Multi-function application with RTR and FanCoil 1816/1.0

Function overview

This application provides you with the following functions:

- Temperature control with various display options
- Scene function
- Disable function for push-buttons
- Time control
- Valve control
- Push-button functions

All buttons can be assigned different functions independently. You can do the following:

- Switch and toggle
- Dim
- Control blinds
- Save and retrieve scenes
- Select a linear regulator function
- Save edge functions
- Adjust setpoints
- Access operation modes and toggle between them

The status LEDs can also be utilised independently of one another and in a wide variety of ways.

Group addresses

Group addresses are managed dynamically. Maximum number of group addresses and assignments: 254 addresses, 255 connections

Notes on this documentation

This application enables you to implement a multitude of functions with the push-button. However, which functions are possible in each individual case depends on the KNX devices being controlled (e.g. dimming actuators, switch actuators etc.). The functions described here therefore show only the settings for this push-button.

Many parameters and their settings are dependent on the settings you have already made for other parameters. This means that some parameters will appear or disappear and the values available for selection will change according to settings you have already made. These dependencies have not been shown in the table for reasons of clarity. All settings are always shown.

Configurable times (staircase timer, ON delay, OFF delay, cyclic intervals etc.) are set via the base and factor parameters. The actual time is given by the multiplication of the two values. Example: Base = 1 second * factor = 3 gives 3 seconds.

The **bold** values in a table are the values set during factory configuration.

Basic settings

Before you begin, make a few basic settings in the "General" tab. You can specify the following:

- Whether the operational LED switches off or displays that the device is operating.
- How the date and time are received.
- Whether and how the time is sent cyclically to the bus.
- How large the start-up delay for the device should be after bus voltage recovery.

General	
Parameter	**Settings**
Switch on operational LED	**Yes**
	No
Receive date and time	In one communication object
	In two communication objects
Send time cyclically	**No**
	Every minute
	Every hour
	Daily
Format of sent time	Date/time format (8 byte)
	Time format (3 byte)
Device's start-up delay in s (0-255)	2-255, **4**

General settings for date and time

Here you set whether the date and time are received by two communication objects (3 byte each) or by only one communication object. In the latter case, there is an 8 byte object available.

If you send the time cyclically to the bus, you either have an output object with 3 byte or one with 8 byte, depending on which format you selected.

You can find more details on time control, synchronisation and master-slave functions in the "Time control" chapter.

Start-up delay

The start-up delay is the time delay between the bus voltage recovery and the device's functional start. Set a time after which the current values can be read by other KNX devices.

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Time control	Date object input	3 byte	Low	WC	Receive
Time control	Time object input	3 byte	Low	WC	Receive
Time control	Date/time object input	8 byte	Low	WC	Receive
Time control	Date/time object output	8 byte	Low	CT	Transmit
Time control	Time object output	3 byte	Low	CT	Transmit

Setting the display

Here, you set the display options and display functions:

- You can disable the control menu so that the menu buttons cannot be used to make further changes to the settings.
- You use the display mode to specify which values you want to display on the basic display (actual temperature, setpoint temperature, time, date, external temperature and fan speed). You can display these values either individually or in alternation. You can also set the display rhythm for alternating displays. The display mode can also be set using the control menu on the push-button.
- For the weekday display, you can set day "1". This means that the first weekday can be defined as: Friday, Saturday, Sunday or Monday. Each weekday is then shown in the display with a different number accordingly.
- Time format and unit of temperature display (Celsius/ Fahrenheit).
- The background lighting on the display can either be switched off or on at all times, or can be switched on during operation. You can also set the persistence period and the brightness of the background lighting.

The parameters for the background illumination and display mode can be changed using the menu buttons on the push-button (see operating instructions).

If you use room temperature control ("Control general" tab), additional possible settings appear:

- You can specify whether or not the operation mode can be changed when the frost or heat protection operation mode is activated.
- "Access menu setpoint temperature/operation mode directly" Here, you specify which operation mode you want to select with a short push of a button on the menu buttons. You can choose between the control menu "Set setpoint temperature" or "Set operation mode".
- On the display, a symbol can show either the current controller status or whether heating or cooling is activated.

Display heating and cooling symbol = shows current controller status

Display	Type of control
	Heating active, correcting variable ≠0
	Cooling active, correcting variable ≠0
1 2 (below the symbols)	1 = Heating/cooling active, correcting variable ≠0 For two-stage heating/cooling 1 = Basic level active, correcting variable ≠0 2 = Basic level and additional level active, correcting variable ≠0
--	Insensitive zone, correcting variables = 0

Display heating and cooling symbol = shows heating/cooling

Display	Type of control
	Heating
	Cooling
1 2 (below the symbols)	1 = Heating/cooling active, correcting variable ≠0 For two-stage heating/cooling: 1 = Basic level active, correcting variable ≠0 2 = Basic level and additional level active, correcting variable ≠0

Display	
Parameter	**Settings**
User menu enabled	**Yes** No
Adjust operation mode during frost / heat protection	Yes **No**
Set display mode (multiple names appear during the change)	**Actual temperature** Setpoint temperature External temperature Date Time Fan speed Date/time Date, time, fan speed Actual temperature, setpoint temperature Actual temperature, setpoint temperature, time Actual temperature, setpoint temperature, fan speed External temperature, actual temperature External temperature, actual temperature, time Actual temperature, setpoint temperature, time Actual temperature, setpoint temperature, fan speed, time External temperature, actual temperature, fan speed, time

Display	
Parameter	**Settings**
Display rhythm	3 s - 10 s, **5 s**
Time display	**00:00 ...23:59** 01:00 ...12:59 (AM/PM)
Access menu setpoint temperature/operation mode directly (menu buttons)	No **Setpoint adjustment** Operation mode changed
Display heating and cooling symbol	Shows the current controller status **Shows heating/cooling**
Unit of temperature display	**°C**, °F
Display "1" corresponds to	Friday Saturday Sunday **Monday**
Switch on background lighting	No Yes **During operation + persistence**
Persistence period in s (1-254)	1-254, **10**
Set brightness (1 = very dark / 10 = very bright)	1-10, **5**

Use signal function

The signal function enables the device to display whether the actual temperature is above or below a preset value. The signal is shown on the display by the bell symbol, and is ended when the actual temperature returns to within the set range.

When an signal is issued, a "1" is transmitted to the bus via the "Signal object output", and a "0" is transmitted after the signal is finished. The "Signal object input" can also be set to "1" by another sensor.

The signal cannot be acknowledged. It is not possible to acknowledge the signal when the alarm is triggered via the feedback object.

Signal function	
Parameter	**Settings**
Use signal function	Yes **No**
Report if actual temperature is higher than	20,0 = 68.0 °F to 40.0 °C = 104.0 °F, **No**
Report if actual temperature is lower than	0,0 = 32,0 °F to 19,0 °C = 166.2 °F, **No**

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Signal function	Feedback object input	1 bit	Low	WC	Receive
Signal function	Feedback object output	1 bit	Low	CT	Transmit

Push-button information

On the “Push-button info” tab you can see which push-button names in ETS correspond to which push-buttons on the device. The names assigned cannot be changed.

Push-button info	
Parameter	
Push-button 1 =	Upper left push-button
Push-button 2 =	Upper right push-button
Push-button 3 =	Lower left push-button
Push-button 4 =	Lower right push-button
Left menu button =	Left push-button next to display
Right menu button =	Right push-button next to display

Sending toggle commands - 1-bit, 1-byte

Each time the button is pressed, the 1-bit object type first inverts the object value and then transmits it to the bus, in other words making a “0” into a “1”. If the same button is pressed again, the “1” turns back into a “0”. The device is thus switched on and off alternately. This switching behaviour is called “toggling”.

For 1-byte object types, you can set two values, which are transmitted alternately after each press of the button.

An update or change to the object values is possible via the bus when another sensor switches the actuator (e.g. via a two-way circuit or a central command). To prevent “incorrect” toggling, the state of the actuator (“1” or “0”) must be tracked in the push-button. To do this, connect the group address of the second sensor to the switch/value object of the push-button.

Two objects (1 bit / 1 byte) can also be transmitted in any combination when the push-button is pressed.

Push-button X	
Parameter	**Settings**
Select push-button function	Toggle

Status indication

The status LED can:

- Be switched on or off continuously.
- Light up when pressed (for a long period), and go out when released.
- Flash.
- Display the status of the switch/value object. When the 1 byte object type is used, the LED lights up if value 1 is greater than zero.
- Display the status of the status feedback object.

Parameters

Parameter	Settings
Number of objects	**One**
	Two
Object A / Object B	**1 bit**
	1 byte in steps 0 % - 100 %
	1 byte continuous 0-255
Value 1	**100 %**, 90 %, 80 %, ..., 0 %, 25 %, 75 %
	255, 254, 253, ...0
Value 2	**0 %**, 10 %, 20 %, ... 100 %, 25 %, 75 %
	0, 1, 2, 3, ... 255
Trigger status LED	Switched on
	Switched off
	From switch/value object A
	From switch/value object B
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	Flashes
	Flashes when switch/value object A not equal to 0
	Flashes when switch/value object B not equal to 0
	Flashes when switch/value object A equals 0
	Flashes when switch/value object B equals 0
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Switch object A/B	1 bit	Low	WCT	Transmit/ receive
Push-button X	Value object A/B	1 byte	Low	WCT	Transmit/ receive
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Sending switching commands - 1-bit, 1-byte

When a push-button is pressed, the following values can be sent via the switch/value object

- An ON or OFF telegram
- 1 byte values in steps (0 % - 100 %)
- 1 byte values, infinitely adjustable (0-255)
- Two objects at the same time (1-bit, 1-byte) in any combination

Push-button X	
Parameter	**Settings**
Select push-button function	**Switching**

Status indication

The status LED can:

- Be switched on or off continuously.
- Light up when pressed (for a long period), and go out when released.
- Flash.
- Display the status of the switch/value object. When the 1 byte object type is used, the LED lights up if value 1 is greater than zero.
- Display the status of the status feedback object.

Parameters

Parameter	Settings
Number of objects	**One**
	Two
Object A / Object B	**1 bit**
	1 byte in steps 0 % - 100 %
	1 byte continuous 0-255
Value	**ON telegram**
	OFF telegram
	100 %, 90 %, 80 %, ..., 0 %, 25 %, 75 %
	255, 254, 253, ...0
Trigger status LED	Switched on
	Switched off
	From switch/value object A
	From switch/value object B
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	Flashes
	Flashes when switch/value object A not equal to 0
	Flashes when switch/value object B not equal to 0
	Flashes when switch/value object A equals 0
	Flashes when switch/value object B equals 0
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Switch object A/B	1 bit	Low	WCT	Transmit/ receive
Push-button X	Value object A/B	1 byte	Low	WCT	Transmit/ receive
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Dimming

You can use the dimming function for the following:

- Dim brighter **and** darker using **one push-button** (single-button dimming)
- Either dim brighter **or** darker. You need a second push-button to dim in the other direction (two-button dimming).

Push-button X	
Parameter	**Settings**
Select push-button function	Dimming

Status indication

The status LED can:

- Display the status of the switch object
- Light up when pressed (for a long period), and go out when released
- Be on or off continuously
- Flash
- Display the status of the status feedback object

Parameter	Settings
Trigger status LED	Switched on
	Switched off
	From switch object
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	Flashes
	Fashes when status feedback object not equal to 0
	Flashes when status feedback object equals 0
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF

Common parameters for single-button and two-button dimming

You can use the corresponding push-button to switch the light on or off (brief press) or dim it (longer press, the exact period can be parameterised). When switching takes place, an ON/OFF telegram is sent via the switch object. When dimming, dimming up or dimming down is carried out via the 4-bit dimming object; the parameters for the dimming steps can be set. You can also transmit the relevant dimming step cyclically for a period of time which can be set as required.

Parameter	Settings
Long operation time equals 100 ms * factor (4-250)	4 - 250, **6**
Dimming direction	Brighter
	Darker
	Brighter and darker
Send dimming levels cyclically	Yes
	No
Cycle time = basis * factor	
Basis	**0.1 s,** 1 s, 1 min
Factor (3-255)	3 - 255, **8**

Single-button dimming

You can dim both lighter **and** darker and also switch both on **and** off using a single push-button.

The current switching or dimming direction is always dependent on the previous action, i.e. if switched off, a brief push of the button will switch the light on and vice versa, and if the light has been dimmed up, prolonged operation of the push-button will dim the light down again. On release after prolonged actuation, a stop telegram will be transmitted via the 4-bit dimming object, thus terminating the dimming procedure in the dimming actuator.

An update or change to the object value is possible via the bus when another sensor switches or dims the actuator (e.g. via a two-way circuit or a central command). To prevent “incorrect” switching/dimming activity, the state of the actuator must be tracked in the push-button. To do this, connect the group address of the second sensor to the switch/dimming object of the push-button.

A single command is sufficient to cycle through the dimming range. This dimming procedure can be used for most applications. The other possible dimming steps (1/2 - 1/64 brighter or darker) dim brighter or darker by the selected step. For example, if the step is set to 1/4, you would need to push the button for a prolonged period four times in succession to dim from minimum to maximum brightness.

Parameter	Setting
Dimming direction	Brighter and darker
Step dimming (brighter)	**To max. brightness**
	1/2 brighter
	1/4 brighter
	1/8 brighter
	1/16 brighter
	1/32 brighter
	1/64 brighter
Step dimming (darker)	**To min. brightness**
	1/2 darker
	1/4 darker
	1/8 darker
	1/16 darker
	1/32 darker
	1/64 darker

Two-button dimming

You can dim either lighter **or** darker and switch either on **or** off with a single push-button. A second push-button for the opposite direction must be parameterised.

You can specify whether a stop telegram is to be transmitted when the push-button is released. If you have enabled the transmission of a stop telegram, a stop telegram will be transmitted via the 4-bit dimming object when the push-button is released after prolonged actuation, thus terminating the dimming procedure in the dimming actuator.

A single command is sufficient to cycle through the dimming range. This dimming procedure can be used for most applications. The other possible dimming steps (1/2 - 1/64 brighter or darker) dim brighter or darker by the selected step. For example, if the step is set to 1/4, you would need to push the button for a prolonged period four times in succession to dim from minimum to maximum brightness.

Parameter	Setting
Dimming direction	Brighter
	Darker
Step dimming (brighter)	**To max. brightness**
	1/2 brighter
	1/4 brighter
	1/8 brighter
	1/16 brighter
	1/32 brighter
	1/64 brighter
Step dimming (darker)	**To min. brightness**
	1/2 darker
	1/4 darker
	1/8 darker
	1/16 darker
	1/32 darker
	1/64 darker
Stop telegram after release	**Yes**
	No

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Switch object	1 bit	Low	WCT	Transmit/receive
Push-button X	Dimming object	4 bit	Low	WCT	Transmit/receive
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Blind control

You can use the blind control function to do the following:

- Raise the blinds/adjust the slats using a single push-button and lower the blinds/adjust the slats using a second push-button (two-button blind operation).
- Move the blind using an individual push-button and adjust the slats (single-button blind operation).
- Move the blind to a pre-specified position.
- Move the blind back and forth between two previously specified positions.

Push-button X	
Parameter	**Setting**
Select push-button function	Blind

Status indication

The status LED can:

- Flash
- Light up when pressed, and go out when released
- Be on or off continuously
- Display the status of the status feedback object

Parameter	**Setting**
Trigger status LED	Switched on
	Switched off
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	ON after long operation / release = OFF
	Flashes
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF

Two-button blind operation

You can either raise **or** lower the blind with a single push-button.

When the corresponding push-button is pressed for a short time, a stop/step telegram is transmitted; when the push-button is pressed for a longer period (the exact period can be parameterised), a movement telegram is transmitted. With this function, you must parameterise a second push-button with the corresponding settings for blind movement in the opposite direction. Both push-buttons must be given the same group addresses.

Parameter	**Setting**
Long operation time equals 100 ms * factor (4-250)	4 - 250, **6**
Direction of movement, blind	Up
	Down

Single-button blind operation

You can both raise **and** lower the blind with a single push-button.

The current direction of movement of the blind, or the direction of the slat adjustment, always depends on the previous action, i.e. when the blind has just been lowered, it will be raised the next time the push-button is activated for a long period (the exact period can be parameterised).

