

## EN – Software Description

Subject to technical alteration  
Version 11.04.07

# 29100...

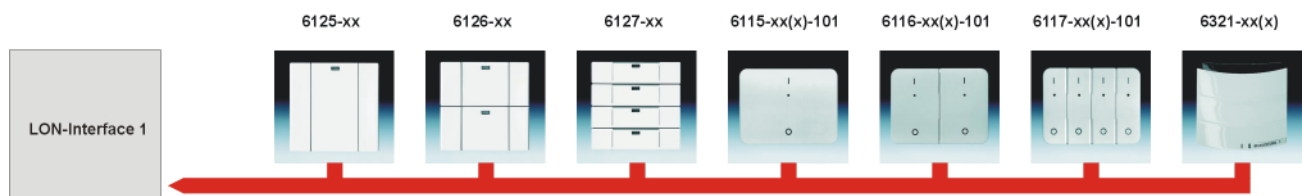


## 1 Application

By means of the LON-Interface 1 (bus coupling unit) the following devices of the Busch-Jaeger EIB programme:

- alpha nea / alpha exclusive: 6115-xx-101, 6116-xx-101, 6117-xx-101
- solo / future / future linear / carat: 6125-xx, 6126-xx, 6127-xx
- triton: 6321-xx

can be coupled to the building communication system LON of the company Echelon. The LON-Interface is designed as a gateway between the LON bus and the specific Busch-Jaeger device.



The defaults of the LonMark® function profiles **3200** „Switch“ and **3250** „Scene Panel“ are considered. For extended adjustment possibilities, there are user configuration properties (UCPT). The UCPTs used are defined in the Thermokon Device Resource Files from Version 2.1 or higher.

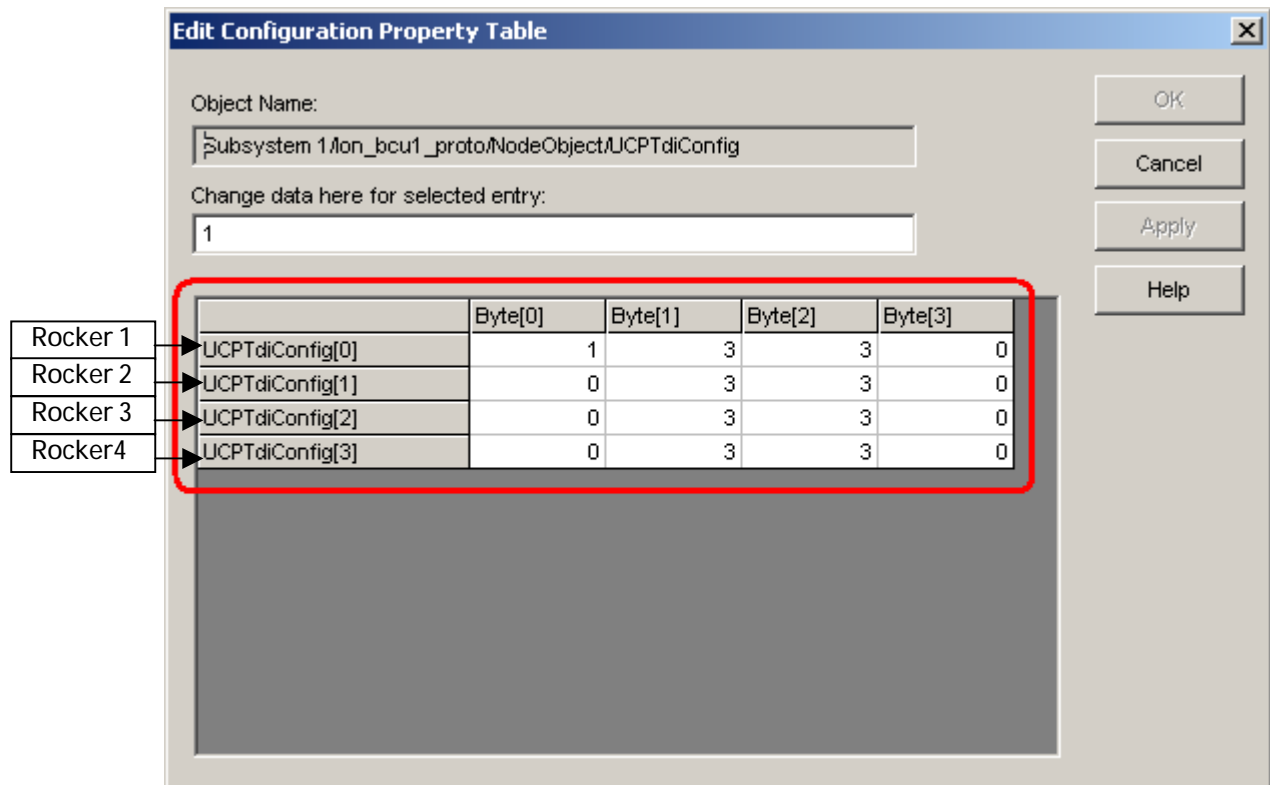
## 2 Types Available

6115-xx-101, 6116-xx-101, 6117-xx-101, 6125-xx, 6126-xx, 6127-xx, 6321-xx

- Universal operating element
- Button function (switch/dim/blind/shutter/send value/light scenes)
- Operating elements: tactile contacts left/right respectively at the top/at the bottom
- Display elements: LED for display of switching status

## 3 Parameterization of Rockers

For the parameterization of the respective rockers, the configuration property *UCPTdiConfig[...]* in the Node Object is used. When using the LONMaker, the configuration window shown below can be opened by a double click on the parameter.



