

## DE - Softwarebeschreibung

Technische Änderungen vorbehalten  
Stand 11.04.07

# 29100...

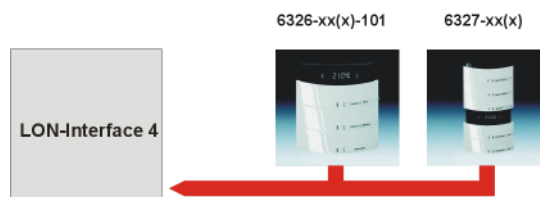


## 1 Anwendung

Mit dem LON-Interface 4 (bus coupling unit) können Geräte aus dem Busch-Jaeger EIB Programm:

- triton: 6326-xx und 6327-xx

an das Gebäude-Kommunikationssystem LON der Fa. Echelon angekoppelt werden. Dabei dient das LON-Interface als Schnittstelle zwischen dem LON Bus und des spezifischen Busch-Jaeger Geräts.



Die Vorgaben des LonMark® Funktionsprofils **3200** „Switch“ und **3250** „Scene Panel“ wurden berücksichtigt. Das Funktionsprofil (UFTP) **20001** „SpaceComfortController“ ist den Vorgaben des LonMark® Funktionsprofils **8500** „SpaceComfortController“ angepasst. Für erweiterte Einstellmöglichkeiten werden benutzerdefinierte Konfigurationsparameter (UCPT) genutzt. Die hier verwendeten UCPTs sind in den Thermokon Device Recource Files ab Version 2.1 oder höher definiert.

## 2 Geräteübersicht

6326-xx Raumtemperaturregler mit Tastsensor 3fach mit Heiz-/Kühlbetrieb

6327-xx Raumtemperaturregler mit Tastsensor 5fach mit Heiz-/Kühlbetrieb mit IR

- Universal Bedienelement mit Raumtemperaturreglerfunktion, 6327 mit IR-Empfang
- Zum Heizen und Kühlen (PI, PWM oder 2-Punkt)
- Tasterfunktion (Schalten/Dimmen/Jalousie/Wert senden/Lichtszenen/Lüftungsfunktion)
- Bedienelemente: Tastkontakte links/rechts, auch zur Sollwert- und Betriebsartenwahl
- Anzeigeelemente: Anzeige der Betriebsart, Temperatur, Lüfterstellung über LCD

## 3 Parametrierung der Wippen

### Sollwert, Raumbelugung

Um den Sollwert bzw. die Raumbelugung zu ändern muss über die Zusatztaste der Betriebsartenmodus gewechselt werden. Der Sollwert ist fest der Wippe 1 und die Raumbelugung ist fest der Wippe 2 zugeordnet.



### Tasterfunktion

Zum Parametrieren der einzelnen Wippen wird der Konfigurationsparameter *UCPTdiConfig[...]* im Node Object verwendet. Bei der Benutzung des LONMakers kann durch einen Doppelklick auf den Parameter das unten dargestellte Konfigurierungsfenster geöffnet werden.

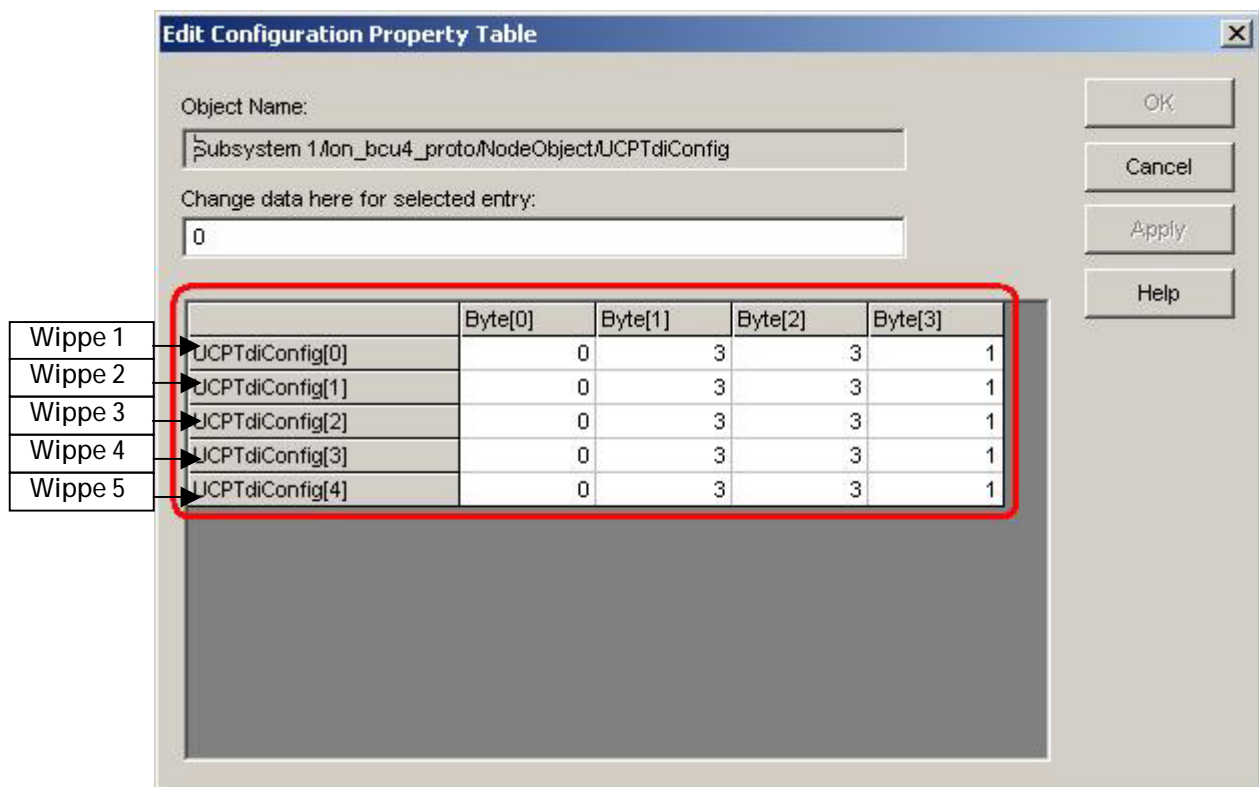


Abbildung 3-1: LONmaker

### UCPTdiConfig[x]

Dabei wird die:

- Wippe 1 über UCPTdiConfig[0]
  - Wippe 2 über UCPTdiConfig[1]
  - Wippe 3 über UCPTdiConfig[2]
  - Wippe 4 über UCPTdiConfig[3]
  - Wippe 5 über UCPTdiConfig[4]
- konfiguriert.

### Byte[0]

Die Funktionen der Wippe wird über das Byte[0] konfiguriert. Folgende Funktionen sind möglich:

- 0 - Schalten
- 1 – Dimmen
- 2 – Jalousie
- 3 - Rollladen
- 4 – Szene
- 5 - Lüfterverstellung

Beispiel: Die Wippe 1 soll als Dimmfunktion und die Wippe 2 als Jalousie konfiguriert werden.  
UCPTdiConfig[0].Byte[0] = 1  
UCPTdiConfig[1].Byte[0] = 2

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

Über das Byte[1] wird die Funktion der linken und mit Byte[2] die Funktion der rechten Taste der Wippe konfiguriert. Es kann z.B. ausgewählt werden, ob die linke Taste das Licht Aus (Byte[1] = 6) oder Ein (Byte[1] = 4) schalten soll. (siehe auch Kapitel:4.1.3)

Beispiel: Die linke Taste soll das Licht Ein / heller und die rechte Aus / dunkler schalten / dimmen.  
UCPTdiConfig[0].Byte[1] = 3  
UCPTdiConfig[0].Byte[2] = 7

### Byte[3]

Die LED der Wippe wird im Byte[3] eingestellt. So kann ausgewählt werden, ob die Farbe zwischen rot und grün bei einer Tastenbetätigung geändert werden soll. (siehe auch Kapitel:4.1.3)

Beispiel: Die LED soll bei einem linken Tastendruck rot und bei einem rechten grün sein.  
UCPTdiConfig[0].Byte[3] = 1

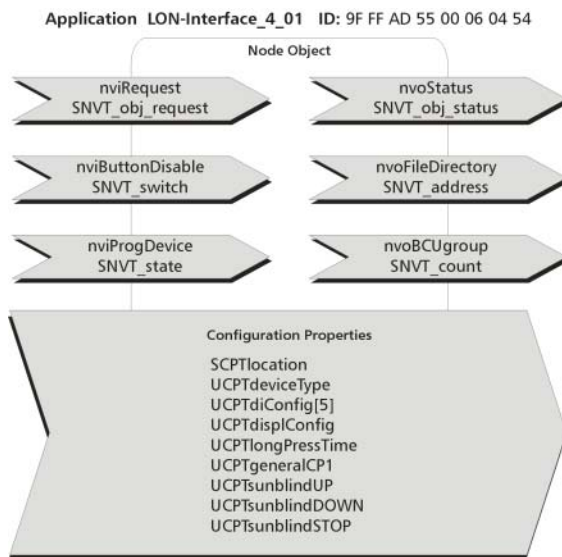
### Übertragen der Daten

Nachdem Parameter geändert wurden sind, müssen die Parameter zum Bedienelement übertragen werden. Mit setzen von 0 auf 1 von nviProgDevice.bit0 bzw. nach einem Spannungsreset werden diese neuen Einstellungen in das Bedienteil übernommen.

## 4 Softwarebeschreibung

### 4.1 Node Object

Das Node Objekt überwacht und steuert die Funktionen der einzelnen Objekte im Gerät. Unterstützt wird die von LonMark® geforderte Grundfunktionalität, wobei allgemeine Netzwerkvariablen und Konfigurationsparameter zur Steuerung und Parametrierung des Gerätes eingefügt wurden.



#### Übertragung der Daten:

Nach einer Parametrierung muss in *nviProgDevice.bit0* von 0 auf 1 eingestellt werden, damit die neuen Parameter übernommen werden. Neue Parameter werden auch nach einem Spannungsreset übernommen.

#### Messwertanzeigen:

Neben der Raumtemperatur kann auch der Sollwert angezeigt werden. Diese Konfiguration erfolgt über *UCPTdisplConfig*.

#### Tasterfunktionalität und Gerätevarianten:

Während der Inbetriebnahme muss über den Parameter *UCPTdeviceType* der entsprechend verwendete Gerätetyp (6326-xx, 6327-xx) eingestellt werden. Den Tastern können unterschiedliche Funktionen z.B. Beleuchtungssteuerung oder Szenenaufwurf zugewiesen werden. Mit *UCPTdiConfig[0]*, *UCPTdiConfig[1]*, *UCPTdiConfig[2]*, *UCPTdiConfig[3]* und *UCPTdiConfig[4]* erfolgt die Konfiguration der Wippen 1, 2, 3, 4 und 5.

#### 4.1.1 Eingangsvariablen Node Object:

##### nviRequest

SNVT Typ: SNVT\_obj\_request, Index 92

Funktion: Eingangsvariable mit den Funktionen RQ\_NORMAL, RQ\_UPDATE\_STATUS und RQ\_REPORT\_MASK.

##### nviButtonDisable

SNVT Typ: SNVT\_switch, Index 95

Funktion: Eingangsvariable zum Deaktivieren der Tasten. (Nur für Schalten, Dimmen, Jalousie, Rollläden, Szene)

##### nviProgDevice

SNVT Typ: SNVT\_state, Index 83

Funktion: Eingangsvariable zum Parametrieren des Bedienteils. Nach einer Parametrierung muss *nviProgDevice.bit0* von 0 auf 1 eingestellt werden, damit die neuen Parameter im Bedienteil übernommen werden.

*nviProgDevice.bit0* = 0 → *nviProgDevice.bit0* = 1

Übernahme der neuen Parameter.

#### 4.1.2 Ausgangsvariablen Node Object:

##### nvoStatus

SNVT Typ: SNVT\_obj\_status, Index 93

Funktion: Ausgangsvariable mit den geforderten Status Bits „invalid\_id“ und „invalid\_request“.

##### nvoFileDirectory

SNVT Typ: SNVT\_address, Index 114

Funktion: Die Ausgangsvariable stellt dem LON-Integrationstool die Adressdaten der Konfigurationsparameter im Gerät zur Verfügung.

