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.

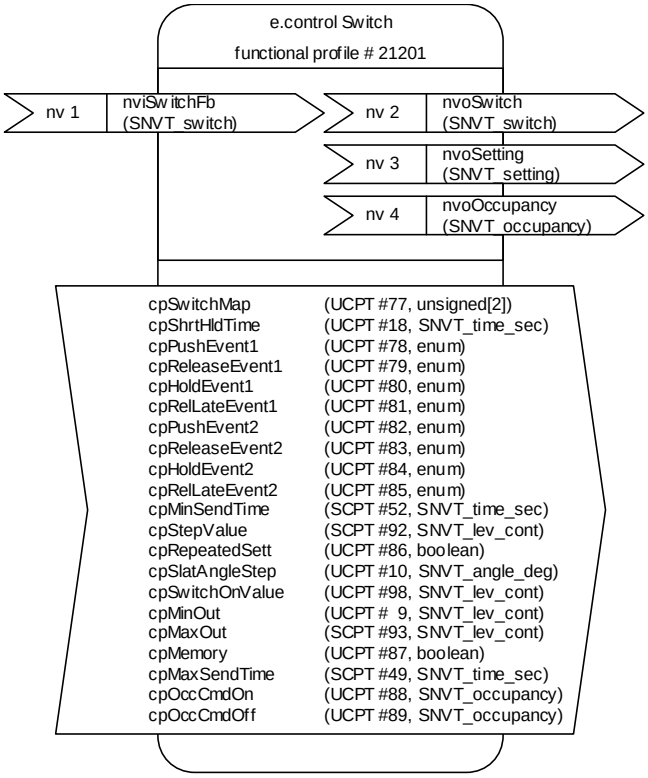
The application complies with LONMARK-Interoperability-Guidelines. With LNS based system integration tools the use of e.control resource files is recommended.

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

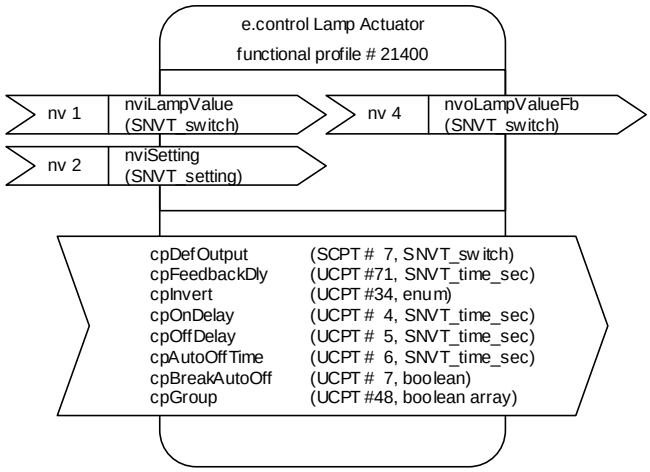
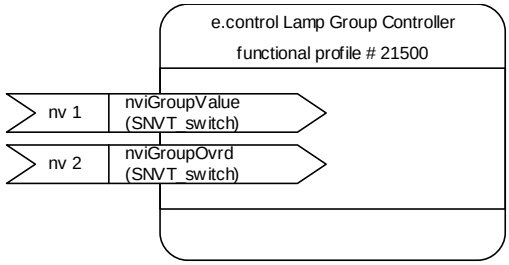
Application files

Files	SC111008SC_12.APB SC111008SC_12.NXE SC111008SC_12.XIF SC111008SC_12.XFB	Application files Interface files
Resource files Plug-ins	e.control resource files v.1.02+ available for all objects	

Overview of implemented software objects

Quantity	Object	Interface
8	#21201 Switch	

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
8	# 21400 Lamp Actuator	
1	# 21500 Lamp Group Controller	

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)