Picture 3-1: LONmaker

### UCPTdiConfig[0], UCPTdiConfig[1]

The rockers are configured as follows:

- Rocker 1 by UCPTdiConfig[0]
- Rocker 2 by UCPTdiConfig[1]
- Rocker 3 by UCPTdiConfig[2]
- Rocker 4 by UCPTdiConfig[3]

### Byte[0]

The functions of the rocker are configured by the Byte[0]. The following functions are possible:

- 0 - switch
- 1 – dim
- 2 – blinds
- 3 - shutter
- 4 – scene

Example: Rocker 1 shall be configured as a dimming and rocker 2 as a blind function.

UCPTdiConfig[0].*Byte[0]* = 1

UCPTdiConfig[1].*Byte[0]* = 2

### Byte[1], Byte[2]

By Byte[1] the function of the left rocker button is configured. By Byte[2] the function of the right rocker button is configured. For example, it is possible to select the left button to switch-off (Byte[1] = 6) or switch-on (Byte[1] = 4) the light. (see chapter: 4.1.3)

Example:       The left button shall switch-on the light/ light it up and the right button shall switch-off the light/ darken the light/ dim the light.  
                  UCPTdiConfig[0].Byte[1] = 3  
                  UCPTdiConfig[0].Byte[2] = 7

### Byte[3]

The LED of the rocker is set in Byte[3]. Thus, it can be selected whether the colour shall be changed between red and green with a button actuation (see chapter 4.13).

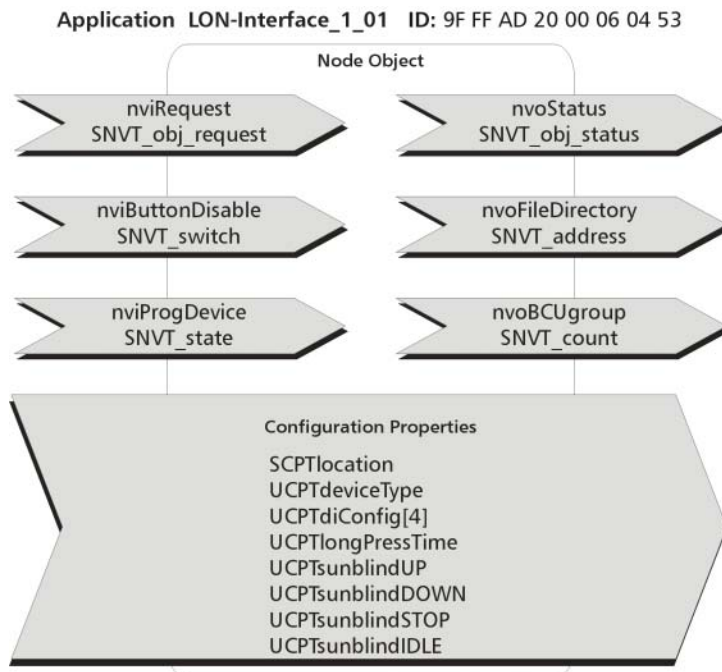
Example:       UCPTdiConfig[0].Byte[3] = 1

### Data Transmission

After having changed the parameters they must be transmitted to the operating element. By setting nviProgDevice.bit0 from 0 to 1 respectively after a voltage reset the new settings are taken over in the control element.

## 4 Software Description

### 4.1 Node Object



The Node Object supervises and controls the functions of the individual objects in the device. The basic functions required by the LonMark® are supported whereas general network variables and configuration properties for the control and parameterization of the device were added. After a parameterization, *nviProgDevice.bit0* must be set from 0 to 1 to take over the new parameter.

#### Button Functionality and Device Types:

During installation the used device type (6115, 6116, 6117, 6125, 6126, 6127, 6321) must be adjusted by the property *UCPTdeviceType*. Different functions, such as light control or scene polling can be allocated to the rockers. By *UCPTdiConfig[x]* the configuration of the rockers is made.

#### 4.1.1 Input Variable Node Object:

##### nviRequest

SNVT Type: SNVT\_obj\_request, Index 92

Function: Input variable including the functions RQ\_NORMAL, RQ\_UPDATE\_STATUS and RQ\_REPORT\_MASK.

##### nviButtonDisable

SNVT Typ: SNVT\_switch, Index 95

Funktion: Input variable for disabling the buttons. (Only for Switch, Dim, Blind, Shutter, Scene)

##### nviProgDevice

SNVT Type: SNVT\_state, Index 83

Function: Input variable for parameterization of the control element. After the parameterization *nviProgDevice.bit0* must be set from 0 to 1, so that the new parameters are taken over in the control element.

nviProgDevice.bit0 = 0 → nviProgDevice.bit0 = 1  
Acceptance of the new parameter.

#### 4.1.2 Output Variable Node Object:

##### nvoStatus

SNVT Type: SNVT\_obj\_status, Index 93

Function: Output variable including the requested status bits „invalid\_id“ and „invalid\_request“.

##### nvoFileDirectory

SNVT Type: SNVT\_address, Index 114

Function: The output variable makes the address data of the configuration property in the device available to the LON integration tool.

##### nvoBCUgroup

SNVT Type: SNVT\_count, Index 8

Function: Output variable with the LON-Interface type. The LON-Interface type cannot be changed.

## Software

### 4.1.3 Configuration Property Node Object:

#### SCPTlocation

SCPT Index: 17, SNVT\_str\_asc

Function: Additional input option to save information on the location in the device.

