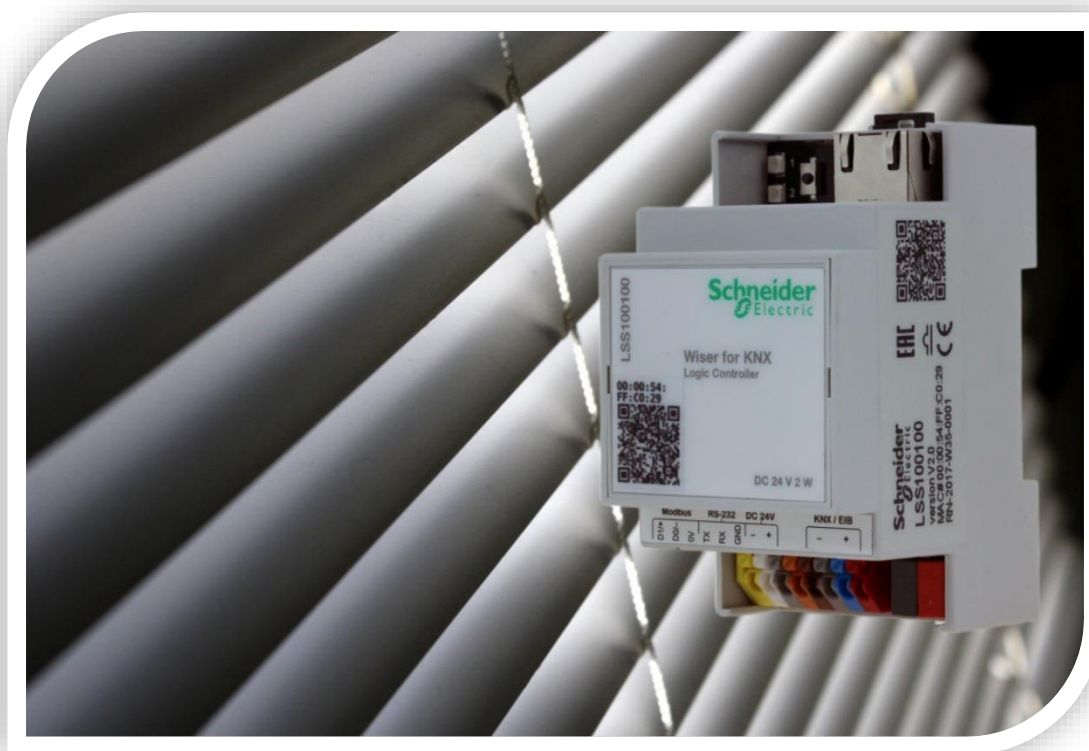


Application note

Lighting solution for comfortable environment using Wiser for KNX II

Applying Constant Light Control algorithm in Wiser for KNX to blinds and lights



Safety Information

Important Information



Read these instructions carefully before trying to install, configure, or operate this software. The following special messages may appear throughout this bulletin or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of either symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, can result in death or serious injury.

CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, can result in minor or moderate injury.

NOTICE


NOTICE is used to address practices not related to physical injury. The safety alert symbol shall not be used with this signal word.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction, installation, and operation of electrical equipment and has received safety training to recognize and avoid the hazards involved.

Safety Precautions

 WARNING
<p>HAZARD OF INCORRECT INFORMATION</p> <ul style="list-style-type: none">• Do not incorrectly configure the software, as this can lead to incorrect reports and/or data results.• Do not base your maintenance or service actions solely on messages and information displayed by the software.• Do not rely solely on software messages and reports to determine if the system is functioning correctly or meeting all applicable standards and requirements.• Consider the implications of unanticipated transmission delays or failures of communications links. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information that is contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2014 Schneider Electric. All rights reserved

Table of Contents

1	Introduction	7
2	Design	9
2.1	Flowchart.....	9
2.2	Flowchart description.....	10
3	ETS Configuration.....	11
3.1	Blind actuator	11
3.1.1	General	11
3.1.2	Channel configuration.....	11
3.1.3	Blind	12
3.1.4	Drive.....	14
3.1.5	Automatic mode	15
3.1.6	Calibration	16
3.1.7	Status reports	17
3.2	Dimming actuator	18
3.2.1	Dimming time reduction.....	19
3.3	Brightness sensor.....	20
3.4	Push-buttons	21
3.4.1	Button for dimming up	21
3.4.2	Button for dimming down.....	22
3.4.3	Button for blind movement up.....	23
3.4.4	Button for blind movement down	23

3.5	Creating group addresses	24
3.6	Linking group addresses	25
3.7	Download devices	27
3.8	Export *.ESF file and import to the Wiser for KNX	27
4	Wiser for KNX configuration	29
4.1	Data type.....	29
4.2	Creating objects in the Wiser for KNX	31
4.3	Blind calibration	31
4.4	Creation of block functions	Error! Bookmark not defined.
4.5	Block functions	Error! Bookmark not defined.
4.5.1	Light control block function	Error! Bookmark not defined.
4.5.2	Blind control block function.....	Error! Bookmark not defined.
4.6	Scripts setting.....	32
4.6.1	Light control.....	32
4.6.2	Blind control	Error! Bookmark not defined.
4.6.3	Auto/manual mode transition	33
4.6.4	Manual/auto mode transition	34
5	User interface.....	36
6	Conclusion.....	38
7	Appendix	38
7.1	Glossary	38

1 Introduction

This application note describes how to integrate and configure blinds, dimming actuators and Wiser for KNX for constant light control functionality in a room. Proper cooperation between brightness sensor and dimmer helps you to maximize usage of the daylight and to reduce the energy costs of artificial lighting.

Moreover, you achieve a better lighting comfort. Integrating blinds with light control allows you to:

- Reduce energy costs by controlling lights to constant light level or switching them off when room is not occupied.
- Get more benefits from daylight and reduce the intensity of artificial lighting according to individual needs of users
- Adjust optimal lighting conditions for efficient and comfortable work
- Protect workplaces from direct sunlight and disruptive reflections
- Reduce costs of devices (PIR with constant light control feature is not necessary)

Part 1 of previous Application note 032 “Lighting solution for comfortable environment using Wiser for KNX I” described light control using CLC algorithm built in the PIR

Part 2 described in this Application note 034 relates to light control using CLC algorithm built in the Wiser for KNX

Proportional regulation described in this Application note is useful in scenario when it is not possible to use presence detector (PIR) with constant light control algorithm (CLC). Wiser for KNX logic controller fully replaces PIR with CLC thanks to programmed CLC function block. Input of measured brightness over controlled area can be collected from any low cost brightness sensor with analog signal converted to KNX bus via Analog Input Converter.

This AN is evolution of AN032 - Lighting solution for comfortable environment using Wiser for KNX I (regulation at constant light level is integrated in PIR)

A glossary is available in the appendix chapter of this document. Please refer to it whenever necessary.

