e.control
lumina T6 – Binary Input Module with constant light controller
SC211 006LC_12.*

Software description

This document describes the behaviour and parameterisation of the software application listed below. The application is split into logical objects according to LONMARK™-Interoperability Guidelines. The objects' behaviour is described separately in this document.

spega offers object oriented LNS plug-ins for all software objects to ensure easy configuration for the system integrator. Refer to the plug-in help system for further information.

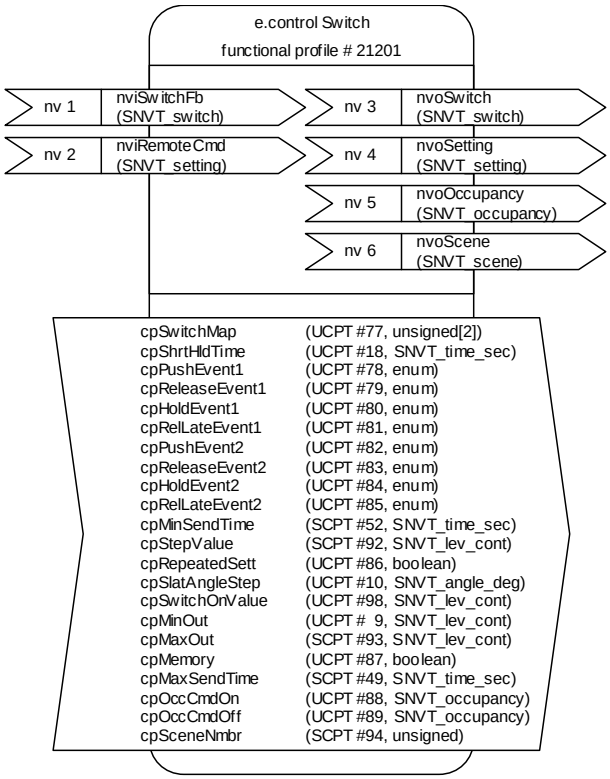
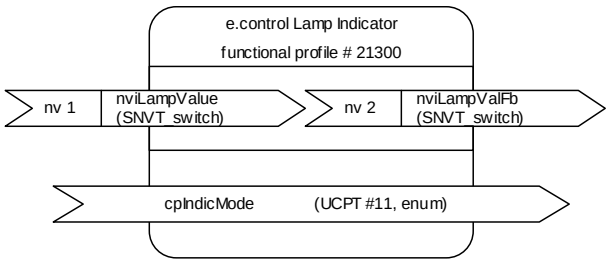
The application complies with LONMARK-Interoperability-Guidelines. With LNS based system integration tools the use of e.control resource files is recommended.

IMPORTANT: This application is applicable for spega device with hardware release 2 only. This is indicated by „HW 2" on the Neuron-ID label.

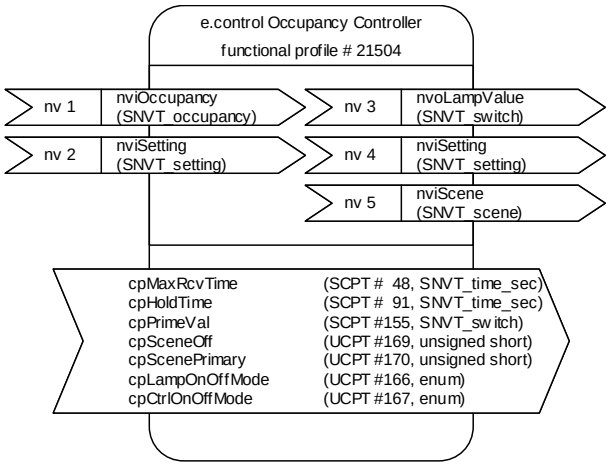
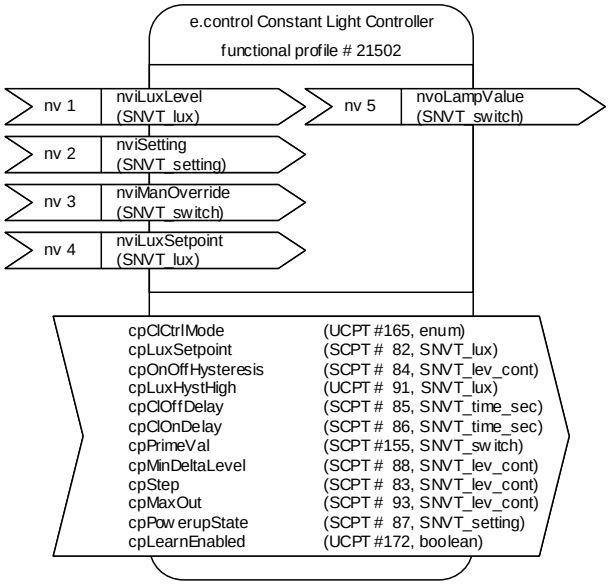
Software files

Files	SC211006LC_12.APB SC211006LC_12.NXE SC211006LC_12.XIF SC211006LC_12.XFB	Application files Interface files
Resource files Plug-ins	e.control resource files version 1.09+ available for all objects	