#### UCPTdeviceType

UCPT Index: 42, SNVT\_count

Function: The adaption of the software is made via this configuration property. Valid input values are:  
for: 6115-xx, 6116-xx, 6117-xx, 6125-xx, 6126-xx, 6127-xx, 6321-xx,

6115-xx	==>	6115
6116-xx	==>	6116
6117-xx	==>	6117
6125-xx	==>	6125
6126-xx	==>	6126
6127-xx	==>	6127
6321-xx	==>	6321

Preset value: 0

#### UCPTlongPressTime

UCPT Index: 71, typedef struct { SNVT\_time\_sec dimming; SNVT\_time\_sec sunblind;  
SNVT\_time\_sec scene; SNVT\_time\_sec universal; }

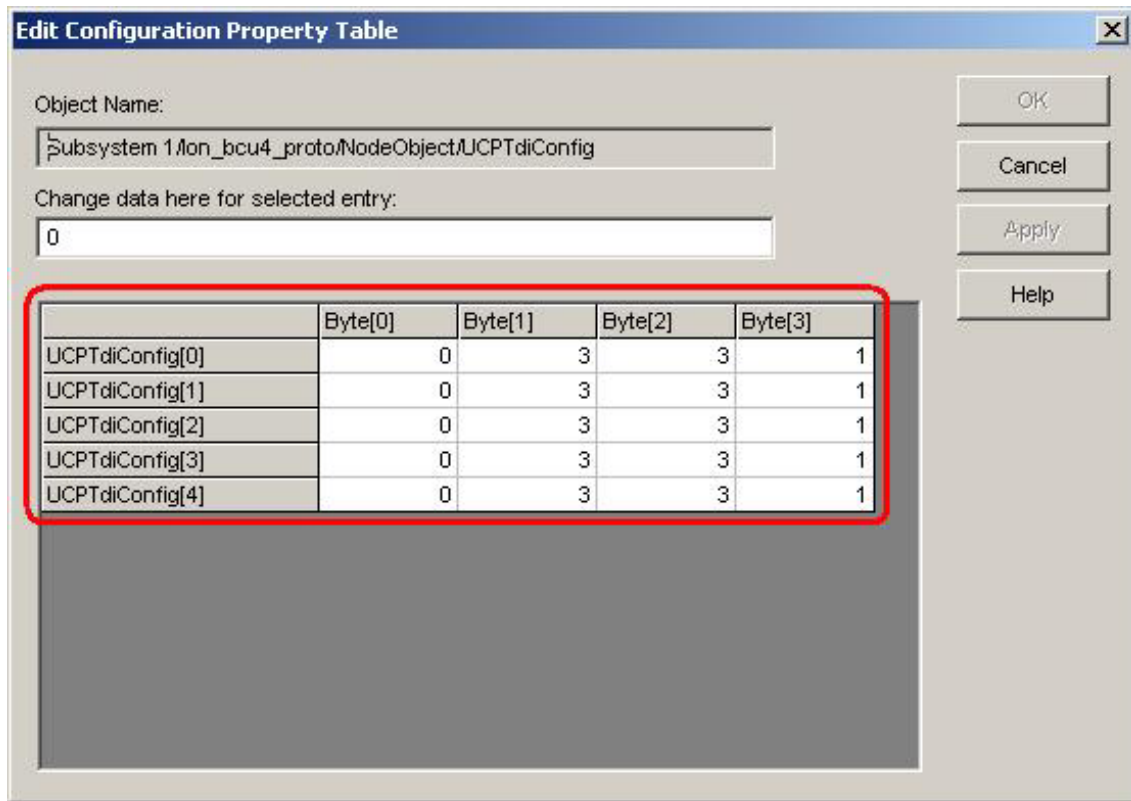
Function: By means of this configuration property the time (in seconds) for dimming, blinds, scene and universal can be input by a long button actuation.  
(preset value: 1.0;2.0;2.0;2.0)

#### UCPTdiConfig[0]...[4]

UCPT Index: 44, typedef struct {unsigned short Byte[4]} UNVT\_str\_hex4

Function: The configuration properties determine the rocker functions and their allocation to the output variables, as well as the control logic of the signal LEDs. UCPTdiConfig[0] is fix allocated to rocker 1, UCPTdiConfig[1] to rocker 2, UCPTdiConfig[2] to rocker 3 and UCPTdiConfig[3] to rocker.

UCPTdiConfig[0]	configured	rocker 1
UCPTdiConfig[1]	configured	rocker 2
UCPTdiConfig[2]	configured	rocker 3
UCPTdiConfig[3]	configured	rocker 4
UCPTdiConfig[x].Byte[0]	configured	function of rocker
UCPTdiConfig[x].Byte[1]	configured	function of left button
UCPTdiConfig[x].Byte[2]	configured	function of right button
UCPTdiConfig[x].Byte[3]	configured	function of LED



Picture 4-1: UCPTdiConfig

## UCPTsunblindUP

UCPT Index: 72, SNVT\_setting

Function: By means of this configuration parameter you can adjust which SNVT\_setting value shall be sent when the blind/shutter is going up. (Preset value: SET\_UP 100.0 0.0)

## UCPTsunblindDOWN

UCPT Index: 73, SNVT\_setting

Function: By means of this configuration parameter you can adjust which SNVT\_setting value shall be sent when the blind/shutter is going down. (Preset value: SET\_DOWN 100.0 0.0)

## UCPTsunblindSTOP

UCPT Index: 74, SNVT\_setting

Function: By means of this configuration parameter you can adjust which SNVT\_setting value shall be sent when the blind/shutter is stopped. (Preset value: SET\_STOP 0.0 0.0)