##### nvoBCUgroup

SNVT Typ: SNVT\_count, Index 8

Funktion: Ausgangsvariable mit dem LON-Interface Typ. Der LON-Interface Typ kann nicht verändert werden.

## Softwarebeschreibung

### 4.1.3 Konfigurationsparameter Node Object:

#### SCPTlocation

SCPT Index: 17, SNVT\_str\_asc

Funktion: Zusätzliche Eingabemöglichkeit um Informationen zur Standortkennung im Gerät speichern zu können.

#### UCPTdeviceType

UCPT Index: 42, SNVT\_count

Funktion: Die Anpassung der Software erfolgt über diesen Konfigurationsparameter. Gültige Eingabewerte sind für:

6326-xx ==> 6326

6327-xx ==> 6327

Voreingestellter Wert: 0

#### UCPTdisplConfig

UCPT Index: 46, SNVT\_state

Funktion: Mit UCPTdisplConfig lässt sich die Darstellungsform konfigurieren.

Temperatur:

UCPTdisplConfig.

bit0 bit1

0 0\*

==> **Raumtemperatur** anzeigen

0 1

==> **keine Temperaturanzeige**

1 0

==> **Solltemperatur** anzeigen

Displayeinstellung:

Bit2 Bit3

0 0

==> **Display-Beleuchtung** immer Aus

0 1\*

==> **Display-Beleuchtung** nur bei Bedienung über Wippe 3

1 0

==> **Display-Beleuchtung** bei Nachtbetrieb aus

1 1

==> **Display-Beleuchtung** immer Ein

\* = Voreingestellte Werte

#### UCPTgeneralCP1

UCPT Index: 7, SNVT\_state

Funktion: Auswahl des IR-Bereichs der Fernbedienung. (IR-Bedienung ist nur beim 6327 möglich)

bit0

0

→ Weiß

1

→ Blau

#### UCPTlongPressTime

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

Funktion: Mit diesem Konfigurationsparameter kann die Zeit für einen langen Tastendruck für Dimmen, Jalousie, Szene und Universell eingegeben werden.

Beim 6326 ist der lange Tastendruck für Jalousie fest auf 900 ms eingestellt.

(Voreingestellter Wert: 1.0;2.0;2.0;2.0)

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

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

Funktion: Diese Konfigurationsparameter bestimmen die Wippenfunktionen und deren Zuordnung zu den Ausgangsvariablen, sowie die Ansteuerlogik der Rückmelde-LEDs. UCPTdiConfig[0] ist fest der Wippe 1, UCPTdiConfig[1] der Wippe 2, UCPTdiConfig[2] der Wippe 3, UCPTdiConfig[3] der Wippe 4, UCPTdiConfig[4] der Wippe 5 zugeordnet..

UCPTdiConfig[0] konfiguriert **Wippe 1**

UCPTdiConfig[1] konfiguriert **Wippe 2**

UCPTdiConfig[2] konfiguriert **Wippe 3**

UCPTdiConfig[3] konfiguriert **Wippe 4**

UCPTdiConfig[4] konfiguriert **Wippe 5**

UCPTdiConfig[x].Byte[0] konfiguriert **Funktion der Wippe**

UCPTdiConfig[x].Byte[1] konfiguriert **Funktion der linken Taste**

UCPTdiConfig[x].Byte[2] konfiguriert **Funktion der rechten Taste**

UCPTdiConfig[x].Byte[3] konfiguriert **Funktion der LED**

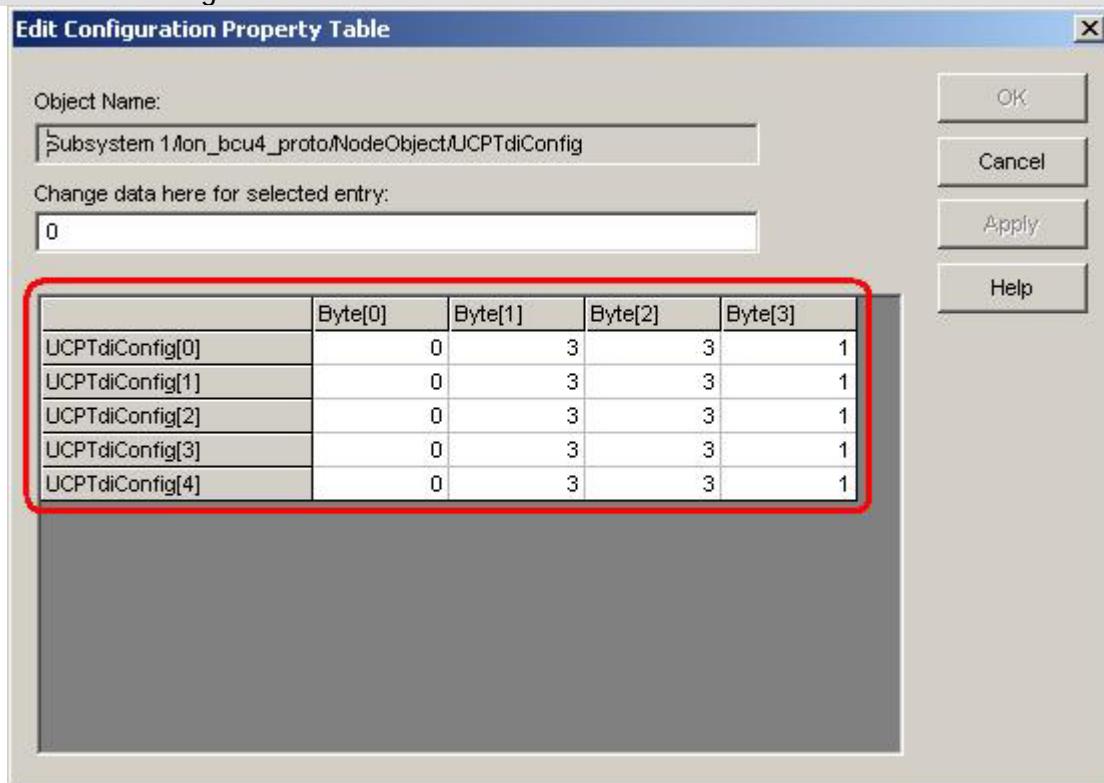


Abbildung 4-1: UCPTdiconfig

## UCPTsunblindUP

UCPT Index: 72, SNVT\_setting

Funktion: Mit diesem Konfigurationsparameter kann eingestellt werden, welcher SNVT\_setting Wert beim Hochfahren der Jalousie / Rollladen gesendet werden soll.  
(Voreingestellter Wert: SET\_UP 100.0 0.0)

## UCPTsunblindDOWN

UCPT Index: 73, SNVT\_setting

Funktion: Mit diesem Konfigurationsparameter kann eingestellt werden, welcher SNVT\_setting Wert beim Runterfahren der Jalousie / Rollladen gesendet werden soll.  
(Voreingestellter Wert: SET\_DOWN 100.0 0.0)

## UCPTsunblindSTOP

UCPT Index: 74, SNVT\_setting

Funktion: Mit diesem Konfigurationsparameter kann eingestellt werden, welcher SNVT\_setting Wert beim Anhalten der Jalousie / Rollladen gesendet werden soll.  
(Voreingestellter Wert: SET\_STOP 0.0 0.0)

## UCPTsunblindIDLE

UCPT Index: 75, SNVT\_setting

Funktion: Mit diesem Konfigurationsparameter kann eingestellt werden, welcher SNVT\_setting Wert für den Ruhezustand der Jalousie / Rollladen gesendet werden soll. *UCPTsunblindIDLE* wird 500ms nach anhalten der Jalousie / Rollladen gesendet, wenn *UCPTsunblindIDLE* ungleich *UCPTsunblindSTOP* ist.  
(Voreingestellter Wert: SET\_NUL 0.0 0.0)

6326-xx Triton Raumtemperaturregler 3-fach

Ein Wippe ist fest zugeordnet zu einem der Switch Objekte zugeordnet.

Funktion Beleuchtung Schalten / Tasten: Byte[0] = 0:

Konfiguration der Taster für Beleuchtung mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
0	Schalten / Tasten	1	Licht nur EIN (Switch_A_n)	1	Licht nur EIN (Switch_A_n)	1	Linke Wippe rot Rechte Wippe grün
		2	Licht nur AUS (Switch_A_n)	2	Licht nur AUS (Switch_A_n)	2	Linke Wippe grün Rechte Wippe rot
		3	Befehl Automatik (= 0.0 –1) (Switch_A_n)	3	Befehl Automatik (= 0.0 –1) (Switch_A_n)	3	Linke Wippe rot Rechte Wippe rot
						4	Linke Wippe grün Rechte Wippe grün

Beispiele:

- A: Die Wippe 1 soll als Schaltfunktion konfiguriert werden. Dabei schaltet die linke Taste das Licht Ein und die rechte Aus.
- UCPTdiConfig[0].Byte[0] = 0 → schalten
  - UCPTdiConfig[0].Byte[1] = 2 → Links EIN
  - UCPTdiConfig[0].Byte[2] = 3 → Rechts AUS
  - UCPTdiConfig[0].Byte[3] = 1 → LED



6327-xx Triton Raumtemperaturregler 5-fach

Ein Wippe ist fest zugeordnet zu einem der Switch Objekte. Jedes Objekt besitzt zwei Ausgangsvariablen (xxx\_A; xxx\_B). Jeder Taste kann somit ein einzelnes Switch-Objekt zugewiesen werden. Dies ist bei einer toggle Funktion sinnvoll. (siehe Kapitel4.2)

Funktion Beleuchtung Schalten / Tasten: Byte[0] = 0:

Konfiguration der Wippe für Beleuchtung EIN / AUS mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion LED	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
0	Schalten / Tasten	1	Licht Toggle EIN / AUS (Switch_A_n)	1	Licht Toggle EIN / AUS (Switch_B_n)	1	Linke Wippe rot Rechte Wippe grün
		2	Licht nur EIN (Switch_A_n)	2	Licht nur EIN (Switch_A_n)	2	Linke Wippe grün Rechte Wippe rot
		3	Licht nur EIN (Switch_B_n)	3	Licht nur EIN (Switch_B_n)	3	Linke Wippe rot Rechte Wippe rot
		4	Licht nur AUS (Switch_A_n)	4	Licht nur AUS (Switch_A_n)	4	Linke Wippe grün Rechte Wippe grün
		5	Licht nur AUS (Switch_B_n)	5	Licht nur AUS (Switch_B_n)		
		6	Befehl Automatik (= 0.0 –1) (Switch_A_n)	6	Befehl Automatik (= 0.0 –1) (Switch_A_n)		
		7	Befehl Automatik (= 0.0 –1) (Switch_B_n)	7	Befehl Automatik (= 0.0 –1) (Switch_B_n)		

Beispiele:

- A: Die Wippe 1 soll als Schaltfunktion konfiguriert werden. Dabei schaltet die linke Taste das Licht Ein und die rechte Aus.  
 UCPTdiConfig[0].Byte[0] = 0 → schalten  
 UCPTdiConfig[0].Byte[1] = 2 → Links EIN  
 UCPTdiConfig[0].Byte[2] = 4 → Rechts AUS  
 UCPTdiConfig[0].Byte[3] = 1 → LED
- B: Die Wippe 2 soll als Schaltfunktion konfiguriert werden. Dabei toggelt die linke und die rechte Taste das Licht Ein und Aus. Die LED soll grün leuchten.  
 UCPTdiConfig[1].Byte[0] = 0 → schalten  
 UCPTdiConfig[1].Byte[1] = 1 → Links toggeln  
 UCPTdiConfig[1].Byte[2] = 1 → Rechts toggeln  
 UCPTdiConfig[1].Byte[3] = 4 → LED
- C: Die Wippe 3 soll als Schaltfunktion konfiguriert werden. Dabei toggelt die linke Taste das Licht Ein und Aus. Die rechte Taste setzt die Beleuchtung in den Automatik-Modus. Die LED soll im toggle-Modus grün und im Automatik-Modus rot leuchten.  
 UCPTdiConfig[2].Byte[0] = 0 → schalten  
 UCPTdiConfig[2].Byte[1] = 1 → Links toggeln  
 UCPTdiConfig[2].Byte[2] = 6 → Rechts Automatik  
 UCPTdiConfig[2].Byte[3] = 2 → LED



Funktion Beleuchtung Dimmen: Byte[0] = 1:

6326-xx Triton Raumtemperaturregler 3-fach

- Kurze Tastendrucke bewirken ein toggeln des Schaltzustandes

Konfiguration der Wippe für Beleuchtung mit Dimmen mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion LED	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
1	Beleuchtung mit Dimmen	1	Licht heller mit Dimmen (Switch_A_n)	1	Licht heller mit Dimmen (Switch_A_n)	1	Licht Ein rot Licht Aus grün
			Einschaltwert = Max-Wert		Einschaltwert = Max-Wert		
		2	Licht heller mit Dimmen (Switch_A_n)	2	Licht heller mit Dimmen (Switch_A_n)	2	Licht Ein grün Licht Aus rot
			Einschaltwert = letzter Ein-Wert		Einschaltwert = letzter Ein-Wert		
		3	Licht dunkler mit Dimmen (Switch_A_n)	3	Licht nur dunkler mit Dimmen (Switch_A_n)	3	Linke Wippe rot Rechte Wippe rot
						4	Linke Wippe grün Rechte Wippe grün

Beispiele:

A:

Die Wippe 1 soll als Dimmfunktion konfiguriert werden. Die linke Taste dimmt das Licht Heller und die rechte Dunkler.

