

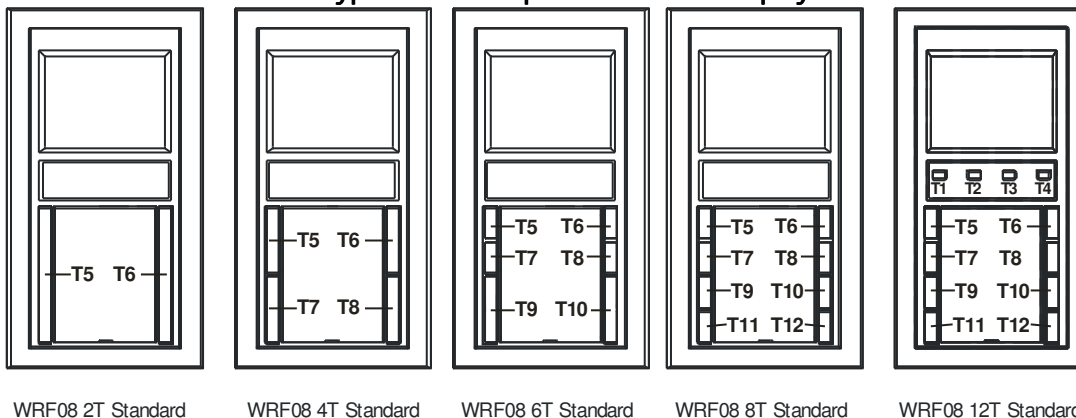
## Software Application wrf08\_06 (Lighting, Blinds, Air-Conditioning Control) For model WRF08

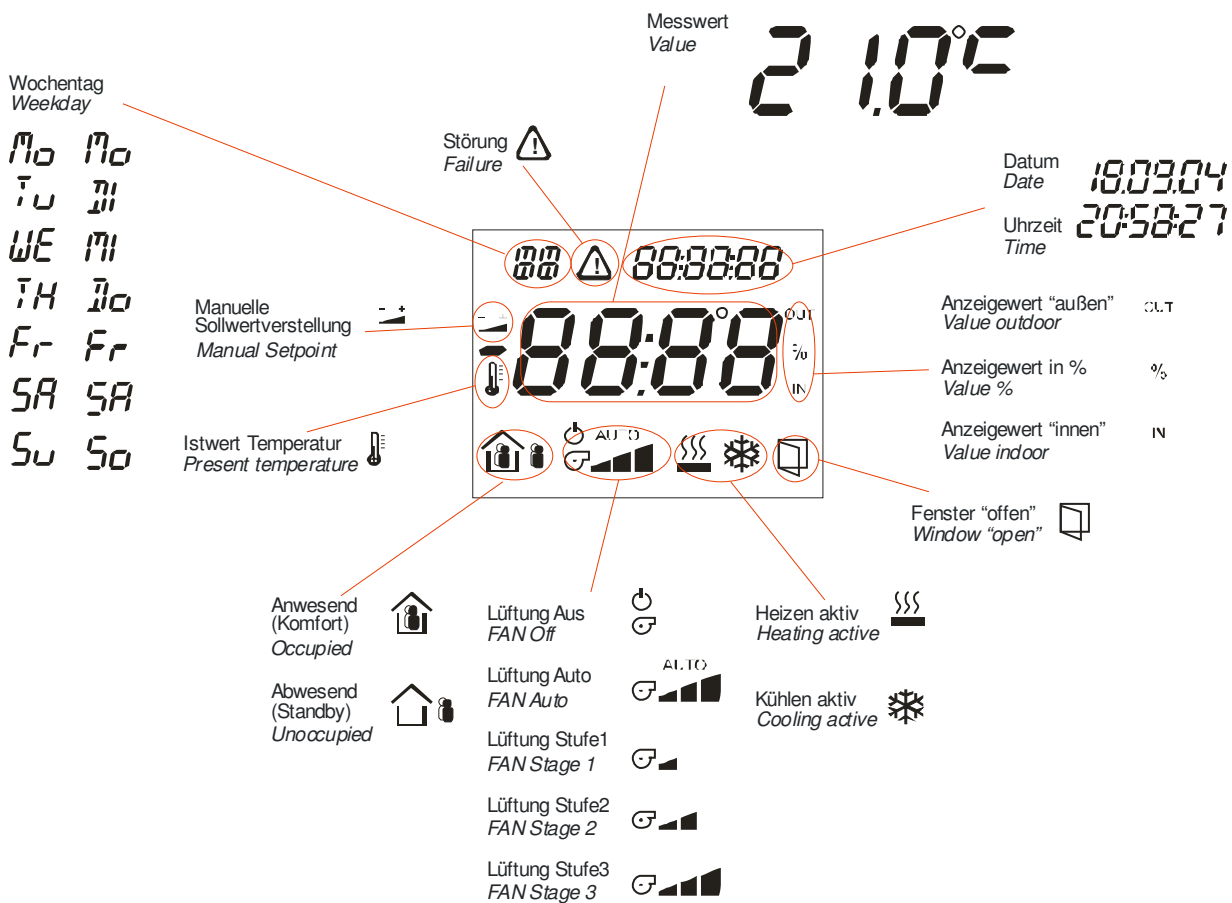
### 1 Overview

The multi-function room operating panel WRF08 is designed for temperature detection and integrated control of HVAC, lighting and blinds for single room control. The operating functions can be flexibly adapted to different room requirements. Thus, various numbers of function buttons are available.

The prescription of the LonMark® function profile **8090** „Space Comfort Control Command Module“, **3200** „Switch“, **1060** „Occupancy Sensor“, and **3250** „Scene Panel“ are considered. For extended adjustment options, user-defined configuration parameters (UCPT) are used. The UCPTs used herewith are defined in the **Thermokon Device Resource Files** from version 1.6 or higher.

