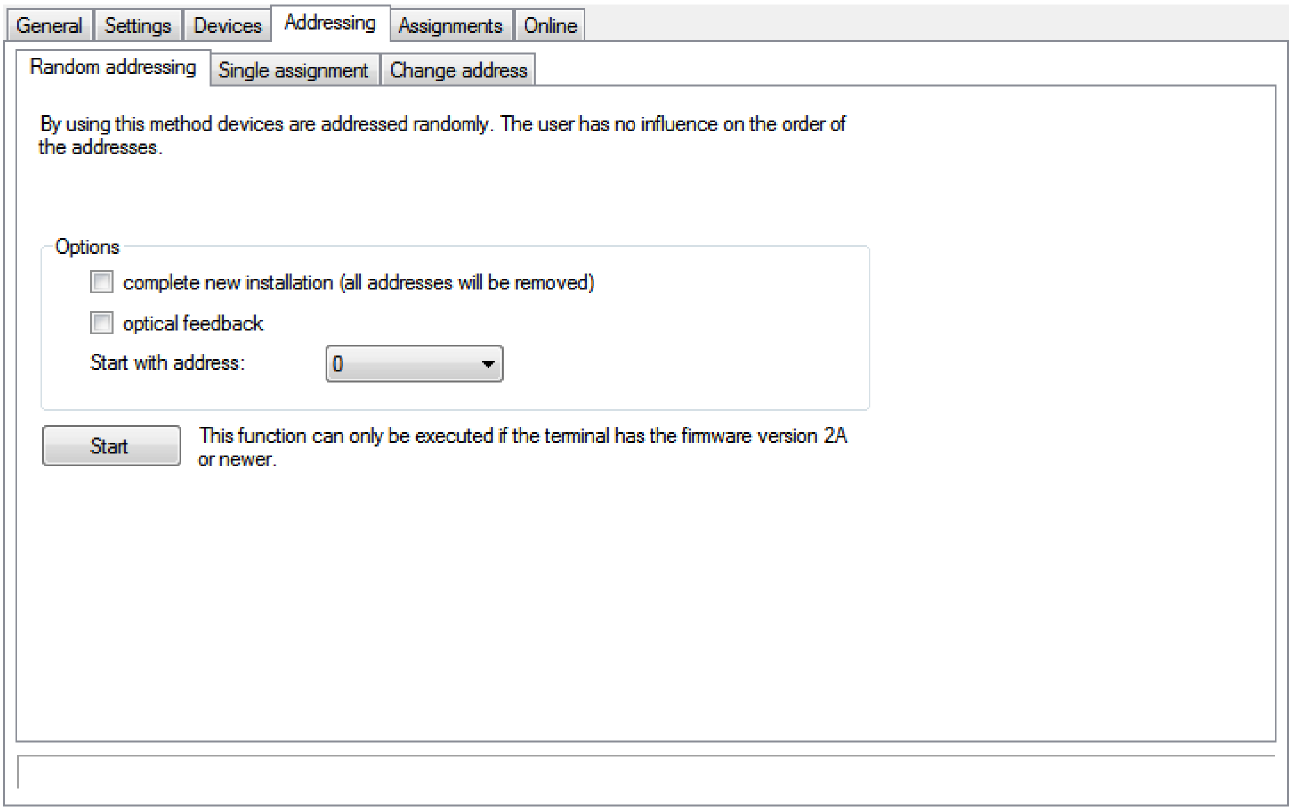
□ Optical selection feedback (Recall Min Level / Recall Max Level)

□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

This dialog lists all scanned DALI-devices and shows all available parameters and commands. Three DALI-types are available:

- Standard [▶ 176]
- Self-contained Emergency Lighting [▶ 179]
- SmartSPOT [▶ 182]

Addressing



By using this method devices are addressed randomly. The user has no influence on the order of the addresses.

Options

□ complete new installation (all addresses will be removed)

□ optical feedback

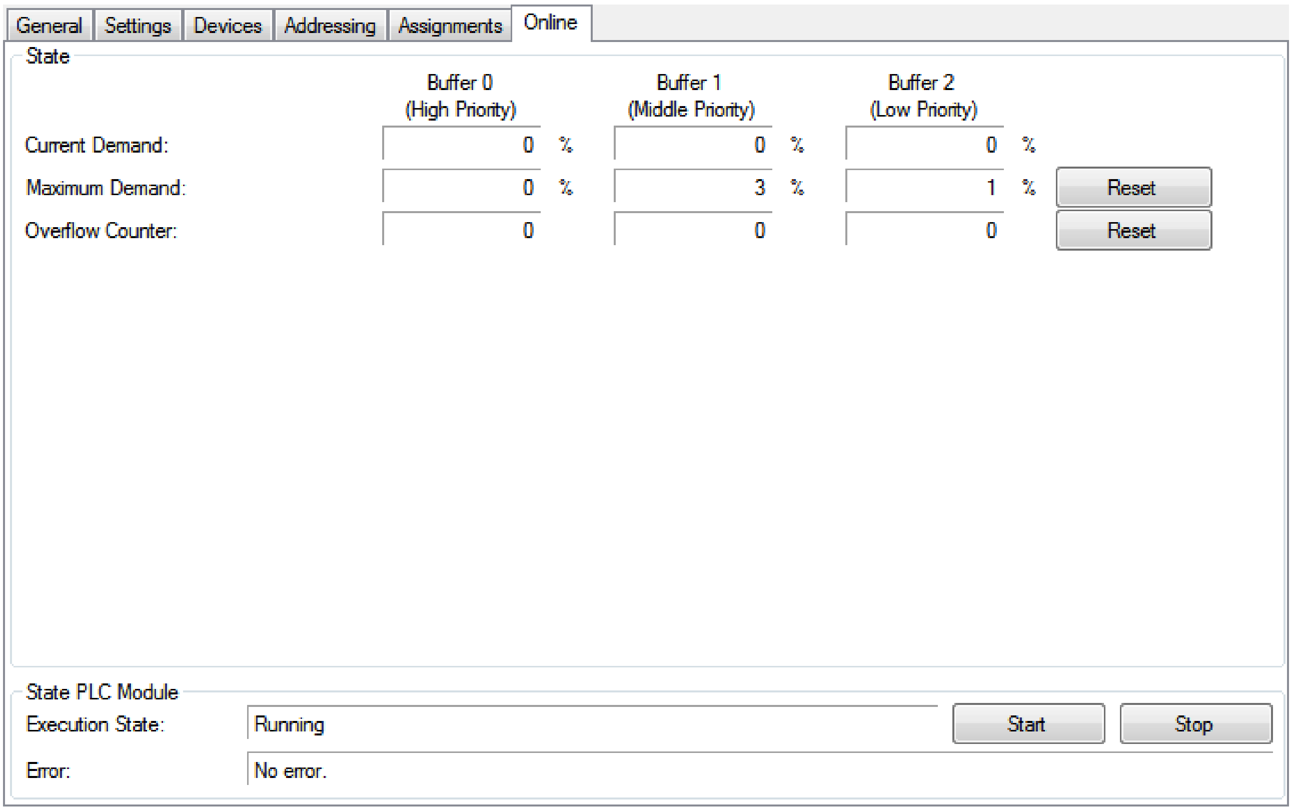
Start with address: 0

This function can only be executed if the terminal has the firmware version 2A or newer.

The *Adressing*-menu provides access to the addressing-options. For more information, please see the following pages:

- Random addressing [▶ 184]
- Single assignment [▶ 184]
- Change address [▶ 185]

Online



State

	Buffer 0 (High Priority)		Buffer 1 (Middle Priority)		Buffer 2 (Low Priority)	
Current Demand:	0	%	0	%	0	%
Maximum Demand:	0	%	3	%	1	%
Overflow Counter:	0		0		0	

State PLC Module

Execution State: Running

Error: No error.

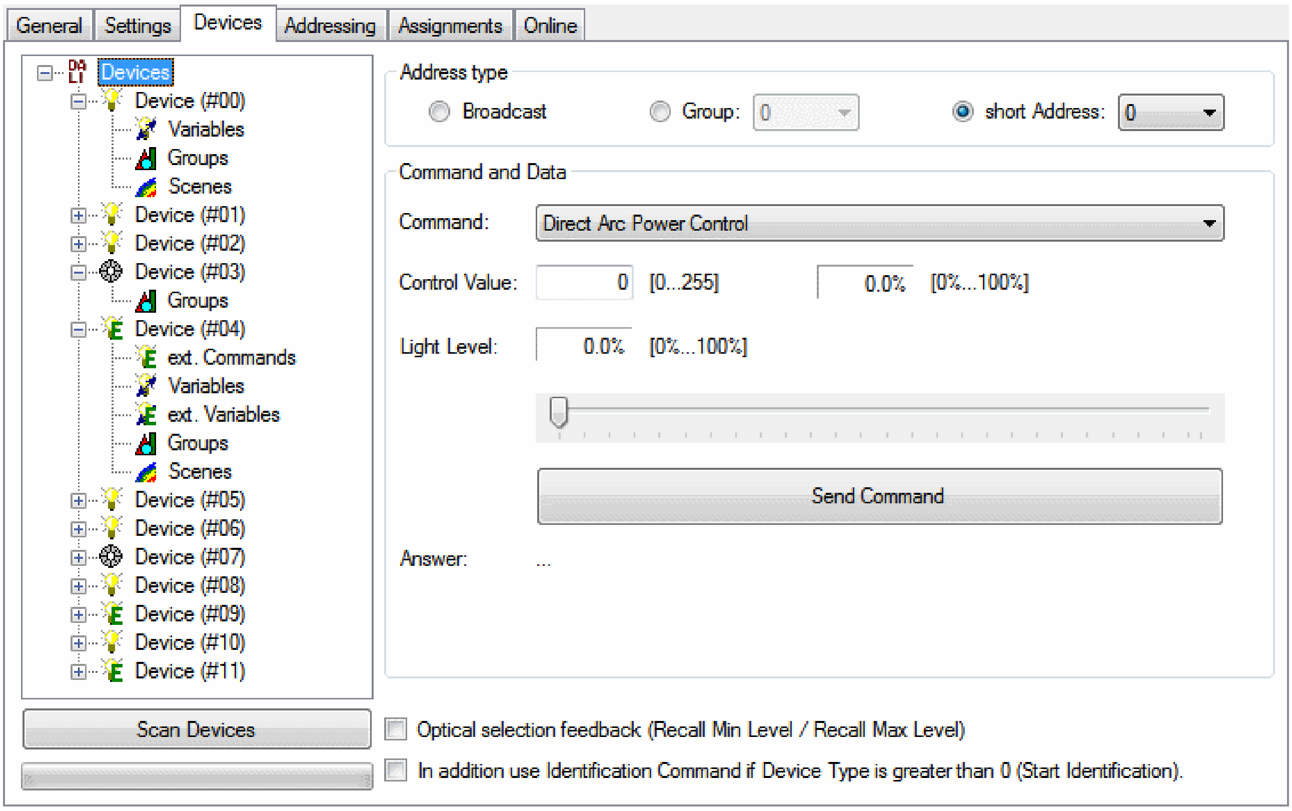
State

- **Current Demand:** The current demand in percentage.
- **Maximum Demand:** The maximum demand in percentage.
- **Overflow Counter:** Displays the current count of recognized overflows.

State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error-code and the error-parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.12.1.1 Devices



Address type: ○ Broadcast ○ Group: 0 ● short Address: 0

Command: Direct Arc Power Control

Control Value: 0 [0...255] 0.0% [0%...100%]

Light Level: 0.0% [0%...100%]

Answer: ...

□ Optical selection feedback (Recall Min Level / Recall Max Level)

□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

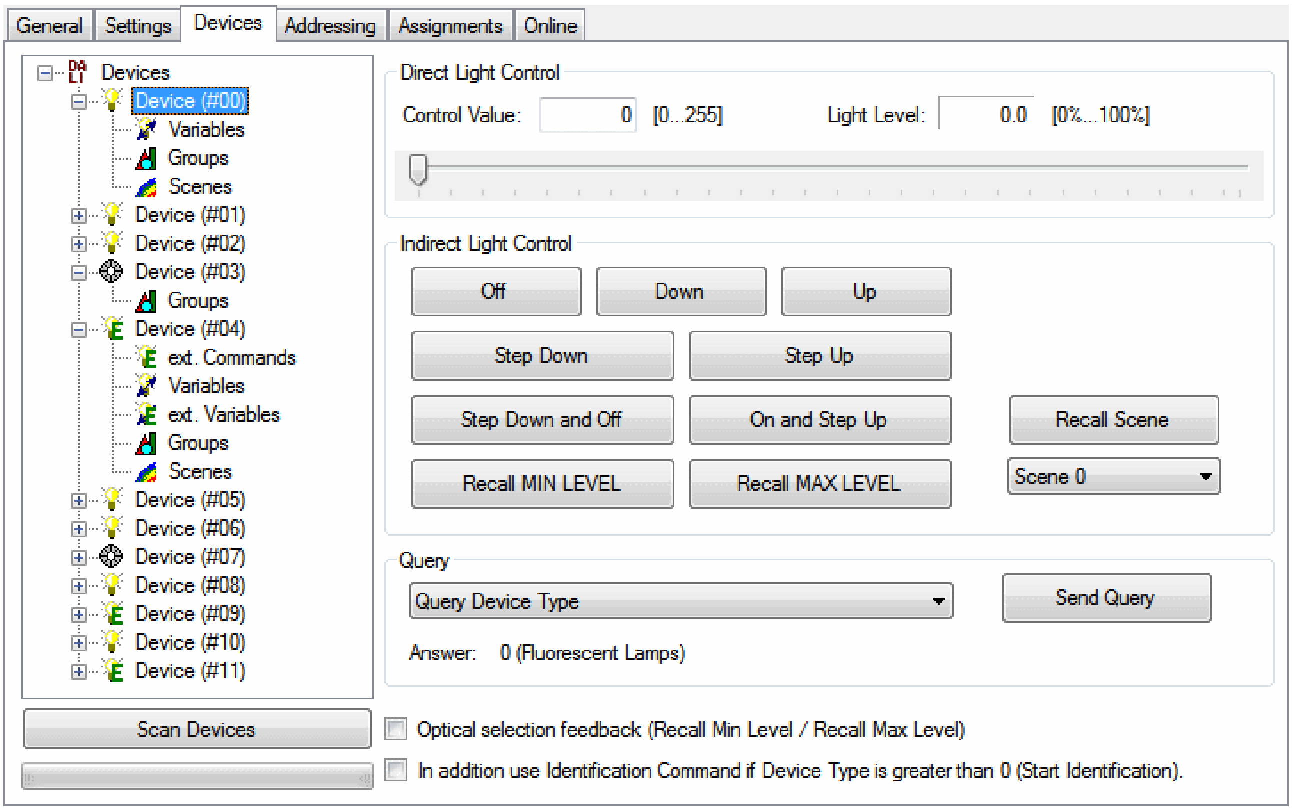
This dialog lists all scanned DALI-devices and shows all available parameters and commands. Three DALI-types are available:

- Standard [▸ 176]
- Self-contained Emergency Lighting [▸ 179]
- SmartSPOT [▸ 182]

7.2.2.12.1.1.1 Standard

The following dialogs allow to configure and to use DALI devices (type 0) with the standard instruction set.

Device



Control Value: 0 [0...255] Light Level: 0.0 [0%...100%]

Scene 0

Query Device Type

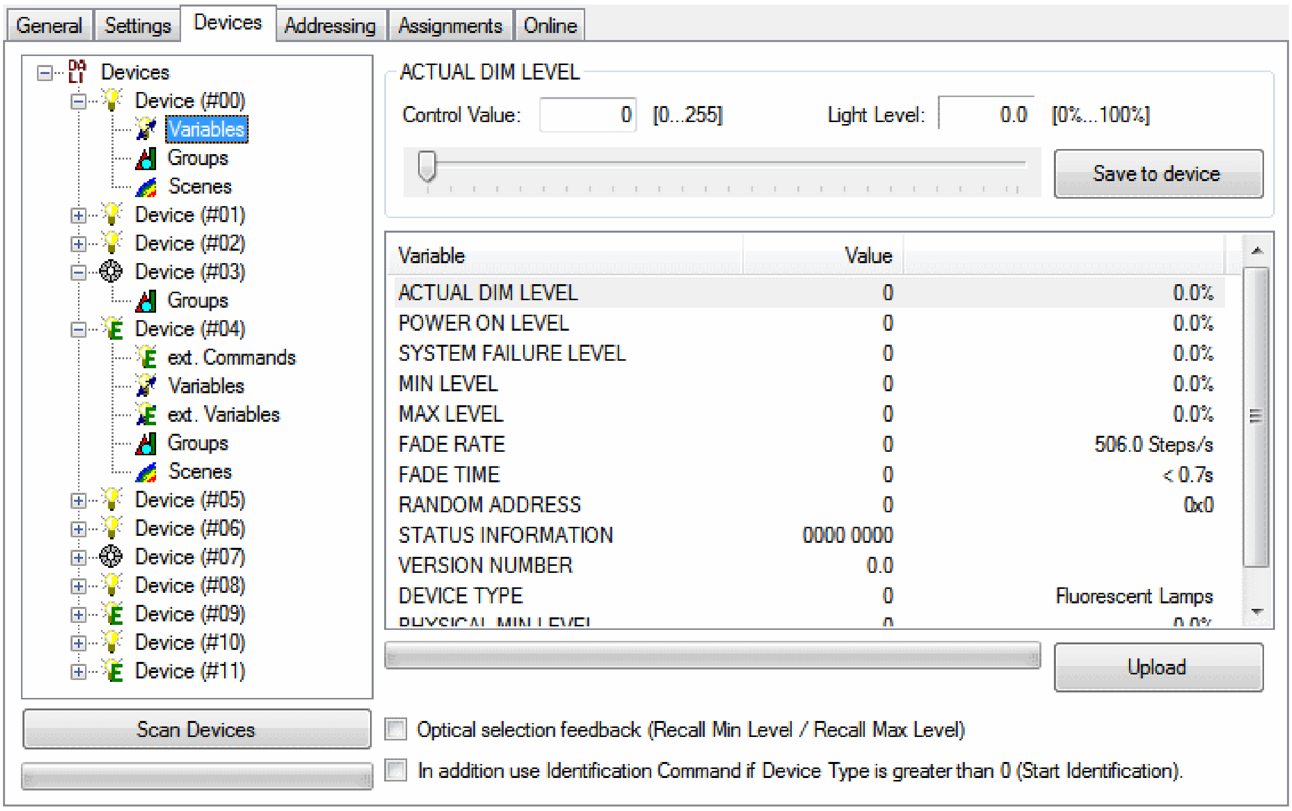
Answer: 0 (Fluorescent Lamps)

☐ Optical selection feedback (Recall Min Level / Recall Max Level)

☐ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

For the most important functions, here are buttons for going to send a corresponding DALI-command to the selected device.

Variables



Control Value: 0 [0...255] Light Level: 0.0 [0%...100%]

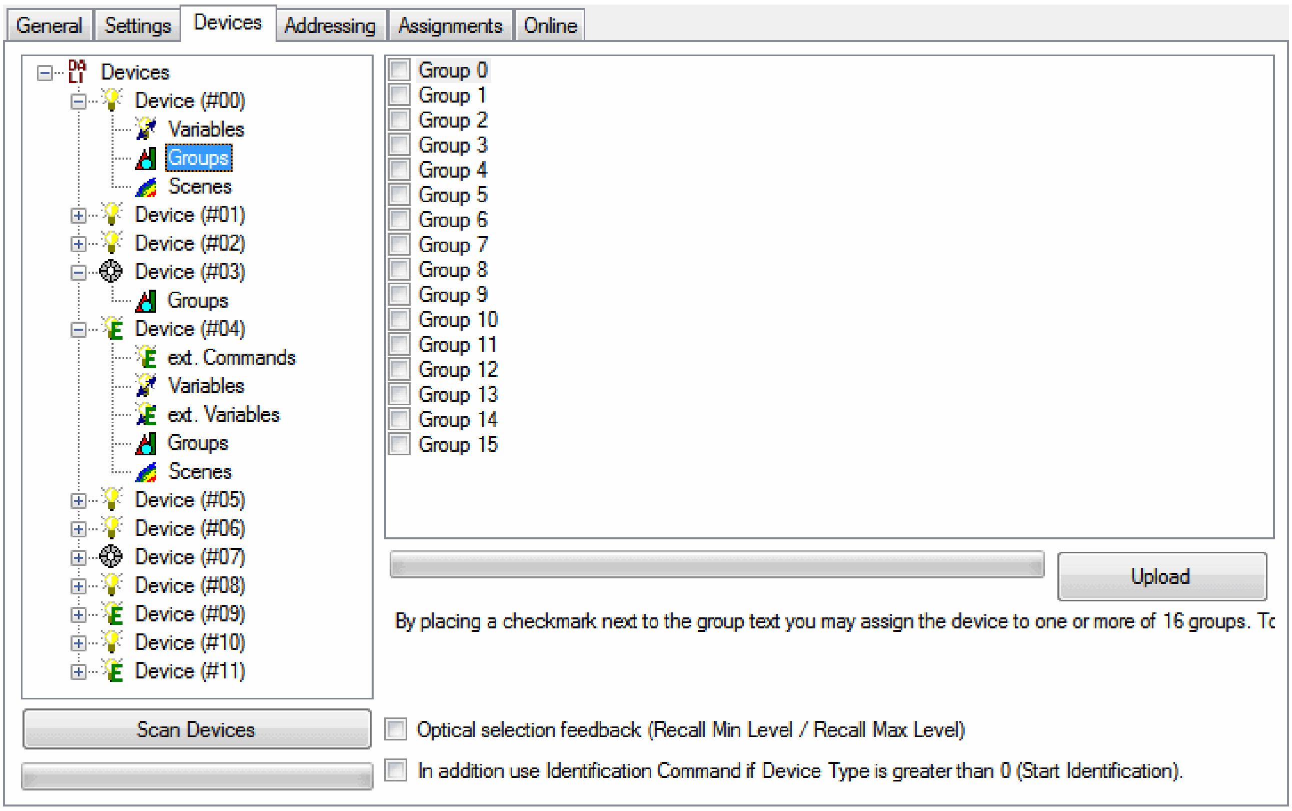
Variable	Value	
ACTUAL DIM LEVEL	0	0.0%
POWER ON LEVEL	0	0.0%
SYSTEM FAILURE LEVEL	0	0.0%
MIN LEVEL	0	0.0%
MAX LEVEL	0	0.0%
FADE RATE	0	506.0 Steps/s
FADE TIME	0	< 0.7s
RANDOM ADDRESS	0	0x0
STATUS INFORMATION	0000 0000	
VERSION NUMBER	0.0	
DEVICE TYPE	0	Fluorescent Lamps
PHYSICAL MIN LEVEL	0	0.0%

☐ Optical selection feedback (Recall Min Level / Recall Max Level)

☐ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

With DALI, the intelligence was not centralized to 100%. This means, that many settings and light-levels in the form of variables are stored in the ballast. The *Variables*-menu displays a dialog, where you can review and edit the variables of the selected DALI-device.

Groups



- □ Group 0
- □ Group 1
- □ Group 2
- □ Group 3
- □ Group 4
- □ Group 5
- □ Group 6
- □ Group 7
- □ Group 8
- □ Group 9
- □ Group 10
- □ Group 11
- □ Group 12
- □ Group 13
- □ Group 14
- □ Group 15

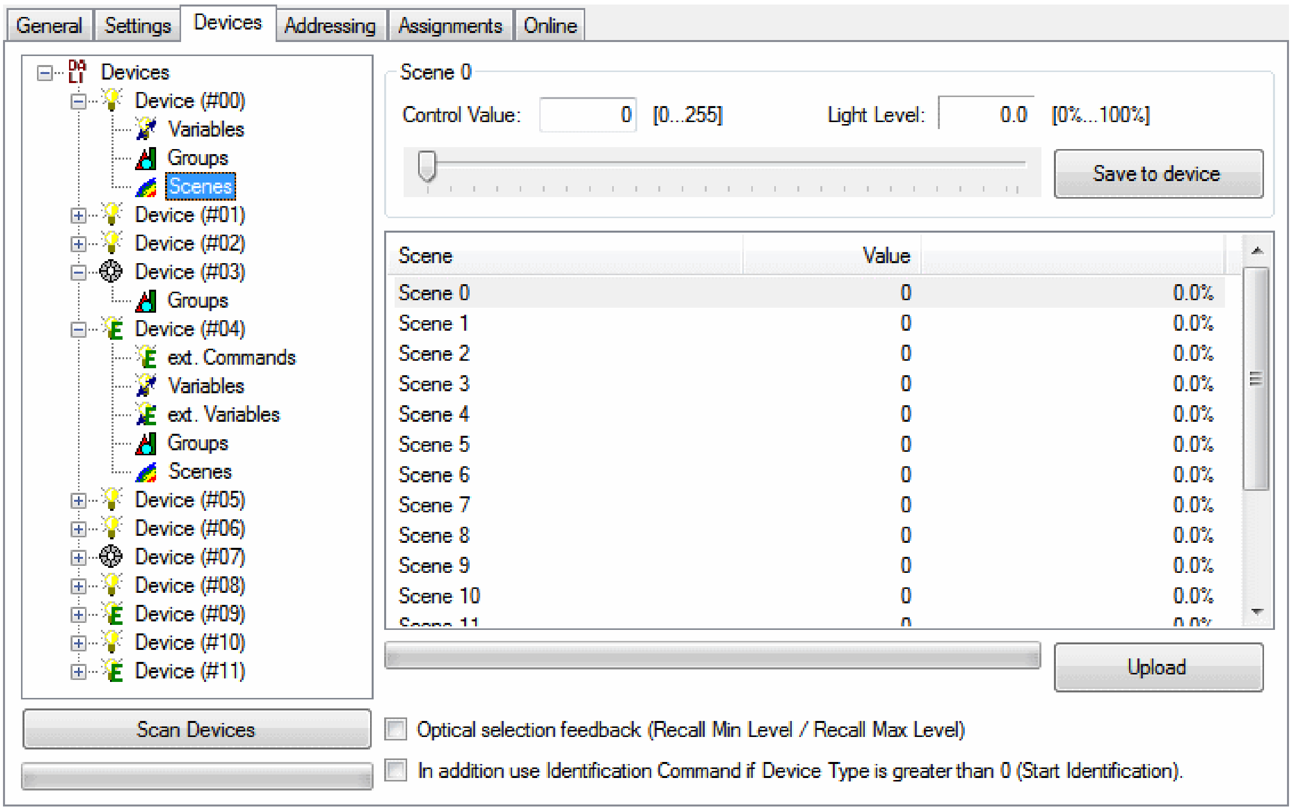
By placing a checkmark next to the group text you may assign the device to one or more of 16 groups. Tc

□ Optical selection feedback (Recall Min Level / Recall Max Level)

□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

In this menu you can add each DALI-device up to 16 groups. When a DALI-device receives a command in conjunction with a parameterized group-number, it will execute this command.

Scenes



Scene 0

Control Value: 0 [0...255] Light Level: 0.0 [0%...100%]

Scene	Value	
Scene 0	0	0.0%
Scene 1	0	0.0%
Scene 2	0	0.0%
Scene 3	0	0.0%
Scene 4	0	0.0%
Scene 5	0	0.0%
Scene 6	0	0.0%
Scene 7	0	0.0%
Scene 8	0	0.0%
Scene 9	0	0.0%
Scene 10	0	0.0%
Scene 11	0	0.0%

□ Optical selection feedback (Recall Min Level / Recall Max Level)

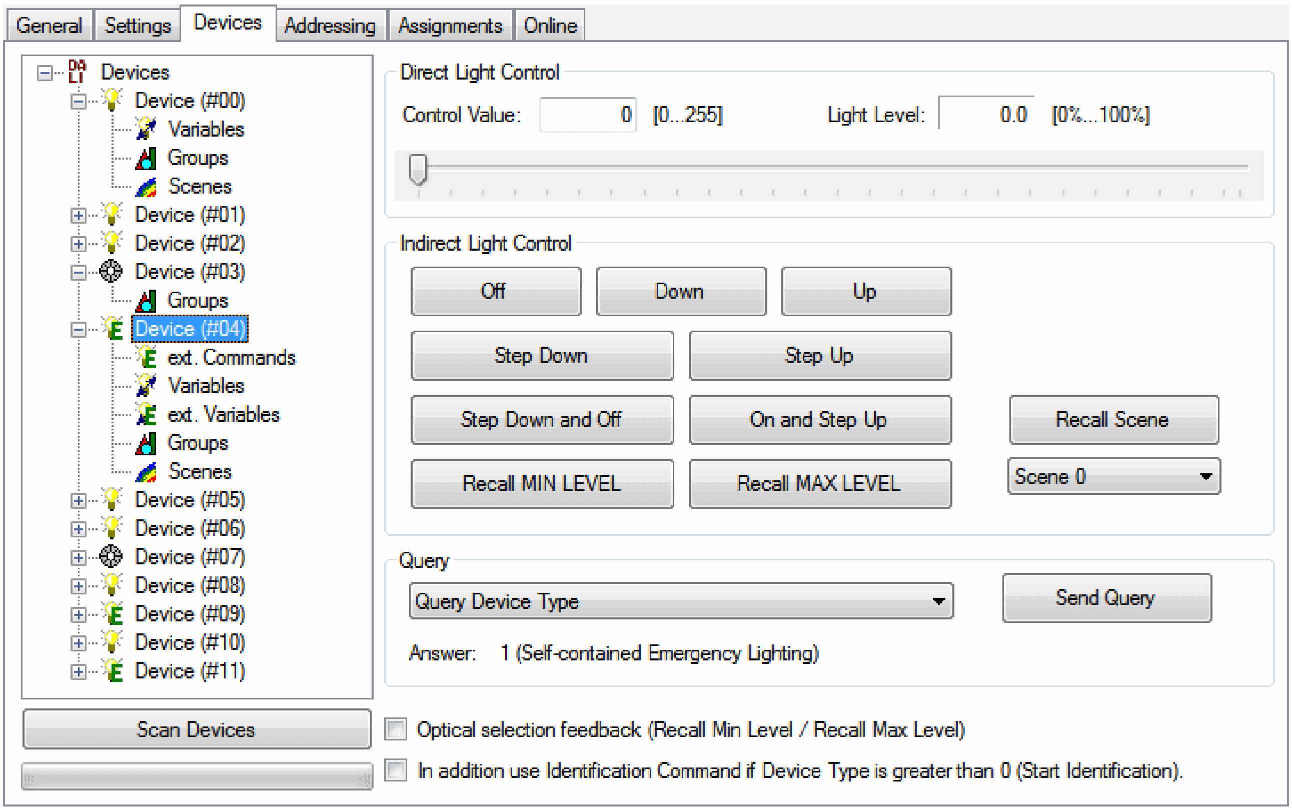
□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

Here you can store 16 different light-levels for scenes on each DALI-device. When a DALI-device receives a parameterized scene-number (e.g., as part of a broadcast), it will set the light-intensity for the scene. If the value of 255 (mask) is stored for this scene, the DALI-device retains the previous value.

7.2.2.12.1.1.2 *Self-contained Emergency Lighting*

In addition to the standard instruction set, the autonomous emergency lighting (DALI device type 1) also supports a type-specific instruction set to meet the requirements for emergency lighting. The following dialogues are available for the configuration and operation.

Device



Control Value: 0 [0...255]

Light Level: 0.0 [0%...100%]

Scene 0

Query Device Type

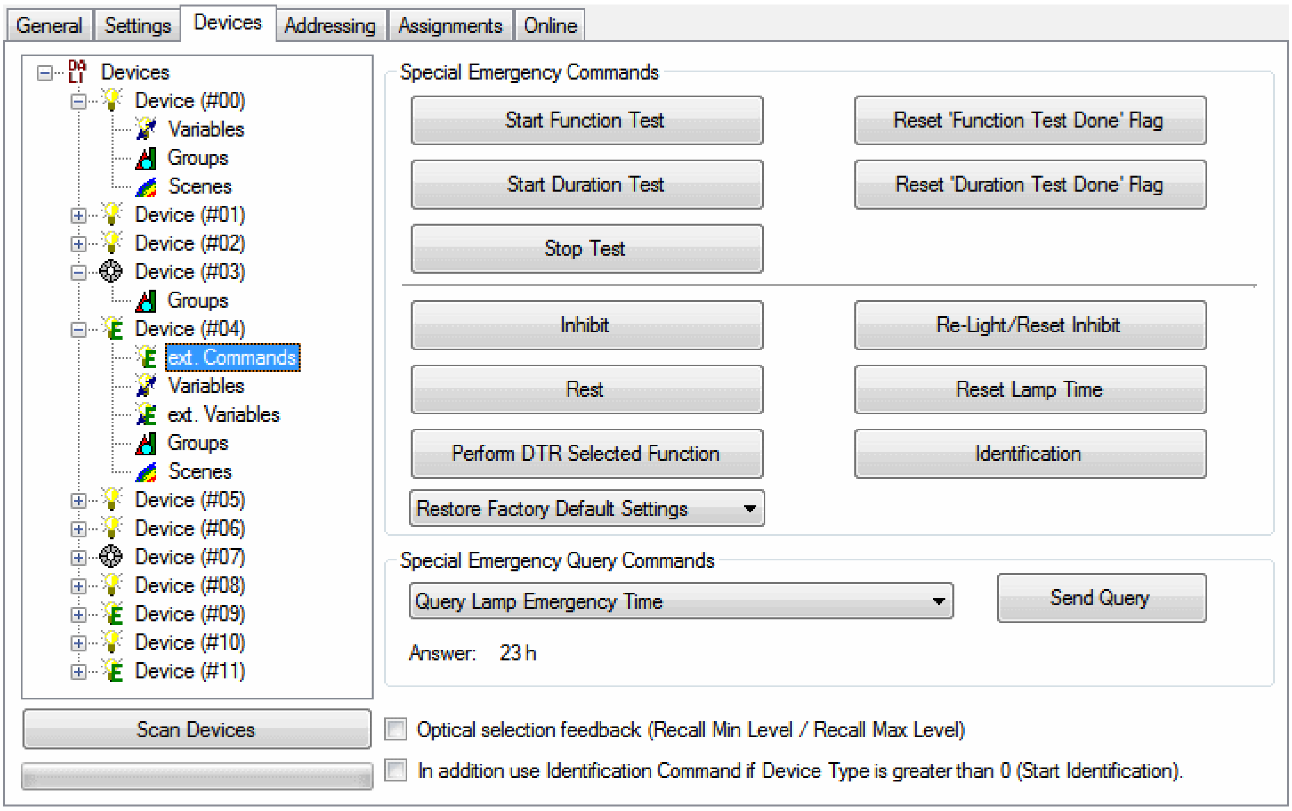
Answer: 1 (Self-contained Emergency Lighting)

☐ Optical selection feedback (Recall Min Level / Recall Max Level)

☐ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

For the most important functions, here are buttons for going to send a corresponding DALI-command to the selected device.

Ext. Commands



Restore Factory Default Settings

Query Lamp Emergency Time

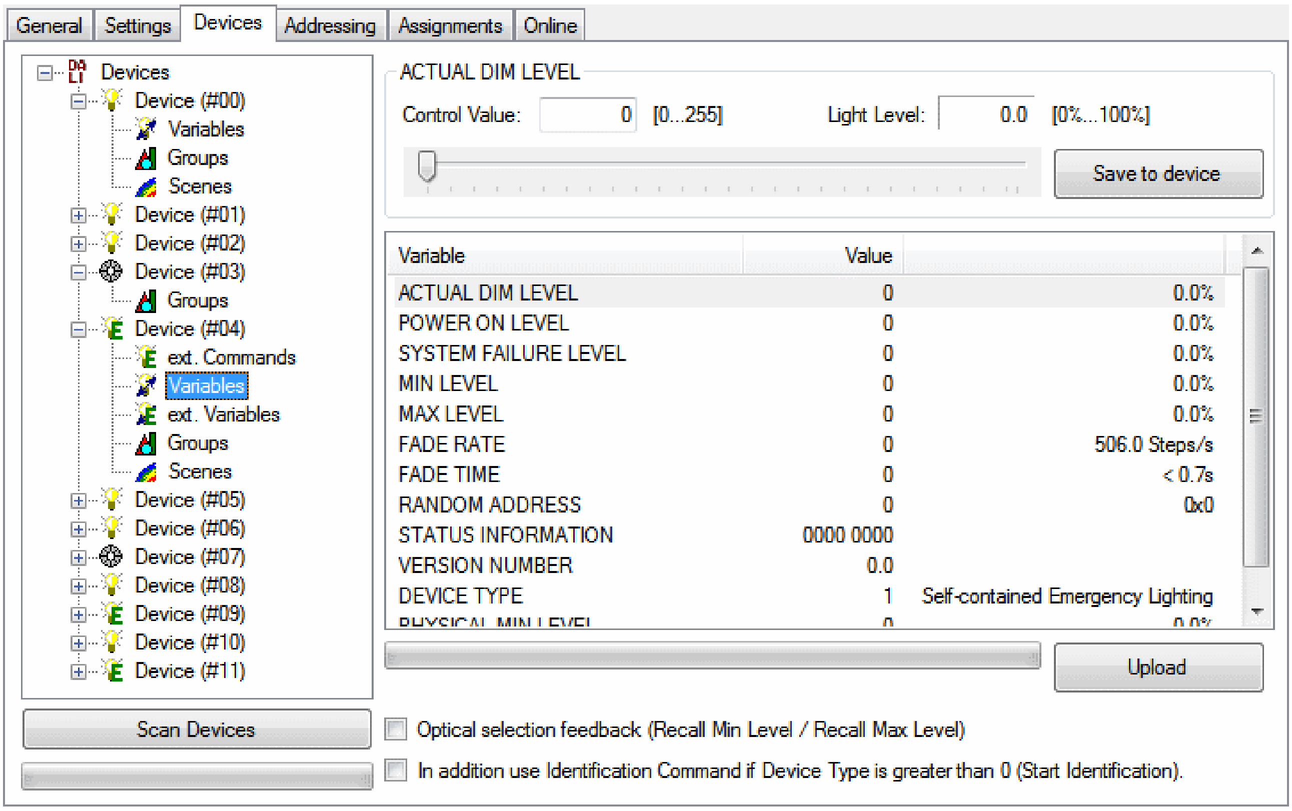
Answer: 23 h

□ Optical selection feedback (Recall Min Level / Recall Max Level)

□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

Provides access to the specific commands of the self-contained emergency lighting.

Variables



ACTUAL DIM LEVEL

Control Value: 0 [0...255] Light Level: 0.0 [0%...100%]

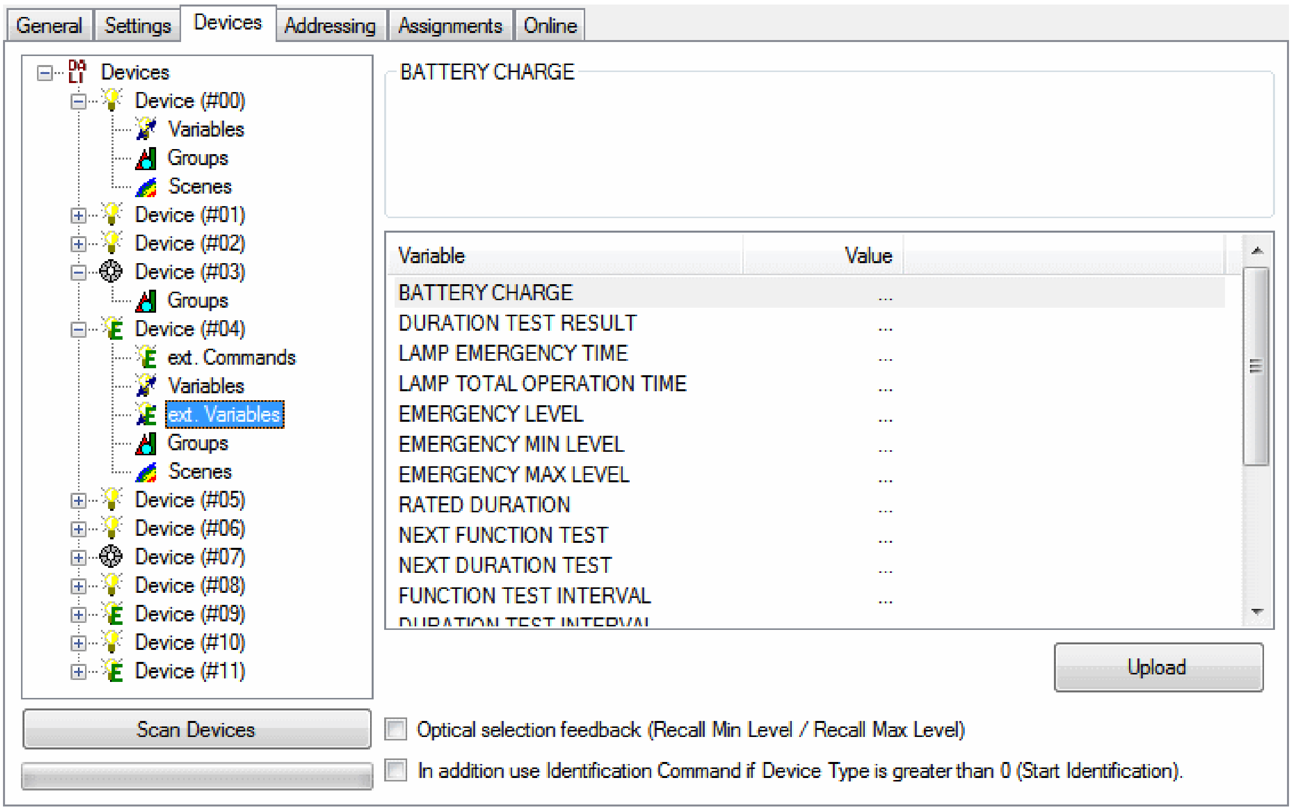
Variable	Value	
ACTUAL DIM LEVEL	0	0.0%
POWER ON LEVEL	0	0.0%
SYSTEM FAILURE LEVEL	0	0.0%
MIN LEVEL	0	0.0%
MAX LEVEL	0	0.0%
FADE RATE	0	506.0 Steps/s
FADE TIME	0	< 0.7s
RANDOM ADDRESS	0	0x0
STATUS INFORMATION	0000 0000	
VERSION NUMBER	0.0	
DEVICE TYPE	1	Self-contained Emergency Lighting
PHYSICAL MIN LEVEL	0	0.0%

□ Optical selection feedback (Recall Min Level / Recall Max Level)

□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

With DALI, the intelligence was not centralized to 100%. This means, that many settings and light-levels in the form of variables are stored in the ballast. The *Variables*-menu displays a dialog where you can review and edit the variables of the selected DALI-device.