UCPTdiConfig[0].Byte[0] = 1 → Dimmen

UCPTdiConfig[0].Byte[1] = 1 → Links Heller

UCPTdiConfig[0].Byte[2] = 3 → Rechts Dunkler

UCPTdiConfig[0].Byte[3] = 1 → LED

Konfiguration der Wippe für Beleuchtung mit Dimmen mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion LED	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
1	Beleuchtung mit Dimmen	1	Licht Toggle mit Dimmen (Switch_A_n)	1	Licht Toggle mit Dimmen (Switch_B_n)	1	Linke Wippe rot Rechte Wippe grün
			Einschaltwert = Max-Wert		Einschaltwert = Max-Wert		
		2	Licht Toggle mit Dimmen (Switch_A_n)	2	Licht Toggle mit Dimmen (Switch_B_n)	2	Linke Wippe grün Rechte Wippe rot
			Einschaltwert = letzter Ein-Wert		Einschaltwert = letzter Ein-Wert		
		3	Licht nur Heller mit Dimmen (Switch_A_n)	3	Licht nur Heller mit Dimmen (Switch_A_n)	3	Linke Wippe rot Rechte Wippe rot
			Einschaltwert = Max-Wert		Einschaltwert = Max-Wert		
		4	Licht nur Heller mit Dimmen (Switch_B_n)	4	Licht nur Heller mit Dimmen (Switch_B_n)	4	Linke Wippe grün Rechte Wippe grün
			Einschaltwert = Max-Wert		Einschaltwert = Max-Wert		
		5	Licht nur Heller mit Dimmen (Switch_A_n)	5	Licht nur Heller mit Dimmen (Switch_A_n)		
			Einschaltwert = letzter Ein-Wert		Einschaltwert = letzter Ein-Wert		
		6	Licht nur Heller mit Dimmen (Switch_B_n)	6	Licht nur Heller mit Dimmen (Switch_B_n)		
			Einschaltwert = letzter Ein-Wert		Einschaltwert = letzter Ein-Wert		
		7	Licht nur dunkler mit Dimmen (Switch_A_n)	7	Licht nur dunkler mit Dimmen (Switch_A_n)		
		8	Licht nur dunkler mit Dimmen (Switch_B_n)	8	Licht nur dunkler mit Dimmen (Switch_B_n)		
		9	Befehl Automatik (= 0.0 – 1) (Switch_A_n)	9	Befehl Automatik (= 0.0 – 1) (Switch_A_n)		
		A	Befehl Automatik (= 0.0 – 1) (Switch_B_n)	A	Befehl Automatik (= 0.0 – 1) (Switch_B_n)		

Beispiele:

A: Die Wippe 1 soll als Dimmfunktion konfiguriert werden. Dabei schaltet die linke Taste das Licht Ein / Heller und die rechte Aus / Dunkler.

UCPTdiConfig[0].Byte[0] = 1 → Dimmen  
 UCPTdiConfig[0].Byte[1] = 3 → Links Ein / Heller  
 UCPTdiConfig[0].Byte[2] = 7 → Rechts Aus / Dunkler  
 UCPTdiConfig[0].Byte[3] = 1 → LED

B: Die Wippe 2 soll als Schaltfunktion konfiguriert werden. Dabei toggelt die linke Taste mit *nvoSwitch\_A\_n* und die rechte Taste mit *nvoSwitch\_B\_n* das Licht Ein bzw. Aus. Die LED soll grün leuchten.

UCPTdiConfig[1].Byte[0] = 1 → Dimmen  
 UCPTdiConfig[1].Byte[1] = 1 → Links Toggle mit Dimmen  
 UCPTdiConfig[1].Byte[2] = 1 → Rechts Toggle mit Dimmen  
 UCPTdiConfig[1].Byte[3] = 4 → LED

Funktion Jalousie: Byte[0] = 2:

Konfiguration der Taster für Jalousie mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion LED	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
2	Jalousie	1	Jalousie AUF (Setting_A_n)	1	Jalousie AUF (Setting_A_n)	1	Linke Wippe rot Rechte Wippe grün
		2	Jalousie AB (Setting_A_n)	2	Jalousie AB (Setting_A_n)	2	Linke Wippe grün Rechte Wippe rot
		3	Jalousie AUF (Setting_B_n)	3	Jalousie AUF (Setting_B_n)	3	Linke Wippe rot Rechte Wippe rot
		4	Jalousie AB (Setting_B_n)	4	Jalousie AB (Setting_B_n)	4	Linke Wippe grün Rechte Wippe grün

Beispiel: Die Wippe 1 soll als Jalousiefunktion konfiguriert werden. Dabei schaltet die linke Taste die Jalousie Auf die rechte Ab. Die LED soll rot leuchten.

UCPTdiConfig[0].Byte[0] = 2 → Jalousie  
 UCPTdiConfig[0].Byte[1] = 1 → Links Auf  
 UCPTdiConfig[0].Byte[2] = 2 → Rechts Ab  
 UCPTdiConfig[0].Byte[3] = 3 → LED

Funktion Rollladen: Byte[0] = 3:

Konfiguration der Wippe für Jalousie mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion LED	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
3	Rollladen	1	Rollladen AUF (Setting_A_n)	1	Rollladen AUF (Setting_A_n)	1	Linke Wippe rot Rechte Wippe grün
		2	Rollladen AB (Setting_A_n)	2	Rollladen AB (Setting_A_n)	2	Linke Wippe grün Rechte Wippe rot
		3	Rollladen AUF (Setting_B_n)	3	Rollladen AUF (Setting_B_n)	3	Linke Wippe rot Rechte Wippe rot
		4	Rollladen AB (Setting_B_n)	4	Rollladen AB (Setting_B_n)	4	Linke Wippe grün Rechte Wippe grün

Beispiel: Die Wippe 3 soll als Rollladenfunktion konfiguriert werden. Dabei schaltet die linke Taste den Rollladen Auf die rechte Ab. Die LED soll rot sein.

UCPTdiConfig[2].Byte[0] = 3 → Rollladen  
 UCPTdiConfig[2].Byte[1] = 1 → Links Auf  
 UCPTdiConfig[2].Byte[2] = 2 → Rechts Ab  
 UCPTdiConfig[2].Byte[3] = 3 → LED

Funktion Szenensteuerung: Byte[0] = 4:

Konfiguration der Taster für Szenensteuerung mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
4	Szene	01-FE	Szenen-Nummer	01-FE	Szenen-Nummer	1	Linke Wippe rot Rechte Wippe grün
						2	Linke Wippe grün Rechte Wippe rot
						3	Linke Wippe rot Rechte Wippe rot
						4	Linke Wippe grün Rechte Wippe grün

Beispiel: Die Wippe 2 soll als Szenenaufruf konfiguriert werden. Dabei ruft die linke Taste die Szene 5 auf und die rechte Taste die Szene 12. Die LED soll rot leuchten.

UCPTdiConfig[1].Byte[0] = 4 → Szene  
 UCPTdiConfig[1].Byte[1] = 5 dez → Szene 5 aufrufen  
 UCPTdiConfig[1].Byte[2] = 12 dez → Szene 12 aufrufen  
 UCPTdiConfig[1].Byte[3] = 3 → LED

Funktion Lüfterstufe: Byte[0] = 5:

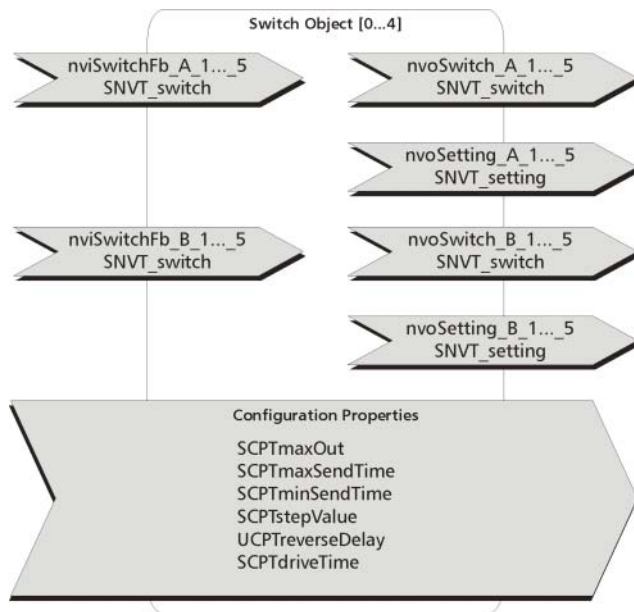
Die Lüfterstufenstellung wird stetig in 5 % - Schritten verändert. Ein Automatikbetrieb ist nicht möglich.

Konfiguration der Wippe für stetige Lüfterstufensteuerung mit UCPTdiConfig							
Zuordnung Wippen - Funktionsblock		Funktion linke Taste		Funktion rechte Taste		Funktion LED	
Byte[0]	Beschreibung	Byte[1]	Beschreibung	Byte[2]	Beschreibung	Byte[3]	Beschreibung
5	Lüfterstufe	1	Lüfterstufe -5 % (nvoFanSpeed)	1	Lüfterstufe +5 % (nvoFanSpeed)	1	Linke Wippe rot Rechte Wippe grün
						2	Linke Wippe grün Rechte Wippe rot
						3	Linke Wippe rot Rechte Wippe rot
						4	Linke Wippe grün Rechte Wippe grün

Beispiel: Die Wippe 2 soll für eine Lüfterstufenverstellung konfiguriert werden. Die LED soll rot leuchten.

UCPTdiConfig[1].Byte[0] = 5 → Lüfterverstellung  
 UCPTdiConfig[1].Byte[1] = 1 → Links + 5 % Schritte  
 UCPTdiConfig[1].Byte[2] = 1 → Rechts - 5 % Schritte  
 UCPTdiConfig[1].Byte[3] = 3 → LED

## 4.2 Switch Object



Fünf identische Switch-Objekte für die Beleuchtungs- und Jalousiesteuerung. Die Switch-Objekte werden direkt den Wippen zugeordnet. Dabei gehört zu den einzelnen Objekten:

- Switch Object[0] = Wippe 1
- Switch Object[1] = Wippe 2
- Switch Object[2] = Wippe 3
- Switch Object[3] = Wippe 4
- Switch Object[4] = Wippe 5

Die Zuordnung der Wippen zu den einzelnen Funktionen erfolgt wie zuvor beschrieben mit den Konfigurationsparametern UCPTdiConfig[0]...[4] im Node Object. Mögliche Funktionen sind Beleuchtungssteuerung EIN/AUS, Beleuchtungssteuerung EIN/AUS mit Dimmen oder Jalousie AUF/AB mit Automatiklauf. Die Lüfterstufenregelung befindet sich im Space Comfort Object.