#### Presentation of the Device Types and Description of the LCD-Display:

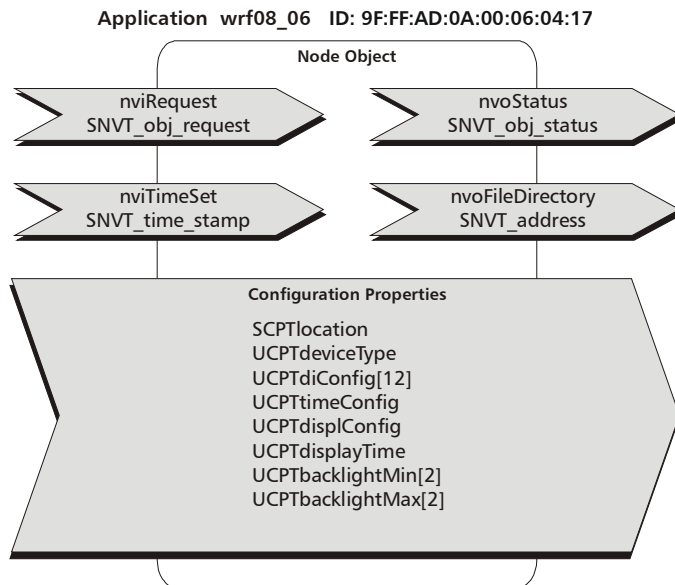




### Control of Segments

Time, date, weekday:	nviTimeSet, UCPTtimeConfig
Measuring value:	nvoSpaceTemp, nvoSetptEffect, nvoSetptOffset, nviTempOutdoor, nviRH_Indoor, nviRH_Outdoor, UCPTdisplConfig
Room occupancy:	nvoEffectOccup, UCPTdisplConfig
Fan symbols:	nviFanSpeed, nvoFanSpeed, UCPTdisplConfig
Failure:	nviAlarm, nviCmd_all
Heating, cooling:	nviUnitStatus, nviCmd_all
Window:	nviEnergyHoldOff, nviCmd_all

## 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 parameters for the control and parameterization of the device were added.

**Time / Date:** After a module reset, the display for time and date is faded out. Only after receipt of a message via the input variable *nviTimeSet* the time is faded in. The time should be synchronized by a LON system clock once a day. The display of the time can be adapted to individual requests by *UCPTtimeConfig*. Please see the parameter description on page 4.

**Display of Measuring Value:** Besides the room temperature it is also possible to display for example the set point, outdoor temperature or the relative humidity. This configuration is made by means of *UCPTdisplConfig*, whereas the display is changing between the different display values in the time interval *UCPTdisplayTime*.

**Background Illumination:** The background illumination of the LCD and the text field can be selected separately. In the stable state, i.e. after a module reset or if no button actuation is made for a time exceeding 15 seconds, the value *UCPTbacklightMin[0]* determines the brightness of the LCD and the value *UCPTbacklightMin[1]* determines the brightness of the label area. By actuating one button, the stable state is left and the values *UCPTbacklightMax* determine the brightness of the LCD and the label area.

**Button Functionality and Device Types:** During installation, the device type used (2T, 4T, 8T, 12T) must be adjusted by the parameter *UCPTdeviceType*. Different functions and objects such as set point adjustment or light control can be allocated to the buttons. The configuration of the buttons T1 to T12 is made by *UCPTdiConfig[0]* to *UCPTdiConfig[11]*.

### Input Variables 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.

#### nviTimeSet

SNVT Type: SNVT\_time\_stamp, Index 84

Function: Input variable for synchronisation of the time displayed by means of a LON system clock. After a module reset the time is faded out as long as a first valid value is received.

## 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 parameter in the device available to the LON-integration tool.

## Configuration Parameter Node Object:

### SCPTlocation

SCPT Index: 17, SNVT\_str\_asc

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

### UCPTdeviceType

UCPT Index: 42, SNVT\_count

Function: The adaption to the software is made by this configuration parameter. Valid input values are:

Device type 2T ==>	UCPTdeviceType = 2
Device type 4T ==>	UCPTdeviceType = 4
Device type 8T ==>	UCPTdeviceType = 8
Device type 12T ==>	UCPTdeviceType = 12

Preset value: 8

### UCPTdiConfig[0]...[11]

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

Function: These configuration parameters determine the button functions and their allocation to the output variables and the control logic of the button LEDs.

UCPTdiConfig[0]	configures	<b>button T1</b>
UCPTdiConfig[1]	configures	<b>button T2</b>
:		
UCPTdiConfig[11]	configures	<b>button T12</b>

Example: Button **T5** configures for *Switch-Object-0* as *light button with toggle function*.  
UCPTdiConfig[4].*Byte[0]* = 0 and UCPTdiConfig[4].*Byte[1]* = 2

UCPTdiConfig, Configuration of Buttons for Lighting and Sunblind					
Allocation Button - Function Block		Button - Function		LED - Function	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description
0	Switch-Object 0	1	actuated / not actuated	1	LED = ON, if button is pressed LED = OFF, if button is not pressed
1	Switch-Object 1	2	Light Toggle	2	LED = ON, if light = ON LED = OFF, if light = OFF (analyze of nvoSwitch and nviSwitchFb)
2	Switch-Object 2	3	Light only ON	3	LED = ON, if nvoSetting.function = SET_UP LED = OFF, if nvoSetting.function = SET_STOP
3	Switch-Object 3	4	Light only OFF	4	LED = ON, if nvoSetting.function = SET_DOWN LED = OFF, if nvoSetting.function = SET_STOP
4	Switch-Object 4	5	Light Toggle with dimming Switch-on value = Max-value	5	LED = ON, if nvoSwitch.state = -1 LED = OFF, if nvoSwitch.state ≠ -1
5	Switch-Object 5	6	Light Toggle with dimming Switch-on value = last ON-value		
6	Switch-Object 6	7	Light only brighter by dimming Switch-on value = Max-value		
7	Switch-Object 7	8	Light only brighter by dimming Switch-on value = last ON-value		
		9	Light only darker by dimming		
		0A	Blind UP		
		0B	Blind DOWN		
		0C	Command automatic (= 0.0 –1)		

UCPTdiConfig, Configuration of Buttons for Scene Control					
Allocation Button - Function Block		Button - Function		LED - Function	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description
8	Scene Panel	01-FE	Scenen-Number	1	LED = ON, if button is pressed LED = OFF, if button is not pressed
				2	LED = ON, if button is pressed or if .scene_number = Byte[1] LED = OFF, if .scene_number ≠ Byte[1] (analyze of nvoScScene and nviScSceneFb)