Ext. Variables



BATTERY CHARGE

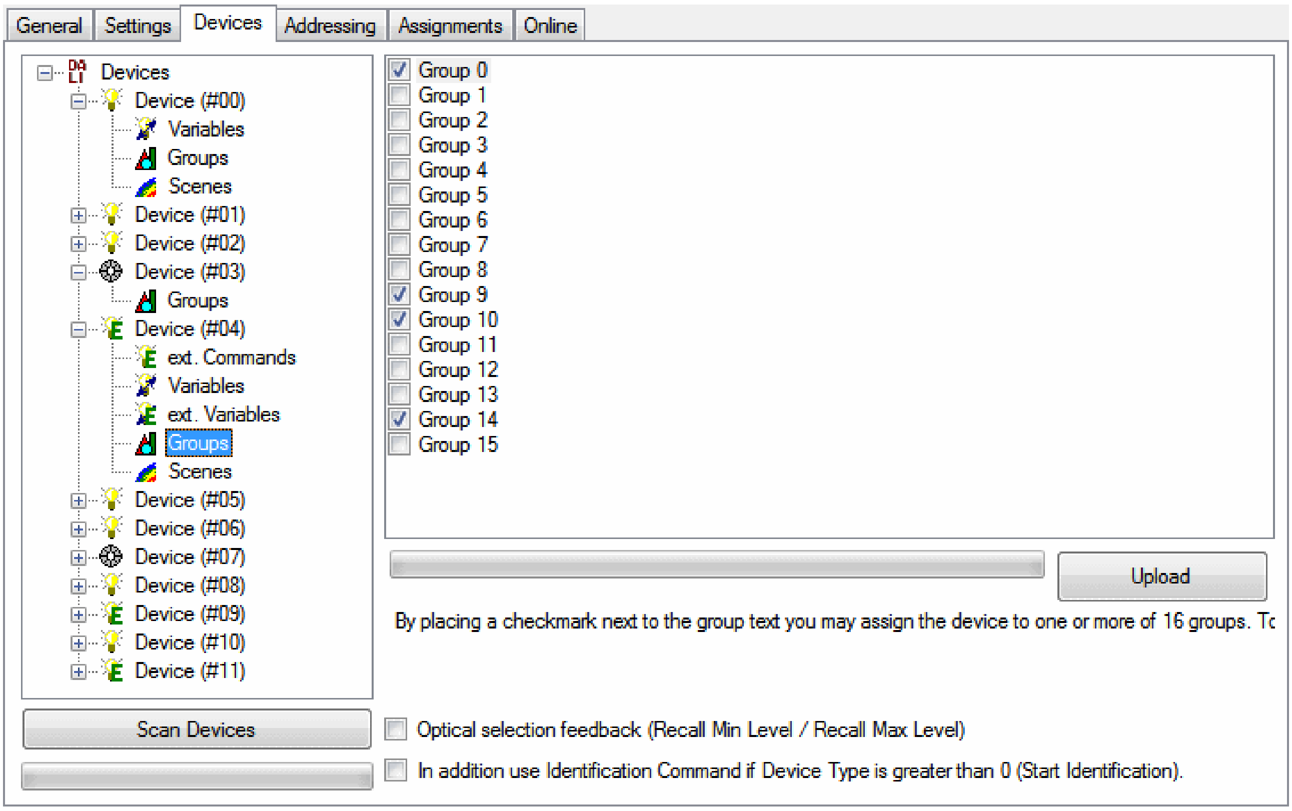
Variable	Value
BATTERY CHARGE	...
DURATION TEST RESULT	...
LAMP EMERGENCY TIME	...
LAMP TOTAL OPERATION TIME	...
EMERGENCY LEVEL	...
EMERGENCY MIN LEVEL	...
EMERGENCY MAX LEVEL	...
RATED DURATION	...
NEXT FUNCTION TEST	...
NEXT DURATION TEST	...
FUNCTION TEST INTERVAL	...
DURATION TEST INTERVAL	

□ Optical selection feedback (Recall Min Level / Recall Max Level)

□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

Displays all values of the extended variables.

Groups



☑ Group 0
□ Group 1
□ Group 2
□ Group 3
□ Group 4
□ Group 5
□ Group 6
□ Group 7
□ Group 8
☑ Group 9
☑ Group 10
□ Group 11
□ Group 12
□ Group 13
☑ Group 14
□ Group 15

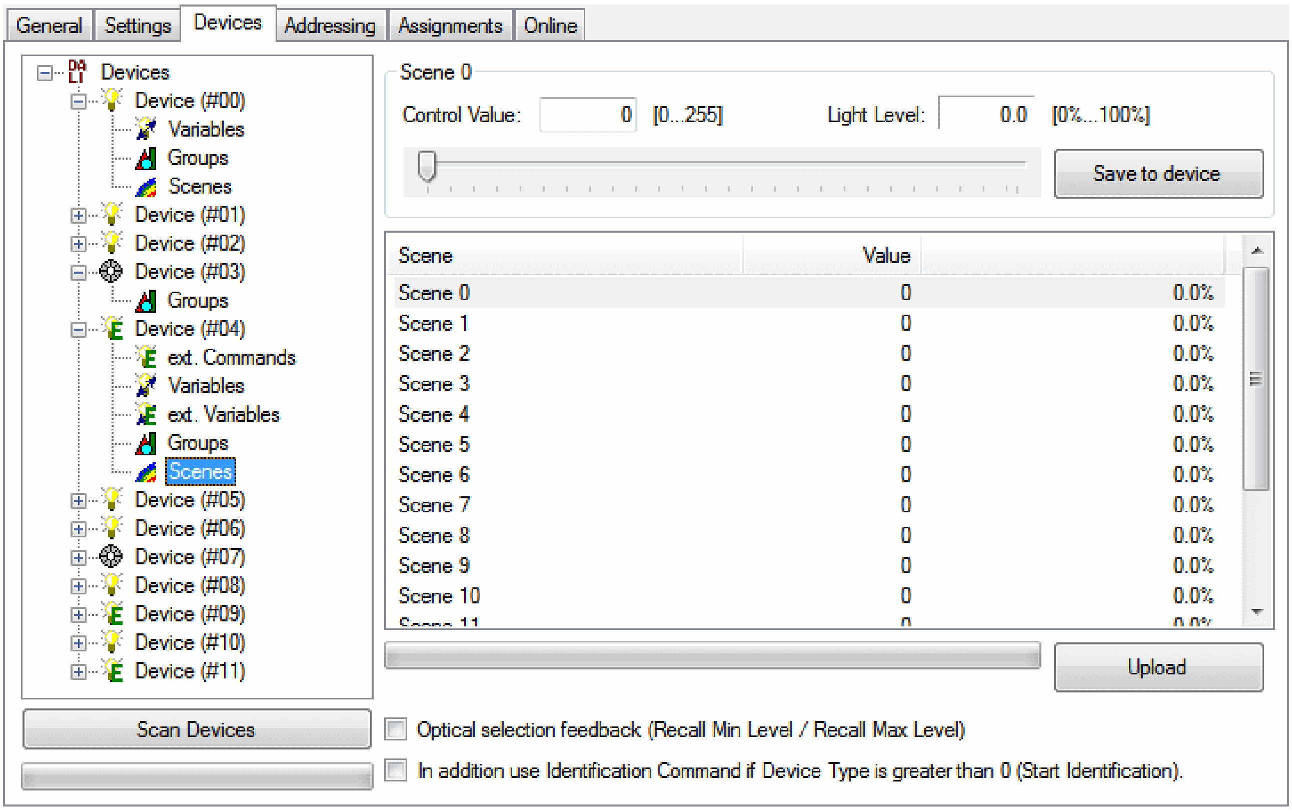
By placing a checkmark next to the group text you may assign the device to one or more of 16 groups. To

□ Optical selection feedback (Recall Min Level / Recall Max Level)

□ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

In this menu you can add each DALI-device up to 16 groups. When a DALI-device receives a command in conjunction with a parameterized group-number, it will execute this command.

Scenes



Control Value: 0 [0...255]

Light Level: 0.0 [0%...100%]

Scene	Value	
Scene 0	0	0.0%
Scene 1	0	0.0%
Scene 2	0	0.0%
Scene 3	0	0.0%
Scene 4	0	0.0%
Scene 5	0	0.0%
Scene 6	0	0.0%
Scene 7	0	0.0%
Scene 8	0	0.0%
Scene 9	0	0.0%
Scene 10	0	0.0%

☐ Optical selection feedback (Recall Min Level / Recall Max Level)

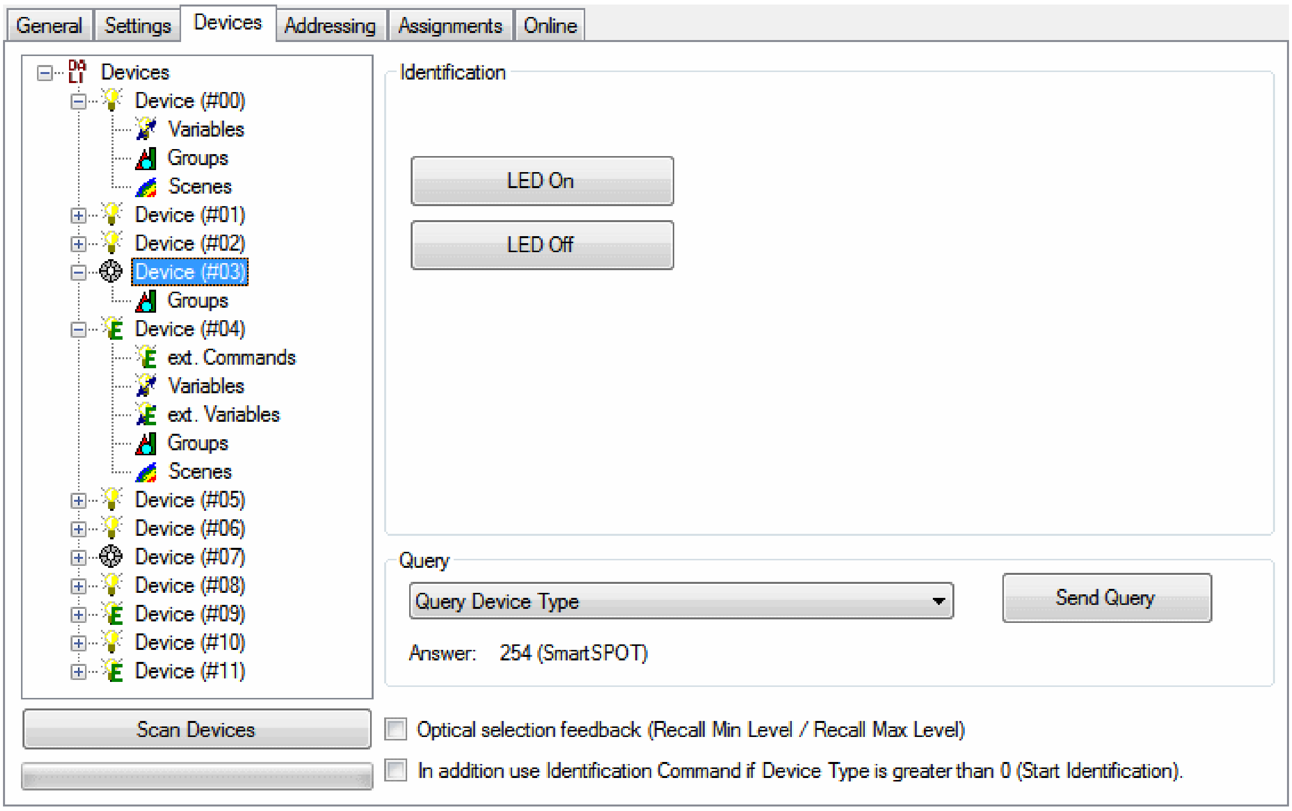
☐ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

Here you can store 16 different light-levels for scenes on each DALI-device. When a DALI-device receives a parameterized scene-number (e.g. as part of a broadcast), it will set the light-intensity for the scene. If the value of 255 (mask) is stored for this scene, the DALI-device retains the previous value.

7.2.2.12.1.1.3 SmartSPOT

The smartSPOT (DALI device type 254) for motion detection offers only a few configuration and operating options compared to other DALI devices. The following dialogues are available for the necessary adjustments.

Device



Query: Query Device Type

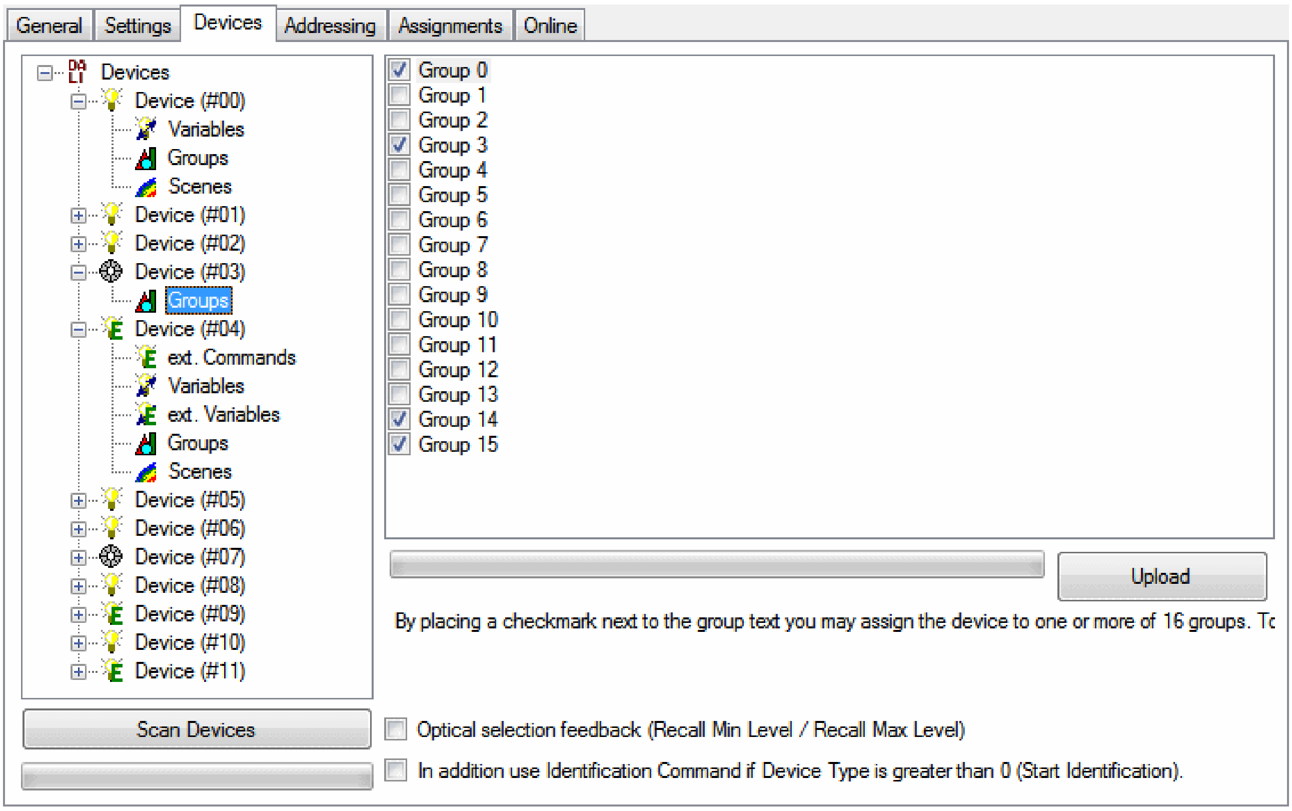
Answer: 254 (SmartSPOT)

☐ Optical selection feedback (Recall Min Level / Recall Max Level)

☐ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

For the most important functions, here are buttons for going to send a corresponding DALI-command to the selected device.

Groups



☑ Group 0
☐ Group 1
☐ Group 2
☑ Group 3
☐ Group 4
☐ Group 5
☐ Group 6
☐ Group 7
☐ Group 8
☐ Group 9
☐ Group 10
☐ Group 11
☐ Group 12
☐ Group 13
☑ Group 14
☑ Group 15

By placing a checkmark next to the group text you may assign the device to one or more of 16 groups. To

☐ Optical selection feedback (Recall Min Level / Recall Max Level)

☐ In addition use Identification Command if Device Type is greater than 0 (Start Identification).

In this menu you can add each DALI-device up to 16 groups. When a DALI-device receives a command in conjunction with a parameterized group-number, it will execute this command.

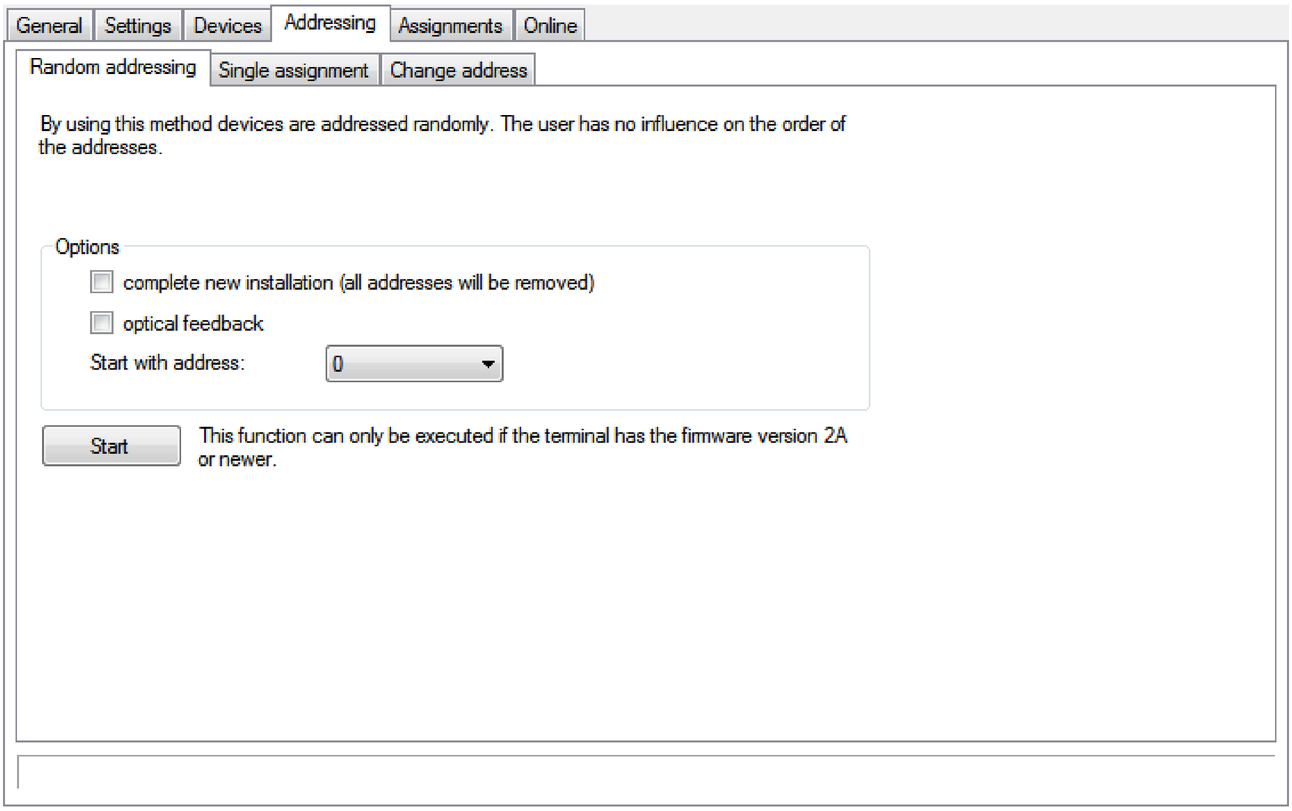
7.2.2.12.1.2 Addressing

The following Addressing options are available:

- Random addressing [⏵ 184]
- Single assignment [⏵ 184]
- Change address [⏵ 185]

7.2.2.12.1.2.1 Random addressing

The start of the random address assignment for the device offers this dialog.



By using this method devices are addressed randomly. The user has no influence on the order of the addresses.

Options

□ complete new installation (all addresses will be removed)

□ optical feedback

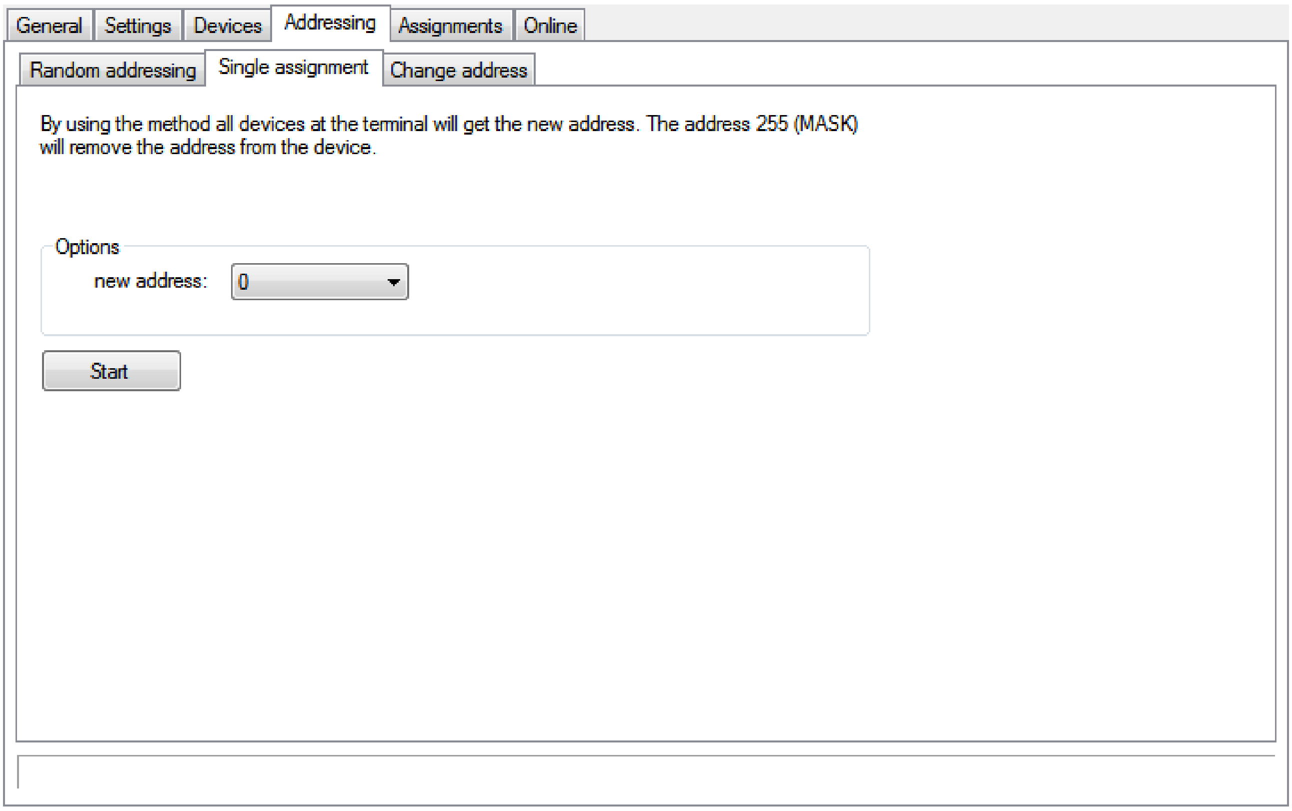
Start with address: 0

This function can only be executed if the terminal has the firmware version 2A or newer.

With this method, the addresses are assigned randomly. It can be taken no influence on the order of address assignment. The addressing may be adjusted later in the *Change address*-menu.

7.2.2.12.1.2.2 Single assignment

In this menu, the addressing can be made individually for each device.



By using the method all devices at the terminal will get the new address. The address 255 (MASK) will remove the address from the device.

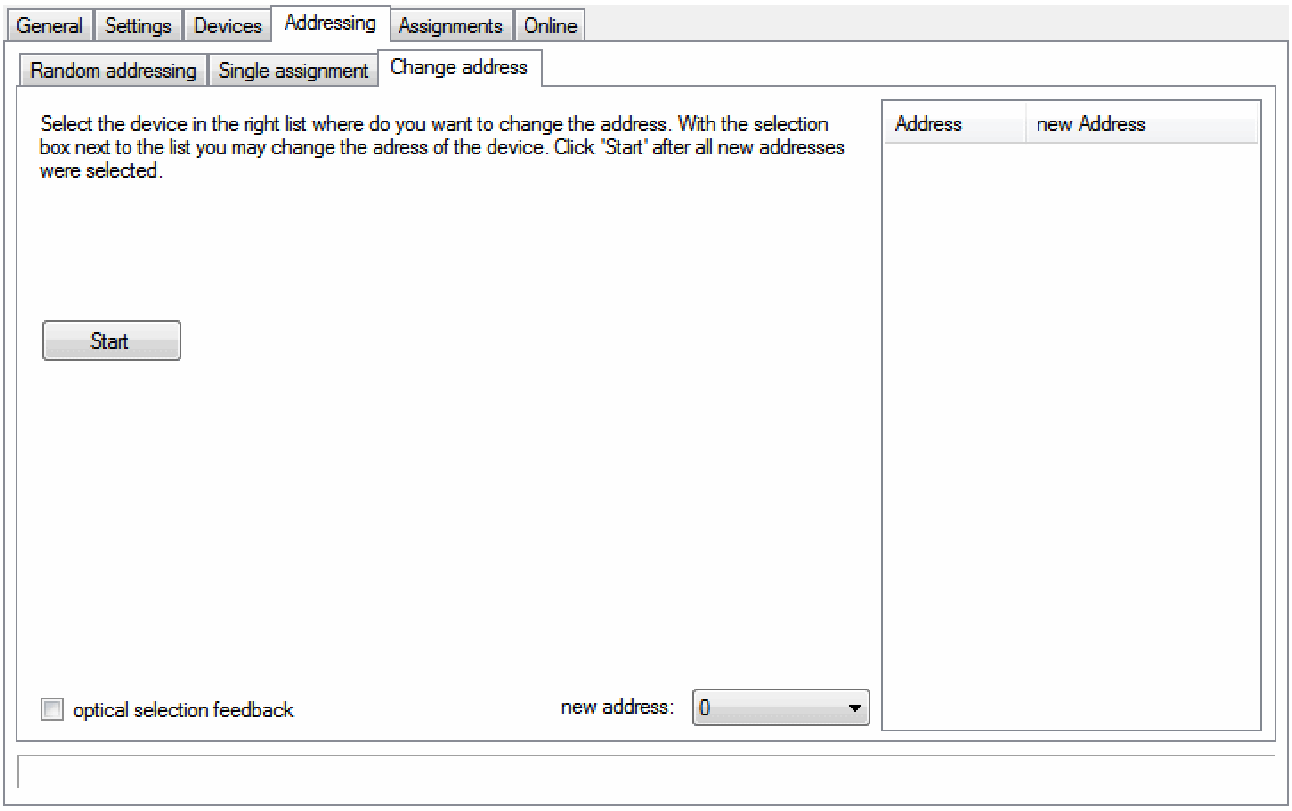
Options

new address: 0

All available DALI-slaves connected to the KL6811 will have the same new address allocated. In practice, you will connect only one DALI-slave, who should receive this address, before pressing the *Start*-button.

7.2.2.12.1.2.3 Change address

Changing the address of a device is supported by this menu.



Select the device in the right list where do you want to change the address. With the selection box next to the list you may change the adress of the device. Click 'Start' after all new addresses were selected.

Address	new Address

□ optical selection feedback

new address: 0

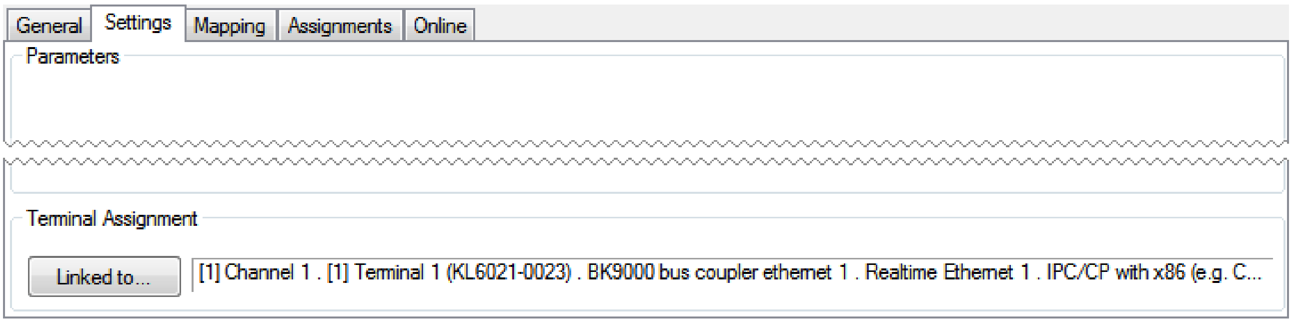
Select from the list the device, where you want to change the address. Choose a new address for the selected device via the combo-box *new address*. After you have selected all the new addresses, click the *Start*-button to assign these addresses to the devices.

Thus, the addresses can be changed, at least one unused address in the DALI-network needs to be present.

7.2.2.12.2 EnOcean

EnOcean modules enable radio transmission without batteries or an external power supply in the transmitter. Supported EnOcean modules can be integrated by the KL6021-0023 or KL6581 via an EnOcean line.

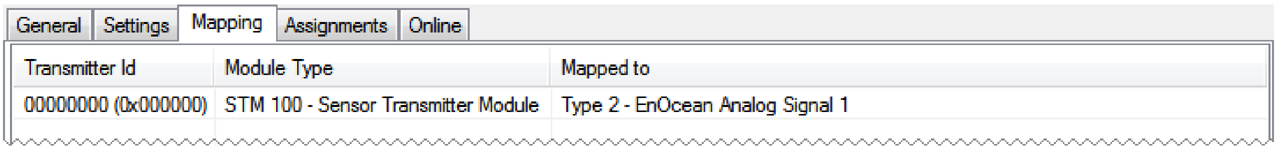
Settings



[1] Channel 1 . [1] Terminal 1 (KL6021-0023) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. C...

Pressing the *Linked to...*-button opens a dialog, where a compatible channel of a terminal from the I/O devices can be assigned to this module. The state of the sensor now corresponds with this channel.

Mapping



Transmitter Id	Module Type	Mapped to
00000000 (0x000000)	STM 100 - Sensor Transmitter Module	Type 2 - EnOcean Analog Signal 1

The list shows all sensors that are linked with the EnOcean-line.

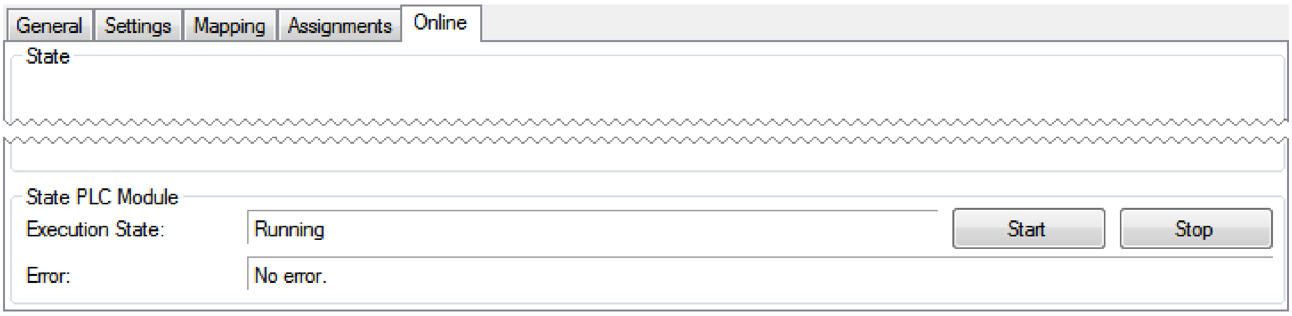
- **Transmitter Id:** Defines the unique transmitter id, specified by the company EnOcean.
- **Module Type:** Describes the module type of EnOcean.
- **Linked Sensor:** Shows which digital signal is configured as EnOcean-module and linked to the EnOCean-line.

Supports EnOcean module types:

- PTM 100 - Pushbutton Transmitter Module
- PTM 200 - Pushbutton Transmitter Module
- PTM 250 - Pushbutton Transmitter Module
- STM 100 - Sensor Transmitter Module
- STM 250 - Wireless Magnet Contact

Further information can be found on the official EnOcean website.

Online



State PLC Module

Execution State: Running

Error: No error.

State PLC Module

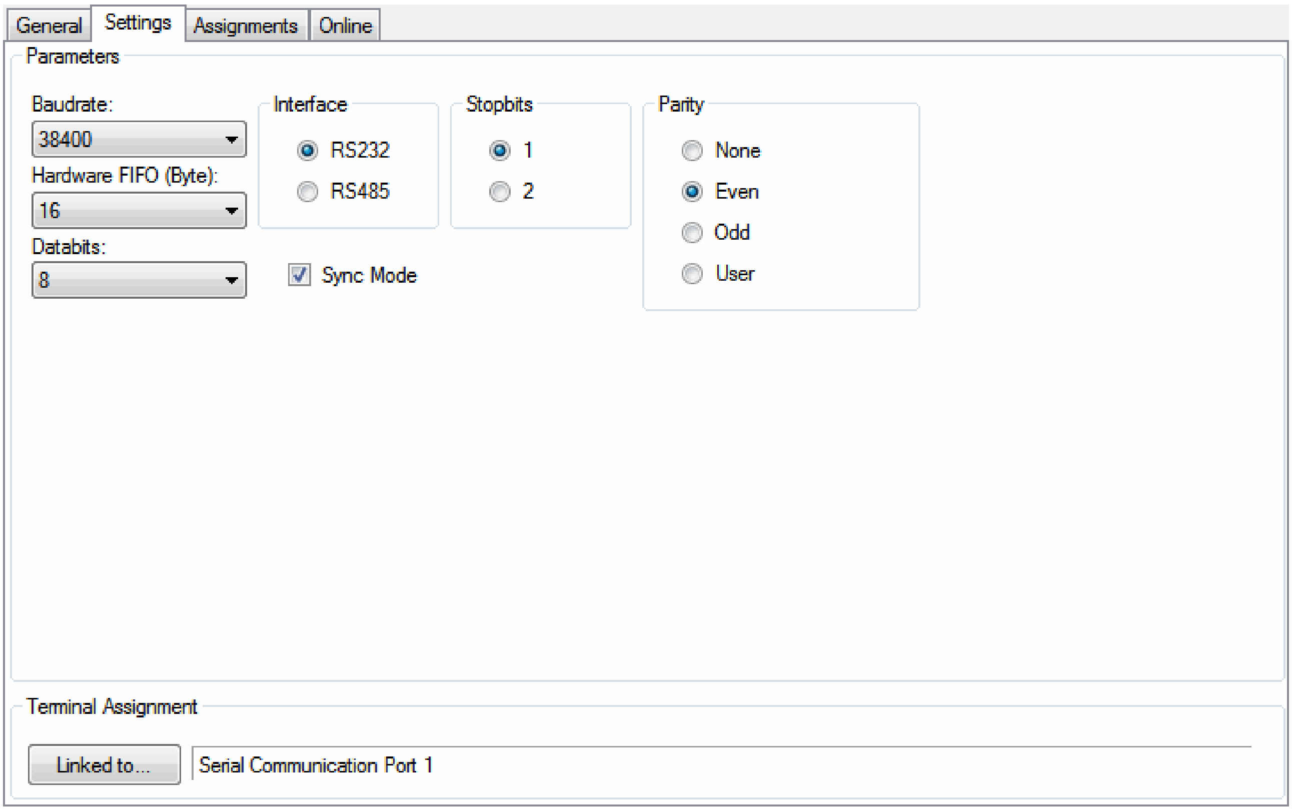
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.2.12.3 Serial Communication

Serial communication enables data acquisition as per RS232/RS485 via a COM port (64 bytes), the KL6031/KL6041 (22 bytes) or the KL6001/KL6021 (5 bytes).

Settings

Parameter settings of *Type 1 - RS232/RS485 via COM Port (64 Bytes)*.



Baudrate: 38400

Hardware FIFO (Byte): 16

Databits: 8

Interface: ● RS232 ○ RS485

Stopbits: ● 1 ○ 2

Parity: ○ None ● Even ○ Odd ○ User

☑ Sync Mode

Terminal Assignment

Serial Communication Port 1

Parameters

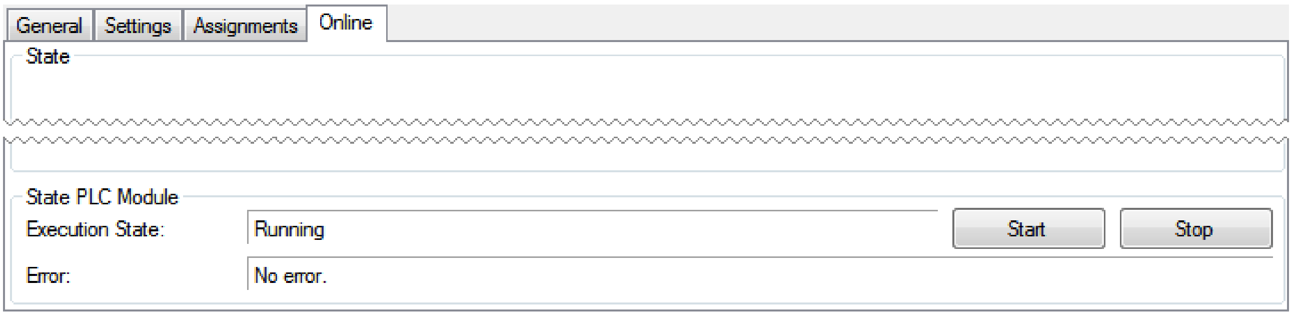
- **Baudrate:** Select the desired baud rate. The following values are available: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 230400, 460800.
- **Hardware FIFO (Byte):** Select the desired byte size of the Fifo. The following values are available: 16, 128.
- **Databits:** Select the desired number of data bits per byte. The following values are available: 8, 7, 6, 5.
- **Interface:** Select the type of RS to be used. The following values are available: RS232, RS485.

- **Stopbits:** Select the desired number of stop bits. The following values are available: 1, 2.
- **Parity:** Select the parity test to be used. The following options are available: None, Even, Odd, User.
- **Sync Mode:** The exchange of process data with the PLC is synchronised if *Sync Mode* is switched on.

Pressing the *Linked to...*-button opens a dialog, where the designated serial communication port can be assigned to this module.

In case of using RS232/RS485 via KL6031/KL6041 or KL6001/KL6021, there are no parameters to configure. Only select the appropriate channel of the terminal and it will be assigned to this module.

Online



Execution State: Running

Error: No error.

State PLC Module

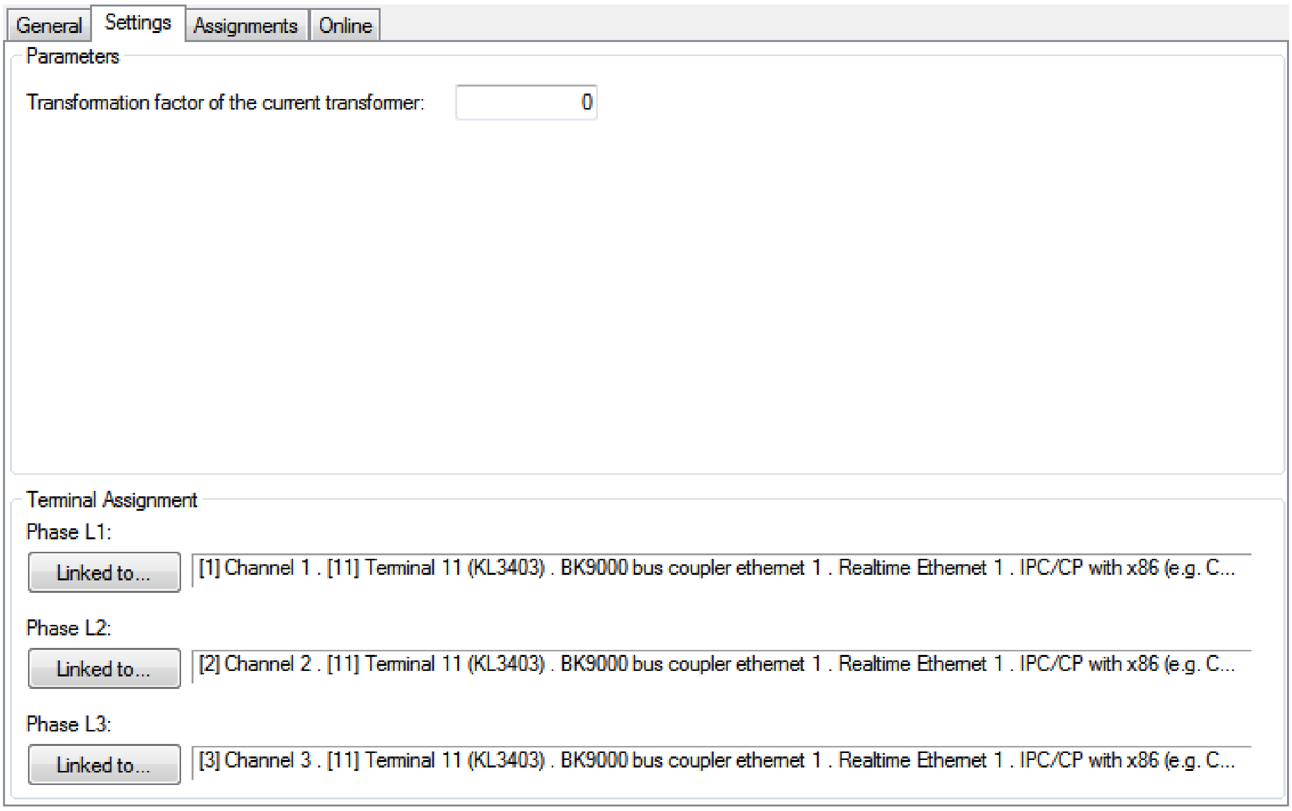
- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

This behavior is the same for RS232/RS485 via KL6031/KL6041 and KL6001/KL6021.

7.2.2.12.4 Power Measurement

The power measurement subsystem enables the measurement of electrical data for a three-phase supply network. Power is supplied via the 3-phase KL3403 power measurement terminal.

Settings



Parameters

Transformation factor of the current transformer: 0

Terminal Assignment

Phase L1:

[1] Channel 1 . [11] Terminal 11 (KL3403) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. C...

Phase L2:

[2] Channel 2 . [11] Terminal 11 (KL3403) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. C...

Phase L3:

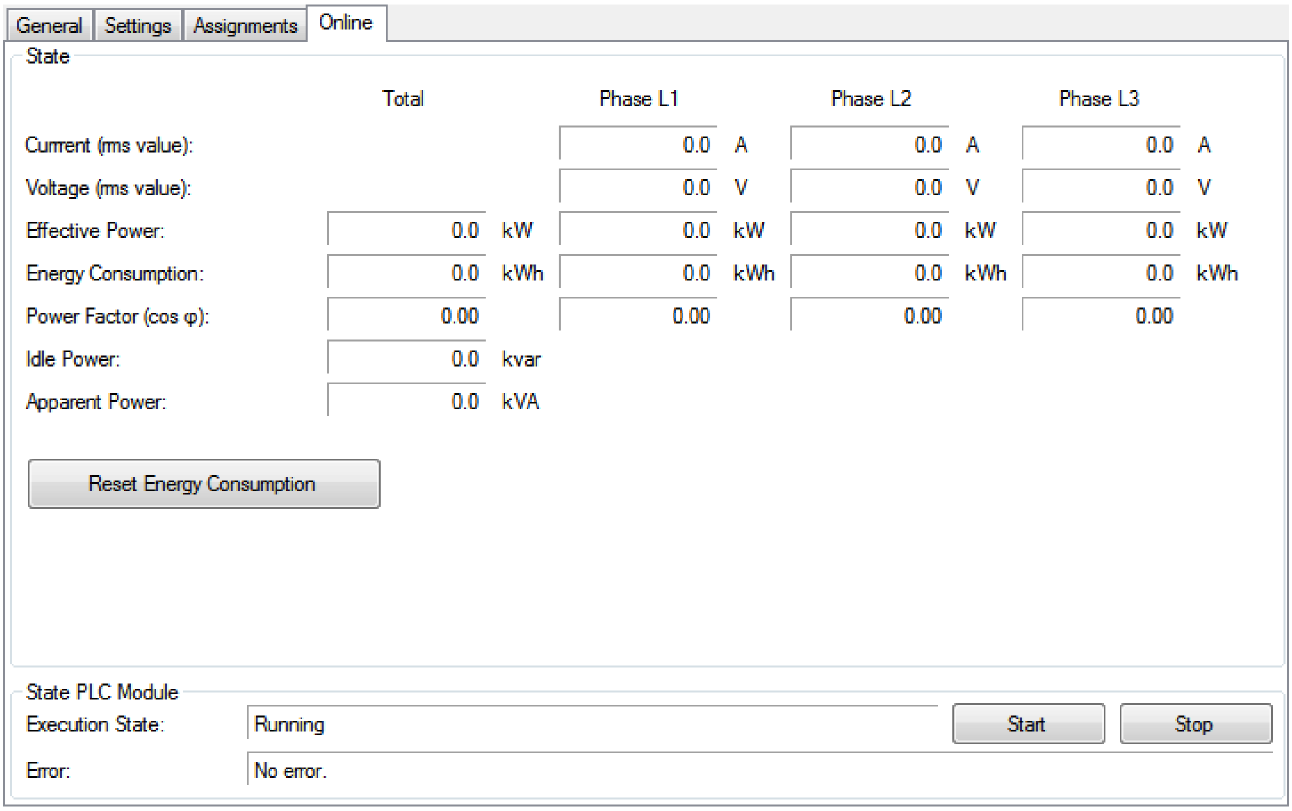
[3] Channel 3 . [11] Terminal 11 (KL3403) . BK9000 bus coupler ethernet 1 . Realtime Ethernet 1 . IPC/CP with x86 (e.g. C...