## UCPTsunblindIDLE

UCPT Index: 75, SNVT\_setting

Function: By means of this configuration parameter you can adjust which SNVT\_setting value shall be sent for the stand-by mode of the blind/shutter. *UCPTsunblindIDLE* is sent 500ms after the stop of the blind/shutter, if *UCPTsunblindIDLE* is unequal *UCPTsunblindSTOP*. (Preset value: SET\_NUL 0.0 0.0)

A rocker is fix allocated to a switch object. Each object has two output variables (xxx\_A; xxx\_B). Thus, a individual swtich object can be allocated to each button. This is useful for a toggle function. (see chapter 4.2)

Function: Switch light / buttons: Byte[0] = 0:

Configuration of the rocker for lighting ON / OFF by UCPTdiConfig							
Allocation of the rocker function block		Function left button		Function right button		Function LED	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description	Byte[3]	Description
0	Switching / Buttons	1	Light toggle ON / OFF (Switch_A_n)	1	Light toggle ON / OFF (Switch_B_n)	1	Left rocker red Right rocker green
		2	Light only ON (Switch_A_n)	2	Light only ON (Switch_A_n)	2	Left rocker green Right rocker red
		3	Light only ON (Switch_B_n)	3	Light only ON (Switch_B_n)	3	Left rocker red Right rocker red
		4	Light only OFF(Switch_A_n)	4	Light only OFF (Switch_A_n)	4	Left rocker green Right rocker green
		5	Light only OFF (Switch_B_n)	5	Light only OFF (Switch_B_n)		
		6	Command Automatic (= 0.0 -1) (Switch_A_n)	6	Command Automatic (= 0.0 -1) (Switch_A_n)		
		7	Command Automatic (= 0.0 -1) (Switch_B_n)	7	Command Automatic (= 0.0 -1) (Switch_B_n)		

Examples:

A: Rocker 1 shall be configured as a switching function. The left button turns the light on and the right button turns the light off.  
 UCPTdiConfig[0].Byte[0] = 0 → switch  
 UCPTdiConfig[0].Byte[1] = 2 → left ON  
 UCPTdiConfig[0].Byte[2] = 4 → right OFF  
 UCPTdiConfig[0].Byte[3] = 1 → LED

B: Rocker 2 shall be configured as a switching function. The left and right button toggles the light on and off. The LED shall be green.  
 UCPTdiConfig[1].Byte[0] = 0 → switch  
 UCPTdiConfig[1].Byte[1] = 1 → toggle left  
 UCPTdiConfig[1].Byte[2] = 1 → toggle right  
 UCPTdiConfig[1].Byte[3] = 4 → LED

C: Rocker 3 shall be configured as a switching function. The left button toggles the light on and off. The right button puts the lighting in the automatic mode. The LED shall be red.  
 UCPTdiConfig[2].Byte[0] = 0 → switch  
 UCPTdiConfig[2].Byte[1] = 1 → toggle left  
 UCPTdiConfig[2].Byte[2] = 6 → toggle automatic  
 UCPTdiConfig[2].Byte[3] = 3 → LED

Function: Dim Lighting: Byte[0] = 1:

Configuration of the rocker for lighting with dimming by UCPTdiConfig							
Allocation of rocker function block		Function left button		Function right button		Function LED	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description	Byte[3]	Description
1	Lighting with Dimming	1	Toggle light by dimming (Switch_A_n)	1	Toggle light by dimming (Switch_B_n)	1	Left rocker red Right rocker green
			Turn-on value = Max-value		Turn-on = Max-value		
		2	Toggle light by dimming (Switch_A_n)	2	Toggle light by dimming (Switch_B_n)	2	Left rocker green Right rocker red
			Turn-on value = last ON value		Turn-on value = last ON value		
		3	Light brighter only by dimming (Switch_A_n)	3	Light brighter only by dimming (Switch_A_n)	3	Left rocker red Right rocker red
			Turn-on value = Max-value		Turn-on value = Max-value		
		4	Light only brighter by dimming (Switch_B_n)	4	Light only brighter by dimming (Switch_B_n)	4	Left rocker green Right rocker green
			Turn-on value = Max-value		Turn-on value = Max-value		
		5	Light only brighter by dimming (Switch_A_n)	5	Light only brighter by dimming (Switch_A_n)		
			Turn-on value = last ON value		Turn-on value = last ON value		
		6	Light only brighter by dimming (Switch_B_n)	6	Light only brighter by dimming (Switch_B_n)		
			Turn-on value = last ON value		Turn-on value = last ON value		
		7	Light only darker by dimming (Switch_A_n)	7	Light only darker by dimming (Switch_A_n)		
		8	Light only darker by dimming (Switch_B_n)	8	Light only darker by dimming (Switch_B_n)		
		9	Command Automatic (= 0.0 – 1) (Switch_A_n)	9	Command Automatic (= 0.0 – 1) (Switch_A_n)		
		A	Command Automatic (= 0.0 – 1) (Switch_B_n)	A	Command Automatic (= 0.0 – 1) (Switch_B_n)		

Examples:

A: Rocker 1 shall be configured as a dimming function. Thus, the left button turns the led green and the right button turns the led red.

UCPTdiConfig[0].Byte[0] = 1 → Dim  
 UCPTdiConfig[0].Byte[1] = 3 → Left: On/Brighter  
 UCPTdiConfig[0].Byte[2] = 7 → Right: Off/Darker  
 UCPTdiConfig[0].Byte[3] = 1 → LED

B: Rocker 2 shall be configured as a switching function. Thus, the left button toggles the light on by *nvoSwitch\_A\_n* and the right button toggles the light off by *nvoSwitch\_B\_n*. The LED shall be green.