Competencies

This document is intended for readers who have been trained on Wiser for KNX, spaceLYnk products. The integration should not be attempted by someone who is new to the installation of either products. In addition we recommend that you are familiar with:

- Concepts of the KNX bus
- The ETS configuration software
- Lua scripting

System prerequisites

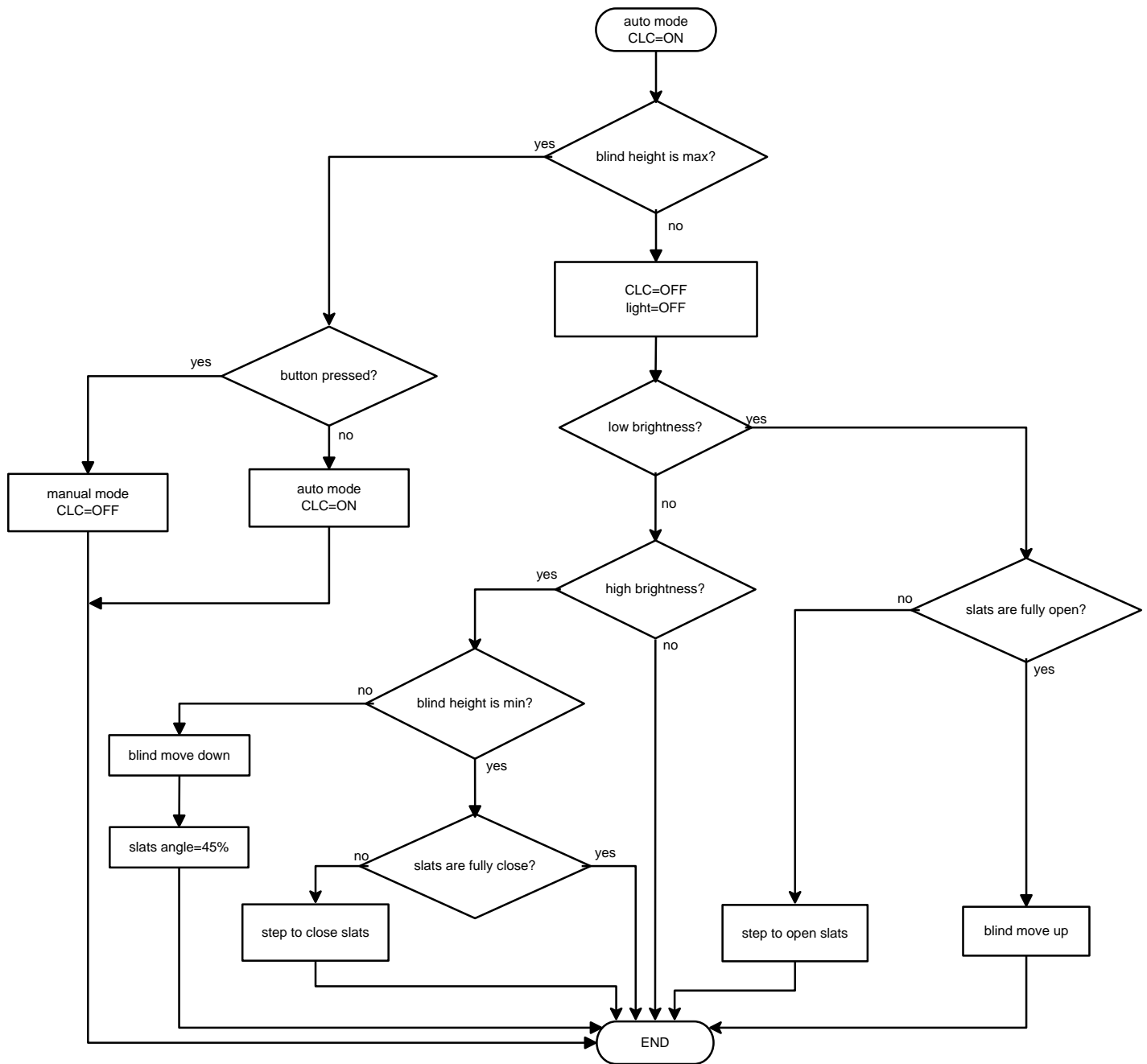
Software	Version	Download
Wiser for KNX	2.1 and newer	http://www.schneider-electric.com
ETS	5.3.3	http://www.knx.org/
Hardware	Order number	Manufacturer
Blind actuator REG-K/4x/10 with manual mode	MTN649804	Schneider Electric
Universal dimming actuator REG-K/4x230/250W	MTN649325	Schneider Electric
Wiser for KNX logic controller	LSS100100	Schneider Electric
KNX Push-button, 2-gang	MGU3.531.12	Schneider Electric
Power supply	MTN684016	Schneider Electric
KNX brightness and temperature sensor	MTN663991	Schneider Electric

Table 1: System prerequisites

NOTE: It is not possible to use combined switch/blind actuators (MTN649908 a MTN649912)

2 Design

2.1 Flowchart



Picture 1: Blind & Light control algorithm

2.2 Flowchart description

The main script for blind control is resident. It changes the height of blinds and tilt angle of slats depending on brightness measured by brightness sensor. It also enables the Constant Light Control (CLC) block function if blind is fully open and there are still insufficient light conditions.

Assuming the light control was not switched OFF in manual mode by pressing push button, the CLC block function is enabled if the blinds are in the top position. Pressing the push-button disables CLC and switches the light control to manual mode and starts staircase timer. The manual mode is switched back to the automatic one after time out.

If the CLC is disabled, the light conditions are regulated by adjustment of the blinds.

If there are inadequate light conditions in the room the blind actuator first tries to adjust slat angle. If the slat angle is maximal (slats are opened) and there are still poor light conditions the blind starts to move up until the light conditions are acceptable or blinds reached the top position.

Otherwise if there is too much light the blind actuator moves blind to bottom position and tilts slats to 45° angle. If there is still too much light the blind actuator continues closing slats.

NOTE: It is not possible to change slat angle at the minimal height of blind. Therefore, if the blind height is less than 3%, at first the minimum height is set to 3% before the slat angle is changed.

The maximum opening angle is set to 95% and the minimum opening angle to 5%. If your slats can work in full range change values to 0% and 100%.

3 ETS Configuration

First, import required devices to ETS project (see **Table 1**) and set properly all parameters as it is described in the following chapters.

3.1 Blind actuator

3.1.1 General

The screenshot shows the 'General' settings for a '1.1.4 Blind actuator REG-K/4x/10 with manual mode'. The left sidebar contains a tree view with 'General' selected, and other options like 'Channel config.', '1: Blind', '1: Drive', '1: Automatic mode', '1: Calibration', and '1: Status reports'. The main area displays various configuration parameters:

Parameter	Value / Options
Manual operation type	<input checked="" type="radio"/> Bus and manual operation <input type="radio"/> Manual operation only
Manual operation enabled	enabled (dropdown)
Time-dependent reset of manual operation	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Scenes in general	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Ext. input scenes	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Status of mains voltage	disabled (dropdown)
Minimum interval for status reports	200 ms (dropdown)
Set the order of priority for higher-level functions	1.Alarm 2.Weather alarm 3.Locking 4.Movement (dropdown)
Weather alarm function in general	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Reference movement in general	<input type="radio"/> disabled <input checked="" type="radio"/> enabled

Picture 2: General setting for blind actuator

For proper calibration enable **Reference movement** for calibration all channels at once.