When a stop/step telegram has been transmitted to adjust the slats, a stop/step telegram for the same direction of movement can be generated by pressing the push-button again, as long as this subsequent push-button action is carried out within a set time period (which can be parameterised). If that time period has elapsed, the direction of rotation of the slats will change when the push-button is pressed briefly.

The push-button can receive telegrams via the stop/step and movement object, and can generate corresponding telegrams when the push-button is pressed, according to the values received. An update or change to the object values is possible via the bus when another sensor switches the actuator (e.g. via a two-way circuit or a central command). To prevent “incorrect” movement, the state of the actuator must be tracked in the push-button. To do this, connect the group address of the second sensor to the stop/step and movement object of the push-button.

Parameter	**Setting**
Long operation time equals 100 ms * factor (4-250)	4 - 250, **6**
Direction of movement, blind	Up and Down
Pause for slat - change of direction 100 ms * factor (5-50)	5 - 50, **10**

Moving the blind to a pre-specified position

If the blind actuator is capable of moving to specific position, you can use this function to specify one or two positions to which the blind can be moved using 1 byte position values with a push-button action. The position values can be set in steps between 0% and 100%, or infinitely from 0-255.

When moving to a position, the set value for the blind position and the slat position is transmitted using a short (or long) push-button action.

To address two positions, enter the required blind position and slat position for both. Position value 1 is transmitted with a short push-button action, while position value 2 is transmitted with a long push-button action. No movement or stop/step objects exist with these set parameters.

Parameter	Setting
Direction of movement, blind	With positional values
Select number of positionings	One position (short operation) Two positions (distinction between short/long operation)
Positional value 1 (short operation)	**In steps of 0% - 100%** Continuous 0-255
Position of blind	**100 %**, 90 %, 80 %, ..., 0 %, 25 %, 75 % **255**, 254, 253, ...0
Position of slats	**0 %**, 10 %, 20 %, ... 100 %, 25 %, 75 % **0**, 1, 2, 3, ... 255
Positional value 2 (long operation)	**In steps of 0% - 100%** Continuous 0-255
Position of blind	**100 %**, 90 %, 80 %, ..., 0 %, 25 %, 75 % **255**, 254, 253, ...0
Position of slats	**0 %**, 10 %, 20 %, ... 100 %, 25 %, 75 % **0**, 1, 2, 3, ... 255

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Stop/step object	1 bit	Low	WCT	Transmit/ receive
Push-button X	Movement object	1 bit	Low	WCT	Transmit/ receive
Push-button X	Blind position	1 byte	Low	CT	Transmit
Push-button X	Slat position	1 byte	Low	CT	Transmit
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Sending edge commands - 1 bit, 2 bit (priority), 4 bit, 1 byte

With this edge function you can transmit one or two objects simultaneously, and select the size of the objects required as needed (1 bit, 2 bit priority control, 4 bit or 1 byte in steps or infinitely). A distinction is made between the normal edge function and the extended edge function:

- With the normal edge function, you can specify which actions should be carried out when a push-button is pressed, and which should be carried out when a push-button is released.
- With the extended edge function you can also parameterise different actions to take place upon short and long operation of the push-button.

Push-button X	
Parameter	**Setting**
Select push-button function	Edges 1 bit, 2 bit (prio), 4 bit, 1 byte values
Select edge function	**Normal (operate, release)** Extended (+ long and short operation)

Status indication

The status LED can:

- Be switched on or off continuously.
- Light up when pressed (for a long period), and go out when released.
- Flash.
- Display the status of object A/B.
- Display the status of the status feedback object.

Parameter	Setting
Trigger status LED	Switched on Switched off **From object A** From object B From status feedback object Operation = ON / release = OFF Long operation = ON / release = OFF Flashes Flashes when object A not equal to 0 Flashes when object B not equal to 0 Flashes when object A equals 0 Flashes when object B equals 0 Flashes when status feedback object equals 1 Flashes when status feedback object equals 0 Operation = flash / release = OFF Long operation = flash / release = OFF

Normal edge function

With the normal edge function, you can specify which actions should be carried out when a push-button is pressed, and which should be carried out when a push-button is released. These actions could include:

- Send 1 or 0 (with 1 bit)
- Send value 1 or value 2 (with 2 bit, 4 bit or 1 byte): You can enter two values and set whether and how they are to be transmitted.
- Object sends its value: The object transmits the value which it currently has. Therefore you can, for example, transmit a value with the sending group address which was previously received via another group address.
- Toggle: The current object value is inverted and then transmitted. The device is thus switched on/off alternately or transmitted value 1/value 2 (toggling). The value can be modified via the bus.
- No action

The values available to you are 1 bit, 2 bit (priority control), 4 bit, 1 byte in steps or infinitely.

Push-button X	
Parameter	**Setting**
Edge function	Normal (operate, release)
Number of objects	**One**
	Two

Push-button X - edges object A/B	
Parameter	**Setting**
Object A / Object B	**1 bit**
	2 bit (priority control)
	4 bit
	1 byte in steps 0 % - 100 %
	1 byte continuous 0-255
Action on operation	**Sends 1**
	Sends 0
	Toggles
	Sends its value
	None
	Sends value 1
	Sends value 2
Action at release	Sends 1
	Sends 0
	Toggles
	Sends its value
	None
	Sends value 1
	Sends value 2

Push-button X - edges object A/B	
Parameter	**Setting**
Value 1	**Switch on with priority (11)**
	Switch off with priority (10)
	Remove priority control(00)
	Dim-darker-stop
	To min. brightness
	1/2 darker
	1/8 darker
	1/16 darker
	1/32 darker
	1/64 darker
	1/4 darker
	Dim-brighter-stop
	To max. brightness
	1/2 brighter
	1/4 brighter
	1/8 brighter
	1/16 brighter
	1/32 brighter
	1/64 brighter
	100 %, 90 %, 80 %, ..., 0 %, 25 %, 75 %
	255, 254, 253, ...0
Value 2	Switch on with priority (11)
	Switch off with priority (10)
	Remove priority control (00)
	Dim-darker-stop
	To min. brightness
	1/2 darker
	1/8 darker
	1/16 darker
	1/32 darker
	1/64 darker
	1/4 darker
	Dim-brighter-stop
	To max. brightness
	1/2 brighter
	1/4 brighter
	1/8 brighter
	1/16 brighter
	1/32 brighter
	1/64 brighter
	100 %, 90 %, 80 %, ..., **0 %**, 25 %, 75 %
	255, 254, 253, ...**0**

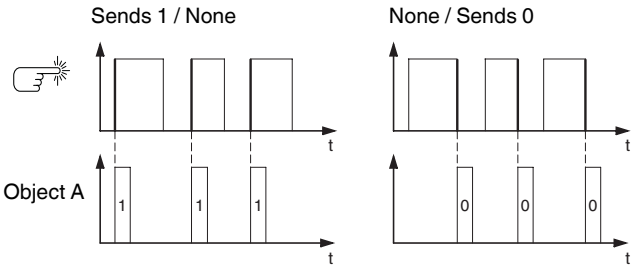
Principle of the edge function

Using the following diagrams, you can see how the edge function behaves when edges rise or fall.

The settings for “Action on operation / Action at release” are shown directly above each diagram.

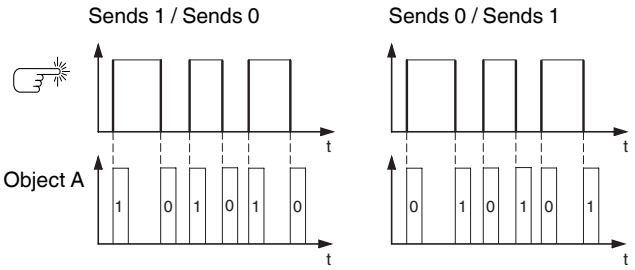
Example 1

Object A = 1 bit



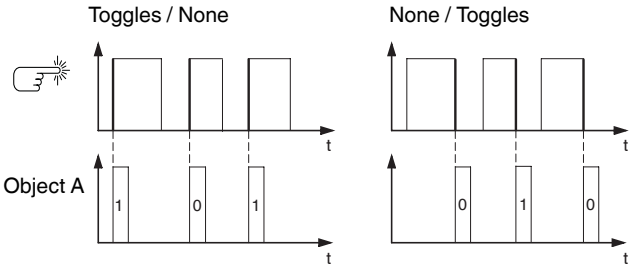
Example 2

Object A = 1 bit



Example 3

Object A = 1 bit

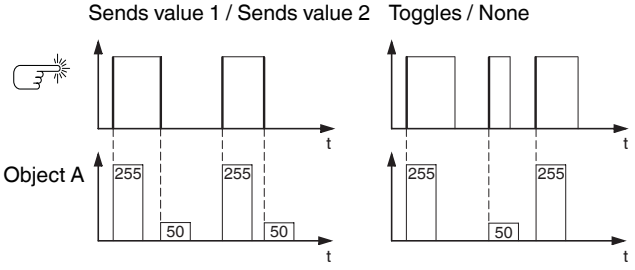


Example 4

Object A = 1 byte continuous 0-255

Value 1 = 255

Value 2 = 50



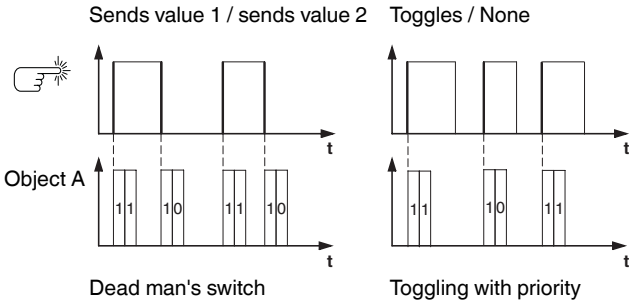
Example 4

Example 5

Object A = 2 bit (priority control)

Value 1 = 11 (switch on with priority)

Value 2 = 10 (switch off with priority)



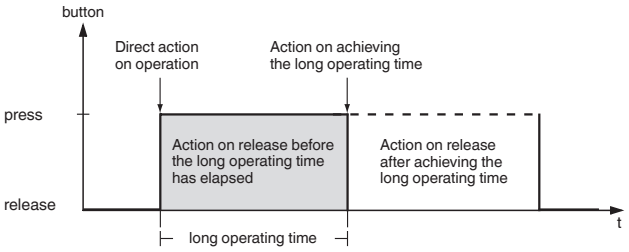
Extended edge function

With the extended edge function, you have a wider range of functions available. For example, you can set different actions for short and long presses of a push-button, both for when the push-button is pressed and for when it is released. You can also set a cycle time which can be parameterised for each object.

When parameterising, bear in mind that you need to set all four types of push-button operation (short/long press, pressing and releasing the button) in order to ensure that the push-button functions as required.

In order to read the object values, you may need to set the Read flags manually.

The following activation sequence chart shows the phases into which the pulse edge function is divided:



Push-button X	
Parameter	**Setting**
Edge function	Extended (+ long and short operation)
Long operation time equals 100 ms * factor (4-250)	4 - 250, **6**
Number of objects	**One** Two

Push-button X - edges object A/B	
Parameter	**Setting**
Object A/B	**1 bit**
	2 bit (priority control)
	4 bit
	1 byte in steps 0 % - 100 %
	1 byte contiuous 0-255
Direct action on operation	**Sends 1**
Action on release before the long operating time has elapsed	Sends 1 immediately and then cyclically
Action on achieving the long operating time	Sends 1 only cyclically
Action on release after achieving the long operating time	Sets object value to 1 (readable only)
	Sends 0
	Sends 0 immediately and then cyclically
	Sends 0 only cyclically
	Sets object value to 0 (readable only)
	Sends value 1
	Sends value 1 immediately and then cyclically
	Sends value 1 only cyclically
	Sets object value to value 1 (readable only)
	Sends value 2
	Sends value 2 immediately and then cyclically
	Sends value 2 only cyclically
	Sets object value to value 2 (readable only)
	Toggles
	Toggles, sends immediately, then cyclically
	Toggles, only sends cyclically
	Toggles and is not sent
	Toggles cyclically, sends immediately, then cyclically
	Toggles cyclically, only sends cyclically
	Toggles cyclically and is not sent
	Sends its value
	Sends its value immediately and then cyclically
	Sends 1 and after a cycle time 0
	Sends value 1, then value 2 after a cyclic time
	Cyclically increase the current object value by 1
	Cyclically reduce the current object value by 2
	None (stops cyclical sending)
	No change
	None (stop after current cycle time)

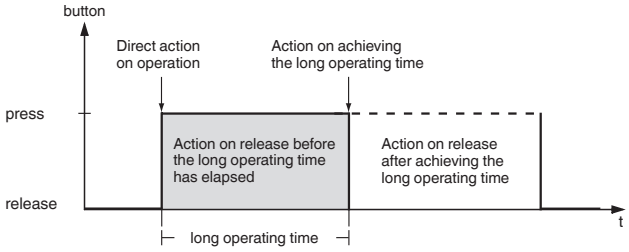
Push-button X - edges object A/B	
Parameter	**Setting**
Value 1	**Switch on with priority (11)**
	Switch off with priority (10)
	Remove priority control (00)
	Dim-darker-stop
	To min. brightness
	1/2 darker
	1/8 darker
	1/16 darker
	1/32 darker
	1/64 darker
	1/4 darker
	Dim-brighter-stop
	To max. brightness
	1/2 brighter
	1/4 brighter
	1/8 brighter
	1/16 brighter
	1/32 brighter
	1/64 brighter
	100 %, 90 %, 80 %, ..., 0 %, 25 %, 75 %
	255, 254, 253, ...0
Value 2	Switch on with priority (11)
	Switch off with priority (10)
	Remove priority control (00)
	Dim-darker-stop
	To min. brightness
	1/2 darker
	1/8 darker
	1/16 darker
	1/32 darker
	1/64 darker
	1/4 darker
	Dim-brighter-stop
	To max. brightness
	1/2 brighter
	1/4 brighter
	1/8 brighter
	1/16 brighter
	1/32 brighter
	1/64 brighter
	100 %, 90 %, 80 %, ..., **0 %**, 25 %, 75 %,
	255, 254, 253, ...**0**
Cycle time = basis * factor	
Basis	0.1 s, 1 s, **1 min**, 1 h, 1 day
Factor (3-255)	3-255, **10**

A description of the most important actions is given below:

- Sends [value]:
  Transmits the current value and stops a cyclical transmission.
- Sends [value] immediately and then cyclically:
  If no cycle time is running, [value] is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, [value] is transmitted and a new cycle time is started.
- Sends [value] only cyclically:
  If no cycle time is running, [value] is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is **not** interrupted; [value] is transmitted after the current cycle time has elapsed, and a new cycle time is started.
- Sets object value to [value] (readable only)
  [value] is written into the object and is not transmitted. Any active cycle time is terminated.
- Toggles:
  Compares the current object value with [value]. If both are the same, value 1 or value 2 is transmitted. If they are different, [value] is transmitted.
- Toggles, sends immediately, then cyclically:
  If no cycle time is running, the value is toggled (see “toggles”), transmitted immediately, and a new cycle time is started. If a cycle time is already running, it is interrupted, the toggled value is transmitted and a new cycle time is started. Subsequently, the value which has already been toggled is always transmitted cyclically.
- Toggles, only sends cyclically:
  If no cycle time is running, the toggled value is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is **not** interrupted; the toggled value is transmitted after the current cycle time has elapsed, and a new cycle time is started. Subsequently, the value which has already been toggled is always transmitted cyclically.
- Toggles and is not sent:
  The toggled value is written into the object and is not transmitted. Any active cycle time is terminated.
- Toggles cyclically, sends immediately, then cyclically:
  If no cycle time is running, the value is toggled (see “toggles”), transmitted immediately, and a new cycle time is started. If a cycle time is already running, it is interrupted, the toggled value is transmitted and a new cycle time is started. Subsequently, it is always toggled cyclically and the new value is transmitted.
- Toggles cyclically, only sends cyclically:
  If no cycle time is running, the toggled value is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is **not** interrupted; the toggled value is transmitted after the current cycle time has elapsed, and a new cycle time is started. Subsequently, it is always toggled cyclically and the new value is transmitted.
- Toggles cyclically and is not sent:
  The toggled value is written into the object and is **not** transmitted. Subsequently, it is always toggled cyclically and the new value is written into the object.
- Sends its value:
  The current object value is transmitted. Any active cycle time is terminated.
- Sends its value immediately and then cyclically:
  If no cycle time is running, the current object value is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, the current object value is transmitted and a new cycle time is started. Subsequently, the current object value is always transmitted cyclically.
- Cyclically increase the current object value by [value]:
  If no cycle time is running, [value] is added to the current object value, the object value is transmitted, and a new cycle time is started. If a cycle time is already running, it is **not** interrupted; the current object value with [value] added is transmitted and a new cycle time is started.
- Reduce the current object value by [value] cyclically:
  If no cycle time is running, [value] is subtracted from the current object value, the object value is transmitted, and a new cycle time is started. If a cycle time is already running, it is **not** interrupted; the current object value with [value] subtracted is transmitted and a new cycle time is started.
- Sends [value A] and after a cycle time [value B]:
  [value A] is transmitted immediately, and [value B] is transmitted after **one** cycle time, regardless of whether a cycle time is already running or not (staircase lighting timer function).
- None (stops cyclical sending):
  No action is carried out, and any active cycle time is stopped.
- No change:
  The current action remains unchanged (e.g. “sends value 1, then value 2 after a cycle time”).
- None (stop after current cycle time):
  No action is currently carried out, but any active cycle time is **not** stopped. It runs through until the end, and then transmits the corresponding value.