Parameters

- **Transformation factor of the current transformer:** Input field for the transformation factor of the current transformer.

Pressing the *Linked to...*-button opens a dialog, where the channels of an power measuring terminal can be assigned to this module.

Online



State

	Total		Phase L1		Phase L2		Phase L3	
Current (rms value):			0.0	A	0.0	A	0.0	A
Voltage (rms value):			0.0	V	0.0	V	0.0	V
Effective Power:	0.0	kW	0.0	kW	0.0	kW	0.0	kW
Energy Consumption:	0.0	kWh	0.0	kWh	0.0	kWh	0.0	kWh
Power Factor (cos φ):	0.00		0.00		0.00		0.00	
Idle Power:	0.0	kvar						
Apparent Power:	0.0	kVA						

State PLC Module

Execution State: Running

Error: No error.

Commands

- **Reset Energy Consumption:** Resets the energy consumption values.

State

- **Current (rms value):** In sum current.
- **Voltage (rms value):** Voltage value of all phases.
- **Effective Power:** In sum effective power.
- **Energy Consumption:** Overall energy consumption.
- **Power Factor (cos φ):** Power factor of all phases.
- **Idle Power:** Overall idle power.
- **Apparent Power:** In sum apparent power.

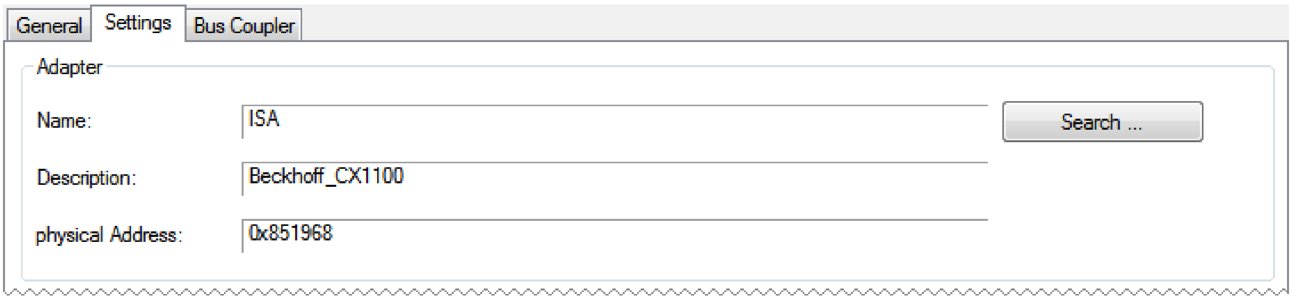
State PLC Module

- **Execution State:** Shows if the module is running or stopped.
- **Error:** In case of an error, the error code and the error parameter are displayed in this box.
- **Start/Stop:** Starts/stops the module.

7.2.3 I/O Configuration

7.2.3.1 I/O Devices CX1000\CX9000 Realtime Ethernet, CX1100-KB, CX9000 Power Supplier/Terminal Device

Settings



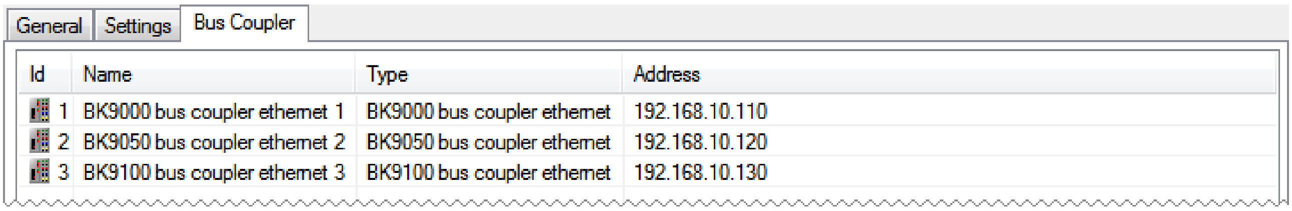
Name: ISA

Description: Beckhoff_CX1100

physical Address: 0x851968

If the search of an I/O device was successful, its information will appear here.

Bus Coupler

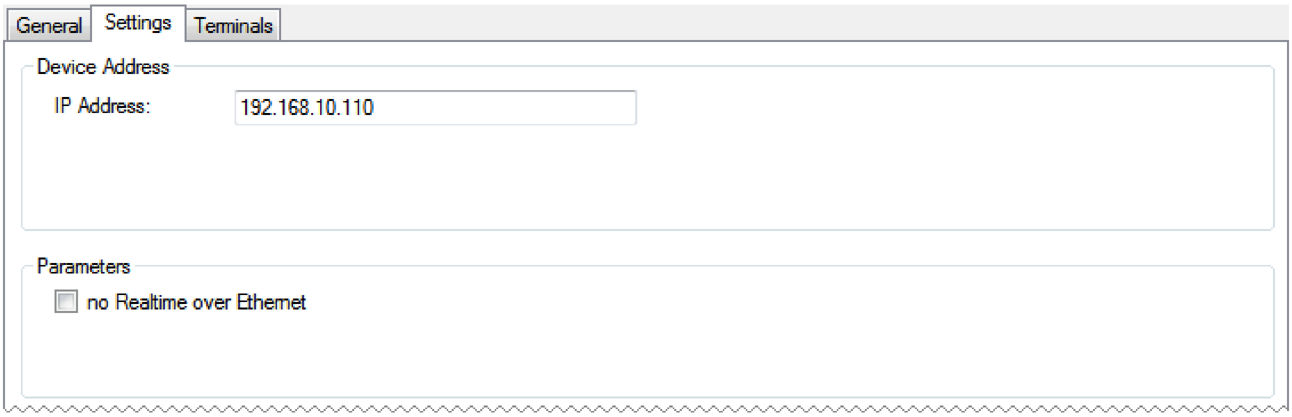


Id	Name	Type	Address
1	BK9000 bus coupler ethernet 1	BK9000 bus coupler ethernet	192.168.10.110
2	BK9050 bus coupler ethernet 2	BK9050 bus coupler ethernet	192.168.10.120
3	BK9100 bus coupler ethernet 3	BK9100 bus coupler ethernet	192.168.10.130

The *Bus Coupler*-menu lists all bus coupler of an selected I/O device.

7.2.3.2 Bus Coupler BK9000, BK9010, BK9050

Settings



Device Address

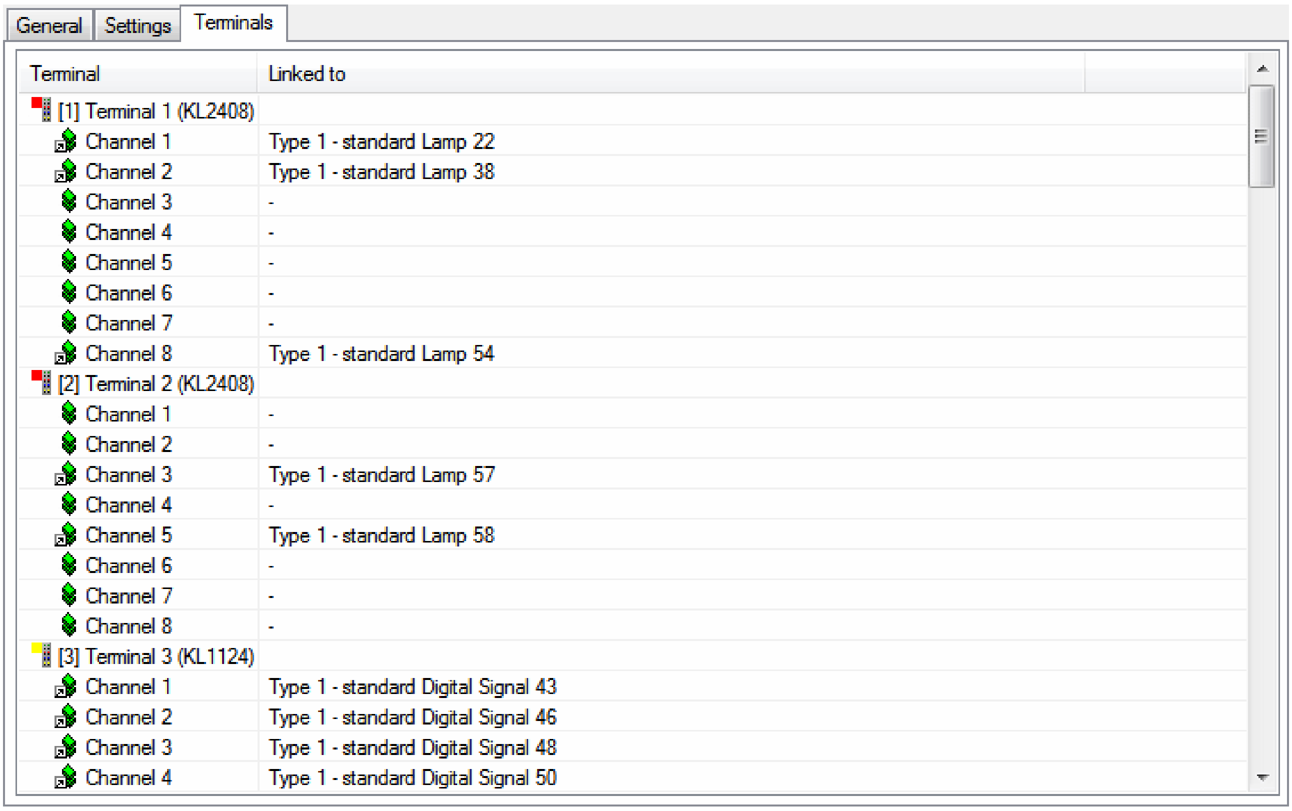
IP Address: 192.168.10.110

Parameters

□ no Realtime over Ethernet

Address and real time ethernet behavior of a bus coupler can be changed in this dialog.

Terminals



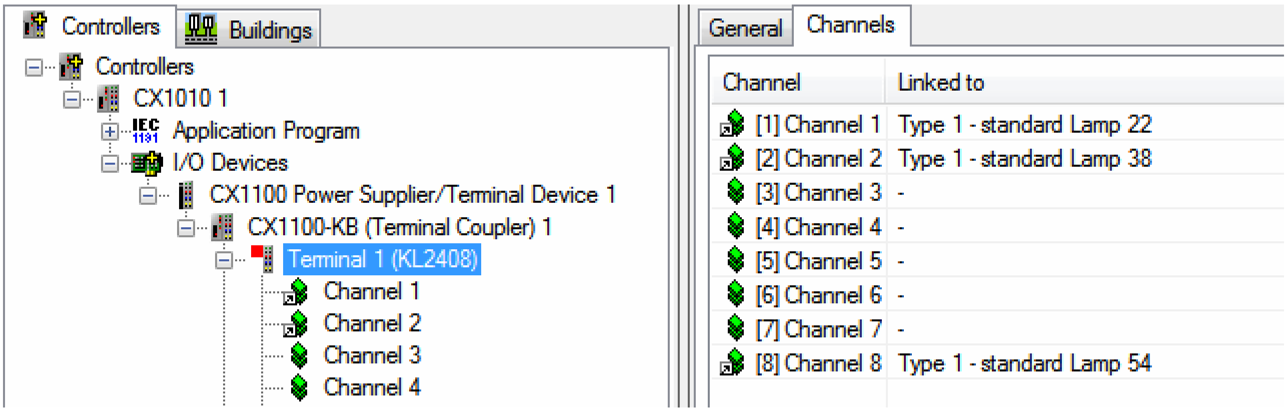
Terminal	Linked to
[1] Terminal 1 (KL2408)	
Channel 1	Type 1 - standard Lamp 22
Channel 2	Type 1 - standard Lamp 38
Channel 3	-
Channel 4	-
Channel 5	-
Channel 6	-
Channel 7	-
Channel 8	Type 1 - standard Lamp 54
[2] Terminal 2 (KL2408)	
Channel 1	-
Channel 2	-
Channel 3	Type 1 - standard Lamp 57
Channel 4	-
Channel 5	Type 1 - standard Lamp 58
Channel 6	-
Channel 7	-
Channel 8	-
[3] Terminal 3 (KL1124)	
Channel 1	Type 1 - standard Digital Signal 43
Channel 2	Type 1 - standard Digital Signal 46
Channel 3	Type 1 - standard Digital Signal 48
Channel 4	Type 1 - standard Digital Signal 50

The *Terminals*-menu lists all terminals.

To print the terminal-list of a coupler, click this coupler with the right mouse button. The known context menu with three additional entries appears:

- **Print...:** Prints the terminal-list of this coupler.
- **Print Preview...:** Shows the print-preview.
- **Page Setup...:** Settings of paper-size and margin.

7.2.3.3 Terminals

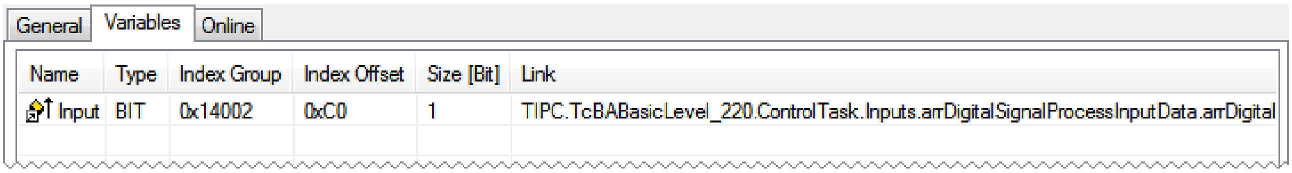


- Controllers
  - CX1010 1
    - Application Program
    - I/O Devices
      - CX1100 Power Supplier/Terminal Device 1
        - CX1100-KB (Terminal Coupler) 1
          - Terminal 1 (KL2408)
            - Channel 1
            - Channel 2
            - Channel 3
            - Channel 4

Channel	Linked to
[1] Channel 1	Type 1 - standard Lamp 22
[2] Channel 2	Type 1 - standard Lamp 38
[3] Channel 3	-
[4] Channel 4	-
[5] Channel 5	-
[6] Channel 6	-
[7] Channel 7	-
[8] Channel 8	Type 1 - standard Lamp 54

By clicking on a terminal in the navigation tree will show its channels and where they linked to. More information about a channel can be optained by clicking on it.

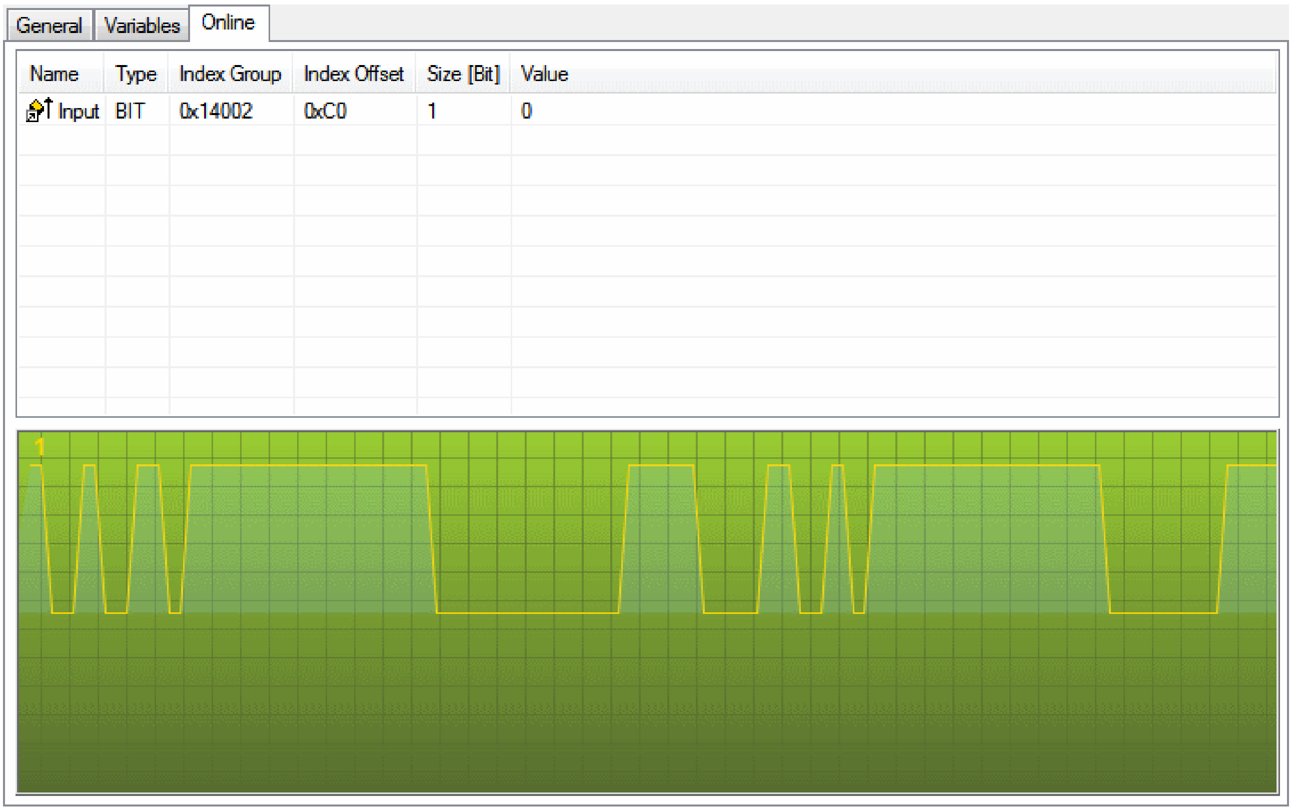
Variables



Name	Type	Index Group	Index Offset	Size [Bit]	Link
Input	BIT	0x14002	0xC0	1	TIPC.TcBABasicLevel_220.ControlTask.Inputs.arrDigitalSignalProcessInputData.arrDigital

This dialog provides basic information about the channel and where it is linked to.

Online



Name	Type	Index Group	Index Offset	Size [Bit]	Value
Input	BIT	0x14002	0xC0	1	0

The *Online* menu also provides basic information about the channel and the value of the channel.

7.3 Application Examples

7.3.1 Dyn. group allocation for lamps

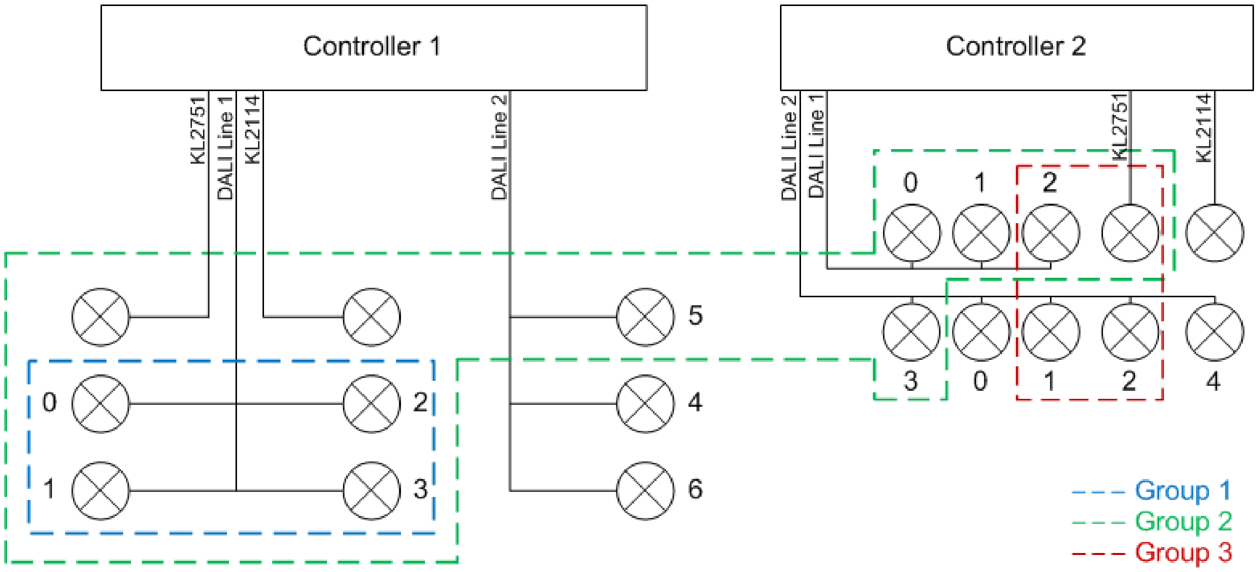
The realisation of lighting tasks can become a time-consuming task as soon as lamp groups are distributed over several controllers, in particular if the lamps are connected via subsystems such as DALI and a group of lamps extends over several DALI lines.

The TwinCAT Building Automation Framework is specialised for simplifying the commissioning of such applications as far as possible. This is achieved by various means:

- the *lamp group* object supports various lamp types (DALI, dimmer terminal and digital output). If need be these can be combined into a lamp group.
- the *lamp group* object can manage lamps that are distributed over different lines and also over different controllers. The TwinCAT Building Automation Manager automatically generates the necessary network variables.
- the *lamp group* object assigns the individual DALI lamps to the necessary DALI groups via the *Subsystem DALI* object. This takes place automatically while the configuration is activated. It is not necessary to implement the assignment of the DALI lamps to the DALI groups separately (possibly with a further commissioning tool).

Grouping

The combination of several lamps of a controller into a group represents the simplest form of grouping. Independent of the actual type of lamp, the desired lamps can be grouped and operated, even beyond the limits of the controller. Nesting is possible for more complex lighting scenarios, whereby lamps that already belong to a group can be assigned to further groups.



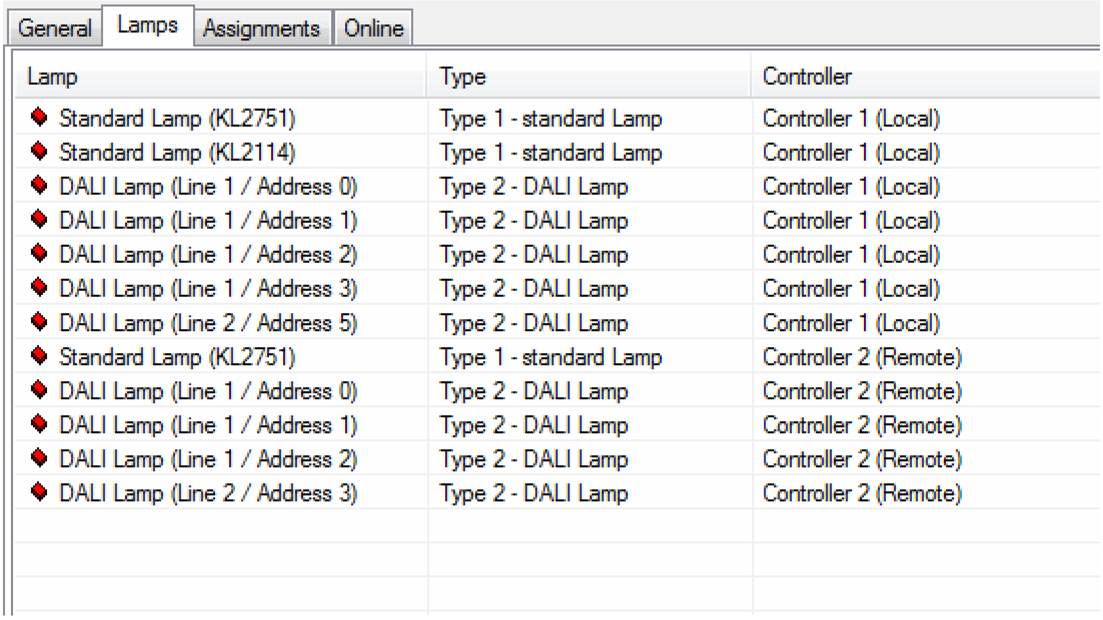
Group 1

A simple combination of DALI lamps on a DALI line into a group. Since all DALI lamps are located on one DALI line, these are assigned to one DALI group.

Group 2

Lamp group 2 represents the most extensive variant. 5 DALI lamps from controller 1 are assigned to the lamp group. Furthermore, a non-dimmable standard lamp (KL2114) and a dimmable standard lamp (KL2751) from controller 1 are in lamp group 2.

From controller 2, three DALI lamps from DALI line 1 and one DALI lamp from DALI line 2 are assigned to lamp group 2. In addition, a further standard lamp (KL2751) is added.



Lamp	Type	Controller
◆ Standard Lamp (KL2751)	Type 1 - standard Lamp	Controller 1 (Local)
◆ Standard Lamp (KL2114)	Type 1 - standard Lamp	Controller 1 (Local)
◆ DALI Lamp (Line 1 / Address 0)	Type 2 - DALI Lamp	Controller 1 (Local)
◆ DALI Lamp (Line 1 / Address 1)	Type 2 - DALI Lamp	Controller 1 (Local)
◆ DALI Lamp (Line 1 / Address 2)	Type 2 - DALI Lamp	Controller 1 (Local)
◆ DALI Lamp (Line 1 / Address 3)	Type 2 - DALI Lamp	Controller 1 (Local)
◆ DALI Lamp (Line 2 / Address 5)	Type 2 - DALI Lamp	Controller 1 (Local)
◆ Standard Lamp (KL2751)	Type 1 - standard Lamp	Controller 2 (Remote)
◆ DALI Lamp (Line 1 / Address 0)	Type 2 - DALI Lamp	Controller 2 (Remote)
◆ DALI Lamp (Line 1 / Address 1)	Type 2 - DALI Lamp	Controller 2 (Remote)
◆ DALI Lamp (Line 1 / Address 2)	Type 2 - DALI Lamp	Controller 2 (Remote)
◆ DALI Lamp (Line 2 / Address 3)	Type 2 - DALI Lamp	Controller 2 (Remote)

The 4 DALI lamps (on DALI line 1) from controller 1 are assigned to one DALI group. This assignment takes place automatically on activation of the configuration. The individual DALI lamps from controller 1 and the DALI lamps from controller 2 are addressed by short address.

The *lamp group* object ensures the correct distribution of the necessary commands. It is irrelevant whether the object is operated via the online dialogue of the TwinCAT Building Automation Manager or by the PLC function block FB_BAFLampGroupStandard [▶ 230].

As soon as two or more DALI lamps of a DALI line are assigned to a local lamp group, the TwinCAT Building Automation Manager assigns these to a DALI group. In applications with many groups of lamps it may be necessary to raise the threshold value (the number of DALI lamps starting from which the lamps are assigned to a DALI group). Thus, a DALI lamp can be assigned to more than 16 lamp groups. The parameters can be individually set for each DALI line (see DALI line subsystem [▶ 172]).

Group 3

Lamp group 3 contains four lamps. two DALI lamps from DALI line 2, one DALI lamp from DALI line 1 and a dimmable standard lamp (KL2751). The DALI lamps from DALI line 2 are assigned to a DALI group. The DALI lamp from line 1 is addressed by short address, since it is the only DALI lamp from its DALI line in this lamp group.

7.3.2 Scene cascading

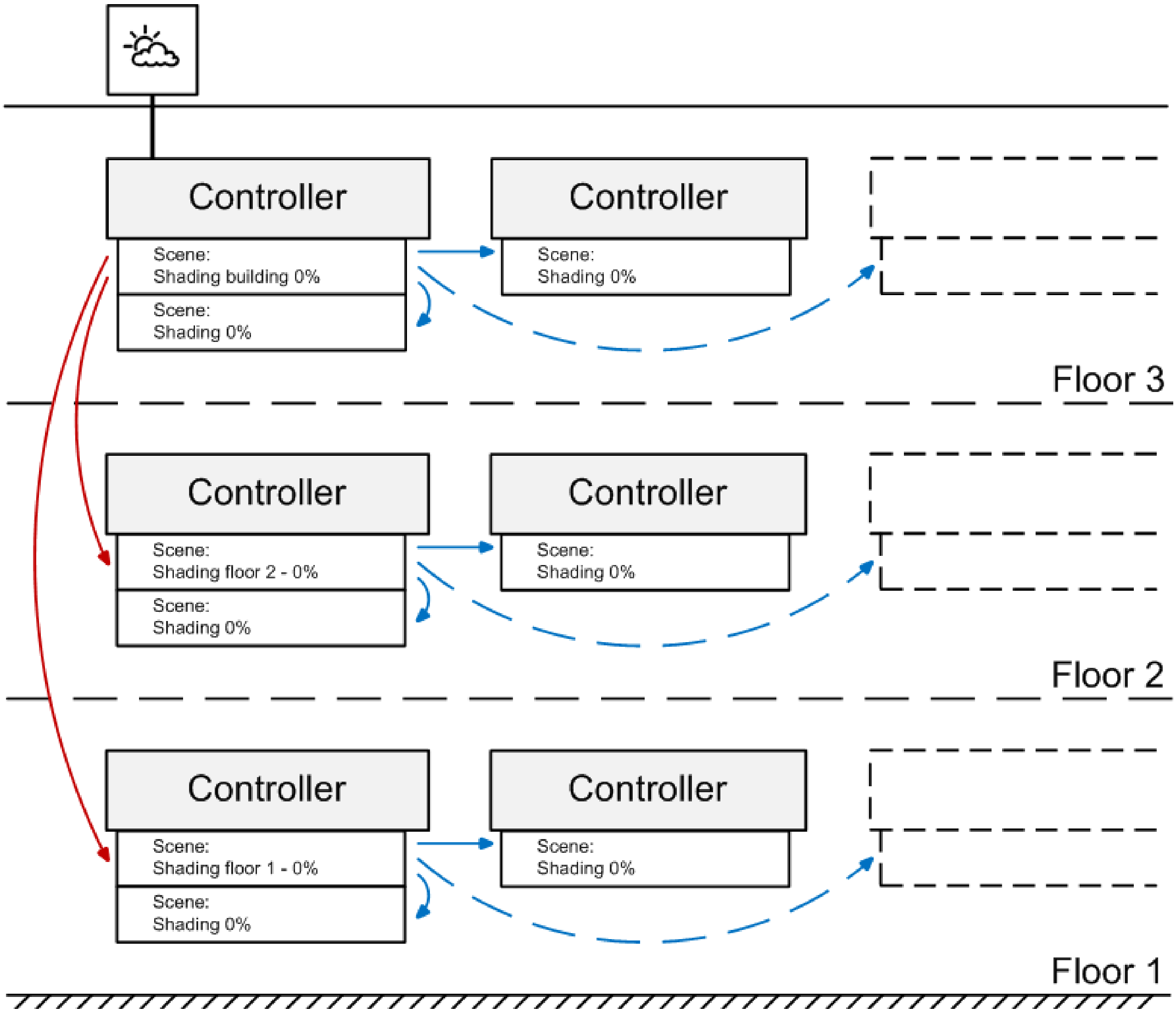
The *Scene* object offers the possibility to define a series of commands (see Standard Scene [▶ 150]). Hence actuators can be controlled by the call of a scene. The specialty of the TwinCAT Building Automation Framework's scene management is the possibility for one scene to call another. This scene can also be defined on another controller. The TwinCAT Building Automation Manager automatically creates the necessary network variables.

The following example shows how a building with several controllers automatically reacts to events with the aid of scenes. If the wind velocity exceeds a certain threshold value, all sunblinds are to be automatically driven upwards.

A useful variant is to create a scene on each controller that drives all the controller's blinds upwards. The controller that evaluates the data from the weather station activates this scene on all controllers. The TwinCAT Building Automation Manager creates the necessary network variables on activation of the configuration. These are also optimized in such a way that the load on the network is minimal (e.g., using multicast network variables). Hence, all scenes in the building are activated in a simple manner to drive the sunblinds upwards.

A further variant involves the cascading of scene calls. In this example the cascading takes place over two levels. On each floor there is one controller that contains the scene with which all controllers on the respective floor are addressed. The controller with the weather station now no longer needs to call the scenes on all controllers. It suffices when the scene is called that activates all further scenes on the floor.

Cascading offers the possibility to react more flexibly to changes. If a certain floor is no longer to be included, this can be achieved by removing a single command from the scene in the controller with the weather station.



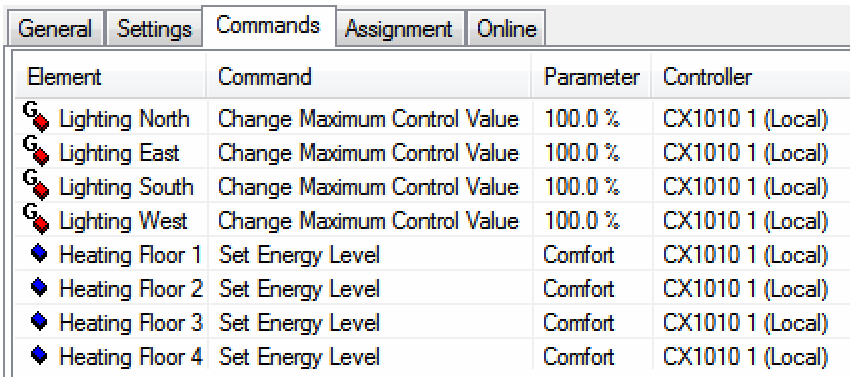
7.3.3 Change of operating states

The efficiency of a building is determined by a great many factors. Among other things by the building automation employed. Depending on the structure and complexity, efficiency during operation can be increased and costs can be lowered. A simple and effective method is the temporal division of the building into two operating conditions: productive and non-productive time.

The *Productive time* and *Non-productive time* scenes and a weekly timer channel (individual days of the week), *Mode change*, are added to the project. The *Productive time* scene is configured first.

Productive time

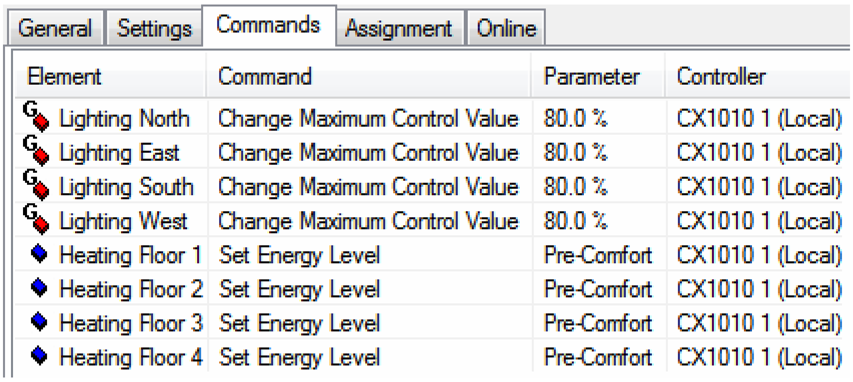
The brightness of the lighting in the building should be 100 % during the productive time. To this end the *Productive time* scene contains a command entry in the *Commands* dialogue for each lamp group that changes the maximum control value. Accordingly, the energy level for the heaters is set to *Comfort*.



Element	Command	Parameter	Controller
Lighting North	Change Maximum Control Value	100.0 %	CX1010 1 (Local)
Lighting East	Change Maximum Control Value	100.0 %	CX1010 1 (Local)
Lighting South	Change Maximum Control Value	100.0 %	CX1010 1 (Local)
Lighting West	Change Maximum Control Value	100.0 %	CX1010 1 (Local)
Heating Floor 1	Set Energy Level	Comfort	CX1010 1 (Local)
Heating Floor 2	Set Energy Level	Comfort	CX1010 1 (Local)
Heating Floor 3	Set Energy Level	Comfort	CX1010 1 (Local)
Heating Floor 4	Set Energy Level	Comfort	CX1010 1 (Local)

Non-productive time

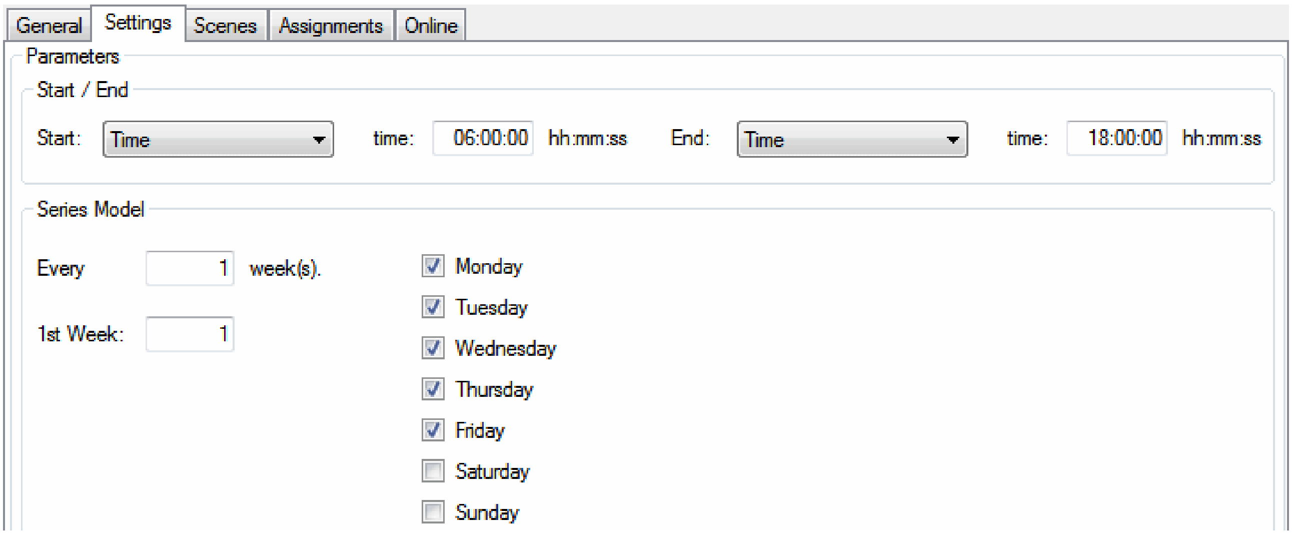
During the non-productive time all lamp groups are set to a maximum control value of 80 % and the heaters to the *Standby* energy level.



Element	Command	Parameter	Controller
Lighting North	Change Maximum Control Value	80.0 %	CX1010 1 (Local)
Lighting East	Change Maximum Control Value	80.0 %	CX1010 1 (Local)
Lighting South	Change Maximum Control Value	80.0 %	CX1010 1 (Local)
Lighting West	Change Maximum Control Value	80.0 %	CX1010 1 (Local)
Heating Floor 1	Set Energy Level	Pre-Comfort	CX1010 1 (Local)
Heating Floor 2	Set Energy Level	Pre-Comfort	CX1010 1 (Local)
Heating Floor 3	Set Energy Level	Pre-Comfort	CX1010 1 (Local)
Heating Floor 4	Set Energy Level	Pre-Comfort	CX1010 1 (Local)

Timer switch channel

Finally, the timer switch channel is configured in the *Settings* dialogue so that the scene can be called depending on the day of the week and the time. A normal working week from Monday to Friday is assumed in which the main operating time of the building is defined from 6 am to 6 pm.



Parameters

Start / End

Start: Time time: 06:00:00 hh:mm:ss End: Time time: 18:00:00 hh:mm:ss

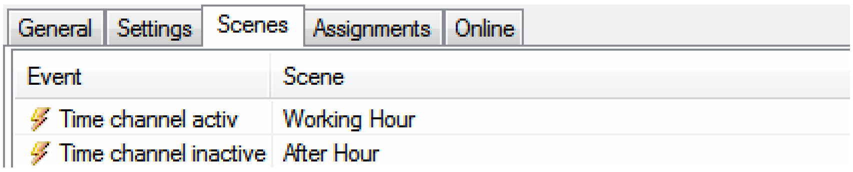
Series Model

Every 1 week(s).

1st Week: 1

☑ Monday
☑ Tuesday
☑ Wednesday
☑ Thursday
☑ Friday
☐ Saturday
☐ Sunday

The assignment of the scene calls takes place in the *Scenes* dialogue, depending on the state of the timer switch channel. The *Productive time* scene is to be called when switching on and the *Non-productive time* scene when switching off.



Event	Scene
Time channel activ	Working Hour
Time channel inactive	After Hour

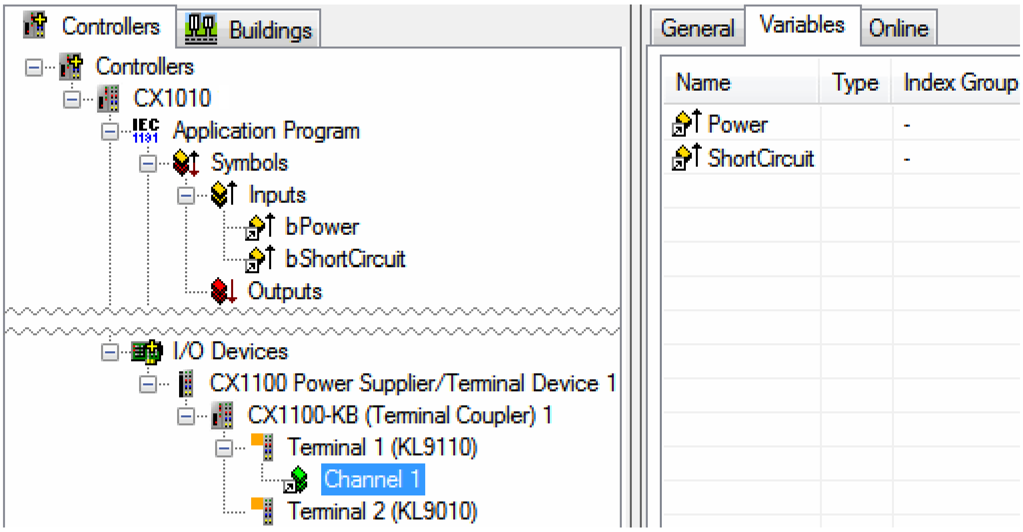
After activation of the configuration the building, controlled by the timer, will change its operating states for the specified time and day of the week.

7.3.4 Linking of power supply terminals with diagnostics

The power supply terminals with diagnostics have a channel with two input bits and therefore cannot easily be linked to a digital signal in the TwinCAT Building Automation Manager. The procedure to obtain the values is slightly different from the normal way.

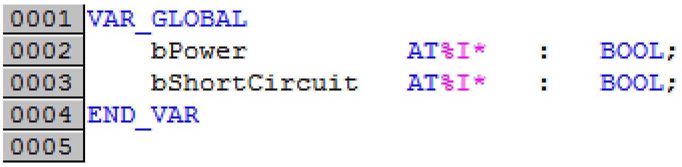
Procedure

First, the input bits of the terminal will be routed by PLC symbols into the PLC. Create two input variables of type BOOL within the PLC program and re-read the *.tpy file in the TwinCAT Building Automation Manager. Then link the PLC symbols to the input bits of the terminal.

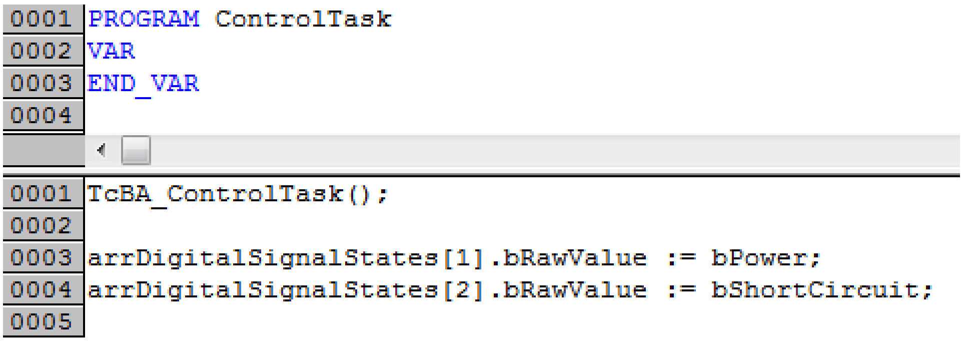


Name	Type	Index Group
Power		-
ShortCircuit		-

Thus, the PLC symbols can be used like normal digital signals, you have to create two digital signals and keep the object ids (see Object Id [▶ 200]) in mind. With this object ids now the created digital signals can be overwritten with the values of the PLC symbols anywhere in the PLC program.



```
VAR_GLOBAL
    bPower          AT%I*   :   BOOL;
    bShortCircuit   AT%I*   :   BOOL;
END_VAR

```



```
PROGRAM ControlTask
VAR
END_VAR

```

```
TcBA_ControlTask();

arrDigitalSignalStates[1].bRawValue := bPower;
arrDigitalSignalStates[2].bRawValue := bShortCircuit;

```

Download area

Download area

Example based on the KL9110.