Go to → **Parameter** → **General** → **Reference movement in general** → **enabled**

3.1.2 Channel configuration

At channel configuration tab set at least one channel operation mode to Blind

Go to → **Parameter** → **Channel config.** → **Channel 1 operation mode** → **Blind**

The following settings must be performed individually for each channel.

3.1.3 Blind

Set the required behavior of the blind channel. Enable automatic mode, calibration and status messages.

1.1.4 Blind actuator REG-K/4x/10 with manual mode > 1: Blind		
General	How does the existing blind move?	downwards closed / upwards horizontal ▼
Channel config.	Slat position after movement	last slat position ▼
1: Blind		
1: Drive	Automatic controls / Presets	Automatic controls ▼
1: Automatic mode	Scenes	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
1: Calibration	Manual locking	inactive ▼
1: Status reports	Calibration	<input type="radio"/> disabled <input checked="" type="radio"/> enabled
	Weather alarm	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
	Alarm function	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
	Disable function	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
	Movement range limits	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
	Failure mode	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
	Status signals	<input type="radio"/> disabled <input checked="" type="radio"/> enabled
	Manual operation when bus voltage fails (mains voltage present)	<input checked="" type="radio"/> disabled <input type="radio"/> enabled

Picture 3: Setting for blind actuator channel

Go to → **Parameter** → **Blind** → **Automatic mode / Presets** → **Automatic controls**

Go to → **Parameter** → **Blind** → **Calibration** → **enabled**

Go to → **Parameter** → **Blind** → **Status signals** → **enabled**

Height and the slat position are periodically detected.

Optionally you can set the alarm function if there is a device which is able to detect dangerous conditions (fire, wind, frost etc.) to avoid damaging the blinds or easy access of firefighters in case of fire.

Go to → **Parameter** → **Blind** → **alarm function** → **enable**

Go to → **Parameter** → **Blind** → **weather alarm** → **enable**

In the **Alarm function** tab set behavior of the blinds when the alarm is triggered.

1.1.4 Blind actuator REG-K/4x/10 with manual mode > 1: Alarm function	
General	Alarm <input type="radio"/> at object value "0" <input checked="" type="radio"/> at object value "1"
Channel config.	
1: Blind	Behaviour at start of alarm UP
1: Drive	Behaviour at the end of alarm no reaction
1: Automatic mode	Alarm status in event of bus voltage recovery inactive
1: Calibration	
1: Weather alarm	
1: Alarm function	
1: Status reports	

Picture 4: Alarm function for blind actuator channel

Go to → **Parameter** → **Alarm function** → **behavior at the start of alarm** → **up**

Go to → **Parameter** → **Alarm function** → **behavior at the end of alarm** → **no reaction**

Blinds will move up to maximum height position in case of alarm.

3.1.4 Drive

Following setting is important for correct positioning of the blinds!

In the **Drive** tab set downward running time for blinds and additional time for moving upward.

The **factor of running time of height adjustment** is measured time of blinds when move from upper to bottom position in *ms* divided by **Time base**.

1.1.4 Blind actuator REG-K/4x/6 > 1: Drive

General

Channel config

1: Blind

1: Drive

1: Automatic mode

1: Locking function

1: Status reports

Extended drive parameters

☒ disable

☐ enable

Time base for running time of height adjustment

☐ 10 ms

☒ 100 ms

Factor for running time of height adjustment (10-64000) 1 second = 1000 ms

235

Common time base for additional running time / idle time

☒ 10 ms

☐ 100 ms

Factor for additional running time in upward direction

20

Time base for step interval of slat

☒ 10 ms

☐ 100 ms

Factor for step interval of slat (5-255)

10

Time base for running time of slat

☒ 10 ms

☐ 100 ms

Factor for running time of slat (5-255)

100

Pause on reverse on change in direction (1-255) factor * 100 ms, manuf. data

5

Picture 5: Blind drive setting

After correct setting real height of blinds and slats angle corresponds to the values of status objects.

3.1.5 Automatic mode

In tab **Automatic mode** make sure that the automatic locking mode is at object value 1.

Go to → **Parameter** → **Automatic mode** → **Automatic locking** → **at object value 1**

Picture 6: Automatic mode

Set behavior on deactivating the automatic locking.

Go to → **Parameter** → **Automatic mode** → **Behavior on deactivating...** → **accept current automatic...**

Set behavior on manual control (push buttons)

Go to → **Parameter** → **Automatic mode** → **Reaction in ...** → **automatic mode temporarily disabled**

Set how long will be automatic mode disabled after manual action.

Go to → **Parameter** → **Automatic mode** → **Time base for deactivation time...** → **1 hr**

Go to → **Parameter** → **Automatic mode** → **Factor for deactivation time...** → **8**

3.1.6 Calibration

Set the calibration of blinds.

1.1.4 Blind actuator REG-K/4x/10 with manual mode > 1: Calibration		
General	Reference movement to	Activate value "1" at reference movement object ▼
Channel config.	Delay for reference movement via object in seconds	disabled ▼
1: Blind	Reference position	upper ▼
1: Drive	Position after reference movement via object	position prior to reference movement ▼
1: Automatic mode		
1: Calibration	Automatic calibration	<input checked="" type="radio"/> upper limit position <input type="radio"/> upper and lower limit positions

Picture 7: Calibration of blind

Go to → **Parameter** → **Calibration** → **Reference movement to** → **Activate value "1"...**

Go to → **Parameter** → **Calibration** → **Delay for reference...** → **disabled**

Go to → **Parameter** → **Calibration** → **Reference position** → **upper**

Go to → **Parameter** → **Calibration** → **Position after...** → **position prior to reference movement**

Go to → **Parameter** → **Calibration** → **Automatic mode calibration** → **upper limit position**

The blinds will be calibrated when the reference movement object is 1.

3.1.7 Status reports

In the **Status reports** tab set all columns to **active status**.

1.1.3 Blind actuator REG-K/4x/10 with manual mode > 1: Status reports

General	Status of height	active status response object ▼
Channel config.	Status of slat	active status response object ▼
1: Blind	Status of automatic mode	active status response object ▼
1: Drive	Type of locking signal	<input checked="" type="radio"/> drive locking <input type="radio"/> movement range limit
1: Automatic mode	Status of drive locking	active status response object ▼
1: Calibration		
1: Status reports		

Picture 8: Status reports of dimming actuator

Go to → **Parameter** → **Status report** → **Status of height** → **active status response object**

Go to → **Parameter** → **Status report** → **Status of slat** → **active status response object**

Go to → **Parameter** → **Status report** → **Status of automatic mode** → **active status response object**

Go to → **Parameter** → **Status report** → **Type of locking signal** → **drive locking**

Go to → **Parameter** → **Status report** → **Status of drive locking** → **active status response object**

3.2 Dimming actuator

Configure dimmer actuator for periodical detection of its status.