Jedes Switch Object hat 2 Variablengruppen die durch den Anhang *xxx\_A* und *xxx\_B* unterschieden werden. Dadurch kann jedem Taster eine Netzwerkvariable zugeordnet werden.

### 4.2.1 Eingangsvariablen Switch Object:

**nviSwitchFb\_A\_1...5, nviSwitchFb\_B\_1...5**

SNVT Typ: SNVT\_switch, Index 95

Funktion: Eingangsvariablen für den aktuellen Zustand der mit nvoSwitch\_A\_1...5 bzw, nvoSwitchFb\_B\_1...5 angesteuerten Beleuchtungsgruppen.

### 4.2.2 Ausgangsvariablen Switch Object:

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

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

Funktion: Ausgangsvariablen zur Ansteuerung von Beleuchtungsgruppen und von Beleuchtungs- und Jalousiecontrollern. Die Ausgabewerte sind von den Funktionseinstellungen abhängig. Jeder Wippe kann ein Switch Object zugewiesen werden. Jedes Switch Object hat zwei verschiedene Variablengruppen, welche sich durch den Anhang *\_A* bzw. *\_B* unterscheiden. Dadurch kann jedem Taster eine Netzwerkvariable zugeordnet werden.

Mit

UCPTdiConfig[x].Byte[0] werden Wippen einer Funktion zugeordnet.

UCPTdiConfig[x].Byte[1] weist der linken Taste eine Funktion zu.

UCPTdiConfig[x].Byte[2] weist der rechten Taste eine Funktion zu.

UCPTdiConfig[x].Byte[3] weist der LED die Funktion zu.

## Schalten/ Tasten

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

### Taste gedrückt/ nicht gedrückt

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

Taste gedrückt	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;
Taste nicht gedrückt	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>

Taste gedrückt	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;
Taste nicht gedrückt	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0
	nvoSetting_B_x.function	= SET_OFF;
	nvoSetting_B_x.setting	= 0;

## Beleuchtung Toggle

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

Jede Tastbetätigung führt zum Umschalten der Beleuchtung, d.h. zwischen EIN und AUS

Beleuchtung EIN	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;
Beleuchtung AUS	nvoSwitch_A_x.value	= 0
	nvoSwitch_A_x.state	= 0
	nvoSetting_A_x.function	= SET_OFF;
	nvoSetting_A_x.setting	= 0;

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

Jede Tastbetätigung führt zum Umschalten der Beleuchtung, d.h. zwischen EIN und AUS

Beleuchtung EIN	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;
Beleuchtung AUS	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0
	nvoSetting_B_x.function	= SET_OFF;
	nvoSetting_B_x.setting	= 0;

## Beleuchtung EIN

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

Jede Tastbetätigung führt zum Einschalten der Beleuchtung

Beleuchtung EIN	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>

Jede Tastbetätigung führt zum Einschalten der Beleuchtung

Beleuchtung EIN	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;

## Beleuchtung AUS

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

Jede Tastbetätigung führt zum Ausschalten der Beleuchtung

Beleuchtung AUS	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>

Jede Tastbetätigung führt zum Ausschalten der Beleuchtung

Beleuchtung AUS	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0
	nvoSetting_B_x.function	= SET_OFF;
	nvoSetting_B_x.setting	= 0;

## Automatik

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

Die Betätigung einer „Automatik-Taste“ schaltet die Variablen nvoSwitch\_A\_x auf den Wert 0,0 -1. Damit kann z.B. ein Beleuchtungsregler von ext. Übersteuerung wieder in den Automatikmodus gebracht werden.

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

Die Betätigung einer „Automatik-Taste“ schaltet die Variablen nvoSwitch\_B\_x auf den Wert 0,0 -1. Damit kann z.B. ein Beleuchtungsregler von ext. Übersteuerung wieder in den Automatikmodus gebracht werden.

## Dimmen

### 6327-xx Triton Raumtemperaturregler 5-fach

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

## Beleuchtung Togglen mit Dimmen, Einschaltwert = max. Wert

Kurze Tastbetätigungen (< 1 s) führen zum Umschalten des aktuellen Beleuchtungszustandes, wobei der .value - Einschaltwert immer SCPTmaxOut beträgt. Mit längeren Tastbetätigungen (> 1 s) wird die Dimm-Funktion aktiviert, d.h. ausgehend vom aktuellen Beleuchtungszustand wird der .value-Wert der Switch-Variablen in Prozentschritten von UCPTstepValue erhöht oder verringert und zwar so lange wie die Taste gedrückt wird.

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

Beleuchtung auf Maximalwert	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
Beleuchtung auf 50%	nvoSwitch_A_x.value	= 50,0
	nvoSwitch_A_x.state	= 1
Beleuchtung AUS	nvoSwitch_A_x.value	= 0
	nvoSwitch_A_x.state	= 0

Die Variablen nvoSetting\_A\_x werden in dieser Einstellung nicht verändert.

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

Beleuchtung auf Maximalwert	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
Beleuchtung auf 50%	nvoSwitch_B_x.value	= 50,0
	nvoSwitch_B_x.state	= 1
Beleuchtung AUS	nvoSwitch_B_x.value	= 0
	nvoSwitch_B_x.state	= 0



## Softwarebeschreibung

Die Variablen nvoSetting\_B\_x werden in dieser Einstellung nicht verändert.

### Beleuchtung Togglen mit Dimmen, Einschaltwert = letzter Ein-Wert

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

Funktion wie bei 01<sub>hex</sub>, nur mit dem Unterschied, dass beim Einschalten die Beleuchtung nicht mit dem Wert SCPTmaxOut angesteuert wird, sondern mit dem letzten Einschaltwert. Der kleinste Einschaltwert ist hierbei auf 20% begrenzt.

### Beleuchtung EIN mit Heller-Dimmen, Einschaltwert = max. Wert

Ist die Beleuchtung ausgeschaltet, dann führt eine Tastbetätigung zum sofortigen Einschalten der Beleuchtung. Mit längeren Tastbetätigungen (> 1 s) wird die Funktion „Heller-Dimmen“ aktiviert, d.h. ausgehend vom aktuellen Beleuchtungszustand wird der .value - Wert der Switch-Variablen in Prozent - Schritten von UCPTstepValue erhöht und zwar so lange bis der Maximalwert SCPTmaxOut erreicht wird. Das Sendeintervall im Modus Dimmen wird mit SCPTminSendTime eingestellt und beträgt voreingestellt ca. 300ms.

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

Beleuchtung Einschalten	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;
Beleuchtung Heller-Dimmen	nvoSwitch_A_x.value	= letzter Wert + 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>

Beleuchtung Einschalten	nvoSwitch_B_x.value	= SCPTmaxOut
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_ON;
	nvoSetting_B_x.setting	= SCPTmaxOut;
Beleuchtung Heller-Dimmen	nvoSwitch_B_x.value	= letzter Wert + UCPTstepValue
	nvoSwitch_B_x.state	= 1
	nvoSetting_B_x.function	= SET_UP;
	nvoSetting_B_x.setting	= UCPTstepValue;

### Beleuchtung EIN mit Heller-Dimmen, Einschaltwert = letzter Ein-Wert

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

Funktion wie bei 03<sub>hex</sub>, 04<sub>hex</sub> nur mit dem Unterschied, dass beim Einschalten die Beleuchtung nicht mit dem Wert SCPTmaxOut angesteuert wird, sondern mit dem letzten Einschaltwert. Der kleinste Einschaltwert ist hierbei auf 20% begrenzt.

### Beleuchtung AUS mit Dunkler-Dimmen

Ist die Beleuchtung eingeschaltet, dann führt eine kurze Tastbetätigung (< 1 s) zum sofortigen Ausschalten der Beleuchtung. Mit längeren Tastbetätigungen (> 1 s) wird die Funktion „Dunkler-Dimmen“ aktiviert, d.h. ausgehend vom aktuellen Beleuchtungszustand wird der .value - Wert der Switch-Variablen in Prozent - Schritten von UCPTstepValue verringert und zwar so lange bis der Wert 0 erreicht wird. Das Sendeintervall im Modus Dimmen wird mit SCPTminSendTime eingestellt und beträgt voreingestellt ca. 300ms.

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

Beleuchtung Ausschalten	nvoSwitch_A_x.value	= 0
	nvoSwitch_A_x.state	= 0
	nvoSetting_A_x.function	= SET_OFF;
	nvoSetting_A_x.setting	= 0;
Beleuchtung Dunkler-Dimmen	nvoSwitch_A_x.value	= letzter Wert - 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>**

Beleuchtung Ausschalten

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

Beleuchtung Dunkler-Dimmen

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

## Automatik

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

Die Betätigung einer „Automatik-Taste“ schaltet die Variable nvoSwitch\_A\_x auf den Wert 0,0 -1. Damit kann z.B. ein Beleuchtungsregler von ext. Übersteuerung wieder in den Automatikmodus gebracht werden.

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

Die Betätigung einer „Automatik-Taste“ schaltet die Variable nvoSwitch\_B\_x auf den Wert 0,0 -1. Damit kann z.B. ein Beleuchtungsregler von ext. Übersteuerung wieder in den Automatikmodus gebracht werden.

## 6326-xx Triton Raumtemperaturregler 3-fach

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

### Beleuchtung Heller-Dimmen, Einschaltwert = max. Wert

Kurze Tastbetätigungen (< 1 s) führen zum Umschalten des aktuellen Beleuchtungszustandes, wobei der .value - Einschaltwert immer SCPTmaxOut beträgt. Mit längeren Tastbetätigungen (> 1 s) wird die Funktion „Heller-Dimmen“ aktiviert, d.h. ausgehend vom aktuellen Beleuchtungszustand wird der .value - Wert der Switch-Variablen in Prozent - Schritten von UCPTstepValue erhöht und zwar so lange bis der Maximalwert SCPTmaxOut erreicht wird. Das Sendeintervall im Modus Dimmen wird mit SCPTminSendTime eingestellt und beträgt voreingestellt ca. 300ms.

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

Beleuchtung Einschalten

nvoSwitch\_A\_x.value = SCPTmaxOut  
nvoSwitch\_A\_x.state = 1  
nvoSetting\_A\_x.function = SET\_ON;  
nvoSetting\_A\_x.setting = SCPTmaxOut;

Beleuchtung AUS

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

Beleuchtung Heller-Dimmen

nvoSwitch\_A\_x.value = letzter Wert + UCPTstepValue  
nvoSwitch\_A\_x.state = 1  
nvoSetting\_A\_x.function = SET\_UP;  
nvoSetting\_A\_x.setting = UCPTstepValue;

### Beleuchtung Heller-Dimmen, Einschaltwert = letzter Ein-Wert

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

Funktion wie bei 01hex nur mit dem Unterschied, dass beim Einschalten die Beleuchtung nicht mit dem Wert SCPTmaxOut angesteuert wird, sondern mit dem letzten Einschaltwert. Der kleinste Einschaltwert ist hierbei auf 20% begrenzt.

## Beleuchtung Dunkler-Dimmen, Einschaltwert = max. Wert

Kurze Tastbetätigungen (< 1 s) führen zum Umschalten des aktuellen Beleuchtungszustandes, wobei der .value - Einschaltwert immer SCPTmaxOut beträgt. Mit längeren Tastbetätigungen (> 1 s) wird die Funktion „Dunkler-Dimmen“ aktiviert, d.h. ausgehend vom aktuellen Beleuchtungszustand wird der .value - Wert der Switch-Variablen in Prozent - Schritten von UCPTstepValue verringert und zwar so lange bis der Wert 0 erreicht wird. Das Sendeintervall im Modus Dimmen wird mit SCPTminSendTime eingestellt und beträgt voreingestellt ca. 300ms.