UCPTdiConfig[1].Byte[0] = 1 → Dim  
 UCPTdiConfig[1].Byte[1] = 1 → Left: Toggle by dimming  
 UCPTdiConfig[1].Byte[2] = 1 → Right: Toggle by dimming  
 UCPTdiConfig[1].Byte[3] = 4 → LED



Function Blinds: Byte[0] = 2:

Configuration of button for blind with UCPTdiConfig							
Allocation of rocker function block		Function left button		Function right button		Function LED	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description	Byte[3]	Description
2	Blind	1	Blind Up (Setting_A_n)	1	Blind Up (Setting_A_n)	1	Left rocker red Right rocker green
		2	Blind Down(Setting_A_n)	2	Blind Down (Setting_A_n)	2	Left rocker green Right rocker red
		3	Blind Up (Setting_B_n)	3	Blind Up (Setting_B_n)	3	Left rocker red Right rocker red
		4	Blind Down(Setting_B_n)	4	Blind Down (Setting_B_n)	4	Left rocker green Right rocker green

Example: Rocker 1 shall be configured as a blind function. Thus, the left button switches the blind up and the right button switches the blind down. The LED shall be red.

UCPTdiConfig[0].Byte[0] = 2 → Blind  
 UCPTdiConfig[0].Byte[1] = 1 → Left: Up  
 UCPTdiConfig[0].Byte[2] = 2 → Right: Down  
 UCPTdiConfig[0].Byte[3] = 3 → LED

Function: Shutter: Byte[0] = 3:

Configuration of rocker for shutter with UCPTdiConfig							
Allocation rocker function block		Function left button		Function right button		Function LED	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description	Byte[3]	Description
3	Shutter	1	Shutter Up (Setting_A_n)	1	Shutter Down (Setting_A_n)	1	Left rocker red Right rocker green
		2	Shutter Up (Setting_A_n)	2	Shuter Down (Setting_A_n)	2	Left rocker green Right rocker red
		3	Shutter Up (Setting_B_n)	3	Shutter Down (Setting_B_n)	3	Left rocker red Right rocker red
		4	Shutter Up (Setting_B_n)	4	Shuter Down (Setting_B_n)	4	Left rocker green Right rocker green

Example: Rocker 3 shall be configured as a shutter function. Thus, the left button switches the shutter up and the right button switchs the shutter down. The LED shall be red.

UCPTdiConfig[2].Byte[0] = 3 → Shutter  
 UCPTdiConfig[2].Byte[1] = 1 → Left Up  
 UCPTdiConfig[2].Byte[2] = 2 → Right Down  
 UCPTdiConfig[2].Byte[3] = 3 → LED

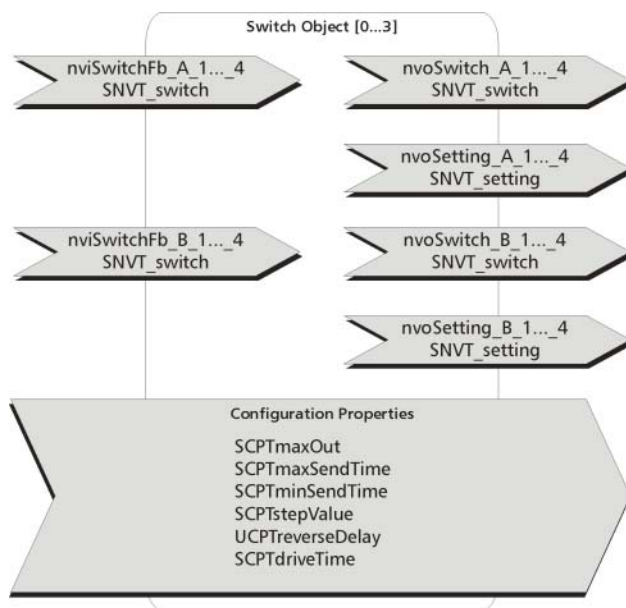
Function Scene Control: Byte[0] = 4:

Configuration of button for scene control with UCPTdiConfig							
Allocation of rocker function block		Function left button		Function right button		Function	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description	Byte[3]	Description
4	Scene	01-FE	Scene number	01-FE	Scene number	1	Left rocker red Right rocker green
						2	Left rocker green Right rocker red
						3	Left rocker red Right rocker red
						4	Left rocker green Right rocker green

Example: Rocker 2 shall be configured as a scene call. Thus, the left button calls scene 5 and the right button calls scene 12. The LED shall be red.

UCPTdiConfig[1].Byte[0] = 4 → Scene  
 UCPTdiConfig[1].Byte[1] = 5 dez → Call scene 5  
 UCPTdiConfig[1].Byte[2] = 12 dez → Call scene 12  
 UCPTdiConfig[1].Byte[3] = 3 → LED

## 4.2 Switch Object



Four identical switch objects for light and blind control. The switch objects are directly allocated to the rockers:

Switch Object[0] = Rocker 1  
 Switch Object[1] = Rocker 2  
 Switch Object[2] = Rocker 3  
 Switch Object[3] = Rocker 4

The allocation of the rockers to the individual functions is made as described above by means of the configuration properties UCPTdiConfig[0]...[4] in the Node Object. Possible functions are light control ON/OFF, light control ON/OFF by dimming or blinds UP/DOWN by automatic run. Each switch object has 2 variable groups which are distinguished by the appendix *xxx\_A* and *xxx\_B*. Thus, a network variable can be allocated to each button.