Activate at least one channel to be able to set following parameters.

The screenshot shows a configuration window for a dimming actuator. On the left is a sidebar with a 'General' tab selected, and below it, a list of channels: '1: General', '1: Base dimming curve', and '1: Dimming time reduction'. The main area displays parameters for the selected channel. The 'Central function' is set to 'disabled' (radio button). 'Higher priority function' is set to 'deactivated' (text field). 'Disable function' is set to 'deactivated' (radio button). 'Behaviour on bus voltage recovery and download' is set to 'no reaction' (text field). 'Status switch' is set to 'active status response object' (text field). 'Status value object/brightness value' is set to 'active status response object' (text field).

Parameter	Value
Central function	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Higher priority function	deactivated
Disable function	<input checked="" type="radio"/> deactivated <input type="radio"/> activated
Behaviour on bus voltage recovery and download	no reaction
Status switch	active status response object
Status value object/brightness value	active status response object

Picture 9: Status objects of dimming actuator

In **General tab** set both status objects.

Go to → **Parameter** → **General** → **Status switch**→ active status response object

Go to → **Parameter** → **General** → **Status value object/brightness value**→ active status response object

The status feedback of objects is following status of dimmer.

3.2.1 Dimming time reduction

There is necessary to choose the format of dimming time in the **Dimming time reduction** tab.

General	Dimming time reduction object for dimming curve	<input checked="" type="radio"/> deactivated <input type="radio"/> activated
1: General	Sets for dimming time reduction	
1: Base dimming curve	Format of dimming time reduction	<input checked="" type="radio"/> 1 - 100 % <input type="radio"/> 1 - 255 (corresponds to 1-100 %)
1: Dimming time reduction		

Picture 10: Dimming time reduction format of dimming actuator

Go to → **Parameter** → **Dimming time reduction** → **Format for dimming actuator**→ 1 -100 %

3.3 Brightness sensor

Send brightness value cyclically

1.1.1 KNX brightness and temperature sensor > Measured values

Measured values	Send brightness value on change of	not due to a change
Channel use	Send brightness value cyclically	Every minute
C1 Brightness	Send temperature on change of	not due to a change
C1.1	Send temperature cyclically	don't send cyclically
C2 Temperature	Temperature offset in 1/10°C (-64 .. 64)	0
C2.1		

Picture 11: Brightness sensor setting

Go to → Parameter → Measured values → Send brightness value cyclically → Every minute

NOTE: The accuracy of lighting in the workplace depends on the location of brightness sensor and reflection of the light in the room.

3.4 Push-buttons

Configure push-buttons for manual control

3.4.1 Button for dimming up

1.1.2 Button KNX Push-button, 2-gang > Push-button 1		
Push-button info	Select push-button function	Dimming ▼
Push-button 1	Long operation time equals 100 ms * factor (4-250)	6 ▲▼
Push-button 2		
Push-button 3	Dimming direction	Brighter ▼
Push-button 4	Step dimming (brighter)	To max. brightness ▼
Disable function for push-butt...	Stop telegram after release	<input checked="" type="radio"/> Yes <input type="radio"/> No
Scene module	Send dimming levels cyclically	<input type="radio"/> Yes <input checked="" type="radio"/> No
	Trigger status LED	From switch object ▼

Picture 12: Push-button dimming up

Go to → **Parameter** → **Push-button 1** → **Select push-button function** → **dimming**

Go to → **Parameter** → **Push-button 1** → **dimming direction** → **brighter**

Go to → **Parameter** → **Push-button 1** → **step dimming** → **to max. brightness**

Go to → **Parameter** → **Push-button 1** → **stop telegram after release** → **Yes**

3.4.2 Button for dimming down

1.1.2 Button KNX Push-button, 2-gang > Push-button 2		
Push-button info	Select push-button function	Dimming ▼
Push-button 1	Long operation time equals 100 ms * factor (4-250)	6 ▲▼
Push-button 2		
Push-button 3	Dimming direction	Darker ▼
Push-button 4	Step dimming (darker)	To min. brightness ▼
Disable function for push-butt...	Stop telegram after release	<input checked="" type="radio"/> Yes <input type="radio"/> No
Scene module	Send dimming levels cyclically	<input type="radio"/> Yes <input checked="" type="radio"/> No
	Trigger status LED	From switch object ▼

Picture 13: Push-button dimming down

Go to → **Parameter** → **Push-button 1** → **Select push-button function** → **dimming**

Go to → **Parameter** → **Push-button 1** → **dimming direction** → **darker**

Go to → **Parameter** → **Push-button 1** → **step dimming** → **to min. brightness**

Go to → **Parameter** → **Push-button 1** → **stop telegram after release** → **Yes**

The objects of button are ready for connection with dimming actuator.

3.4.3 Button for blind movement up

1.1.2 Button KNX Push-button, 2-gang > Push-button 3		
Push-button info	Select push-button function	Blind ▼
Push-button 1	Long operation time equals 100 ms * factor (4-250)	6 ▲▼
Push-button 2		
Push-button 3	Direction of movement, blind	Up ▼
Push-button 4	Trigger status LED	Long operation = ON / release = OFF ▼
Disable function for push-butt...		
Scene module		

Picture 14: Push-button blind movement up

3.4.4 Button for blind movement down

1.1.2 Button KNX Push-button, 2-gang > Push-button 4		
Push-button info	Select push-button function	Blind ▼
Push-button 1	Long operation time equals 100 ms * factor (4-250)	6 ▲▼
Push-button 2		
Push-button 3	Direction of movement, blind	Down ▼
Push-button 4	Trigger status LED	Long operation = ON / release = OFF ▼
Disable function for push-butt...		
Scene module		

Picture 15: Push-button blind movement down

The objects of button are ready for connection with blind actuator.

3.5 Creating group addresses

Create topology for designed room. Create the group addresses for all important objects in the device.

NOTE: We strongly recommend you to follow group addresses during testing possibilities of this solution.