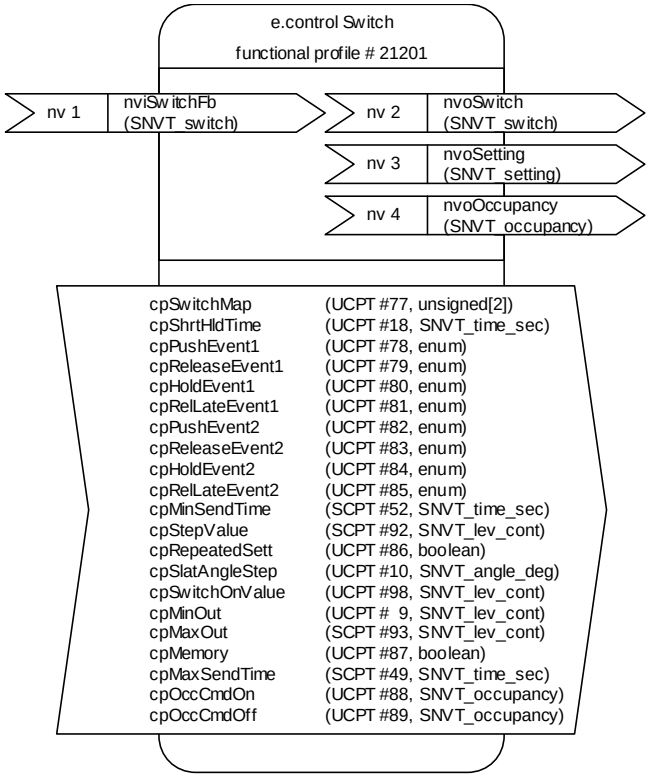
Version/Status

3.2	01.10.2003

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. 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)

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 (*cpMin-SendTime)* 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

`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)

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

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
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)

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)

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

`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)

`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

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 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 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.

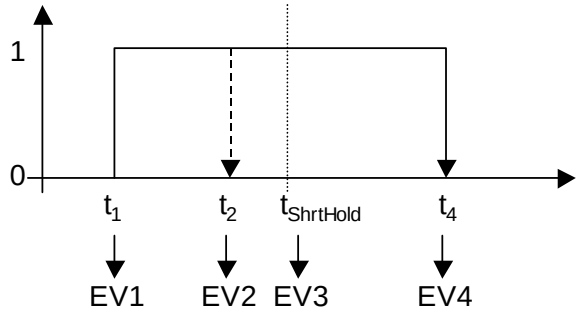
E.g.: {3 4} assigns inputs 3 (1st button) and 4 (2nd button) to the object, i.e. two-button station function

{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)

Action	Function
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_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

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.

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 Actuator (with Relays)
Profile # 21400 (corresponds to LonMark # 3040)

Version/Status

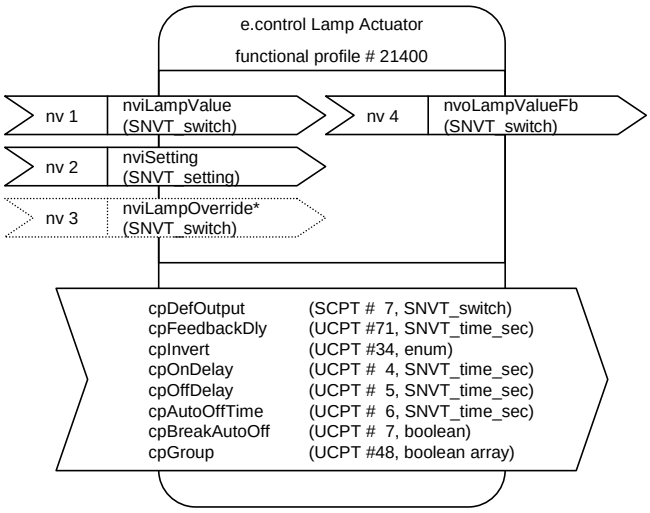
2.0	30.04.2003

Description

The lamp actuator object of the e.control system is used for switching actuators. The object has a switch-on/switch-off delay as well as an automatic time function (stairwell light).

In the automatic mode, the actuator switches off automatically after a parameterised time. Moreover, it is possible to determine via the configuration whether a switch-off command is being executed within this time.

Network interface



* located on Lamp Group Controller (#21500)

Network variables

Input variables

`nviLampValue` Switching input of the actuator

Type: SNVT_switch

Range of values: SNVT_switch

Presetting: (0.0, 0)

`nviSetting` Control input of the actuator

Type: SNVT_setting

Range of values: SNVT_setting (*function)* with the following meanings*:

0	SET_OFF	Off (with memory)
1	SET_ON	Switch on memory
2	SET_UP	Dim up
3	SET_DOWN	Dim down
?	SET_NUL	Cancel

* Explanation: see back

Presetting: (SET_OFF, 0, 0)

`nviLampOverride` Control input of the actuator (located on group controller)

Type: SNVT_switch

Range of values: SNVT_switch

State	Value	Status	Block
0	?	off	yes
1	X	on	yes
-1	?	unchang.	no

Presetting: (0.0, -1)

Output variables

`nvoLampValueFb` Checkback output for tactile sensors when using several buttons for a light circuit (so-called two-way/intermediate switch circuits)

Type: SNVT_switch

Range of values: SNVT_switch

Presetting: (0.0, 0)

Transmission: Delayed after new 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

cpDefOutput Switch-on value after reset / power restoration

Type: SNVT_switch (SCPT #7)

Range of values: SNVT_switch

Presetting: OFF (0.0, 0)

cpFeedbackDly Feedback delay

Type: SNVT_time_sec (UCPT #71)