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

Beleuchtung Einschalten	nvoSwitch_A_x.value	= SCPTmaxOut
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_ON;
	nvoSetting_A_x.setting	= SCPTmaxOut;
Beleuchtung Ausschalten	nvoSwitch_A_x.value	= 0
	nvoSwitch_A_x.state	= 0
	nvoSetting_A_x.function	= SET_OFF;
	nvoSetting_A_x.setting	= 0;
Beleuchtung Dunkler-Dimmen	nvoSwitch_A_x.value	= letzter Wert - UCPTstepValue
	nvoSwitch_A_x.state	= 1
	nvoSetting_A_x.function	= SET_DOWN;
	nvoSetting_A_x.setting	= UCPTstepValue;

## Beleuchtung Dunkler-Dimmen, Einschaltwert = letzter Ein-Wert

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

Funktion wie bei 03<sub>hex</sub> nur mit dem Unterschied, dass beim Einschalten die Beleuchtung nicht mit dem Wert SCPTmaxOut angesteuert wird, sondern mit dem letzten Einschaltwert. Der kleinste Einschaltwert ist hierbei auf 20% begrenzt.

## Jalousie

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

#### Jalousie AUF

Im Konfigurationsmodus „Jalousie AUF“ werden nur die nvoSetting-Variablen verändert und gesendet. Kurze Tastbetätigungen (< 2 s) dienen zur Feineinstellung der Lamellen. Eine lange Tastbetätigung (> 2 s) startet den Automatiklauf und steuert die Jalousie für die Zeit SCPTdriveTime dauerhaft in Richtung öffnen an. Der Automatiklauf kann durch erneute Tastbetätigung angehalten werden.

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

Jalousie öffnen	nvoSetting_A_x.function	= UCPTsunblindUP;
Jalousie anhalten	nvoSetting_A_x.function	= UCPTsunblindSTOP;

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

Jalousie öffnen	nvoSetting_B_x.function	= UCPTsunblindUP;
Jalousie anhalten	nvoSetting_B_x.function	= UCPTsunblindSTOP;

Mit einer Verzögerung von 500ms kann nach senden des Befehls UCPTsunblindSTOP der Befehl UCPTsunblindIDLE gesendet, wenn dieser ungleich UCPTsunblindSTOP ist.

#### Jalousie AB

Im Konfigurationsmodus „Jalousie AB“ werden nur die nvoSetting-Variablen verändert und gesendet. Kurze Tastbetätigungen (< 2 s) dienen zur Feineinstellung der Lamellen. Eine lange Tastbetätigung (> 2 s) startet den Automatiklauf und steuert die Jalousie für die Zeit SCPTdriveTime dauerhaft in Richtung schließen an. Der Automatiklauf kann durch erneute Tastbetätigung angehalten werden.

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

## Softwarebeschreibung

Jalousie schließen	nvoSetting_A_x.function	= UCPTsunblindDOWN;
Jalousie anhalten	nvoSetting_A_x.function	= UCPTsunblindSTOP;

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

Jalousie schließen	nvoSetting_B_x.function	= UCPTsunblindDOWN;
Jalousie anhalten	nvoSetting_B_x.function	= UCPTsunblindSTOP;

Mit einer Verzögerung von 500ms kann nach senden des Befehls UCPTsunblindSTOP der Befehl UCPTsunblindIDLE gesendet, wenn dieser ungleich UCPTsunblindSTOP ist.

## Rollladen

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

### Rollladen AUF

Im Konfigurationsmodus „Rollladen AUF“ werden nur die nvoSetting-Variablen verändert und gesendet. Kurze Tastbetätigungen (< 2 s) startet den Automatiklauf und steuert den Rollladen für die Zeit SCPTdriveTime dauerhaft in Richtung öffnen an. Der Automatiklauf kann durch erneute Tastbetätigung angehalten werden. Mit einer langen Tastbetätigung (> 2 s) kann die Position des Rollladens individuell eingestellt werden.

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

Rollladen öffnen	nvoSetting_A_x.function	= UCPTsunblindUP;
Rollladen anhalten	nvoSetting_A_x.function	= UCPTsunblindSTOP;

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

Rollladen öffnen	nvoSetting_B_x.function	= UCPTsunblindUP;
Rollladen anhalten	nvoSetting_B_x.function	= UCPTsunblindSTOP;

Mit einer Verzögerung von 500ms kann nach senden des Befehls UCPTsunblindSTOP der Befehl UCPTsunblindIDLE gesendet, wenn dieser ungleich UCPTsunblindSTOP ist.

### Rollladen AB

Im Konfigurationsmodus „Rollladen AB“ werden nur die nvoSetting-Variablen verändert und gesendet. Kurze Tastbetätigungen (< 2 s) startet den Automatiklauf und steuert den Rollladen für die Zeit SCPTdriveTime dauerhaft in Richtung schließen an. Der Automatiklauf kann durch erneute Tastbetätigung angehalten werden. Mit einer langen Tastbetätigung (> 2 s) kann die Position des Rollladens individuell eingestellt werden

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

Rollladen schließen	nvoSetting_A_x.function	= UCPTsunblindDOWN;
Rollladen anhalten	nvoSetting_A_x.function	= UCPTsunblindSTOP;

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

Rollladen schließen	nvoSetting_B_x.function	= UCPTsunblindDOWN;
Rollladen anhalten	nvoSetting_B_x.function	= UCPTsunblindSTOP;

Mit einer Verzögerung von 500ms kann nach senden des Befehls UCPTsunblindSTOP der Befehl UCPTsunblindIDLE gesendet, wenn dieser ungleich UCPTsunblindSTOP ist.

## 4.2.3 Konfigurationsparameter Switch Object:

### SCPTmaxOut

SCPT Index: 93, SNVT\_lev\_cont

Funktion: Dieser Konfigurationsparameter bestimmt den maximalen Ausgabewert der Variablen nvoSwitch.value. (Voreingestellter Wert: 100.0)

### SCPTmaxSendTime

SCPT Index: 49, SNVT\_time\_sec

Funktion: Heartbeatfunktion. Dieser Konfigurationsparameter legt die Intervallzeit fest, nach der die Ausgangsvariablen gesendet werden. Mit Eingabewerten = 0 wird die Heartbeatfunktion deaktiviert. (Voreingestellter Wert: 0,0 s)

### SCPTminSendTime

SCPT Index: 52, SNVT\_time\_sec

Funktion: Dieser Konfigurationsparameter legt das Sendeintervall der Ausgangsvariablen im Modus Dimmen fest. Mit Eingabewerten = 0 wird die Funktion deaktiviert. (Voreingestellter Wert: 0,3 s)

### SCPTstepValue

SCPT Index: 92, SNVT\_lev\_cont

Funktion: Dieser Konfigurationsparameter definiert die Schrittweite der Variablen nvoSwitch.value im Modus Dimmen. (Voreingestellter Wert: 5.0)

### UCPTreverseDelay

UCPT Index: 14, SNVT\_count

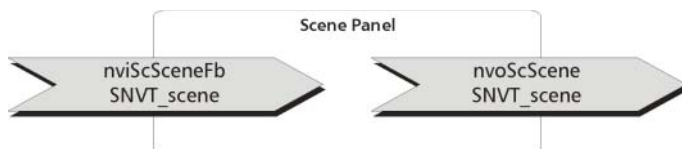
Funktion: Dieser Konfigurationsparameter definiert die Umschaltverzögerung bei Drehrichtungs-umkehr der Jalousiemotoren in ms. Damit wird ein Wechselbefehl z.B. von nvoSetting = SET\_UP auf nvoSetting = SET\_DOWN verzögert ausgegeben. (Voreingestellter Wert: 500 ms)

### SCPTdriveTime

UCPT Index: 45, SNVT\_time\_sec

Funktion: Dieser Konfigurationsparameter definiert die maximale Einschaltzeit der Jalousiemotoren im Automatiklauf. (Voreingestellter Wert: 100,0 s)

## 4.3 Scene Panel



### 4.3.1 Eingangsvariablen Scene Panel

#### nviScSceneFb

SNVT Typ: SNVT\_scene, Index 115

Funktion: Eingangsvariable mit der aktuellen Beleuchtungsszene im Raum.

### 4.3.2 Ausgangsvariablen Scene Panel:

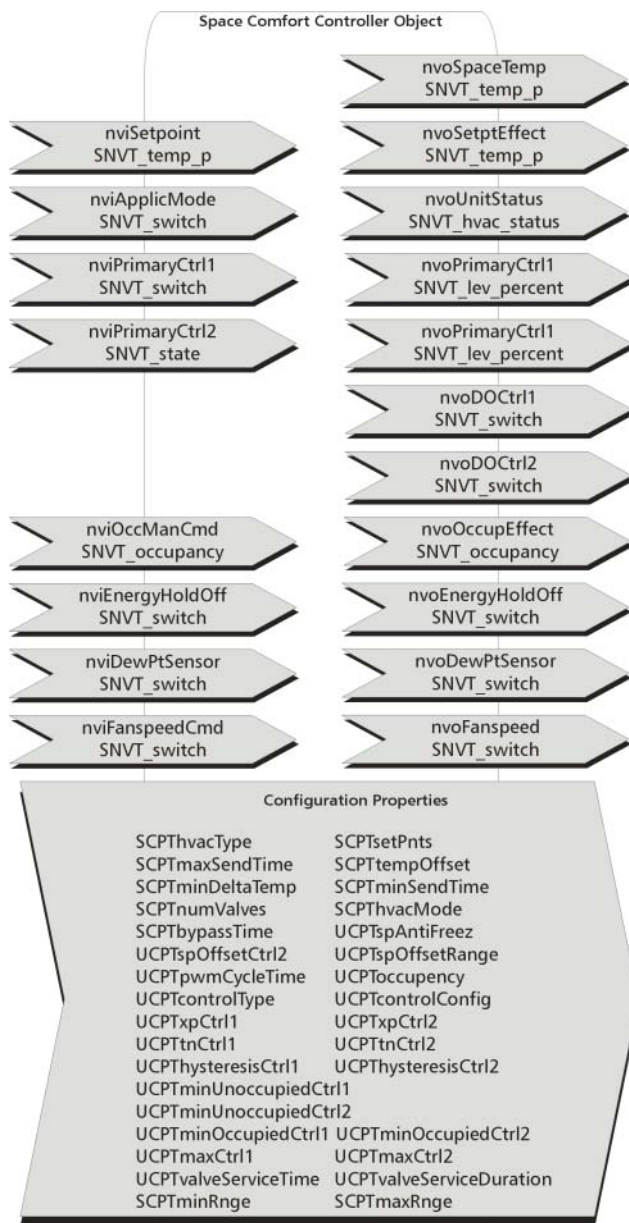
#### nvoScScene

SNVT Typ: SNVT\_scene, Index 115

Funktion: Ausgangsvariable zur Ansteuerung eines Szene-Controllers. Die Ausgabewerte sind von den Funktionseinstellungen abhängig. Mit UCPTdiConfig[x].Byte[0] = 3 werden Taster dem Scene-Object zugeordnet.

UCPTdiConfig[x].Byte[1...2] = 01<sub>hex</sub> - FE<sub>hex</sub> weist diesen Tastern dann eine Szenennummer zu, die bei kurzen Tastbetätigungen mit SC\_RECALL aufgerufen wird. Bei langen Tastbetätigungen (> 2s) wird die Szene mit SC\_LEARN neu gelernt. Es können auch mehrere Tasten dem Scene Panel zugeordnet werden.

## 4.4 Space Comfort Controller Object



### Temperatur

Die Temperaturerfassung erfolgt mit dem internen Temperaturfühler. Eine externe Temperaturmessung ist nicht möglich.

Für eine nachträgliche Kalibrierung des internen Sensors steht der Konfigurationsparameter *SCPTtempOffset* zur Verfügung.

### Sollwert

Der effektive Sollwert *nvoSetptEffect* errechnet sich in Abhängigkeit zur Raumbelastung (*nvoOccupEffect*), aus den Sollwertvorgaben über *SCPTsetPnts* bzw. *nviSetpoint* und dem Offsetwert (manuelle Verstellung).

### Regelung

Für die Temperaturregelung verwendet der Regelalgorithmus den oben beschriebenen effektiven Sollwert. Die neutrale (energiefreie) Zone um den Basissollwert passt sich automatisch der aktuellen Raumbelastung an und ist ebenfalls über *SCPTsetPnts* parametrierbar.