Later, you can change addresses according project needs. BTN = button, PIR= presence sensor, BL= blind actuator

group name	group address	Group description
LS_brightness value	0/0/1	Actual value of brightness
BL_movement object for manual	0/0/2	Moving blinds in manual mode
BL_stop/step object in manual	0/0/3	Stopping blind movement in manual mode
BL_height position in manual	0/0/4	Setting height of blind in manual mode
BL_slats position in manual	0/0/5	Changing slats angle in manual mode
BL_movement object for auto	0/0/6	Moving blinds in automatic mode
BL_stop/step object in auto	0/0/7	Stopping blind movement in automatic mode
BL_height position in auto	0/0/8	Setting height of blind in automatic mode
BL_slats position in auto	0/0/9	Changing slats angle in automatic mode
BL_automatic locking	0/0/10	Locking automatic movements
BL_status feedback for height	0/0/11	Status of blind height
BL_status feedback for slats	0/0/12	Status of slats angle
BL_status feedback for drive locking	0/0/13	Status of blind drive locking
BL_calibration	0/0/14	Variable for blind calibration
DIM_switch object	0/0/15	Switching of dimming actuator
DIM_dimming object	0/0/16	Dimming input of dimming actuator
DIM_value object	0/0/17	Value input of dimming actuator
DIM_status feedback switch	0/0/18	Status of dimming actuator state
DIM_status feedback value	0/0/19	Status of dimming actuator value
BTN_switch light on	0/0/20	Short button press – switch light on
BTN_switch light off	0/0/21	Short button press – switch light off
BTN_dimming brighter	0/0/22	Long button press - dimming light brighter
BTN_dimming darker	0/0/23	Long button press - dimming light darker
BTN_move up	0/0/24	Long button press – blind move up
BTN_move down	0/0/25	Long button press – blind move down
BTN_slats step to open	0/0/26	Short button press - step to open slats
BTN_slats step to close	0/0/27	Short button press - step to close slats

Table 2: Group address summary

3.6 Linking group addresses

Link ETS objects to appropriate group addresses.

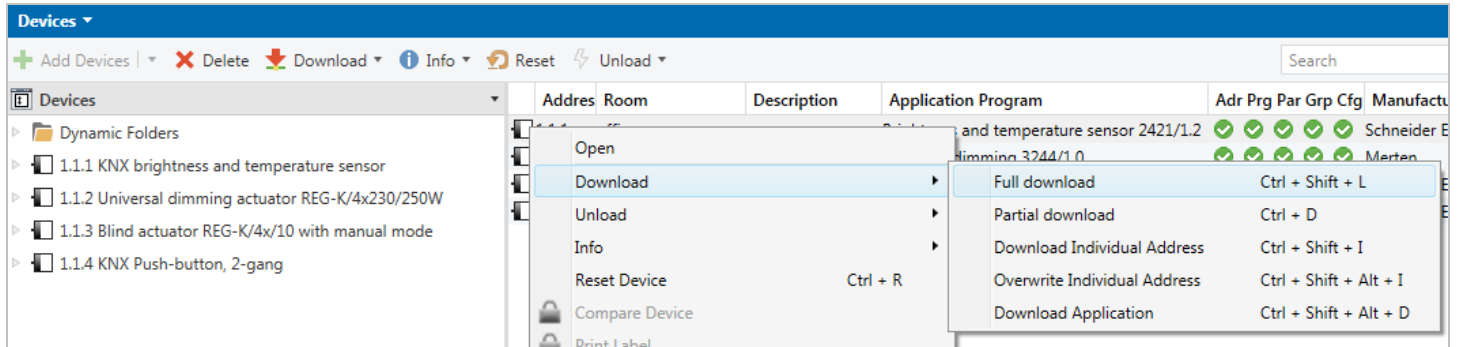
Group name	group address	Objects
LS_brightness value	0/0/1	KNX brightness and temperature sensor 0: Brightness value - Physical value
BL_movement object for manual	0/0/2	Blind actuator REG-K/4x/10 with manual mode 0: Movement object in manual mode - Channel 1
BL_stop/step object in manual	0/0/3	Blind actuator REG-K/4x/10 with manual mode 1: Stop/step object in manual mode - Channel 1
BL_height position in manual	0/0/4	Blind actuator REG-K/4x/10 with manual mode 2: Height position in manual mode - Channel 1
BL_slat position in manual	0/0/5	Blind actuator REG-K/4x/10 with manual mode 3: Slat position in manual mode - Channel 1
BL_movement object for auto	0/0/6	Blind actuator REG-K/4x/10 with manual mode 4: Movement object in automatic mode - Channel 1
BL_stop/step object in auto	0/0/7	Blind actuator REG-K/4x/10 with manual mode 5: Stop/step object in automatic mode - Channel 1
BL_height position in auto	0/0/8	Blind actuator REG-K/4x/10 with manual mode 6: Height position in automatic mode - Channel 1
BL_slat position in auto	0/0/9	Blind actuator REG-K/4x/10 with manual mode 7: Slat position in automatic mode - Channel 1
BL_automatic locking	0/0/10	Blind actuator REG-K/4x/10 with manual mode 12: Automatic locking - Channel 1
BL_status feedback for height	0/0/11	Blind actuator REG-K/4x/10 with manual mode 16: Status feedback for height - Channel 1
BL_status feedback for slats	0/0/12	Blind actuator REG-K/4x/10 with manual mode 17: Status feedback for slats - Channel 1
BL_status feedback for drive locking	0/0/13	Blind actuator REG-K/4x/10 with manual mode 18: Status feedback for drive locking - Channel 1
BL_calibration	0/0/14	Blind actuator REG-K/4x/10 with manual mode 233: Activate reference movement - Calibration of all channels
DIM_switch object	0/0/15	Universal dimming actuator REGK/4x230/250W 0: Switch object - Channel 1, general
DIM_dimming object	0/0/16	Universal dimming actuator REGK/4x230/250W 1: Dimming object - Channel 1, general
DIM_value object	0/0/17	Universal dimming actuator REGK/4x230/250W 2: Value object - Channel 1, general
DIM_status feedback switch	0/0/18	Universal dimming actuator REGK/4x230/250W 8: Status feedback switch - Channel 1, status feedback
DIM_status feedback value	0/0/19	Universal dimming actuator REGK/4x230/250W 9: Status feedback value object/brightness value - Channel 1, status feedback

BTN_switch light on	0/0/20	Universal dimming actuator REGK/4x230/250W 0: Switch object - Channel 1, general KNX Push-button, 2-gang 0: Switch object - Push-button 1
BTN_switch light off	0/0/21	Universal dimming actuator REGK/4x230/250W 0: Switch object - Channel 1, general KNX Push-button, 2-gang 3: Switch object - Push-button 2
BTN_dimming brighter	0/0/22	Universal dimming actuator REGK/4x230/250W 1: Dimming object - Channel 1, general KNX Push-button, 2-gang 1: Dimming object - Push-button 1
BTN_dimming darker	0/0/23	Universal dimming actuator REGK/4x230/250W 1: Dimming object - Channel 1, general KNX Push-button, 2-gang 4: Dimming object - Push-button 2
BTN_move up	0/0/24	Blind actuator REG-K/4x/10 with manual mode 0: Movement object in manual mode - Channel 1 KNX Push-button, 2-gang 7: Movement object - Push-button 3
BTN_move down	0/0/25	Blind actuator REG-K/4x/10 with manual mode 0: Movement object in manual mode - Channel 1 KNX Push-button, 2-gang 10: Movement object - Push-button 4
BTN_slats step to open	0/0/26	Blind actuator REG-K/4x/10 with manual mode 1: Stop/step object in manual mode - Channel 1 KNX Push-button, 2-gang 6: Stop/step object - Push-button 3
BTN_slats step to close	0/0/27	Blind actuator REG-K/4x/10 with manual mode 1: Stop/step object in manual mode - Channel 1 KNX Push-button, 2-gang 9: Stop/step object - Push-button 4

