# KNX eTR 102 FC

## Room Temperature Controller for Fan Coil A/C

Item numbers 71320 (white), 71322 (black)


elsner

**Manual**

This manual is amended periodically and will be brought into line with new software releases. The change status (software version and date) can be found in the contents footer. If you have a device with a later software version, please check **www.elsner-elektronik.de** in the menu area "Service" to find out whether a more up-to-date version of the manual is available.

## Clarification of signs used in this manual

Safety advice.

Safety advice for working on electrical connections, components, etc.

**DANGER!** ... indicates an immediately hazardous situation which will lead to death or severe injuries if it is not avoided.

**WARNING!** ... indicates a potentially hazardous situation which may lead to death or severe injuries if it is not avoided.

**CAUTION!** ... indicates a potentially hazardous situation which may lead to trivial or minor injuries if it is not avoided.


**ATTENTION!** ... indicates a situation which may lead to damage to property if it is not avoided.

ETS In the ETS tables, the parameter default settings are marked by underlining.

# 1. Safety and operating instructions

Installation, testing, operational start-up and troubleshooting should only be performed by an authorised electrician.

**CAUTION!**
**Live voltage!**

- Inspect the device for damage before installation. Only put undamaged devices into operation.
- Comply with the locally applicable directives, regulations and provisions for electrical installation.
- Immediately take the device or system out of service and secure it against unintentional switch-on if risk-free operation is no longer guaranteed.

Use the device exclusively for building automation and observe the operating instructions. Improper use, modifications to the device or failure to observe the operating instructions will invalidate any warranty or guarantee claims.
Operate the device only as a fixed-site installation, i.e. only in assembled condition and after conclusion of all installation and operational start-up tasks, and only in the surroundings designated for it.
Elsner Elektronik is not liable for any changes in norms and standards which may occur after publication of these operating instructions.

**For information on installation, maintenance, disposal, scope of delivery and technical data, please refer to the installation instructions.**

# 2. Description

The room temperature controller for the KNX building bus system automatically controls air conditioning with fan coil units. It has touch-sensitive buttons with which the room climate settings can be manually readjusted. The fan speed (1-3), the setpoint temperature (value, touch buttons +/-) and the mode (automatic on/off, air conditioning on/off) can be set. LEDs show the current settings. The brightness and switch-off behaviour of the LEDs can be adjusted.

A temperature sensor is integrated in the unit, which can be used for control. To obtain a mixed value, the unit can receive another measured value via the bus and process it with its own value to obtain an overall temperature.

The automatic room climate control includes a PI controller for a heating and a cooling (one- or two-stage). This control contains parameters specifically for air conditioning with fan coil systems (systems with fans).

The room temperature controller can also be used as an extension unit, i.e. it transmits the manual inputs to another control device in the KNX system, which then takes over the control.

Communication objects can be linked in integrated AND and OR logic gates.

***Functions:***

- **Operating zone for setting 3 blower levels**
- **Operating zone for temperature control** with 2 areas (+ warmer, - cooler)
- **Operating zone for mode setting** with 2 areas (automatic on/off, air-conditioning on/off)
- The lighting behaviour of all LEDs is adjustable, so they can be used as ambient lighting, for example, or switched off as long as there is no input
- **Temperature** measurements. **Mixed value** from own measured value and external values (proportion can be set by percentage), output of minimum and maximum values
- **PI-controller for heating** (one or two-level) and **cooling** (one or two-level) according to temperature. Regulation according to separate setpoints or basic setpoint temperature. Parameters specifically for fan coil control for fan coil units
- **2 AND and 2 OR logic gates** each with 4 inputs. 8 logic inputs (in the form of communications objects) can be used as inputs for the logic gates. The output from each gate can be configured optionally as 1-bit or 2 x 8-bit

# 3. Commissioning

Configuration is made using the KNX software as of ETS 5. The **product file** can be downloaded from the ETS online catalogue and the Elsner Elektronik website on **www.elsner-elektronik.de**.

After the bus voltage has been applied, the device will enter an initialisation phase lasting approx. 5 seconds. During this phase no information can be received or sent via the bus.

# 4. Addressing the device

The device is delivered ex works with the bus address 15.15.255. You program a different address in the ETS by overwriting the address 15.15.255 or teach the device using the programming button.