Examples of use for the edge function

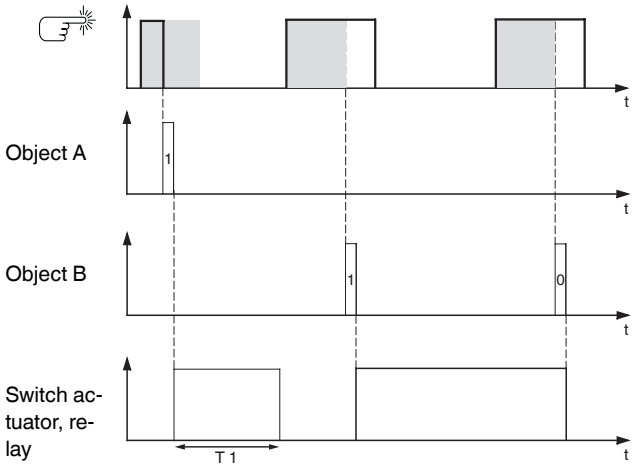
The following activation sequence chart shows the phases into which the pulse edge function is divided:



Staircase lighting function with cleaning light function

With a brief press of a push-button, the switch actuator switches on the light. A long press of the push-button extends the staircase lighting function (= cleaning light function) until a second long press of the button switches off the actuator. The switch actuator requires a staircase lighting function and a disable function for this function.

Number of objects = 2 (object A/B)
Object A/B = 1 bit
Object A: Action on release before the long operating time has elapsed = Sends 1
Object B: Action on achieving the long operating time = Toggles
Connect object A with the switch object and object B with the disable object of the switch actuator.



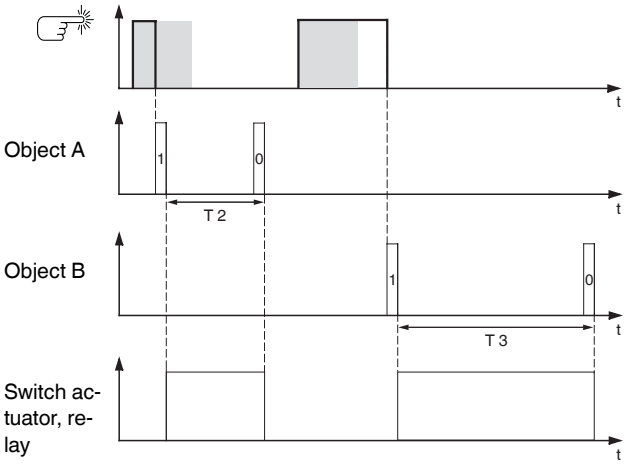
T 1 = Staircase timer period

Short and long staircase timer

You can use this function to produce a brief and a long staircase lighting time with the push-button. The switch actuator requires no staircase lighting function for this request.

With a brief press of the push-button, the switch actuator switches on the light, and after a parameterised cycle time (e.g. 3 minutes), it switches it back off again. With a long press of the push-button, the same function is carried out, but with a longer cycle time (e.g. 6 minutes).

Number of objects = 2 (object A/B)
Object A/B = 1 bit
Object A: Action on release before the long operating time has elapsed = Sends 1 and after a cycle time 0.
Cycle time = e.g. 3 minutes
Object B: Action on release after achieving the long operating time = Sends 1 and after a cycle time 0.
Cycle time = e.g. 6 minutes
Connect object A and object B with the switch object of the switch actuator.



T 2 = Short cycle time
T 3 = Long cycle time

Switching the light on/off permanently, or switching off after a cycle time has elapsed

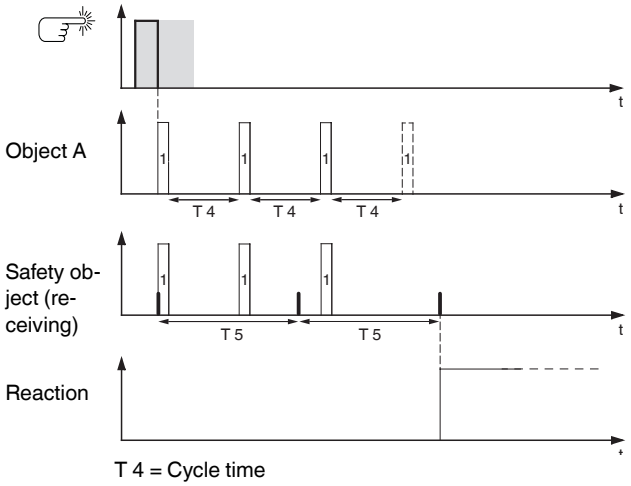
With a brief press of a push-button, the switch actuator switches the light on or off permanently. With a long press of a push-button, the light switches on, and after a parameterised cycle time (e.g. 6 minutes), it switches back off again. Due to the cycle time in the push-button which can be parameterised, the switch actuator requires no staircase lighting function for this function.

Number of objects = 2 (object A/B)
Object A/B = 1 bit
Object A: Action on release before the long operating time has elapsed = toggles
Object B: Action on achieving the long operating interval = sends 1 and after a cycle time 0.
Action on release after achieving the long operating time = no change.
Cycle time = e.g. 6 minutes.
Connect object A and object B with the switch object of the switch actuator.

Electronic protection against theft

This example will show you how to program electronic theft protection for the push-button. It is activated by a brief push-button action and then transmits cyclically. As soon as the push-button is forcibly separated from the bus, this can be reported or an alarm can be triggered.

Number of objects = 1 (object A)
Object A = 1 bit
Object A: Action on release before the long operating time has elapsed = Sends 1 immediately and then cyclically.
Action on achieving the long operating time = No change.
Action on release after achieving the long operating time = No change.
Cycle time = e.g. 10 minutes.
Connect object A with an object that listens cyclically for telegrams (e.g. a safety object). The monitoring time set on the safety object must be longer than the cycle time of the push-button. If the safety object receives no telegrams from the push-button during this time, a reaction which can be parameterised is activated (e.g. channel is switched on).

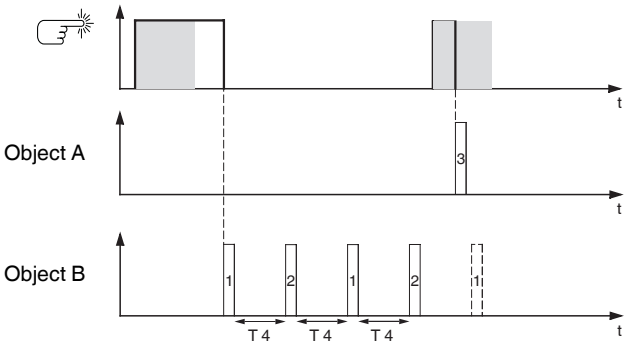


T 4 = Cycle time
T 5 = Monitoring time

Effect lighting

This example shows you how to program effect lighting, for example for a display window. A long push-button action switches between two different lighting scenes. A short push-button action stops the toggling and transmits a scene which switches off everything. The scene module of the push-button is used to retrieve the scene.

Number of objects = 2 (object A/B)
Object A/B = 1 byte contiuous 0-255
Object A: Direct action on operation = None (stops cyclical sending).
Action on release before the long operating time has elapsed = Sends value 1.
Action on achieving the long operating time = None (stops cyclical sending).
Action on release after achieving the long operating time = None (stops cyclical sending).
Value 1 = 3
Object B: Direct action on operation = None (stops cyclical sending).
Action on release before the long operating time has elapsed = None (stops cyclical sending).
Action on achieving the long operating time = None (stops cyclical sending).
Action on release after achieving the long operating time = Toggles cyclically, sends immediately, then cyclically.
Value 1 = 1
Value 2 = 2
Cycle time = e.g. 1 minute.
Connect object A and object B with the extension unit object of the scene function.



T 4 = Cycle time

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Object A/B	1 bit	Low	WCT	Transmit/ receive
Push-button X	Object A/B	2 bit	Low	WCT	Transmit/ receive
Push-button X	Object A/B	1 byte	Low	WCT	Transmit/ receive
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Sending edge commands - 2 byte

With this edge function, you can send a 2-byte object in floating point format or in integer format (with or without sign). A distinction is made between the normal edge function and the extended edge function:

- With the normal edge function, you can specify which actions should be carried out when a push-button is pressed, and which should be carried out when a push-button is released.
- With the extended edge function, you can also set the actions before and after the long button actuation period is completed.

Push-button X	
Parameter	**Setting**
Select push-button function	Edges with 2 byte values
Select edge function	**Normal (operate, release)**
	Extended (+ long and short operation)

Status indication

The status LED can:

- Be switched on or off continuously.
- Light up when pressed (for a long period), and go out when released.
- Flash.
- Display the status of the status feedback object.

Parameter	Setting
Trigger status LED	Switched on
	Switched off
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	Flashes
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF

Normal edge function

With the normal edge function, you can specify which actions should be carried out when a push-button is pressed, and which should be carried out when a push-button is released. These actions could include:

- Send value 1 or value 2:
  You can specify two values and set whether and how they are to be transmitted.
- Object sends its value:
  The object transmits the value which it currently has. Therefore you can, for example, transmit a value with the sending group address which was previously received via another group address.
- No action

Available values are the floating point value or integer values with/without sign.

Push-button X	
Parameter	**Setting**
Select edge function	**Normal (operate, release)**
Action on operation	**Sends value 1**
	Sends value 2
	Sends its value
	None
Action at release	Sends value 1
	Sends value 2
	Sends its value
	None

Push-button X - edges values	
Parameter	**Setting**
Object type value	**Floating point**
	Integer with sign (-32768...32767)
	Integer without sign (0 … 65535)
Value 1 = basis * factor	
Basis (possible values in brackets)	0,01, ... 327,68; **0,01**
Factor (0-2047)	0 - 2047, **1000**
Value 2 = basis * factor	
Basis (possible values in brackets)	0,01, ... 327,68; **0,01**
Factor (0-2047)	0 - 2047, **2000**
Value 1 (-32768 - 32767)	-32768...32767, **32767**
Value 2 (-32768 - 32767)	-32768...32767, **-32768**
Value 1 (0-65535)	0-65535, **65535**
Value 2 (0-65535)	0-65535, **0**

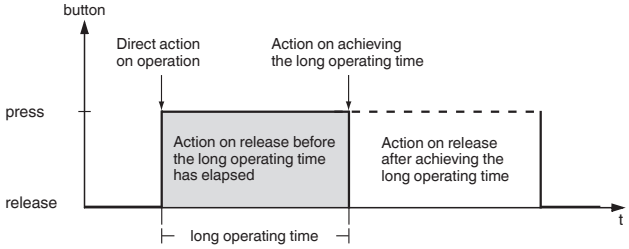
Extended edge function

With the extended edge function, you have a wider range of functions available. For example, you can set different actions for short and long presses of a push-button, both for when the push-button is pressed and for when it is released. You can also set a cycle time which can be parameterised for the object.

When parameterising, bear in mind that you need to set all four types of push-button operation (short/long press, pressing and releasing the button) in order to ensure that the push-button functions as required.

In order to read the object values, you may need to set the Read flags manually.

The following activation sequence chart shows the phases into which the pulse edge function is divided:



Push-button X	
Parameter	**Setting**
Select edge function	Extended (+ long and short operation)
Long operation time equals 100 ms * factor (4-250)	4 - 250, **6**
Direct action on operation Action on release before the long operating time has elapsed Action on achieving the long operating time Action on release after achieving the long operating time	**Sends value 1** Sends value 1 immediately and then cyclically Sends value 1 only cyclically Sets object value to value 1 (readable only) Sends value 2 Sends value 2 immediately and then cyclically Sends value 2 only cyclically Sets object value to value 2 (readable only) Sends its value Sends value 1, then value 2 after cycle time **None (stops cyclical sending)** No change
Cycle time = basis * factor	
Basis	0.1 s, 1 s, **1 min**, 1 h, 1 day
Factor (3-255)	3-255, **10**

A description of the actions is given below:

- Sends [value]:
  Transmits the current value and stops a cyclical transmission.
- Sends [value] immediately and then cyclically:
  If no cycle time is running, [value] is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, [value] is transmitted and a new cycle time is started.
- Sends [value] only cyclically:
  If no cycle time is running, [value] is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is **not** interrupted; [value] is transmitted after the current cycle time has elapsed, and a new cycle time is started.
- Sets object value to [value] (readable only)
  [value] is written into the object and is not transmitted. Any active cycle time is terminated.
- Sends its value:
  The current object value is transmitted. Any active cycle time is terminated.
- Sends [value A] and after cycle time [value B]:
  [value A] is transmitted immediately, and [value B] is transmitted after **one** cycle time, regardless of whether a cycle time is already running or not (staircase lighting timer function).
- None (stops cyclical sending):
  No action is carried out, and any active cycle time is stopped.
- No change:
  The current action remains unchanged (e.g. “sends value 1, then value 2 after a cycle time”).

Push-button X - edges, values	
Parameter	**Setting**
Object type value	**Floating point** Integer with sign (-32768...32767) Integer without sign (0 … 65535)
Value 1 = basis * factor	
Basis (possible values in brackets)	0,01, ... 327,68; **0,01**
Factor (0-2047)	0 - 2047, **1000**
Value 2 = basis * factor	
Basis (possible values in brackets)	0,01, ... 327,68; **0,01**
Factor (0-2047)	0 - 2047, **2000**
Value 1 (-32768 - 32767)	-32768...32767, **32767**
Value 2 (-32768 - 32767)	-32768...32767, **-32768**
Value 1 (0-65535)	0-65535, **65535**
Value 2 (0-65535)	0-65535, **0**

Communication objects

You can select the following communication objects:

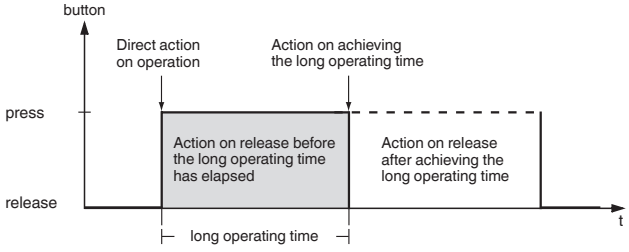
Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Value object A	2 byte	Low	WCT	Transmit/ receive
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Setting the parameters for the 8 bit slider

With this function you can program a push-button as a slider, allowing you to automatically increase or reduce object values cyclically (for example). The slider function can be parameterised with or without limit values for all four actions: when pressing/releasing and with a short or long button operating time (brief/long press).

Push-button X	
Parameter	**Setting**
Select push-button function	8 bit slider
Long operation time equals 100 ms * factor (4-250)	4 - 250, **6**

The following activation sequence chart shows the phases into which the slider function is divided:



Status indication

The status LED can:

- Be switched on or off continuously.
- Light up when pressed (for a long period), and go out when released.
- Flash.
- Display the status of the status feedback object.
- Display the status of the value object.