UCPTdiConfig, Configuration of Buttons for Control of Room Occupancy					
Allocation Button - Function Block		Button - Function		LED - Function	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description
9	Occupancy Sensor	1	OCCUPIED, overtime function	1	LED = ON, if button is pressed LED = OFF, if button is not pressed
		2	Toggle Occupied/Unoccupied	2	LED = ON, if nvoOccup = OC_OCCUPIED LED = OFF, if nvoOccup ≠ OC_OCCUPIED
		3	only OCCUPIED	3	LED = ON, if nvoEffectOccup = OC_OCCUPIED LED = OFF, if nvoEffectOccup ≠ OC_OCCUPIED
		4	only UNOCCUPIED		
UCPTdiConfig, Configuration of Buttons for Air-Conditioning					
Allocation Button - Function Block		Button - Function		LED - Function	
Byte[0]	Description	Byte[1]	Description	Byte[2]	Description
0A	Space Comfort Control Command Module	1	Set temperature Plus	1	LED = ON, if button is pressed LED = OFF, if button is not pressed
		2	Set temperature Minus	2	LED = ON, if button is pressed or if nvoFanSpeed.state = -1 LED = OFF, if nvoFanSpeed.state ≠ -1
		3	Fan stage Plus incl. AUTO		
		4	Fan stage Minus incl. AUTO		
		5	Fan stage Plus <b>without</b> AUTO		
		6	Fan stage Minus <b>without</b> AUTO		
		7	Fan stage "only AUTO"		
Notice: The release of the operating buttons for air-conditioning control is depending on the current room occupancy (see Space Comfort Control Command Module). An adjustment of the values is only possible with nvoEffectOccup = OC_OCCUPIED.					

## UCPTtimeConfig

UCPT Index: 45, typedef struct {unsigned short Byte[4]} UNVT\_str\_hex4

Function: By UCPTtimeConfig the display mode of weekday, date and time can be configured.

!!Please note: After a module reset, the display for date and time is faded out. The ime is only faded in after

!! receipt of a telegram via the input variable *nviTimeSet*. The time should be synchronised by a LON system

!! clock once a day.

UCPTtimeConfig.Byte[0]	=	0	==>	<b>Weekday</b> faded out
UCPTtimeConfig.Byte[0]	=	1	==>	<b>Weekday</b> English
UCPTtimeConfig.Byte[0]	=	2	==>	<b>Weekday</b> German
UCPTtimeConfig.Byte[1]	=	0	==>	<b>Date</b> faded out
UCPTtimeConfig.Byte[1]	=	1	==>	<b>Date</b> English (JJ.MM.TT)
UCPTtimeConfig.Byte[1]	=	2	==>	<b>Datum</b> German (TT.MM.JJ)
UCPTtimeConfig.Byte[2]	=	0	==>	<b>Time</b> faded out
UCPTtimeConfig.Byte[2]	=	1	==>	<b>Time</b> with seconds
UCPTtimeConfig.Byte[2]	=	2	==>	<b>Time</b> without seconds
UCPTtimeConfig.Byte[3]	=	0	==>	<b>Time</b> 24-hours mode
UCPTtimeConfig.Byte[3]	=	1	==>	<b>Time</b> 12-hours mode

Preset values: .Byte[0] = 2, .Byte[1] = 2, .Byte[2] = 2, .Byte[3] = 0

## UCPTdisplConfig

UCPT Index: 46, SNVT\_state

Function: By means of UCPTtimeConfig the presentation mode of the weekday, date and time can be configured. If several measuring values are indicated, the display is switching back and forth between the measuring values in the time interval *UCPTdisplayTime*.

UCPTdisplConfig.bit[0]	=	1*	==>	<b>Room temperature</b> displayed
UCPTdisplConfig.bit[0]	=	0	==>	<b>Room temperature</b> not displayed
UCPTdisplConfig.bit[1]	=	1	==>	<b>Outdoor temperature</b> displayed
UCPTdisplConfig.bit[1]	=	0*	==>	<b>Outdoor temperature</b> not displayed
UCPTdisplConfig.bit[2]	=	1	==>	<b>Set temperature effective</b> displayed
UCPTdisplConfig.bit[2]	=	0*	==>	<b>Set temperature effective</b> not displayed
UCPTdisplConfig.bit[3]	=	1	==>	<b>Set point offset</b> displayed
UCPTdisplConfig.bit[3]	=	0*	==>	<b>Set point offset</b> not displayed
If bit2 = 1 and bit3 = 1, then the effective set point temperature is displayed				
UCPTdisplConfig.bit[4]	=	1*	==>	<b>°C</b> unit of measurement for the temperature
UCPTdisplConfig.bit[4]	=	0	==>	<b>°F</b> unit of measurement for the temperature
UCPTdisplConfig.bit[5]	=	1*	==>	<b>Decimal point for temperature</b> displayed
UCPTdisplConfig.bit[5]	=	0	==>	<b>Decimal point for temperature</b> not displayed
UCPTdisplConfig.bit[6]	=	1	==>	<b>Relative humidity</b> of the room air displayed
UCPTdisplConfig.bit[6]	=	0*	==>	<b>Relative humidity</b> of room air not displayed
UCPTdisplConfig.bit[7]	=	1	==>	<b>Relative humidity</b> of outdoor air displayed
UCPTdisplConfig.bit[7]	=	0*	==>	<b>Relative humidity</b> of outdoor air not displayed
UCPTdisplConfig.bit[8]	=	1*	==>	<b>Decimal point for relative humidity</b> displayed
UCPTdisplConfig.bit[8]	=	0	==>	<b>Decimal point for relative humidity</b> not displayed
UCPTdisplConfig.bit[9]	=	1*	==>	<b>Symbols room occupancy</b> displayed (status of nvoEffectOccup)
UCPTdisplConfig.bit[9]	=	0	==>	<b>Symbols room occupancy</b> not displayed
UCPTdisplConfig.bit[10]	=	1	==>	<b>Symbols fan</b> displayed
UCPTdisplConfig.bit[10]	=	0*	==>	<b>Symbols fan</b> not displayed
UCPTdisplConfig.bit[11]	=	1	==>	<b>Display set point offset</b> with set point adjustment
UCPTdisplConfig.bit[11]	=	0*	==>	<b>Set temperature effective</b> with set point adjustment
UCPTdisplConfig.bit[12]	=	1	==>	<b>Display setpoint</b> by first button press
UCPTdisplConfig.bit[12]	=	0*	==>	<b>Change setpoint</b> by first button press
UCPTdisplConfig.bit[13]	=	1	==>	<b>Use effective setpoint</b> for nviTempSetpoint
UCPTdisplConfig.bit[13]	=	0*	==>	<b>Use basic setpoint</b> for nviTempSetpoint

\* = preset values

## UCPTdisplayTime

UCPT Index: 16, SNVT\_time\_sec

Function: This configuration parameter defines the period of time of a display message respectively the up-date interval of the display. (preset value: 5,0 sec)

## UCPTbacklightMin[0],[1]

UCPT Index: 48, SNVT\_lev\_cont

Function: In the stable state, i.e. after module reset or if no button actuation is made for a period longer than 15 sec. T

The value *UCPTbacklightMin[0]* determines the background illumination of the LCD.