The programming button is located at the bottom of the device and is recessed. Use a thin object to reach the button, e. g. a 1.5 mm² wire. When the button is pressed, the temperature display on the front flashes.

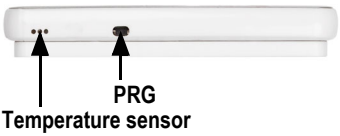



*Fig. 1*
*View from bottom*

# 5. Display and operation at the device

## 5.1. Adjust room temperature

Depending on the setting of the "Temperature display" parameter in the device application, the **Room Temperature Controller KNX eTR 102 FC** displays the current room temperature value (or mixed value), the target value or the shift in relation to the basic setpoint. The display can be dimmed and switched off via the bus so that *no* value is displayed even when touched.

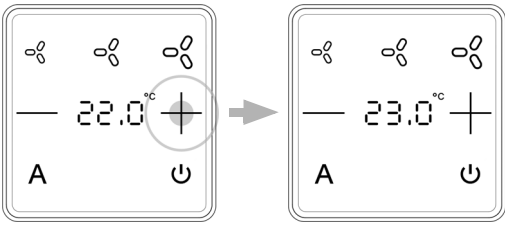
***Option A: Display of actual temperature (room temperature)***

The current room temperature is displayed. It is *not* possible to change the room temperature manually using the +/- buttons.

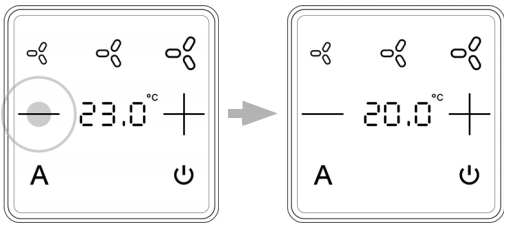
***Option B: Display of target temperature or basic setpoint shift***

Depending on the setting, the current target value or the shift relative to the base setpoint is displayed. The temperature can be changed by touching the +/- buttons.

**Target value** display (absolute value):

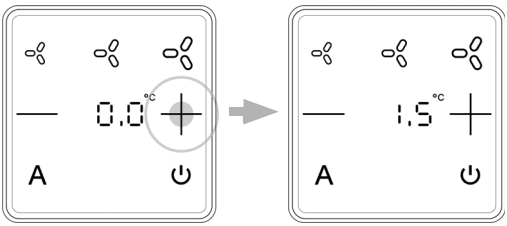



Tap +:
Increase room temperature (target temperature is increased)

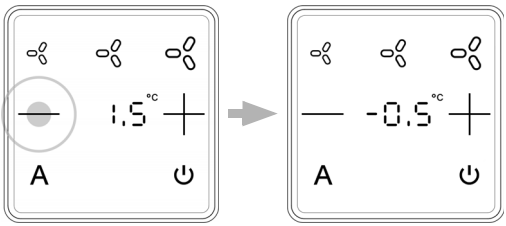



Tap -:
Lower room temperature (target temperature is lowered)

Display of the **basic setpoint shift** (change compared to the basic setpoint of the control):



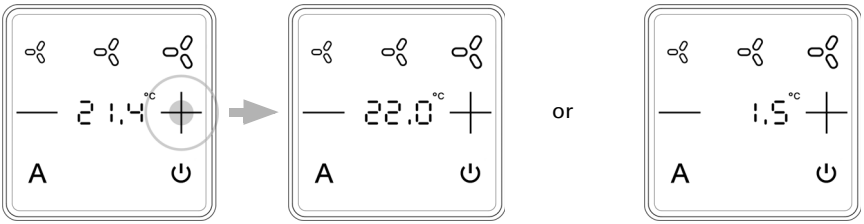

Tap +:
Increase room temperature
(Basic setpoint shift direction PLUS)




Tap -:
Lower room temperature
(Basic setpoint shift direction MINUS)

***Option C: Display of actual temperature and target temperature/basic setpoint shift***

During normal operation, the current room temperature is displayed. By touching the buttons, the display jumps to the target temperature or to the basic setpoint shift, depending on the presetting. Changes with + or - are visible. The display returns to the room temperature if no button is touched for 7 seconds.




Touch the **+ or -** button briefly: The current **target temperature** (or the basic setpoint shift) is displayed.