Parameter	Setting
Trigger status LED	Switched on
	Switched off
	From value object A
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	Flashes
	Flashes when value object A not equal to 0
	Flashes when value object A equals 0
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF

Push-button X slider	
Parameter	**Setting**
Slider function	**With limit values**
	Without limit values
Direct action on operation	Send value 1, then increase cyclically by step width
Action on release before the long operating time has elapsed	Send value 2, then reduce cyclically by step width
Action on achieving the long operating time	Increase current object value cyclically
Action on release after achieving the long operating time	Increase current object value once
	Reduce current object value cyclically
	Reduce current object value once
	Reverse slide direction and send cyclically
	Reverse slide direction and increase/decrease cyclically
	Stepwise to the limit values and back again
	Increase stepwise within limits
	Decrease stepwise within limits
	None (stops cyclical sending)
	no change
Value 1	0-255, **0**
Set step value	0-255, **10**
Value 2	0-255, **100**
Cycle time = basis * factor	
Basis	**0.1 s**, 1 s, 1 min, 1 h, 1 day
Factor (3-255)	3-255, **5**

A description of the actions is given below:

- Send value 1, then increase cyclically by step width:
  If no cycle time is running, value 1 is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, value 1 is transmitted and a new cycle time is started.
- Send value 2, then reduce cyclically by step width:
  If no cycle time is running, value 2 is transmitted immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, value 2 is transmitted and a new cycle time is started.
- Increase current object value cyclically:
  Increase the current object value cyclically by the parameterised step value.
- Increase current object value once:
  Increase the current object value once by the parameterised step value. Any active cycle time is terminated.
- Reduce current object value cyclically:
  Reduce the current object value cyclically by the parameterised step value.
- Reduce current object value once:
  Reduce the current object value once by the parameterised step value. Any active cycle time is terminated.
- Reverse slide direction and send cyclically:
  If no cycle time is running, the slide is pushed in the

opposite direction (of this push-button) and a new cycle time is started. If a cycle time is already running, it is interrupted, the slide is immediately pushed in the opposite direction (of this push-button) and a new cycle time is started. Cyclic transmission is stopped when the maximum/minimum value is reached.

- Reverse slide direction and increase/decrease cyclically:
  If no cycle time is running, the slide is pushed in the opposite direction (of this push-button) and a new cycle time is started. If a cycle time is already running, it is interrupted, the slide is immediately pushed in the opposite direction (of this push-button) and a new cycle time is started. Cyclic transmission is not stopped when the maximum/minimum value is reached. When an incrementing value reaches the maximum value, the value is set to the minimum value and cyclic transmission continues. When an decrementing value reaches the minimum value, the value is set to the maximum value and cyclic transmission continues.
- Stepwise to the limit values and back again:
  The limit values are approached by one step at a time. When a limit value is reached, the sliding direction is reversed for the next action.
- Increase stepwise within limits:
  The value is incremented by one step value at a time, within the limits. The limits are not exceeded; instead value 1 is sent again after the last possible step.
  Example: Value 1: “0”, value 2: “255”, step size: “100”; the following values are sent: 39%, 78%, 0%, 39%, 78%, 0%, etc.
- Decrease stepwise within limits:
  The value is reduced by one step value at a time, within the limits. The limits are not exceeded; instead value 2 is sent again after the last possible step.
  Example: Value 1: “0”, value 2: “255”, step size: „100“. The following values are sent: 100%, 61%, 22%, 100%, 61%, 22%, etc.
- None (stops cyclical sending):
  No action is carried out, and any active cycle time is stopped.
- No change:
  No action is carried out, and any active cycle time is continued.

Keeping within the limits and toggling to a new slide direction are only possible with local, on-site operation!

Example: Implementing a step dimmer with slider function

It is possible to dim a dimming actuator in several “steps” using a push-button. Push-button 1 is used as an 8 bit slider. The status LED can be controlled by the status feedback object of the dimmer.

“Push-button 1” tab:

Push-button function = 8 bit slider

“Push-button 1 slider” tab:

Slider function: “With limit values”

Direct action on rocker operation = Stepwise to the limit values and back again

Action on release, on or after achieving the long operating time = No change

Value 1 = 0

Step value = 51

Value 2 = 255

The cycle time is not required for this function.

Connect the push-button value object to the dimming actuator value object.

Every new press of the push-button sends a new dimming value, in the following steps: 20%, 40%, 60%, 80%, 100%, 80%, 60%, 40%, 20%, 0%, 20%, etc.

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Value object A	1 byte	Low	WCT	Transmit/ receive
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Retrieving scenes

Retrieving scenes by push-button does not access the internal scene module, but rather only accesses the bus externally via communication objects. If you therefore wish to retrieve scenes stored in the internal scene module using a push-button, you must connect the corresponding communication object with the extension unit object of the scene function.

There are two types of scene function:

- Normal
- Extended

Push-button X	
Parameter	**Setting**
Select push-button function	Scene
Select scene function	**Normal (short = recall/long = save)**
	Extended

Status indication

The status LED can:

- Be switched on or off continuously.
- Light up when pressed (for a long period), and go out when released.
- Flash.
- Display the status of the status feedback object.
- Display the status of object A/B.

Parameter	Setting
Trigger status LED	Switched on
	Switched off
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	Flashes
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF
	From object A
	From object B
	Flashes when object A not equal to 0
	Flashes when object B not equal to 0

Normal scene function

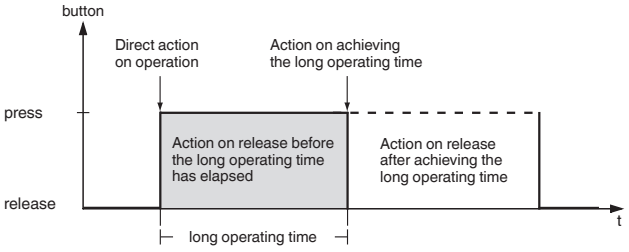
With the normal scene function, a scene is retrieved by a brief push-button action and a long push-button action is used to save a scene. You merely have to set the time after which a push-button action is identified as being long, together with the triggering of the status LED and the scene address.

Push-button X	
Parameter	**Setting**
Select scene function	Normal (short = recall/long = save)
Long operation time equals 100 ms * factor (4-250)	4 - 250, **6**
Scene address (0-63)	0-63, **0**

Extended scene function

With the extended scene function, you can set different actions for short and long presses of a push-button, both for when the push-button is pressed and for when it is released. You can also set a cycle time which can be parameterised for each object.

The following activation sequence chart shows the phases into which the scene function is divided:



Push-button X	
Parameter	**Setting**
Select scene function	Extended
Long operation defined as 100 ms * factor (4-250)	4 - 250, **30**
Number of objects	**one**
	two

Push-button X - scene object A/B	
Parameter	**Setting**
Direct action on operation	Sends value 1
Action on release before the long operating time has elapsed	Sends value 2
Action on achieving the long operating time	Toggles
Action on release after achieving the long operating time	Toggles cyclically, sends immediately, then cyclically
	Sends value 1, then value 2 after a cycle time
	None (stops cyclical sending)
	No change
Value 1 Scene address (0-63)	0-63, **0**
Value 1 to retrieve/save the scene	**Retrieve**
	Save
Value 2 Scene address (0-63)	0-63, **0**
Value 2 to retrieve/save the scene	Retrieve
	Save
Cycle time = basis * factor	
Basis	0.1 s, **1 s**, 1 min, 1 h, 1 day
Factor (3-255)	3-255, **10**

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Object A	1 byte	Low	WCT	Transmit/ receive
Push-button X	Object B	1 byte	Low	WCT	Transmit/ receive
Push-button X	Status feedback object	1 bit	Low	WC	Receive

Change setpoint

You can change the setpoint for the integrated room temperature control unit by pressing a push-button. Whether this change affects the current operation mode or all operation modes depends on the setting you make on the "Control general - On what the setpoint adjustment has an effect" tab.

You can also trigger setpoint adjustment using an external push-button or you send the values to the bus in order to change the setpoint for another push-button. There is a 1 bit object available for increasing the setpoint and a 1 bit object for reducing it.

Push-button X	
Parameter	**Settings**
Select push-button function	Setpoint adjustment

Status feedback

The status LED can:

- Be switched on or off continuously.
- Light up when pressed (for a long period), and go out when released.
- Flash.
- Display the status of the setpoint adjustment object.
- Display the status of the status feedback object.

Parameter

Parameter	Settings
Setpoint adjustment	**Increase setpoint**
	Reduce setpoint
Set step width	**0.5 K**
	1 K
Trigger status LED	Switched on
	Switched off
	From the setpoint adjustment object
	From status feedback object
	Operation = ON / release = OFF
	Long operation = ON / release = OFF
	Flashes
	Flashes when obj. setpoint adjustm. not equal to 0
	Flashes when obj. setpoint adjustment equals 0
	Flashes when status feedback object equals 1
	Flashes when status feedback object equals 0
	Operation = flash / release = OFF
	Long operation = flash / release = OFF

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Push-button X	Increase setpoint	1 bit	Low	WCT	Transmit/ receive
Push-button X	Reduce setpoint	1 bit	Low	WCT	Transmit/ receive

Toggle operation modes

Use this function to toggle operation modes with a single push-button.

Push-button X	
Parameter	**Settings**
Select push-button function	Operation mode

Parameter	Settings
Number of operation modes between which you want to toggle.	One Two Three **Four**
Operation mode 1	**Comfort extension operation** Comfort operation Standby operation Night operation
Operation mode 2	Comfort extension operation **Comfort operation** Standby operation Night operation
Operation mode 3	Comfort extension operation Comfort operation **Standby operation** Night operation
Operation mode 4	Comfort extension operation Comfort operation Standby operation **Night operation**
Trigger status LED	Switched on Switched off **Operation = ON / release = OFF** Flashes Operation = flash / release = OFF Switched on in comfort extension operation Switched on in comfort operation Switched on in standby operation Switched on in night operation Switched on in frost/heat protection operation

Setting the parameters for the disable function for push-buttons

You can use the disable function to disable the push-buttons in three different ways:

1.For each push-button separately

2.All push-buttons function like a predefined master push-button

3.Toggle between two local scenes.

You can determine whether disabling should occur when disable object = 0 or when disable object = 1.

When a disable function is activated via the disable object, all current push-button functions (including cyclical actions) are suppressed.

Disable function for push-buttons	
Parameter	**Setting**
Apply disable function	**No**
	Yes
Set disable function	
Execute disable function	At object value 0
	At object value 1
Type of blocking	Set separately for each push-button
	All push-buttons function like master
	Toggle between two scenes (scene addresses)

For each push-button separately

With this function you can disable each push-button individually. When a push-button is disabled, it does not execute a function when pressed.

Disable function for push-buttons	
Parameter	**Setting**
Type of blocking	Set separately for each push-button
Push-button 1 disable Push-button 2 disable Push-button 3 disable Push-button 4 disable	**Yes** No
Include menu buttons in the lock	**Yes**
	No

All push-buttons function like master

You can use this function to specify one push-button as a master push-button. When any push-button is pressed, the function that was parameterised for the master key is carried out.

Disable function for push-buttons	
Parameter	**Setting**
Type of blocking	All push-buttons function like master
Master push-button =	Push-button 1
	Push-button 2
	Push-button 3
	Push-button 4
Include menu buttons in the lock	**Yes**
	No

Toggle between two scenes (scene addresses)

With this action you can toggle between two scenes which are parameterised in the scene module. When any push-button is pressed, one or the other scene is retrieved in alternation.

The scene addresses entered must be known to the push-button's internal scene module, and must be identical to the scene addresses in the module. The scene addresses entered with this function are not transmitted to the bus.

Disable function for push-buttons	
Parameter	**Setting**
Type of blocking	Toggle between two scenes (scene addresses)
First scene address	0-63, **0**
Second scene address	0-63, **1**
Include menu buttons in the lock	**Yes**
	No

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Disable function	Locking object	1 bit	Low	WC	Receive

Setting the parameters for scenes in the scene module

The push-button is fitted with its own scene module, which enables you to save up to eight scenes permanently. The saved scenes can be overwritten if you have parameterised a release for this purpose.

The entire scene function is controlled via the extension object (1 byte). The following objects are also available for sending scene values to the bus:

- An object for programming release
- Eight objects for values with 1 bit, 2 bit and 1 bytes
- One object (Actuator group 7) for values with 2 bytes

You can set the time between the actuator read telegrams. This makes sense, e.g. when the anticipated response can last a long time (line coupler, area coupler).

If a read request is lost, or is not responded to, the current object value is saved in the scene (either through a read request, or written via an output). To check the correct saving procedure, you should retrieve the scene last saved on the push-button. If this remains unchanged, the individual saving procedure has been completed free of errors. If there is a difference, this means that a read request was not responded to correctly.

If the push-button works through a scene, and a further scene is retrieved, the current process is interrupted and the last retrieved scene is worked through.

Scene module	
Parameter	**Setting**
Apply scene module	**No**
	Yes
Save scenes	**Yes**
	Yes, if enable object = 1
	No
Time between 2 read telegrams 100 ms * factor (2-255)	2-255, **10**

Specifying scene actuator groups

In this card, you can specify the data type of the eight actuator groups. Actuator group 7 is a special group which allows you to transmit values with 16 bits.

Scene actuator groups	
Parameter	**Setting**
Object types of the actuator groups	
Actuator group 1 Actuator group 2 Actuator group 3 Actuator group 4 Actuator group 5 Actuator group 6 Actuator group 8	**Switch object** Value object (8 bit in steps) Value object (8 bit stepless) Priority object
Actuator group 7 (also 16 bit possible)	**Switch object** Value object (8 bit in steps) Value object (8 bit stepless) Priority object Value object (16 bit without sign) Value object (16 bit with sign) Value object (16-bit floating point value)

Specifying scene addresses and values

For each scene, you specify the scene address via which the scene on the extension object should be retrieved. You also specify the time between the individual scene telegrams.

Make sure that you always enter unique scene addresses for this device, i.e. no scene address should be allocated more than once.

Scene X	
Parameter	**Setting**
Scene address (0-63)	0-63
Time between scene telegrams 100 ms * factor (2-255)	2-255, **10**

Finally, specify the actuator groups and their values for this scene. These only remain valid up to the first time the scene is saved.

The value range which can be set depends on the data type set for the “scene actuator groups”.

Scene X - values	
Parameter	**Setting**
Value 1 sending Value 2 sending Value 3 sending Value 4 sending Value 5 sending Value 6 sending Value 8 sending	ON telegram OFF telegram **No telegram** 0 % - 100 % 0-254 Switch on with priority (11) Switch off with priority (10) Remove priority (00)

Scene X - values	
Parameter	**Setting**
Value 7 sending	ON telegram
	OFF telegram
	No telegram
	0 % - 100 %
	0-254
	Switch on with priority (11)
	Switch off with priority (10)
	Remove priority (00)
	Send telegram
Value 7 sending (0-65535)	0-65535, **65535**
Value 7 sending (-32768-32767)	-32768...32767, **32767**
Value 7 = basis * factor	
Basis (possible values in brackets)	0,01...327,68, **0,01**
Factor (0-2047)	0-2047, **1000**

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Save scenes	Enable object	1 bit	Low	WC	Receive
Scene function	Extension object	1 byte	Low	WC	Receive
Switching	Actuator group 1-8	1 bit	Low	WCT	Transmit/ receive
Transmit value	Actuator group 1-8	1 byte	Low	WCT	Transmit/ receive
Transmit value	Actuator group 7	2 byte	Low	WCT	Transmit/ receive
Priority operation	Actuator group 1-8	2 bit	Low	WCT	Transmit/ receive

Activating the time control

Two time-switch channels are available, each with four programmable switch times, in order to trigger actions with minute-by-minute precision.

The push-button can be linked to an external clock via

- The date and time object
- The object for requesting time
- The object for labelling a working day / holiday

This link synchronises the internal clock.

After a reset, the time is set to 0:00, and the time symbol in the display flashes. If no time synchronisation is completed within 24 hours, switching commands can continue to be carried out or suppressed, depending on the setting. In this case, the time symbol also flashes.

Time control	
Parameter	**Settings**
Use time control	Yes
	No

Request time synchronisation via the bus

After a download or when the bus voltage is switched on, the push-button can transmit a telegram to the bus to request the current time and date. This synchronises the time and date in the push-button. Make the following settings:

① "General" tab: For the "Receive date and time" parameter, select whether the data is received in one or two communication objects.

② "Time control" tab: Set the "Request time synchronisation via the bus" parameter to "Yes".

③ "Time control" tab: For the "Behaviour when synchronisation fails" parameter, select whether the switching commands are carried out or suppressed.

④ Connect the objects "Time object input", "Date object input", "Date/time object input" and "Request time" to the corresponding objects of a year time switch.

The function "Request time synchronisation via the bus" only works in conjunction with an appropriate year time switch.