Overview of implemented software objects

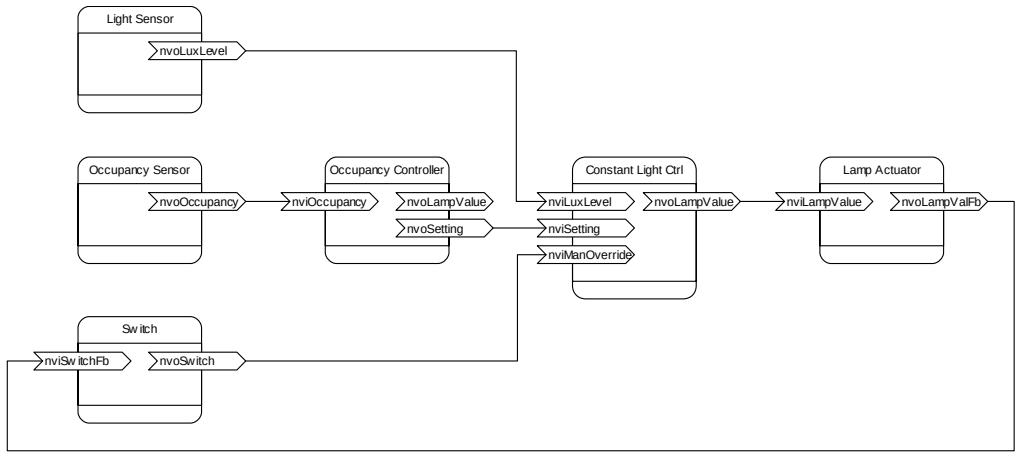
Quantity	Object	Interface
6	#21201 Switch	
2	#21300 Lamp Indicator	

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

Quantity	Object	Interface
1	#21504 Occupancy Controller	
1	#21502 Constant Light Controller	

Example

The following example shows a constant light control application combined with an occupancy controller for automatic activation on presence. An override switch button can be applied to give manual control to the user.



spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

e.control
Switch
Profile # 21201 (corresponds to LONMARK # 3200)

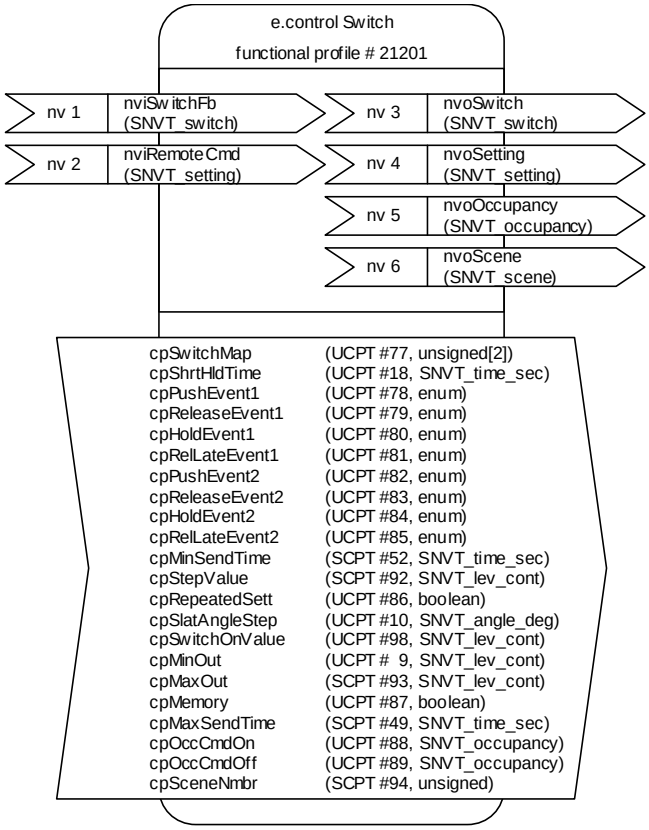
Version/Status

3.0	05.05.2003

Description

Using the switch object of the e.control system, it is possible, with conventional installation switches, to control or dim lights and other consumers as well as to control blinds and retrieve light scenes. Absence or presence reports can also be made to other system components (such as thermostats for example).

Network interface



Network variables

Input variables

`nviSwitchFb` — Checkback input for switching/dimming actuators where several switches are being used for one light circuit)

Type: SNVT_switch

Range of values: SNVT_switch

Presetting: (0.0, 0)

`nviRemoteCmd` — Input variable for forwarding central management commands to connected actuators

Type: SNVT_setting

Range of values: SNVT_setting

Presetting: (SET_OFF, 0.0, 0.00)

Output variables

`nvoSwitch` — Output of switching values for controlling the light (switching/dimming)

Type: SNVT_switch

Range of values: SNVT_switch

Presetting: (0.0, 0)

Transmission: Once in the event of a switching edge as well as in cycles (*cpMinSendTime)* on dimming commands and in cycles after the time set in *cpMaxSendTime* (heartbeat)

`nvoSetting` — Control output for sun blinds or switching functions

Type: SNVT_setting

Range of values: SNVT_setting (*function)* with the relevant meanings as follows*:

0	SET_OFF	Device off
1	SET_ON	Device on
2	SET_DOWN	Decrement value
3	SET_UP	Increment value
4	SET_STOP	Stop action

Presetting: (SET_OFF, 0, 0)

Transmission: Once in the event of alterations, in cycles (*cpMinSendTime*) during dimming procedures if *cpRepeatedSett* set

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

`nvoOccupancy` Output for reporting the room occupancy status, e.g. for air conditioning functions

Type: SNVT_occupancy

Range of values: SNVT_occupancy (s. *cpOccCmdOn* / *cpOccCmdOff*)

Presetting: *cpOccCmdOff*

Transmission: Once in the event of a switching edge as well as in cycles after the cpMaxSendTime (heartbeat)

`nvoScene` Scene output for recalling or storing allocated scenes

Type: SNVT_scene

Range of values: SNVT_scene (*function*), however with the following functions only:

0 SC_RECALL Recall scene
1 SC_LEARN Store scene
Scene number off *cpSceneNmbr*

Presetting: (SC_RECALL, 0)

Transmission: Once on pushing the button

Configuration parameters

`cpSwitchMap` Assignment of the pushbutton inputs to the object. Depending on functionality, one (switch1) or two switches (switch1 and switch2) can be allocated.