Tap +: Increase room temperature
(target temperature/basic setpoint shift is increased).

Tap -: Lower room temperature
(target temperature/basic setpoint shift is lowered).

***General:***

The step size for the change and the possible setting range are defined in the device application (ETS). There you can also define whether the manually changed values are retained after a mode change (e.g. Eco mode overnight) or reset to the stored values.

# 6. Transmission protocol

***Units:***

*Temperatures in degrees Celsius*

## 6.1. List of all communication objects

***Abbreviations Flags:***

*C Communication*
*R Read*
*W Write*
*T Transmit*
*U Update*

| No | Text | Function | Flags | DPT type | Size |
|---|---|---|---|---|---|
| 0 | Software version | Output | R-CT- | [217.1] DPT_Version | 2 Bytes |
| 20 | Temperature sensor: malfunction | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 21 | Temperature sensor: measured value external | Input | -WCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 22 | Temperature sensor: measured value | Output | R-CT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 23 | Temperature sensor: measured value total | Output | R-CT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 24 | Temperature sensor: measured value min./max. query | Input | -WC-- | [1.17] DPT_Trigger | 1 Bit |
| 25 | Temperature sensor: measured value minimum | Output | R-CT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 26 | Temperature sensor: measured value maximum | Output | R-CT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 27 | Temperature sensor: measured value min./max. reset | Input | -WC-- | [1.17] DPT_Trigger | 1 Bit |
| 30 | Temp. thresholdV 1: Absolute value | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 31 | Temp. thresholdV 1: (1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 32 | Temp. thresholdV 1: Switching delay from 0 to 1 | Input | -WC-- | [7.5] DPT_TimePeriodSec | 2 Bytes |
| 33 | Temp. thresholdV 1: Switching delay from 1 to 0 | Input | -WC-- | [7.5] DPT_TimePeriodSec | 2 Bytes |
| 34 | Temp. thresholdV 1: Switching output | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 35 | Temp. thresholdV 1: Switching output block | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 36 | Temp. thresholdV 2: Absolute value | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |

| No | Text | Function | Flags | DPT type | Size |
|---|---|---|---|---|---|
| 37 | Temp. thresholdV 2: (1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 38 | Temp. thresholdV 2: Switching delay from 0 to 1 | Input | -WC-- | [7.5] DPT_TimePeriod-Sec | 2 Bytes |
| 39 | Temp. thresholdV 2: Switching delay from 1 to 0 | Input | -WC-- | [7.5] DPT_TimePeriod-Sec | 2 Bytes |
| 40 | Temp. thresholdV 2: Switching output | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 41 | Temp. thresholdV 2: Switching output block | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 50 | Temp.control: HVAC mode (priority 1) | Input / Output | RWCTU | depending on setting | 1 Byte |
| 51 | Temp.control: HVAC mode (priority 2) | Input / Output | RWCT- | depending on setting | 1 Byte |
| 52 | Temp.control: Mode frost/heat protection activt. | Input | RWCT- | [1.1] DPT_Switch | 1 Bit |
| 53 | Temp. control: On/Off | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 54 | Temp.control: Current setpoint | Output | R-CT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 55 | Temp.control: Switch./Status | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 56 | Temp.control: Setpoint Comfort heating | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 57 | Temp.control: Setpoint Comfort heat.(1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 58 | Temp.control: Setpoint Comfort cooling | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 59 | Temp.control: Setpoint Comfort cool.(1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 60 | Temp.control: Basic 16-bit setpoint shift | Input | -WC-- | [9.1] DPT_Value_Temp | 2 Bytes |
| 61 | Temp.control: Setpoint Standby heating | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 62 | Temp.control: Setpoint Standby heat.(1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 63 | Temp.control: Setpoint Standby cooling | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 64 | Temp.control: Setpoint Standby cool. (1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 65 | Temp.control: Setpoint Eco heating | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 66 | Temp.control: Setpoint Eco heating (1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 67 | Temp.control: Setpoint Eco cooling | Input / Output | RWCT- | [9.1] DPT_Value_Temp | 2 Bytes |
| 68 | Temp.control: Setpoint Eco cooling (1:+ \| 0:-) | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |

| No | Text | Function | Flags | DPT type | Size |
|---|---|---|---|---|---|
| 69 | Temp.control: Control variable heating (level 1) | Output | R-CT- | [5.1] DPT_Scaling | 1 Byte |
| 70 | Temp.control: Control variable heating (level 2) | Output | R-CT- | [5.1] DPT_Scaling | 1 Byte |
| 71 | Temp.control: Control variable cooling (level 1) | Output | R-CT- | [5.1] DPT_Scaling | 1 Byte |
| 72 | Temp.control: Control variable cooling (level 2) | Output | R-CT- | [5.1] DPT_Scaling | 1 Byte |
| 73 | Temperature control: Variable for 4/6-way valve | Output | R-CT- | [5.1] DPT_Scaling | 1 Byte |
| 74 | Temp.control: Status Heat. level 1 (1=ON\|0=OFF) | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 75 | Temp.control: Status Heat. level 2 (1=ON\|0=OFF) | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 76 | Temp.control: Status Cool. level 1 (1=ON\|0=OFF) | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 77 | Temp.control: Status Cool. level 2 (1=ON\|0=OFF) | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 78 | Temp.control: Comfort extension status | Input / Output | RWCT- | [1.1] DPT_Switch | 1 Bit |
| 79 | Temp.control: Comfort Extension time | Input | RWCT- | [7.5] DPT_TimePeriodSec | 2 Bytes |
| 80 | Temp. Controller: Fan coil levels 0 to 3 | Output | R-CT- | [5.1] DPT_Scaling | 1 Byte |
| 81 | Temp. Controller: Fan coil level 1 | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 82 | Temp. Controller: Fan coil level 2 | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 83 | Temp. Controller: Fan coil level 3 | Output | R-CT- | [1.1] DPT_Switch | 1 Bit |
| 84 | Temp. Controller: Fan coil auto=1 manual=0 | Input / Output | RWCT- | [1.1] DPT_Switch | 1 Bit |
| 86 | All LEDs On/Off | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 87 | All LEDs Brightness | Input | -WC-- | [5.1] DPT_Scaling | 1 Byte |
| 107 | Logic input 1 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |
| 108 | Logic input 2 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |
| 109 | Logic input 3 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |
| 110 | Logic input 4 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |
| 111 | Logic input 5 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |
| 112 | Logic input 6 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |
| 113 | Logic input 7 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |
| 114 | Logic input 8 | Input | -WC-- | [1.2] DPT_Bool | 1 Bit |

| No | Text | Function | Flags | DPT type | Size |
|---|---|---|---|---|---|
| 117 | AND logic 1: 1 bit switching output | Output | R-CT- | [1.2] DPT_Bool | 1 Bit |
| 118 | AND logic 1: 8 bit output A | Output | R-CT- | depending on setting | 1 Byte |
| 119 | AND logic 1: 8 bit output B | Output | R-CT- | depending on setting | 1 Byte |
| 120 | AND logic 1: Block | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 121 | AND logic 2: 1 bit switching output | Output | R-CT- | [1.2] DPT_Bool | 1 Bit |
| 122 | AND logic 2: 8 bit output A | Output | R-CT- | depending on setting | 1 Byte |
| 123 | AND logic 2: 8 bit output B | Output | R-CT- | depending on setting | 1 Byte |
| 124 | AND logic 2: Block | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 125 | OR logic 1: 1 bit switching output | Output | R-CT- | [1.2] DPT_Bool | 1 Bit |
| 126 | OR logic 1: 8 bit output A | Output | R-CT- | depending on setting | 1 Byte |
| 127 | OR logic 1: 8 bit output B | Output | R-CT- | depending on setting | 1 Byte |
| 128 | OR logic 1: Block | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |
| 129 | OR logic 2: 1 bit switching output | Output | R-CT- | [1.2] DPT_Bool | 1 Bit |
| 130 | OR logic 2: 8 bit output A | Output | R-CT- | depending on setting | 1 Byte |
| 131 | OR logic 2: 8 bit output B | Output | R-CT- | depending on setting | 1 Byte |
| 132 | OR logic 2: Block | Input | -WC-- | [1.1] DPT_Switch | 1 Bit |

# 7. Setting the parameters

## 7.1. Behaviour on power failure/ restoration of power

***Behaviour following a failure of the bus power supply:***

The device sends nothing.

***Behaviour on bus restoration of power and following programming or reset:***

The device sends all outputs according to their send behaviour set in the parameters. Delays established in the "General settings" parameter block are taken into account.