- https://infosys.beckhoff.com/content/1033/tcbaframework/Resources/11368459403/.zip

8 PLC Function Blocks

The objects configured in the TwinCAT Building Automation Framework can also be used with the following function blocks from the PLC. Performance features that are not covered by the standard functions of the TwinCAT Building Automation Framework can thus be implemented with the help of these function blocks. The function blocks are contained in the PLC library TcBAFramework2.20.lib.

The function blocks are firmly bound to the respective type. This means, for example, that only one DALI lamp can be addressed with the function block*FB_BAFLampDALI()*. An error message is returned if an attempt is made to address a standard lamp object or a DALI lamp for the security lighting.

Sensors

Name of the Function Block	Description
FB_BAFDigitalSignalStandard [▸ 206]	Function block for controlling a digital signal.
FB_BAFDigitalSignalEnOcean [▸ 207]	Function block for controlling a digital EnOcean signal.
FB_BAFDigitalSignalSmartSPOT [▸ 208]	Function block for controlling a digital signal (occupancy) of a smartSPOT.
FB_BAFDigitalSignalUpDownCounterKL1512 [▸ 209]	Function block for controlling an up/down counter via KL1512.
FB_BAFAnalogSignalStandard [▸ 210]	Function block for controlling an analog signal.
FB_BAFAnalogSignalEnOcean [▸ 211]	Function block for controlling an analog EnOcean signal.
FB_BAFAnalogSignalSmartSPOT [▸ 212]	Function block for controlling an analog signal (brightness) of a smartSPOT.

Actuators

Name of the Function Block	Description
FB_BAFLampStandard [▸ 214]	Function block for controlling a lamp.
FB_BAFLampDALI [▸ 216]	Function block for controlling a DALI lamp.
FB_BAFLampDALIEmergencyLighting [▸ 218]	Function block for controlling a DALI lamp for emergency-lighting.
FB_BAFBlindDriveStandard [▸ 220]	Function block for controlling a blind drive.
FB_BAFBlindDriveKL2532 [▸ 221]	Function block for controlling a blind drive via KL2532.
FB_BAFWindowDriveStandard [▸ 223]	Function block for controlling a window drive.
FB_BAFValveActuator2Point [▸ 224]	Function block for controlling a 2-point valve actuator.
FB_BAFValveActuator3Point [▸ 225]	Function block for controlling a 3-point valve actuator.
FB_BAFValveActuatorContinuous [▸ 226]	Function block for controlling a continuous valve actuator.

Sensor Groups

Name of the Function Block	Description
FB_BAFDigitalSignalGroupStandard [▸ 227]	Function block for controlling a digital signal group.
FB_BAFAnalogSignalGroupStandard [▸ 228]	Function block for controlling an analog signal group.

Actuator Groups

Name of the Function Block	Description
FB_BAFLampGroupStandard [▶ 230]	Function block for controlling a lamp group.
FB_BAFBlindDriveGroupStandard [▶ 232]	Function block for controlling a blind drive group.
FB_BAFWindowDriveGroupStandard [▶ 233]	Function block for controlling a window drive group.
FB_BAFValveActuatorGroupStandard [▶ 235]	Function block for controlling a valve actuator group.

Functional Units

Name of the Function Block	Description
FB_BAFLightingDimmerSwitch [▶ 236]	Function block for controlling a lighting with dimmer function.
FB_BAFLightingStairwellDimmer [▶ 237]	Function block for controlling a stairwell lighting.
FB_BAFLightingTwilightSwitch [▶ 238]	Function block for controlling a twighlight switch.
FB_BAFLightingDaylightLightControl [▶ 241]	Function block for controlling a daylight dependent light control.
FB_BAFLightingConstantLightRegulation [▶ 243]	Function block for controlling a constant light control.
FB_BAFLightingTestsForEmergencyLighting [▶ 245]	Function block for controlling function- and duration tests of DALI lamps.
FB_BAFSunblindStandard [▶ 246]	Function block for controlling a sunblind.
FB_BAFSunblindSuntracking [▶ 247]	Function block for controlling a sunblind with sun position tracking.
FB_BAFWindowStandard [▶ 249]	Function block for controlling a window unit.
FB_BAFHVACHeatingCooling [▶ 250]	Function block for controlling a temperature controller for heating and cooling.

Scenes

Name of the Function Block	Description
FB_BAFSceneStandard [▶ 252]	Function block for controlling a scene.

Time Scheduler Channels

Name of the Function Block	Description
FB_BAFSchedulerDaily [▶ 253]	Function block for controlling a daily time scheduler channel.
FB_BAFSchedulerWeeklyIndividual [▶ 254]	Function block for controlling a weekly time scheduler channel for individual days of the week.
FB_BAFSchedulerWeeklyContinual [▶ 254]	Function block for controlling a weekly time scheduler channel for continual days of the week.
FB_BAFSchedulerMonthlyNthDay [▶ 255]	Function block for controlling a monthly time scheduler channel for the nth day of the month.
FB_BAFSchedulerMonthlyIndividualWeekday [▶ 256]	Function block for controlling a monthly time scheduler channel for an individual weekday of the month.

Weather Stations

Name of the Function Block	Description
FB_BAFWeatherStationVirtual [▸ 257]	Function block for controlling a virtual weather station.
FB_BAFWeatherStationElsnerP03 [▸ 260]	Function block for controlling a P03 weather station of the Elsner Company.
FB_BAFWeatherStationThiesClimaSensorD [▸ 264]	Function block for controlling a Sensor D weather station of the Thies Clima Company.
FB_BAFWeatherStationAnalogDigitalSignals [▸ 268]	Function block for controlling a weather station composed of analog and digital signals.

Subsystems

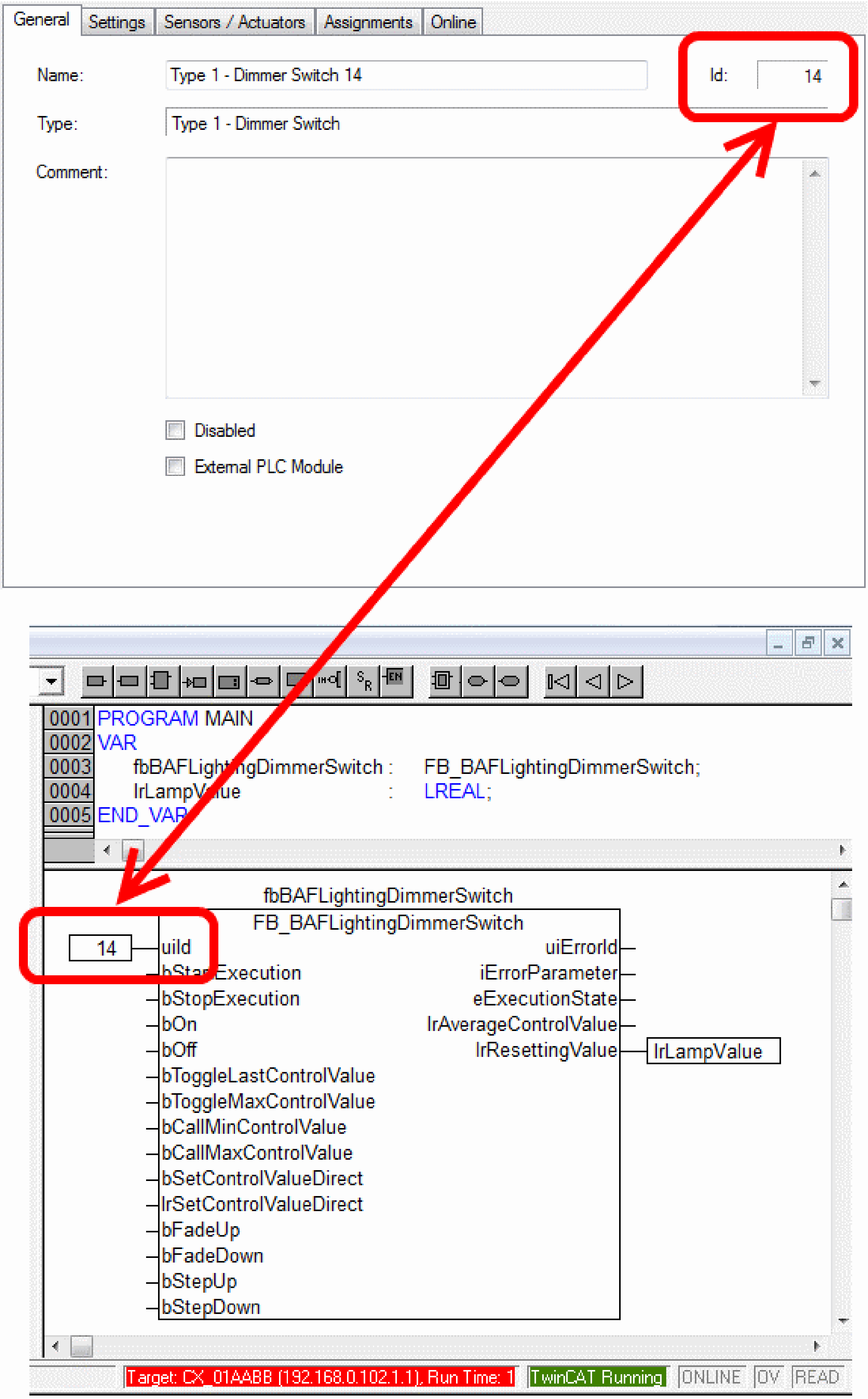
Name of the Function Block	Description
FB_BAFPowerMeasurementKL3403 [▸ 271]	Function block for controlling a power measurement unit via KL3403.

Application

Name of the Function Block	Description
FB_BAFApplicationProgram [▸ 273]	Function block for controlling global services of the application program.

8.1 Object Id

Each object possesses an ID, with whose assistance the function blocks address the individual objects. You will find the ID in the same dialog in the TwinCAT Building Automation Manager in which the name of the object is also specified. This ID is also specified as an input parameter for each function block. The connection between the function block and the configured object in the TwinCAT Building Automation Manager is established with the help of this ID.



Name: Type 1 - Dimmer Switch 14

Id: 14

Type: Type 1 - Dimmer Switch

Comment:

□ Disabled

□ External PLC Module

```
PROGRAM MAIN
VAR
    fbBAFLightingDimmerSwitch :    FB_BAFLightingDimmerSwitch;
    lrLampValue               :    LREAL;
END_VAR
```

fbBAFLightingDimmerSwitch

FB_BAFLightingDimmerSwitch

Inputs	Outputs
uiId = 14	uiErrorId
bStartExecution	iErrorParameter
bStopExecution	eExecutionState
bOn	lrAverageControlValue
bOff	lrResettingValue → lrLampValue
bToggleLastControlValue	
bToggleMaxControlValue	
bCallMinControlValue	
bCallMaxControlValue	
bSetControlValueDirect	
lrSetControlValueDirect	
bFadeUp	
bFadeDown	
bStepUp	
bStepDown	

Each ID begins with 1. The upper limit can be freely specified by defining certain constants in the PLC. Further information on the constants can be found in the Appendix [▸ 202].

In the TwinCAT Building Automation Manager, the upper limit for each object type is displayed in the application program area under the Performance features [▸ 57] dialog. Note that only objects that were created and activated in the TwinCAT Building Automation Manager can be addressed from the PLC. The PLC block returns an error message [▸ 276] if an attempt is made to address an object that does not exist, has not yet been activated or whose ID lies outside the permitted range. Also, the objects can be addressed only if the entire application program is in the RUN state.

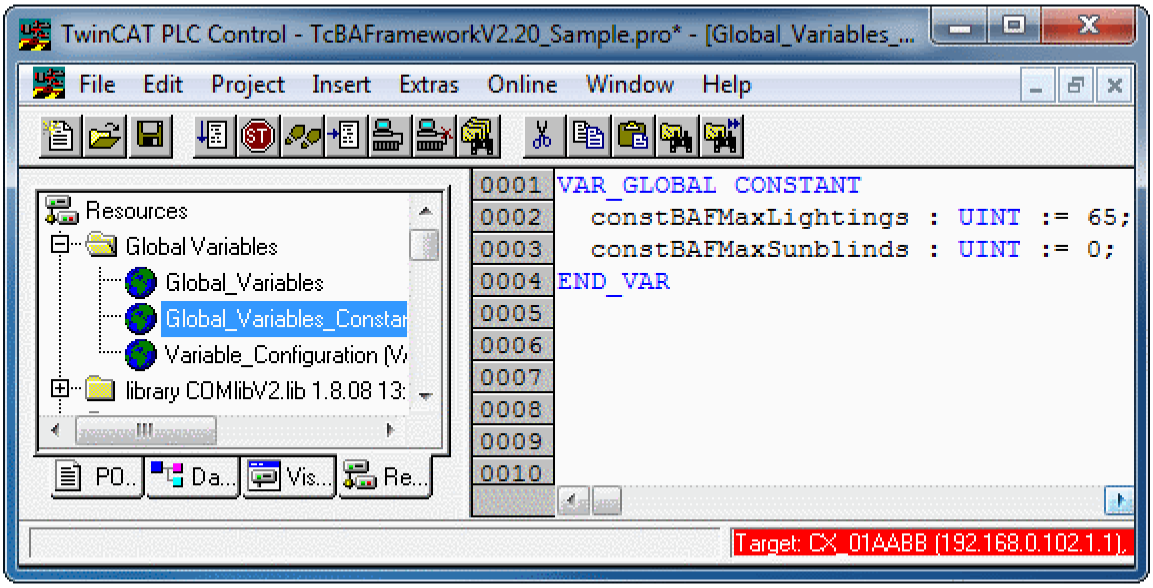
8.2 Constants

By defining constants in the PLC it is possible to specify how many objects of a certain type can be created. If the appropriate constant does not exist, then the default value listed below is used.

If you change one of the constants you must recompile the PLC program. Subsequently, the TPY file must be read in again and the project re-activated in the TwinCAT Building Automation Manager.

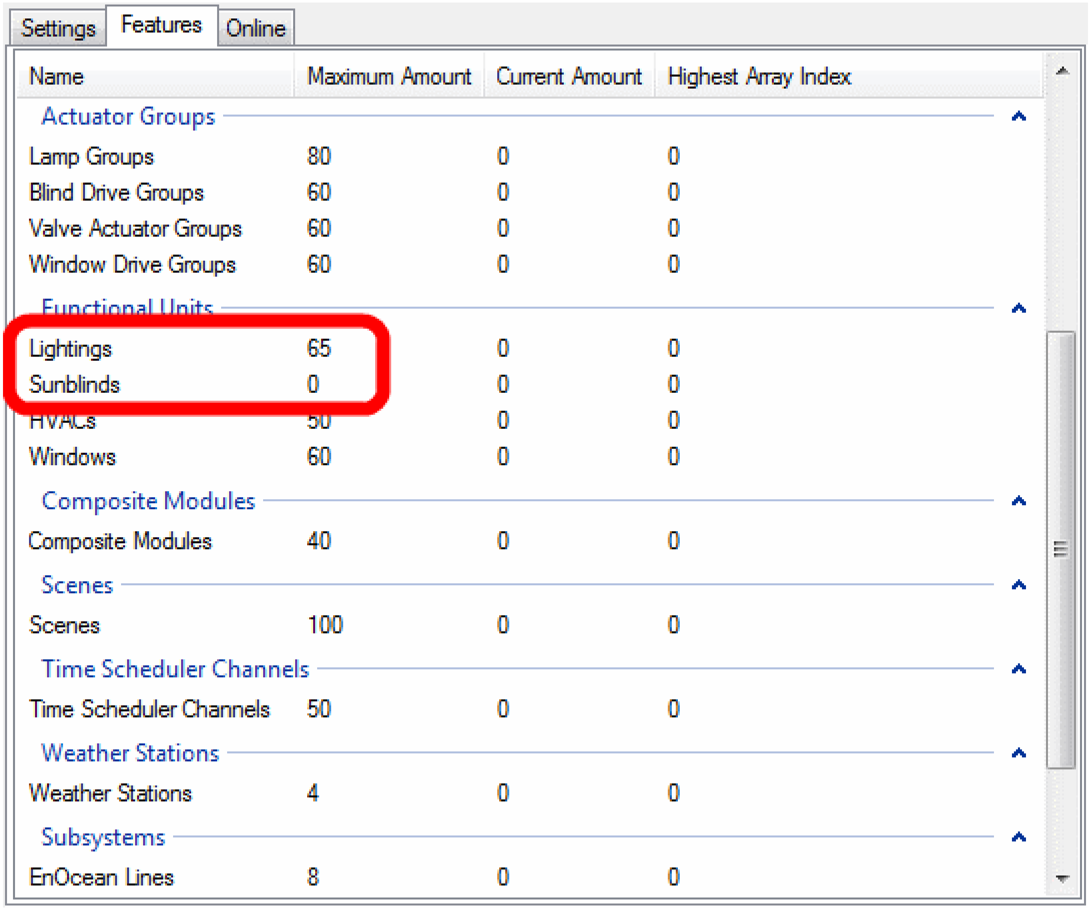
Unused objects also use resources on the target system. Do not set the constants any highly than is necessary for the application. A reserve of approx. 10% can be taken as a guide value here. If it is to be expected that, for example, you will have a maximum of 142 lighting actuators, then you should set the constant *constBAFMaxLamps* to 157.

In the example below, the maximum number of lights is set to 65 and that of the sun blinds to 0.


TwinCAT PLC Control - TcBAFrameworkV2.20_Sample.pro* - [Global_Variables_...
File
Edit
Project
Insert
Extras
Online
Window
Help
Resources
Global Variables
Global_Variables
Global_Variables_Constar
Variable_Configuration (V
library COMlibV2.lib 1.8.08 13:
PO..
Da...
Vis...
Re...
Target: CX_01AABB (192.168.0.102.1.1),

```
VAR_GLOBAL CONSTANT
   constBAFMaxLightings : UINT := 65;
   constBAFMaxSunblinds : UINT := 0;
END_VAR
```

In the TwinCAT Building Automation Manager, the value of the respective constant is displayed in the application program area under the Performance features [▸ 57] dialog.



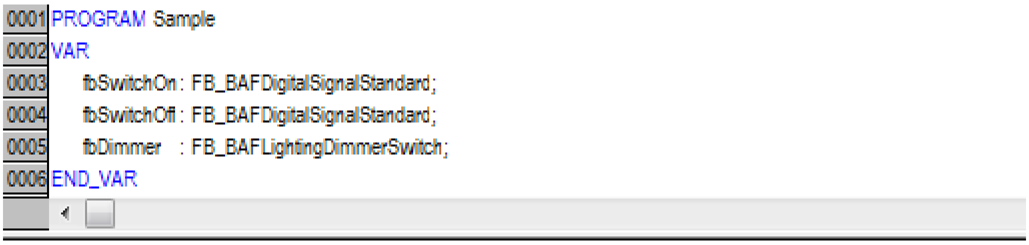
Name	Maximum Amount	Current Amount	Highest Array Index
Actuator Groups			
Lamp Groups	80	0	0
Blind Drive Groups	60	0	0
Valve Actuator Groups	60	0	0
Window Drive Groups	60	0	0
Functional Units			
Lightings	65	0	0
Sunblinds	0	0	0
HVACs	50	0	0
Windows	60	0	0
Composite Modules			
Composite Modules	40	0	0
Scenes			
Scenes	100	0	0
Time Scheduler Channels			
Time Scheduler Channels	50	0	0
Weather Stations			
Weather Stations	4	0	0
Subsystems			
EnOcean Lines	8	0	0

The table shows the names of the individual constants and their default values. If a constant is not re-declared, then the appropriate default value is used.

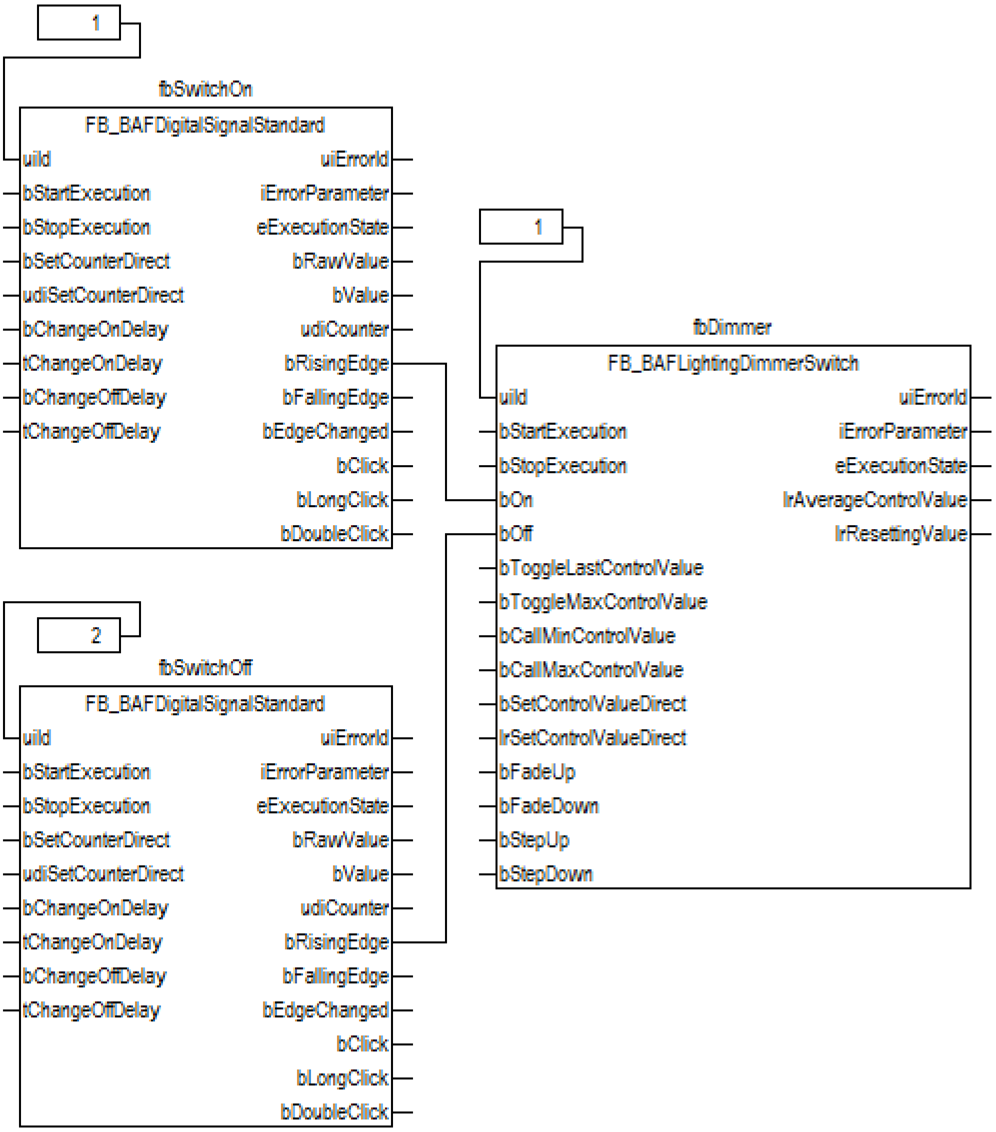
Sensors	Name of the constant	Default value
Digital Signals	constBAFMaxDigitalSignals	80
Analog Signals	constBAFMaxAnalogSignals	50
Actuators		
Lamps	constBAFMaxLamps	80
Blind Drives	constBAFMaxBlindDrives	60
Valve Actuators	constBAFMaxValveActuators	60
Window Drives	constBAFMaxWindowDrives	60
Sensor Groups		
Digital Signal Groups	constBAFMaxDigitalSignalGroups	80
Analog Signal Groups	constBAFMaxAnalogSignalGroups	50
Actuator Groups		
Lamp Groups	constBAFMaxLampGroups	80
Blind Drive Groups	constBAFMaxBlindDriveGroups	60
Valve Drive Groups	constBAFMaxValveActuatorGroups	60
Window Drive Groups	constBAFMaxWindowDriveGroups	60
Functional Units		
Lightings	constBAFMaxLightings	80
Sunblinds	constBAFMaxSunblinds	60
HVAC	constBAFMaxHVACs	50
Windows	constBAFMaxWindows	60
Composite Modules		
Composite Modules	constBAFMaxCompositeModules	40
Scenes		
Scenes	constBAFMaxScenes	100
Time Scheduler Channels		
Time Scheduler Channels	constBAFMaxTimeSchedulerChannels	50
Weather Stations		
Weather Stations	constBAFMaxWeatherStations	4
Subsystems		
EnOcean Lines	constBAFMaxEnOceanLines	8
DALI Lines	constBAFMaxDALILines	8
Serial Communication Lines	constBAFMaxSerialCommunicationLines	4
Power Measurement Lines	constBAFMaxPowerMeasurementLines	4

8.3 CFC Sample

In the example below, two digital signals (with the ID 1 and the ID 2) control lighting (with the ID 1). In this case the buttons and the lighting are not linked in the TwinCAT Building Automation Manager, but by the manual connection of the two PLC blocks in the PLC program.



```
PROGRAM Sample
VAR
    fbSwitchOn : FB_BAFDigitalSignalStandard;
    fbSwitchOff : FB_BAFDigitalSignalStandard;
    fbDimmer    : FB_BAFLightingDimmerSwitch;
END_VAR
```

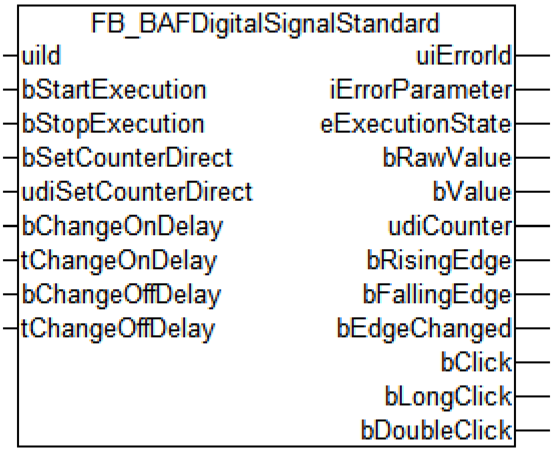


The free extendability by the user’s own PLC blocks makes it possible to implement performance features that are not possible by configuration in the TwinCAT Building Automation Manager alone. All types of IEC61131-3 representation are supported. In addition, all TwinCAT supplements and TwinCAT PLC libraries are at the programmer’s disposal.

8.4 Sensors

8.4.1 Digital Signals

8.4.1.1 FB_BAFDigitalSignalStandard



Function-block for controlling a Standard Digital Signal [▶ 61] directly out of the PLC-program.

VAR_INPUT

```
uiId                      : UINT;
bStartExecution           : BOOL;
bStopExecution            : BOOL;
bSetCounterDirect         : BOOL;
udiSetCounterDirect       : UDINT;
bChangeOnDelay            : BOOL;
tChangeOnDelay            : TIME;
bChangeOffDelay           : BOOL;
tChangeOffDelay           : TIME;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the digital signal.

bStopExecution: Stops the instance of the digital signal.

bSetCounterDirect: Sets the counter to the value of the input *udiSetCounterDirect*.

udiSetCounterDirect: See *bSetCounterDirect*.

bChangeOnDelay: Sets the time delay-on to the value of the input *tChangeOnDelay*.

tChangeOnDelay: See *bChangeOnDelay*.

bChangeOffDelay: Sets the time delay-off to the value of the input *tChangeOffDelay*.

tChangeOffDelay: See *bChangeOffDelay*.

VAR_OUTPUT

```
uiErrorId                 : UINT;
iErrorParameter           : INT;
eExecutionState           : E_BAF_ExecutionState;
bRawValue                 : BOOL;
bValue                    : BOOL;
udiCounter                : UDINT;
bRisingEdge               : BOOL;
bFallingEdge              : BOOL;
bEdgeChanged              : BOOL;
bClick                    : BOOL;
bLongClick                : BOOL;
bDoubleClick              : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the digital signal.

bRawValue: The raw value of the signal.

bValue: The output-state of the signal.

udiCounter: The counter value of the signal.

bRisingEdge: If the output *bValue* becomes true, this output is for one PLC cycle true.

bFallingEdge: If the output *bValue* becomes false, this output is for one PLC cycle true.

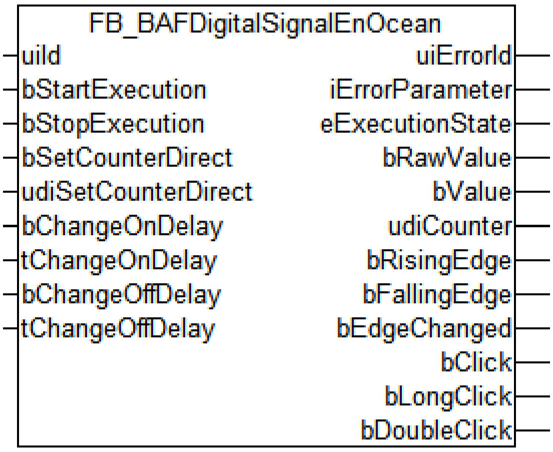
bEdgeChanged: If the output *bValue* changes, this output is for one PLC cycle true.

bClick: If a click (not a double click and not a long click) is detected, this output is for one PLC cycle true.

bLongClick: If a long click is detected, this output is for one PLC cycle true.

bDoubleClick: If a double click is detected, this output is for one PLC cycle true.

8.4.1.2 FB_BAFDigitalSignalEnOcean



Function-block for controlling an EnOcean Digital Signal [▶ 63] directly out of the PLC-program.

VAR_INPUT

```
uiId                      : UINT;
bStartExecution           : BOOL;
bStopExecution            : BOOL;
bSetCounterDirect         : BOOL;
udiSetCounterDirect       : UDINT;
bChangeOnDelay            : BOOL;
tChangeOnDelay            : TIME;
bChangeOffDelay           : BOOL;
tChangeOffDelay           : TIME;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the digital signal.

bStopExecution: Stops the instance of the digital signal.

bSetCounterDirect: Sets the counter to the value of the input *udiSetCounterDirect*.

udiSetCounterDirect: See *bSetCounterDirect*.

bChangeOnDelay: Sets the time delay-on to the value of the input *tChangeOnDelay*.

tChangeOnDelay: See *bChangeOnDelay*.

bChangeOffDelay: Sets the time delay-off to the value of the input *tChangeOffDelay*.

tChangeOffDelay: See *bChangeOffDelay*.

VAR_OUTPUT

```
uiErrorId                  : UINT;
iErrorParameter            : INT;
eExecutionState            : E_BAF_ExecutionState;
bRawValue                  : BOOL;
bValue                     : BOOL;
udiCounter                 : UDINT;
bRisingEdge                : BOOL;
bFallingEdge               : BOOL;
bEdgeChanged               : BOOL;
bClick                     : BOOL;
bLongClick                 : BOOL;
bDoubleClick               : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the digital signal.

bRawValue: The raw value of the signal.

bValue: The output-state of the signal.

udiCounter: the counter value of the signal.

bRisingEdge: If the output *bValue* becomes true, this output is for one PLC cycle true.

bFallingEdge: If the output *bValue* becomes false, this output is for one PLC cycle true.

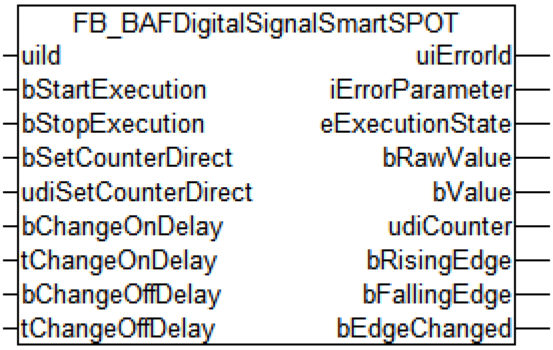
bEdgeChanged: If the output *bValue* changes, this output is for one PLC cycle true.

bClick: If a click (not a double click and not a long click) is detected, this output is for one PLC cycle true.

bLongClick: If a long click is detected, this output is for one PLC cycle true.

bDoubleClick: If a double click is detected, this output is for one PLC cycle true.

8.4.1.3 FB_BAFDigitalSignalSmartSPOT



Function-block for controlling an DALI smartSPOT Digital Signal [▶ 66] directly out of the PLC-program.

VAR_INPUT

```
uiId                   : UINT;
bStartExecution        : BOOL;
bStopExecution         : BOOL;
bSetCounterDirect      : BOOL;
udiSetCounterDirect    : UDINT;
bChangeOnDelay         : BOOL;
tChangeOnDelay         : TIME;
bChangeOffDelay        : BOOL;
tChangeOffDelay        : TIME;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the digital signal.

bStopExecution: Stops the instance of the digital signal.

bSetCounterDirect: Sets the counter to the value of the input *udiSetCounterDirect*.

udiSetCounterDirect: See *bSetCounterDirect*.

bChangeOnDelay: Sets the time delay-on to the value of the input *tChangeOnDelay*.

tChangeOnDelay: See *bChangeOnDelay*.

bChangeOffDelay: Sets the time delay-off to the value of the input *tChangeOffDelay*.

tChangeOffDelay: See *bChangeOffDelay*.

VAR_OUTPUT

```
uiErrorId                : UINT;
iErrorParameter          : INT;
eExecutionState          : E_BAF_ExecutionState;
bRawValue                : BOOL;
bValue                   : BOOL;
udiCounter               : UDINT;
bRisingEdge              : BOOL;
bFallingEdge             : BOOL;
bEdgeChanged             : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the digital signal.

bRawValue: The raw value of the signal.

bValue: The output-state of the signal.

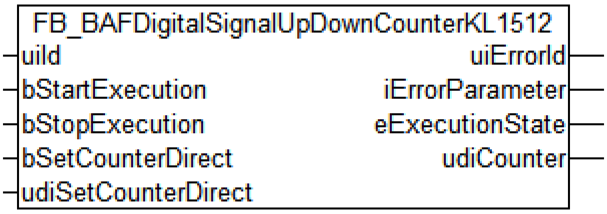
udiCounter: the counter value of the signal.

bRisingEdge: If the output *bValue* becomes true, this output is for one PLC cycle true.

bFallingEdge: If the output *bValue* becomes false, this output is for one PLC cycle true.

bEdgeChanged: If the output *bValue* changes, this output is for one PLC cycle true.

8.4.1.4 FB_BAFDigitalSignalCounterKL1512



Function-block for controlling a Up/Down counter via KL1512 Digital Signal [▶ 69] directly out of the PLC-program.

VAR_INPUT

```
uiId                     : UINT;
bStartExecution          : BOOL;
bStopExecution           : BOOL;
bSetCounterDirect        : BOOL;
udiSetCounterDirect      : UDINT;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the digital signal.

bStopExecution: Stops the instance of the digital signal.

bSetCounterDirect: Sets the counter to the value of the input *udiSetCounterDirect*.

udiSetCounterDirect: See *bSetCounterDirect*.

VAR_OUTPUT

```
uiErrorId              : UINT;
iErrorParameter        : INT;
eExecutionState        : E_BAF_ExecutionState;
udiCounter             : UDINT;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

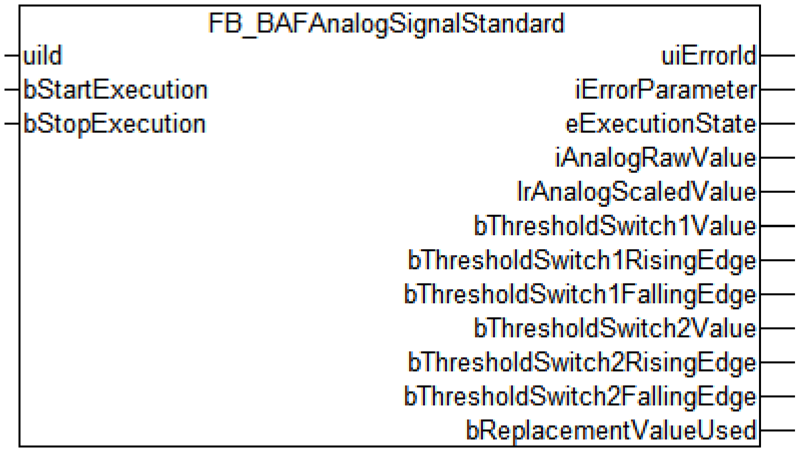
iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the digital signal.

udiCounter: The counter value of the signal.

8.4.2 Analog Signals

8.4.2.1 FB_BAFAnalogSignalStandard



Function-block for controlling a Standard Analog Signal [▶ 70] directly out of the PLC-program.

VAR_INPUT

```
uiId                   : UINT;
bStartExecution        : BOOL;
bStopExecution         : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the analog signal.

bStopExecution: Stops the instance of the analog signal.

VAR_OUTPUT

```
uiErrorId                    : UINT;
iErrorParameter              : INT;
eExecutionState              : E_BAF_ExecutionState;
iAnalogRawValue              : INT;
lrAnalogScaledValue          : LREAL;
bThresholdSwitch1Value       : BOOL;
bThresholdSwitch1RisingEdge  : BOOL;
bThresholdSwitch1FallingEdge : BOOL;
bThresholdSwitch2Value       : BOOL;
```

```
bThresholdSwitch2RisingEdge  : BOOL;
bThresholdSwitch2FallingEdge : BOOL;
bReplacementValueUsed        : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the analog signal.

iAnalogRawValue : The raw-value of the analog signal.

lrAnalogScaledValue: The scaled value of the analog signal.

bThresholdSwitch1Value: The current value of the threshold switch 1.

bThresholdSwitch1RisingEdge: If the threshold switch 1 becomes true, this output is for one PLC cycle true.

bThresholdSwitch1FallingEdge: If the threshold switch 1 becomes false, this output is for one PLC cycle true.

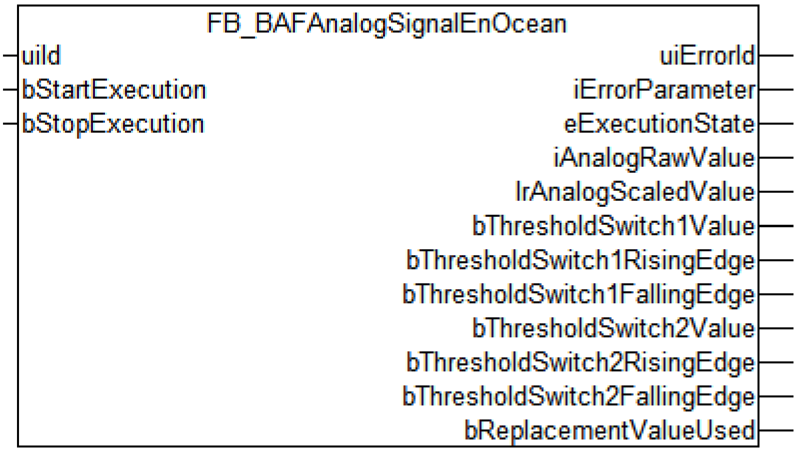
bThresholdSwitch2Value: The current value of the threshold switch 2.

bThresholdSwitch2RisingEdge: If the threshold switch 2 becomes true, this output is for one PLC cycle true.

bThresholdSwitch2FallingEdge: If the threshold switch 2 becomes false, this output is for one PLC cycle true.

bReplacementValueUsed: Indicates if the replacement value is used.

8.4.2.2 FB_BAFAnalogSignalEnOcean



Function-block for controlling an EnOcean Analog Signal [▶ 73] directly out of the PLC-program.

VAR_INPUT

```
uiId                : UINT;
bStartExecution     : BOOL;
bStopExecution      : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the analog signal.

bStopExecution: Stops the instance of the analog signal.

VAR_OUTPUT

```
uiErrorId                    : UINT;
iErrorParameter              : INT;
eExecutionState              : E_BAF_ExecutionState;
iAnalogRawValue              : INT;
```

```
lrAnalogScaledValue          : LREAL;
bThresholdSwitch1Value       : BOOL;
bThresholdSwitch1RisingEdge  : BOOL;
bThresholdSwitch1FallingEdge : BOOL;
bThresholdSwitch2Value       : BOOL;
bThresholdSwitch2RisingEdge  : BOOL;
bThresholdSwitch2FallingEdge : BOOL;
bReplacementValueUsed        : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the analog signal.

iAnalogRawValue: The raw-value of the analog signal.

lrAnalogScaledValue: The scaled value of the analog signal.

bThresholdSwitch1Value: The current value of the threshold switch 1.

bThresholdSwitch1RisingEdge: If the threshold switch 1 becomes true, this output is for one PLC cycle true.

bThresholdSwitch1FallingEdge: If the threshold switch 1 becomes false, this output is for one PLC cycle true.

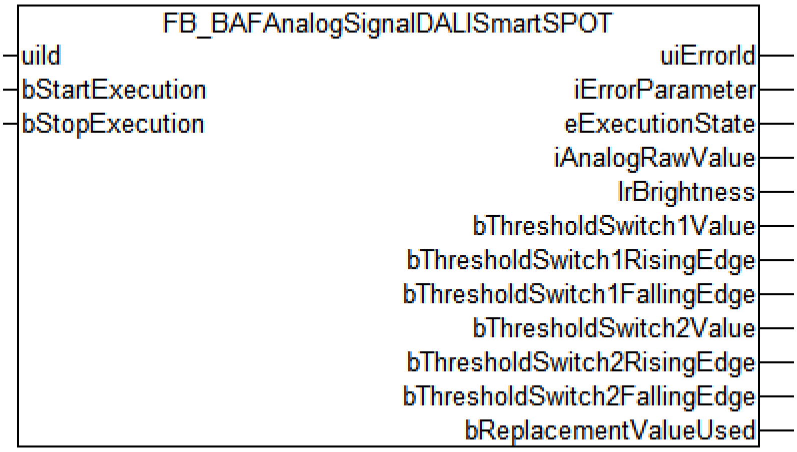
bThresholdSwitch2Value: The current value of the threshold switch 2.

bThresholdSwitch2RisingEdge: If the threshold switch 2 becomes true, this output is for one PLC cycle true.

bThresholdSwitch2FallingEdge: If the threshold switch 2 becomes false, this output is for one PLC cycle true.

bReplacementValueUsed: Indicates if the replacement value is used.

8.4.2.3 FB_BAFAnalogSignalDALISmartSPOT



Function-block for controlling a DALI smartSPOT [▶ 76] directly out of the PLC-program.

VAR_INPUT

```
uiId                  : UINT;
bStartExecution       : BOOL;
bStopExecution        : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the analog signal.

bStopExecution: Stops the instance of the analog signal.

VAR_OUTPUT

```
uiErrorId                    : UINT;
iErrorParameter              : INT;
eExecutionState              : E_BAF_ExecutionState;
iAnalogRawValue              : INT;
lrBrightness                 : LREAL;
bThresholdSwitch1Value       : BOOL;
bThresholdSwitch1RisingEdge  : BOOL;
bThresholdSwitch1FallingEdge : BOOL;
bThresholdSwitch2Value       : BOOL;
bThresholdSwitch2RisingEdge  : BOOL;
bThresholdSwitch2FallingEdge : BOOL;
bReplacementValueUsed        : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the analog signal.

iAnalogRawValue: The raw value of the smartSPOT.

lrBrightness: The brightness in lux.

bThresholdSwitch1Value: The current value of the threshold switch 1.

bThresholdSwitch1RisingEdge: If the threshold switch 1 becomes true, this output is for one PLC cycle true.

bThresholdSwitch1FallingEdge: If the threshold switch 1 becomes false, this output is for one PLC cycle true.

bThresholdSwitch2Value: The current value of the threshold switch 2.

bThresholdSwitch2RisingEdge: If the threshold switch 2 becomes true, this output is for one PLC cycle true.