General	
Parameter	**Settings**
Receive date and time	In one communication object
	In two communication objects

Time control	
Parameter	**Settings**
Request time synchronisation via the bus	**Yes**
	No

Time control	
Parameter	**Settings**
Behaviour when synchronisation fails	**Switching commands are still carried out**
	Switching commands are suppressed

Use push-button as master clock

Set a push-button as the master clock in order to synchronise the time for other push-button (slave clocks).

Make the following settings:

① "General" tab for the master clock: Set the parameter "Send time cyclically" to the value "Every minute", "Every hour" or "Daily".

② "General" tab for the master clock: Select the format for the time that is sent.

③ "General" tab for the slave clocks: Set the parameter "Send time cyclically" to the value "No".

④ Connect the "Date/time object output" objects to each other, or the "Time object output" objects, as the case may be.

⑤ Set the time on the master clock.

The time is synchronised on all the other push-buttons.

This only applies to transmitting the time. The date is not transmitted to the bus and cannot be set with the menu buttons.

General	
Parameter	**Settings**
Send time cyclically	**No**
	Every minute
	Every hour
	Daily
Format of sent time	**Time format (3 byte)**
	Date/time format (8 byte)

Parameters for the switching times

Time control is deactivated by default. Furthermore, it is initially not possible to specify the switching times via the control menu (-:- is displayed)..

If you want to use time control, you have to overwrite the switching times once with an ETS download.

The following switching times are set by default:

- Switching time 1 = 06:00
- Switching time 2 = 12:00
- Switching time 3 = 18:00
- Switching time 4 = 22:00

The switching times are not carried out until the time has been set once via the control menu or via the time object.

When a change is made from a "holiday" to a "working day" or vice-versa:
The push-button carries out the last switch times that are programmed up to the current time, taking into account the new state.

Time control	
Parameter	**Settings**
Number of time switch channels	**1**
	2
Time channel 1 / Time channel 2	
Number of switching times	1
	2
	3
	4
Actuator group	**Switch object**
	Value object (8 bit in steps)
	Value object (8 bit continuous)
	Priority object
	Value object (16-bit integer without sign)
	Value object (16-bit integer with sign)
	Value object (16-bit floating point value)

Time channel X - switching time X	
Parameter	**Settings**
Overwrite switching times	**Yes**
	No
Switching time X	
Hour (0-23)	0-23, **6, 12, 18, 22**
Minute (0-59)	0-59, **0**
Execute switch time	**On working day**
	On holiday
	Always
Value	ON telegram
	OFF telegram
	100 %, 90 %, 80 %, ..., 0 %, 25 %, 75 %
	0-255, **255**
	Switch on with priority (11)
	Switch off with priority (10)
	Remove priority (00)
	0-65535, **65535**
	-32768 ... 32767, **32767**
Value = base * factor	
Base (possible values in brackets)	0.01, ... 327.68, **0.01**
Factor (0-2047)	0-2047, **1000**
Select scene address internally	0-63, **No**

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Time control	Time object input	3 byte	Low	WC	Receive
Time control	Date object input	3 byte	Low	WC	Receive
Time control	Date/time object input	8 byte	Low	WC	Receive
Time control	Time object output	3 byte	Low	CT	Transmit
Time control	Date/time object output	8 byte	Low	CT	Transmit
Time control	Time request	1 bit	Low	CT	Transmit
Time control	Working day/holiday	1 bit	Low	WCT	Transmit/ receive
Time control	Switch object A/B	1 bit	Low	WCT	Transmit/ receive
Time control	Value object A/B	1 byte	Low	WCT	Transmit/ receive
Time control	Value object A/B	2 byte	Low	WCT	Transmit/ receive
Time control	Priority object A/B	1 byte	Low	WCT	Transmit/ receive

Communication object "Working day/holiday"

The value 1 (1 bit) switches the clock to "Working day" mode. The value 0 (1 bit) switches the clock to "Holiday" mode.

If the communication object "Working day/holiday" does not have a group address then the default value "0" is used. This means that the clock is in the "Holiday" mode..
Only the switching times parameterised for "On holiday" or "Always" are executed.

This object should be controlled by an appropriate year time switch.

Getting to know and parameterising the room temperature control unit

To activate the room temperature controller, you must set the "Use control" parameter to "Yes" in the "Control general" tab. Once you have done this, further configurable tabs are available. As for all parameters, the recommended settings are already specified here. However, you must check all parameters to make sure that they are set correctly and appropriately for your installation's local conditions.

Control general	
Parameter	**Settings**
Use control	Yes
	No

How the room temperature controller functions

There are many factors that can affect room temperature. The task of the control is to detect the actual temperature constantly, and to ensure that the heating or cooling system receives new information accordingly. The heating or cooling system converts this information and adjusts the room temperature to the preconfigured setpoints.

The actual temperature is continuously measured by the temperature sensor integrated into the push-button. However, you can also measure the temperature via an external sensor and transfer it to the controller via the bus, which then takes it fully or partially into account when assessing the actual temperature.

The controller can control the connected heating/cooling systems via corresponding switch telegrams or continuous correcting variables. In this way, both PI controls and 2-step controls can be parameterised.

Four operation modes (comfort, standby, night and frost/heat protection) for which setpoints can be set in each case are available for differentiated control with different requirements.

Additional functions of the room temperature control unit are comfort extension, shared/separate correcting variable output, selection of the operation mode after reset, offset of the setpoint temperatures, 1 bit/1 byte status objects, taking into account a temperature which has been measured separately, temperature drop detection, and valve protection.

Setpoints and operation modes

Four operation modes are available to help you control the room temperature:

- Comfort mode
  Controls the room temperature when the room is being used.
- Standby mode
  Lowers temperature slightly when the room is not being used.
- Night operation
  Lowers temperature significantly, e.g. at night or over the weekend.
- Frost/heat protection
  Automatically switches on the heating or cooling when adjustable temperature threshold values are not reached or are exceeded.

The additional "comfort extension" operation mode acts in the same way as the comfort mode, but is exited automatically after a time period that you can set.

You can switch back and forth between these operation modes in different ways:

- Via the communication objects, by using the time control on the push-button, for example
- Via the control menu

You can specify a setpoint for each operation mode. When changing the operation mode, the relevant setpoint for continued room temperature control is used. The setpoints for all operation modes, except for frost/heat protection, can be manually altered within adjustable limits using the control menu on the push-button, or can be adjusted via the "Setpoint adjustment input" object. You can also specify whether setpoint adjustment affects

- The current operation mode only or
- All operation modes

Setpoint adjustment affects current operation mode only

In this setting, the setpoint temperature of the current operation mode is changed. You can select whether or not the setpoint adjustment is retained after the operation mode is switched.

The operation mode switch via frost/heat protection does not affect the setpoint adjustment.

Setpoint adjustment maintained after change in operation mode = No

Comfort = 2 °C Standby = 0 °C Night = 0 °C	Comfort -> Standby ->Comfort	Comfort = **0 °C** Standby = 0 °C Night = 0 °C
Comfort = 2 °C Standby = 0 °C Night = 0 °C	Comfort -> Frost protection- >Comfort	Comfort = **2 °C** Standby = 0 °C Night = 0 °C

Setpoint adjustment maintained after change in operation mode = Yes

Comfort = 2 °C Standby = 0 °C Night = 0 °C	Comfort -> Standby ->Comfort	Comfort = **2 °C** Standby = 0 °C Night = 0 °C

You can specify the setpoint adjustment directly via the "Setpoint adjustment input" object and the control menu. By comparison, you use the "Current setpoint temperature input" object to specify a new setpoint temperature. Setpoint adjustment is determined here by the difference between the current setpoint temperature and the object value.

Example 1

Heating, current operation mode = standby
Limits of setpoint adjustment = +3 K/-3 K
Setpoint adjustment: "Setpoint adjustment input" object = +3 °C

Initial status	Result
Comfort = 21 °C	Comfort = 21 °C
Standby = 19 °C	Standby = **22 °C**
Night = 17 °C	Night = 17 °C
Frost protection = 7 °C	Frost protection = 7 °C

Example 2

Heating, current operation mode = comfort
Limits of setpoint adjustment = +5 K/-5 K
New set value: object "Current setpoint temperature input" = +30 °C

Initial status	Result
Comfort = 21 °C	Comfort = **26 °C**
Standby = 19 °C	Standby = 19 °C
Night = 17 °C	Night = 17 °C
Frost protection = 7 °C	Frost protection = 7 °C

Setpoint adjustment affects all operation modes

In this setting, you not only change the setpoint temperature for the current operation mode, you change all the setpoint temperatures in the same way and at the same time. The only setpoint temperatures that are not affected are those for the frost/heat protection. These operation modes also specify the limits of the setpoint adjustment. It is therefore not possible to set setpoint temperatures lower than the frost protection or higher than the heat protection.

You can specify the setpoint adjustment directly via the "Setpoint adjustment input" object and the control menu. By comparison, you use the "Current setpoint temperature input" object to specify a new setpoint temperature. Setpoint adjustment is determined here by the difference between the current setpoint temperature and the "reference setpoint for calculating the setpoint adjustment".

Example 1

Cooling/heating
Limits of setpoint adjustment = +3 K/-3 K
Setpoint adjustment: "Setpoint adjustment input" object = +5 °C

Initial status	Result
Cooling:	Cooling:
Heating protection = 35 °C	Heat protection = 35 °C
Night = 28 °C	Night = **31 °C**
Standby = 26 °C	Standby = **29 °C**
Comfort = 24 °C	Comfort = **27 °C**
Heating:	Heating
Comfort = 21 °C	Comfort = **24 °C**
Standby = 19 °C	Standby = **22 °C**
Night = 17 °C	Night = **20 °C**
Frost protection = 7 °C	Frost protection = 7 °C

Example 2

Cooling/heating
Limits of setpoint adjustment = +10 K/-10 K
Setpoint adjustment: "Setpoint adjustment input" object = +20 °C

Initial status	Result
Cooling:	Cooling:
Heating protection = 35 °C	Heat protection = 35 °C
Night = 28 °C	Night = **35 °C**
Standby = 26 °C	Standby = **33 °C**
Comfort = 24 °C	Comfort = **31 °C**
Heating:	Heating
Comfort = 21 °C	Comfort = **28 °C**
Standby = 19 °C	Standby = **26 °C**
Night = 17 °C	Night = **24 °C**
Frost protection = 7 °C	Frost protection = 7 °C

Example 3

Cooling/heating
Limits of setpoint adjustment = +3 K/-3 K
New set value: object "Current setpoint temperature input" = 24 °C
Reference setpoint for calculating the setpoint adjustment = 21 °C
Calculated setpoint adjustment = +3 °C

Initial status	Result
Cooling:	Cooling:
Heating protection = 35 °C	Heat protection = 35 °C
Night = 28 °C	Night = **31 °C**
Standby = 26 °C	Standby = **29 °C**
Comfort = 24 °C	Comfort = **27 °C**
Heating:	Heating
Comfort = 21 °C	Comfort = **24 °C**
Standby = 19 °C	Standby = **22 °C**
Night = 17 °C	Night = **20 °C**
Frost protection = 7 °C	Frost protection = 7 °C

The active operating state of the controller is determined by the states of the communication objects: "Comfort", "Night reduction", "Frost/heat protection" and "Dewpoint alarm".

The highest priority when calculating the setpoints is the dewpoint alarm. If it occurs, heating continues to be possible but cooling is deactivated ("0" to the controller output). The dewpoint alarm is terminated when its communication object is set to "0".

After a reset, the operation mode you preconfigured is active. The corresponding setpoints then also apply. If the setpoint that was set is changed via the control menu and the value is higher or lower than the set limit, an acoustic signal can inform you of this (you can set this on the "Signal function" tab).

When a setpoint adjustment is received via the bus, the controller checks whether it lies within the parameterised limits, and if necessary, adjusts it to the corresponding limits.

Comfort mode

The symbol in the display indicates that the controller is in the "Comfort" operation mode. This operation mode is used to control the room temperature when the room is being used.

Comfort mode is active

- If (for example a presence detector) reports that someone is present via the "Comfort input" object. An external push-button is also an option.
- If you select the operation mode "Comfort" in the control menu.
- If you activate a push-button for which the push-button function operation mode = comfort mode was parameterised.

Ending the comfort mode via the "Comfort input" object (value = 0) results in the standby or night mode being activated. This is useful as an office application for central resetting, for example.

You can set the controller to automatically switch to this state after a reset or a download.

Control general	
Parameter	**Settings**
Operation mode after reset	Comfort operation **Standby operation** Night operation Frost/heat protection Last operation
Operation mode after download	**Comfort operation** Standby operation Night operation Frost/heat protection

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Control	Comfort input	1 bit	Low	WC	Receive
Control	Comfort output	1 bit	Low	CRT	Transmit

Comfort extension (symbol flashes)

Comfort extension is indicated by the flashing symbol. The comfort extension operation mode is largely the same as the comfort mode. However, the comfort extension is exited automatically after a time period that you can set. It temporarily suppresses the night operation mode when the room is used for longer during the evening, for example.

You can access the comfort extension via:

- The control menu,
- A push-button (operation mode push-button function = comfort extension operation) or
- The bus (object "Comfort extension")

The parameterised time for the comfort extension runs to an end and can then be restarted by activating the comfort extension again in the control menu, for example.

If you select the comfort extension via the control menu, the symbol flashes.

The comfort extension is terminated:

- When the parameterised time has elapsed.
- When the "Night operation", "Comfort" or "Standby" operation mode is selected in the control menu.

You can parameterise the controller so that, once the comfort extension has ended, the controller:

- Switches to standby mode.
- Switches to night operation.
- Switches to the operation mode specified by the current value. A precondition for this is that the parameter "Termination of comfort extension via objects" is set to "No".

Control general	
Parameter	**Settings**
Duration of comfort extension	None Test mode (1 min) 30 min to 4.0 h, **1.0 h**
Termination of comfort extension via objects* *Objects: comfort, standby, operation mode	Yes **No**
Operation mode after comfort extension	Standby operation Night operation **Current object values**

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Control	Comfort extension input	1 bit	Low	WC	Receive
Control	Comfort extension output	1 bit	Low	CRT	Transmit

Standby operation ⌂

The ⌂ symbol in the display indicates that the controller is in the "Standby" operation mode. This operation mode enables you to reduce or increase the room temperature to a parameterised level as soon as the room is no longer in use. A brief heating period or cooling period is triggered by the low difference in temperature to the comfort mode.

Standby operation is activated

- If all the operation mode objects equal "0", i.e. the operation modes "Dewpoint alarm", "Night reduction", "Frost/heat protection" or "Comfort" are inactive.
- If you select the operation mode "Standby" in the control menu.
- If you activate a push-button for which the push-button function operation mode = standby mode was parameterised.

You can set the controller to automatically switch to this state after a reset or a download.

Control general	
Parameter	**Settings**
Operation mode after reset	Comfort operation **Standby operation** Night operation Frost/heat protection Last operation
Operation mode after download	**Comfort operation** Standby operation Night operation Frost/heat protection

Night operation ☽

The ☽ symbol in the display indicates that the controller is in the "night operation" mode. This operation mode enables you to reduce or increase the room temperature to a greater extent during the night or over the weekend. In this operation mode, you use a "1" telegram to switch via the "Night reduction" object.

The night operation mode is active when the comfort object is set to "0" and

- the "night reduction input" object is set to "1", or
- When you select "Night operation" mode in the control menu or
- If you push a push-button that was parameterised for the push-button function operation mode = night operation.

Night operation ends

- When the "Night reduction input" object is set to "0", or
- When the Comfort extension", "Comfort" or "Standby" operation mode is selected in the control menu.

You can set the controller to switch automatically to this state after a reset or after a download.

Control general	
Parameter	**Settings**
Operation mode after reset	Comfort operation **Standby operation** Night operation Frost/heat protection Last operation
Operation mode after download	**Comfort operation** Standby operation Night operation Frost/heat protection operation

Frost/heat protection

The parameterised values for frost protection (e.g. +7°C) or heat protection (e.g. +35°C) are set as new setpoints with a "1" telegram to the "Frost/heat protection input" object. This prevents the room from becoming overheated or the heating from freezing. A "0" telegram terminates the "frost/heat protection" and the new operation mode is set again. The operation mode is the result of current information from the objects "Comfort extension input", "Comfort input" and "Night reduction input". If no change occurs, the previous operation mode is set. This does not apply when the "Dewpoint alarm" operation mode is also active.

On the "Display" tab, you can specify whether the user is allowed to adjust the operation mode during frost/ heat protection. "No" is the default setting here.