### 4.2.1 Input Variable Switch Object:

**nviSwitchFb\_A\_1...4, nviSwitchFb\_B\_1...4**

SNVT Type: SNVT\_switch, Index 95

Function: Input variable for the current status of the light groups controlled by nvoSwitch\_A\_1...4 respectively nvoSwitchFb\_B\_1...4.

## 4.2.2 Output Variable Switch Object:

**nvoSwitch\_A\_1...5, nvoSetting\_A\_1...4, nvoSwitch\_B\_1...4, nvoSetting\_B\_1...4**

SNVT Type: SNVT\_switch, Index 95; SNVT\_setting, Index 117

Function: Output variable for control of light groups and of light and blind controllers. The output values are depending on the function setting. A switch object can be allocated to each rocker. Each switch object has two different variable groups, which are distinguished by the appendix *\_A* respectively *\_B*. Thus, a network variable can be allocated to each button.

BY

UCPTdiConfig[x].Byte[0] a function is allocated to the rockers.

UCPTdiConfig[x].Byte[1] a function is allocated to the left button.

UCPTdiConfig[x].Byte[2] a function is allocated to the right button.

UCPTdiConfig[x].Byte[3] a function is allocated to the LED.

### Switching/ Buttons

UCPTdiConfig[x].Byte[0] = 00<sub>hex</sub>

#### Button pressed/ not pressed

UCPTdiConfig[x].Byte[1...2] = 01<sub>hex</sub>

Button pressed	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;

Button not pressed	nvoSwitch_A_x.value	= 0
	nvoSwitch_A_x.state	= 0
	nvoSetting_A_x.function	= SET_OFF;
	nvoSetting_A_x.setting	= 0;

UCPTdiConfig[x].Byte[1...2] = 02<sub>hex</sub>

Button pressed	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;

Button not pressed	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0
	nvoSetting_B_x.function	= SET_OFF;
	nvoSetting_B_x.setting	= 0;

#### Toggle of Lighting

UCPTdiConfig[x].Byte[1] = 03<sub>hex</sub>

Each button actuation results in a toggling of the lighting, i.e. between ON and OFF

Lighting ON	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;

Lighting OFF	nvoSwitch_A_x.value	= 0
	nvoSwitch_A_x.state	= 0
	nvoSetting_A_x.function	= SET_OFF;
	nvoSetting_A_x.setting	= 0;

## Software

### UCPTdiConfig[x].Byte[2] = 03<sub>hex</sub>

Each button actuation results in a toggling of the lighting, i.e. between ON and OFF.

Lighting ON	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;
Lighting OFF	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0
	nvoSetting_B_x.function	= SET_OFF;
	nvoSetting_B_x.setting	= 0;

### Lighting ON

### UCPTdiConfig[x].Byte[1...2] = 04<sub>hex</sub>

Each button actuation leads to a switching-on of the lighting.

Lighting ON	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;

### UCPTdiConfig[x].Byte[1...2] = 05<sub>hex</sub>

Each button actuation results in a switching-on of the lighting.

Lighting ON	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;

### Lighting OFF

### UCPTdiConfig[x].Byte[1...2] = 06<sub>hex</sub>

Each button actuation results in a switching-off of the lighting.

Lighting OFF	nvoSwitch_A_x.value	= 0
	nvoSwitch_A_x.state	= 0
	nvoSetting_A_x.function	= SET_OFF;
	nvoSetting_A_x.setting	= 0;

### UCPTdiConfig[x].Byte[1...2] = 07<sub>hex</sub>

Each button actuation results in a switching-off of the lighting.

Lighting OFF	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0
	nvoSetting_B_x.function	= SET_OFF;
	nvoSetting_B_x.setting	= 0;

## Automatic

### UCPTdiConfig[x].Byte[1..2] = 08<sub>hex</sub>

An actuation of an "automatic button" switches the variables nvoSwitch\_A\_x to the value 0,0 -1. Thus, for example a light controller can be reset from an external override to the automatic mode.

### UCPTdiConfig[x].Byte[1...2] = 08<sub>hex</sub>

The actuation of an "automatic button" switches the variables nvoSwitch\_B\_x to the value 0,0 -1. Thus, for example a light controller can be reset from an external override to the automatic mode.

## Dim

### UCPTdiConfig[x].Byte[0] = 01<sub>hex</sub>

### Toggle lighting by dimming, turn-on value = max. value

Short button actuations (< 1 s) lead to a toggling of the current light status, whereas the .value – turn-on value is always SCPTmaxOut. By longer button actuations (> 1 s) the dimming function is activated, i.e. based on the current lighting status, the .value-value of the switch variables is increased or reduced in percent steps of UCPTstepValue as long as the button is pressed.

## UCPTdiConfig[x].Byte[1] = 01<sub>hex</sub>

Lighting on maximum value	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
Lighting on 50%	nvoSwitch_A_x.value	= 50,0
	nvoSwitch_A_x.state	= 1
Lighting OFF	nvoSwitch.value	= 0
	nvoSwitch_A_x.state	= 0

The variables nvoSetting\_A\_x are not changed in this setting.

## UCPTdiConfig[x].Byte[2] = 01<sub>hex</sub>,

Lighting on maximum value	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
Lighting on 50%	nvoSwitch_B_x.value	= 50,0
	nvoSwitch_B_x.state	= 1
Lighting OFF	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0

The variables nvoSetting\_B\_x are not changed in this setting.