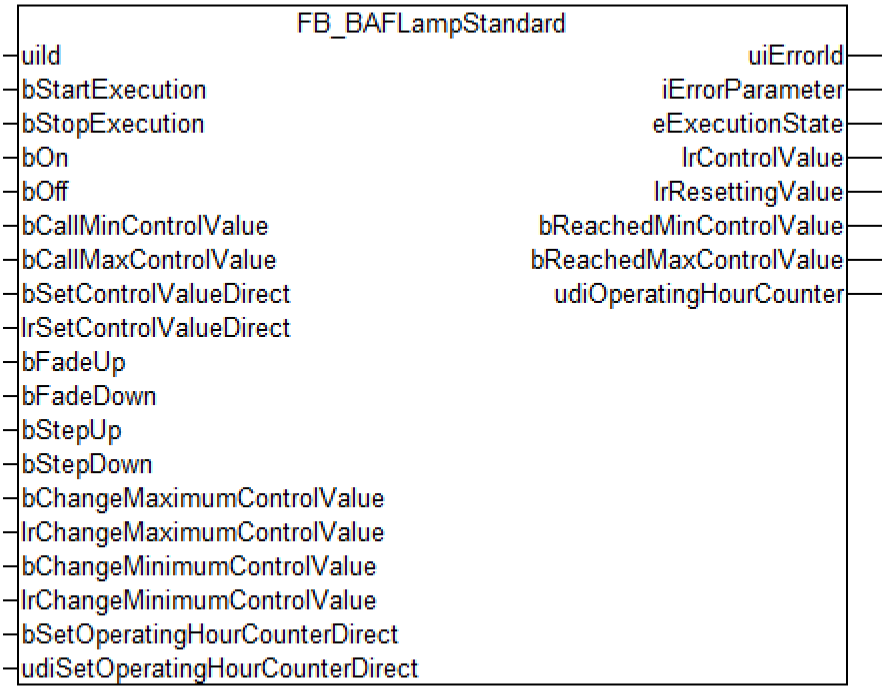
bThresholdSwitch2FallingEdge: If the threshold switch 2 becomes false, this output is for one PLC cycle true.

bReplacementValueUsed: Indicates if the replacement value is used.

8.5 Actuators

8.5.1 Lamps

8.5.1.1 FB_BAFLampStandard



Function-block for controlling a Standard Lamp [▶ 79] directly out of the PLC-program.

VAR_INPUT

```
uiId                             : UINT;
bStartExecution                  : BOOL;
bStopExecution                   : BOOL;
bOn                              : BOOL;
bOff                             : BOOL;
bCallMinControlValue             : BOOL;
bCallMaxControlValue             : BOOL;
bSetControlValueDirect           : BOOL;
lrSetControlValueDirect          : LREAL;
bFadeUp                          : BOOL;
bFadeDown                        : BOOL;
bStepUp                          : BOOL;
bStepDown                        : BOOL;
bChangeMaximumControlValue       : BOOL;
lrChangeMaximumControlValue      : LREAL
bChangeMinimumControlValue       : BOOL;
lrChangeMinimumControlValue      : LREAL;
bSetOperatingHourCounterDirect   : BOOL;
udiSetOperatingHourCounterDirect : UDINT;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the lamp.

bStopExecution: Stops the instance of the lamp.

bOn: If *iResettingMode* = 1: The lamp will be set to the previously stored *lrResettingValue*.
If *iResettingMode* = 0: The lamp will be set to *lrMaxControlValue.*

bOff: If *iResettingMode* = 1: The lamp will be turned off. Before, the control value will be stored in *lrResettingValue*.
If *iResettingMode* = 0: The lamp will be turned off without saving the last control value.

bCallMinControlValue: The control value of the lamp will be set to *lrMinControlValue*.

bCallMaxControlValue: The control value of the lamp will be set to *lrMaxControlValue*.

bSetControlValueDirect: The control value of the lamp will be set to *lrSetControlValueDirect*.

lrSetControlValueDirect: The value (0% - 100%) to which the lamp is set after the command *bSetControlValueDirect* was given. Sending the value "0" to the lamp will turn the device directly off. All other given control-values below *lrMinControlValue* will be automatically increased to *lrMinControlValue*, values above *lrMaxControlValue* will automatically decreased to *lrMaxControlValue*.

bFadeUp: Lets the control value be increased by *lrFadeSteps*. If the lamp is turned off, this command will not be executed.

bFadeDown: Lets the control value be decreased by *lrFadeSteps*. It is not possible to turn off the lamp with this command.

bStepUp: Steps the lamp up by the entered step range.

bStepDown: Steps the lamp down by the entered step range.

bChangeMaximumControlValue: The maximum control value of the lamp will be set to *lrChangeMaximumControlValue*.

lrChangeMaximumControlValue: Level (0% - 100%) to which the lamps maximum control level will be set, depending on the state of *bChangeMaximumControlValue*.

bChangeMinimumControlValue: The minimum control value of the lamp will be set to *lrChangeMinimumControlValue*.

lrChangeMaxMinControlValue: Level (0% - 100%) to which the lamps minimum control level will be set, depending on the state of *bChangeMinimumControlValue*.

bSetOperatingHourCounterDirect: The operating hours counter for the lamp is set to *udiSetOperatingHourCounterDirect*.

udiSetOperatingHourCounterDirect: The value to which the operating hours counter for the lamp is set by the input *bSetOperatingHourCounterDirect* set. Unit is seconds.

VAR_OUTPUT

```
uiErrorId                  : UINT;
iErrorParameter            : INT;
eExecutionState            : E_BAF_ExecutionState;
lrControlValue             : LREAL;
lrResettingValue           : LREAL;
bReachedMinControlValue    : BOOL;
bReachedMaxControlValue    : BOOL;
udiOperatingHourCounter    : UDINT;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution state of the instance of the lamp.

lrControlValue: The control-value (0% - 100%) of the lamp.

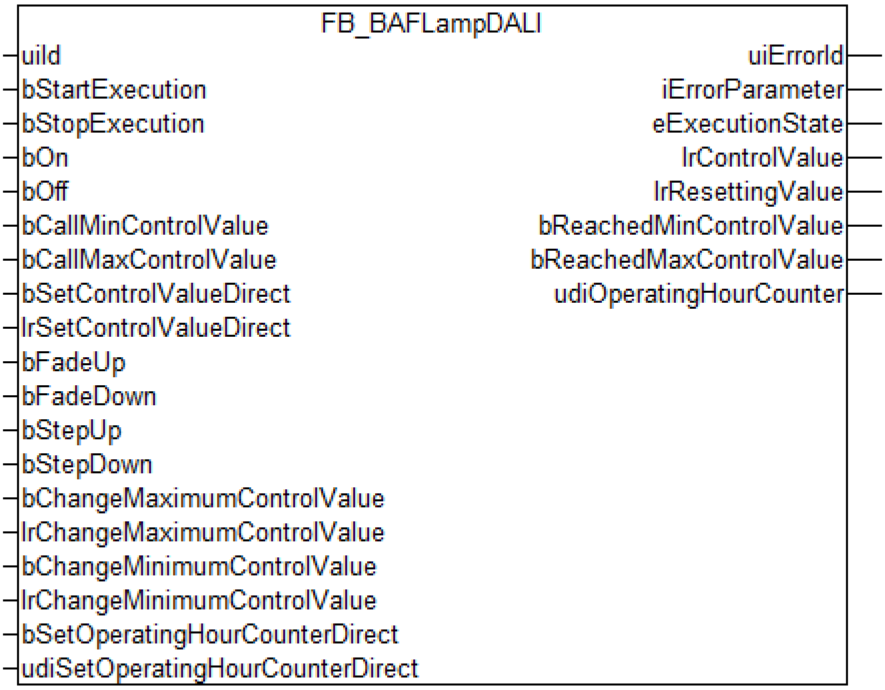
iResettingValue: The resetting value (0% - 100%) of the lamp.

bReachedMinControlValue: The control value of the lamp *lrControlValue* is equal to the parameter *lrMinControlValue*.

bReachedMaxControlValue: The control value of the lamp *lrControlValue* is equal to the parameter *lrMaxControlValue*.

udiOperatingHourCounter: operating hours counter for the lamp. Unit is seconds.

8.5.1.2 FB_BAFLampDALI



Function-block for controlling a DALI Lamp [▸ 82] directly out of the PLC-program.

VAR_INPUT

```
uiId                             : UINT;
bStartExecution                  : BOOL;
bStopExecution                   : BOOL;
bOn                              : BOOL;
bOff                             : BOOL;
bCallMinControlValue             : BOOL;
bCallMaxControlValue             : BOOL;
bSetControlValueDirect           : BOOL;
lrSetControlValueDirect          : LREAL;
bFadeUp                          : BOOL;
bFadeDown                        : BOOL;
bChangeMaximumControlValue       : BOOL;
lrChangeMaximumControlValue      : LREAL
bChangeMinimumControlValue       : BOOL;
lrChangeMinimumControlValue      : LREAL;
bSetOperatingHourCounterDirect   : BOOL;
udiSetOperatingHourCounterDirect : UDINT;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the lamp.

bStopExecution: Stops the instance of the lamp.

bOn: If *iResettingMode* = 1: The control value will be set to the previously stored *lrResettingValue*. Then this value will be internally adjusted to the DALI-range (0..254) and sent to the lamp with the DALI-command DIRECT ARC POWER.
If *iResettingMode* = 0: The DALI-Command RECALL MAX LEVEL will be sent.

bOff: If *iResettingMode* = 1: The lamp will be turned off by sending the DALI-command OFF. Before, the control value will be stored in *lrResettingValue*.
If *iResettingMode* = 0: The lamp will be turned off without saving the last control-value.

bCallMinControlValue: The DALI-command RECALL MIN LEVEL will be sent.

bCallMaxControlValue: The DALI-command RECALL MAX LEVEL will be sent.

bSetControlValueDirect: The control-value of the lamp will be set to *lrSetControlValueDirect*.

lrSetControlValueDirect: If *bSetControlValueDirect* is set, the value (0% - 100%) will be internally adjusted to the DALI-Range (0 - 254) and then sent to the lamp with the DALI-command DIRECT ARC POWER.

bFadeUp: The DALI-command UP will be sent.

bFadeDown: The DALI-command DOWN will be sent.

bChangeMaximumControlValue: The maximum control value of the lamp will be set to *lrChangeMaximumControlValue*.

lrChangeMaximumControlValue: Level (0% - 100%) to which the lamps maximum control level will be set, depending on the state of *bChangeMaximumControlValue*.

bChangeMinimumControlValue: The minimum control value of the lamp will be set to *lrChangeMinimumControlValue*.

lrChangeMaxMinControlValue: Level (0% - 100%) to which the lamps minimum control level will be set, depending on the state of *bChangeMinimumControlValue*.

bSetOperatingHourCounterDirect: The operating hours counter for the lamp is set to *udiSetOperatingHourCounterDirect*.

udiSetOperatingHourCounterDirect: The value to which the operating hours counter for the lamp is set by the input *bSetOperatingHourCounterDirect* set. Unit is seconds.

VAR_OUTPUT

```
uiErrorId                 : UINT;
iErrorParameter           : INT;
eExecutionState           : E_BAF_ExecutionState;
lrControlValue            : LREAL;
lrResettingValue          : LREAL;
bReachedMinControlValue   : BOOL;
bReachedMaxControlValue   : BOOL;
udiOperatingHourCounter   : UDINT;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the lamp.

lrControlValue: The control value (0% - 100%) of the lamp.

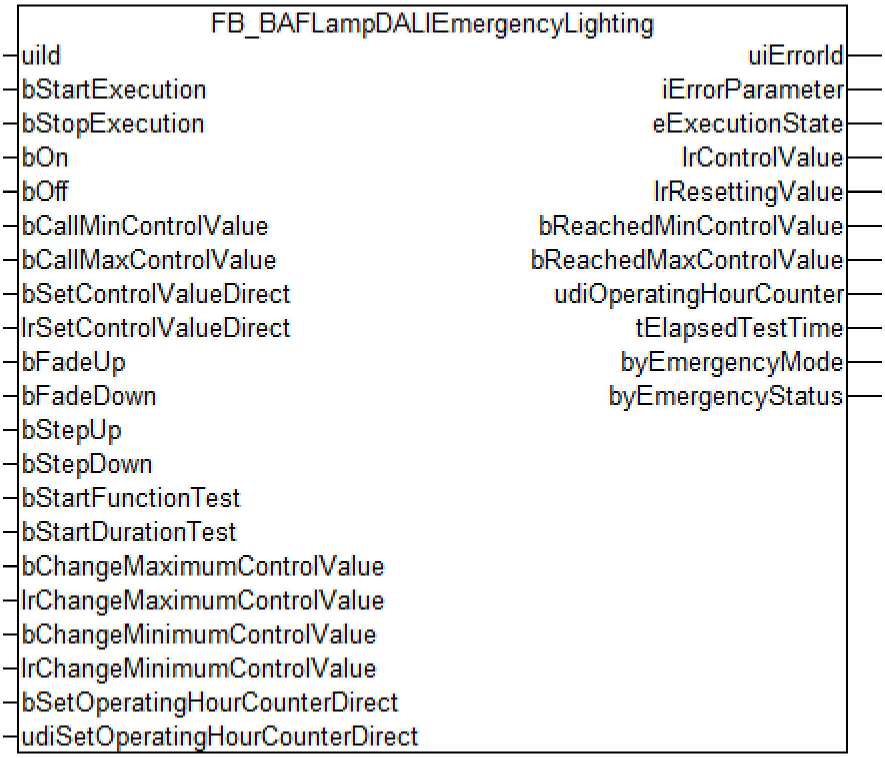
iResettingValue: The resetting value (0% - 100%) of the lamp.

bReachedMinControlValue: The control value of the lamp *lrControlValue* is equal to the parameter *lrMinControlValue*.

bReachedMaxControlValue: The control value of the lamp *lrControlValue* is equal to the parameter *lrMaxControlValue*.

udiOperatingHourCounter: operating hours counter for the lamp. Unit is seconds.

8.5.1.3 FB_BAFLampDALIEmergencyLighting



Function-block for controlling a DALI Lamp for Emergency Lighting [▸ 84] directly out of the PLC-program.

VAR_INPUT

```
uiId                             : UINT;
bStartExecution                  : BOOL;
bStopExecution                   : BOOL;
bOn                              : BOOL;
bOff                             : BOOL;
bCallMinControlValue             : BOOL;
bCallMaxControlValue             : BOOL;
bSetControlValueDirect           : BOOL;
lrSetControlValueDirect          : LREAL;
bFadeUp                          : BOOL;
bFadeDown                        : BOOL;
bStepUp                          : BOOL;
bStepDown                        : BOOL;
bStartFunctionTest               : BOOL;
bStartDurationTest               : BOOL;
bChangeMaximumControlValue       : BOOL;
lrChangeMaximumControlValue      : LREAL
bChangeMinimumControlValue       : BOOL;
lrChangeMinimumControlValue      : LREAL;
bSetOperatingHourCounterDirect   : BOOL;
udiSetOperatingHourCounterDirect : UDINT;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the lamp.

bStopExecution: Stops the instance of the lamp.

bOn: If *iResettingMode* = 1: The control-value will be set to the previously stored *lrResettingValue*. Then this value will be internally adjusted to the DALI-Range (0 - 254) and sent to the lamp with the DALI-command DIRECT ARC POWER.
If *iResettingMode* = 0: The DALI-Command RECALL MAX LEVEL will be sent.

bOff: If *iResettingMode* = 1: The lamp will be turned off by sending the DALI-command OFF. Before, the control value will be stored in *lrResettingValue*.
If *iResettingMode* = 0: The lamp will be turned off without saving the last control value.

bCallMinControlValue: The DALI-Command RECALL MIN LEVEL will be sent.

bCallMaxControlValue: The DALI-Command RECALL MAX LEVEL will be sent.

bSetControlValueDirect: The control value of the lamp will be set to *lrSetControlValueDirect*.

lrSetControlValueDirect: If bSetControlValueDirect is set, the value (0% - 100%) will be internally adjusted to the DALI-Range (0 - 254) and then sent to the lamp with the DALI-command DIRECT ARC POWER.

bFadeUp: Increases the light value by the entered fade-rate.

bFadeDown: Decreases the light value by the entered fade-rate.

bStepUp: Steps the lamp up by the entered step range.

bStepDown: Steps the lamp down by the entered step range.

bStartFunctionTest: A function test will be started.

bStartDurationTest: A duration test will be started.

bChangeMaximumControlValue: The maximum control-value of the lamp will be set to *lrChangeMaxMinControlValue*.

bChangeMinimumControlValue: The minimum control-value of the lamp will be set to *lrChangeMaxMinControlValue*.

lrChangeMaxMinControlValue: Level (0% - 100%) to which the lamps minimum- or maximum control level will be set, depending on the states of *bChangeMaximumControlValue* and *bChangeMinimumControlValue*.

bSetOperatingHourCounterDirect: The operating hours counter for the lamp is set to *udiSetOperatingHourCounterDirect*.

udiSetOperatingHourCounterDirect: The value to which the operating hours counter for the lamp is set by the input *bSetOperatingHourCounterDirect* set. Unit is seconds.

VAR_OUTPUT

```
uiErrorId                 : UINT;
iErrorParameter           : INT;
eExecutionState           : E_BAF_ExecutionState;
lrControlValue            : LREAL;
lrResettingValue          : LREAL;
bReachedMinControlValue   : BOOL;
bReachedMaxControlValue   : BOOL;
udiOperatingHourCounter   : UDINT;
tElapsedTestTime          : TIME;
byEmergencyMode           : BYTE;
byEmergencyStatus         : BYTE;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution state of the instance of the lamp.

lrControlValue: The control-value (0% - 100%) of the lamp.

iResettingValue: The resetting-value (0% - 100%) of the lamp.

bReachedMinControlValue: The control value of the lamp *lrControlValue* is equal to the parameter *lrMinControlValue*.

bReachedMaxControlValue: The control value of the lamp *lrControlValue* is equal to the parameter *lrMaxControlValue*.

udiOperatingHourCounter: operating hours counter for the lamp. Unit is seconds.

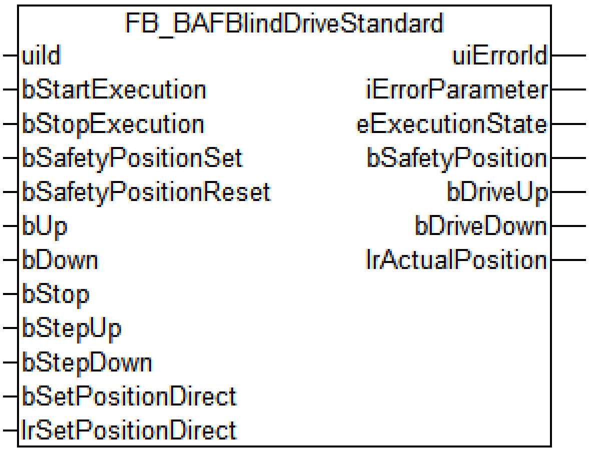
tElapsedTestTime: If a test is actually running, this output will show the elapsed test-time. This output will maintain its contents after the test is finished.

byEmergencyMode: Shows the Emergency-mode during a test. If not in test-mode, this DALI-specific state-byte will not be automatically updated.

byEmergencyState: Shows the Emergency-state during a test. If not in test-mode, this DALI-specific state-byte will not be automatically updated.

8.5.2 Blind Drives

8.5.2.1 FB_BAFBlindDriveStandard



Function-block for controlling a Standard Blind Drive [▶ 86] directly out of the PLC-program.

VAR_INPUT

```
uiId                 : UINT;
bStartExecution      : BOOL;
bStopExecution       : BOOL;
bSafetyPositionSet   : BOOL;
bSafetyPositionReset : BOOL;
bUp                  : BOOL;
bDown                : BOOL;
bStop                : BOOL;
bStepUp              : BOOL;
bStepDown            : BOOL;
bSetPositionDirect   : BOOL;
lrSetPositionDirect  : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the blind drive.

bStopExecution: Stops the instance of the blind drive.

bSafetyPositionSet: Driving to safety-position is activated. For this purpose the blind drive will be moved up for *tDriveTime* + 10%. No other commands except *bSafetyPositionReset* are accepted then.

bSafetyPositionReset: Releases the safety-position.

bUp: The blind will be driven upwards.

bDown: The blind will be driven downwards.

bStop: Stops the blind.

bStepUp: The blind will be driven upwards for *tStepTime*.

bStepDown: The blind will be driven downwards for *tStepTime*.

bSetPositionDirect: The blind drive will be moved to the position given by *lrSetPositionDirect*.

lrSetPositionDirect: See *bSetPositionDirect*. 0% means completely opened, 100% completely closed.

VAR_OUTPUT

```
uiErrorId            : UINT;
iErrorParameter      : INT;
eExecutionState      : E_BAF_ExecutionState;
bSafetyPosition      : BOOL;
bDriveUp             : BOOL;
bDriveDown           : BOOL;
lrActualPosition     : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the blind drive.

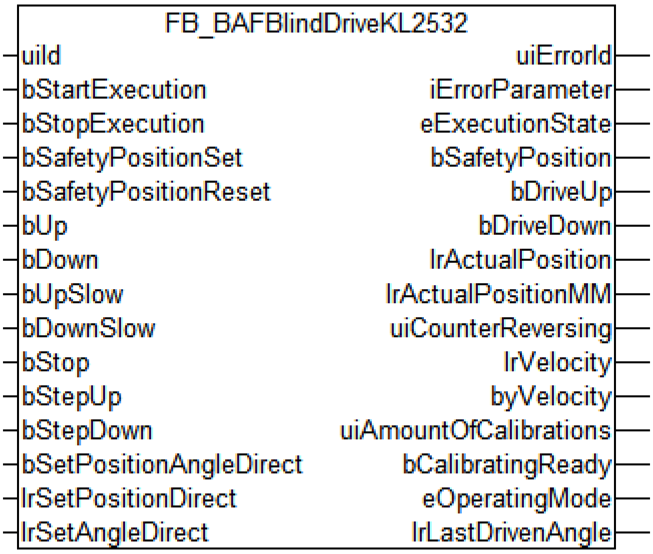
bSafetyPosition : The safety-position-driving has been activated.

bDriveUp: The blind is moving upwards.

bDriveDown: The blind is moving downwards.

lrActualPosition: The position (0% - 100%) of this blind drive.

8.5.2.2 FB_BAFBlindDriveKL2532



Function-block for controlling a Blind Drive connected with KL2532 [▶ 88] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
bSafetyPositionSet      : BOOL;
bSafetyPositionReset    : BOOL;
bUp                     : BOOL;
bDown                   : BOOL;
bUpSlow                 : BOOL;
bDownSlow               : BOOL;
bStop                   : BOOL;
bStepUp                 : BOOL;
bStepDown               : BOOL;
bSetPositionAngleDirect : BOOL;
lrSetPositionDirect     : LREAL;
lrSetAngleDirect        : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the blind drive.

bStopExecution: Stops the instance of the blind drive.

bSafetyPositionSet: Driving to safety-position is activated. For this purpose the blind drive will be moved up for *tDriveTime* + 10%. No other commands except *bSafetyPositionReset* are accepted then.

bSafetyPositionReset: Releases the safety-position.

bUp: The blind will be driven upwards.

bDown: The blind will be driven downwards.

bUpSlow: The blind will be slowly driven upwards.

bDownSlow: The blind will be slowly driven downwards.

bStop: Stops the blind.

bStepUp: The blind will be driven upwards for *tStepTime*.

bStepDown: The blind will be driven downwards for *tStepTime*.

bSetPositionAngleDirect: The blind drive will be moved to the angle entered under *lrSetAngleDirect*. As the BA-Framework-element *Standard Blind Drive* does not support angle-driving functionality - unlike *Blind Drive connected with KL2532* - this command will only have an effect on the position.

lrSetPositionDirect: 0% means completely opened, 100% completely closed.

lrSetAngleDirect: See *bSetPositionAngleDirect*.

VAR_OUTPUT

```
uiErrorId                : UINT;
iErrorParameter          : INT;
eExecutionState          : E_BAF_ExecutionState;
bSafetyPosition          : BOOL;
bDriveUp                 : BOOL;
bDriveDown               : BOOL;
lrActualPosition         : LREAL;
lrActualPositionMM       : LREAL;
uiCounterReversing       : UINT;
lrVelocity               : LREAL;
byVelocity               : BYTE;
uiAmountOfCalibrations   : UINT;
bCalibratingReady        : BOOL;
eOperatingMode           : E_BAF_SuntrackingOperatingMode;
lrLastDrivenAngle        : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution state of the instance of the blind drive.

bSafetyPosition: The safety-position-driving has been activated

bDriveUp: The blind is moving upwards.

bDriveDown: The blind is moving downwards.

lrActualPosition: The actual position in percentage.

lrActualPositionMM: The actual position in millimetres.

uiCounterReversing: The counter in reversing-mode.

lrVelocity: Actual speed in [mm/s].

byVelocity: Actual speed in [%].

uiAmountOfCalibrations: Number of calibrations.

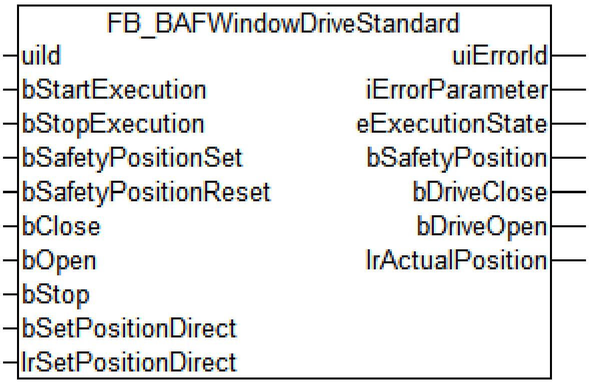
bCalibratingReady: Axis is calibrated.

eOperatingMode: Actual operating-mode.

lrLastDrivenAngle: The last driven angle.

8.5.3 Window Drives

8.5.3.1 FB_BAFWindowDriveStandard



Function-block for controlling a Standard Window Drive [▶ 98] directly out of the PLC-program.

VAR_INPUT

```
uiId                 : UINT;
bStartExecution      : BOOL;
bStopExecution       : BOOL;
bSafetyPositionSet   : BOOL;
bSafetyPositionReset : BOOL;
bClose               : BOOL;
bOpen                : BOOL;
bStop                : BOOL;
bSetPositionDirect   : BOOL;
lrSetPositionDirect  : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the window drive.

bStopExecution: Stops the instance of the window drive.

bSafetyPositionSet: Driving to safety-position is activated. For this purpose the window drive will be moved up for *tDriveTime* + 10%. No other commands except *bSafetyPositionReset* are accepted then.

bSafetyPositionReset: Releases the safety-position.

bClose: The window will be closed.

bOpen: The window will be opened.

bStop: Stops the window-drive.

bSetPositionDirect: The window drive will be moved to the position entered under *lrSetPositionDirect*.

lrSetPositionDirect: See *bSetPositionDirect*. 0% means completely closed, 100% completely opened.

VAR_OUTPUT

```
uiErrorId                 : UINT;
iErrorParameter           : INT;
eExecutionState           : E_BAF_ExecutionState;
```

```
bSafetyPosition          : BOOL;
bDriveClose              : BOOL;
bDriveOpen               : BOOL;
lrActualPosition         : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the window drive.

bSafetyPosition: The safety-position-driving has been activated.

bDriveClose: The window is closing.

bDriveOpen: The window is opening.

lrActualPosition: The position (0% - 100%) of this window drive.

8.5.4 Valve Actuators

8.5.4.1 FB_BAFValveActuator2Point

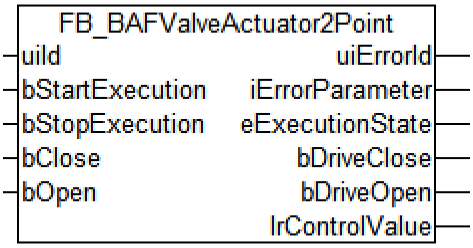


Fig. 2: TcBAPlc_FB_BAFValveActuator2Point

Function-block for controlling a 2-point Valve Actuator [▶ 94] directly out of the PLC-program.

VAR_INPUT

```
uiId                  : UINT;
bStartExecution       : BOOL;
bStopExecution        : BOOL;
bClose                : BOOL;
bOpen                 : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the valve actuator.

bStopExecution: Stops the instance of the valve actuator.

bClose: Closes the valve.

bOpen: Opens the valve.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
bDriveClose           : BOOL;
bDriveOpen            : BOOL;
lrControlValue        : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the valve.

bDriveClose: The valve closes.

bDriveOpen: The valve opens.

lrControlValue: Control value between 0% and 100%.

8.5.4.2 FB_BAFValveActuator3Point

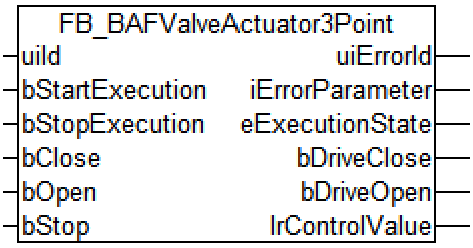


Fig. 3: TcBAPlc_FB_BAFValveActuator3Point

Function-block for controlling a 3-point Valve Actuator [▸ 95] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
bClose                  : BOOL;
bOpen                   : BOOL;
bStop                   : BOOL;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the valve actuator.

bStopExecution: Stops the instance of the valve actuator.

bClose: Closes the valve.

bOpen: Opens the valve.

bStop: Stops opening/closing the valve.

VAR_OUTPUT

```
uiErrorId               : UINT;
iErrorParameter         : INT;
eExecutionState         : E_BAF_ExecutionState;
bDriveClose             : BOOL;
bDriveOpen              : BOOL;
lrControlValue          : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution state of the instance of the valve.

bDriveClose: The valve closes.

bDriveOpen: The valve opens.

lrControlValue: Control value between 0% and 100%.

8.5.4.3 FB_BAFValveActuatorContinuous

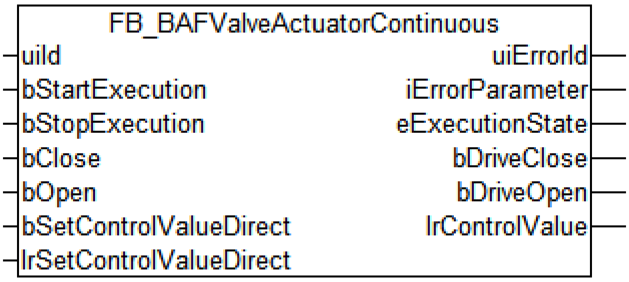


Fig. 4: TcBAPlc_FB_BAFValveActuatorContinuous

Function-block for controlling a Continuous Valve Actuator [▶ 97] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
bClose                  : BOOL;
bOpen                   : BOOL;
bSetControlValueDirect  : BOOL;
lrSetControlValueDirect : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the valve actuator.

bStopExecution: Stops the instance of the valve actuator.

bSetPositionDirect: The valve will open up to the position entered under *lrSetPositionDirect*.

lrSetPositionDirect: See *bSetPositionDirect*. 0% means completely closed, 100% fully opened.

VAR_OUTPUT

```
uiErrorId           : UINT;
iErrorParameter     : INT;
eExecutionState     : E_BAF_ExecutionState;
lrControlValue      : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

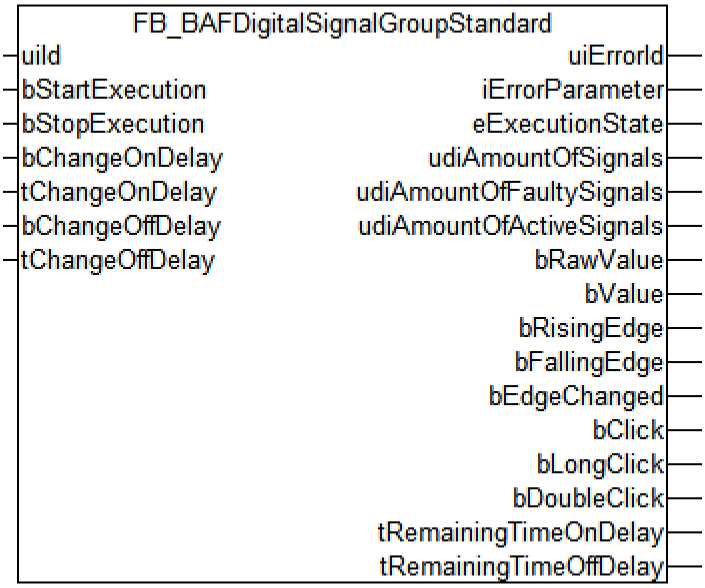
eExecutionState: Contains the actual execution state of the instance of the valve.

lrControlValue: Control value between 0% and 100%.

8.6 Sensor Groups

8.6.1 Digital Signal Groups

8.6.1.1 FB_BAFDigitalSignalGroupStandard



Function-block for controlling a Standard Digital Signal Group [▸ 100] directly out of the PLC-program.

VAR_INPUT

```
uiId                 : UINT;
bStartExecution      : BOOL;
bStopExecution       : BOOL;
bChangeOnDelay       : BOOL;
tChangeOnDelay       : TIME;
bChangeOffDelay      : BOOL;
tChangeOffDelay      : TIME;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the digital signal group.

bStopExecution: Stops the instance of the digital signal group.

bChangeOnDelay: Sets the time delay-on to the value of the input *tChangeOnDelay*.

tChangeOnDelay: See *bChangeOnDelay*.

bChangeOffDelay: Sets the time delay-off to the value of the input *tChangeOffDelay*.

tChangeOffDelay: See *bChangeOffDelay*.

VAR_OUTPUT

```
uiErrorId                : UINT;
iErrorParameter          : INT;
eExecutionState          : E_BAF_ExecutionState;
udiAmountOfSignals       : UDINT;
udiAmountOfFaultySignals : UDINT;
udiAmountOfActiveSignals : UDINT;
bRawValue                : BOOL;
bValue                   : BOOL;
bRisingEdge              : BOOL;
bFallingEdge             : BOOL;
bEdgeChanged             : BOOL;
bClick                   : BOOL;
bLongClick               : BOOL;
```

```
bDoubleClick              : BOOL;
tRemainingTimeOnDelay     : TIME;
tRemainingTimeOffDelay    : TIME;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the digital signal group.

udiAmountOfSignals: The amount of all digital signals in this group.

udiAmountOfFaultySignals: The amount of all faulty signals in this group (*uiErrorId* > 0).

udiAmountOfActiveSignals : The amount of all active signals in this group.

bRawValue: The *bValue*-output of a digital signal can be delayed. This output however shows the result of the OR-connection of the non-delayed signal-states of the digital signals in this group.

bValue : This signal is the result of an OR-connection of all *bValue*-outputs of the digital signals in this group.

bRisingEdge : The rising trigger of *bValue* of this group.

bFallingEdge : The falling trigger of *bValue* of this group.

bEdgeChanged : The edge-change of *bValue* of this group.

bClick : A short-click in this group was detected.

bLongClick : A long-click in this group was detected.

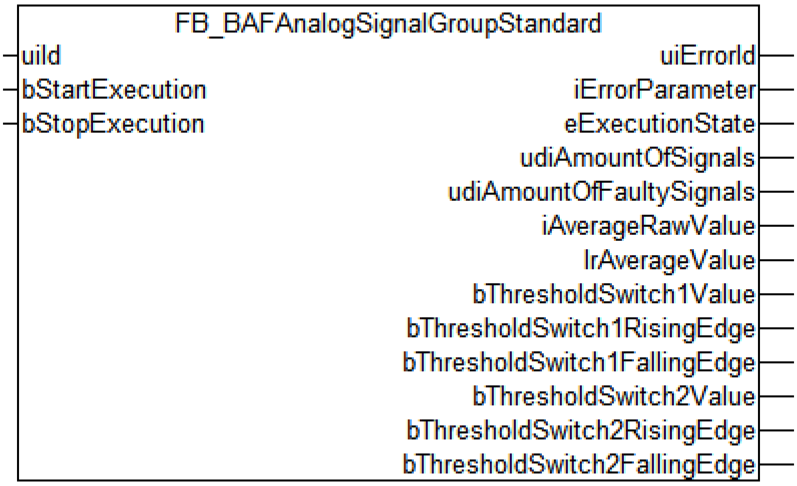
bDoubleClick : A double-click in this group was detected.

tRemainingTimeOnDelay: Shows the remaining time of the on-delay-timer.

tRemainingTimeOffDelay: Shows the remaining time of the off-delay-timer.

8.6.2 Analog Signal Groups

8.6.2.1 FB_BAFAnalogSignalGroupStandard



Function-block for controlling a Standard Analog Signal Group [▶ 102] directly out of the PLC-program.

VAR_INPUT

```
uiId                  : UINT;
bStartExecution       : BOOL;
bStopExecution        : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the analog signal group.

bStopExecution: Stops the instance of the analog signal group.

VAR_OUTPUT

```
uiErrorId                    : UINT;
iErrorParameter              : INT;
eExecutionState              : E_BAF_ExecutionState;
udiAmountOfSignals           : UDINT;
udiAmountOfFaultySignals     : UDINT;
iAverageRawValue             : INT;
lrAverageValue               : LREAL;
bThresholdSwitch1Value       : BOOL;
bThresholdSwitch1RisingEdge  : BOOL;
bThresholdSwitch1FallingEdge : BOOL;
bThresholdSwitch2Value       : BOOL;
bThresholdSwitch2RisingEdge  : BOOL;
bThresholdSwitch2FallingEdge : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [⏵ 276]).

iErrorParameter : Contains the specific parameter of the error code (see Error codes [⏵ 276]).

eExecutionState : Contains the actual execution state of the instance of the analog signal group.

udiAmountOfSignals : The amount of all analog signals in this group.

udiAmountOfFaultySignals : The amount of all faulty signals in this group (*uiErrorId* > 0).

iAverageRawValue: The average value of raw values of all analog signals in this group.

lrAverageValue : The average value of all analog signals in this group.

bThresholdSwitch1Value: Threshold-level 1 is reached.

bThresholdSwitch1RisingEdge: Threshold-level 1 is reached - Rising edge.

bThresholdSwitch1FallingEdge: Threshold-level 1 is reached - Falling edge.

bThresholdSwitch2Value: Threshold-level 2 is reached.

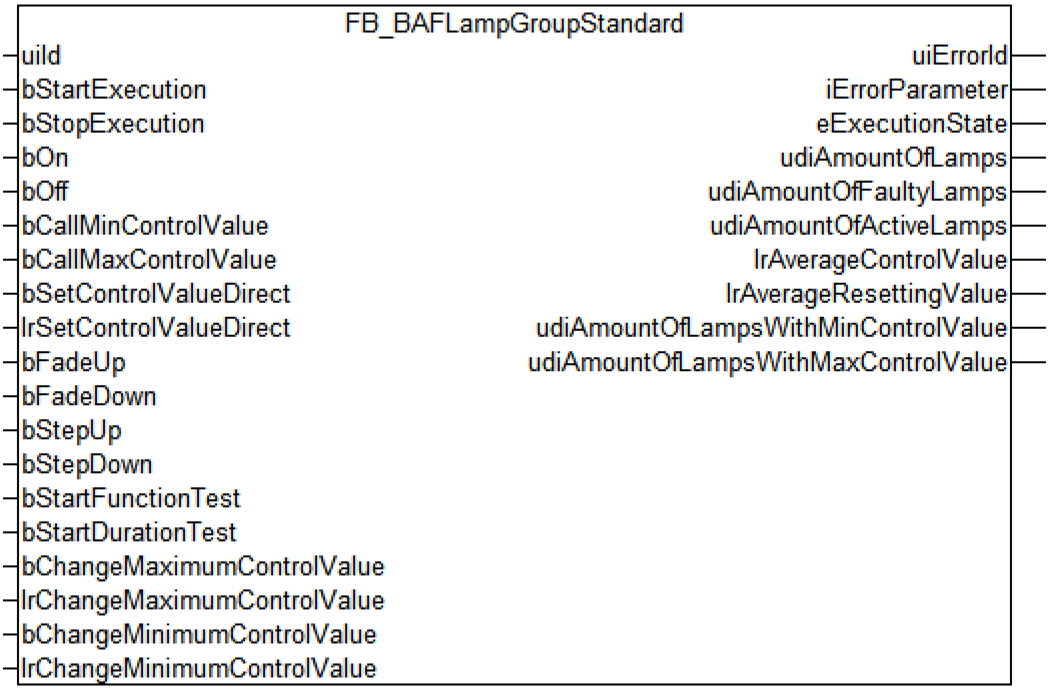
bThresholdSwitch2RisingEdge: Threshold-level 2 is reached - Rising edge.

bThresholdSwitch2FallingEdge: Threshold-level 2 is reached - Falling edge.

8.7 Actuator Groups

8.7.1 Lamp Groups

8.7.1.1 FB_BAFLampGroupStandard



Function-block for controlling a Standard Lamp Group [▶ 104] directly out of the PLC-program.

VAR_INPUT

```
uiId                          : UINT;
bStartExecution               : BOOL;
bStopExecution                : BOOL;
bOn                           : BOOL;
bOff                          : BOOL;
bCallMinControlValue          : BOOL;
bCallMaxControlValue          : BOOL;
bSetControlValueDirect        : BOOL;
lrSetControlValueDirect       : LREAL;
bFadeUp                       : BOOL;
bFadeDown                     : BOOL;
bStepUp                       : BOOL;
bStepDown                     : BOOL;
bStartFunctionTest            : BOOL;
bStartDurationTest            : BOOL;
bChangeMaximumControlValue    : BOOL;
lrChangeMaximumControlValue   : LREAL;
bChangeMinimumControlValue    : BOOL;
lrChangeMinimumControlValue   : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the lamp group.

bStopExecution: Stops the instance of the lamp group.

bOn : The *bOn*-command of all lamps in this group will be executed.

bOff : The *bOff*-command of all lamps in this group will be executed.

bCallMinControlValue: The *bCallMinControlValue*-command of all lamps in this group will be executed.

bCallMaxControlValue: The *bCallMaxControlValue*-command of all lamps in this group will be executed.

bSetControlValueDirect: The *bSetControlValueDirect*-command of all lamps in this group will be executed.

lrSetControlValueDirect: If *bSetControlValueDirect* is set, the value of this variable (0% - 100%) will be put through to all lamps of the group.

bFadeUp: The *bFadeUp*-command of all lamps in this group will be executed.

bFadeDown: The *bFadeDown*-command of all lamps in this group will be executed.

bStepUp: The *bStepUp*-command of all lamps in this group will be executed.

bStepDown: The *bStepDown*-command of all lamps in this group will be executed.

bStartFunctionTest: A function-test will be started at all DALI-Emergency-devices. All other lamps in this group will simply ignore this command.

bStartDurationTest: A duration-test will be started at all DALI-Emergency-devices. All other lamps in this group will simply ignore this command.

bChangeMaximumControlValue: The *bChangeMaximumControlValue*-command of all lamps in this group will be executed.

lrChangeMaximumControlValue: If *bChangeMaximumControlValue* is set, the value of this variable (0% - 100%) will be taken as the lamps´ maximum value.

bChangeMinimumControlValue: The *bChangeMinimumControlValue*-command of all lamps in this group will be executed.

lrChangeMinimumControlValue: If *bChangeMinimumControlValue* is set, the value of this variable (0% - 100%) will be taken as the lamps´ minimum value.

VAR_OUTPUT

```
uiErrorId                           : UINT;
iErrorParameter                     : INT;
eExecutionState                     : E_BAF_ExecutionState;
udiAmountOfLamps                    : UDINT;
udiAmountOfFaultyLamps              : UDINT;
udiAmountOfActiveLamps              : UDINT;
lrAverageControlValue               : LREAL;
lrAverageResettingValue             : LREAL;
udiAmountOfLampsWithMinControlValue : UDINT;
udiAmountOfLampsWithMaxControlValue : UDINT;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution-state of the instance of the lamp group.

udiAmountOfLamps: The amount of all lamps in this group.

udiAmountOfFaultyLamps: The amount of all faulty lamps in this group (*uiErrorId* > 0).

udiAmountOfActiveLamps: The amount of all active lamps in this group. The control-value has to be greater than 0 and the *nErrorId* has to be 0.

lrAverageControlValue: The average control-value of all lamps in this group, which are not in an error-state.