Der Regler kann mit *UCPTcontrolType* für Heizen und Kühlen oder für ein zweistufiges Heizen konfiguriert werden. Die Stellgrößen des PI-Reglers für Heizen und Kühlen werden mit Variablen vom Typ *SNVT\_lev\_percent* für stetige Stellantriebe und mit Variablen vom Typ *SNVT\_switch* für thermische Zweipunkt-Stellantriebe (PWM-Ansteuerung und Zweipunkt-Regelung) ausgegeben. Die Regelparameter können hierbei den räumlichen Gegebenheiten individuell angepasst werden. Die Überwachung von Fensterkontakt und Taupunktwärter erfolgt mit den Eingangsvariablen *nviEnergyHoldOff* und *nviDewPtSensor*.

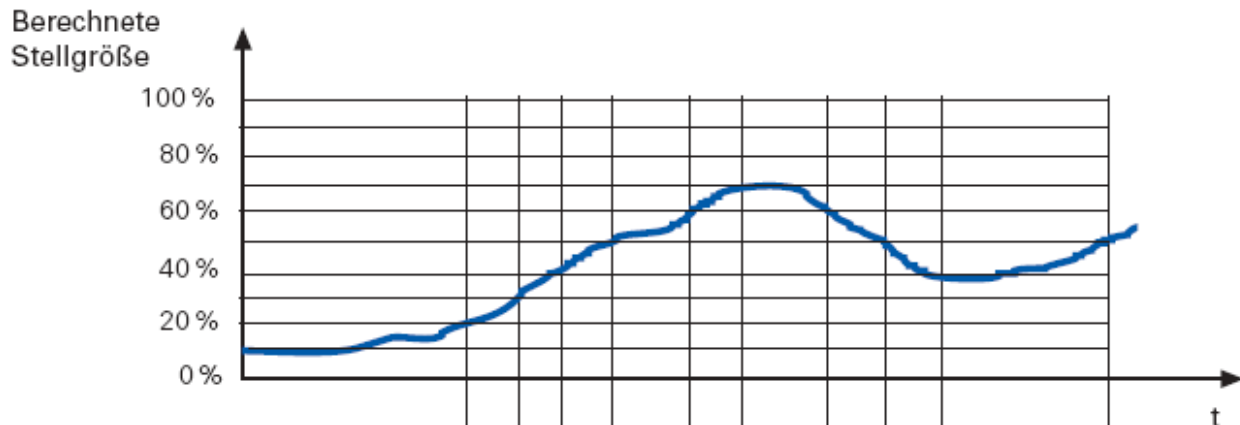
### Änderung von Konfigurationsparametern

Nachdem Parameter geändert worden sind, müssen die Änderungen aktiviert werden. Mit setzen von 0 auf 1 von *nviProgDevice.bit0* bzw. nach einem Spannungsreset werden diese neuen Einstellungen in das Bedienteil übernommen.



## Stetige PI-Regelung

Der Stetigregler gibt die Stellgröße auf den Bus aus. Die Raumtemperatur wird durch den Regelalgorithmus konstant gehalten. Die gesendete Stellgröße wirkt auf einen stetigen Stellantrieb, der auf einem Eckventil montiert ist. Dieser dosiert, gemäß der Stellgröße, die Wärmemenge durch den Heizkörper (0 bis 100%). Je mehr der Sollwert von der Temperatur abweicht, desto größer wird die Stellgröße.



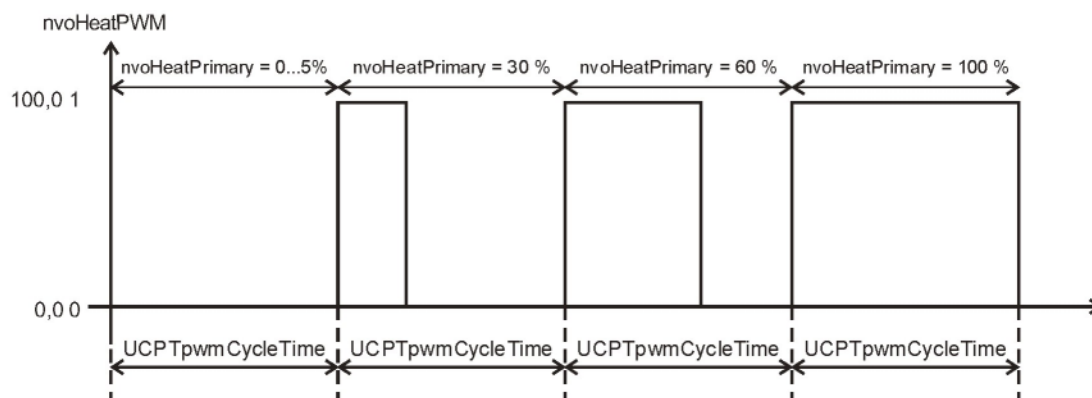
## Schaltende PI-Regelung (PWM-Regelung)

Bei der PWM-Regelung (Puls-Weiten-Modulation) werden die vom Regelalgorithmus berechneten Werte (0...100%) in eine PWM umgewandelt. Diese basiert immer auf einer konstanten Zykluszeit. Berechnet der Regler eine Stellgröße von 20%, dann wird bei einer „Zykluszeit der schaltenden Stellgröße“ von 15 Minuten eine logische „1“ für 3 Minuten (20% von 15 Minuten) und anschließend eine „0“ für 12 Minuten gesendet. Nach Ablauf der Zykluszeit wird die aktuelle Stellgröße des Reglers wieder in eine neue PWM umgewandelt.

Die Raumtemperatur wird auch durch den Regelalgorithmus konstant gehalten. Gemittelt über die Zeit, ergibt sich das gleiche Verhalten des Regelsystems wie mit einem stetigen Regler.

Bei der PWM- und der 2-Punkt-Regelung steuert der Raumtemperaturregler einen Schaltaktor an. Der Schaltaktor öffnet oder schließt den Stellantrieb.

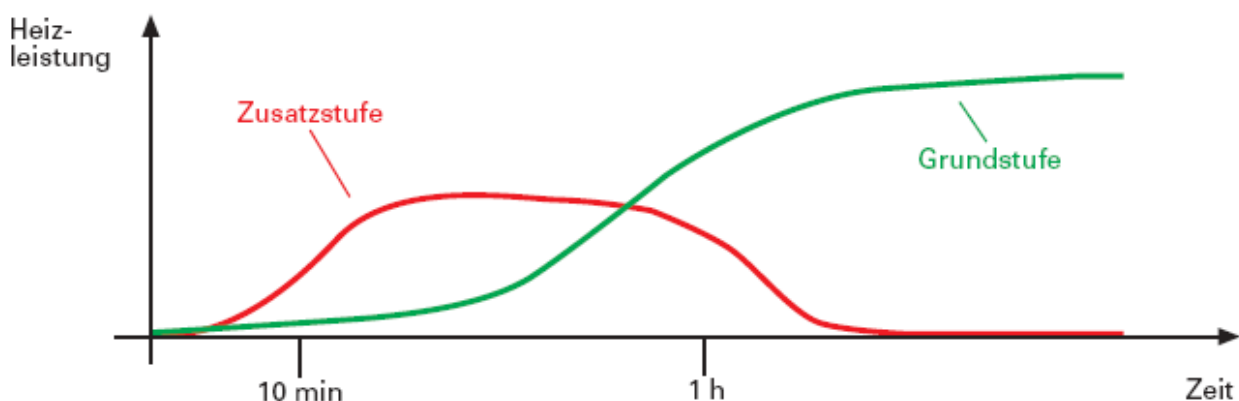
Beispiel PWM-Ansteuerung für Regler 1 (Ctrl1):





## 2-stufiges Heizen

Das 2-stufige Heizen kommt sehr häufig in Verbindung mit einer Fußbodenheizung zum Einsatz. Die Fußbodenheizung ist ein sehr träges System. Das Aufheizen eines Raums dauert dementsprechend lang (bis zu mehreren Stunden). Um die Aufheizphase zu verkürzen, wird zusätzlich ein flinkes Heizungssystem (z. B. Warmwasser-Konvektorheizung) eingesetzt. Sobald man die Solltemperatur wesentlich erhöht, wird die Zusatzstufe (flinkes Heizungssystem) zusammen mit der Grundstufe (z. B. Fußbodenheizung) eingeschaltet. Der Raum wird dann hauptsächlich über die Zusatzstufe aufgeheizt, da diese flinker ist und somit schneller Heizleistung zur Verfügung stellt. Ist der Raum bis zu einer gewissen Temperatur (z. B. Solltemperatur  $-1\text{ K}$ , parametrierbar) aufgeheizt, wird die Zusatzstufe abgeschaltet. In der Zwischenzeit kann die Grundstufe die benötigte Wärmeleistung dem Raum zuführen und übernimmt dann allein die Regelung. Die Grundstufe wird ganz normal parametrierbar (z. B. bei Fußbodenheizung: PWM mit 30 Minuten Zykluszeit). Für die Zusatzheizung ist eine 2-Punkt-Regelung absolut ausreichend, da diese nicht für die Regelung genutzt wird, sondern nur für die Aufheizphase.



Thermisches Verhalten, sobald Grund- und Zusatzstufe gleichzeitig eingeschaltet werden, z. B. wenn morgens die Heizung aktiviert wird

### 4.4.1 Eingangsvariablen Space Comfort Controller Object

#### nviApplicMode

SNVT Typ: SNVT\_hvac\_mode, Index 108

Funktion: Eingangsvariable zur Auswahl des Betriebsmodus des Reglers.

HVAC\_AUTO ==> automatisches Umschalten zwischen Heizen und Kühlen

HVAC\_HEAT ==> nur Heizen

HVAC\_COOL ==> nur Kühlen

HVAC\_OFF ==> Stellgrößen werden auf 0% gesetzt

Der Initialisierungszustand nach Reset wird durch den Konfigurationsparameter *SCPTHvacMode* bestimmt.

**!! Bei einer Änderung von HVAC\_AUTO auf HVAC\_HEAT bzw. HVAC\_COOL sowie von**

**!! HVAC\_HEAT bzw. HVAC\_COOL auf HVAC\_AUTO wird im Bedienteil ein Reset durchgeführt.**

#### nviPrimaryCtrl1

SNVT Typ: SNVT\_lev\_percent, Index 81

Funktion: Steuervariable für die Netzwerkvariablen *nvoPrimaryCtrl1* und *nvoDOCtrl1*.

*nviPrimaryCtrl1* = 0x7FFF (163,835 %) ==> interner Regler Kühlen EIN  
(Initialisierungswert nach Reset)

*nviPrimaryCtrl1* = 0 ... 100 % ==> interner Regler AUS

==> *nviPrimaryCtrl1* bestimmt die Ausgangsgrößen

**!! Die externe Übersteuerung hat höchste Priorität, auch eine gleichzeitige**

**!! Ansteuerung von Heiz- und Kühlventil ist daher möglich.**

## Softwarebeschreibung

### nviPrimaryCtrl2

SNVT Typ: SNVT\_lev\_percent, Index 81

Funktion: Steuervariable für die Netzwerkvariable nvoPrimaryCtrl2 und *nvoDOCtrl2*.  
nvoPrimaryCtrl2 = 0x7FFF (163,835 %) ==> interner Regler Heizen EIN  
(Initialisierungswert nach Reset)  
nvoPrimaryCtrl2 = 0 ... 100 % ==> interner Regler AUS  
==> nvoPrimaryCtrl2 bestimmt die Ausgangsgrößen  
**!! Die externe Übersteuerung hat höchste Priorität, auch eine gleichzeitige  
!! Ansteuerung von Heiz- und Kühlventil ist daher möglich.**

### nviOccManCmd

SNVT Typ: SNVT\_occupancy, Index 109

Funktion: Eingangsvariablen zur Vorgabe der Raumbelegung. Die aktuelle Raumbelegung bestimmt die Größen der Regelparameter „effektiver Sollwert“ und „Neutrale Zone“ und damit die Sollwerte für Heizen und Kühlen (siehe Tabelle 1). Initialisierungswert einstellbar über *UCPToccupancy*.

nviOccManCmd: Vorgabe über GLT mit: OC\_OCCUPIED, OC\_STANDBY, OC\_UNOCCUPIED

nviOccManCmd	Wippe	>>>	nvoOccupEffect	NvoSetptEffect
OC_OCCUPIED	****	>>>	OCCUPIED	SCPTsetpoint + manuelle Sollwertverschiebung <b>oder</b> nviSetpoint + manuelle Sollwertverschiebung
****	OC_OCCUPIED	>>>		
OC_STANDBY	****	>>>	STANDBY	SCPTsetpoint + manuelle Sollwertverschiebung <b>oder</b> nviSetpoint + manuelle Sollwertverschiebung
OC_OCCUPIED	OC_STANDBY	>>>		
OC_UNOCCUPIED	****	>>>	UNOCCUPIED	SCPTsetpoint + manuelle Sollwertverschiebung <b>oder</b> nviSetpoint + manuelle Sollwertverschiebung
OC_UNOCCUPIED	OC_STANDBY	>>>		

**Tabelle 1: Sollwert, Raumbelegung**

### nviSetpoint

SNVT Typ: SNVT\_temp\_p, Index 105

Funktion: Eingangsvariable zur Vorgabe der Basis-Sollwerttemperatur.  
Es ist nicht zwingend erforderlich diese Netzwerkvariable mit einem übergeordneten Knoten zu binden. Wenn für nviSetpoint kein Update erfolgt, dann bleibt der Initialisierungswert 0x7FFF (=327,67°C) erhalten und es werden zur Berechnung des effektiven Sollwertes (Basis-Sollwert + Offset) die Werte des Konfigurationsparameters *SCPTsetPnt*s herangezogen. Erhält *nviSetpoint* ein Update mit einem gültigen Sollwert, dann wird der effektive Sollwert mit dem Wert der Eingangsvariablen berechnet.