## 7.2. General settings

First set the send delays after reset/bus restoration here.
These delays should be coordinated with the entire KNX-system, i.e. in a KNX system with many participants, care should be taken that the bus is not overloaded after a KNX-bus reset. The messages of the individual participants should be sent offset.

| | |
|---|---|
| Transmission delay after reset/bus restoration | 5 s • ... • 300 s |

The bus load is limited with the aid of the maximum message rate. Many messages per second put a strain on the bus, but ensure faster data transmission.

| | |
|---|---|
| Maximum message rate | • 1 message per second<br>• ...<br>• 10 messages per second<br>• ...<br>• 50 messages per second |

## 7.3. Temperature measured value

Select, whether a **malfunction object** is to be sent if the sensor is faulty.

| | |
|---|---|
| Use malfunction object | No • Yes |

When **measuring temperature**, the self-heating of the device is considered by the electronics. The heating is compensated for by the device.

Use **Offsets** to adjust the readings to be sent. Permanent measurement variations can be corrected in this way.

| | |
|---|---|
| Offset in 0.1°C | -50…50; 0 |

The unit can calculate a **mixed value** from its own reading and an external value, e.g. in order to determine a room average. Set the mixed value calculation if desired. If an external portion is used, all of the following settings (threshold values, etc.) are related to the overall reading.

| | |
|---|---|
| Use external measured value | No • Yes |

The percentage of the external measured value share of the total value is set here.

| | |
|---|---|
| Ext. Reading proportion of the total reading | 5% • 10% • ... • 50% • ... • 100% |

The internal measured value and the total measured value can be sent to the bus and further processed there by other participants.

| | |
|---|---|
| Sending pattern for internal and total measured value | • never<br>• periodically<br>• on change<br>• on change and periodically |

When sending on change, the temperature values are sent to the bus as soon as it changes by the value set here.

| | |
|---|---|
| At and above change of<br>*(if sent on change)* | 0.1°C • 0.2°C • 0.5°C • ... • 5.0°C |

When sending periodically, the temperature values are sent to the bus in a fixed cycle that can be set here.

| | |
|---|---|
| Send cycle<br>*(if sent periodically)* | 5 s • 10 s • ... • 2 h |

The highest (max.) and the lowest (min.) temperature value since programming or a reset can be sent to the bus. The two values can be reset via object no. 27 “Temperature sensor: measured value min./max. reset”.

| | |
|---|---|
| Use minimum and maximum value | No • Yes |

# 7.4. Temperature threshold values

The temperature threshold values are used to carry out certain actions in the KNX system when a temperature value is exceeded or not reached.

| | |
|---|---|
| Use threshold value 1 / 2 | Yes • No |

## 7.4.1. Temperature threshold value 1 / 2

**Threshold value:**

Decide when **threshold values and delay times** received are to be kept per object. The parameter is only taken into consideration if the setting by object is activated further down. Please note that the setting "After power restoration and programming" should not be used for the initial start-up, as the factory settings are always used until the first call (setting via objects is ignored).

| | |
|---|---|
| The nominal values and delay times | |
| received by the communication object should be retained | • never<br>• after restoration of power<br>• after power restoration and programming |
| . | |

Set the threshold values directly in the application program using parameters or define them via the bus using a communication object.

| | |
|---|---|
| Threshold value setpoint using | Parameter • Communication objects |

***Threshold value setpoint using parameter:***

When the **threshold value per parameter** is specified, then the value is set.

| | |
|---|---|
| Threshold value in 0.1°C | -300 … 800; 200 |

***Threshold value setpoint using a communication object:***

During initial commissioning, a threshold value must be defined which will be valid until the first call with a new threshold value. For units which have already been taken into

service, the last communicated threshold value can be used. Basically, a range is given in which the threshold value can be changed (object value limit).
A set threshold value will be retained until a new value or change is transferred. The current value is saved so that it is retained in the event of a power supply failure and will be available again once the supply voltage is restored.

| | |
|---|---|
| Start threshold value in 0.1°C<br>valid until the first communication | -300 ... 800; 200 |

Minimum value that can be set via object.

| | |
|---|---|
| Limitation of object value (min) in 0.1°C | -300…800 |

Maximum value that can be set via object.