### Toggle lighting by dimming, turn-on value = last on-value

#### UCPTdiConfig[x].Byte[1] = 02<sub>hex</sub>

Function as with 01<sub>hex</sub>, but with the difference, that not the value SCPTmaxOut but the last turn-on value is taken over. The smallest turn-on value is limited to 20%.

### Lighting ON by dimming brighter, turn-on value = max. value

If the lighting is switched-off, a button actuation results in an immediate switching-on of the lighting. By longer button actuations (> 1 s) the function "dim brighter" is activated, i.e. based on the current light status the .value – value of the switch variable is reduced in present steps of UCPTstepValue as long as the maximum value SCPTmaxOut is reached. The sending interval in the mode dimming is adjusted by SCPTminSendTime and is preadjusted to approx. 300ms.

#### UCPTdiConfig[x].Byte[1...2] = 03<sub>hex</sub>

Switch-on of lighting	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;
Dim lighting brighter	nvoSwitch_A_x.value	= last value + UCPTstepValue
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_UP;
	nvoSetting_A_x.setting	= UCPTstepValue;

#### UCPTdiConfig[x].Byte[1...2] = 04<sub>hex</sub>

Switch-on of lighting	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;
Dim lighting brighter	nvoSwitch_B_x.value	= last value + UCPTstepValue
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_UP;
	nvoSetting_B_x.setting	= UCPTstepValue;

### Lighting ON by dimming brighter, turn-on value = last on-value

#### UCPTdiConfig[x].Byte[1...2] = 05<sub>hex</sub>, 06<sub>hex</sub>,

Function as with 03<sub>hex</sub>, but with the difference, that not the value SCPTmaxOut is taken over when switching-on the light, but the last turn-on value. The smallest turn-on value is limited to 20%.

## Lighting OFF by dimming darker

If the lighting is turned-on, a short button actuation (< 1 s) leads to an immediate switching-off of the lighting. By longer button actuations (> 1 s) the function "dim darker" is activated, i.e. based on the current lighting status the .value -value of the switch variables is reduced in percent steps of UCPTstepValue as long as the value 0 is reached. The sending interval in the mode dimming is adjusted by SCPTminSendTime and amounts to approx. 300ms preset.

### UCPTdiConfig[x].Byte[1...2] = 07<sub>hex</sub>

Switch-off lighting

nvoSwitch\_A\_x.value = 0  
nvoSwitch\_A\_x.state = 0  
nvoSetting\_A\_x.function = SET\_OFF;  
nvoSetting\_A\_x.setting = 0;

Dim lighting darker

nvoSwitch\_A\_x.value = last value - UCPTstepValue  
nvoSwitch\_A\_x.state = 1  
nvoSetting\_A\_x.function = SET\_DOWN;  
nvoSetting\_A\_x.setting = UCPTstepValue;

### UCPTdiConfig[x].Byte[1...2] = 08<sub>hex</sub>

Switch-off lighting

nvoSwitch\_B\_x.value = 0  
nvoSwitch\_B\_x.state = 0  
nvoSetting\_B\_x.function = SET\_OFF;  
nvoSetting\_B\_x.setting = 0;

nvoSwitch\_B\_x.value = last value - UCPTstepValue  
nvoSwitch\_B\_x.state = 1  
nvoSetting\_B\_x.function = SET\_DOWN;  
nvoSetting\_B\_x.setting = UCPTstepValue;

## Automatic

### UCPTdiConfig[x].Byte[1...2] = 09<sub>hex</sub>

The actuation of an "automatic button" switches the variable nvoSwitch\_A\_x to the value 0,0 -1. Thus, a light controller can be reset from an external override to the automatic mode.

### UCPTdiConfig[x].Byte[1...2] = 0A<sub>hex</sub>

The actuation of an "automatic button" switches the variable nvoSwitch\_B\_x to the value 0,0 -1. Thus, a light controller can be reset from external override to the automatic mode.

## Blind

### UCPTdiConfig[x].Byte[0] = 02<sub>hex</sub>

#### Blind UP

In the configuration mode "blind UP" only the nvoSetting variables are changed and sent. Short button actuations (< 2 s) are used for a fine adjustment of the lamellas. A long button actuation (>2 s) starts the automatic run and drives the blind continuously in the direction open for the time SCPTdriveTime. The automatic run can be stopped by a renewed button actuation.

### UCPTdiConfig[x].Byte[1...2] = 01<sub>hex</sub>

Open blind nvoSetting\_A\_x.function = UCPTsunblindUP;  
Stop blind nvoSetting\_A\_x.function = UCPTsunblindSTOP;

### UCPTdiConfig[x].Byte[1...2] = 03<sub>hex</sub>

Open blind nvoSetting\_B\_x.function = UCPTsunblindUP;  
Stop blind nvoSetting\_B\_x.function = UCPTsunblindSTOP;

With a delay of 500ms the command *UCPTsunblindIDLE* for idle mode is sent after the command *UCPTsunblindSTOP* if *UCPTsunblindIDLE* is unequal to *UCPTsunblindSTOP*.

## Blind DOWN

In the configuration mode "blind DOWN" only the nvoSetting variables are changed and sent. Short button actuations (< 2 s) are for the fine adjustment of the lamellas. A long button actuation (> 2 s) starts the automatic run and drives the blind for the time SCPTdriveTime continuously into the direction close. The automatic run can be stopped by a renewed button actuation.