Dewpoint alarm

The dewpoint operation mode is used to switch off the cooling in all circumstances. A "1" telegram to the "Dewpoint alarm" object switches off the cooling when there is condensation in the cooler. This operation mode has the highest priority. A "0" telegram terminates the "dewpoint alarm" and the new operation mode is set. The operation mode is the result of current information from the objects "Comfort input" and "Night reduction input". If no change occurs, the previous operation mode is set.

Locking object

You can use the control's locking object to activate the frost/heat protection with priority.

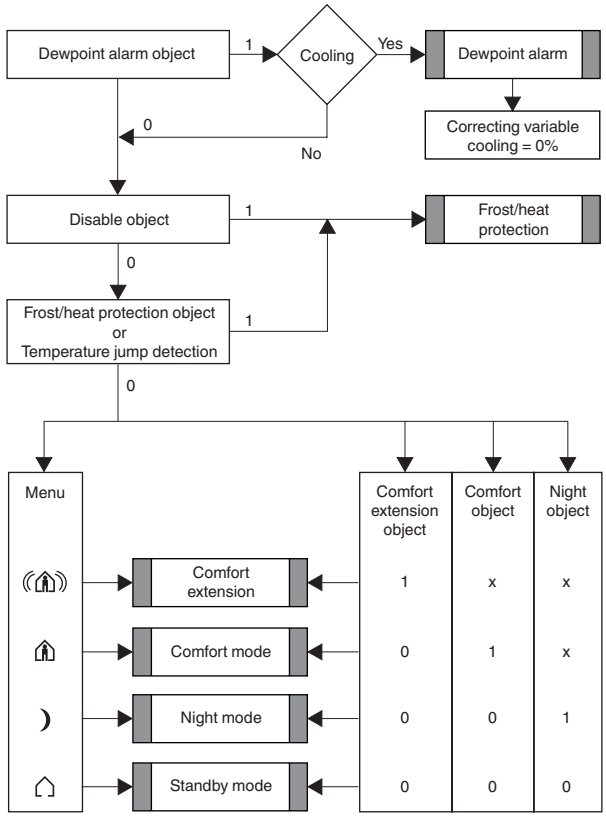
Application: A push-button at the building's exit enables you to switch the system to the away setting, for example. In this case, the heating only comes on during extreme cold. If the window contacts are monitored and the windows are only closed after the away setting is activated, the frost/heat protection remains active anyway.

Toggling between operation modes via 1 bit

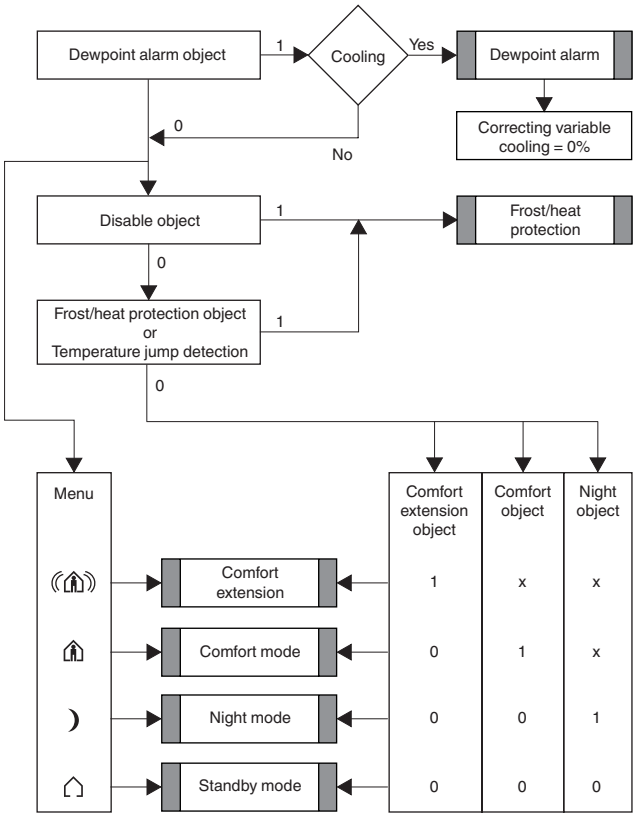
The following shows toggling between operation modes via 1 bit. Different processes occur during toggling between operation modes, depending on the parameter "Adjust frost/heat protection operation mode".

Display	
Parameter	**Settings**
Adjust operation mode during frost / heat protection	Yes **No**

Adjust operation mode during frost / heat protection = No



Adjust operation mode during frost / heat protection = Yes



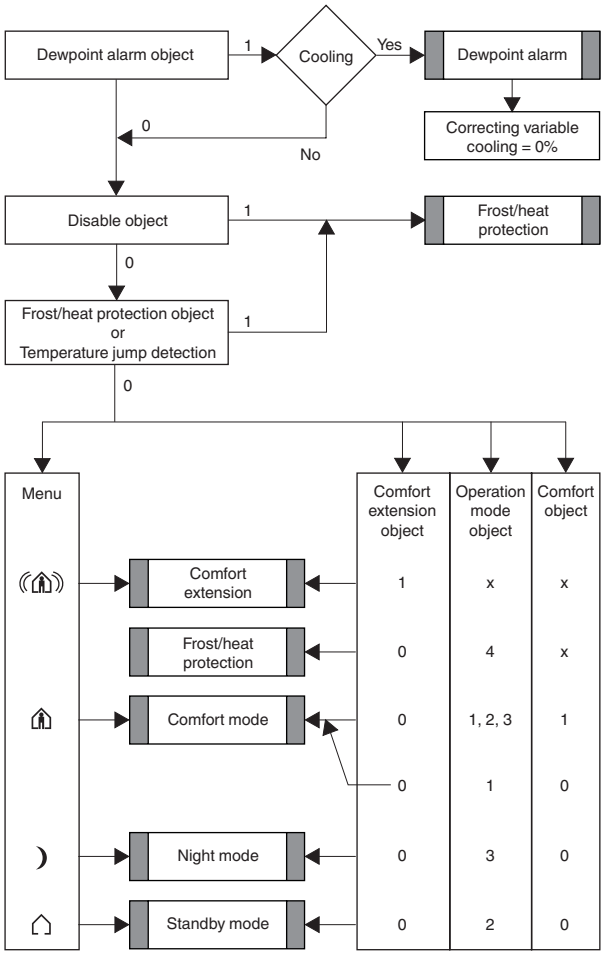
Toggling between operation modes via 1 byte

The following shows toggling between operation modes via 1 byte. Different processes occur during toggling between operation modes, depending on the parameter "Adjust frost/heat protection operation mode".

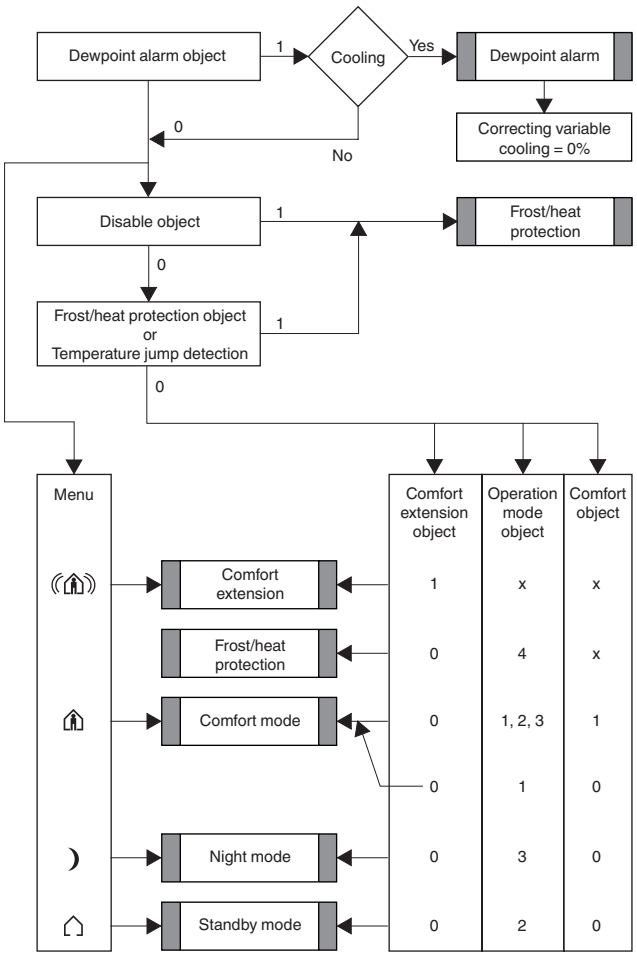
Display	
Parameter	**Settings**
Adjust operation mode during frost / heat protection	Yes **No**

- "4" = Frost/heat protection
- "3" = Night reduction
- "2" = Standby
- "1" = Comfort

Adjust operation mode during frost / heat protection = No



Adjust operation mode during frost / heat protection = Yes



Heating and cooling

Heating

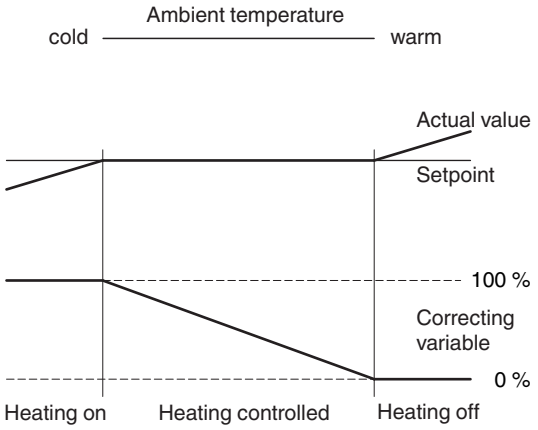
In the heating control mode, the current actual temperature is compared with the current setpoint temperature. If the actual temperature lies below the setpoint temperature, this difference is counteracted by issuing a correcting variable that does not equal "0".

Heating with constant correcting variables (e.g. EMO valve drive):

- Radiator/convector warm water heating
- Underfloor warm water heating
- 2-circuit underfloor warm water heating
- Air convectors

Heating with switching correcting variables (e.g. switch actuator):

- Electric convector
- Night storage heating
- Ceiling heating



Cooling

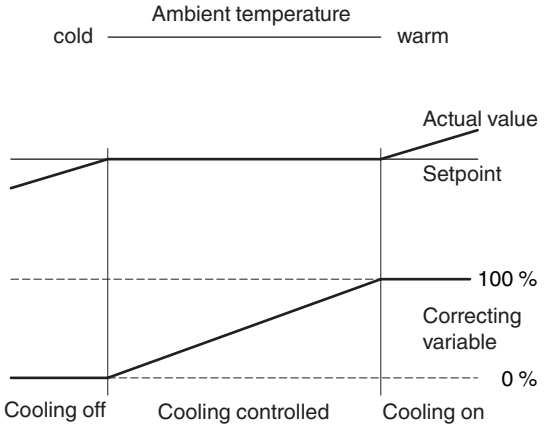
In the cooling control mode, the current actual temperature is compared with the current setpoint temperature. If the actual temperature is more then the setpoint temperature, this difference is counteracted by issuing a correcting variable that does not equal "0".

Cooling with constant correcting variables (e.g. EMO valve drive):

- Cooling ceiling
- Air convectors

Cooling with switching correcting variables (e.g. switch actuator):

- Cooling ceiling
- Air convectors



Heating and cooling

You can use the parameter "Switch between heating and cooling" to set whether heating and cooling are

- Set automatically by the controller or
- Set externally via the "Heating/cooling" object

If you select the "Heating/cooling" object, you can only force the controller into the heating or cooling mode via the object value.

If you have set the parameter "Switch between heating and cooling" to "Externally (via Heating/cooling object)" then after a download or restoration of bus voltage, a read request is transmitted to the bus by the "Heating/cooling" object.

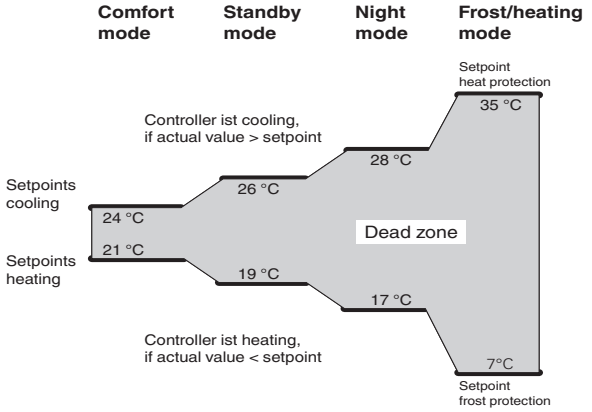
- If the object does not receive any status feedback after a download, the controller switches to "Heating" and the object sends a "1" to the bus.
- If the object does not receive any status feedback after bus voltage recovery, the controller switches to the last mode.
- If the object receives a status feedback, the object's operation mode is set.

If automatic mode was selected, the controller decides which control mode is suitable based on the parameterised setpoints, the insensitive zone and the current actual temperature.

The insensitive zone

The insensitive zone prevents the controller from switching constantly between heating and cooling. For example, if a heater is used for heating, it has sufficient thermal energy after the valve has been closed to continue to heat the room above the setpoint temperature. If you have configured the heating and cooling setpoint temperatures to be the same, the insensitive zone is set to "0 K". The air conditioning unit cools immediately because the setpoint for cooling has been exceeded. The procedure repeats itself again and again. This error is displayed as "Er 2" in the display.

Another error occurs if the heating setpoint was set higher than the cooling setpoint. This is displayed as error message "Er 2" after a reset. The control remains inactive until you rectify the error in the ETS and re-load the parameters.

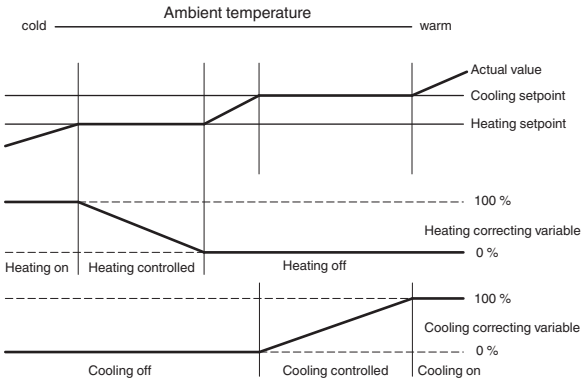


Heating and cooling with constant correcting variables (e.g. EMO valve drive):

- 2-pipe fan coil
- 4-pipe fan coil (with external switching between heating and cooling)
- 4-pipe fan coil (with automatic switching between heating and cooling)
- 1-circuit air conditioned ceiling
- Cooling ceiling with combined warm water heating
- Cooling ceiling with combined underfloor heating
- Variable air volume

Heating and cooling with switching correcting variables (e.g. switch actuator):

- Cooling ceiling
- Air convectors



Adjust the setpoint ranges for heating and cooling together

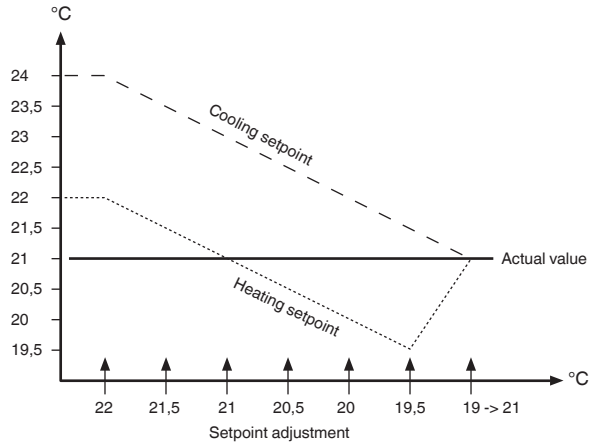
The difference between the two setpoints (heating and cooling) is interpreted as the insensitive zone.

Example:

The upper and lower setpoint adjustment is 3 K respectively.

Actual value = 21 °C; Heating setpoint = 22 °C; Cooling setpoint = 24 °C, this results in an insensitive zone of 2 K.

If you now adjust the setpoint temperature downwards using the display, then the following values are displayed: 22,0; 21,5; 21,0; 20,5; 20,0; 19,5; 21,0.