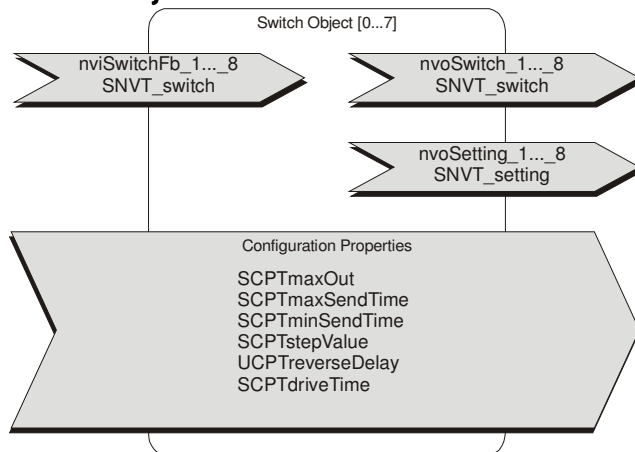
The value *UCPTbacklightMin[1]* determines the brightness of the label area.

## UCPTbacklightMax[0],[1]

UCPT Index: 48, SNVT\_lev\_cont

Function: If the stable state is left by actuation of a button, the value *UCPTbacklightMax[0]* determines the brightness of the background illumination of the LCD. The value *UCPTbacklightMax[1]* determines the brightness of the label area.

## Switch Object



Eight identical switch objects for light and blind control. Both, the allocation of the buttons to the individual objects and the determination of the functions are made by the configuration parameters UCPTdiConfig[0]...[11] in the Node Object.

Possible functions are light control ON/OFF, light control ON/OFF with dimming or blinds UP/DOWN with automatic run.

## Input Variables Switch Object:

### nviSwitchFb\_1...8

SNVT Type: SNVT\_switch, Index 95

Function: Input variable for the current status of the light groups controlled by Switch\_1...8.

## Output Variables Switch Object:

### nvoSwitch\_1...8, nvoSetting\_1...8

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

Function: Output variables for control of light groups and for control of light and blind controllers.

The output values are depending on the function adjustments. By UCPTdiConfig.Byte[0] = 0 - 7 buttons are allocated to a switch object. UCPTdiConfig.Byte[1] = 0 - 0C hex allocates a function to these buttons. It is also possible to allocate several buttons to a switch object, e.g. T5 for switching-off and T6 for switching-on the light.

### UCPTdiConfig.Byte[1] = 01<sub>hex</sub> Button actuated/ not actuated

Button actuated	nvoSwitch.value	= SCPTmaxOut
	nvoSwitch.state	= 1
	nvoSetting.function	= SET_ON;
	nvoSetting.setting	= SCPTmaxOut;
Button not actuated	nvoSwitch.value	= 0
	nvoSwitch.state	= 0

nvoSetting.function = SET\_OFF;  
nvoSetting.setting = 0;

#### **UCPTdiConfig.Byte[1] = 02<sub>hex</sub>, Light Toggle**

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

Lighting ON	nvoSwitch.value	= SCPTmaxOut
	nvoSwitch.state	= 1
Lighting OFF	nvoSetting.function	= SET_ON;
	nvoSetting.setting	= SCPTmaxOut;
	nvoSwitch.value	= 0
	nvoSwitch.state	= 0
	nvoSetting.function	= SET_OFF;
	nvoSetting.setting	= 0;

#### **UCPTdiConfig.Byte[1] = 03<sub>hex</sub>, Lighting ON**

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

Lighting ON	nvoSwitch.value	= SCPTmaxOut
	nvoSwitch.state	= 1
	nvoSetting.function	= SET_ON;
	nvoSetting.setting	= SCPTmaxOut;

#### **UCPTdiConfig.Byte[1] = 04<sub>hex</sub>, Lighting OFF**

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

Lighting OFF	nvoSwitch.value	= 0
	nvoSwitch.state	= 0
	nvoSetting.function	= SET_OFF;
	nvoSetting.setting	= 0;

#### **UCPTdiConfig.Byte[1] = 05<sub>hex</sub>, Light Toggle with Dimming, Switch-on value = max. Value**

Short-term button actuations (< 1 s) lead to a toggle of the current light status, whereas the .value - switching-on value always amounts to SCPTmaxOut. By a longer button actuation (> 1 s) the dimming function is activated, i.e. based on the current light status, the .value-value of the switch variable is increased or reduced as long as the button is actuated.

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

The variables nvoSetting are not changed in this setting.

#### **UCPTdiConfig.Byte[1] = 06<sub>hex</sub>, Light Toggle with Dimming, Switch-on value = last On-value**

Function as with 05hex, but with the difference, that the value SCPTmaxOut is not taken over when switching-on the light, but the last switch-value. The smallest switch-on value is limited to 20%.

#### **UCPTdiConfig.Byte[1] = 07<sub>hex</sub>, Lighting ON with Brighter Dimming, Switch-on value = max. Value**

If the lighting is switched-off, a button actuation leads to an immediate switching-on of the lighting. By longer button actuations (> 1 s) the function „brighter dimming“ is activated, i.e. based on the current light status, the .value-value of the switch variables is increased in percent steps by UCPTstepValue as long as the maximum value SCPTmaxOut is reached. The

sending interval in the dimming mode is set by SCPTminSendTime and amounts to approx. 300ms., preset. ca. 300ms.

Switching-on of light	nvoSwitch.value	= SCPTmaxOut
	nvoSwitch.state	= 1
Brighter dimming	nvoSetting.function	= SET_ON;
	nvoSetting.setting	= SCPTmaxOut;
	nvoSwitch.value	= last value + UCPTstepValue
	nvoSwitch.state	= 1
	nvoSetting.function	= SET_UP;
	nvoSetting.setting	= UCPTstepValue;

**UCPTdiConfig.Byte[1] = 08<sub>hex</sub> Lighting ON with Brighter Dimming, Switch-on value = last On-value**

Function as with 07hex, but with the difference, that the value SCPTmaxOut is not taken over when switching-on the light, but the last switch-on value. The smallest switch-on value is limited to 20%.