Table 3: The assignment of objects to group addresses

3.7 Download devices

- Go to **Devices**
- Right click device you want to download.
- Click **Download**
- Click **Full download**



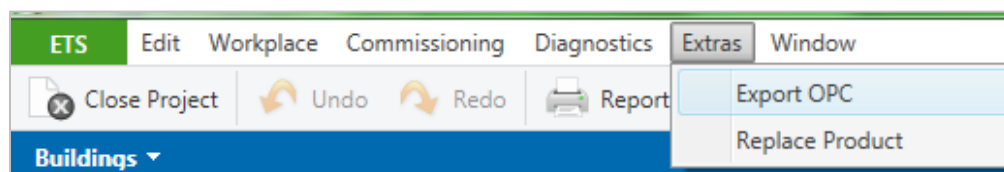
Picture 16: ETS downloading

Follow the ETS instruction. Repeat for all the devices.

3.8 Export *.ESF file and import to the Wiser for KNX

STEP 1: Export created group addresses to ESF file.

Go to → **Extras** → **Export OPC**

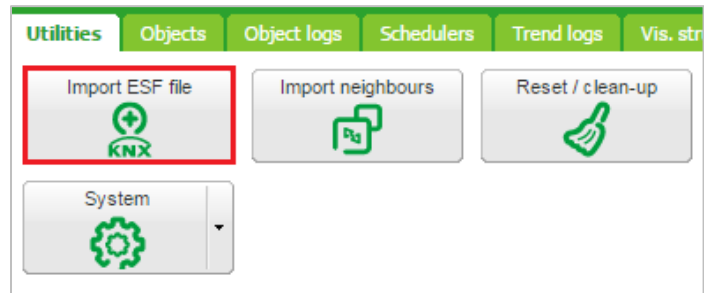


Picture 17: Exporting objects from ETS

RESULT: ESF file was created.

STEP 2: Import ESF to the Wiser for KNX.

Go to → **Configurator** → **Utilities** → **Import ESF file** → **Choose your ESF file**



Picture 18: Importing *.esf file to Wiser for KNX

RESULT: ESF file was imported.

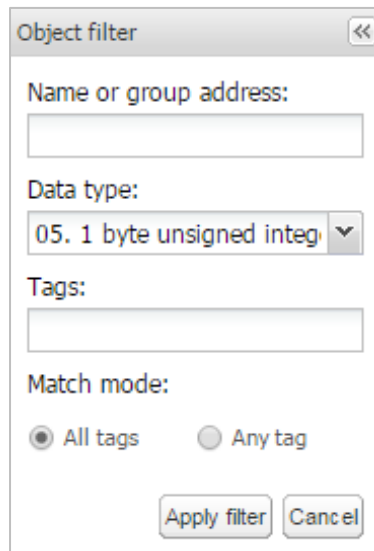
4 Wiser for KNX configuration

4.1 Data type

Change the all **1 byte unsigned** objects' data type to scale to display values in percentages.

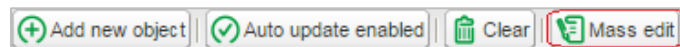
You can use **mass edit** function

STEP 1: Filter all **1 byte unsigned** objects



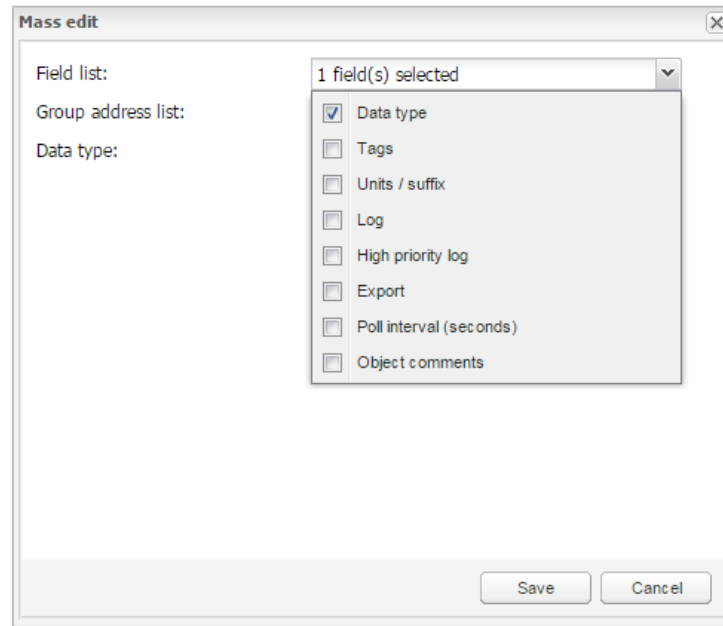
Picture 19: Object filter

STEP 2: Click **mass edit** button



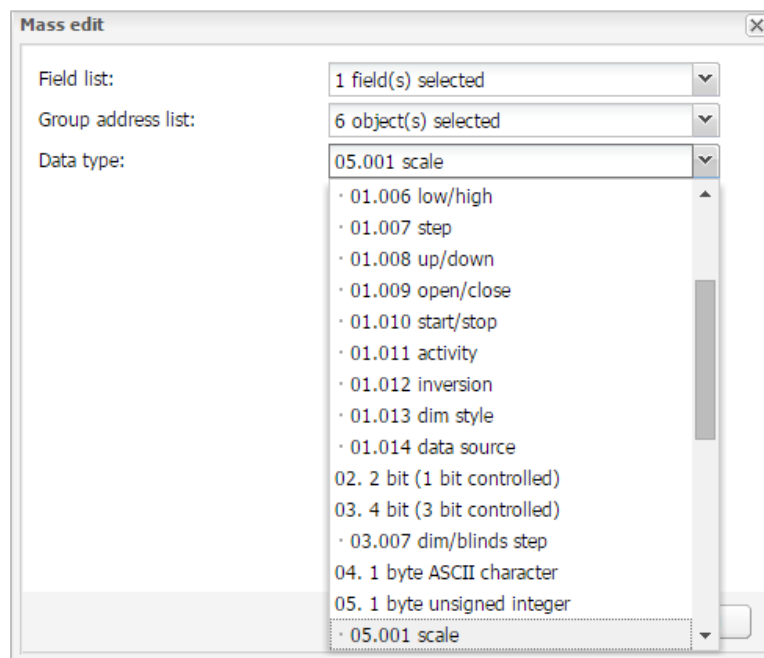
Picture 20: Mass edit 1

STEP 3: Set Field list to **Data type**



Picture 21: Mass edit 2

STEP 4: Set Data type to **scale**



Picture 22: Mass edit 3

STEP 5: Click **SAVE**

4.2 Creating objects in the Wiser for KNX

Create additional objects for user customization and proper script functionality.