Type: typedef struct {
unsigned switch1;
unsigned switch2 } (UCPT #77)

Range of values:
0 No input assigned
n=1...x Input assigned

Presetting: {0 0] No inputs assigned

`cpShrtHldTime` Time threshold between the short and long hold function of the button

Type: SNVT_time_sec (UCPT #18)

Range of values: 0.1 ... 30.0 s

Presetting: 0.5 s (5)

`cpPushEvent1` Event on pushing the first button

Type: Enumeration (UCPT # 78)

Range of values:

0	EV_OFF	Switch off
1	EV_ON	Switch on
2	EV_DIM_DOWN	Dim down
3	EV_DIM_UP	Dim up
4	EV_STOP	Stop command
5	EV_SB_DOWN	Run down
6	EV_SB_UP	Run up
7	EV_SLAT_DWN	Slat down
8	EV_SLAT_UP	Slat up
9	EV_TOGGLE	Change over
10	EV_DIM	Dim up/down
11	EV_SB_TOGGLE	Run up/down
12	EV_SCENE_RCL	Recall scene
13	EV_SCENE_LRN	Learn scene
14	EV_NO_MSG	No event
-1	EV_NUL	Send invalid

Presetting: 14 EV_NO_MSG (no event)

`cpReleaseEvent1` Event on releasing the first button before expiry of the short hold time

Type: Enumeration (UCPT # 79)

Range of values: See cpPushEvent1

Presetting: 9 EV_TOGGLE (Switch on/off)

`cpHoldEvent1` Event on reaching the short hold time with the first button depressed

Type: Enumeration (UCPT # 80)

Range of values: See cpPushEvent1

Presetting: 10EV_TOGGLE (Dim up/down)

`cpRelLateEvent1` Event on releasing the first button after expiry of the short hold time

Type: Enumeration (UCPT # 81)

Range of values: See cpPushEvent1

Presetting: 4 EV_STOP (Send stop command)

`cpPushEvent2` Event on pushing the second button

Type: Enumeration (UCPT # 82)

Range of values: See *cpPushEvent1*

Presetting: 14 EV_NO_MSG (No event)

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

`cpReleaseEvent2` Event on releasing the second button before expiry of the short hold time

Type: Enumeration (UCPT # 83)

Range of values: See *cpReleaseEvent1*

Presetting: 14EV_NO_MSG (no event)

`cpHoldEvent2` Event on reaching the short hold time with the second pushbutton depressed

Type: Enumeration (UCPT # 84)

Range of values: See *cpHoldEvent1*

Presetting: 14EV_NO_MSG (no event)

`cpRelLateEvent2` Event on releasing the second pushbutton after expiry of the short hold time

Type: Enumeration (UCPT # 85)

Range of values: See *cpRelLateEvent1*

Presetting: 14EV_NO_MSG (no event)

`cpMinSendTime` Indicates the time between two dimming telegrams

Type: SNVT_time_sec (SCPT #52)

Range of values: 0 ... 6553.5 s

Presetting: 0.1 s (1)

`cpStepValue` Indicates the step value between two dimming telegrams

Type: SNVT_lev_cont (SCPT #92)

Range of values: 0.5 ... 100%

Presetting: 4 (2 %)

`cpRepeatedSett` Indicates whether the control output nvoSetting is to be sent in cycles in the event of dimming commands

Type: Boolean (UCPT #86)

Range of values: 0 FALSE no sending in cycles
1 TRUE sending in cycles

Presetting: TRUE (1)

`cpSlatAngleStep` Indicates the step angle for adjusting the slats in the event of _SLAT_xxx

Type: SNVT_angle_deg (UCPT #10)

Range of values: - 90° ...+ 90°

Presetting: 10° (500)

`cpSwitchOnValue` Indicates the switch-on value for switching events

Type: SNVT_lev_cont (UCPT #98)

Range of values: 0 ... 100%

Presetting: 100 % (200)

`cpMinOut` Indicates the lower threshold for dimming procedures

Type: SNVT_lev_cont (UCPT # 9)

Range of values: 0 ... 100%

Presetting: 0 % (0)

`cpMaxOut` Indicates the upper threshold for dimming and switching procedures

Type: SNVT_lev_cont (SCPT #93)

Range of values: 0 ... 100%

Presetting: 100 % (200)

`cpMemory` Indicates whether the last brightness value should be recalled in the event of starting commands (Memory)

Type: Boolean (UCPT #87)

Range of values: 0 FALSE send *cpSwitchOnV*
1 TRUE send memory value

Presetting: TRUE (1)

`cpMaxSendTime` Heartbeat time in the event of activated *nvoSwitch* and *nvoOccupancy* output network variables

Type: SNVT_time_sec (SCPT #49)

Range of values: 0 No heartbeat
0.5 ... 6535.5 s Repeat time in sec.

Presetting: 0 (no heartbeat)

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

`cpOccCmdOn` Indicates the occupancy command, sent parallel to a starting event via *nvoOccupancy*

Type: SNVT_occupancy (UCPT #88)

Range of values: See SNVT_occupancy

Presetting: OC_OCCUPIED (0) - occupied

`cpOccCmdOff` Indicates the occupancy command sent via *nvoOccupancy,* parallel to a switching-off event

Type: SNVT_occupancy (UCPT #89)

Range of values: See SNVT_occupancy

Presetting: OC_UNOCCUPIED (1) - unoccupied

`CpSceneNmbr` Scene number

Type: unsigned short (SCPT #94)

Range of values: 0 Scene invalid
1...255 Scene number

Presetting: 0

Functional description

The switch object sends – as a function of the parameter setting – switching or control commands for starting up light or blind actuators as well as for controlling scenes or manual occupancy reporting.

Conventional switches, pushbuttons or other floating contacts can be connected to the pushbutton interface. It is possible to select the assignment of the pushbutton inputs to the relevant software objects.

Assignment of the inputs to the switch object

The switch object supports operation with one or two pushbuttons. Assignment takes place by specifying the relevant input port in *cpSwitchMap*. If no hardware input has been assigned, "0" should be selected.