Range of values: 0 ...6553.5 s

Presetting: 0.3s (3)

cpInvert Inverting of the relay operation (only in the case of switching actuators)

Type: Enumeration (UCPT #34)

Range of values:
0 LO_NORM Make contact
1 LO_INVERT Break contact

Presetting: 0 LO_NORM

cpOnDelay Switch-on delay of the actuator

Type: SNVT_time_sec (UCPT #4)

Range of values:
0 no delay
0 ...6553.5 s delay

Presetting: No delay (0)

cpOffDelay Switch-off delay of the actuator or warning duration for an automatic light

Type: SNVT_time_sec (UCPT #5)

Range of values:
0 no delay
0 ...6553.5 s delay

Presetting: No delay (0)

cpAutoOffDly Stairwell light duration

Type: SNVT_time_sec (UCPT #6)

Range of values:
0 Automatic mode deactivated
0.1...6553.5 secs Stairwell light duration

Presetting: Deactivated (0)

cpBreakAutoOff Indicates whether switch-off commands are executed during the stairwell light duration

Type: Boolean (UCPT #7)

Range of values:
0 FALSE Off command Aus-Command is ignored
1 TRUE Carry out Off command

Presetting: 0 FALSE

cpGroup Object belongs to a group (21500) (only in case of multiple actuators)

Type: Boolean[3] (UCPT #48)

Range of values: Boolean[3] (max. 3 group objects)
0 FALSE Group not assigned
1 TRUE Group assigned

Presetting: {TRUE, FALSE, FALSE} (member of first group only)

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 actuator object can be operated within the following functions:

Normal operation	The actuator is operated as a switching actuator for electrical consumers
Stairwell light	The actuator switches off automatically after a set time

Normal operation

The actuator is activated via *nviLampValue*. The field *nviLampValue.state* determines the actuator status:

NV input *nviLampValue.state*	Switching actuator
0	OFF (opened)*
1	ON (closed)**

* Switch-off time delayed in acc. with *cpOnDelay*
** Switch-on point delayed in acc. with *cpOffDelay*

The switching value is displayed as a feedback signal via *nvoLampValueFb*. Here either 0 (0%) or 200 (100%) is displayed in the field *nvoLampValueFb.value*. The feedback signal is delayed by the time given in *cpFeedbackDly*, with the result that intermediate telegrams can be avoided.

The switching-on and off procedures can be delayed via *cpOnDelay* and *cpOffDelay*.

The actuator can also be controlled via the control input *nviSetting*. The input has the same priority as *nviLampValue*, however switching-on and switching-off delays are ignored. The actuator responds as follows:

NV input *nviSetting.function*	Response	NV output *nvoLampValueFb*
SET_OFF	OFF	(0.0, 0)
SET_ON	Memory (last value before SET_OFF)	Memory

Stairwell light/automatic function

Configuration of the parameter *cpAutoOffTime* causes the actuator to operate in the stairwell light mode, i.e. it switches the output off automatically after the set time. If a repeated switch-on command is received while the light is switched on, the timer running time is reset. Switch-off commands during the switch-on duration are treated as a function of the parameter *cpBreakAutoOff*. The setting “TRUE” causes the actuator to be prematurely switched off as soon as a switch-off command is received, in the case of “FALSE“, the command is ignored.

The switch-on duration – if parameterised in *cpOffDelay* – is followed by a warning time, during which the user is informed that switch-off is imminent. In this case the relay is switched off every 5 seconds for 0.1 second.

Override

Normal operation via *nviLampValue* or nviSetting may be overridden by means of valid telegrams to *nviLampOverride* via group controller. The actuator accepts the received value and blocks the evaluation of telegrams received at both network variables. The block is cancelled on receipt of an “Invalid” telegram (.state = -1), whilst the actuator status is retained.

Inverting of the switching output

The switching method of the output relay can be inverted via the parameter *cpInvert*. With this it is possible to operate the relay logically as a break contact (standard setting) or break contact.

NOTE: With the aid of a coupling relay, for example, emergency lighting, which is switched on or retained in the event of a failure of the bus voltage, is possible with this function.

Assignment to group controllers

The object can be assigned to up to three *lamp group controllers* (# 21500) via the parameter *cpGroup*. The group controllers are located on the same device, as the lamp actuator object is operated directly and without network variables from the relevant group controllers. The assignment of the 3 parameter values corresponds to the sequence of the group controller on the device.

NOTE: Thanks to the option of operating all outputs via a group object, it is possible to create central commands in such a way that only one network variable has to be linked for each device. In this way, group linking can be used, which does not require any alias entries on the transmitting device. The extent of the central function, therefore, is unlimited.

Reset response

The output is set according to the value in *cpDefValue* and sent via *nvoLampValueFb*.

Troubleshooting

Faulty input telegrams (see LonMark specifications) are ignored.

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