It is necessary to create these objects for each separately controlled blind or for the group of blinds with identical behavior.

CUS_Button pressed – 1 bit - If variable is true, the manual mode is enabled and countdown timer is triggered.

CUS_Light control ON/OFF – 1 bit - If variable is true, the CLC is enabled.

For user customization, you can also create:

CUS_Desired brightness – 2 byte floating point - Brightness desired by user

CUS_Maximal brightness – 2 byte floating point - Maximum brightness acceptable by user

4.3 Blind calibration

After *.ESF file import the blind height feedback and slat angle feedback does not show correct values. Set value of the object **BL_calibration** to 1 and wait until the calibration is complete. **BL_status feedback for height** and **BL_status feedback for slats** will be updated. After calibration set value of the object **BL_calibration** to 0.

NOTE: Greater value of *Proportional coefficient* can speed up the regulation. Large value may cause instability of regulation. Value should be around 1.

Example (*Proportional coefficient* = 1) : When deviation from *Desired brightness* changes x%, dimmer changes value x%. Recommended value is in range from 0.1 to 1.0 according to the required speed of control. Default value is 1.0.

Conditions:

Maximum brightness = 1000 lx.

Light sensor brightness value = 300 lx

Desired brightness = 600 lx

Light lit on 10%

$$\text{regulation output} = K \times \frac{\text{Desired brightness} - \text{Light sensor brightness value}}{\text{Maximum brightness}} = 1 \times \frac{600 - 300}{1000} = 30\%$$

Value of regulation output is added: 10% + 30% = 40% and Light sensor brightness value is 500 lx.

4.4 Scripts setting

4.4.1 Light control

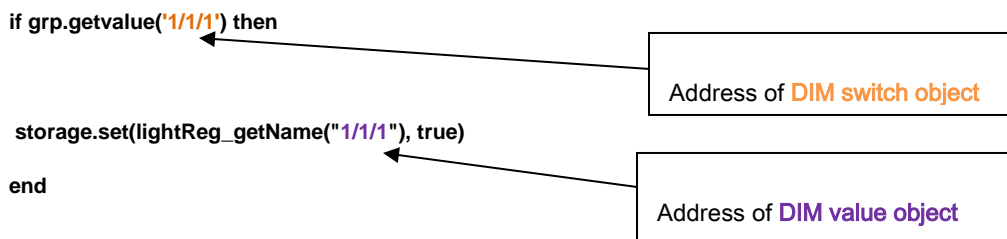
For proper function of light control is necessary to set following objects:

- DIM Switch object
- LS Brightness value
- CUS Desired value (2 byte floating point)
- DIM Status feedback value:
- DIM Value object

SCRIPT 1:

Set event-based script on object DIM Status feedback value:

Open script editor and fill following code:



SCRIPT 2:

- 1) Set same tag to objects LS brightness value, CUS desired brightness a DIM switch object.
- 2) Set event-based script on tag

Scripting → Event-based → Add new script

Insert Constant light control block function and fill all the parameters

4.4.2 Auto/manual mode transition

Create **button** tags on push buttons which control the lighting.

Group address ▾	Object name	Event sc...	Data type	Current value	Log	Export	Tags
0/0/26	BTN_dimming brighter		03.007 dim/blinds step	Stop	<input type="checkbox"/>	<input type="checkbox"/>	button
0/0/27	BTN_dimming darker		03.007 dim/blinds step	Stop	<input type="checkbox"/>	<input type="checkbox"/>	button
0/0/28	BTN_switch light ON		01. 1 bit (boolean)	1	<input type="checkbox"/>	<input type="checkbox"/>	button
0/0/29	BTN_switch light OFF		01. 1 bit (boolean)	0	<input type="checkbox"/>	<input type="checkbox"/>	button

Picture 23: Creating of button tags

Create event script on **button** tag

Go to → **Wiser for KNX web interface** → **Configurator** → **Scripting** → **Even-based** → **Add new script**

NOTE: Button tag must be unique for each button.

Event-based script

Script name:

Group address / tag:

Active: ☒

Execute on group read: ☐

Category:

Description:

Save Cancel

Picture 24: Creating of event script on button tag

Script will set **CUS_Button pressed** to true and switch OFF Light control every time if the button is pressed.

Light control is OFF until the value of object **CUS_Button pressed** is changed to 0.

Copy code to script editor:

```
grp.write('1/1/1', true)
grp.write('1/1/3', false)
```

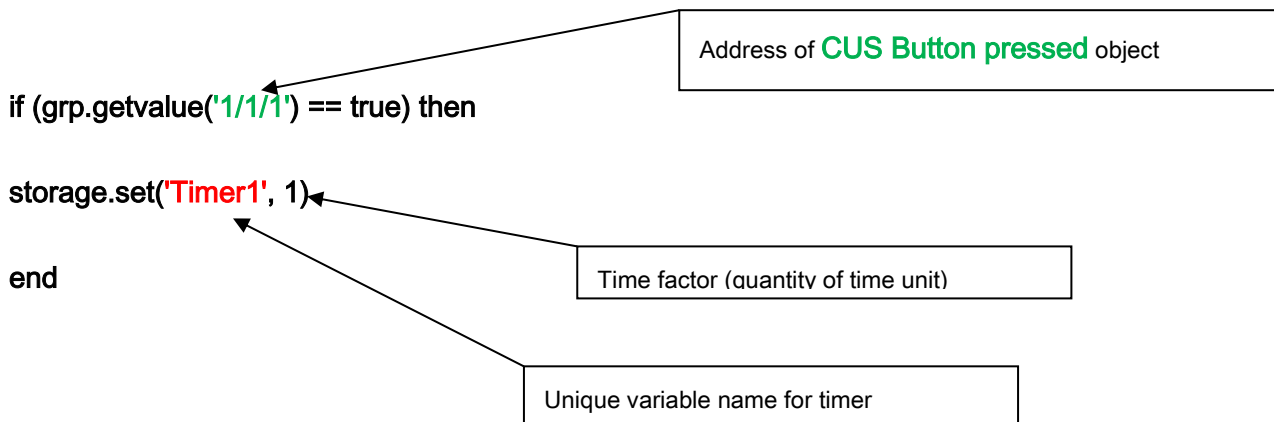
Address of **CUS Button pressed** object

Address of **CUS Light control ON/OFF** object

4.4.3 Manual/auto mode transition

To automatic mode can be switched by changing the **Automatic/manual object**. This can be done via a dedicated button or via script that will do so after a certain time. The following section is an example using scheduled script.

STEP 1: Create script on 'CUS_Button pressed' object which will set your own timer if the button is pressed.