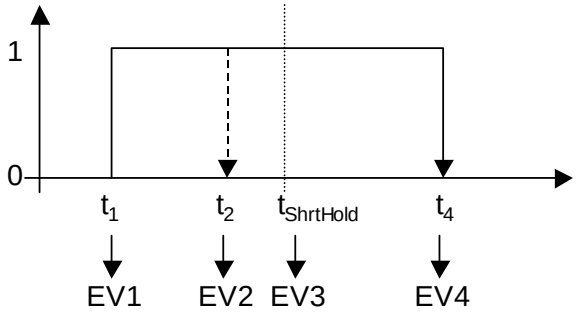
E.g.: {3 4} assigns inputs 3 (1st button) and 4 (2nd button) to the object, i.e. two-button station function

{2 0} assigns the object input 2 as the first button, a second input is not assigned

Assignment of actions to pushbutton events

1. Pushbutton events

The button-pushing process generates up to four different events, depending on the profile.



EV1 Pushing the pushbutton
EV2 Releasing the pushbutton before expiry of the short hold time
EV3 Expiry of the short hold time
EV4 Releasing the pushbutton after expiry of the short hold time

By assigning certain actions which are performed by the object during each respective event, the functionality of the object can be freely configured.

Assignment takes place on the basis of the following parameters:

Parameter	**Description of the event**
cpPushEvent1	Pushing the first button
cpReleaseEvent1	Releasing the first button before expiry of the short hold time
cpHoldEvent1	Reaching the short hold time with the first button depressed
cpRelLateEvent1	Releasing the first button after expiry of the short hold time
Parameter	**Description of the event**
cpPushEvent2	Pushing the second button
cpReleaseEvent2	Releasing the second button before expiry of the short hold time
cpHoldEvent2	Reaching the short hold time with the second button depressed
cpRelLateEvent2	Releasing the second button after expiry of the short hold time

2. Selection of the short hold time

By specifying the short hold time $t_{ShrtHold}$ in the parameter *cpShrtHldTime*, the period of time, during which duration the action of event EV3 (*cpHoldEvent*) is executed, is selected. If this short hold time is eliminated ("0"), events EV2 and EV3 (*cpReleaseEvent*, *cpHoldTime*) are omitted. In this case, the object only processes edge change via events EV1 (*cpPushEvent*) and EV4 (*cpRelLateEvent*).

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

3. Selection of the action to be executed

The following actions, which are executed if the previously mentioned events occur, are available to the object:

Action	Function
EV_OFF “Switch off “	The object sends a switch-off command as well as the assigned occupancy status: *nvoSwitch* = (0, 0) *nvoSetting* = (SET_OFF, 0, 0) *nvoOccupancy* = *cpOccCmdOff*
EV_ON “Switch on “	The object sends a switch-on command as well as the assigned occupancy status: *nvoSwitch* = (X*, 1) *nvoSetting* = (SET_ON, X*, 0) *nvoOccupancy* = *cpOccCmdOn* * The switch-on value X is determined as follows: cpMemory: TRUE FALSE Switch-on: Memory cpSwitchOnV.
EV_DIM_DOWN “Dim down “	The object sends a dim value decremented on the basis of the *cpStepValue* in cycles (*cpMinSendTime*). The dimming process is limited by *cpMinOut* (“0%“ results in a switch-off command in the event of the limit being reached. The transmission response of *nvoSetting* can be parameterised. *nvoSwitch* = (Value-cpStepValue, 1) *nvoSetting* = (SET_DOWN, cpStepValue, 0)* (or) = (SET_DOWN, 0, 0)** * in cycles, if *cpRepeatedSett* TRUE ** once, if *cpRepeatedSett* FALSE
EV_DIM_UP “Dim up “	The object sends a dim value incremented by the *cpStepValue* in cycles (*cpMinSendTime*). The dimming procedure is limited by *cpMaxOut*. The transmission response of *nvoSetting* can be parameterised. *nvoSwitch* = (Value+cpStepValue, 1) *nvoSetting* = (SET_UP, cpStepValue, 0)* (or) = (SET_UP, 0, 0)** * in cycles, if *cpRepeatedSett* TRUE ** once, if *cpRepeatedSett* FALSE
EV_STOP “Stop“	The object sends a stop command *nvoSetting* = (SET_STOP, 0, 0)
EV_SB_DOWN “Run down“	The object sends a run-down command for sunblind systems. *nvoSetting* = (SET_DOWN, 100%, 0)
EV_SB_UP “Run up “	The object sends a run-up command for sunblind systems. *nvoSetting* = (SET_UP, 100%, 0)
EV_SLAT_DOWN “Slat down “	The object sends a slat-down command for adjusting sunblind systems. *nvoSetting* = (SET_DOWN, 0%, *cpSlatAngleStep*)
EV_SLAT_UP “Slat up “	The object sends a slat turning command for adjusting sunblind systems. *nvoSetting* = (SET_UP, 0%, *cpSlatAngleStep*)
EV_TOGGLE “Change over “	The object alternately sends a switch-off or switch-on command and the assigned occupancy status: (See EV_OFF / EV_ON)
EV_DIM “Dim up/down“	The object alternately sends a dim up/dim down command: (s. EV_DIM_DOWN / EV_DIM_UP)
EV_SB_TOGGLE “Run up/down“	The object alternately sends a run-up/run-down command: (s. EV_SB_DOWN / EV_SB_UP)
EV_SCENE_RCL ”Recall scene”	The object sends a command to recall a scene. In addition, a switch-on telegram is sent to *nvoSwitch*. *nvoSwitch* = (100%, 1) *nvoScene* = (SC_RECALL, *cpSceneNmbr*)
EV_SCENE_LRN “Learn scene “	The object sends a command to learn a scene. *nvoScene* = (SC_LEARN, *cpSceneNmbr*)
EV_NO_MSG ”No function”	The object does not send a command. However, previous cyclical commands are completed.
EV_NUL “Send invalid “	The object sends an “invalid” command, e.g. for cancelling priority commands. *nvoSwitch* = (0, -1) *nvoSetting* = (SET_NUL, 0, 0) *nvoOccupancy* = *cpOccCmdOff*