**UCPTdiConfig.Byte[1] = 09<sub>hex</sub> Lighting OFF with Darker Dimming**

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

Switching-off of lighting	nvoSwitch.value	= 0
	nvoSwitch.state	= 0
Darker dimming of lighting	nvoSetting.function	= SET_OFF;
	nvoSetting.setting	= 0;
	nvoSwitch.value	= last value - UCPTstepValue
	nvoSwitch.state	= 1
	nvoSetting.function	= SET_DOWN;
	nvoSetting.setting	= UCPTstepValue;

**UCPTdiConfig.Byte[1] = 0A<sub>hex</sub> Blind UP**

In the configuration mode „Blind UP“ only the nvoSetting-Variables are changed and sent. Short-term button actuations (< 2 s) are designed for the fine adjustment of the lamellas. A long-term button actuation (> 2 s) starts the automatic run and controls the blind for the time SCPTdriveTime continuously into the direction open. The automatic run can be stopped by a renewed button actuation.

Run-up of blind	nvoSetting.function	= SET_UP;
Stop blind	nvoSetting.function	= SET_STOP;

The command SET\_NUL is sent with a delay of 500ms after the command SET\_STOP was sent .

**UCPTdiConfig.Byte[1] = 0B<sub>hex</sub> Blind DOW**

In the configuration mode „Blind DOWN“ only the nvoSetting-Variables are changed and sent. Short-term button acutations (< 2 s) are designed for the fine adjustment of the lamellas. A long-term button actuation (> 2 s) starts the automatic run and controls the blind for the time SCPTdriveTime continuously into the direction close. The automatic run can be stopped by a renewed button actuation.

Run-down of blind	nvoSetting.function	= SET_DOWN;
Stop blind	nvoSetting.function	= SET_STOP;

The command SET\_NULL is sent with a delay of 500ms after the command SET\_STOP is sent.

**UCPTdiConfig.Byte[1] = 0C<sub>hex</sub> Command Automatic**

The actuation of an „Automatic-Button“ switches the variables nvoSwitch to the value 0,0 -1. Thus, it is possible to set e.g. a light controller from an external override back to the automatic run.

**Configuration Parameter Switch Object:**

**SCPTmaxOut**

SCPT Index: 93, SNVT\_lev\_cont

Function: This configuration parameter determines the maximum output value of the variable nvoSwitch.value.

**SCPTmaxSendTime**

SCPT Index: 49, SNVT\_time\_sec

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

**SCPTminSendTime**

SCPT Index: 52, SNVT\_time\_sec

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

**SCPTstepValue**

SCPT Index: 92, SNVT\_lev\_cont

Function: This configuration parameter defines the step size of the variables nvoSwitch.value in the dimming mode.

**UCPTreverseDelay**

UCPT Index: 14, SNVT\_count

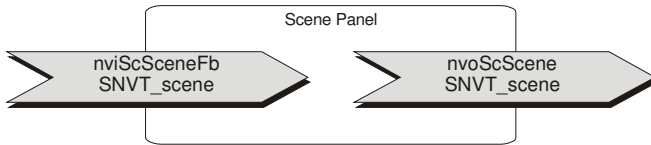
Function: The configuration parameter defines the reverse delay with a rotation reverse of the blind motors in ms. Thus an alternate command e.g. from nvoSetting = SET\_UP to nvoSetting = SET\_DOWN is output with a delay. (Preset value: 500 ms)

**SCPTdriveTime**

UCPT Index: 45, SNVT\_time\_sec

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

## Scene Panel



### Input Variables Scene Panel:

#### nviScSceneFb

SNVT Type: SNVT\_scene, Index 115

Function: Input variable for current illumination scene. If a new input value is received the button an led configuration will be checked. If configured, leds indicates the current scene.

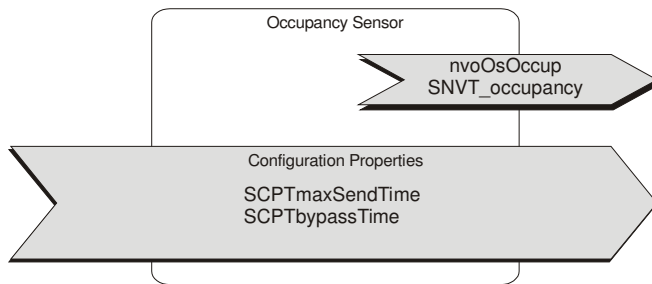
### Output Variables Scene Panel:

#### nvoScScene

SNVT Type: SNVT\_scene, Index 115

Function: Output variable for control of a scene panel. The output values are depending on the function adjustments. By UCPTdiConfig.Byte[0] = 8 the buttons are allocated to the scene object. UCPTdiConfig.Byte[1] = 01<sub>hex</sub> - FF<sub>hex</sub> allocates a scene number to these buttons, which are called-off with short-term button actuations by SC\_RECALL. With long-term button actuations (> 2s) the scene is learned in.

## Occupancy Sensor Object



### Output Variables Occupancy Sensor Object:

#### nvoOsOccup

SNVT Type: SNVT\_occupancy, Index 109

Function: Output variable for presence status message in rooms. The output values are depending on the function settings. By UCPTdiConfig.Byte[0] = 9 buttons are allocated to the occupancy sensor object. UCPTdiConfig.Byte[1] = 01<sub>hex</sub> - 04<sub>hex</sub> allocates a function to these buttons. It is also possible to allocate several buttons to an occupancy sensor object. nvoOsOccup can be bound on the input variable nviOccSensor for local presence status message.

#### UCPTdiConfig.Byte[1] = 01<sub>hex</sub>, Presence button with overtime function

By button actuation the output variable nvoOsOccup receives the value OC\_OCCUPIED and the reset time is started. After expiration of the delay time SCPTbypassTime the output variable is set back to the value UNOCCUPIED. Each button actuation restarts the reset time.

#### UCPTdiConfig.Byte[1] = 02<sub>hex</sub>, Presence button with toggle function

By button actuation the output variable nvoOsOccup switches forth and back between the values OC\_OCCUPIED and OC\_UNOCCUPIED.

#### UCPTdiConfig.Byte[1] = 03<sub>hex</sub>, Presence button OCCUPIED

By button actuation the output variable nvoOsOccup receives the value OC\_OCCUPIED.

#### UCPTdiConfig.Byte[1] = 04<sub>hex</sub>, Presence button UNOCCUPIED

By button actuation the output variable nvoOsOccup receives the value OC\_UNOCCUPIED.