### nviEnergyHoldOff

SNVT Typ: SNVT\_switch, Index 95

Funktion: Eingangsvariable von z.B. Fenster- oder Türkontakt zur Aktivierung der Energiesparfunktion. Mit *nviEnergyHoldOff* = 100.0 1 wird die Funktion aktiviert und die Stellgrößen Heizen/Kühlen werden auf ihre Minimalwerte zurückgesetzt. Bei aktiver Energiesparfunktion wird die Frostschutzfunktion eingeschaltet (siehe UCPTspAntiFreez). Nach Deaktivierung der Energy-Hold-Off Funktion wird die Temperaturregelung neu gestartet.

### nviDewPtSensor

SNVT Typ: SNVT\_switch, Index 95

Funktion: Eingangsvariable zur Auswertung eines Kondensationswächters im Betriebsmodus Kühlen. Mit *nviDewPtSensor* = 100.0 1 wird die Stellgröße Kühlen auf ihren Minimalwert zurückgesetzt. Nach Deaktivierung dieser Funktion wird die Temperaturregelung neu gestartet.

### nviFanSpeedCmd

SNVT Typ: SNVT\_switch, Index 95

Funktion: Eingangsvariable zur externen Vorgabe der angezeigten und mit *nvoFanSpeed* ausgegebenen Lüftereinstellung. Der Wertebereich entspricht dem der Ausgangsvariablen *nvoFanSpeed*.

## 4.4.2 Ausgangsvariablen Space Comfort Controller Object

### nvoSpaceTemp

SNVT Typ: SNVT\_temp\_p, Index 105

Funktion: Ausgangsvariable für den gemessenen Temperaturwert. Messbereich 0 - 50°C, Auflösung 1/100 °C. Die Datenausgabe erfolgt in Abhängigkeit von *SCPTmaxSendTime* und 5s - 15s nach Reset.

### nvoUnitStatus

SNVT Typ: SNVT\_hvac\_status, Index 112

Funktion: Ausgangsvariable für den Betriebsstatus und den Stellgrößen Heizen/Kühlen des Reglers.

.mode = HVAC_AUTO	==>	automatisches Umschalten zwischen Heizen und Kühlen
HVAC_HEAT	==>	nur Heizen
HVAC_COOL	==>	nur Kühlen
HVAC_OFF	==>	Regelung ausgeschaltet
.heat_output_primary	0...100 %	==> Stellgröße Heizen
.cool_output_primary	0...100 %	==> Stellgröße Kühlen

### nvoPrimaryCtrl1

SNVT Typ: SNVT\_lev\_percent, Index 81

Funktion: Ausgangsvariable mit der Stellgröße des Reglers 1 (Heizen) zur Ansteuerung eines stetigen Stellantriebs. Die Datenausgabe erfolgt in Abhängigkeit von *SCPTmaxSendTime* und 5s - 15s nach Reset.

### nvoDOCtrl1

SNVT Typ: SNVT\_switch, Index 95

Funktion: Ausgangsvariable mit der Stellgröße des Reglers 1 (Heizen) zur pulsweitenmodulierten Ansteuerung oder zur Zweipunkt-Ansteuerung eines thermischen Zweipunkt-Stellantriebs. Die Datenausgabe erfolgt sofort bei einem anstehenden Schaltbefehl und ansonsten in Abhängigkeit von *SCPTmaxSendTime* und 5s - 15s nach Reset.

### nvoPrimaryCtrl2

SNVT Typ: SNVT\_lev\_percent, Index 81

Funktion: Ausgangsvariable mit der Stellgröße des Reglers 2 (2-stufiges Heizen oder Kühlen) zur Ansteuerung eines stetigen Stellantriebs. Die Datenausgabe erfolgt in Abhängigkeit von *SCPTmaxSendTime* und 5s - 15s nach Reset.

### nvoDOCtrl2

SNVT Typ: SNVT\_switch, Index 95

Funktion: Ausgangsvariable mit der Stellgröße des Reglers 2 (2-stufiges Heizen oder Kühlen) zur pulsweitenmodulierten Ansteuerung oder zur Zweipunkt-Ansteuerung eines thermischen Zweipunkt-Stellantriebs. Die Datenausgabe erfolgt sofort bei einem anstehenden Schaltbefehl und ansonsten in Abhängigkeit von *SCPTmaxSendTime* und 5s - 15s nach Reset.

### nvoSetptEffect

SNVT Typ: SNVT\_temp\_p, Index 105

Funktion: Die Ausgangsvariable sendet den vom Regelalgorithmus verwendeten Sollwert. Die Ausgabe ist vom Betriebsmodus des Reglers abhängig:

nviApplicMode = HVAC\_AUTO ==> nvoSetptEffect = Basissollwert  
= (Sollwert Heizen + Sollwert Kühlen)/2  
nviApplicMode = HVAC\_HEAT ==> nvoSetptEffect = Sollwert Heizen  
nviApplicMode = HVAC\_Cool ==> nvoSetptEffect = Sollwert Kühlen

Der effektive Sollwert wird in Abhängigkeit von *nviSetpoint*, *nviOccManCmd*, *SCPTsetPnts* und der manuellen Verstellung berechnet (siehe Tabelle 1). Die Datenausgabe erfolgt in Abhängigkeit von *SCPTmaxSendTime*, bei Wertänderungen und 5s - 15s nach Reset.

### nvoOccupEffect

SNVT Typ: SNVT\_occupancy, Index 109

Funktion: Ausgangsvariablen für die effektive Raumbesetzung (siehe Tabelle 1). Die Datenausgabe erfolgt in Abhängigkeit von *SCPTmaxSendTime*, bei Wertänderungen und 5s - 15s nach Reset.

### nvoEnergyHoldOff

SNVT Typ: SNVT\_switch, Index 95

Funktion: Ausgangsvariable zur Statusanzeige der Energiesparfunktion.

nvoEnergyHoldOff = 0.0 0 ==> Fensterkontakt nicht aktiv  
nvoEnergyHoldOff = 100.0 1 ==> Fensterkontakt aktiv

Die Datenausgabe erfolgt in Abhängigkeit von *SCPTmaxSendTime*, bei Wertänderungen und 5s - 15s nach Reset.

### nvoDewPtSensor

SNVT Typ: SNVT\_switch, Index 95

Funktion: Ausgangsvariable zur Statusanzeige des Kondensationswächters.

nvoDewPtSensor = 0.0 0 ==> Kondensationswächter nicht aktiv  
nvoDewPtSensor = 100.0 1 ==> Kondensationswächter aktiv

Die Datenausgabe erfolgt in Abhängigkeit von *SCPTmaxSendTime*, bei Wertänderungen und 5s - 15s nach Reset.

### nvoFanSpeed

SNVT Typ: SNVT\_switch, Index 95

Funktion: Ausgangsvariablen für die eingestellte Lüftergeschwindigkeit. Die Anzeige im Display zeigt die aktuell eingestellte Lüftergeschwindigkeit an. Änderung in 5 %-Schritten.

Wertebereich: 0 - 100 %

## 4.4.3 Konfigurationsparameter Space Comfort Controller Object - Allgemein

### SCPTHvacType - Reglertyp

Index: 169, SNVT\_hvac\_type

Funktion: Konfigurationsparameter zur Kennzeichnung des Reglertyps.  
Eingestellter Wert: SCPTHvacType = HVT\_GENERIC

### SCPTHvacMode - Initialisierung

Index: 74, SNVT\_hvac\_mode

Funktion: Der Konfigurationsparameter bestimmt den Initialisierungszustand der Eingangsvariablen *nviApplicMode* und damit auch die Startkonfiguration des Temperaturreglers.  
(Voreingestellter Wert: HVAC\_AUTO)

### SCPTmaxSendTime - Heartbeat

Index: 49, SNVT\_time\_sec

Funktion: Der Konfigurationsparameter definiert den Sendezeitpunkt der Ausgangsvariablen. Mit Eingabewerten = 0 wird die Datenausgabe deaktiviert. (Voreingestellter Wert: 30 s)

### SCPTtempOffset - Temperatur Offset

Index: 272, SNVT\_temp\_diff\_p

Funktion: Offset für den Temperaturwert. Mit diesem Parameter ist eine softwareseitige Kalibrierung möglich. (Voreingestellter Wert: 0.0 °C)

Wertebereich: -12,8 K .... 12,7 K

## Softwarebeschreibung

### SCPTminSendTime

Index: 52, SNVT\_time\_sec

Funktion: Legt das kleinste Update-Intervall der Ausgangsvariablen nvoSpaceTemp fest. Ein Update erfolgt nach Ablauf von „SCPTminSendTime“, wenn sich der Temperaturwert der Ausgangsvariablen um mehr als „SCPTminDeltaTemp“ geändert hat. Mit Eingabewerten = 0 wird die Funktion deaktiviert. (Voreingestellter Wert: 5,0 sec)

### SCPTminDeltaTemp

Index: 64, SNVT\_temp\_p

Funktion: Wenn sich die Temperatur um den eingestellten Wert „SCPTminDeltaTemp“ verändert, wird der neue Temperaturwert übertragen. Die Funktion ist abhängig von der Einstellung des Parameters „SCPTminSendTime“. (Wertebereich  $\geq 0$  °C; Voreingestellter Wert: 0,30 °C)

### SCPTnumValves - Heizungsart

Index: 59, SNVT\_count

Funktion: Der Konfigurationsparameter dient zur Auswahl von 2-Rohr- oder 4-Rohr-Systemen. Wird ein 2-Rohr-System betrieben (1 Ventil), dann erhalten die Ausgangsvariablen mit den Stellgrößen für Heizen und Kühlen die gleichen Werte.

SCPTnumValves = 1: ==> 2-Rohr-System

Modus Heizen: nvoPrimaryCtrl1 = nvoPrimaryCtrl2 = Stellgröße Heizen

Modus Kühlen: nvoPrimaryCtrl1 = nvoPrimaryCtrl2 = Stellgröße Kühlen

SCPTnumValves = 2: ==> 4-Rohr-System (Standardwert)

Modus Heizen: nvoPrimaryCtrl1 = Stellgröße Heizen

Modus Kühlen: nvoPrimaryCtrl2 = Stellgröße Kühlen

### SCPTminRnge

Index: 23, SNVT\_switch

Funktion: Untere Begrenzung der Lüfterstufenverstellung.

Wertebereich: 0 – 30 %

### SCPTmaxRnge

Index: 20, SNVT\_switch

Funktion: Obere Begrenzung der Lüfterstufenverstellung.