NOTE: Time factor results in total time before the mode is switched back to automatic mode

STEP 2: Create a scheduled script that will countdown your timer

Scheduled script

Script name: Timercountdown

Minute: *

Hour: *

Day of the month: *

Month of the year: Every month of the year

Day of the week: Every day of the week

Active: ☒

Category:

Description: Decreasing countdown timer every minute

Save Cancel

Picture 25: Countdown timer setting for Time base 1 minute

NOTE: You can set the script evaluation to every minute or every hour according the Time base (time unit).

Total time = Time factor x Time base

STEP 3: Fill the scheduled script

```
local timer = storage.get('Timer1',0)
```

```
if timer==0 and grp.read('1/1/1')==true then
```

```
  grp.write('1/1/1', false)
```

```
  os.sleep(2)
```

```
  grp.write('1/1/3', true)
```

Address of **CUS Button pressed** object

```
else
```

```
  timer=timer-1
```

```
  os.sleep(2)
```

```
  storage.set('Timer1', timer)
```

```
end
```

Address of **CUS Light control ON/OFF** object

Save and activate script.

RESULT:

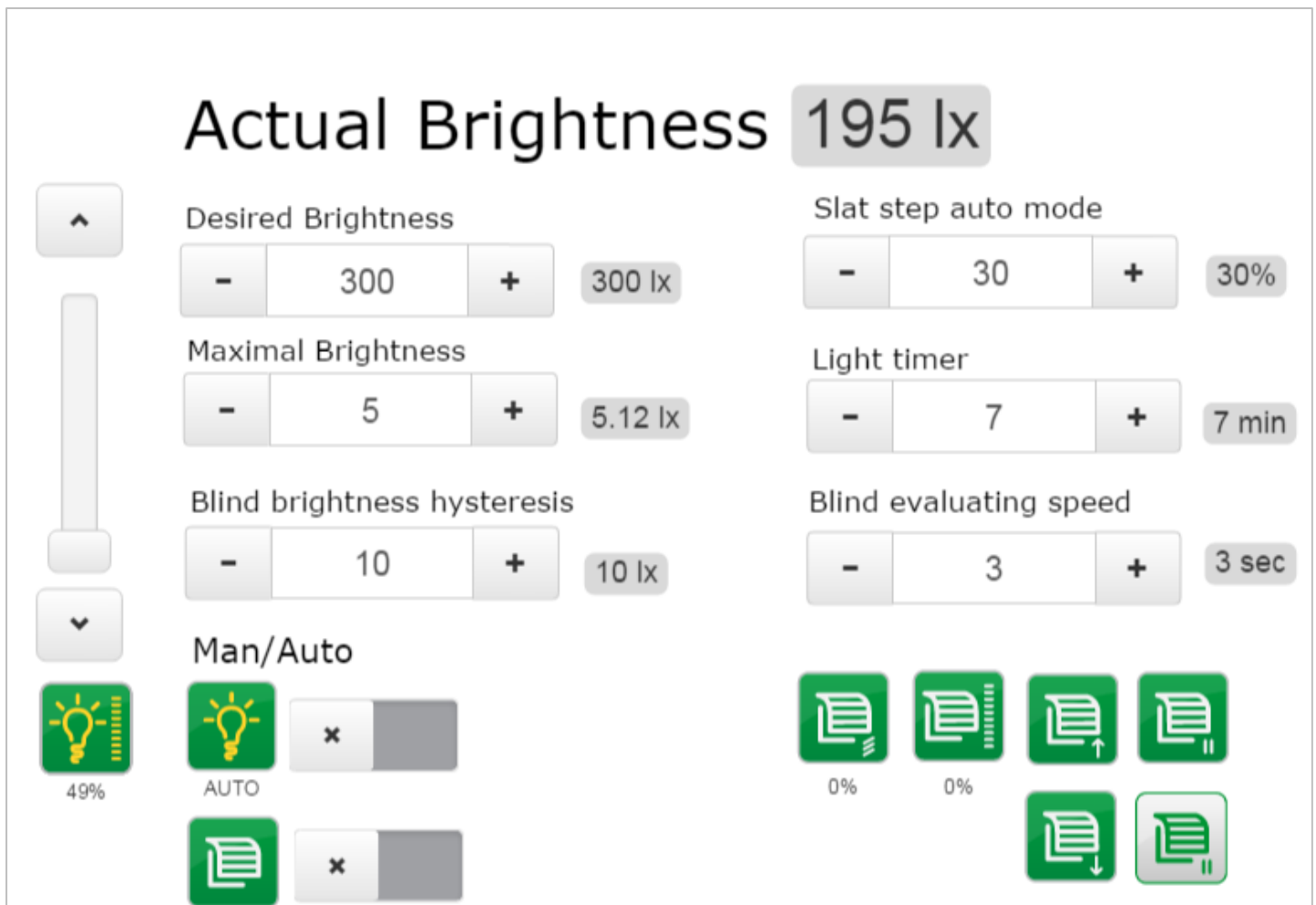
After pressing of push button the **countdown timer** is set to defined time.

Scheduled script is counting timer down.

After countdown, the object **CUS Button pressed** is changed to 0 and CLC is **enabled**.

5 User interface

The following picture shows the possibilities of user control. The user can fully control both lighting and blinds or can turn on the automatic mode to keep defined brightness level.



Picture 26: Basic user interface

Setting of parameters to customize light control according to user needs:

The **Actual brightness** shows the actual level of brightness.

The **Desired brightness** determines the minimal level of brightness. CLC keep intensity of artificial lighting at this level of brightness. Nevertheless, the actual brightness can be higher if the lighting is off and day light doesn't exceed maximal brightness.

The **Maximal brightness** determines the limit beyond which light intensity is disruptive.

The **Blind brightness hysteresis** determines how much can be desired brightness lower than actual brightness and how much can be actual brightness higher than maximal brightness before the blinds starts to move.

The **Slat step auto mode** determines minimal percent of slat tilt in one step in automatic control mode.

The **Light timer** determines how long the light stays ON without motion detection.

The **Blind evaluating speed** determines the period of blind adjustment.

NOTE: Blinds and light can be switched to manual mode and fully controlled by the user interface or physical pushbuttons.

6 Conclusion

The brightness of workplaces should be between 200 and 600 lx according the work activity. Avoid excessively small range between the desired and the maximal brightness. It should be set sufficiently large hysteresis to avoid too frequent movements of the blinds.

We recommend setting the starting behavior of the dimmer to the last known value which makes smooth transitions between daylight and artificial light.

7 Appendix

7.1 Glossary

The following table describes the acronyms and defines the specific terms used in this document.

Abbreviation	Description
CLC	Constant light control
LUA	Programming language
ETS	Engineering Tool Software
KNX	Network communications protocol for intelligent buildings
PIR	Passive infrared sensor

Table 4: specific terms

Schneider Electric Industries SAS

Head Office

35, rue Joseph Monier

92506 Rueil-Malmaison Cedex

FRANCE

www.schneider-electric.com