### Configuration Parameter Occupancy Sensor Object:

#### SCPTbypassTime

SCPT Index: 34, SNVT\_time\_min

Function: Delay time in minutes. After expiration of SCPTbypassTime the output variable nvoOsOccup is set back to OC\_UNOCCUPIED. (Preset value: 90 min)

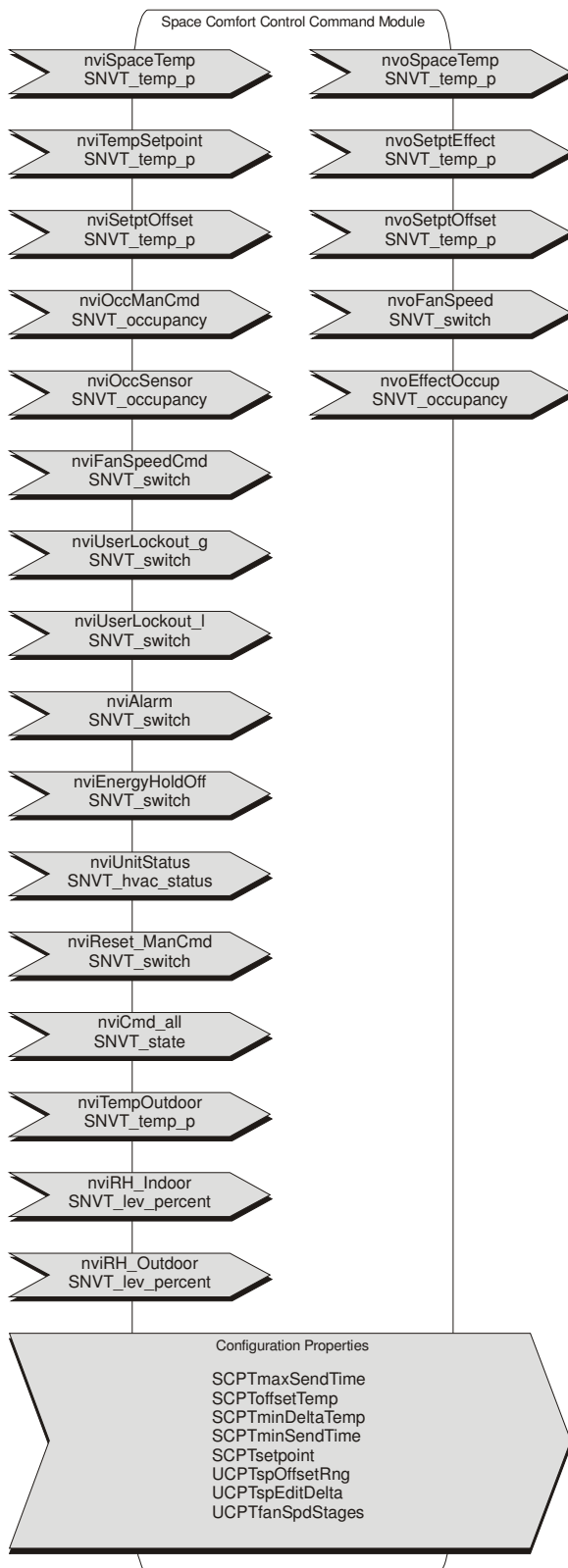
#### SCPTmaxSendTime

SCPT Index: 49, SNVT\_time\_sec

Function: Heartbeat function. Stipulates the interval time after which the output variables are sent. By means of input values = 0, the heartbeat function is deactivated. (Preset value: 120s)

## Space Comfort Control Command Module:

### Space Comfort Control Command Module:



**Temperature Detection:** Temperature detection is either made by the integrated temperature sensor or via the input variable *nviSpaceTemp* by an external LON sensor. For an afterwards calibration of the internal sensor the configuration parameter *SCPTOffsetTemp* is used.

**Measurement Indication:** Additionally to the measured room temperature it is possible to show the input variables *nviTempOutdoor*, *nviRH\_Indoor* and *nviRH\_Outdoor* in the display. The configuration of the display is made in the Node Object by means of the parameters *UCPTdisplConfig* and *UCPTdisplayTime*. If more than one measuring value should be shown, the display switches forth and back between the measuring values in the time interval.

**Set Point Default:** The effective set point (basic set point) *nvoSetptEffect* is calculated in reliance on the input variables to the room occupancy (*nviOccManCmd* and *nviOccSensor*) of the set point defaults *SCPTsetpoint* respectively *nviTempSetpoint* and the offset value *nvoSetptOffset* (see table 1).

If multiple devices are used for setpoint adjustment in a single room it's possible to synchronize the user settings by means of *nviSetptOffset* and *nvoSetptOffset*.

For the evaluation of internal occupancy status message the output variable *nvoOcOccup* of the „Occupancy Sensor“ must be combined with the input variable *nviOccSensor*.

The offset value *nvoSetptOffset* can be raised respectively lowered via appropriate parameterized operating buttons within a range *UCPTspOffsetRng*.

**Fan Adjustment:** The output value for the fan stages *nvoFanSpeed* can be adjusted by appropriate configured buttons. The manual user settings can be override by means of *nviFanSpeedCmd*. Both display and output variable are take over the values of *nviFanSpeedCmd*.

## Input Variables Space Comfort Control Command Module:

### nviSpaceTemp

SNVT Type: SNVT\_temp\_p, Index 105

Function: Input variable for connection of an external LON temperature sensor. The external value is taken over, if the initialisation value 0x7FFF (=327,67 °C) was changed by a NV-Update after reset. As long as the initialisation value is not changed after reset, the internal temperature sensor remains active.

### nviTempSetpoint

SNVT Type: SNVT\_temp\_p, Index 105

Function: Input variable for default of the basic set point temperature.

It is not obligatory necessary to bind this network variable with a higher-ranged node. If no update is made for nviSetpoint, the initialisation value 0x7FFF (=327,67°C) is kept and the value of the configuration parameter **SCPTsetpoint** is used for the calculation of the effective set point (basic setpoint + offset).

If nviTempSetpoint receives an update with a valid set point, the effective set point is calculated by the value of the input variables.

### nviSetptOffset

SNVT Type: SNVT\_temp\_p, Index 105

Function: Input variable for external default of the setpoint correction. If a new value is received (*nviSetptOffset* unequal *nvoSetptOffset*), *nvoSetptOffset* takes over the value of *nviSetptOffset*. Thus it's possible to synchronize multiple devices in a single room.