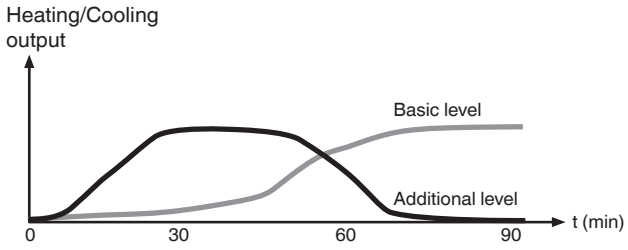


The jump from 19.5 to 21.0 can be explained by the fact that the cooling setpoint is relevant now because the actual temperature is more than or the same as the cooling setpoint temperature. Depending on the setting, this setpoint adjustment affects all the operation modes or just the current operation mode.

Setting "Setpoint adjustment affects current operation mode only": If comfort mode is currently activated, you can adjust the comfort setpoints for heating and cooling together, but you can not adjust the values for standby or night operation.

Two-stage heating or cooling

In order to shorten the heat-up phase with slow heating systems (e.g. underfloor heating), a second, more responsive heating system that heats up faster during the long start-up period of the main system (basic level) is frequently used.



The same behaviour applies with cooling systems.

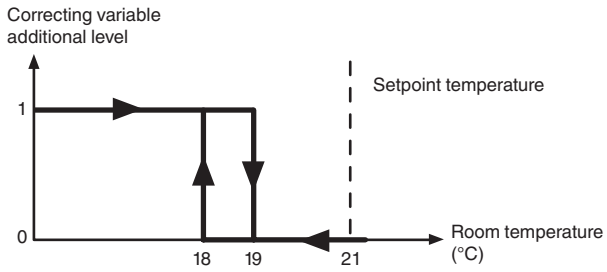
The additional level, which is controlled via 2-step control, remains switched on until a parameterised interval below the basic level is reached (e.g. 2 K), and then switches off. Only the basic level then remains switched on.

Example:

- Setpoint temperature: 21 °C
- Interval between basic level - additional level: 2 K
- Hysteresis of additional level: 1 K

The additional level remains switched on until "Setpoint temperature minus interval" (21 °C - 2 K = 19 °C) is reached. The additional level is then switched off.

It is only switched on again when the actual temperature is lower than the "setpoint temperature minus interval minus hysteresis" (21 °C - 2 K - 1 K = 18 °C).



The push-button displays the active basic level with a "1" and the active additional level with a "2".

Display

On the display, a symbol can show either the current controller status or whether heating or cooling is activated.

Display heating and cooling symbol = shows current controller status

Display	Mode
	Heating active, correcting variable ≠0
	Cooling active, correcting variable ≠0
1 2 (below the symbols)	1 = Heating/cooling active, correcting variable ≠0 For two-stage heating/cooling 1 = Basic level active, correcting variable ≠0 2 = Basic level and additional level active, correcting variable ≠0
--	Insensitive zone, correcting variables = 0

Display heating and cooling symbol = shows heating/cooling

Display	Mode
	Heating
	Cooling
1 2 (below the symbols)	1 = Heating/cooling active, correcting variable ≠0 For two-stage heating/cooling: 1 = Basic level active, correcting variable ≠0 2 = Basic level and additional level active, correcting variable ≠0

Controller types

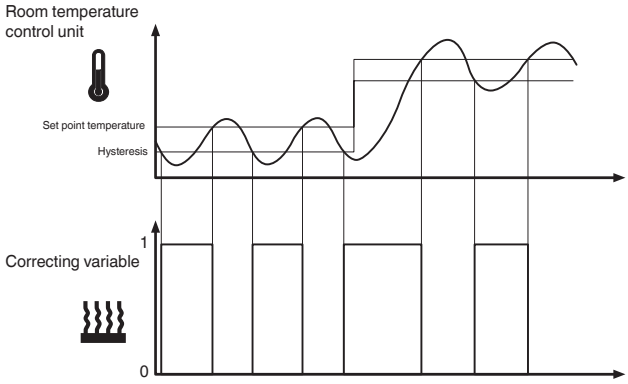
The room temperature control unit transmits correcting variables to the bus via various communication objects, which you can use to control different controller types with switching commands or by specifying percentage values:

- Continuous 2-step control
- Switching 2-step control
- Continuous PI control
- Switching PI control

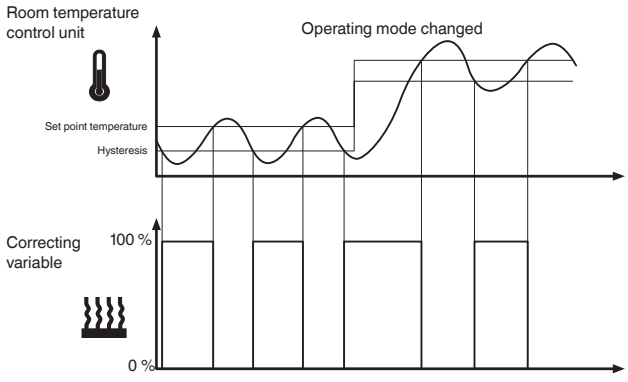
Continuous and switching 2-step control

The 2-step control is the simplest type of control. The heating switches on as soon as the actual temperature falls below a specific value, and switches off as soon as the setpoint temperature has been exceeded.

Switching 2-step control:



Continuous 2-step control:



The disadvantage of simple control, in contrast to its advantage, is that the room temperature is not constant but changes continuously, reducing comfort particularly when heating and cooling systems are slow to react. To counteract this effect, you can set a sufficiently small hysteresis. However, this leads to an increase in switching frequency, and therefore to increased wear of the drives.

The temperature overshoot above or below the hysteresis apparent in the diagram is caused when the heating/cooling system continues to emit heat or cold into the room after it has been switched off.

Setting rules for the 2-step control

"Hysteresis of the 2-step control" parameter:

- Small hysteresis:
  Leads to small fluctuations, but frequent switching.
- Large hysteresis:
  Leads to large fluctuations, but infrequent switching.

In general, due to the influences of the heating system and the room, the temperature fluctuations in the room are significantly higher than the hysteresis.

Continuous and switching PI control

For the PI control, the correcting variable is calculated from a proportional and an integral share. The calculation is governed by parameters such as:

- Temperature difference between actual value and setpoint
- Proportional range
- Reset time

In this way, the controller can correct the room temperature quickly and accurately. The corresponding correcting variable is transferred via a 1 bit/1 byte value to the bus.

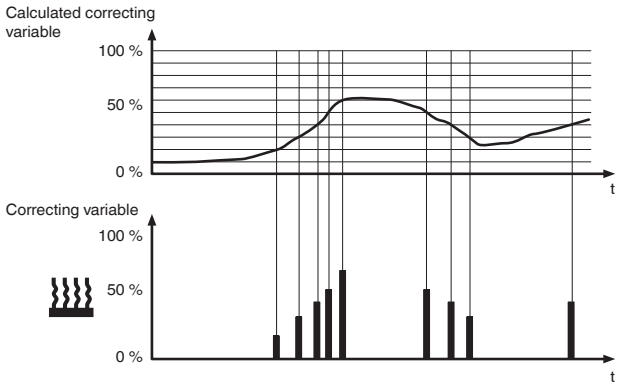
The standard control parameters for the most common system types are already installed in the controller:

- Warm water heating
- Underfloor heating
- Electric heating
- Fan convector
- Split unit
- Cooling ceiling

You can also set the control parameters for the proportional range and the reset time manually, but you should know exactly which actuators are connected and the control conditions in the room.

Continuous PI control

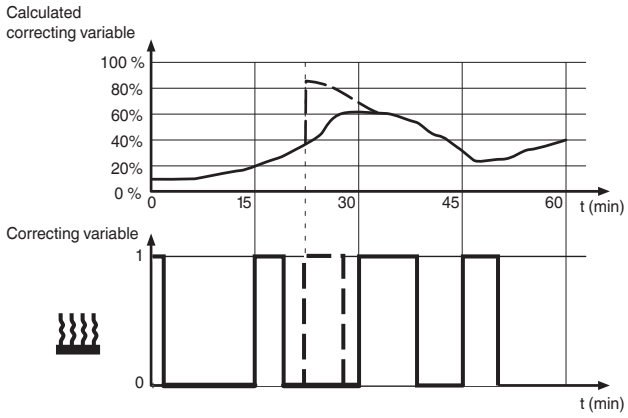
For the continuous PI control, the corresponding 1 byte correcting variable is transmitted 0-100% directly via the bus to the heating actuator or an EMO valve drive, which convert the correcting variable directly to a degree of opening. However, this is only transmitted when the newly calculated correcting variable has changed by a specified percentage.



Switching PI control (PWM)

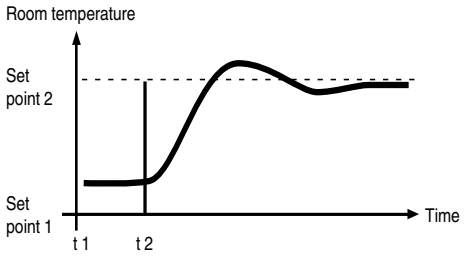
With the switching PI control, also known as the PWM control, the correcting variables calculated by the controller (0-100%) are converted into a pulse-width modulation (PWM). Within a constant, defined cycle time, the control actuator is opened ("1") and then closed again ("0") for the calculated percentage period. For example, when a correcting variable of 25% is calculated for a cycle time of 12 minutes, a "1" is transmitted at the beginning of the cycle time, and a "0" is transmitted after three minutes (= 25% of 12 minutes)

.



When the setpoint temperature changes, the controller recalculates the required correcting variable and transmits it still within the current cycle (broken line).

Setting rules for the PI control



In general:

- Large system increases (e.g. high heating output, steep characteristic curves for valves) are controlled with large proportional ranges.
- Slow heating systems (e.g. underfloor heating) are controlled with high-level reset times.

If no satisfactory control result is achieved by selecting an appropriate heating or cooling system, you can improve the adaptation "via control parameters":

- Small proportional range:
  Large overshoot for setpoint changes (also continuous oscillation under certain circumstances), rapid adjustment to the setpoint.
- Large proportional range:
  No (or little) overshooting, but slow adjustment.
- Short reset time:
  Rapid correction of control deviations (ambient conditions), risk of continuous oscillation.
- Long reset time:
  Slow correction of control deviations.

The framework conditions for setting the cycle time are as follows:

- For small values, the switching frequency and the bus load are increased.
- For large values, temperature fluctuations are created in the room.
- A short cycle time for rapid heating systems (e.g. electric heating).
- A long cycle time for slow heating systems (e.g. underfloor warm water heating).

Examples

Warm water radiator heating with motorised valve drives:

Characteristics	Parameter	Setting
Heating only	Controller type	Heating
	Correcting variable output	Continuous PI control
	Adjust the controller to the heating system	Warm water heating (5 K/150 min)

Cooling ceiling with motorised valve drives:

Characteristics	Parameter	Setting
Cooling only	Controller type	Cooling
	Correcting variable output	Continuous PI control
	Adjust the controller to the cooling system	Adjustment via control parameter
	Cooling proportional range	Approx. 5 K (depending on the application)
	Reset time for cooling	Approx. 240 min. (depending on the application)

Switching electric radiator heating:

Characteristics	Parameter	Setting
Heating only	Controller type	Heating
	Correcting variable output	Switching PI control
	Adjust the controller to the heating system	Electric heating (4 K/100 min)

Air conditioning with 4-duct (2-circuit) air convector system (e.g. switching valve drives):

Characteristics	Parameter	Setting
Heating or cooling as required, with automatic switching	Controller type	Heating and cooling
	Correcting variable output - heating	e.g. switching PI control
	Adjust the controller to the heating system	Air convector (4 K/90 min)
	Correcting variable output - cooling	e.g. switching PI control
	Adjust the controller to the cooling system	Air convector (4 K/90 min)
e.g. automatically switch between heating and cooling	Switch between heating and cooling	automatically via the controller

Temperature limitation using shading facility:

Characteristics	Parameter	Setting
Cooling only	Controller type	Cooling
	Correcting variable output - heating	Switching 2-step control
	Hysteresis	Large (e.g. 2 K)

Setting the room temperature control unit

Setting the general control parameters

Set the control first. Then specify which control type you want to plan.

Specify

- Whether and how the comfort extension should function, and which operation mode the device should go to after a reset.
- Whether setpoint adjustments via the control menu should be saved, or whether each setpoint you set should apply again when the operation mode is changed.
- How large a setpoint adjustment is allowed.
- Whether the setpoint adjustment affects the current operation mode only or all operation modes.

Control general	
Parameter	**Setting**
Use controller control	Yes
	No
Controller type	**Heating**
	Cooling
	Heating and cooling
Duration of comfort extension	None
	Test mode (1 min)
	30 min to 4.0 h, **1.0 h**
Termination of comfort extension via objects*	Yes
*Objects: comfort, standby, operation mode	**No**
Operation mode after comfort extension	Standby operation
	Night operation
	Current object values
Operation mode after reset	Comfort operation
	Standby operation
	Night operation
	Frost/heat protection operation
	Last operation
Operation mode after download	Comfort operation
	Standby operation
	Night operation
	Frost/heat protection operation
On what the setpoint adjustment has an effect	**Current operation mode**
	All operation modes
Setpoint adjustment maintained after change in operation mode	Yes
	No
Max. upper setpoint adjustment	0 - 10 K, **3 K**
Max. lower setpoint adjustment	0 - 10 K, **3 K**
Switch between heating and cooling	Automatically (via the controller)
	Externally (via heating/cooling object)
Heating/cooling read request after bus voltage recovery	Yes
	No
Waiting time after switching-over (heating/cooling)	Yes
	No
Waiting time (1-60 min)	1 ... 60, **10**

If you have parameterised the **"Heating and cooling"** controller type, either the controller automatically switches between heating and cooling or it is done by the "Heating/cooling" object accordingly.
If the controller switches automatically between heating and cooling, the controller is either in heating or cooling mode. The correcting variable of the non-active mode is switched to 0% (off).

When switching externally, switch to the heating mode with a "1" telegram, and to cooling mode with a "0" telegram.

If the same transmitting group address is used for the correcting variables for heating and cooling, the "0" telegrams of the non-active controller type are automatically suppressed.

Setting the operation mode and status

Here you can set whether to toggle between operation modes via 1 bit or 1 byte. Additionally, you define the 1 bit status object here.

If you want to display the system's status using visualisation software, there is one 1 byte status object and one 2 byte status object available for this purpose.

Operation mode/status	
Parameter	**Setting**
Toggle operation mode via 1 bit/ 1 byte	**1 bit**
	1 byte
Define 1 bit status object	Bit 0: Comfort
	Bit 1: Standby
	Bit 2: Night operation
	Bit 3: Frost/heat protection
	Bit 4: Dewpoint alarm
	Bit 5: Heating (1)/cooling (0)
	Bit 6: Controller inactive
	Bit 7: Frost alarm (1)

Structure of the 1 byte status object:

Bit 0 Comfort (1/0)
Bit 1 Standby (1/0)
Bit 2 Night operation (1/0)
Bit 3 Frost/heat protection (1/0)
Bit 4 Dewpoint alarm 1/0
Bit 5 Heating (1)/cooling (0)
Bit 6 Controller inactive (1/0)
Bit 7 Frost alarm (1/0)

Structure of the 2 byte status object:

Bit 00 Error (1/0)
Bit 01 * (0)
Bit 02 * (0)
Bit 03 * (0)
Bit 04 Additional heating level (1/0)
Bit 05 * (0)
Bit 06 * (0)
Bit 07 Heating inactive (1/0)
Bit 08 Heating (1)/cooling (0)
Bit 09 * (0)
Bit 10 Additional cooling level (1/0)
Bit 11 Cooling inactive (1/0)
Bit 12 Dewpoint alarm (1/0)
Bit 13 Frost alarm (1/0)
Bit 14 Temperature alarm (1/0)
Bit 15 * (0)

*not supported

Setting the setpoints

For each operation mode, there is a setpoint available for temperature control when changing operation mode automatically or manually. You have to specify this setpoint. You can change the setpoint manually via the control menu on the push-button within setpoint adjustment (see "Control general" tab). There is no setpoint adjustment for frost or heat protection.

Reference value for calculating the setpoint adjustment:

This parameter is only activated if the setpoint adjustment is to affect all operation modes equally ("Control general" tab). If you specify a new setpoint temperature via the object "Current setpoint temperature input", the setpoint adjustment is calculated as the difference between the reference value and the object value.