### UCPTdiConfig[x].Byte[1...2] = 02<sub>hex</sub>

Close blind	nvoSetting_A_x.function	= UCPTsunblindDOWN;
Stop blind	nvoSetting_A_x.function	= UCPTsunblindSTOP;

### UCPTdiConfig[x].Byte[1...2] = 04<sub>hex</sub>

Close blind	nvoSetting_B_x.function	= UCPTsunblindDOWN;
Stop blind	nvoSetting_B_x.function	= UCPTsunblindSTOP;

With a delay of 500ms the command *UCPTsunblindIDLE* for idle mode is sent after the command *UCPTsunblindSTOP* if *UCPTsunblindIDLE* is unequal to *UCPTsunblindSTOP*.

## Shutter

### UCPTdiConfig[x].Byte[0] = 03<sub>hex</sub>

## Shutter UP

In the configuration mode "Shutter UP" only the nvoSetting variables are changed and sent. Short button actuation (< 2 s) starts the automatic run and drives the shutter continuously in the direction open for the time SCPTdriveTime. The automatic run can be stopped by a renewed button actuation. By a long button actuation (> 2 s) the position of the shutter can be individually adjusted.

### UCPTdiConfig[x].Byte[1...2] = 01<sub>hex</sub>

Open shutter	nvoSetting_A_x.function	= UCPTsunblindUP;
Stop shutter	nvoSetting_A_x.function	= UCPTsunblindSTOP;

### UCPTdiConfig[x].Byte[1...2] = 03<sub>hex</sub>

Open shutter	nvoSetting_B_x.function	= UCPTsunblindUP;
Stop shutter	nvoSetting_B_x.function	= UCPTsunblindSTOP;

With a delay of 500ms the command *UCPTsunblindIDLE* for idle mode is sent after the command *UCPTsunblindSTOP* if *UCPTsunblindIDLE* is unequal to *UCPTsunblindSTOP*.

## Shutter DOWN

In the configuration mode "shutter DOWN" only the nvoSetting variables are changed and sent. Short button actuation (< 2 s) starts the automatic run and drives the shutter continuously into the direction close for the time SCPTdriveTime. The automatic run can be stopped by a renewed button actuation. By a long button actuation (> 2 s) the position of the shutter can be adjusted individually.

### UCPTdiConfig[x].Byte[1...2] = 02<sub>hex</sub>

Close shutter	nvoSetting_A_x.function	= UCPTsunblindDOWN;
Stop shutter	nvoSetting_A_x.function	= UCPTsunblindSTOP;

### UCPTdiConfig[x].Byte[1...2] = 04<sub>hex</sub>

Close shutter	nvoSetting_B_x.function	= UCPTsunblindDOWN;
Stop shutter	nvoSetting_B_x.function	= UCPTsunblindSTOP;

With a delay of 500ms the command *UCPTsunblindIDLE* for idle mode is sent after the command *UCPTsunblindSTOP* if *UCPTsunblindIDLE* is unequal to *UCPTsunblindSTOP*.



## Software

### 4.2.3 Configuration Property Switch Object:

#### SCPTmaxOut

SCPT Index: 93, SNVT\_lev\_cont

Function: This configuration property determines the maximum output value of the variable nvoSwitch.value. (Preset value: 100.0)

#### SCPTmaxSendTime

SCPT Index: 49, SNVT\_time\_sec

Function: Heartbeat function. This configuration property stipulates the interval time after which the output variable is sent. By input values = 0, the heartbeat function is deactivated. (Preset value: 0,0 s)

#### SCPTminSendTime

SCPT Index: 52, SNVT\_time\_sec

Function: This configuration property stipulates the sending interval of the output variable in the mode dimming. By input values = 0, the function is deactivated. (Preset value: 0,3 s)

#### SCPTstepValue

SCPT Index: 92, SNVT\_lev\_cont

Function: This configuration property defines the step size of the variable nvoSwitch.value in the mode dimming. (Preset value: 5.0)

#### UCPTreverseDelay

UCPT Index: 14, SNVT\_count

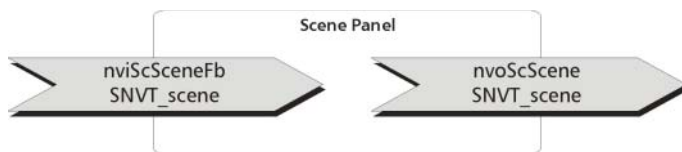
Function: This configuration property defines the toggling delay with a rotation reversing of the blind motors. Thus, a change command from e.g. nvoSetting = SET\_UP to nvoSetting = SET\_DOWN is output delayed. (Preset value: 500 ms)

#### SCPTdriveTime

UCPT Index: 45, SNVT\_time\_sec

Function: This configuration property defines the maximum turn-on time of the blind motors in the automatic run. (Preset value: 100,0 s)

## 4.3 Scene Panel



### 4.3.1 Input Variable Scene Panel

#### nviScSceneFb

SNVT Type: SNVT\_scene, Index 115

Function: Input variables with the current lighting scene in the room.

### 4.3.2 Output Variable Scene Panel:

#### nvoScScene

SNVT Type: SNVT\_scene, Index 115

Function: Output variable for driving a scene controller. The output variables are depending on the function adjustments. By UCPTdiConfig[x].Byte[0] = 3 buttons are allocated to the Scene-Object.

UCPTdiConfig[x].Byte[1...2] = 01<sub>hex</sub> - FE<sub>hex</sub> allocates a scene number to these buttons, which is called by short button actuations with SC\_RECALL. With longer button actuations (> 2s) the scene is relearned with SC\_LEARN. It is also possible to allocate several buttons to the scene panel.