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

4. Configuration for standard functions

The parameterisation of the events for all standard control functions with one or two-button stations is shown below.

a) Switching and pushbutton functions

Changing over with one pushbutton				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	TOGGLE	NO_MSG	NO_MSG	NO_MSG
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

Changing over with a two-button station				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	ON	NO_MSG	NO_MSG	NO_MSG
Button 2	OFF	NO_MSG	NO_MSG	NO_MSG

Changing over with a switch				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	TOGGLE	TOGGLE	NO_MSG	NO_MSG
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

Switching on with a pushbutton				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	ON	NO_MSG	NO_MSG	NO_MSG
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

Switching off with a pushbutton				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	OFF	NO_MSG	NO_MSG	NO_MSG
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

b) Binary contacts

Make contact				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	ON	OFF	NO_MSG	OFF
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

Break contact				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	OFF	ON	NO_MSG	ON
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

c) Dimming functions

Dimming with pushbutton				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	NO_MSG	TOGGLE	DIM.	STOP
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

Dimming with two-button station				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	NO_MSG	ON	DIM_UP	STOP
Button 2	NO_MSG	OFF	DIM_DOWN	STOP

d) Sunblind system functions

Sunblind system with a pushbutton				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	STOP	NO_MSG	SB_TOG.	NO_MSG
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

Sunblind system with two-button station				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	SLAT_UP	NO_MSG	SB_UP	NO_MSG
Button 2	SLAT_DWN	NO_MSG	SB_DOWN	NO_MSG

e) Scene function

Scene pushbutton				
	pushEvent	releaseEvent	holdEvent	relLateEvent
Button 1	NO_MSG	SCENE_RCL	NO_MSG	SCENE_LRN
Button 2	NO_MSG	NO_MSG	NO_MSG	NO_MSG

Use of the heartbeat

If the object is used as a signalling device for transmitting binary statuses (e.g. window contact) or assignments, the use of a heartbeat may be necessary. The corresponding time can be adjusted via *cpMaxSendTime* and causes the network variables *nvoSwitch* and *nvoOccupancy* to be sent again after the set period of time.

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

Use of nviRemoteCmd

It is possible via *nviRemoteCmd* to send commands, e.g. from a central control station, via the pushbutton to the connected actuators. In this way, for example, commands can be sent to a group of actuators, via *nviRemoteCmd* from the central control station to the (group) pushbutton, which then sends this command to the connected actuators via *nvoSwitch* and *nvoSetting*. All setting commands are forwarded unchanged. In the case of SET_NUL, SET_OFF and SET_ON, the analog command is also sent via *nvoSwitch*:

nvoSwitch.state = nvoSetting.function
nvoSwitch.value = nvoSetting.setting

Reset response

nviSwitchFb is polled in order to determine the current lighting status of the actuators.

If a heartbeat time (*cpMaxSendTime*) is set, the following action is executed – depending on the status of the first input – in order to adjust the network variables to suit the input status (for window contacts, occupancy sensors etc.):

First input opened:	*cpRelLateEvent1*
First input closed:	*cpPushEvent1*

Troubleshooting

No troubleshooting is required

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

e.control
Lamp Indicator
Profile # 21300 (corresponds to LonMark # 3040)

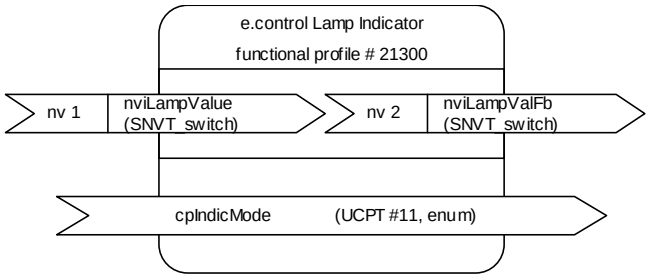
Version/Status

2.0	02.05.2003

Description

The object lamp indicator serves to display certain system statuses. The method of signalling can be adjusted via the parameters.

Network interface



Network variables

Input variable

`nviLampValue` Switching input for indicator lights (or similar consumers)

Type: SNVT_switch

Range of values: SNVT_switch

Presetting: (0.0, 0)

Output variable

`nvoLampValFb` Checkback output of status

Type: SNVT_switch

Range of values: SNVT_switch

Presetting: (0.0, 0)

Transmission: After each variation

Configuration parameters

`cpIndicMode` Mode of operation of display

Type: Enumeration (UCPT #11)

Range of values:

0	LED_SYNC	Light follows input
1	LED_INVERT	Light is inverted
2	LED_BLINK	Light flashes state = 1
3	LED_OFF	Light always off
4	LED_ON	Light always on
0xFF	LED_NUL	Value not valid

Presetting: LED_SYNC

Functional description

The switching output is operated on the basis of the input variable *nviLampValue*, as a function of the configuration parameter, as follows:

Input variable *nviLampValue*	Configuration parameter. *cpIndicMode*	Switching output
.state = 0; .value = don't care	LED_SYNC	OFF
	LED_INVERT	ON
	LED_BLINK	OFF
	LED_OFF	OFF
	LED_ON	ON
.state = 1; .value = don't care	LED_SYNC	ON
	LED_INVERT	OFF
	LED_BLINK	FLASHES
	LED_OFF	OFF
	LED_ON	ON

Reset response

Reset the variables to the default values.

Troubleshooting

No troubleshooting is required.