Example

Cooling/heating
Limits of setpoint adjustment = +3 K/-3 K
New set value: object "Current setpoint temperature input" = 24 °C
Reference setpoint for calculating the setpoint adjustment = 21 °C
Calculated setpoint adjustment = +3 °C

Initial status	Result
Cooling:	Cooling:
Heating protection = 35 °C	Heat protection = 35 °C
Night = 28 °C	Night = **31 °C**
Standby = 26 °C	Standby = **29 °C**
Comfort = 24 °C	Comfort = **27 °C**
Heating:	Heating
Comfort = 21 °C	Comfort = **24 °C**
Standby = 19 °C	Standby = **22 °C**
Night = 17 °C	Night = **20 °C**
Frost protection = 7 °C	Frost protection = 7 °C

Setpoints	
Parameter	**Setting**
Reference value for calculating the setpoint adjustment* *based on object "Current setpoint temperature input"	5.0 - 40 °C in 0.5 degree steps, **21.0 °C = 69.8 °F**
Heating	
Comfort setpoint	5.0 - 40 °C in 0.5 degree steps, **21.0 °C = 69.8 °F**
Standby setpoint	5.0 - 40 °C in 0.5 degree steps, **19.0 °C = 66.2 °F**
Night setpoint	5.0 - 40 °C in 0.5 degree steps, **17.0 °C = 62.6 °F**
Frost protection setpoint	0 - 15 °C in 1.0 degree steps, **7.0 °C = 44.6 °F**
Cooling	
Comfort setpoint	5.0 - 40 °C in 0.5 degree steps, **24.0 °C = 75.2 °F**
Standby setpoint	5.0 - 40 °C in 0.5 degree steps, **26.0 °C = 78.8 °F**
Night setpoint	5.0 - 40 °C in 0.5 degree steps, **28.0 °C = 82.4 °F**
Heat protection setpoint	18 - 40 °C in 1.0 degree steps, **35.0 °C = 95.0 °F**

Correct and send actual temperature

The actual temperature is affected by the following:

- Actual temperature inside (measured by internal sensors)
- Actual temperature outside (measured by external temperature sensors)
- Combination of internal and external actual temperature

You can set the temperature difference (the last difference transmitted compared to the current actual temperature) at which the actual temperature is transmitted, and the interval at which it should automatically be transmitted (e.g. to visualisation software).

Here, you can also set a correction value for the temperature sensor installed in the room temperature control unit. This is useful if it is installed in an unsuitable place in the room where the temperature is different from other places in the room (e.g. due to a draught or heat sources nearby), for example. The following formula applies:

Actual temperature = measured temperature + correction value

If you also use an external temperature sensor, you can set the percentage proportion at which the external actual temperature should be included in the current actual temperature. The external value is received via the "Current actual temperature input" object, read by the room temperature control unit and calculated according to the set weighting. The "Current actual temperature input" object is then overwritten by the calculated actual value.

You can set the system to monitor the actual external temperature cyclically. If the controller does not receive any new values during this time, a read request is sent. If no new value is received in response to this, the actual external temperature is equalised with the actual internal temperature.

If you do not want the system to monitor the actual external temperature, set the value here to "0".

Actual temperature (resulting)	
Parameter	**Setting**
Correct internal actual temperature Factor (-128 ... 127) * 0.1 K	-128 ... 127, **0**
Take actual external temperature partially into account	5 % to 100 %, **No**
Monitor actual external temperature every ... min (0-255)	0 ... 255, **60**
Send actual temperature if difference is ... K	No 0.1 to 2.0 K, **0.2 K**
Send actual temperature every ... min	No 3 to 60 min, **10 min**

Set temperature drop detection

When temperature drop detection is switched on, the room temperature control unit checks whether the temperature has changed by the set temperature difference within three minutes. If this is the case, the system switches to frost/heat protection mode for a period that you can set. After this time has elapsed, the controller switches back to the operation mode that was set previously.

You can set which temperature value or sensor the temperature for the measurement is taken from.

If using an external sensor, the actual temperature can consist of the temperature measured by the room temperature control unit and the external temperature, depending on the parameterised weighting (see "Correcting and setting the actual temperature").

Temperature jump	
Parameter	**Setting**
Temperature jump detection	**No** +/- 0.2 K / 3 min ... +/- 4.0 K / 3 min
Duration of the frost/heat protection in event of temperature jump (10-60 min) Frost protection during heating operation Heat protection during cooling operation	10 - 60 min, **20 min**
Temperature measurement	Of actual internal temperature Of actual external temperature Of actual internal or external temperature **Of (resulting) internal temperature**

Set closed-loop control for heating and additional level

This tab only appears if you set the "heating" or "heating and cooling" control type in the "Control general" tab. Here, you can set which heating control type should be activated. For PI controls, you can select between five standard system types, for which the recommended parameters have already been preconfigured. However, if you have sufficient specialised knowledge, you can also set the control parameters as required.

For 2-step control, set the hysteresis here.

Control heating	
Parameter	**Setting**
Basic level	
Direction of the controller	**Normal** Inverted
Correcting variable output	PI control (switching) **PI control (continuous)** 2-step control (switching) 2-step control (continuous)
Select heating system	Adjustment via control parameter **Warm water heating (5 K/150 min)** Underfloor heating (5 K/ 240 min) Electric heating (4 K/100 min) Air convector (4 K/90 min) Split unit (4 K/90 min)
Hysteresis	0.3 K - 2.0 K, **0.5 K**
Proportional range for heating in 0.1 K (10-255)	10 - 255, **40**
Reset time for heating (1-255 min)	**No**, 1 - 255

Here, specify the settings for the second heating level.

Control heating	
Parameter	**Setting**
Use additional level	Yes **No**
Direction of the controller	**Normal** Inverted
Correcting variable output	**2-step control (switching)** 2-step control (continuous)
Interval Factor (10 ... 100) * 0.1 K	10 ... 100, **20**
Hysteresis	0.3 K - 2.0 K, **0.5 K**

Set control cooling and additional level

This tab only appears if you set the "cooling" or "heating and cooling" control type in the "Control general" tab. Here, you can set which cooling control type should be activated. For PI controls, you can select between three standard system types, for which the recommended parameters have already been preconfigured. However, if you have sufficient specialised knowledge, you can also set the control parameters as required.

For 2-step control, set the hysteresis here.

Control cooling	
Parameter	**Setting**
Basic level	
Direction of the controller	**Normal**
	Inverted
Correcting variable output	PI control (switching)
	PI control (continuous)
	2-step control (switching)
	2-step control (continuous)
Select cooling system	Adjustment via control parameter
	Air convector (4 K/90 min)
	Split unit (4 K/90 min)
	Cooling ceiling (5 K/240 min)
Hysteresis	0.3 K - 2.0 K, **0.5 K**
Proportional range for cooling in 0.1 K (10-255)	10 - 255, **40**
Reset time for cooling (1-255 min)	**No**, 1 - 255

Here, specify the settings for the second cooling level.

Control heating	
Parameter	**Setting**
Use additional level	Yes
	No
Direction of the controller	**Normal**
	Inverted
Correcting variable output	**2-step control (switching)**
	2-step control (continuous)
Interval Factor (10 ... 100) * 0.1 K	10 ... 100, **20**
Hysteresis	0.3 K - 2.0 K, **0.5 K**

Set correcting variables and valve protection

Note that you need to set different parameters for 2-step control than you do for a PI control.

For "Cycle time of switching correcting variable", set the duration for the PI control. The calculated correcting variable is always transmitted at the start of a cycle time. If the valve drive has not received a value (e.g. during commissioning), the room could continuously heat up or cool down. To prevent this, set the "Cycle time for automatic sending of correcting variable". The correcting variable is transmitted again within the set time (as a precaution).

The minimum correcting variable must always be smaller than the maximum correcting variable! If not, the message "Er 4" appears on the display.

Valve protection

Valve protection prevents the valves on the heaters becoming stuck due to deposits in the heating water when the heating is switched off for a longer period of time (e.g. over the summer). When the valve protection is switched on, the valves are opened for a preset duration (100% on the controller output) after a preset number of days, and are then closed again (0% on the controller output).

The following settings apply for the "Heating" and "Cooling" controller types:

Correcting variables	
Parameter	**Setting**
Basic level	
Select a minimum correcting variable that is smaller than the maximum correcting variable.	
Cycle time of switching correcting variable (2-60 min)	2-60, **15**
Range of minimum correcting variable from 0 % to ... %	0 % - 100 %, **30 %**
Minimum correcting variable (0 % - 100 %)	0 % - 100 %, **30 %**
Range of maximum correcting variable from 100 % to ... %	0 % - 100 %, **70 %**
Maximum correcting variable (100 % - 0%)	0 % - 100 %, **70 %**
Change for which correcting variable is sent	2 % - 10 %, **3 %**
Send correcting variable cyclically	**Yes**
	No
Cycle time for automatic sending of correcting variable in min (1-60)	1-60, **30**
Send inactive correcting variable cyclically	**Yes**
	No
Use valve protection	Yes
	No
Activate valve protection every ... days (1 - 30)	1-30, **15**
Approach end position for ... min (1-30)	1-30, **4**

Correcting variables	
Parameter	**Setting**
Additional level	
Send correcting variable cyclically	**Yes** No
Cycle time for automatic sending of correcting variable in min (1-60)	1-60, **30**
Send inactive correcting variable cyclically	**Yes** No
Use valve protection	Yes **No**
Activate valve protection every ... days (1 - 30)	1-30, **15**
Approach end position for ... min (1-30)	1-30, **4**

Control problems - what to do if

Problem	**Possible solution**
The controller switches constantly between heating and cooling.	Increase insensitive zone between heating and cooling or increase "Waiting time after switching-over".
The setpoint is only reached very slowly.	Decrease the proportional range, either by selecting a system type with a smaller proportional range in the system-specific selection "Adjustment of the controller to the heating/ cooling system", or by directly reducing the proportional range via control parameters in the adjustment, or by decreasing the integral time.
The room temperature exceeds the limits when changes are made to the setpoint.	Increase the proportional range, either by selecting a system type with a larger proportional range in the system-specific selection "Adjustment of the controller to the heating/ cooling system", or by directly increasing the proportional range via control parameters in the adjustment, or by increasing the integral time.
It is always too hot or too cold in the room.	Compensate for the room temperature measurement by changing the "Compensation of the internal actual temperature" parameters accordingly.

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Control	Current actual temperature output	2 byte	Low	CRT	Transmit/ read out
Control	Current actual temperature input	2 byte	Low	WCT+	Transmit/ receive
Control	Current setpoint temperature output	2 byte	Low	CRT	Transmit/ read out
Control	Current setpoint temperature input	2 byte	Low	WC	Receive
Control	Operation mode output	1 byte	Low	CRT	Transmit/ read out
Control	Operation mode input	1 byte	Low	WC	Receive
Control	Frost/heat protection output	1 bit	Low	CRT	Transmit/ read out
Control	Frost/heat protection input	1 bit	Low	WC	Receive
Control	Heating/cooling input	1 bit	Low	WCT+	Transmit/ receive
Control	Heating/cooling output	1 bit	Low	CRT	Transmit/ read out
Control	Comfort output	1 bit	Low	CRT	Transmit/ read out
Control	Comfort input	1 bit	Low	WC	Receive
Control	Comfort extension output	1 bit	Low	CRT	Transmit/ read out
Control	Comfort extension input	1 bit	Low	WC	Receive
Control	Night reduction output	1 bit	Low	CRT	Transmit/ read out
Control	Night reduction input	1 bit	Low	WC	Receive
Control	Setpoint adjustment output	2 byte	Low	CRT	Transmit/ read out
Control	Setpoint adjustment input	2 byte	Low	WC	Receive
Control	Disable object for output	1 bit	Low	CRT	Transmit/ read out
Control	Disable object for input	1 bit	Low	WC	Receive
Control	Status	1 byte	Low	CRT	Transmit/ read out
Control	Status	2 byte	Low	CRT	Transmit/ read out
Control	Status (comfort)	1 bit	Low	CRT	Transmit/ read out
Control	Status (standby)	1 bit	Low	CRT	Transmit/ read out
Control	Status (night operation)	1 bit	Low	CRT	Transmit/ read out
Control	Status (frost/heat protection)	1 bit	Low	CRT	Transmit/ read out
Control	Status (dewpoint alarm)	1 bit	Low	CRT	Transmit/ read out
Control	Status (heating/cooling)	1 bit	Low	CRT	Transmit/ read out
Control	Status (controller inactive)	1 bit	Low	CRT	Transmit/ read out
Control	Status (frost alarm)	1 bit	Low	CRT	Transmit/ read out
Control	Heating status (basic level)	1 byte	Low	CRT	Transmit/ read out

Function	Object name	Type	Prio	Flags	Behaviour
Control	Cooling status (basic level)	1 byte	Low	CRT	Transmit/ read out
Control	Correcting variable heating (basic level)	1 bit	Low	CRT	Transmit/ read out
Control	Correcting variable heating (basic level)	1 byte	Low	CRT	Transmit/ read out
Control	Correcting variable heating (additional level)	1 bit	Low	CRT	Transmit/ read out
Control	Correcting variable heating (additional level)	1 byte	Low	CRT	Transmit/ read out
Control	Correcting variable cooling (basic level)	1 bit	Low	CRT	Transmit/ read out
Control	Correcting variable cooling (basic level)	1 byte	Low	CRT	Transmit/ read out
Control	Correcting variable cooling (additional level)	1 bit	Low	CRT	Transmit/ read out
Control	Correcting variable cooling (additional level)	1 byte	Low	CRT	Transmit/ read out
Control	Dewpoint alarm	1 bit	Low	WC	Receive

Display external temperature in display

You can cyclically read an external temperature value (2-byte value) via the bus and display it. The external temperature can be the outdoor temperature transmitted from a weather station, for example.

Display external temperature	
Parameter	**Settings**
Read external temperature cyclically	Yes
	No
Cycle time = base * factor	
Base	1 s
	1 min
	1 h
	1 day
Factor (3-255)	3-255, **3**

The external temperature is only shown in the display, this temperature has no effect on the control behaviour of the push-button.

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Display external temperature	External temperature	2 byte	Low	WCT	Transmit/ receive

Controlling and displaying the fan speed

You can use this setting in combination with the Fan Coil actuator, for example. If you combine it with the "Linear regulator" push-button function, you can easily control fan speeds with one push-button.

The display can be used to show whether the fan of the fan coil actuator is working in automatic or manual mode and which fan speed is currently activated. The status feedback object can be used to send the status to the LED.

In the parameters, you can set the percentage values at which the individual levels are displayed.

Fan speed	
Parameter	**Setting**
Number of fan speeds	1 - 7, **3**
Value range for speed display: 1-100 %	
Display fan speed 1 up to ... %	1 - 100%, **34%**
Display fan speed 2 up to ... %	1 - 100%, **67%**
Display fan speed 3 up to ... %	1 - 100%, **100%**
Display "Auto" in display	**If fan status automatic is "0"**
	If fan status automatic is "1"

Communication objects

You can select the following communication objects:

Function	Object name	Type	Prio	Flags	Behaviour
Display of fan speed	Fan 0 -100 %	1 byte	Low	WC	Receive
Display automatic	Fan status automatic	1 bit	Low	WC	Receive

Behaviour on bus voltage recovery / bus voltage failure

Behaviour on application/recovery of the bus voltage

Depending on the application settings:

- The status LEDs are switched on or off or they flash.
- The operational LED can be switched on.
- One of the following operation modes is activated: comfort, standby, night, frost/heat protection or the last operation mode.
- The clock symbol may flash in the display when the time has not yet been automatically synchronised or manually set.
- The background lighting of the display may be switched on.

Telegrams

- A telegram can be sent to request time synchronisation.
- Telegrams for the control function (actual temperature, correcting variables etc.) may be transmitted.

Behaviour when bus voltage fails

The operation LEDs, status LEDs are switched off together with the display, including the background lighting.

Displays and error messages

Er 2	A contradiction has been detected in the heating parameters (setpoints or insensitive zone are inconsistent), e.g. heating setpoint temperature ≥ cooling setpoint temperature
Er 3	The software in the push-button is not compatible with the ETS application
Er 4	Upper control value range ≤ lower control value range
Er 5	Memory error
Er 6	Error in temperature sensor
Er 7	STACK error
Er 8	RAM error
APL	Application not loaded or faulty

Schneider Electric Industries SAS

If you have technical questions, please contact the Customer Care Center in your country.

www.schneider-electric.com

This product must be installed, connected and used in compliance with prevailing standards and/or installation regulations. As standards, specifications and designs develop from time to time, always ask for confirmation of the information given in this publication.