Wertebereich: 70 – 100 %

### UCPTpwmCycleTime – PWM-Zykluszeit

Index: 35, SNVT\_time\_min

Funktion: Der Konfigurationsparameter bestimmt die Zykluszeit zur pulswidenmodulierten Ansteuerung der Stellantriebe mit *nvoDOCtrl1* und *nvoDOCtrl2*. (Voreingestellter Wert: 15 min)

### UCPTcontrolType - Reglertyp

Index: 51, typedef struct {unsigned short Byte[4]} UNVT\_str\_hex4

Funktion: Mit UCPTcontrolType.Byte[0] kann der Reglertyp 1 und mit UCPTcontrolType.Byte[1] der Reglertyp 2 eingestellt werden. Regler 1 (Ctrl1) kann nur Heizen während der Regler 2 (Ctrl2) 2-stufig Heizen oder Kühlen kann. Beide Regler können stetig oder mit Zweipunkt-Verhalten regeln. Bei einer stetigen Regelung wird automatisch über *nvoDOCtrl1* bzw. *nvoDOCtrl2* die pulswidenmodulierte Ansteuerung ausgegeben (Tabelle 2).

Konfiguration der Regler zur Klimasteuerung UCPTcontrolType			
Zuordnung Taster - Funktionsblock		Funktion	
Byte[0]	Beschreibung	Byte[1]	Beschreibung
0	kein Regler	0	kein Regler
1	stetig Heizen	1	stetig Kühlen
2	2-Punkt Heizen	2	2-Punkt Kühlen
		3	2. Heizstufe stetig Heizen
		4	2. Heizstufe 2-Punkt Heizen

**Tabelle 2: Reglerauswahl**

## UCPTcontrolConfig - Reglereinstellungen

Index: 61, SNVT\_state

Funktion: Mit UCPTcontrolConfig können Reglereinstellungen getätigt werden.  
UCPTcontrolConfig.

bit0		
0*	==>	bei Empfang eines nviSetpoint manuelle Sollwertverstellung zurücksetzen
1	==>	bei Empfang eines nviSetpoint manuelle Sollwertverstellung unverändert
bit1		
0*	==>	Partyfunktion nicht bei Frostschutz
1	==>	Partyfunktion auch bei Frostschutz
bit2		
0*	==>	Normale Heizen Ausgabe
1	==>	Invertierung Heizen Ausgabe
bit3		
0*	==>	Normale Kühlen Ausgabe
1	==>	Invertierung Kühlen Ausgabe

## UCPTvalveServiceTime - Wartungsintervall

Index: 68, SNVT\_time\_hour

Funktion: Der Konfigurationsparameter bestimmt das Wartungsintervall für das Ventil. Nach Ablauf des Wartungsintervalls wird das Ventil einmal komplett geöffnet und wieder geschlossen. Dies dient zum Schutz des Ventils und beugt Kalkablagerungen vor.

## UCPTvalveServiceDuration – Dauer Wartungsintervall

Index: 69, SNVT\_time\_min

Funktion: Der Konfigurationsparameter bestimmt die Zeitspanne, wie lange das Ventil zum Schutz vor Ablagerungen geöffnet bleiben soll. Diese Zeit sollte mindestens ein komplettes Öffnen des Ventils gewährleisten.

### 4.4.4 Konfigurationsparameter Space Comfort Controller Object – Sollwert

#### SCPTsetPnts - Sollwerte

Index: 60, SNVT\_temp\_setpt

Funktion: Konfigurationsparameter zur Vorgabe der Sollwerte für Heizen und Kühlen in Abhängigkeit der Raumbellegung. Mit nviSetpoint können die Werte überschrieben werden.

Voreingestellte Werte:

.occupied_heat	21 °C
.standby_heat	19 °C
.unoccupied_heat	16 °C
.occupied_cool	23 °C
.standby_cool	25 °C
.unoccupied_cool	28 °C

Wertebereich:

.occupied_heat	16 ... 31 °C
.standby_heat	occupied_heat - 1...8 K
.unoccupied_heat	occupied_heat - 1...8 K
.occupied_cool	occupied_heat + 1...8 K
.standby_cool	occupied_cool + 1...8 K
.unoccupied_cool	occupied_cool + 1...8 K

Achtung: Die Nachkommastelle wird nicht berücksichtigt.

#### SCPTbypassTime - Partyverlängerung

Index: 34, SNVT\_time\_min

Funktion: Konfigurationsparameter zur Vorgabe der Partyzeitverlängerung. Im Nachtbetrieb kann über das Bediengerät für die eingestellte Zeit in den Komfortbetrieb umgeschaltet werden.

Wertebereich: 0 min; 30 min; 60 min; 90 min; 120 min; 150 min; 180 min; 240 min

#### UCPTspAntiFreez - Frostschutz

Index: 18, SNVT\_temp\_p

Funktion: Sollwert für Heizen zur Frostschutzfunktion bei geöffnetem Fensterkontakt, d.h. bei aktiver Energiesparfunktion. (Voreingestellter Wert: 10 °C)

Wertebereich: 5 ... 10 °C

#### UCPToccupancy – Raumbellegung nach Reset

Index: 60, SNVT\_occupancy

Funktion: Konfigurationsparameter zur Vorgabe der Raumbellegung nach einem Reset. (Voreingestellter Wert: OC\_OCCUPIED)

Wertebereich: OCCUPIED, STANDBY

#### UCPTspOffsetRng – manuelle Sollwertverstellung

Index: 12, SNVT\_temp\_p

Funktion: Konfigurationsparameter für den Wertebereich der einstellbaren Sollwertkorrektur, d.h. der vorgegebene Sollwert lässt sich um den Wert +/- UCPTspOffsetRng durch den Benutzer verändern. (Voreingestellter Wert: 3K)

Wertebereich: +/- 1K; +/- 3K; +/- 5K

#### UCPTspOffsetCtrl2 – Sollwert zweite Heizstufe

Index: 58, SNVT\_temp\_p

Funktion: Konfigurationsparameter für 2-stufiges Heizen. Gibt den Stufenabstand von der Grundstufe zur Zusatzstufe an. (Voreingestellter Wert: 2,0 K)

Wertebereich: 1...8 K



## 4.4.5 Konfigurationsparameter Space Comfort Controller Object –Stetiger-Regler

### Einstellbare Parameter

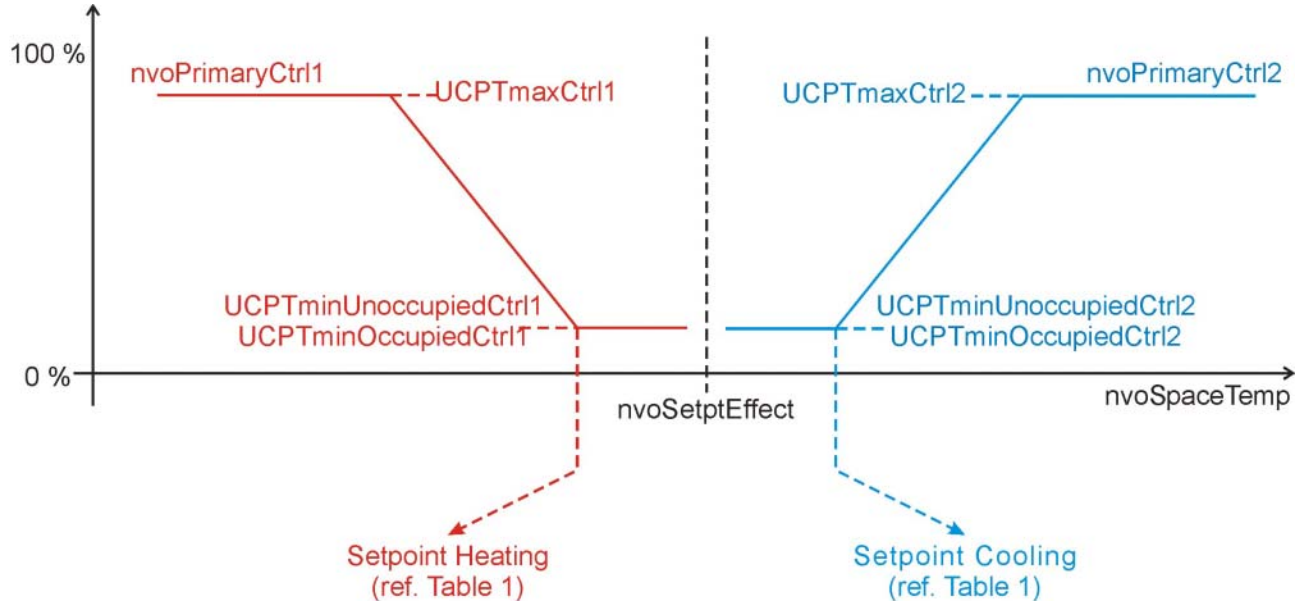


Abbildung 4-2: Parameter

#### UCPTxpCtrl1 – Proportionalbereich Regler 1

Index: 54, SNVT\_temp\_p

Funktion: Parameter zur Einstellung des Proportionalbereichs. Mit UCPTxpCtrl1 Xp = 0 wird der Regler 1 (für Heizen) deaktiviert. (Voreingestellter Wert: 4 K)

Wertebereich: 0...10 K

#### UCPTtnCtrl1 – Nachstellzeit Regler 1

Index: 55, SNVT\_time\_min

Funktion: Parameter zur Einstellung der Nachstellzeit des I-Anteils. Mit Eingabewerten = 0 wird der I-Anteil deaktiviert. (Voreingestellter Wert: 100 min)

Wertebereich: 0 min ... 240 min

#### UCPTminUnoccupiedCtrl1 - Stellgrößenbeschränkung

Index: 62, SNVT\_lev\_percent

Funktion: Stellgrößenbeschränkung nach Unten im Betriebsmodus UNOCCUPIED. (Voreingestellter Wert: 0 %)

#### UCPTminOccupiedCtrl1- Stellgrößenbeschränkung

Index: 64, SNVT\_lev\_percent

Funktion: Stellgrößenbeschränkung nach Unten in den Betriebsmodi OCCUPIED und STANDBY. (Voreingestellter Wert: 0 %)

#### UCPTmaxCtrl1- Stellgrößenbeschränkung

Index: 66, SNVT\_lev\_percent

Funktion: Stellgrößenbeschränkung nach Oben. (Voreingestellter Wert: 100 %)

#### UCPTxpCtrl2– Proportionalbereich Regler 1

Index: 56, SNVT\_temp\_p

Funktion: Parameter zur Einstellung des Proportionalbereichs. Mit UCPTxpCtrl2 = 0 wird der Regler 2 deaktiviert. (Voreingestellter Wert: 4 K)

Wertebereich: 0...10 K

## Softwarebeschreibung

### **UCPTtnCtrl2**– Nachstellzeit Regler 2

Index: 57, SNVT\_time\_min

Funktion: Parameter zur Einstellung der Nachstellzeit des I-Anteils.  
Mit Eingabewerten = 0 wird der I-Anteil deaktiviert. (Voreingestellter Wert: 100 min)

Wertebereich: 0 min ... 240 min

### **UCPTminUnoccupiedCtrl2** - Stellgrößenbeschränkung

Index: 63, SNVT\_lev\_percent

Funktion: Stellgrößenbeschränkung nach Unten im Betriebsmodus UNOCCUPIED.  
(Voreingestellter Wert: 0 %)

### **UCPTminOccupiedCtrl2** - Stellgrößenbeschränkung

Index: 65, SNVT\_lev\_percent

Funktion: Stellgrößenbeschränkung nach Unten in den Betriebsmodi OCCUPIED und STANDBY.  
(Voreingestellter Wert: 0 %)

### **UCPTmaxCtrl2** - Stellgrößenbeschränkung

Index: 67, SNVT\_lev\_percent

Funktion: Stellgrößenbeschränkung nach Oben. (Voreingestellter Wert: 100 %)

## **4.4.6 Konfigurationsparameter Space Comfort Controller Object - Zweipunkt-Regler**

### **UCPTHysteresisCtrl1** - Hysterese

Index: 52, SNVT\_temp\_p

Funktion: Parameter zur Einstellung der Hystereseschleife. (Voreingestellter Wert: 1 K)

Wertebereich: 0...10 K

### **UCPTHysteresisCtrl2** - Hysterese

Index: 53, SNVT\_temp\_p

Funktion: Parameter zur Einstellung der Hystereseschleife. (Voreingestellter Wert: 1 K)

Wertebereich: 0...10 K