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

e.control
Occupancy Controller
Profile # 21504 (corresponds to LONMARK #3071)

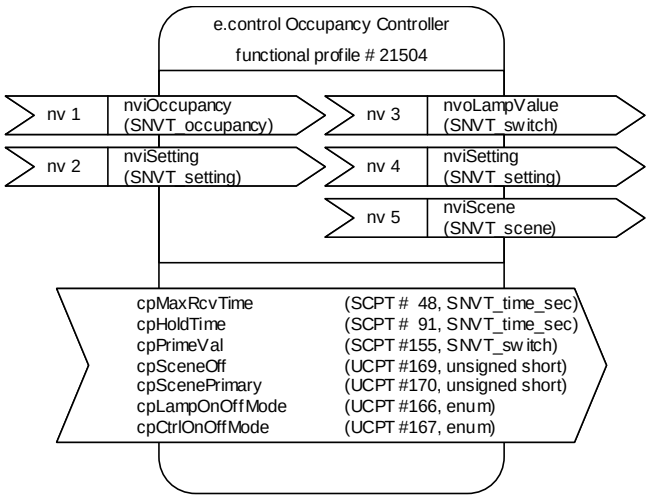
Version/Status

1.00	24.05.2004

Description

The occupancy controller object of the e.control system is used for occupancy-dependent switching of lighting equipment or of other controllers (e.g. constant light controllers).

Network interface



Network variables

Input variables

nviOccupancy — Input for the occupancy status of the room (occupancy)

Type: SNVT_occupancy

Range of values:

0	OC_OCCUPIED	occupied
1	OC_UNOCCUPIED	unoccupied
2	OC_BYPASS	occupied
3	OC_STANDBY	unoccupied

Presetting: OC_UNOCCUPIED (1)

nviSetting — Control input of controller for switching-on or switching-off

Type: SNVT_setting

Range of values: SNVT_setting with the following significance*:

SET_OFF	Controller off
SET_ON	Controller on

Presetting: (SET_ON, 0, 0.00)

Output variables

nvoLampValue — Lamp switching output

Type: SNVT_switch

Range of values: SNVT_switch with following values:
{0.0 0} if room is unoccupied
cpPrimeVal if room is occupied

Presetting: (0.0, 0)

Transmission: After change of status

nvoSetting — Switching output for further controllers (e.g. constant light controllers)

Type: SNVT_setting

Range of values: SNVT_setting with following values:
SET_OFF if room is unoccupied
SET_ON if room is occupied

Presetting: (SET_OFF, 0.0, 0.00)

Transmission: After change of status

nvoScene — Scene output

Type: SNVT_scene

Range of values: SNVT_scene with following values:
(RECALL *cpSceneOff*) if unoccupied
(RECALL *cpScenePrim*) if occupied

Presetting: (SC_RECALL 0)

Transmission: After change of status

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

Configuration parameters

`cpMaxRvcTime` Duration of validity for "occupied" telegrams on *nviOc-Occupancy*; serves parallel-connection of several sensors

Type: SNVT_time_sec (SCPT #48)

Range of values: 0.0 unlimited duration of validity
0.1...6553.5s duration of validity

Presetting: 0.0 (0)

`cpHoldTime` Delay time before an "unoccupied" telegram causes switch-off of the output NVs.

Type: SNVT_time_sec (SCPT #91)

Range of values: 0.0 no delay in switch-off
0.1 ...6553.5s delay

Presetting: 600.0 s (6000) [= 10 min]

`cpPrimeVal` Starting value of the lamp switching output *nvoOcLampValue*

Type: SNVT_switch (SCPT #155)

Range of values: see SNVT_switch.

Presetting: (100.0 1)

`cpSceneOff` Scene if room is unoccupied

Type: unsigned short (UCPT #169)

Range of values: 0 invalid
1...255 scene number

Presetting: 1

`cpScenePrimary` Scene if room is occupied

Type: unsigned short (UCPT #170)

Range of values: 0 invalid
1...255 scene number

Presetting: 2

`cpLampOnOff-Mode` Behaviour of lamp switching output if there is a change in room occupancy

Type: enumeration (UCPT #166)

Range of values: 0 OCL_ONOFF sends if occupied and unoccupied
1 OCL_OFFONLY sends only upon leaving the room (unoccupied)

Presetting: OCL_ONOFF

`cpCtrlOnOff-Mode` Behaviour of controller output if there is a change in room occupancy

Type: enumeration (UCPT #167)

Range of values: 0 OCL_ONOFF sends if occupied and unoccupied
1 OCL_OFFONLY sends only upon leaving the room (unoccupied)

Presetting: OCL_ONOFF

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

Functional description

The occupancy controller object of the e.control system is used for occupancy-dependent switching of lighting equipment or of other controllers (e.g. constant light controllers). For this purpose it gets the room occupancy status which is determined by one or more occupancy sensors. In case of changes, the lighting can be operated directly, activated or deactivated and scenes can be called up.

Telegram evaluation in the event of several occupancy sensors

For the parallel evaluation of several occupancy sensors via *nviOccupancy*, it is necessary to cyclically receive the occupancy telegram of each sensor. At the controller, the validity of an "occupied" telegram is also assigned a duration of validity (*cpMaxRcvTime*) which must be higher than the repetition rate of the sensors.

Deactivation of the controller

The controller can be deactivated via the input *nviSetting* (temporarily). All output variables are switched off in this case. In the subsequent stages the controller no longer reacts to the change in room occupancy until it has been re-activated again.

Adaptation of the behaviour of the outputs

The behaviour of the output variables *nvoLampValue* and *nvoSetting* can be adapted via the parameters *cpLamp-OnOffMode* and. *cpCtrlOnOffMode* in such a way that semi or fully automatic operation becomes possible. Furthermore, the scenes to be activated can be determined:

nvoLampValue		
Status	*OCL_ONOFF*	*OCL_OFFONLY*
"Occupied"	*cpPrimeVal*	---
"Unoccupied"	(0.0 0)	(0.0 0)

nvoSetting		
Status	*OCL_ONOFF*	*OCL_OFFONLY*
"Occupied"	(SET_ON x* 0.00) * *cpPrimeVal.value*	---
"Unoccupied"	(SET_OFF 0.0 0.00)	(SET_OFF 0.0 0.00)

nvoScene	
Status	
"Occupied"	*(SC_RECALL cpScenePrimary)*
"Unoccupied"	*(SC_RECALL cpSceneOff)*

Reset response

The controller adopts the default status.