### nviOccManCmd and nviOccSensor

SNVT Type: SNVT\_occupancy, Index 109

Function: Input variable for default of the room occupancy. The current room occupancy determines the values of the control parameter „effective set point“ and „neutral zone“ and thus the set points for heating and cooling (see table 1). Initialisation value for both variables: OC\_NUL

nviOccManCmd: default via building control technic:

OC\_OCCUPIED, OC\_BYPASS, OC\_STANDBY, OC\_UNOCCUPIED

nviOccSensor: presence status message in rooms:

OC\_OCCUPIED, OC\_UNOCCUPIED

nviOccManCmd	nviOccSensor	>>>	nvoEffectOccup	nvoSetptEffect
OC_NUL	OC_NUL	>>>	OCCUPIED	SCPTsetpoint + nvoSetptOffset  or nviTempSetpoint + nvoSetptOffset
OC_OCCUPIED	****	>>>		
OC_BYPASS	OC_OCCUPIED	>>>		
OC_NUL	OC_OCCUPIED	>>>		
OC_STANDBY	****	>>>	STANDBY	SCPTsetpoint + nvoSetptOffset  or nviTempSetpoint + nvoSetptOffset
OC_NUL	OC_UNOCCUPIED	>>>	UNOCCUPIED	SCPTsetpoint  or nviTempSetpoint
OC_BYPASS	OC_UNOCCUPIED	>>>		
OC_UNOCCUPIED	****	>>>		

**Table 1: Effective set point depending on the room occupancy**

### **nviFanSpeedCmd**

SNVT Type: SNVT\_switch, Index 95

Function: Input variable to set a new fan speed value for display and nvoFanSpeed.

### **nviUserLockout\_g**

SNVT Type: SNVT\_switch, Index 95

Function: Input variable for locking the operating buttons for scene control, room occupancy and air-conditioning control.

nviUserLockout_g	=	100.0 1	=>	buttons are locked
nviUserLockout_g	=	0.0 0	=>	buttons are released

### **nviUserLockout\_l**

SNVT Type: SNVT\_switch, Index 95

Function: Input variable for locking the operating buttons for light and blind control.

nviUserLockout_l	=	100.0 1	=>	buttons are locked
nviUserLockout_l	=	0.0 0	=>	buttons are released

### **nviAlarm**

SNVT Type: SNVT\_switch, Index 95

Function: The input variable controls the LCD symbol „failure“:

nviAlarm =	100.0	1	==>	symbol „failure“ on
nviAlarm =	50.0	1	==>	symbol „failure“ blinks
nviAlarm =	0.0	0	==>	symbol „failure“ off

### **nviEnergyHoldOff**

SNVT Type: SNVT\_switch, Index 95

Function: The input variable controls the LCD symbol „window“:

nviEnergyHoldOff =	100.0	1	==>	symbol „window“ on
nviEnergyHoldOff =	0.0	1	==>	symbol „window“ off

### **nviUnitStatus**

SNVT Type: SNVT\_hvac\_status, Index 112

Function: The input variable controls the LCD symbol „heating“ and „cooling“:

nviUnitStatus.heat_output_primary	>	0	==>	symbol „heating“ ON
nviUnitStatus.heat_output_primary	=	0	==>	symbol „heating“ OFF
nviUnitStatus.cool_output	>	0	==>	symbol „heating“ ON
nviUnitStatus.cool_output	=	0	==>	symbol „heating“ OFF

### **nviReset\_ManCmd**

SNVT Type: SNVT\_switch, Index 95

Function: An update of the variables from 0.0 0 to 100.0 1 leads to a setting back to the user adjustments.

Set point offset temperature: nvoSetptEffect, nvoSetptOffset = 0

Room occupancy: nvoOsOccup = OC\_UNOCCUPIED

Fan stage: nvoFanSpeed = 0,0 -1 (AUTO)

### **nviCmd\_all**

SNVT Type: SNVT\_state, Index 83

Function:

nviCmd_all.bit[0]	=	1	==>	see <b>nviUserLockout_g</b>	=	100.0	1
nviCmd_all.bit[0]	=	0	==>	see <b>nviUserLockout_g</b>	=	0.0	0
nviCmd_all.bit[1]	=	1	==>	see <b>nviUserLockout_l</b>	=	100.0	1
nviCmd_all.bit[1]	=	0	==>	see <b>nviUserLockout_l</b>	=	0.0	0
nviCmd_all.bit[2]	=	1	==>	see <b>nviReset_ManCmd</b>	=	100.0	1
nviCmd_all.bit[2]	=	0	==>	see <b>nviReset_ManCmd</b>	=	0.0	0
nviCmd_all.bit[3]	=	1	==>	switch-on symbol <b>Heating</b>			
nviCmd_all.bit[3]	=	0	==>	switch-off symbol <b>Heating</b>			
nviCmd_all.bit[4]	=	1	==>	switch-on symbol <b>Cooling</b>			
nviCmd_all.bit[4]	=	0	==>	switch-off symbol <b>Cooling</b>			
nviCmd_all.bit[5]	=	1	==>	switch-on symbol <b>Window</b>			
nviCmd_all.bit[5]	=	0	==>	switch-off symbol <b>Window</b>			
nviCmd_all.bit[6]	=	1	==>	switch-on symbol <b>Failure</b>			
nviCmd_all.bit[6]	=	0	==>	switch-off symbol <b>Failure</b>			

### **nviTempOutdoor**

SNVT Type: SNVT\_temp\_p, Index 105

Function: By this input variable the outdoor temperature can be prescribed additionally.

### **nviRH\_Indoor**

SNVT Type: SNVT\_lev\_percent, Index 81

Function: By this input variable the relative humidity of the room air can be prescribed additionally.

### nviRH\_Outdoor

SNVT Type: SNVT\_lev\_percent, Index 81

Function: By this input variable the relative humidity of the outdoor air can be prescribed additionally.

## Output Variables Space Comfort Control Command Module:

### nvoSpaceTemp

SNVT Type: SNVT\_temp\_p, Index 105

Function: Output variable for measured temperature value. Measuring range 0 - 50°C, resolution 1/100 °C. Data output is made depending on the configuration parameter SCPTmaxSendTime, SCPTminDeltaTemp and SCPTminSendTime and approx. 1,5s - 4,0s after reset.

### nvoSetptEffect

SNVT Type: SNVT\_temp\_p, Index 105

Function: Output variable for effective set point. The effective set point is calculated via the defaults of nviTempSetpoint respectively SCPTsetpoint and set point shift adjusted at the device (see table 1). Data output is made after value change by the user, depending on the configuration parameters SCPTmaxSendTime and approx. 1,5s - 4,0s after reset.