| | |
|---|---|
| Limitation of object value (max) in 0.1°C | -300…800 |

Enter how the threshold value will be received from the bus beforehand. Basically, a new absolute value can be received, or simply a command to increase or decrease.

| | |
|---|---|
| Type of threshold value change | Absolute value • Increase/decrease |

Choose the step size.

| | |
|---|---|
| Step size<br>*(upon increase/decrease change)* | 0.1°C • 0.2°C • 0.3°C • 0.4°C • 0.5°C • 1°C • 2°C • 3°C • 4°C • 5°C |

The switching distance (hysteresis) is important for the output of the value at the switching output.
The switching distance prevents the switching output of the threshold value from changing too often in the event of temperature fluctuations. When the temperature value drops, the switching output does not react until the switching distance falls below the threshold value (options 1 and 2 in the first parameter of switching output). When the temperature value increases, the switching output does not react until the switching distance exceeds the threshold value (options 3 and 4 in the first parameter of switching output).

| | |
|---|---|
| Setting the switching distance | in % • absolute |

Set the value of the switching distance.

| | |
|---|---|
| Switching distance in 0.1°C | 0...1100; 50 |
| Switching distance in % of the threshold value | 0 ... 50; 20 |

### Switching output:

Here it is set which value the output transmits if the threshold value is exceeded or undercut.

| | |
|---|---|
| When the following conditions apply, the output is<br>(TV = Threshold value) | • TV above = 1 \| TV - Hyst. below = 0<br>• TV above = 0 \| TV - Hyst. Below = 1<br>• TV below = 1 \| TV + Hyst. above = 0<br>• Below TV = 1 \| Above TV + Hyst. = 0 |

Here it is set whether switching delays can be set via objects.

| | |
|---|---|
| Delays can be set via objects<br>(in seconds) | No • Yes |

Switching command delays ignore short-term temperature fluctuations around the threshold value or threshold value and switching distance for the switching output.

| | |
|---|---|
| Switching delay from 0 to 1 | none • 5 s ... 2 h |
| Switching delay from 1 to 0 | none • 5 s ... 2 h |

Here you set the cases in which the switching output is to be sent to the bus.

| | |
|---|---|
| Switching output sends | • If there is a change<br>• on change to 1<br>• on change to 0<br>• on change and periodically<br>• on change to 1 and periodically<br>• on change to 0 and periodically |

When sending periodically, the temperature threshold value switching output is sent on the bus in a fixed cycle that can be set here.

| | |
|---|---|
| Send cycle<br>*(is sent only if "periodically" is selected)* | 5 s ... 2 h |

### Block:

With the help of the "Blocking" input object, the switching output can be blocked, e.g. by a manual command (push button).

| | |
|---|---|
| Use switching output block | No • **Yes** |

The block can take effect at value 0 or 1, depending on the intended use.

| | |
|---|---|
| Assessment of the block object | • At value 1: block \| At value 0: release<br>• At value 0: block \| At value 1: release |

An object value up to the 1st communication is specified here.

| | |
|---|---|
| Blocking object value before first communication | 0 • 1 |

The behaviour of the switching output during locking can be set.

| Switching output behaviour | |
|---|---|
| On blocking | • Do not send message<br>• send 0<br>• send 1 |

The behaviour of the switching output on release, i.e. when the lock is removed, is dependent on the value of the parameter "Switching output sends" (see "Switching output").

| Switching output behaviour | |
|---|---|
| On release<br>(with 2 second release delay) | [Dependent on the "Switching output sends" setting] |

| | |
|---|---|
| Switching output sends on change | do not send message •<br>Status object/s send/s |
| Switching output sends on change to 1 | do not send message •<br>if switching output = 1 ➔ send 1 |
| Switching output sends on change to 0 | do not send message •<br>if switching output = 0 ➔ send 0 |
| Switching output sends on change and periodically | Send switching output status |
| Switching output sends on change to 1 and periodically | if switching output = 1 ➔ send 1 |
| Switching output sends on change to 0 and periodically | if switching output = 0 ➔ send 0 |

## 7.5. Temperature PI control – Independent controller

Activate the control.

| Use control | No • Yes |
|---|---|

Determine if this device should take over the temperature control (stand-alone controller), or if this device should act as an extension for operating another controller.

| Intended as a | • **Stand-alone controller**<br>• Controller extension (for operating a stand-alone controller only) |
|---|---|

The settings for the 'Stand-alone controller' option are described below. For configuration as an extension, please see Chapter *Temperature PI control – Controller extension unit*, page28.