Troubleshooting

--

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

e.control
Constant Light Controller
Profile # 21502 (corresponds to LONMARK # 3050)

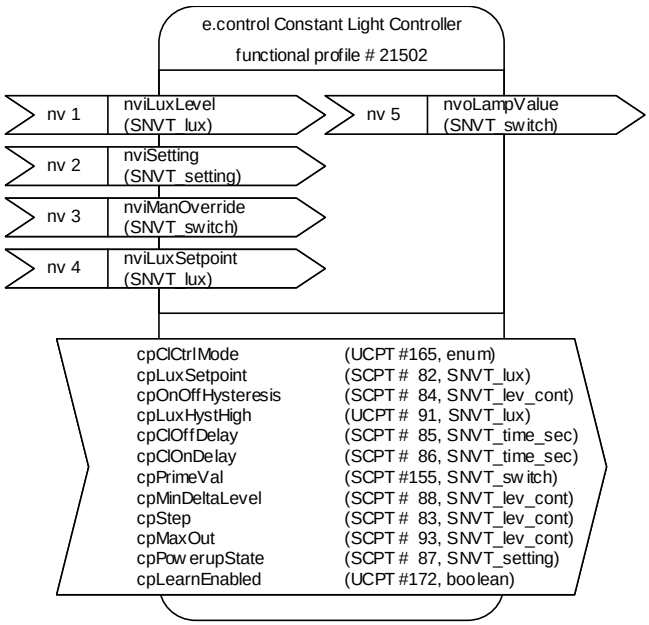
Version/Status

1.00	24.05.2004

Description

The constant light controller object of the e.control system is used for brightness-related control of lighting equipment. The controller can be either used for switch-on or switch-off or for constant light control.

Network interface



Network variables

Input variables

`nviLuxLevel` — Input for room brightness

Type:	SNVT_lux
Range of values:	0 ... 65535 lux
Presetting:	0 lux

`nviSetting` — Control input for activation and deactivation of the controller

Type:	SNVT_setting
Range of values:	SNVT_setting with the following significance*:

SET_OFF	Controller inactive
SET_ON	Controller active
SET_UP	Increase setpoint
SET_DOWN	Decrease setpoint

Presetting: acc. to *cpPowerupState*

`nviManOverride` — Manual override of the controller

Type:	SNVT_switch		
Range of values:	.value	= 0 ... 100%	
	.state	= 0	Light off
		= 1	Light on
		= -1	Control active
Presetting:	(0.0, -1) Control active		

`nviLuxSetpoint` — Permanent setpoint input (overwrites *cpLuxSetpoint*)

Type:	SNVT_lux
Range of values:	0 ... 1500 lux
Presetting:	0 lux (inactive)

Output variables

`nvoLampValue` — Output for activation of light switching or dimming actuators

Type:	SNVT_switch
Range of values:	SNVT_switch
Presetting:	(0.0, 0)
Transmission:	After every change of the manipulated variable > *cpMinDeltaLevel*

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

Configuration parameters

cpClCtrlMode — Operating mode of the controller

Type: Enumeration (UCPT #165)

Range of values:
0 CL_CONSTANT Constant light
1 CL_ONOFF Switching mode

Presetting: 0 CL_CONSTANT

cpLuxSetpoint — Setpoint for constant light mode or minimum brightness in switching mode

Type: SNVT_lux (SCPT #82)

Range of values: SNVT_lux

Presetting: 500 lux

cpOnOffHyste-resis — Switch hysteresis for switching lighting on and off in constant light mode

Type: SNVT_lev_cont (SCPT #84)

Range of values:
0 No automatic switching
1 ... 100 % below/above setpoint

Presetting: 0

cpLuxHystHigh — Upper switch-off value of room brightness in switching mode (is learnt automatically)

Type: SNVT_lux (UCPT #91)

Range of values: SNVT_lux

Presetting: 800 lux

cpClOffDelay — Switch-off delay on the dimming value of "0%" being attained in the constant light mode or after the switch-off limit has been reached in the switching mode

Type: SNVT_time_sec (SCPT #85)

Range of values: 0.0 ...6553.5 s

Presetting: 600.0 s (6000)

cpClOnDelay — Switch-on delay after the lower hysteresis value is reached in the constant light mode or after the value falls below the setpoint in the switching mode.

Type: SNVT_time_sec (SCPT #86)

Range of values: 0.0 ...6553.5 sec.

Presetting: 5.0 sec.

cpPrimeVal — Switch-on or starting value of lighting.

Type: SNVT_switch (SCPT #155)

Range of values: SNVT_switch

Presetting: (50.0% 1) (100 1)

cpMinDeltaLevel — Indicates the minimum deviation required to send a new dimming telegram

Type: SNVT_lev_cont (SCPT #88)

Range of values:
0 send each change immed.
0.5 ...10 % Minimum change

Presetting: 0.5 % (1)

cpStep — Dimming step size in constant light mode

Type: SNVT_lev_cont (SCPT #83)

Range of values: 0.5 ...10 %

Presetting: 2 % (4)

cpMaxOut — Maximum value of lighting in constant light mode (brightness restriction)

Type: SNVT_lev_cont (SCPT #93)

Range of values: 0.5 ...100 %

Presetting: 100 % (200)

cpPowerupState — Specifies status of the controller after power restoration

Type: SNVT_setting (SCPT #87)

Range of values: SNVT_setting with following significance*:

SET_OFF Controller off
SET_ON Controller on

Presetting: (SET_OFF, 0, 0.00)

cpLearnEnabled — Enabling learning of the temporary room brightness as a permanent setpoint (overwrites *cpLuxSetpoint*)

Type: Boolean (UCPT #172)

Range of values:
0 FALSE Learning disabled
1 TRUE Learning enabled

Presetting: 0 FALSE

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

Functional description

The constant light controller object of e.control system is used for brightness-related control of lighting equipment. The controller can be either used for switch-on or switch-off (switching mode) or for constant light control.