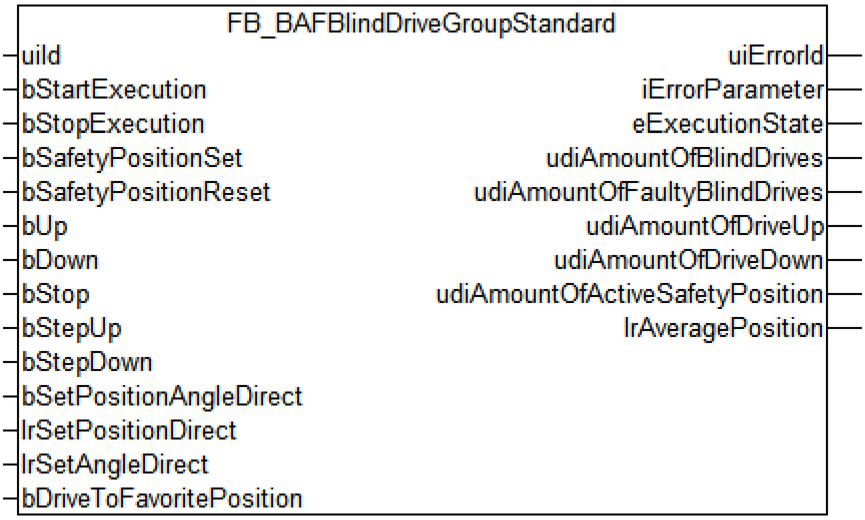
lrAverageResettingValue: The average resetting-value of all lamps in this group, which are not in an error-state.

udiAmountOfLampsWithMinControlValue: The amount of all lamps in this group which are not in an error-state and have the *bReachedMinControlValue*-output set.

udiAmountOfLampsWithMaxControlValue: The amount of all lamps in this group which are not in an error-state and have the *bReachedMaxControlValue*-output set.

8.7.2 Blind Drive Groups

8.7.2.1 FB_BAFBlindDriveGroupStandard



Function-block for controlling a Standard Blind Drive Group [▸ 106] directly out of the PLC-program.

VAR_INPUT

```
uiId                       : UINT;
bStartExecution            : BOOL;
bStopExecution             : BOOL;
bSafetyPositionSet         : BOOL;
bSafetyPositionReset       : BOOL;
bUp                        : BOOL;
bDown                      : BOOL;
bStop                      : BOOL;
bStepUp                    : BOOL;
bStepDown                  : BOOL;
bSetPositionAngleDirect    : BOOL;
lrSetPositionDirect        : LREAL;
lrSetAngleDirect           : LREAL;
bDriveToFavoritePosition   : BOOL;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the blind drive group.

bStopExecution: Stops the instance of the blind drive group.

bSafetyPositionSet: Executes the *bSafetyPositionSet*-command for all blind drives in this group.

bSafetyPositionReset : Executes the *bSafetyPositionReset*-command for all blind drives in this group.

bUp : Executes the *bUp*-command for all blind drives in this group.

bDown: Executes the *bDown*-command for all blind drives in this group.

bStop : Executes the *bStop*-command for all blind drives in this group.

bStepUp : Executes the *bStepUp*-command for all blind drives in this group.

bStepDown: Executes the *bStepDown*-command for all blind drives in this group.

bSetPositionAngleDirect: The blind drives will be moved to the position [0%..100%] entered under *lrSetPositionDirect*. As the BA-Framework-element *Standard Blind Drive* does not support angle-driving functionality - unlike *Blind Drive connected with KL2532* - this command will only have an effect on the position.

lrSetPositionDirect: Specific position where the blinds will be driven to.

lrSetAngleDirect: Specific angle where the blinds will be driven to.

bDriveToFavoritePosition: The blind drives will be moved to the favourite position.

VAR_OUTPUT

```
uiErrorId                         : UINT;
iErrorParameter                   : INT;
eExecutionState                   : E_BAF_ExecutionState;
udiAmountOfBlindDrives            : UDINT;
udiAmountOfFaultyBlindDrives      : UDINT;
udiAmountOfDriveUp                : UDINT;
udiAmountOfDriveDown              : UDINT;
udiAmountOfActiveSafetyPosition   : UDINT;
lrAveragePosition                 : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution-state of the instance of the blind drive group.

udiAmountOfBlindDrives: The number of blind drives in this group.

udiAmountOfFaultyBlindDrives: The amount of faulty blind drives in this group (*uiErrorId* > 0).

udiAmountOfDriveUp: The number of blind drives in this group, which are actually opening (*uiErrorId* has to be 0).

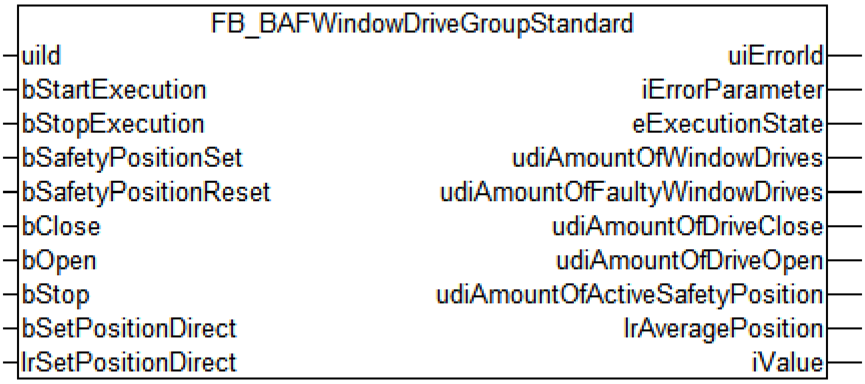
udiAmountOfDriveDown: The number of blind drives in this group, which are actually closing (*uiErrorId* has to be 0).

udiAmountOfActiveSafetyPosition: The number of blind drives in this group standing in safety-position (*uiErrorId* has to be 0).

lrAveragePosition: The average position (0% - 100%) of all active blind drives in this group.

8.7.3 Window Drive Groups

8.7.3.1 FB_BAFWindowDriveGroupStandard



Function-block for controlling a Standard Window Drive Group [▶ 109] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
bSafetyPositionSet      : BOOL;
bSafetyPositionReset    : BOOL;
bClose                  : BOOL;
bOpen                   : BOOL;
bStop                   : BOOL;
bSetPositionDirect      : BOOL;
lrSetPositionDirect     : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the window drive group.

bStopExecution: Stops the instance of the window drive group.

bSafetyPositionSet: Executes the *bSafetyPositionSet*-command for all window drives in this group.

bSafetyPositionReset: Executes the *bSafetyPositionReset*-command for all window drives in this group.

bClose: Executes the *bClose*-command for all window drives in this group.

bOpen: Executes the *bOpen*-command for all window drives in this group.

bStop: Executes the *bStop*-command for all window drives in this group.

bSetPositionDirect: All window drives in this group will be moved to the position entered under *lrSetPositionDirect*.

lrSetPositionDirect: See *bSetPositionDirect*. 0% means completely closed, 100% completely opened.

VAR_OUTPUT

```
uiErrorId                        : UINT;
iErrorParameter                  : INT;
eExecutionState                  : E_BAF_ExecutionState;
udiAmountOfWindowDrives          : UDINT;
udiAmountOfFaultyWindowDrives    : UDINT;
udiAmountOfDriveClose            : UDINT;
udiAmountOfDriveOpen             : UDINT;
udiAmountOfActiveSafetyPosition  : UDINT;
lrAveragePosition                : LREAL;
iValue                           : INT;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution-state of the instance of the window drive group.

udiAmountOfWindowDrives: The amount of window drives in this group.

udiAmountOfFaultyWindowDrives: The amount of faulty window drives in this group (*uiErrorId* > 0).

udiAmountOfDriveClose: The amount of window drives in this group, which are actually closing (*uiErrorId* has to be 0).

udiAmountOfDriveOpen: The amount of window drives in this group, which are opening (*uiErrorId* has to be 0).

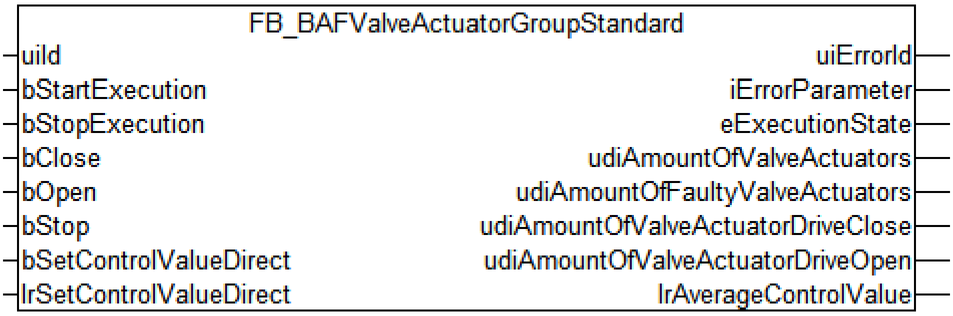
udiAmountOfActiveSafetyPosition: The amount of window drives in this group standing in safety-position (*uiErrorId* has to be 0).

lrAveragePosition: The average position (0% - 100%) of all active window drives in this group.

iValue: Value of the window drive group.

8.7.4 Valve Actuator Groups

8.7.4.1 FB_BAFValveActuatorGroupStandard


FB_BAFValveActuatorGroupStandard
uiId
bStartExecution
bStopExecution
bClose
bOpen
bStop
bSetControlValueDirect
lrSetControlValueDirect
uiErrorId
iErrorParameter
eExecutionState
udiAmountOfValveActuators
udiAmountOfFaultyValveActuators
udiAmountOfValveActuatorDriveClose
udiAmountOfValveActuatorDriveOpen
lrAverageControlValue

Function-block for controlling a Standard Valve Actuator Group [▶ 108] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
bClose                  : BOOL;
bOpen                   : BOOL;
bStop                   : BOOL;
bSetControlValueDirect  : BOOL;
lrSetControlValueDirect : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the valve actuator group.

bStopExecution: Stops the instance of the valve actuator group.

bClose: The *bClose*-command of all valves in this group are executed.

bOpen: The *bOpen*-command of all valves in this group are executed.

bStop: The *bStop*-command of all valves in this group are executed.

bSetControlValueDirect: The *bSetControlValueDirect*-command of all valves in this group are executed.

lrSetControlValueDirect: After sending the command *bSetControlValueDirect* all valves in this group are supposed to drive to this value. 0% completely closed, 100% means fully opened.

VAR_OUTPUT

```
uiErrorId                          : UINT;
iErrorParameter                    : INT;
eExecutionState                    : E_BAF_ExecutionState;
udiAmountOfValveActuators          : UDINT;
udiAmountOfFaultyValveActuators    : UDINT;
udiAmountOfValveActuatorDriveClose : UDINT;
udiAmountOfValveActuatorDriveOpen  : UDINT;
lrAverageControlValue              : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the valve group.

udiAmountOfValveActuators: The amount of valves in this group.

udiAmountOfFaultyValveActuators: The number of faulty valves in this group (*uiErrorId* > 0).

udiAmountOfValveActuatorDriveClose: The number of valves in this group, which are actually closing (*uiErrorId* has to be 0).

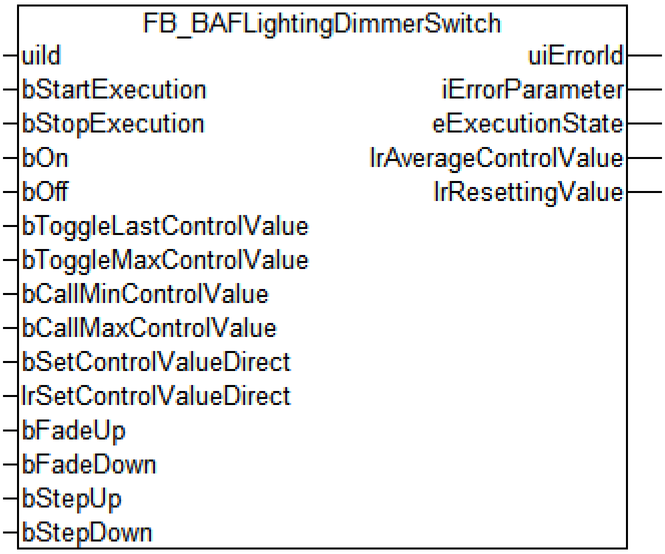
udiAmountOfValveActuatorDriveOpen: The number of valves in this group, which are actually opening (*uiErrorId* has to be 0).

lrAverageControlValue: The average position of all valves in this group, which are not in an error-state.

8.8 Functional Units

8.8.1 Lightings

8.8.1.1 FB_BAFLightingDimmerSwitch



Function-block for controlling a Dimmer-Switch [▶ 111] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
bOn                     : BOOL;
bOff                    : BOOL;
bToggleLastControlValue : BOOL;
bToggleMaxControlValue  : BOOL;
bCallMinControlValue    : BOOL;
bCallMaxControlValue    : BOOL;
bSetControlValueDirect  : BOOL;
lrSetControlValueDirect : LREAL;
bFadeUp                 : BOOL;
bFadeDown               : BOOL;
bStepUp                 : BOOL;
bStepDown               : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the lighting.

bStopExecution: Stops the instance of the lighting.

bOn: The assigned lamp group will be turned on.

bOff: The assigned lamp group will be turned off. If the average value of the lamp group (*lrAverageControlValue*) is greater than 0, it will be stored in *lrResettingValue*.

bToggleLastControlValue: *uiToggleMode* = 0 (toggling between On and Off): the assigned lamp group will be turned on if the average control value is 0 and turned off if it is greater than 0. These actions will be done by sending either the *bOn* or the *bOff*-command.

uiToggleMode = 0 (toggling between *lrResettingValue* and Off): If the average control value of the assigned lamp group is 0, the group will be set to *lrResettingValue* by sending the *bSetControlValueDirect*-command. The group will be turned off, if its average control value is greater than 0 by sending the *bOff*-command.

bToggleMaxControlValue: The assigned lamp group will be switched between Off and the maximum control value. If the average control-value is greater than 0, the *bOff*-command will be sent to the specific group. If it is 0, the command *bCallMaxControlValue* will be executed.

bCallMinControlValue: The command *bCallMinControlValue* will be sent to the assigned lamp group.

bCallMaxControlValue: The command *bCallMaxControlValue* will be sent to the assigned lamp group.

bSetControlValueDirect: The command *bSetControlValueDirect* will be sent to the assigned lamp group. If *lrSetControlValueDirect* is 0, *lrAverageControlValue* will be saved in *lrResettingValue*.

lrSetControlValueDirect: Control value, which will be sent to the assigned lamp group, if *bSetControlValueDirect* was executed.

bFadeUp: The command *bFadeUp* will be sent to the assigned lamp group.

bFadeDown: The command *bFadeDown* will be sent to the assigned lamp group.

bStepUp: The *bStepUp*-command of all lamps of the assigned lamp group will be executed.

bStepDown: The *bStepDown*-command of all lamps of the assigned lamp group will be executed.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
lrAverageControlValue : LREAL;
lrResettingValue      : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

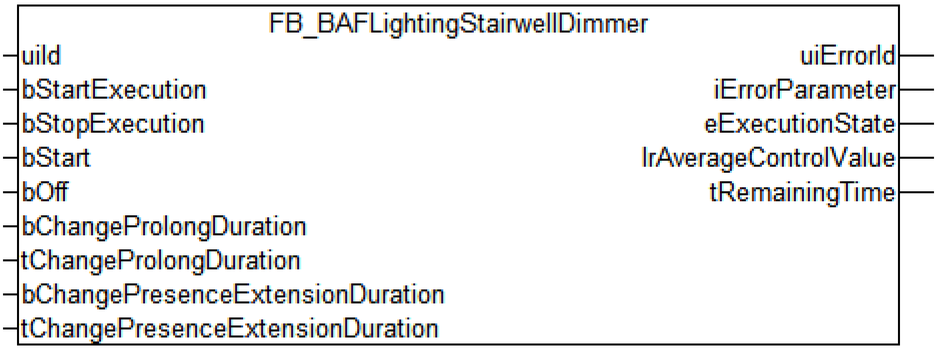
iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the lighting.

lrAverageControlValue: The average control value (0% - 100%) of the assigned lamp group.

lrResettingValue: The resetting value (0% - 100%) of the lighting. If the lighting is turned off, the average value *lrAverageControlValue* will be stored in this variable.

8.8.1.2 FB_BAFLightingStairwellDimmer



Function-block for controlling a Stairwell-Dimmer [▶ 113] directly out of the PLC-program.

VAR_INPUT

```
uiId                              : UINT;
bStartExecution                   : BOOL;
bStopExecution                    : BOOL;
bStart                            : BOOL;
bOff                              : BOOL;
bChangeProlongDuration            : BOOL;
```

```
tChangeProlongDuration            : TIME;
bChangePresenceExtensionDuration : BOOL;
tChangePresenceExtensionDuration : TIME;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the lighting.

bStopExecution: Stops the instance of the lighting.

bStart: The stairwell sequence will be started for the assigned lamp group.

bOff: The stairwell sequence will be terminated and the assigned lamp group will be turned off.

bChangeProlongDuration: The prolong duration of the lighting will be set to *tChangeProlongDuration*.

tChangeProlongDuration: See *bChangeProlongDuration*.

bChangePresenceExtensionDuration: The presence extension duration of the lighting will be set to *tChangePresenceExtensionDuration*.

tChangePresenceExtensionDuration: See *bChangePresenceExtensionDuration*.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
lrAverageControlValue : LREAL;
tRemainingTime        : TIME;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

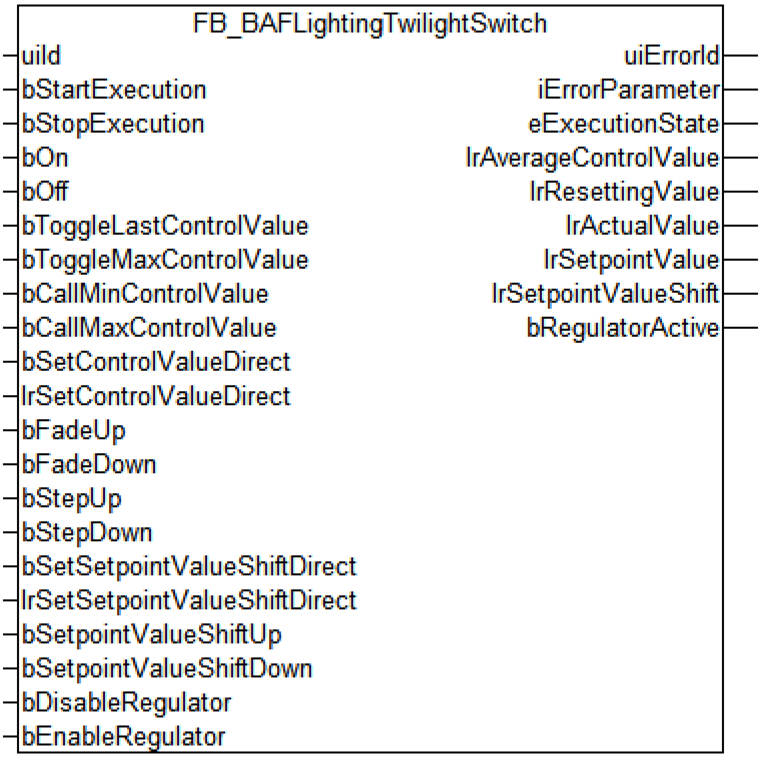
iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution state of the instance of the lighting.

lrAverageControlValue: The average control value (0% - 100%) of the assigned lamp group.

tRemainingTime: Time remaining until the lighting turns off.

8.8.1.3 FB_BAFLightingTwilightSwitch



Function-block for controlling a Twighlight-Switch [▶ 115] directly out of the PLC-program.

VAR_INPUT

```
uiId                          : UINT;
bStartExecution               : BOOL;
bStopExecution                : BOOL;
bOn                           : BOOL;
bOff                          : BOOL;
bToggleLastControlValue       : BOOL;
bToggleMaxControlValue        : BOOL;
bCallMinControlValue          : BOOL;
bCallMaxControlValue          : BOOL;
bSetControlValueDirect        : BOOL;
lrSetControlValueDirect       : LREAL;
bFadeUp                       : BOOL;
bFadeDown                     : BOOL;
bStepUp                       : BOOL;
bStepDown                     : BOOL;
bSetSetpointValueShiftDirect  : BOOL;
lrSetSetpointValueShiftDirect : LREAL;
bSetpointValueShiftUp         : BOOL;
bSetpointValueShiftDown       : BOOL;
bDisableRegulator             : BOOL;
bEnableRegulator              : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the lighting.

bStopExecution: Stops the instance of the lighting.

bOn: The assigned lamp group will be turned on.

bOff: The assigned lamp group will be turned off. If the average value of the lamp group (*lrAverageControlValue*) is greater than 0, it will be stored in *lrResettingValue*.

bToggleLastControlValue: *uiToggleMode* = 0 (toggling between On and Off): the assigned lamp group will be turned on if the average control value is 0 and turned off if it is greater than 0. These actions will be done by sending either the *bOn* or the *bOff*-command.
uiToggleMode = 0 (toggling between *lrResettingValue* and Off): If the average control value of the assigned lamp group is 0, the group will be set to *lrResettingValue* by sending the *bSetControlValueDirect*-command. The group will be turned off, if its average control value is greater than 0 by sending the *bOff*-command.

bToggleMaxControlValue: The assigned lamp group will be switched between Off and the maximum control value. If the average control-value is greater than 0, the *bOff*-command will be sent to the specific group. If it is 0, the command *bCallMaxControlValue* will be executed.

bCallMinControlValue: The command *bCallMinControlValue* will be sent to the assigned lamp group.

bCallMaxControlValue: The command *bCallMaxControlValue* will be sent to the assigned lamp group.

bSetControlValueDirect: The command *bSetControlValueDirect* will be sent to the assigned lamp group. If *lrSetControlValueDirect* is 0, *lrAverageControlValue* will be saved in *lrResettingValue*.

lrSetControlValueDirect: Control value, which will be sent to the assigned lamp group, if *bSetControlValueDirect* was executed.

bFadeUp: The command *bFadeUp* will be sent to the assigned lamp group.

bFadeDown: The command *bFadeDown* will be sent to the assigned lamp group.

bStepUp: The *bStepUp*-command of all lamps of the assigned lamp group will be executed.

bStepDown: The *bStepDown*-command of all lamps of the assigned lamp group will be executed.

bSetSetpointValueShiftDirect: The switching value shift, entered in *lrSetSetpointValueShiftDirect* will be taken over.

lrSetSetpointValueShiftDirect: See *lrSetSetpointValueShiftDirect*.

bSetpointValueShiftUp: The switching value shift will be increased by *lrSetpointValueShiftStepSize* (parameter structure).

bSetpointValueShiftDown: The switching value shift will be decreased by *lrSetpointValueShiftStepSize* (parameter structure).

bDisableRegulator: A rising edge at this input disables the twilight control.

bEnableRegulator: A rising edge at this input activates the twilight control.

Note : All commands, which change the light control value directly (*bOn, bOff, bToggleLastControlValue, bToggleMaxControlValue, bCallMinControlValue, bCallMaxControlValue, bSetControlValueDirect, bFadeUp* and *bFadeDown*) are disabled as long as the control is active.

The automatic switching of the assigned lamp group happens with a programmed hysteresis (*lrControlHysteresis*, parameter structure) around the switching value.

VAR_OUTPUT

```
uiErrorId              : UINT;
iErrorParameter        : INT;
eExecutionState        : E_BAF_ExecutionState;
lrAverageControlValue  : LREAL;
lrResettingValue       : LREAL;
lrActualValue          : LREAL;
lrSetpointValue        : LREAL;
lrSetpointValueShift   : LREAL;
bRegulatorActive       : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution-state of the instance of the lighting.

lrAverageControlValue: The average control value (0% - 100%) of the assigned lamp group.

lrResettingValue: The resetting-value (0% - 100%) of the lighting. If the lighting is turned off, the average value *lrAverageControlValue* will be stored in this variable.

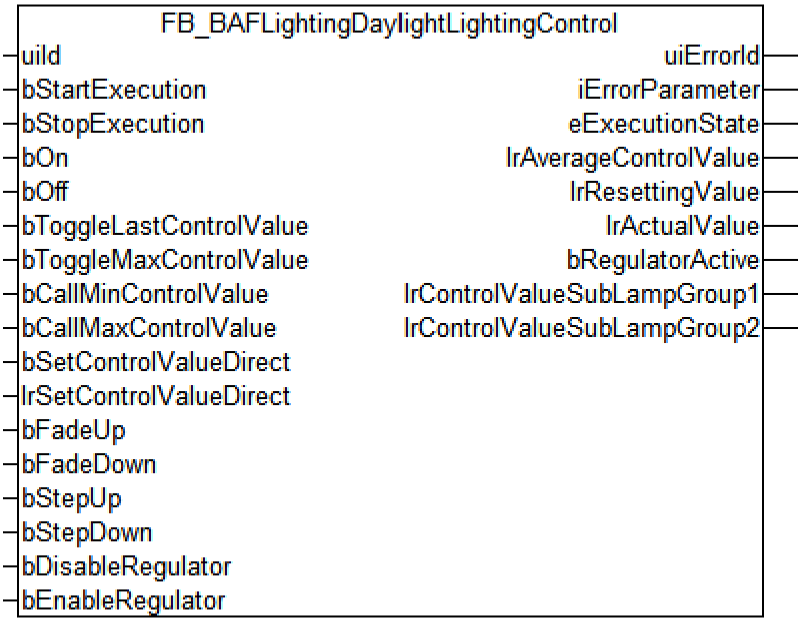
lrActualValue: Measured (day-) light-value of the control.

lrSetpointValue: Switching value. Result of the basic switching value (*lrBasicSetpointValue*) and the switching value shift (*lrSetpointValueShift*).

lrSetpointValueShift: The actual switching value shift.

bRegulatorActive: Indicates, if the control is actually active or disabled.

8.8.1.4 FB_BAFLightingDaylightLightControl



Function-block for controlling a Daylight Lighting-Control [▶ 119] directly out of the PLC-program.

VAR_INPUT

```
uiId                      : UINT;
bStartExecution           : BOOL;
bStopExecution            : BOOL;
bOn                       : BOOL;
bOff                      : BOOL;
bToggleLastControlValue   : BOOL;
bToggleMaxControlValue    : BOOL;
bCallMinControlValue      : BOOL;
bCallMaxControlValue      : BOOL;
bSetControlValueDirect    : BOOL;
lrSetControlValueDirect   : LREAL;
bFadeUp                   : BOOL;
bFadeDown                 : BOOL;
bStepUp                   : BOOL;
bStepDown                 : BOOL;
bDisableRegulator         : BOOL;
bEnableRegulator          : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the lighting.

bStopExecution: Stops the instance of the lighting.

bOn: The assigned lamp group will be turned on.

bOff: The assigned lamp group will be turned off. If the average value of the lamp group (*lrAverageControlValue*) is greater than 0, it will be stored in *lrResettingValue*.
Sending this command, while the light-control is active will turn off all assigned lamp-groups and disable the regulator.

bToggleLastControlValue: *uiToggleMode* = 0 (toggling between On and Off): the assigned lamp group will be turned on if the average control value is 0 and turned off if it is greater than 0. These actions will be done by sending either the *bOn* or the *bOff*-command.
uiToggleMode = 0 (toggling between *lrResettingValue*and Off): If the average control value of the assigned lamp group is 0, the group will be set to *lrResettingValue* by sending the *bSetControlValueDirect*-command. The group will be turned off, if its average control value is greater than 0 by sending the *bOff*-command.

bToggleMaxControlValue: The assigned lamp group will be switched between Off and the maximum control-value. If the average control value is greater than 0, the *bOff*-command will be sent to the specific group. If it is 0, the command *bCallMaxControlValue* will be executed.

bCallMinControlValue: The command *bCallMinControlValue* will be sent to the assigned lamp group.

bCallMaxControlValue: The command *bCallMaxControlValue* will be sent to the assigned lamp group.

bSetControlValueDirect: The command *bSetControlValueDirect* will be sent to the assigned lamp group. If *lrSetControlValueDirect* is 0, *lrAverageControlValue* will be saved in *lrResettingValue*.

lrSetControlValueDirect: Control value, which will be sent to the assigned lamp group, if *bSetControlValueDirect* was executed.

bFadeUp: The command *bFadeUp* will be sent to the assigned lamp group.

bFadeDown: The command *bFadeDown* will be sent to the assigned lamp group.

bStepUp: The *bStepUp*-command of all lamps of the assigned lamp groups will be executed.

bStepDown: The *bStepDown*-command of all lamps of the assigned lamp groups will be executed.

bSetSetpointValueShiftDirect: The setpoint value shift, entered in *lrSetSetpointValueShiftDirect* will be taken over.

lrSetSetpointValueShiftDirect: See *lrSetSetpointValueShiftDirect*.

bSetpointValueShiftUp: The setpoint value shift will, be increased by *lrSetpointValueShiftStepSize* (parameter structure).

bSetpointValueShiftDown: The setpoint value shift will, be decreased by *lrSetpointValueShiftStepSize* (parameter structure).

bDisableRegulator: A rising edge at this input disables the twilight control.

bEnableRegulator: A rising edge at this input activates the twilight control.

All commands, which change the light control value directly (*bOn, bOff, bToggleLastControlValue, bToggleMaxControlValue, bCallMinControlValue, bCallMaxControlValue, bSetControlValueDirect, bFadeUp* and *bFadeDown*) are disabled if the control is active.

VAR_OUTPUT

```
uiErrorId                   : UINT;
iErrorParameter             : INT;
eExecutionState             : E_BAF_ExecutionState;
lrAverageControlValue       : LREAL;
lrResettingValue            : LREAL;
lrActualValue               : LREAL;
bRegulatorActive            : BOOL;
lrControlValueSubLampGroup1 : LREAL;
lrControlValueSubLampGroup2 : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution-state of the instance of the lighting.

lrAverageControlValue: The average control value (0% - 100%) of the assigned lamp group.

lrResettingValue: The resetting value (0% - 100%) of the lighting. If the lighting is turned off, the average value *lrAverageControlValue* will be stored in this variable.

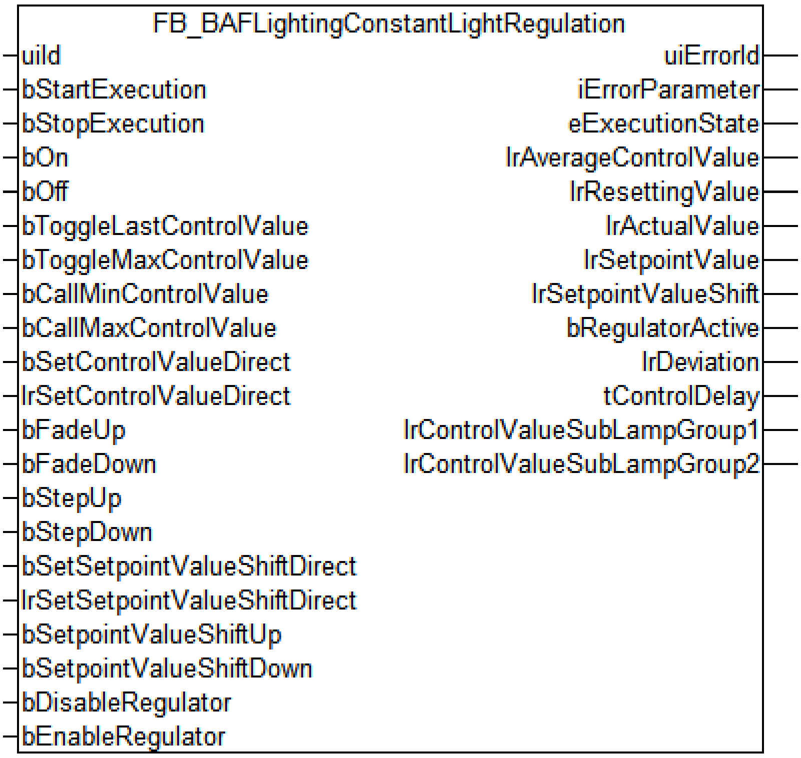
lrActualValue: Measured (day-) light value of the control.

bRegulatorActive: Tells if the control is actually active or disabled.

lrControlValueSubLampGroup1: The average-control-value (0% - 100%) of the assigned sub lamp group 1.

lrControlValueSubLampGroup2: The average-control-value (0% - 100%) of the assigned sub lamp group 2.

8.8.1.5 FB_BAFLightingConstantLightRegulation



Function-block for controlling a Constant Light-Regulation [▶ 124] directly out of the PLC-program.

VAR_INPUT

```
uiId                          : UINT;
bStartExecution               : BOOL;
bStopExecution                : BOOL;
bOn                           : BOOL;
bOff                          : BOOL;
bToggleLastControlValue       : BOOL;
bToggleMaxControlValue        : BOOL;
bCallMinControlValue          : BOOL;
bCallMaxControlValue          : BOOL;
bSetControlValueDirect        : BOOL;
lrSetControlValueDirect       : LREAL;
bFadeUp                       : BOOL;
bFadeDown                     : BOOL;
bStepUp                       : BOOL;
bStepDown                     : BOOL;
bSetSetpointValueShiftDirect  : BOOL;
lrSetSetpointValueShiftDirect : LREAL;
bSetpointValueShiftUp         : BOOL;
bSetpointValueShiftDown       : BOOL;
bDisableRegulator             : BOOL;
bEnableRegulator              : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the lighting.

bStopExecution: Stops the instance of the lighting.

bOn: The assigned lamp group will be turned on.

bOff: The assigned lamp group will be turned off. If the average value of the lamp group (*lrAverageControlValue*) is greater than 0, it will be stored in *lrResettingValue*.

Sending this command, while the light control is active will turn off all assigned lamp groups and disable the regulator.

bToggleLastControlValue: *uiToggleMode* = 0 (toggling between On and Off): the assigned lamp group will be turned on if the average control value is 0 and turned off if it is greater than 0. These actions will be done by sending either the *bOn* or the *bOff*-command.
uiToggleMode = 0 (toggling between *lrResettingValue* and Off): If the average control value of the assigned lamp group is 0, the group will be set to *lrResettingValue* by sending the *bSetControlValueDirect*-command. The group will be turned off, if its average control value is greater than 0 by sending the *bOff*-command.

bToggleMaxControlValue: The assigned lamp group will be switched between Off and the maximum control value. If the average control value is greater than 0, the *bOff*-command will be sent to the specific group. If it is 0, the command *bCallMaxControlValue* will be executed.

bCallMinControlValue: The command *bCallMinControlValue* will be sent to the assigned lamp group.

bCallMaxControlValue: The command *bCallMaxControlValue* will be sent to the assigned lamp group.

bSetControlValueDirect: The command *bSetControlValueDirect* will be sent to the assigned lamp group. If *lrSetControlValueDirect* is 0, *lrAverageControlValue* will be saved in *lrResettingValue*.

lrSetControlValueDirect: Control value, which will be sent to the assigned lamp group, if *bSetControlValueDirect* was executed.

bFadeUp: The command *bFadeUp* will be sent to the assigned lamp group.

bFadeDown: The command *bFadeDown* will be sent to the assigned lamp group.

bStepUp: The *bStepUp*-command of all lamps of the assigned lamp groups will be executed.

bStepDown: The *bStepDown*-command of all lamps of the assigned lamp-groups will be executed.

bSetSetpointValueShiftDirect: The setpoint value shift, entered in *lrSetSetpointValueShiftDirect* will be taken over.

lrSetSetpointValueShiftDirect: See *lrSetSetpointValueShiftDirect*.

bSetpointValueShiftUp: The setpoint value shift will, be increased by *lrSetpointValueShiftStepSize* (parameter structure).

bSetpointValueShiftDown: The setpoint value shift will, be decreased by *lrSetpointValueShiftStepSize* (parameter structure).

bDisableRegulator: A rising edge at this input disables the twilight-control.

bEnableRegulator: A rising edge at this input activates the twilight-control.

All commands, which change the light control value directly (*bOn, bOff, bToggleLastControlValue, bToggleMaxControlValue, bCallMinControlValue, bCallMaxControlValue, bSetControlValueDirect, bFadeUp* and *bFadeDown*) are disabled if the control is active.

The dimming of the assigned lamp groups happens with a programmed hysteresis (*lrControlHysteresis*, parameter structure) around the setpoint value.

VAR_OUTPUT

```
uiErrorId                    : UINT;
iErrorParameter              : INT;
eExecutionState              : E_BAF_ExecutionState;
lrAverageControlValue        : LREAL;
lrResettingValue             : LREAL;
lrActualValue                : LREAL;
lrSetpointValue              : LREAL;
lrSetpointValueShift         : LREAL;
bRegulatorActive             : BOOL;
lrDeviation                  : LREAL;
tControlDelay                : TIME;
lrControlValueSubLampGroup1  : LREAL;
lrControlValueSubLampGroup2  : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution-state of the instance of the lighting.

lrAverageControlValue: The average control value (0% - 100%) of the assigned lamp group.

lrResettingValue: The resetting-value (0% - 100%) of the lighting. If the lighting is turned off, the average value *lrAverageControlValue* will be stored in this variable.

lrActualValue: Measured (day-) light value of the control.

lrSetpointValue: Result of the basic setpoint value (*lrBasicSetpointValue*) and the setpoint value shift (*lrSetpointValueShift*).

lrSetpointValueShift: The actual setpoint value shift.

bRegulatorActive: Tells if the control is actually active or disabled.

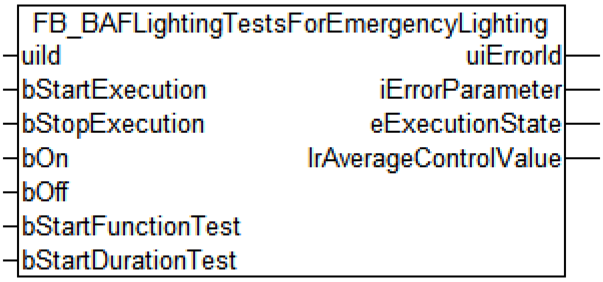
lrDeviation: Control deviation (*lrSetpointValue - lrActualValue*).

tControlDelay: Actual delay to take over a new control-value. This delay is calculated dynamically regarding the actual deviation. The higher the deviation, the shorter the control-delay.

lrControlValueSubLampGroup1: The average control value (0% - 100%) of the assigned sub-lamp-group 1.

lrControlValueSubLampGroup2: The average control value (0% - 100%) of the assigned sub-lamp-group 2.

8.8.1.6 FB_BAFLightingTestsForEmergencyLighting



Function-block for starting a Functiontest and Durationtest for Emergency Lighting directly out of the PLC-program.

VAR_INPUT

```
uiId                  : UINT;
bStartExecution       : BOOL;
bStopExecution        : BOOL;
bOn                   : BOOL;
bOff                  : BOOL;
bStartFunctionTest    : BOOL;
bStartDurationTest    : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the lighting.

bStopExecution: Stops the instance of the lighting.

bOn: Turns the lighting on.

bOff Turns the lighting off.

bStartFunctionTest: A function test will be started at all DALI-emergency-devices in the assigned lamp group.

bStartDurationTest: A duration test will be started at all DALI-emergency-devices in the assigned lamp group.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
lrAverageControlValue : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

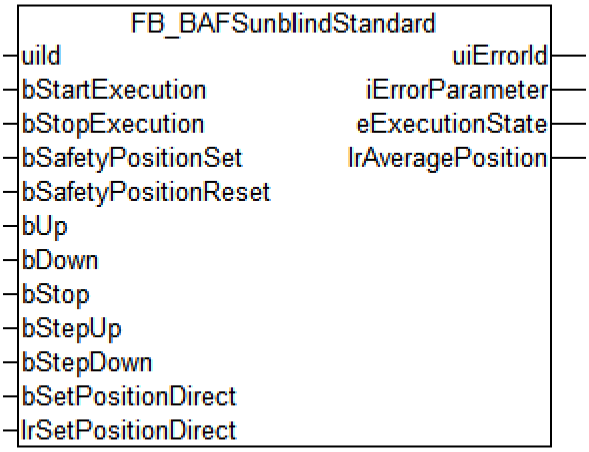
iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the lighting.

lrAverageControlValue: The average control value of the lighting.

8.8.2 Sunblinds

8.8.2.1 FB_BAFSunblindStandard



Function-block for controlling a Standard Sunblind [▶ 130] directly out of the PLC-program.

VAR_INPUT

```
uiId                 : UINT;
bStartExecution      : BOOL;
bStopExecution       : BOOL;
bSafetyPositionSet   : BOOL;
bSafetyPositionReset : BOOL;
bUp                  : BOOL;
bDown                : BOOL;
bStop                : BOOL;
bStepUp              : BOOL;
bStepDown            : BOOL;
bSetPositionDirect   : BOOL;
lrSetPositionDirect  : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the sunblind.

bStopExecution: Stops the instance of the sunblind.

bSafetyPositionSet: All blind drives in the assigned blind drive group (see parameter *uiBlindDriveGroupId*) will move to their safety-positions.

bSafetyPositionReset: All blind drives in the assigned blind drive group are released do move from their safety-positions.

bUp: All blind drives in the assigned blind drive group will move up.

bDown: All blind drives in the assigned blind drive group will move down.

bStop: All blind drives in the assigned blind drive group will stop moving.

bStepUp: The *bStepUp*-command for the assigned blind drive group will be sent.

bStepDown: The *bStepDown*-command for the assigned blind drive group will be sent.

bSetPositionDirect: All blind drives in the assigned group will be driven to the position entered in *lrSetPositionDirect*.

lrSetPositionDirect: See *bSetPositionDirect*. 0% means fully opened, 100% means completely closed.

VAR_OUTPUT

```
uiErrorId              : UINT;
iErrorParameter        : INT;
eExecutionState        : E_BAF_ExecutionState;
lrAveragePosition      : LREAL;
```

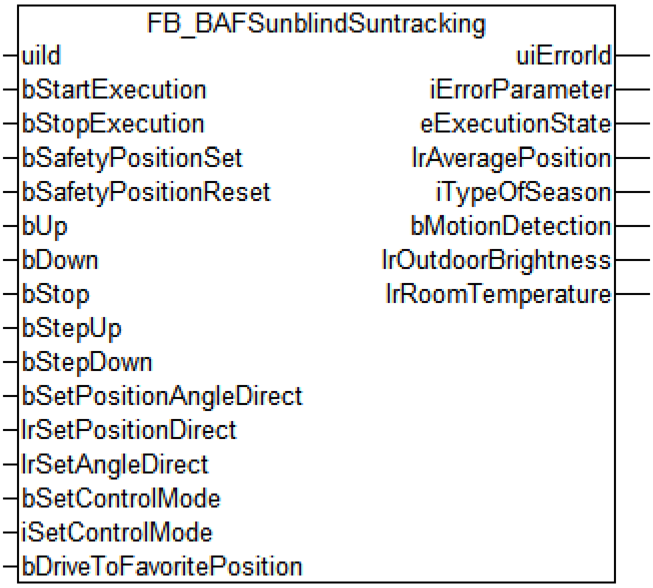
uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the sunblind unit.

lrAveragePosition: The average position (0% - 100%) of the assigned blind drive group.

8.8.2.2 FB_BAFSunblindSuntracking



Function-block for controlling a Shading as a function of the position of the sun [▶ 131] directly out of the PLC-program.

VAR_INPUT

```
uiId                     : UINT;
bStartExecution          : BOOL;
bStopExecution           : BOOL;
bSafetyPositionSet       : BOOL;
bSafetyPositionReset     : BOOL;
bUp                      : BOOL;
bDown                    : BOOL;
bStop                    : BOOL;
bStepUp                  : BOOL;
bStepDown                : BOOL;
bSetPositionAngleDirect  : BOOL;
lrSetPositionDirect      : LREAL;
lrSetAngleDirect         : LREAL;
bSetControlMode          : BOOL;
iSetControlMode          : INT;
bDriveToFavoritePosition : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the sunblind.

bStopExecution: Stops the instance of the sunblind.

bSafetyPositionSet: All blind drives in the assigned blind drive group (see parameter *uiBlindDriveGroupId*) will move to their safety-positions.

bSafetyPositionReset: All blind drives in the assigned blind drive group are released do move from their safety-positions.

bUp: All blind drives in the assigned blind drive group will move up.

bDown: All blind drives in the assigned blind drive group will move down.

bStop: All blind drives in the assigned blind drive group will stop moving.

bStepUp: The *bStepUp*-command for the assigned blind drive group will be sent.

bStepDown: The *bStepDown*-command for the assigned blind drive group will be sent.

bSetPositionAngleDirect: The sunblind will be moved to the angle entered under *lrSetAngleDirect*. As the BA-Framework-element *Standard Blind Drive* does not support angle-driving functionality - unlike *Blind Drive connected with KL2532* - this command will only influence the position.

lrSetPositionDirect: See *bSetPositionAngleDirect*. 0% means fully opened, 100% means completely closed.

lrSetAngleDirect: See *bSetPositionAngleDirect*.

bSetControlMode: Sets the sunblind into the control mode given under *iSetControlMode*.

iSetControlMode: See *bSetControlMode*.

bDriveToFavoritePosition: Drives the sunblind to the favourite position.