### nvoSetptOffset

SNVT Type: SNVT\_temp\_p, Index 105

Function: Output variable for set point correction, which can be changed via the operating buttons. Data output is made with any button actuation. By UCPTdiConfig.Byte[0] = 0A<sub>hex</sub> buttons are allocated to the space comfort control command module. UCPTdiConfig.Byte[1] = 01<sub>hex</sub> - 02<sub>hex</sub> allocates a function for set point adjustment to these buttons. It is also possible to allocate several buttons to the space comfort controller object.

#### UCPTdiConfig.Byte[1] = 01<sub>hex</sub>, Increase set point offset

By button actuation the output variable nvoSetptOffset is increased by the value UCPTspEditDelta. The adjustment range is determined by UCPTspOffsetRng.

#### UCPTdiConfig.Byte[1] = 02<sub>hex</sub>, Reduce set point

By button actuation the output variable nvoSetptOffset is reduced by the value UCPTspEditDelta. The adjustment range is determined by UCPTspOffsetRng.

!! The actuation of the buttons is **only** evaluated by **nvoEffectOccup = OC\_OCCUPIED**. Otherwise button actuations are ignored. For the time of the set point adjustment, the indication in the display either shows the effective set point nvoEffectSetpt or the offset value nvoSetptOffset depending on the configuration.

### nvoEffectOccup

SNVT Type: SNVT\_occupancy, Index 109

Function: Output variable for the effective room occupancy (see table 1).

Data output is made depending on SCPTmaxSendTime, with value changes and approx. 1,5s - 4,0s after reset.

### nvoFanSpeed

SNVT Type: SNVT\_switch, Index 95

Function: Output variable for adjusted fan speed. The number of the fan speed stages is adjusted by the configuration parameter UCPTfanSpdStages. The indication in the display shows the currently adjusted fan speed and whether the controller is in the automatic mode.

UCPTfanSpdStages = 1

Lüfterstufe	nvoMultiOut .value	nvoMultiOut .state
AUTO	0 %	-1
0	0 %	1
1	100 %	1

UCPTfanSpdStages = 2

Lüfterstufe	nvoMultiOut .value	nvoMultiOut .state
AUTO	0 %	-1
0	0 %	1
1	50 %	1
2	100 %	1

UCPTfanSpdStages = 3

Lüfterstufe	nvoMultiOut .value	nvoMultiOut .state
AUTO	0 %	-1
0	0 %	1
1	33,0 %	1
2	66,5 %	1
3	100 %	1

!!The acutation of the buttons is **only** evaluated with **nvoEffectOccup = OC\_OCCUPIED**. Otherwise, button !!actuations are ignored.

By UCPTdiConfig.Byte[0] = 0A<sub>hex</sub> buttons are allocated to the space comfort controller object. UCPTdiConfig.Byte[1] = 03<sub>hex</sub> - 04<sub>hex</sub> allocates a function for fan speed adjustment to these buttons. It is also possible to allocate several buttons to the space comfort controller object.

#### UCPTdiConfig.Byte[1] = 03<sub>hex</sub> Increase Fan Speed Stage incl. AUTO

By button actuation the output variable nvoFanSpeed is increased. A button actuation with max. fan stage switches into the mode "Automatic", i.e. the fan stages are automatically adapted to the control variables for heating and cooling (please see the controller function diagram).

#### UCPTdiConfig.Byte[1] = 04<sub>hex</sub> Reduce Fan Speed Stage incl. AUTO

By button actuation, the output variable nvoFanSpeed is reduced. A button actuation with min. fan stage switches into the mode "Automatic", i.e. the fan stages are automatically adapted to the control variables for heating and cooling (please see the controller function diagram). If the fan is switched-off manually, the control variables for heating and cooling will be set back to 0% as well.

#### UCPTdiConfig.Byte[1] = 05<sub>hex</sub> Increase Fan Speed Stage without AUTO

By button actuation the output variable nvoFanSpeed is increased. A button actuation with max. fan stage switches into the mode „0“. The value AUTO will be jumped over.

#### UCPTdiConfig.Byte[1] = 06<sub>hex</sub> Reduce Fan Speed Stage without AUTO

By button actuation, the output variable nvoFanSpeed is reduced. A button actuation with fan stage „0“ switches into the mode of max. fan stage. The value AUTO will be jumped over.

#### UCPTdiConfig.Byte[1] = 07<sub>hex</sub> Fan Speed AUTO

By button actuation, the output variable nvoFanSpeed is set to 0.0 -1.

## Configuration Parameter Space Comfort Control Command Module

### SCPTmaxSendTime

Index: 49, SNVT\_time\_sec

Function: Heartbeat function. This configuration parameter stipulates the interval time after which the output variables are sent. By means of input values =0, the heartbeat function is deactivated. (Preset value: 300,0 s)

### SCPTminSendTime

Index: 52, SNVT\_time\_sec

Function: Stipulates the smallest update interval of the output variables nvoSpaceTemp. An update is made after expiration of „SCPTminSendTime“, if the temperature value of the output variables has changed by more than „SCPTminDeltaTemp“. By input values = 0 the function is deactivated. (Preset value: 5,0 sec)

### SCPTminDeltaTemp

Index: 64, SNVT\_temp\_p

Function: If the temperature changes by the adjusted value „SCPTminDeltaTemp“, the new temperature value is transmitted. The function is depending on the adjustment of the parameter „SCPTminSendTime“. (Range  $\geq 0\text{ °C}$ ; Preset value: 0,30 °C)

### SCPToffsetTemp

Index: 70, SNVT\_temp\_p

Function: Offset for the temperature value. By this parameter a software calibration is possible. Please see the notices for flush-mounting room temperature sensors with transducer.

### SCPTsetpoint

Index: 213, SNVT\_temp\_p

Function: Configuration parameter for set point default (Preset value: 22,00 °C)

### UCPTspEditDelta

Index: 9, SNVT\_temp\_p

Function: Configuration parameter for definition of temperature jumps via the operating buttons. (Preset value: 0.5K, i.e. the set point is changed by 0,5K with any button actuation).

### UCPTspOffsetRng

Index: 12, SNVT\_temp\_p

Function: Configuration parameter for the value range of the adjusted set point correction, i.e. the prescribed set point can be changed by the user by means of the value +/- UCPTspOffsetRng. (Preset value: 3,0 K)

### UCPTfanSpdStages

Index: 13, SNVT\_count

Function: Configuration parameter for default of fan stages.  
(Preset value: 3 ==> OFF, 33,0 %, 66,5 %, 100,0 %, AUTO)