The controller is in all circumstances activated via *nviSetting* with the telegrams SET_ON or SET_OFF. Alternatively, the IR remote control buttons "0" and "I" which have the same significance can be used. The behaviour in the event of power restoration can be set via *cpPowerupState*.

The two operating modes are separately described below.

Constant light mode
(cpClCtrlMode = CL_CONSTANT)

In the constant light mode, the controller on being activated controls via *nviSetting.function* = SET_ON the linked dimmable lighting via the output *nvoLampValue* in such a way that the preset setpoint *cpLuxSetpoint* is reached and maintained.

Switch-on/switch-off behaviour

The switch-on and switch-off behaviour of the controller can be controlled via the parameters *cpOnOffHysteresis*, *cpClOffDelay* and *cpClOnDelay*.

a) No automatic switch-on/switch-off (cpOnOffHysteresis = 0)

When the controller is activated, it sends the value *cpPrimeVal* to the lamp actuators and commences controlling between the dimming values 0% and *cpMaxOut*. Any relays on the fluorescent tubes remain permanently switched on during the activation (*nvoLampValue.state* = 1). *cpClOffDelay* and *cpClOnDelay* have no significance.

b) Automatic switch-on/switch-off (cpOnOffHysteresis > 0%)

During active operation, the controller checks the switch-on condition

nviLuxLevel < *cpLuxSetpoint* * (1 - *cpOnOffHystersis* / 2)

to switch on the light with *cpPrimeVal* after the time set in *cpClOnDelay*.

If the switch-off conditions

nviLuxLevel > *cpLuxSetpoint* * (1 + *cpOnOffHystersis* / 2) und *nvoLampValue.value* = 0%

become valid for the time set in *cpClOffDelay*, the controller switches the light off.

Change in the setpoint

The setpoint of the constant light control can be changed either temporarily, i.e. only for the duration of an activity phase, or permanently in many ways.

a) Temporary setpoint change by the user

The setpoint can be adapted by the control commands SET_UP or SET_DOWN via *nviSetting*. For this purpose, the current output value is modified by the value indicated in *nviSetting.setting* and the new brightness value which is being set is stored as the temporary setpoint. On deactivating the controller via SET_OFF, the setpoint is reset.

The adaptation can also be attained via the IR remote control with the "Arrow Up" and "Arrow Down" keys. With each operation, the dimming value is modified by 5%.

b) Permanent change

The setpoint can also be changed permanently at *nviLuxSetpoint* by a telegram, e.g. from a central office. The corresponding configuration value *cpLuxSetpoint* is overwritten in this case.

A local modification of the setpoint is likewise possible with the help of the IR remote control if the function is enabled via *cpLearnEnabled*. On operating the "L" button, the current brightness value at *nviLuxLevel* is stored as a new permanent setpoint. *cpLuxSetpoint* is overwritten accordingly.

Switching mode
(cpClCtrlMode = CL_ONOFF)

In the switching mode, the controller on being activated controls the lighting via the *nviSetting.function* = SET_ON on the basis of the measured room brightness so that at least the setpoint *cpLuxSetpoint* is reached. If the brightness falls below the setpoint, the artificial light is switched on. If however the limit brightness *cpLuxHystHigh* is reached with the lighting switched on, the lighting switches itself off.

Switch-on/switch off behaviour

The switch-on and switch-off behaviour of the controller can be controlled via the parameters *cpLuxSetpoint, cpLuxHystHigh*, *cpClOffDelay* and *cpClOnDelay*.

a) *Automatic switch-on/switch-off*

During active operation, the controller checks the switch-on condition

nviLuxLevel < *cpLuxSetpoint*

to switch on the light with *cpPrimeVal* after the time set in *cpClOnDelay*.

If the switch-off conditions

nviLuxLevel > *cpLuxHystHigh*

become valid for the time set in *cpClOffDelay*, the controller switches the light off.

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com

Change in the setpoint (minimum brightness)

The minimum brightness in the switching mode can be changed permanently.

a) *Permanent change*

The setpoint can be changed permanently at *nviLuxSetpoint* by a telegram, e.g. from a central office. The corresponding configuration value *cpLuxSetpoint* is overwritten in this case. The limit brightness *cpLuxHystHigh* is also shifted.

A local modification of the setpoint is likewise possible with the help of the IR remote control if the function is enabled via *cpLearnEnabled*. On operating the "L" button, the current brightness value at *nviLuxLevel* is stored as a new permanent setpoint. *cpLuxSetpoint* and cpLuxHystHigh are overwritten accordingly.

Learning function of the limit brightness

The limit brightness is dynamically adapted in the switching mode For this purpose, the controller measures the difference in brightness 10 s after automatic switch-on or switch-off of the lighting. If this value, which corresponds to the portion of artificial light in the room brightness, is higher than the difference between the parameters *cpLuxHystHigh* and *cpLuxSetpoint*, the limit brightness is calculated again:

cpLuxHystHigh = cpLuxSetpoint + portion of artificial light

This function helps to avoid a likely self-excited cyclical switch-on and switch-off during operation through incorrect parameterisation.

Note: The learning takes place only in case of automatic circuits and not through manual switching telegrams via nviManOverride. When changing the lighting, especially when reducing the capacity, the value of the threshold brightness can be learned again by resetting the parameter cpLuxHystHigh.

Reset response

The controller polls the room brightness (*nviLuxLevel*) and after 2 seconds takes on the status stipulated by the parameter *cpPowerupState*.

Troubleshooting

---

spelsberg gebäudeautomation gmbh + co. kg • zechenstr. 70 • 47443 moers • germany • fon +49 2841 88049-0 • fax +49 2841 88049-49
internet http://www.spega.com • email info@spega.com