VAR_OUTPUT

```
uiErrorId                  : UINT;
iErrorParameter            : INT;
eExecutionState            : E_BAF_ExecutionState;
lrAveragePosition          : LREAL;
iTypeOfSeason              : INT;
bMotionDetection           : BOOL;
lrOutdoorBrightness        : LREAL;
lrRoomTemperature          : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the lighting.

lrAveragePosition: The average position (0% - 100%) of the assigned blind drive group.

iTypeOfSeason: Shows the actual operating-mode: (0: undefined ; 1: winter-mode ; 2: summer-mode).

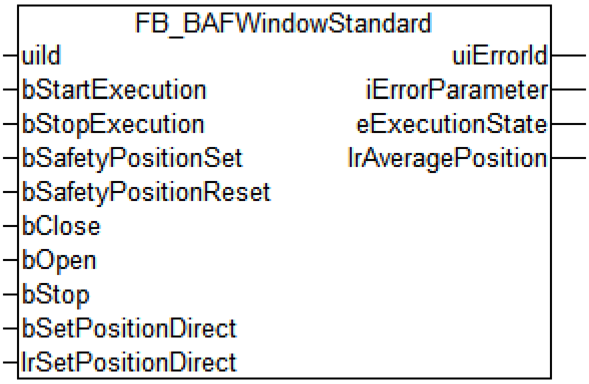
bMotionDetection: Indicates if motion was detected.

lrOutdoorBrightness: Represents the outdoor brightness.

lrRoomTemperature: Shows the room temperature.

8.8.3 Windows

8.8.3.1 FB_BAFWindowStandard



Function-block for controlling a Standard Window [▶ 142] directly out of the PLC-program.

VAR_INPUT

```
uiId                  : UINT;
bStartExecution       : BOOL;
bStopExecution        : BOOL;
bSafetyPositionSet    : BOOL;
bSafetyPositionReset  : BOOL;
bClose                : BOOL;
bOpen                 : BOOL;
bStop                 : BOOL;
bSetPositionDirect    : BOOL;
lrSetPositionDirect   : LREAL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the window unit.

bStopExecution: Stops the instance of the window unit.

bSafetyPositionSet: All window drives in the assigned window drive group will move to their safety-positions.

bSafetyPositionReset: All window drives in the assigned window drive group are released do move from their safety-positions.

bClose: All window-drives in the assigned window drive group will close.

bOpen: All window-drives in the assigned window drive group will open.

bStop: All window-drives in the assigned window drive group will stop moving.

bSetPositionDirect: All window drives in the assigned group will be driven to the position entered in *lrSetPositionDirect*.

lrSetPositionDirect: See *bSetPositionDirect*. 0% means completely closed, 100% means fully opened.

VAR_OUTPUT

```
uiErrorId                : UINT;
iErrorParameter          : INT;
eExecutionState          : E_BAF_ExecutionState;
lrAveragePosition        : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

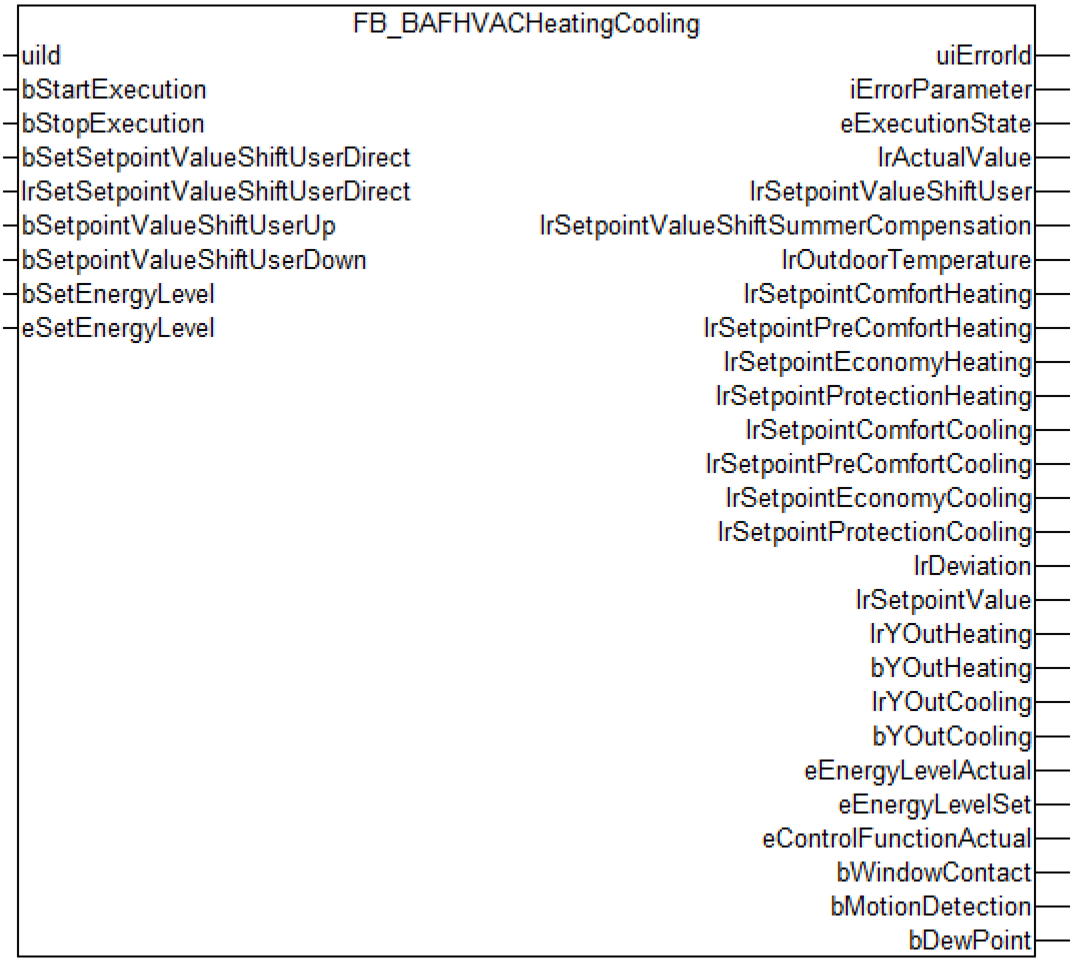
iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the window unit.

lrAveragePosition: The average position (0% - 100%) of the assigned window drive group.

8.8.4 HVAC

8.8.4.1 FB_BAFHVACHeatingCooling



Function block for controlling the functional unit for the heating and cooling [▶ 135] temperature control directly from the PLC program.

VAR_INPUT

```
uiId                              : UINT;
bStartExecution                   : BOOL;
bStopExecution                    : BOOL;
bSetSetpointValueShiftUserDirect  : BOOL;
lrSetSetpointValueShiftUserDirect : LREAL;
bSetpointValueShiftUserUp         : BOOL;
bSetpointValueShiftUserDown       : BOOL;
bSetEnergyLevel                   : BOOL;
eSetEnergyLevel                   : E_BAF_HVACEnergyLevel;
```

uiId: object ID of the instance (see also [▶ 200]).

bStartExecution: starts the instance.

bStopExecution: stops the instance.

bSetSetpointValueShiftUserDirect: on a rising edge the local setpoint value shift (user setpoint value shift) is set to the value entered under *lrSetSetpointValueShiftUserDirect*.

lrSetSetpointValueShiftUserDirect: see *bSetSetpointValueShiftUserDirect*: The value can be limited to a minimum and maximum value by parameters.

bSetpointValueShiftUserUp: the local setpoint value shift (user setpoint value shift) is increased by a rising edge. Parameters can be used to define how much the value should change.

bSetpointValueShiftUserDown: the local setpoint value shift (user setpoint value shift) is decreased by a rising edge. Parameters can be used to define how much the value should change.

bSetEnergyLevel: specification of the energy level. Window contacts or motion sensors, which are linked by digital signals to the heating controller, can overwrite this specification.

eSetEnergyLevel: see *bSetEnergyLevel.*

VAR_OUTPUT

```
uiErrorId                              : UINT;
iErrorParameter                        : INT;
eExecutionState                        : E_BAF_ExecutionState;
lrActualValue                          : LREAL;
lrSetpointValueShiftUser               : LREAL;
lrSetpointValueShiftSummerCompensation : LREAL;
lrOutdoorTemperature                   : LREAL;
lrSetpointComfortHeating               : LREAL;
lrSetpointPreComfortHeating            : LREAL;
lrSetpointEconomyHeating               : LREAL;
lrSetpointProtectionHeating            : LREAL;
lrSetpointComfortCooling               : LREAL;
lrSetpointPreComfortCooling            : LREAL;
lrSetpointEconomyCooling               : LREAL;
lrSetpointProtectionCooling            : LREAL;
lrDeviation                            : LREAL;
lrSetpointValue                        : LREAL;
lrYOutHeating                          : LREAL;
bYOutHeating                           : BOOL;
lrYOutCooling                          : LREAL;
bYOutCooling                           : BOOL;
eEnergyLevelActual                     : E_BAF_HVACEnergyLevel;
eEnergyLevelSet                        : E_BAF_HVACEnergyLevel;
eControlFunctionActual                 : E_BAF_HVACControlFunction;
bWindowContact                         : BOOL;
bMotionDetection                       : BOOL;
bDewPoint                              : BOOL;
```

uiErrorId: contains the error code (see error codes [▶ 276]).

iErrorParameter: contains the specific parameters of the error code (see error codes [▶ 276]).

eExecutionState: contains the current execution status of the instance.

lrActualValue: actual value.

lrSetpointValueShiftUser: local setpoint value shift (user setpoint value shift).

lrSetpointValueShiftSummerCompensation: setpoint value shift due to summer compensation.

lrOutdoorTemperature: outdoor temperature if a weather station has been connected to the air conditioning controller.

lrSetpointComfortHeating: base setpoint value for heating with the *Comfort* energy level.

lrSetpointPreComfortHeating: base setpoint value for heating with the *Pre-comfort* energy level.

lrSetpointEconomyHeating: base setpoint value for heating with the *Economy* energy level.

lrSetpointProtectionHeating: base setpoint value for heating with the *Protection* energy level.

lrSetpointComfortCooling: base setpoint value for cooling with the *Comfort* energy level.

lrSetpointPreComfortCooling: base setpoint value for cooling with the *Pre-comfort* energy level.

lrSetpointEconomyCooling: base setpoint value for cooling with the *Economy* energy level.

lrSetpointProtectionCooling: base setpoint value for cooling with the *Protection* energy level.

lrDeviation: control deviation (setpoint value - actual value).

lrSetpointValue: calculated setpoint value. The resulting setpoint value is composed of one of the 8 base setpoint values, the local setpoint value shift (user setpoint value shift), the central setpoint value shift and the setpoint value shift due to the summer compensation.

lrYOutHeating: analog control value for heating from 0% to 100%

bYOutHeating: digital control value for heating. PWM signal of the analog control value.

lrYOutCooling: analog control value for cooling from 0% to 100%

bYOutCooling: digital control value for cooling. PWM signal of the analog control value.

eEnergyLevelActual: calculated energy level.

eEnergyLevelSet: specification of the energy level.

eControlFunctionActual: calculated control function.

bWindowContact: state of the window contact if window contacts have been connected to the air conditioning controller.

bMotionDetection: state of the motion sensor if motion sensors have been connected to the air conditioning controller.

bDewPointValue: state of the dew point sensor if dew point sensors have been connected to the air conditioning controller.

8.9 Scenes

8.9.1 FB_BAFScenesStandard

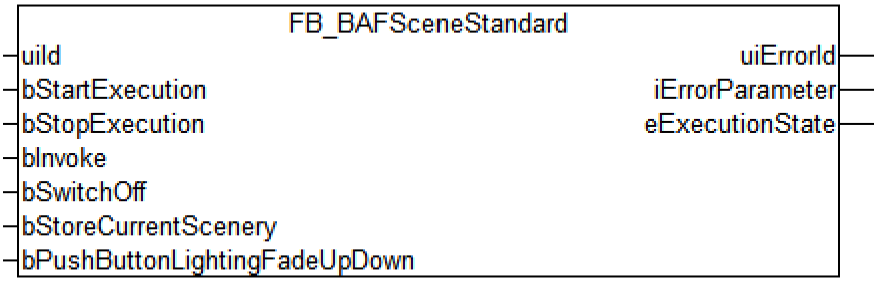


Fig. 5: TcBAPlc_FB_BAFSceneStandard

Function block for controlling a standard scene [▸ 150] directly from the PLC program.

VAR_INPUT

```
uiId                          : UINT;
bStartExecution               : BOOL;
bStopExecution                : BOOL;
bInvoke                       : BOOL;
bSwitchOff                    : BOOL;
bStoreCurrentScenery          : BOOL;
bPushButtonLightingFadeUpDown : BOOL;
```

uiId: object ID of the instance (see also [▸ 200]).

bStartExecution: starts the instance of the scene.

bStopExecution: stops the instance of the scene.

bInvoke: the scene is called.

bSwitchOff: all elements contained in this scene receive the command *Off* or *Stop*. If an element does not support either of the two commands, it is ignored.

bStoreCurrentScenery: The command affects functional units of the type of *illumination*, *sunblind* and *window* only if these are addressed within the scene by the command *Set control value directly* or *Set position directly*. If the *bStoreCurrentScenery* command is executed, then the current state will be used by the respective functional units as the new parameter for the command *Set control value directly* or *Set position directly*.

bPushButtonLightingFadeUpDown: This command affects only functional units of the type *illumination*. The scene is called by a short pulse (push of a button). If the signal is applied continuously, then the lamps of the functional unit are dimmed up and down.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
```

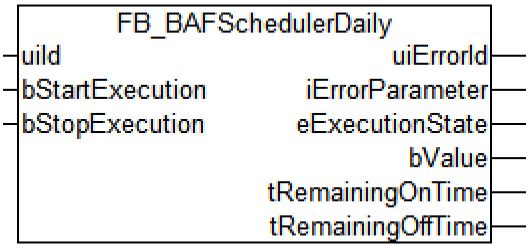
uiErrorId: contains the error code of the last-executed command (see error codes [▶ 276]).

iErrorParameter: contains the specific parameters of the error code (see error codes [▶ 276]).

eExecutionState: contain the current execution status of the instance of the scene.

8.10 Time Scheduler Channels

8.10.1 FB_BAFSchedulerDaily



Function-block for controlling a Daily Time Scheduler [▶ 153] directly out of the PLC-program

VAR_INPUT

```
uiId                  : UINT;
bStartExecution       : BOOL;
bStopExecution        : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the daily time scheduler.

bStopExecution: Stops the instance of the daily time scheduler.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
bValue                : BOOL;
tRamainingOnTime      : TIME;
tRamainingOffTime     : TIME;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

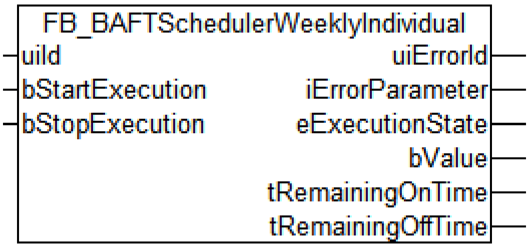
eExecutionState: Contains the actual execution state of the instance of the daily time scheduler.

bValue: The output state of the signal.

tRamainingOnTime: Shows, how long the scheduler channel will remain active.

tRamainingOffTime: Shows, how long the scheduler channel will remain deactivated.

8.10.2 FB_BAFSchedulerWeeklyIndividual



Function-block for controlling a Monthly Time Scheduler (individual weekday of the month) [▶ 155] directly out of the PLC-program.

VAR_INPUT

```
uiId                     : UINT;
bStartExecution          : BOOL;
bStopExecution           : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the monthly time scheduler.

bStopExecution: Stops the instance of the monthly time scheduler.

VAR_OUTPUT

```
uiErrorId                : UINT;
iErrorParameter          : INT;
eExecutionState          : E_BAF_ExecutionState;
bValue                   : INT;
tRamainingOnTime         : TIME;
tRamainingOffTime        : TIME;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

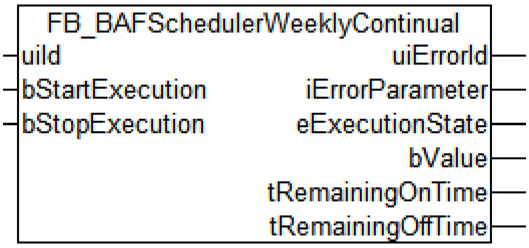
eExecutionState: Contains the actual execution state of the instance of the monthly time scheduler.

bValue: The output state of the signal.

tRamainingOnTime: Shows, how long the scheduler channel will remain active.

tRamainingOffTime: Shows, how long the scheduler channel will remain deactivated.

8.10.3 FB_BAFSchedulerWeeklyContinual



Function-block for controlling a Weekly Time Scheduler (continual days of the week) [▶ 157] directly out of the PLC-program.

VAR_INPUT

```
uiId                     : UINT;
bStartExecution          : BOOL;
bStopExecution           : BOOL;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the weekly time scheduler.

bStopExecution: Stops the instance of the weekly time scheduler.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
bValue                : INT;
tRamainingOnTime      : TIME;
tRamainingOffTime     : TIME;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

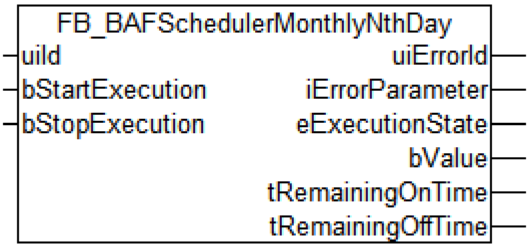
eExecutionState: Contains the actual execution state of the instance of the weekly time scheduler.

bValue: The output state of the signal.

tRamainingOnTime: Shows, how long the scheduler channel will remain active.

tRamainingOffTime: Shows, how long the scheduler channel will remain deactivated.

8.10.4 FB_BAFSchedulerMonthlyNthDay



Function-block for controlling a Monthly Time Scheduler (nth day of the month) [▸ 159] directly out of the PLC-program.

VAR_INPUT

```
uiId                  : UINT;
bStartExecution       : BOOL;
bStopExecution        : BOOL;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the monthly time scheduler.

bStopExecution: Stops the instance of the monthly time scheduler.

VAR_OUTPUT

```
uiErrorId             : UINT;
iErrorParameter       : INT;
eExecutionState       : E_BAF_ExecutionState;
bValue                : INT;
tRamainingOnTime      : TIME;
tRamainingOffTime     : TIME;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

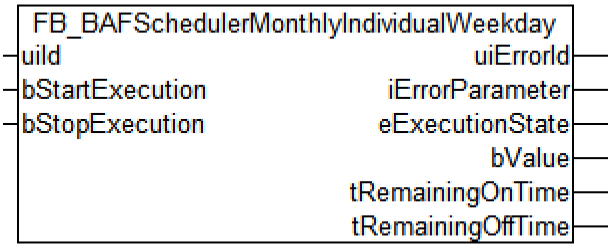
eExecutionState: Contains the actual execution state of the instance of the monthly time scheduler.

bValue: The output state of the signal.

tRamainingOnTime: Shows, how long the scheduler channel will remain active.

tRamainingOffTime: Shows, how long the scheduler channel will remain deactivated.

8.10.5 FB_BAFSchedulerMonthlyIndividualWeekday



Function-block for controlling a Monthly Time Scheduler (individual weekday of the month) [▸ 161] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
```

uiId: Object id of the instance (see object id [▸ 200]).

bStartExecution: Starts the instance of the monthly time scheduler.

bStopExecution: Stops the instance of the monthly time scheduler.

VAR_OUTPUT

```
uiErrorId               : UINT;
iErrorParameter         : INT;
eExecutionState         : E_BAF_ExecutionState;
bValue                  : INT;
tRamainingOnTime        : TIME;
tRamainingOffTime       : TIME;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution state of the instance of the monthly time scheduler.

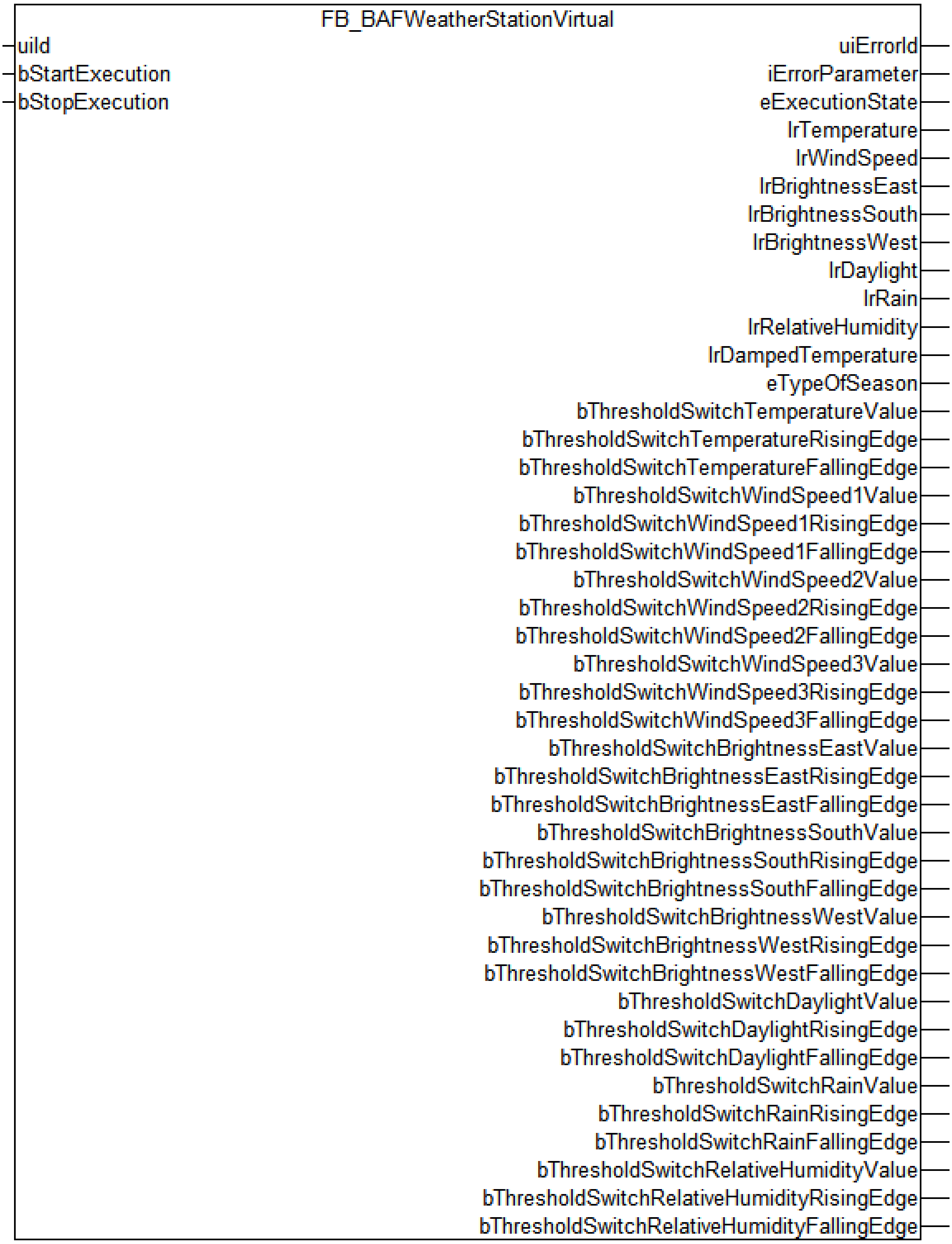
bValue: The output state of the signal.

tRamainingOnTime: Shows, how long the scheduler channel will remain active.

tRamainingOffTime: Shows, how long the scheduler channel will remain deactivated.

8.11 Weather Stations

8.11.1 FB_BAFWeatherStationVirtual


FB_BAFWeatherStationVirtual
uiId
bStartExecution
bStopExecution
uiErrorId
iErrorParameter
eExecutionState
lrTemperature
lrWindSpeed
lrBrightnessEast
lrBrightnessSouth
lrBrightnessWest
lrDaylight
lrRain
lrRelativeHumidity
lrDampedTemperature
eTypeOfSeason
bThresholdSwitchTemperatureValue
bThresholdSwitchTemperatureRisingEdge
bThresholdSwitchTemperatureFallingEdge
bThresholdSwitchWindSpeed1Value
bThresholdSwitchWindSpeed1RisingEdge
bThresholdSwitchWindSpeed1FallingEdge
bThresholdSwitchWindSpeed2Value
bThresholdSwitchWindSpeed2RisingEdge
bThresholdSwitchWindSpeed2FallingEdge
bThresholdSwitchWindSpeed3Value
bThresholdSwitchWindSpeed3RisingEdge
bThresholdSwitchWindSpeed3FallingEdge
bThresholdSwitchBrightnessEastValue
bThresholdSwitchBrightnessEastRisingEdge
bThresholdSwitchBrightnessEastFallingEdge
bThresholdSwitchBrightnessSouthValue
bThresholdSwitchBrightnessSouthRisingEdge
bThresholdSwitchBrightnessSouthFallingEdge
bThresholdSwitchBrightnessWestValue
bThresholdSwitchBrightnessWestRisingEdge
bThresholdSwitchBrightnessWestFallingEdge
bThresholdSwitchDaylightValue
bThresholdSwitchDaylightRisingEdge
bThresholdSwitchDaylightFallingEdge
bThresholdSwitchRainValue
bThresholdSwitchRainRisingEdge
bThresholdSwitchRainFallingEdge
bThresholdSwitchRelativeHumidityValue
bThresholdSwitchRelativeHumidityRisingEdge
bThresholdSwitchRelativeHumidityFallingEdge

Function-block for controlling a Virtual weather station [▶ 163] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the virtual weather station.

bStopExecution: Stops the instance of the virtual weather station.

VAR_OUTPUT

```
uiErrorId                                   : UINT;
iErrorParameter                             : INT;
eExecutionState                             : E_BAF_ExecutionState;
lrTemperature                               : LREAL;
lrWindSpeed                                 : LREAL;
lrBrightnessEast                            : LREAL;
lrBrightnessSouth                           : LREAL;
lrBrightnessWest                            : LREAL;
lrDaylight                                  : LREAL;
lrRain                                      : LREAL;
lrRelativeHumidity                          : LREAL;
lrDampedTemperature                         : LREAL;
eTypeOfSeason                               : E_BAF_WeatherStationTypeOfSeason;
bThresholdSwitchTemperatureValue            : BOOL;
bThresholdSwitchTemperatureRisingEdge       : BOOL;
bThresholdSwitchTemperatureFallingEdge      : BOOL;
bThresholdSwitchWindSpeed1Value             : BOOL;
bThresholdSwitchWindSpeed1RisingEdge        : BOOL;
bThresholdSwitchWindSpeed1FallingEdge       : BOOL;
bThresholdSwitchWindSpeed2Value             : BOOL;
bThresholdSwitchWindSpeed2RisingEdge        : BOOL;
bThresholdSwitchWindSpeed2FallingEdge       : BOOL;
bThresholdSwitchWindSpeed3Value             : BOOL;
bThresholdSwitchWindSpeed3RisingEdge        : BOOL;
bThresholdSwitchWindSpeed3FallingEdge       : BOOL;
bThresholdSwitchBrightnessEastValue         : BOOL;
bThresholdSwitchBrightnessEastRisingEdge    : BOOL;
bThresholdSwitchBrightnessEastFallingEdge   : BOOL;
bThresholdSwitchBrightnessSouthValue        : BOOL;
bThresholdSwitchBrightnessSouthRisingEdge   : BOOL;
bThresholdSwitchBrightnessSouthFallingEdge  : BOOL;
bThresholdSwitchBrightnessWestValue         : BOOL;
bThresholdSwitchBrightnessWestRisingEdge    : BOOL;
bThresholdSwitchBrightnessWestFallingEdge   : BOOL;
bThresholdSwitchDaylightValue               : BOOL;
bThresholdSwitchDaylightRisingEdge          : BOOL;
bThresholdSwitchDaylightFallingEdge         : BOOL;
bThresholdSwitchRainValue                   : BOOL;
bThresholdSwitchRainRisingEdge              : BOOL;
bThresholdSwitchRainFallingEdge             : BOOL;
bThresholdSwitchRelativeHumidityValue       : BOOL;
bThresholdSwitchRelativeHumidityRisingEdge  : BOOL;
bThresholdSwitchRelativeHumidityFallingEdge : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the virtual weather station.

lrTemperature: Actual outside temperature.

lrWindSpeed: Current wind speed.

lrBrightnessEast: Brightness on the east side.

lrBrightnessSouth: Brightness on the south side.

lrBrightnessWest: Brightness on the west side.

lrDaylight: Displays brightness of the day.

lrRain: Shows intensity of rainfall.

lrRelativeHumidity: Measured relative humidity.

lrDampedTemperature: Average of temperature measured during the day.

eTypeOfSeason: Displays the season depending on damped temperature.

bThresholdSwitchTemperatureValue: State of the threshold value switch for monitoring the outside temperature.

bThresholdSwitchTemperatureRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the outside temperature is activated.

bThresholdSwitchTemperatureFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the outside temperature is deactivated.

bThresholdSwitchWindSpeed1Value: State of the first threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed1RisingEdge: The output is set to TRUE for one PLC cycle, if the first threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed1FallingEdge: The output is set to TRUE for one PLC cycle, if the first threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed2Value: State of the second threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed2RisingEdge: The output is set to TRUE for one PLC cycle, if the second threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed2FallingEdge: The output is set to TRUE for one PLC cycle, if the second threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed3Value: State of the third threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed3RisingEdge: The output is set to TRUE for one PLC cycle, if the third threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed3FallingEdge: The output is set to TRUE for one PLC cycle, if the third threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchBrightnessEastValue: State of the threshold value switch for monitoring the light intensity on the east side.

bThresholdSwitchBrightnessEastRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the east side is activated.

bThresholdSwitchBrightnessEastFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the east side is deactivated.

bThresholdSwitchBrightnessSouthValue: State of the threshold value switch for monitoring the light intensity on the south side.

bThresholdSwitchBrightnessSouthRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the south side is activated.

bThresholdSwitchBrightnessSouthFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the south side is deactivated.

bThresholdSwitchBrightnessWestValue: State of the threshold value switch for monitoring the light intensity on the west side.

bThresholdSwitchBrightnessWestRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the west side is activated.

bThresholdSwitchBrightnessWestFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the west side is deactivated.

bThresholdSwitchDaylightValue: State of the threshold value switch for monitoring the daylight level.

bThresholdSwitchDaylightRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the daylight level is activated.

bThresholdSwitchDaylightFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the daylight level is deactivated.

bThresholdSwitchRainValue: State of the threshold value switch for monitoring rainfall.

bThresholdSwitchRainRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring rainfall is activated.

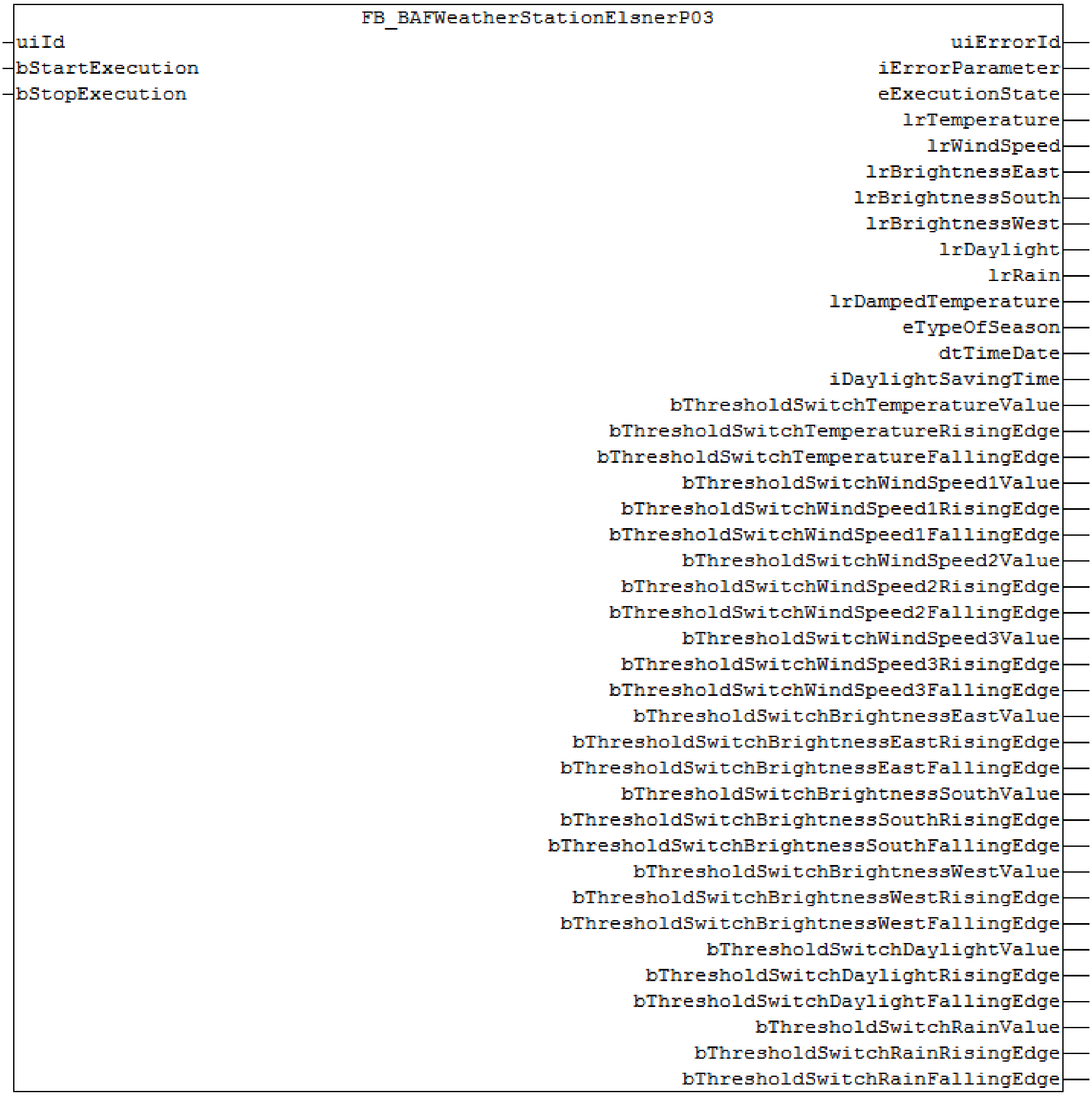
bThresholdSwitchRainFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring rainfall is deactivated.

bThresholdSwitchRelativeHumidityValue: State of the threshold value switch for monitoring the relative humidity.

bThresholdSwitchRelativeHumidityRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the relative humidity is activated.

bThresholdSwitchRelativeHumidityFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the relative humidity is deactivated.

8.11.2 FB_BAFWeatherStationElsnerP03


FB_BAFWeatherStationElsnerP03
uiId
bStartExecution
bStopExecution
uiErrorId
iErrorParameter
eExecutionState
lrTemperature
lrWindSpeed
lrBrightnessEast
lrBrightnessSouth
lrBrightnessWest
lrDaylight
lrRain
lrDampedTemperature
eTypeOfSeason
dtTimeDate
iDaylightSavingTime
bThresholdSwitchTemperatureValue
bThresholdSwitchTemperatureRisingEdge
bThresholdSwitchTemperatureFallingEdge
bThresholdSwitchWindSpeed1Value
bThresholdSwitchWindSpeed1RisingEdge
bThresholdSwitchWindSpeed1FallingEdge
bThresholdSwitchWindSpeed2Value
bThresholdSwitchWindSpeed2RisingEdge
bThresholdSwitchWindSpeed2FallingEdge
bThresholdSwitchWindSpeed3Value
bThresholdSwitchWindSpeed3RisingEdge
bThresholdSwitchWindSpeed3FallingEdge
bThresholdSwitchBrightnessEastValue
bThresholdSwitchBrightnessEastRisingEdge
bThresholdSwitchBrightnessEastFallingEdge
bThresholdSwitchBrightnessSouthValue
bThresholdSwitchBrightnessSouthRisingEdge
bThresholdSwitchBrightnessSouthFallingEdge
bThresholdSwitchBrightnessWestValue
bThresholdSwitchBrightnessWestRisingEdge
bThresholdSwitchBrightnessWestFallingEdge
bThresholdSwitchDaylightValue
bThresholdSwitchDaylightRisingEdge
bThresholdSwitchDaylightFallingEdge
bThresholdSwitchRainValue
bThresholdSwitchRainRisingEdge
bThresholdSwitchRainFallingEdge

Function block for controlling an 'Elsner P03' weather station [▶ 165] directly from the PLC program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
```

uiId: object ID of the instance (see also [▶ 200]).

bStartExecution: starts the instance of the ‘Elsner P03’ weather station.

bStopExecution: stops the instance of the ‘Elsner P03’ weather station.

VAR_OUTPUT

```
uiErrorId                                   : UINT;
iErrorParameter                             : INT;
eExecutionState                             : E_BAF_ExecutionState;
uiErrorId                                   : UINT;
iErrorParameter                             : INT;
eExecutionState                             : E_BAF_ExecutionState;
lrTemperature                               : LREAL;
lrWindSpeed                                 : LREAL;
lrBrightnessEast                            : LREAL;
lrBrightnessSouth                           : LREAL;
lrBrightnessWest                            : LREAL;
lrDaylight                                  : LREAL;
lrRain                                      : LREAL;
lrDampedTemperature                         : LREAL;
eTypeOfSeason                               : E_BAF_WeatherStationTypeOfSeason;
dtTimeDate                                  : DT;
iDaylightSavingTime                         : INT;
bThresholdSwitchTemperatureValue            : BOOL;
bThresholdSwitchTemperatureRisingEdge       : BOOL;
bThresholdSwitchTemperatureFallingEdge      : BOOL;
bThresholdSwitchWindSpeed1Value             : BOOL;
bThresholdSwitchWindSpeed1RisingEdge        : BOOL;
bThresholdSwitchWindSpeed1FallingEdge       : BOOL;
bThresholdSwitchWindSpeed2Value             : BOOL;
bThresholdSwitchWindSpeed2RisingEdge        : BOOL;
bThresholdSwitchWindSpeed2FallingEdge       : BOOL;
bThresholdSwitchWindSpeed3Value             : BOOL;
bThresholdSwitchWindSpeed3RisingEdge        : BOOL;
bThresholdSwitchWindSpeed3FallingEdge       : BOOL;
bThresholdSwitchBrightnessEastValue         : BOOL;
bThresholdSwitchBrightnessEastRisingEdge    : BOOL;
bThresholdSwitchBrightnessEastFallingEdge   : BOOL;
bThresholdSwitchBrightnessSouthValue        : BOOL;
bThresholdSwitchBrightnessSouthRisingEdge   : BOOL;
bThresholdSwitchBrightnessSouthFallingEdge  : BOOL;
bThresholdSwitchBrightnessWestValue         : BOOL;
bThresholdSwitchBrightnessWestRisingEdge    : BOOL;
bThresholdSwitchBrightnessWestFallingEdge   : BOOL;
bThresholdSwitchDaylightValue               : BOOL;
bThresholdSwitchDaylightRisingEdge          : BOOL;
bThresholdSwitchDaylightFallingEdge         : BOOL;
bThresholdSwitchRainValue                   : BOOL;
bThresholdSwitchRainRisingEdge              : BOOL;
bThresholdSwitchRainFallingEdge             : BOOL;
```

uiErrorId: contains the error code of the last-executed command (see error codes [▶ 276]).

iErrorParameter: contains the specific parameters of the error code (see error codes [▶ 276]).

eExecutionState: contains the current execution status of the instance of the 'Elsner P03' weather station.

lrTemperature: current outside temperature [°C].

lrWindSpeed: momentary wind speed [m/s].

lrBrightnessEast: brightness on the east side [klux].

lrBrightnessSouth: brightness on the south side [klux].

lrBrightnessWest: brightness on the west side [klux].

lrDaylight: indicates the daylight level [0-999 lux].

lrRain: is set to 1.0 as soon as precipitation is detected. Please note that with this weather station the precipitation can only adopt the values 0.0 (no rainfall) or 1.0 (rainfall).

lrDampedTemperature: average temperature measured over the course of the day [°C].

eTypeOfSeason: indicates the season in relation to the damped temperature.

dtTimeDate: Date and time received by the time signal transmitter DCF77.

iDaylightSavingTime: Indication whether summer or winter time prevails, received by the time signal transmitter DCF77 (0: undefined, 1: wintertime, 2: summertime).

bThresholdSwitchTemperatureValue: state of the threshold value switch for monitoring the outside temperature.

bThresholdSwitchTemperatureRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the outside temperature is activated.

bThresholdSwitchTemperatureFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the outside temperature is deactivated.

bThresholdSwitchWindSpeed1Value: state of the first threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed1RisingEdge: The output is set to TRUE for a PLC cycle, if the first threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed1FallingEdge: the output is set to TRUE for a PLC cycle, if the first threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed2Value: state of the second threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed2RisingEdge: The output is set to TRUE for a PLC cycle, if the second threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed2FallingEdge: the output is set to TRUE for a PLC cycle, if the second threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed3Value: state of the third threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed3RisingEdge: the output is set to TRUE for a PLC cycle, if the third threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed3FallingEdge: the output is set to TRUE for a PLC cycle, if the third threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchBrightnessEastValue: state of the threshold value switch for monitoring the brightness on the east side.

bThresholdSwitchBrightnessEastRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the east side is activated.

bThresholdSwitchBrightnessEastFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the east side is deactivated.

bThresholdSwitchBrightnessSouthValue: State of the threshold value switch for monitoring the brightness on the south side.

bThresholdSwitchBrightnessSouthRisingEdge: the output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the brightness on the south side is activated.

bThresholdSwitchBrightnessSouthFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the south side is deactivated.

bThresholdSwitchBrightnessWestValue: state of the threshold value switch for monitoring the brightness on the west side.

bThresholdSwitchBrightnessWestRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the west side is activated.

bThresholdSwitchBrightnessWestFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the west side is deactivated.

bThresholdSwitchDaylightValue: state of the threshold value switch for monitoring the daylight level.

bThresholdSwitchDaylightRisingEdge: the output is set to true for a PLC cycle if the threshold switch for monitoring the daylight level switches on.