### General control

Set, in which cases **setpoint values and extension time** received via object are to be retained. The parameter is only taken into consideration if the setting via object is

activated below. Please note that the setting "After power supply restoration and programming" should not be used for the initial start-up, as the factory settings are always used until the 1st communication (setting via objects is ignored).

| Maintain the | |
|---|---|
| Target values and extension time received via communication objects | • never<br>• after power supply restoration<br>• after power supply restoration and programming |
| | |

For an adequate regulation of the ambient temperature, comfort, standby, eco and building protection modes may be used.
**Comfort** when present,
**Standby** during short absences,
**Eco** as a night-time mode and
**Frost/heat protection** (building protection) e. g. with the window open.

The settings for the temperature control include the setpoint temperatures for the individual modes. Objects are used to determine which mode is to be selected. A change of mode may be triggered manually or automatically (e.g. by a timer, window contact).

The **mode** may be switched with two 8 bit objects of different priority. The objects are:
„... HVAC mode (Prio 2)“ for switching in everyday operation and
„... HVAC mode (Prio 1)“ for central switching with higher priority.
The objects are coded as follows:
0 = Auto
1 = Comfort
2 = Standby
3 = Eco
4 = Building Protection

Alternatively, you can use three objects, with one object switching between eco and standby mode and the two others activating comfort mode and frost/heat protection mode respectively. The comfort object blocks the eco/standby object, and the frost/heat protection object has the highest priority. The objects are:
„... Mode (1: Eco, 0: Standby)“,
„... comfort activation mode" and
„... frost/heat protection activation mode"

| Switch mode via | • two 8 Bit objects (HVAC Modes)<br>• three 1 bit objects |
|---|---|

Specify which **mode** the device should be in **after programming or a reset** (e.g. power failure, reset of the line via the bus) (default).

| Mode after reset | • Comfort<br>• Standby<br>• Eco<br>• Building protection |
|---|---|

The control can be deactivated and reactivated by an object from the bus. Configure which value of the on/off object should be used to switch the temperature control **on or off**.

| | |
|---|---|
| Behaviour of the on/off object with value | • 1 = On \| 0 = Off<br>• 0 = On \| 1 = Off |

Set the value of the on/off object after a reset.

| | |
|---|---|
| Value of the on/off object after reset | 0 • 1 |

The device sends the current **control variables** of the control system to the bus. Set the cases in which transmission takes place. Periodic sending is safer, in case a message does not reach a recipient. You may also set up periodical monitoring by the actuator with this setting.

| | |
|---|---|
| Send control variable | • on change<br>• on change and periodically |

When sending on change, the control variables are sent to the bus as soon as they change by the percentage set here.

| | |
|---|---|
| from change (in % absolute) | 1...10; 2 |

With sending periodically, the control variables are sent to the bus in a fixed cycle that can be set here.

| | |
|---|---|
| Cycle<br>*(if sent periodically)* | 5 s • ... • 5 min • ... • 2 h |

The status of the control variable is defined as 0% = OFF and >0% = ON. This status is sent to the bus and may for example be used for visualisation, or to switch off the heating pump as soon as the heating is switched off.

| | |
|---|---|
| Send status objects | • on change<br>• on change to 1<br>• on change to 0<br>• on change and periodically<br>• on change to 1 and periodically<br>• on change to 0 and periodically |

With sending periodically, the status objects are sent to the bus in a fixed cycle that can be set here.

| | |
|---|---|
| Cycle<br>*(if sent periodically)* | 5 s • ... • 5 min • ... • 2 h |

Then define the **type of control**. Heating and/or cooling may be controlled in two levels.

| | |
|---|---|
| Type of control | • Single level heating<br>• Dual-level heating<br>• Single-level cooling<br>• Single-level heating + single-level cooling<br>• Dual-level heating + single-level cooling<br>• Dual-level heating + dual-level cooling |

## General setpoint values

Determine if manually or via the bus modified the modified set point values should be kept after a mode change, or if they should reset to the standard specified here.