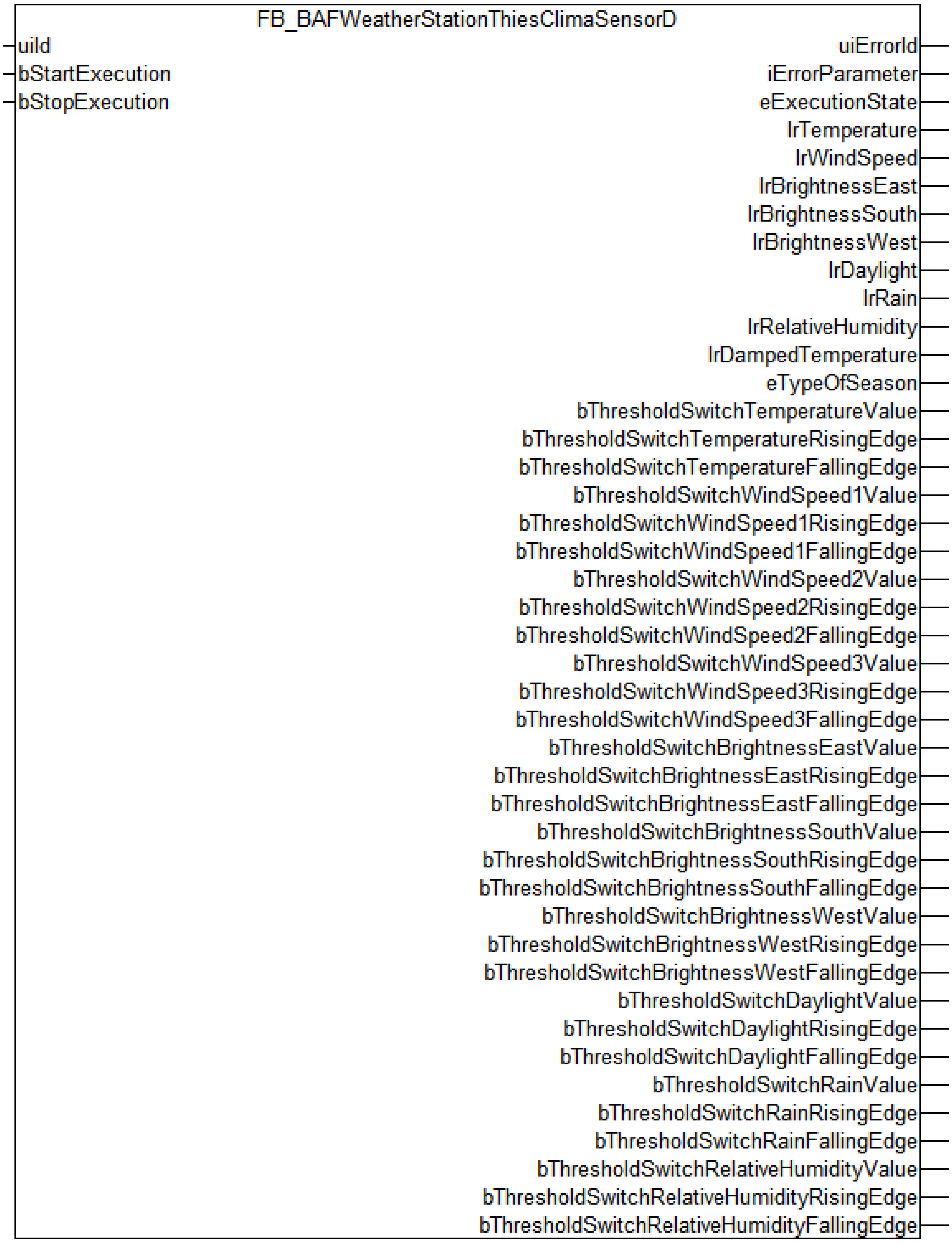
bThresholdSwitchDaylightFallingEdge: the output is set to true for a PLC cycle if the threshold switch for monitoring the daylight level switches off.

bThresholdSwitchRainValue: state of the threshold value switch for monitoring precipitation.

bThresholdSwitchRainRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring precipitation is activated.

bThresholdSwitchRainFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring precipitation is deactivated.

8.11.3 FB_BAFWeatherStationThiesClimaSensorD


FB_BAFWeatherStationThiesClimaSensorD
uiId
bStartExecution
bStopExecution
uiErrorId
iErrorParameter
eExecutionState
lrTemperature
lrWindSpeed
lrBrightnessEast
lrBrightnessSouth
lrBrightnessWest
lrDaylight
lrRain
lrRelativeHumidity
lrDampedTemperature
eTypeOfSeason
bThresholdSwitchTemperatureValue
bThresholdSwitchTemperatureRisingEdge
bThresholdSwitchTemperatureFallingEdge
bThresholdSwitchWindSpeed1Value
bThresholdSwitchWindSpeed1RisingEdge
bThresholdSwitchWindSpeed1FallingEdge
bThresholdSwitchWindSpeed2Value
bThresholdSwitchWindSpeed2RisingEdge
bThresholdSwitchWindSpeed2FallingEdge
bThresholdSwitchWindSpeed3Value
bThresholdSwitchWindSpeed3RisingEdge
bThresholdSwitchWindSpeed3FallingEdge
bThresholdSwitchBrightnessEastValue
bThresholdSwitchBrightnessEastRisingEdge
bThresholdSwitchBrightnessEastFallingEdge
bThresholdSwitchBrightnessSouthValue
bThresholdSwitchBrightnessSouthRisingEdge
bThresholdSwitchBrightnessSouthFallingEdge
bThresholdSwitchBrightnessWestValue
bThresholdSwitchBrightnessWestRisingEdge
bThresholdSwitchBrightnessWestFallingEdge
bThresholdSwitchDaylightValue
bThresholdSwitchDaylightRisingEdge
bThresholdSwitchDaylightFallingEdge
bThresholdSwitchRainValue
bThresholdSwitchRainRisingEdge
bThresholdSwitchRainFallingEdge
bThresholdSwitchRelativeHumidityValue
bThresholdSwitchRelativeHumidityRisingEdge
bThresholdSwitchRelativeHumidityFallingEdge

Function block for controlling a 'Thies Clima Sensor D' weather station [▶ 168] directly from the PLC program.

VAR_INPUT

```
uiId                     : UINT;
bStartExecution          : BOOL;
bStopExecution           : BOOL;
```

uiId: object ID of the instance (see also [▶ 200]).

bStartExecution: starts the instance of the ‘Thies Clima Sensor D’ weather station.

bStopExecution: stops the instance of the ‘Thies Clima Sensor D’ weather station.

VAR_OUTPUT

```
uiErrorId                                   : UINT;
iErrorParameter                             : INT;
eExecutionState                             : E_BAF_ExecutionState;
lrWindSpeed                                 : LREAL;
lrBrightnessEast                            : LREAL;
lrBrightnessSouth                           : LREAL;
lrBrightnessWest                            : LREAL;
lrDaylight                                  : LREAL;
lrRain                                      : LREAL;
lrRelativeHumidity                          : LREAL;
lrDampedTemperature                         : LREAL;
eTypeOfSeason                               : E_BAF_WeatherStationTypeOfSeason;
bThresholdSwitchTemperatureValue            : BOOL;
bThresholdSwitchTemperatureRisingEdge       : BOOL;
bThresholdSwitchTemperatureFallingEdge      : BOOL;
bThresholdSwitchWindSpeed1Value             : BOOL;
bThresholdSwitchWindSpeed1RisingEdge        : BOOL;
bThresholdSwitchWindSpeed1FallingEdge       : BOOL;
bThresholdSwitchWindSpeed2Value             : BOOL;
bThresholdSwitchWindSpeed2RisingEdge        : BOOL;
bThresholdSwitchWindSpeed2FallingEdge       : BOOL;
bThresholdSwitchWindSpeed3Value             : BOOL;
bThresholdSwitchWindSpeed3RisingEdge        : BOOL;
bThresholdSwitchWindSpeed3FallingEdge       : BOOL;
bThresholdSwitchBrightnessEastValue         : BOOL;
bThresholdSwitchBrightnessEastRisingEdge    : BOOL;
bThresholdSwitchBrightnessEastFallingEdge   : BOOL;
bThresholdSwitchBrightnessSouthValue        : BOOL;
bThresholdSwitchBrightnessSouthRisingEdge   : BOOL;
bThresholdSwitchBrightnessSouthFallingEdge  : BOOL;
bThresholdSwitchBrightnessWestValue         : BOOL;
bThresholdSwitchBrightnessWestRisingEdge    : BOOL;
bThresholdSwitchBrightnessWestFallingEdge   : BOOL;
bThresholdSwitchDaylightValue               : BOOL;
bThresholdSwitchDaylightRisingEdge          : BOOL;
bThresholdSwitchDaylightFallingEdge         : BOOL;
bThresholdSwitchRainValue                   : BOOL;
bThresholdSwitchRainRisingEdge              : BOOL;
bThresholdSwitchRainFallingEdge             : BOOL;
bThresholdSwitchRelativeHumidityValue       : BOOL;
bThresholdSwitchRelativeHumidityRisingEdge  : BOOL;
bThresholdSwitchRelativeHumidityFallingEdge : BOOL;
```

uiErrorId: contains the error code of the last-executed command (see error codes [▸ 276]).

iErrorParameter: contains the specific parameters of the error code (see error codes [▸ 276]).

eExecutionState: contains the current execution status of the instance of the 'Thies Clima Sensor D' weather station.

lrTemperature: current outside temperature [°C].

lrWindSpeed: momentary wind speed [m/s].

lrBrightnessEast: brightness on the east side [klux].

lrBrightnessSouth: brightness on the south side [klux].

lrBrightnessWest: brightness on the west side [klux].

lrDaylight: indicates the daylight level [lux].

lr**Rain:** is set to 1.0 as soon as precipitation is detected. Please note that with this weather station the precipitation can only adopt the values 0.0 (no rainfall) or 1.0 (rainfall).

lrRelativeHumidity: measured relative air humidity [%].

lrDampedTemperature: average temperature measured over the course of the day [°C].

eTypeOfSeason: indicates the season in relation to the damped temperature.

bThresholdSwitchTemperatureValue: state of the threshold value switch for monitoring the outside temperature.

bThresholdSwitchTemperatureRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the outside temperature is activated.

bThresholdSwitchTemperatureFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the outside temperature is deactivated.

bThresholdSwitchWindSpeed1Value: state of the first threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed1RisingEdge: The output is set to TRUE for a PLC cycle, if the first threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed1FallingEdge: the output is set to TRUE for a PLC cycle, if the first threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed2Value: state of the second threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed2RisingEdge: The output is set to TRUE for a PLC cycle, if the second threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed2FallingEdge: the output is set to TRUE for a PLC cycle, if the second threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed3Value: state of the third threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed3RisingEdge: the output is set to TRUE for a PLC cycle, if the third threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed3FallingEdge: the output is set to TRUE for a PLC cycle, if the third threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchBrightnessEastValue: state of the threshold value switch for monitoring the brightness on the east side.

bThresholdSwitchBrightnessEastRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the east side is activated.

bThresholdSwitchBrightnessEastFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the east side is deactivated.

bThresholdSwitchBrightnessSouthValue: state of the threshold value switch for monitoring the brightness on the south side.

bThresholdSwitchBrightnessSouthRisingEdge: the output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the brightness on the south side is activated.

bThresholdSwitchBrightnessSouthFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the south side is deactivated.

bThresholdSwitchBrightnessWestValue: state of the threshold value switch for monitoring the brightness on the west side.

bThresholdSwitchBrightnessWestRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the west side is activated.

bThresholdSwitchBrightnessWestFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the brightness on the west side is deactivated.

bThresholdSwitchDaylightValue: state of the threshold value switch for monitoring the daylight level.

bThresholdSwitchDaylightRisingEdge: the output is set to true for a PLC cycle if the threshold switch for monitoring the daylight level switches on.

bThresholdSwitchDaylightFallingEdge: the output is set to true for a PLC cycle if the threshold switch for monitoring the daylight level switches off.

bThresholdSwitchRainValue: state of the threshold value switch for monitoring precipitation.

bThresholdSwitchRainRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring precipitation is activated.

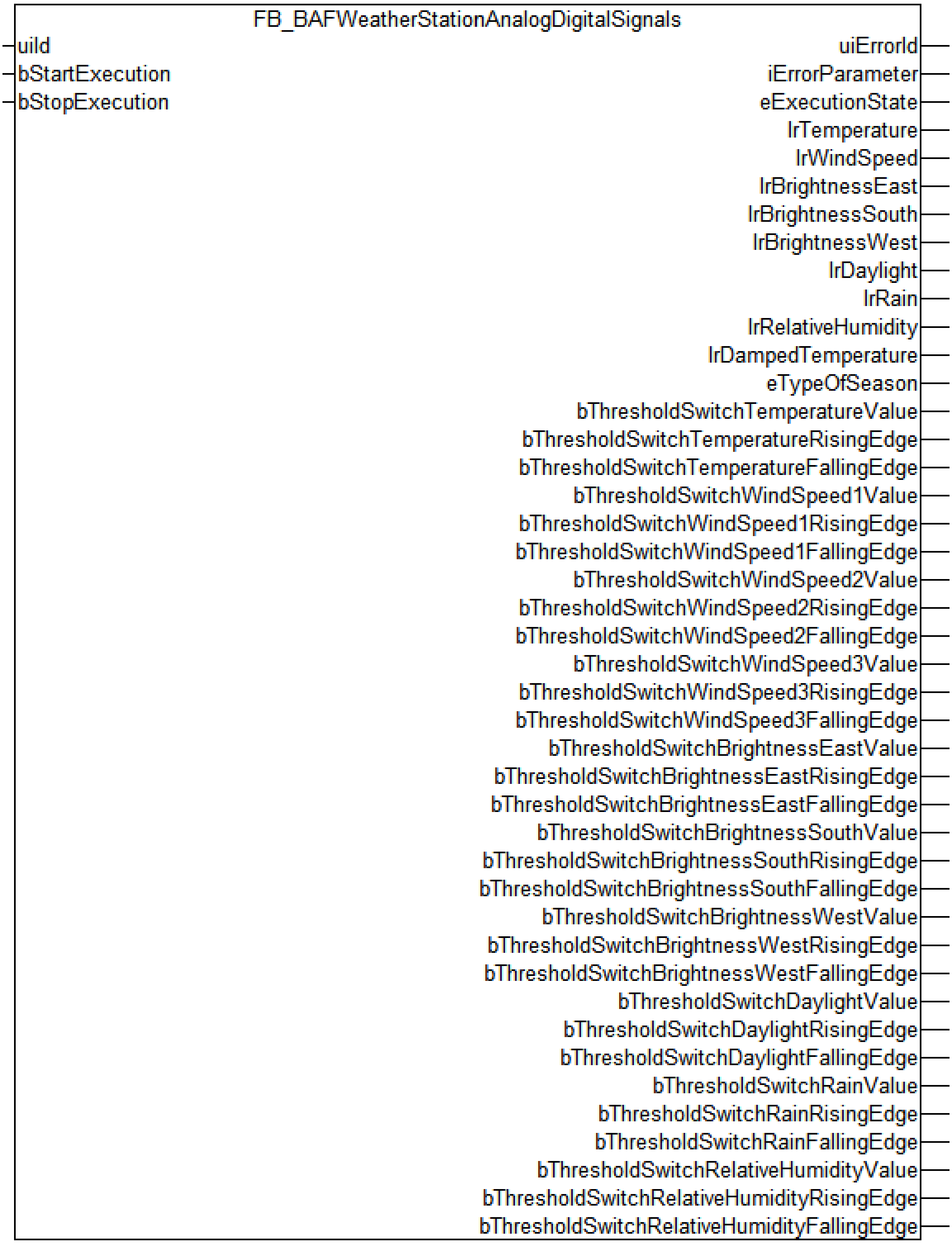
bThresholdSwitchRainFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring precipitation is deactivated.

bThresholdSwitchRelativeHumidityValue: state of the threshold value switch for monitoring the relative air humidity.

bThresholdSwitchRelativeHumidityRisingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the relative air humidity is activated.

bThresholdSwitchRelativeHumidityFallingEdge: the output is set to TRUE for a PLC cycle, if the threshold value switch for monitoring the relative air humidity is deactivated.

8.11.4 FB_BAFWeatherStationAnalogDigitalSignals


FB_BAFWeatherStationAnalogDigitalSignals
uiId
bStartExecution
bStopExecution
uiErrorId
iErrorParameter
eExecutionState
lrTemperature
lrWindSpeed
lrBrightnessEast
lrBrightnessSouth
lrBrightnessWest
lrDaylight
lrRain
lrRelativeHumidity
lrDampedTemperature
eTypeOfSeason
bThresholdSwitchTemperatureValue
bThresholdSwitchTemperatureRisingEdge
bThresholdSwitchTemperatureFallingEdge
bThresholdSwitchWindSpeed1Value
bThresholdSwitchWindSpeed1RisingEdge
bThresholdSwitchWindSpeed1FallingEdge
bThresholdSwitchWindSpeed2Value
bThresholdSwitchWindSpeed2RisingEdge
bThresholdSwitchWindSpeed2FallingEdge
bThresholdSwitchWindSpeed3Value
bThresholdSwitchWindSpeed3RisingEdge
bThresholdSwitchWindSpeed3FallingEdge
bThresholdSwitchBrightnessEastValue
bThresholdSwitchBrightnessEastRisingEdge
bThresholdSwitchBrightnessEastFallingEdge
bThresholdSwitchBrightnessSouthValue
bThresholdSwitchBrightnessSouthRisingEdge
bThresholdSwitchBrightnessSouthFallingEdge
bThresholdSwitchBrightnessWestValue
bThresholdSwitchBrightnessWestRisingEdge
bThresholdSwitchBrightnessWestFallingEdge
bThresholdSwitchDaylightValue
bThresholdSwitchDaylightRisingEdge
bThresholdSwitchDaylightFallingEdge
bThresholdSwitchRainValue
bThresholdSwitchRainRisingEdge
bThresholdSwitchRainFallingEdge
bThresholdSwitchRelativeHumidityValue
bThresholdSwitchRelativeHumidityRisingEdge
bThresholdSwitchRelativeHumidityFallingEdge

Function-block for controlling a Weather station with analog / digital signals [▶ 170] directly out of the PLC-program.

VAR_INPUT

```
uiId                    : UINT;
bStartExecution         : BOOL;
bStopExecution          : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the analog / digital weather station.

bStopExecution: Stops the instance of the analog / digital weather station.

VAR_OUTPUT

```
uiErrorId                                   : UINT;
iErrorParameter                             : INT;
eExecutionState                             : E_BAF_ExecutionState;
lrTemperature                               : LREAL;
lrWindSpeed                                 : LREAL;
lrBrightnessEast                            : LREAL;
lrBrightnessSouth                           : LREAL;
lrBrightnessWest                            : LREAL;
lrDaylight                                  : LREAL;
lrRain                                      : LREAL;
lrRelativeHumidity                          : LREAL;
lrDampedTemperature                         : LREAL;
eTypeOfSeason                               : E_BAF_WeatherStationTypeOfSeason;
bThresholdSwitchTemperatureValue            : BOOL;
bThresholdSwitchTemperatureRisingEdge       : BOOL;
bThresholdSwitchTemperatureFallingEdge      : BOOL;
bThresholdSwitchWindSpeed1Value             : BOOL;
bThresholdSwitchWindSpeed1RisingEdge        : BOOL;
bThresholdSwitchWindSpeed1FallingEdge       : BOOL;
bThresholdSwitchWindSpeed2Value             : BOOL;
bThresholdSwitchWindSpeed2RisingEdge        : BOOL;
bThresholdSwitchWindSpeed2FallingEdge       : BOOL;
bThresholdSwitchWindSpeed3Value             : BOOL;
bThresholdSwitchWindSpeed3RisingEdge        : BOOL;
bThresholdSwitchWindSpeed3FallingEdge       : BOOL;
bThresholdSwitchBrightnessEastValue         : BOOL;
bThresholdSwitchBrightnessEastRisingEdge    : BOOL;
bThresholdSwitchBrightnessEastFallingEdge   : BOOL;
bThresholdSwitchBrightnessSouthValue        : BOOL;
bThresholdSwitchBrightnessSouthRisingEdge   : BOOL;
bThresholdSwitchBrightnessSouthFallingEdge  : BOOL;
bThresholdSwitchBrightnessWestValue         : BOOL;
bThresholdSwitchBrightnessWestRisingEdge    : BOOL;
bThresholdSwitchBrightnessWestFallingEdge   : BOOL;
bThresholdSwitchDaylightValue               : BOOL;
bThresholdSwitchDaylightRisingEdge          : BOOL;
bThresholdSwitchDaylightFallingEdge         : BOOL;
bThresholdSwitchRainValue                   : BOOL;
bThresholdSwitchRainRisingEdge              : BOOL;
bThresholdSwitchRainFallingEdge             : BOOL;
bThresholdSwitchRelativeHumidityValue       : BOOL;
bThresholdSwitchRelativeHumidityRisingEdge  : BOOL;
bThresholdSwitchRelativeHumidityFallingEdge : BOOL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the analog / digital weather station.

lrTemperature: Actual outside temperature.

lrWindSpeed: Current wind speed.

lrBrightnessEast: Brightness on the east side.

lrBrightnessSouth: Brightness on the south side.

lrBrightnessWest: Brightness on the west side.

lrDaylight: Displays brightness of the day.

lrRain: Shows intensity of rainfall.

lrRelativeHumidity: Measured relative humidity.

lrDampedTemperature: Average of temperature measured during the day.

eTypeOfSeason: Displays the season depending on damped temperature.

bThresholdSwitchTemperatureValue: State of the threshold value switch for monitoring the outside temperature.

bThresholdSwitchTemperatureRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the outside temperature is activated.

bThresholdSwitchTemperatureFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the outside temperature is deactivated.

bThresholdSwitchWindSpeed1Value: State of the first threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed1RisingEdge: The output is set to TRUE for one PLC cycle, if the first threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed1FallingEdge: The output is set to TRUE for one PLC cycle, if the first threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed2Value: State of the second threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed2RisingEdge: The output is set to TRUE for one PLC cycle, if the second threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed2FallingEdge: The output is set to TRUE for one PLC cycle, if the second threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchWindSpeed3Value: State of the third threshold value switch for monitoring the wind speed.

bThresholdSwitchWindSpeed3RisingEdge: The output is set to TRUE for one PLC cycle, if the third threshold value switch for monitoring the wind speed is activated.

bThresholdSwitchWindSpeed3FallingEdge: The output is set to TRUE for one PLC cycle, if the third threshold value switch for monitoring the wind speed is deactivated.

bThresholdSwitchBrightnessEastValue: State of the threshold value switch for monitoring the light intensity on the east side.

bThresholdSwitchBrightnessEastRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the east side is activated.

bThresholdSwitchBrightnessEastFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the east side is deactivated.

bThresholdSwitchBrightnessSouthValue: State of the threshold value switch for monitoring the light intensity on the south side.

bThresholdSwitchBrightnessSouthRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the south side is activated.

bThresholdSwitchBrightnessSouthFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the south side is deactivated.

bThresholdSwitchBrightnessWestValue: State of the threshold value switch for monitoring the light intensity on the west side.

bThresholdSwitchBrightnessWestRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the west side is activated.

bThresholdSwitchBrightnessWestFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the light intensity on the west side is deactivated.

bThresholdSwitchDaylightValue: State of the threshold value switch for monitoring the daylight level.

bThresholdSwitchDaylightRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the daylight level is activated.

bThresholdSwitchDaylightFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the daylight level is deactivated.

bThresholdSwitchRainValue: State of the threshold value switch for monitoring rainfall.

bThresholdSwitchRainRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring rainfall is activated.

bThresholdSwitchRainFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring rainfall is deactivated.

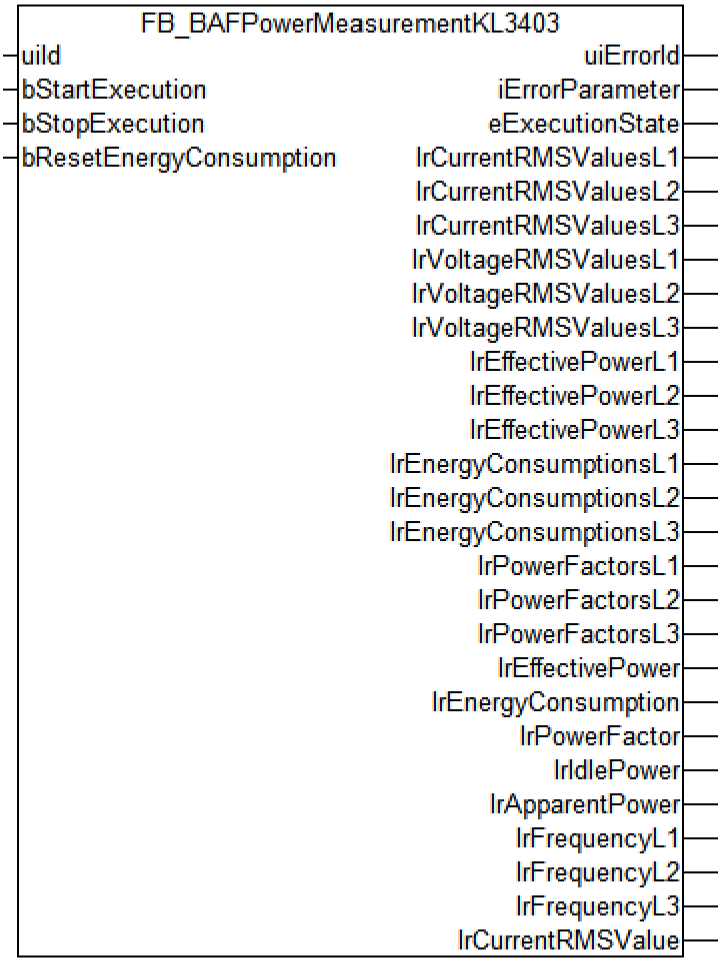
bThresholdSwitchRelativeHumidityValue: State of the threshold value switch for monitoring the relative humidity.

bThresholdSwitchRelativeHumidityRisingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the relative humidity is activated.

bThresholdSwitchRelativeHumidityFallingEdge: The output is set to TRUE for one PLC cycle, if the threshold value switch for monitoring the relative humidity is deactivated.

8.12 Subsystems

8.12.1 FB_BAFPowerMeasurementKL3403



Function-block for controlling a Power Measurement [▶ 188] directly out of the PLC-program.

VAR_INPUT

```
uiId                      : UINT;
bStartExecution           : BOOL;
bStopExecution            : BOOL;
bResetEnergyConsumption   : BOOL;
```

uiId: Object id of the instance (see object id [▶ 200]).

bStartExecution: Starts the instance of the power measurement.

bStopExecution: Stops the instance of the power measurement.

bResetEnergyConsumption: Resets the energy consumption values.

VAR_OUTPUT

```
uiErrorId                : UINT;
iErrorParameter          : INT;
eExecutionState          : E_BAF_ExecutionState;
lrCurrentRMSValuesL1     : LREAL;
lrCurrentRMSValuesL2     : LREAL;
lrCurrentRMSValuesL3     : LREAL;
lrVoltageRMSValuesL1     : LREAL;
lrVoltageRMSValuesL2     : LREAL;
lrVoltageRMSValuesL3     : LREAL;
lrEffectivePowerL1       : LREAL;
lrEffectivePowerL2       : LREAL;
lrEffectivePowerL3       : LREAL;
lrEnergyConsumptionsL1   : LREAL;
lrEnergyConsumptionsL2   : LREAL;
lrEnergyConsumptionsL3   : LREAL;
lrPowerFactorsL1         : LREAL;
lrPowerFactorsL2         : LREAL;
lrPowerFactorsL3         : LREAL;
lrEffectivePower         : LREAL;
lrEnergyConsumption      : LREAL;
lrPowerFactor            : LREAL;
lrIdlePower              : LREAL;
lrApparentPower          : LREAL;
lrFrequencyL1            : LREAL;
lrFrequencyL2            : LREAL;
lrFrequencyL3            : LREAL;
lrCurrentRMSValue        : LREAL;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▶ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▶ 276]).

eExecutionState: Contains the actual execution state of the instance of the power measurement.

lrCurrentRMSValuesL1: Actual current of phase L1.

lrCurrentRMSValuesL2: Actual current of phase L2.

lrCurrentRMSValuesL3: Actual current of phase L3.

lrVoltageRMSValuesL1: Voltage value at phase L1.

lrVoltageRMSValuesL2: Voltage value at phase L2.

lrVoltageRMSValuesL3: Voltage value at phase L3.

lrEffectivePowerL1: Effective power of phase L1.

lrEffectivePowerL2: Effective power of phase L2.

lrEffectivePowerL3: Effective power of phase L3.

lrEnergyConsumptionsL1: Energy consumption at phase L1.

lrEnergyConsumptionsL2: Energy consumption at phase L2.

lrEnergyConsumptionsL3: Energy consumption at phase L3.

lrPowerFactorsL1: Power factor of phase L1.

lrPowerFactorsL2: Power factor of phase L2.

lrPowerFactorsL3: Power factor of phase L3.

lrEffectivePower: Overall power of all phases.

lrEnergyConsumption: Energy consumption of all phases.

lrPowerFactor: Power factor of all phases.

lrIdlePower: In sum idle power.

lrApparentPower: In sum apparent power.

lrFrequencyL1: Frequency at phase L1.

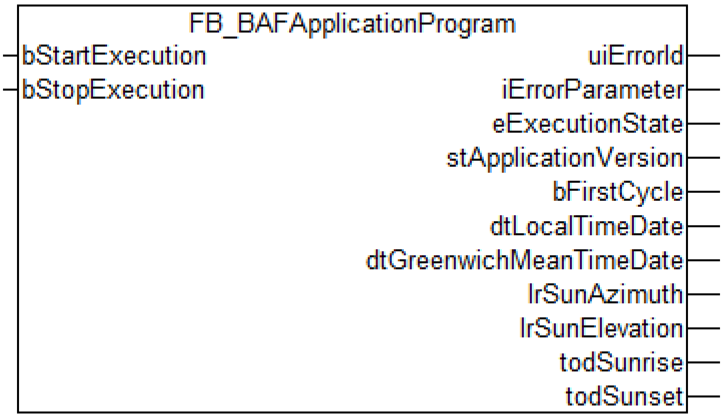
lrFrequencyL2: Frequency at phase L2.

lrFrequencyL3: Frequency at phase L3.

lrCurrentRMSValue: In sum current.

8.13 Application Program

8.13.1 FB_BAFApplicationProgram



Function-block to start or stop the Application Program [▸ 56]. In addition the block provides further, general information to complement the PLC program.

VAR_INPUT

```
bStartExecution             : BOOL;
bStopExecution              : BOOL;
```

bStartExecution: Starts the application.

bStopExecution: Stops the application.

VAR_OUTPUT

```
uiErrorId                   : UINT;
iErrorParameter             : INT;
eExecutionState             : E_BAF_ExecutionState;
stApplicationVersion        : ST_BAF_ApplicationVersion;
bFirstCycle                 : BOOL;
dtLocalTimeDate             : DT;
dtGreenwichMeanTimeDate     : DT;
lrSunAzimuth                : LREAL;
lrSunElevation              : LREAL;
todSunrise                  : TOD;
todSunset                   : TOD;
```

uiErrorId: Contains the error code of the last executed command (see Error codes [▸ 276]).

iErrorParameter: Contains the specific parameter of the error code (see Error codes [▸ 276]).

eExecutionState: Contains the actual execution state of the application.

stApplicationVersion: Contains the version-number of the application.

bFirstCycle: If the application program switches to the execution state *Running*, this output is set for one PLC cycle.

dtLocalTimeDate: The local time.

dtGreenwichMeanTimeDate: Greenwich-time (GMT).

lrSunAzimuth: Sun-direction (0° North / 90° East / 180° South / 270° West).

lrSunElevation: Sun-elevation (0° horizontal / 90° vertical).

todSunrise: the calculated sunrise time. For correct calculation the longitude and latitude must be specified (see *Project* --> *Settings* --> Location).

todSunset: the calculated sunrise time.

8.14 Appendix

8.14.1 Data types

ST_BAF_ApplicationVersion

```
TYPE ST_BAF_ApplicationVersion:
STRUCT
  usiMajor     : USINT;
  usiMinor     : USINT;
  uiBuild      : UINT;
END_STRUCT
END_TYPE
```

E_BAF_ExecutionState

```
TYPE E_BAF_ExecutionState :
(
  eBAFExecutionStateInvalid := 0,
  eBAFExecutionStateStarting,
  eBAFExecutionStateLoadingConfiguration,
  eBAFExecutionStateConfigurationLoaded,
  eBAFExecutionStateCheckingConfiguration,
  eBAFExecutionStateConfigurationChecked,
  eBAFExecutionStateInitializing,
  eBAFExecutionStateInitialized,
  eBAFExecutionStateRunning,
  eBAFExecutionStateStopping,
  eBAFExecutionStateStopped
);
END_TYPE
```

E_BAF_SuntrackingOperatingMode

```
TYPE E_BAF_SuntrackingOperatingMode :
(
  eBAFSuntrackingOperatingModeNormal := 0,
  eBAFSuntrackingOperatingModeCalibrating := 1,
  eBAFSuntrackingOperatingModeReversing := 2,
  eBAFSuntrackingOperatingModeAngleForMaximumShading := 3,
  eBAFSuntrackingOperatingModeAngleForMaximumIllumination := 4,
  eBAFSuntrackingOperatingModeCompletelyClosed := 5,
  eBAFSuntrackingOperatingModeBlindHorizontal := 6,
  eBAFSuntrackingOperatingModeDriveUp := 7,
  eBAFSuntrackingOperatingModeOutOfOperatingRange := 10,
  eBAFSuntrackingOperatingModeTheSunHasSet := 11
);
END_TYPE
```

E_BAF_SuntrackingControlMode

```
TYPE E_BAF_SuntrackingControlMode:
(
  eBAFSuntrackingControlModeManual := 0,
  eBAFSuntrackingControlModeAutomatic := 1
);
END_TYPE
```

E_BAF_HVACEnergyLevel

```
TYPE E_BAF_HVACEnergyLevel :
(
  eBAFHVACEnergyLevelProtection := 1,
  eBAFHVACEnergyLevelEconomy := 2,
  eBAFHVACEnergyLevelPreComfort := 3,
  eBAFHVACEnergyLevelComfort := 4
);
END_TYPE
```

E_BAF_HVACControlFunction

```
TYPE E_BAF_HVACControlFunction :
(
  eBAFHVACControlFunctionOff := 0,
  eBAFHVACControlFunctionHeating := 1,
  eBAFHVACControlFunctionCooling := 2
);
END_TYPE
```

E_BAF_WeatherStationTypeOfSeason

```
TYPE E_BAF_WeatherStationTypeOfSeason :
(
  eBAFWeatherStationTypeOfSeasonUndefined := 0,
  eBAFWeatherStationTypeOfSeasonSummer := 1,
  eBAFWeatherStationTypeOfSeasonWinter := 2
);
END_TYPE
```

8.14.2 Error codes

Hex	Dec	Description	Parameter
0x0000	0	No error	-
0x8001	32769	The PLC function block is sending commands to an object, which is not in *running* state.	-
0x8002	32770	The id is out of range.	-
0x8003	32771	The type of the object is not compatible to the PLC function block.	-
0x8004	32772	Status byte of a terminal channel is not 0.	-
0x8005	32773	No data received yet.	-
0x8006	32774	Watchdog error.	-
0x8007	32775	Wrong boot configuration version.	-
0x8008	32776	Error while opening boot configuration file of the scenes on the target.	error id of *FB_FileOpen()*
0x8009	32777	Error while reading boot configuration file of the scenes on the target	error id of *FB_FileRead()*
0x800A	32778	Error while writing boot configuration file of the scenes on the target.	error id of *FB_FileWrite()*
0x800B	32779	Error while closing boot configuration file of the scenes on the target.	error id of *FB_FileClose()*
0x800C	32780	Error while opening boot configuration file on the target.	error id of *FB_FileOpen()*
0x800D	32781	Error while reading boot configuration file on the target.	error id of *FB_FileRead()*
0x800E	32782	Error while writing boot configuration file on the target.	error id of *FB_FileWrite()*
0x800F	32783	Error while closing boot configuration file on the target.	error id of *FB_FileClose()*
0x8015	32789	The parameters of the sunblind object are invalid.	1: the id of *BlindDriveGroup* is out of range. 2: the id of *DigitalSignalGroupUp* is out of range. 3: the id of *DigitalSignalGroupDown* is out of range. 4: the id of *WeatherStation* is out of range. 5: the id of *RoomTemperatureSignalGroup* is out of range. 6: the id of *MotionDetectionDigitalSignalGroup* is out of range. 7: the id of *DigitalSignalGroupAutomaticOff* is out of range. 8: the id of *DigitalSignalGroupUpDown* is out of range.

Hex	**Dec**	**Description**	**Parameter**
0x8016	32790	The parameters of the lamp object are invalid.	1: *MinControlValue* is bigger or equal *MaxControlValue.* 2: *FadeSteps* is bigger or equal (*MaxControlValue* - *MinControlValue*). 3: the id of *DALILine* is out of range. 4: *DALIShortAddress* is out of range.

Hex	Dec	Description	Parameter
0x8017	32791	The parameters of the lighting object are invalid.	1: the id of *ActualValueSignalGroup* is out of range. 2: the id of *LampGroup* is out of range. 3: the id of *DigitalSignalGroupDimmUp* is out of range. 4: the id of *DigitalSignalGroupDimmDown* is out of range. 5: the id of *DigitalSignalGroupDimmUpDown* is out of range. 6: the id of *DigitalSignalGroupMotionDetection* is out of range. 7: the id of *DigitalSignalGroupMotionDetection* is 0. 8: *PresenceValue* is below 0. 9: *ProlongValue* is below 0. 10: the id of *ActualValueSignalGroupId* is out of range. 11: the id of *ActualValueSignalGroupId* is 0. 12: calculated lower point of hysteresis (*BasicSetpointValue* + *SetpointValueShift* - *ControlHysteresis* / 2) is below 0. 13: *SwitchOnLevel* is below 0. 14: *SetpointValueShiftStepSize* is below 0. 15: *MaximalSetpointValueShift* is below *MinimalSetpointValueShift.* 16: the id of *LampGroupSub1Id* is out of range. 17: the id of *LampGroupSub2Id* is out of range. 18: *FactorControlValueSubLampGroup1* is below 0. 19: *FactorControlValueSubLampGroup2* is below 0. 20: *MaxControlDelay* is below the minimum of t#1s. 21: *ControlHysteresis* is below 0. 22: *ControlOnUpperLimit* is below or equal to 0. 23: the id of *WeatherStation* is out of range. 24: the id of *WeatherStation* is 0.

Hex	Dec	Description	Parameter
0x8018	32792	The parameters of the HVAC object are invalid.	1: the id of *AnalogSignalGroupRoomTemperature* is out of range. 2: the id of *AnalogSignalSetpointValueShift* is out of range. 3: the id of *DigitalSignalGroupDewPoint* is out of range. 4: the id of *DigitalSignalGroupWindowContact* is out of range. 5: the id of *DigitalSignalGroupMotionDetection* is out of range. 6: - 7: - 8: the id of *ValveActuatorGroupHeating2Point* is out of range. 9: the id of *ValveActuatorGroupCooling2Point* is out of range. 10: the id of *ValveActuatorGroupHeatingContinuous* is out of range. 11: the id of *ValveActuatorGroupCoolingContinuous* is out of range. 12: the id of *WeatherStationOutdoorTemperature* is out of range. 13: the parameter *SummerCompensationStartLevel* >= *SummerCompensationEndLevel*. 14: the parameter *SummerCompensationGradient* is out of range. 15: the setpoint values are invalid. 16: the parameter *PWMPeriodDurationCooling* is out of range. 17: the parameter *PWMPeriodDurationHeating* is out of range. 18: the parameter *YMinPIDCooling* is out of range. 19: the parameter *YMaxPIDCooling* is out of range. 20: the parameter *YMinPIDCooling* >= *YMaxPIDCooling*. 21: the parameter *YMinPIDHeating* is out of range. 22: the parameter *YMaxPIDHeating* is out of range. 23: the parameter *YMinPIDHeating* >= *YMaxPIDHeating*.

Hex	Dec	Description	Parameter
0x8019	32793	The parameters of the analog signal object are invalid.	1: *EnOceanMappingListItem* is out of range. 2: *EnOceanDataByte* is out of range. 3: the id of *DALILine* is out of range. 4: the *DALIShortAddress* is out of range. 5: the id of *EnOceanLine* is out of range.
0x801A	32794	The parameters of the digital signal object are invalid.	1: *EnOceanMappingListItem* is out of range. 2: *EnOceanDataByte* is out of range. 3: *EnOceanDataBit* is out of range. 4: the id of *EnOceanLine* is out of range. 5: the id of *DALILine* is out of range. 6: the *DALIShortAddress* is out of range.
0x801B	32795	The signal object of the actual value is in error state.	-
0x801C	32796	The signal object of the dew point value is in error state.	-
0x801D	32797	The parameters of the window object are invalid.	1: the id of *WindowDriveGroup* is out of range. 2: the id of *DigitalSignalGroupClose* is out of range. 3: the id of *DigitalSignalGroupOpen* is out of range.
0x801E	32798	The DALI Ballast is in error state.	-
0x801F	32799	The KL3403 is in error state.	0x100: timeout Error. The allowed execution time elapsed. 0x200: parameter Error (e.g. at an incorrect register number). 0x300: the gathered value differs from the written value (write access to this register is possibly not allowed or has failed).
0x8020	32800	The parameters of the iFront blind drive object are invalid.	-
0x8021	32801	Overload - *LagDistance* greater than allowed.	-
0x8022	32802	Encoder doesn't count.	-
0x8023	32803	Drive not calibrated (at start to target position).	-
0x8024	32804	Terminal not ready.	-
0x8025	32805	Error of terminal.	-

Hex	Dec	Description	Parameter
0x8026	32806	The parameters of the time scheduler channel object are invalid.	1: periodicity is equal to 0 or greater than 365. 2: periodicity is equal to 0 or greater than 52. 3: no weekday selected. 4: a switching event within the next 1500 days cannot be determined. 5: periodicity is equal to 0 or greater than 31. 6: no month selected. 7: periodicity is equal to 0 or greater than 4. 8: the initial day is equal to 0 or greater than 365. 9: the initial week is equal to 0 or greater than 52.
0x8027	32807	The parameters of the weather station object are invalid.	1: the id of *AnalogSignalTemperature* is out of range. 2: the id of *AnalogSignalWindSpeed* is out of range. 3: the id of *AnalogSignalBrightnessEast* is out of range. 4: the id of *AnalogSignalBrightnessSouth* is out of range. 5: the id of *AnalogSignalBrightnessWest* is out of range. 6: the id of *AnalogSignalDaylight* is out of range. 7: the id of *AnalogSignalRain* is out of range. 8: the id of *AnalogSignalRelativeHumidity* is out of range. 9: the id of *SerialCommunicationLine* is out of range.
0x8028	32808	Errror accessing the DALI ballast.	1: no answer from the DALI terminal. 2: no answer from the DALI ballast. 3: communication buffer overflow. 4: no answer from the communication block. 5: DALI collision detected on the backward channel: During the transfer of a DALI telegram, a collision with the send data of another DALI slave was detected. 6: DALI collision detected on the forward channel: During the transfer of a DALI telegram, a collision with the send data of another DALI master was detected. 7: Overload of the internal DALI power supply unit of the KL6811 (bus under-voltage).

Hex	Dec	Description	Parameter
0x8029	32809	Error during receiving data from weather station.	1: watchdog. 2: dataframe has wrong length. 3: error Checksum.
0x802A	32810	The parameters of the composition module object are invalid.	1: the id of *DigitalSignalPushButton* is out of range. 2: the id of *LampFeedbackLED* is out of range. 3: the id of *Lighting* is out of range. 4: the id of *SunblindWindow* is out of range. 5: the id of *Scene* is out of range..
0x802B	32811	The parameters of the blind drive object are invalid.	1: the id of *DigitalSignalGroupLockDriveDown* is out of range.
0x802C	32812	Terminal is not initialized.	-
0x802D	32813	Direction of rotation of motor is wrong. Check wiring.	-

More Information:
www.beckhoff.com/ts8100

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com