| | |
|---|---|
| Keep modified set points after mode change | No • Yes |

You may enter separate setpoint values for each mode or use the comfort setpoint as a basic value.
If you are using the control for both heating *and* cooling, you may also select the setting "separately with switching object". Systems used for cooling in summer and for heating in winter can thus be switched from one to the other.
If no switching object is selected, the actual temperature determines whether heating or cooling takes place. If the actual value is between the heating and cooling setpoint values, the existing operating mode is retained. If heating was previously used, the system remains in heating mode and continues to aim for this set point value. Only when the cooling setpoint is reached does the operating mode switch to cooling.
If cooling has been used so far, the system remains in cooling mode and continues to aim for this set point value. Only when the heating setpoint is reached does the operating mode switch to heating.
If the actual temperature is above the cooling set point value, cooling takes place; if it is below the heating set point value, heating takes place. The difference between the heating set point value and the cooling set point value or the dead zone should be at least 1 °C. This prevents the control from switching too often between heating and cooling in the event of minor temperature fluctuations.
If you are using the basic value, only the deviation from the comfort setpoint value is listed for the other modes (e.g, 2°C less for standby mode).

| | |
|---|---|
| Setting the setpoint values | • with separate setpoint values with Switching object<br>• with separate setpoint values without Switching object<br>• with comfort setpoint as a basis with Switching object<br>• with comfort setpoint as a basis without Switching object |

If the switching object is used, set two parameters for this object: Configure at which value of the switching object **heating or cooling** should take place.

| | |
|---|---|
| Behaviour of the switching object at value *(with switching object)* | • 0 = Heating \| 1 = Cooling<br>• 1 = Heating \| 0 = Cooling |

Set the value of the switching object after a reset.

| | |
|---|---|
| Value of the switching object after reset<br>*(with switching object)* | 0 • 1 |

If the set point values are set without a switching object, there is a status object. This sends information to the bus as to whether the heating or cooling setpoints are currently being used.
Predefine the **increment** for the setpoint changes.

| | |
|---|---|
| Increment for setpoint changes<br>(in 0.1 °C) | 1... 50; 10 |

The control may be reset to comfort mode from eco mode, which is used as night mode, via the **comfort extension**. This allows the user to maintain the comfort setpoint value for a longer time, e.g. when having guests. To do this, place a comfort extension switch on a visualisation or a push-button, for example. The duration of this comfort extension period is set. After the comfort extension period expires, the system returns to eco mode.
Whether the change only remains active temporarily (do not save) or remains saved after voltage recovery (and programming) has already been defined in the first section of "General control".

| | |
|---|---|
| Comfort extension time in seconds<br>(can only be activated from eco mode) | 1...36000; 3600 |

## Comfort Setpoint

Comfort mode is usually used for daytime mode when people are present. A starting value is defined for the comfort setpoint as well as subsequently a temperature range in which the setpoint value may be modified.

| | |
|---|---|
| Starting heating/cooling setpoint (in 0.1 °C) valid until 1st communication<br>*(not upon saving the setpoint value after programming)* | -300...800; 210 |

***If setpoint values are entered separately:***

Minimum value that can be set via object.

| | |
|---|---|
| Min. object value heating/cooling<br>(in 0.1 °C) | -300...800; 160 |

Maximum value that can be set via object.

| | |
|---|---|
| Max. object value heating/cooling<br>(in 0.1 °C) | -300...800; 280 |

***If the comfort setpoint value is used as a basis:***

Minimum value that can be set via object.

| | |
|---|---|
| Minimum base setpoint (in 0.1°C) | -300...800; 160 |

Maximum value that can be set via object.

| | |
|---|---|
| Maximum base setpoint (in 0.1°C) | -300…800; 280 |

If the comfort setpoint value is used as a basis, the reduction of the value is set.

| | |
|---|---|
| Reduction by up to (in 0.1°C) | 0…200; 50 |

If the comfort setpoint value is used as a basis, the increment of the value is set.

| | |
|---|---|
| Increase by up to (in 0.1°C) | 0…200; 50 |

If the comfort setpoint is used as the basis without a switching object, a dead zone is specified for the control mode "heating *and* cooling" to avoid direct switching from heating to cooling.

| | |
|---|---|
| Dead zone between heating and cooling *(only if both heating AND cooling are used)* | 1…100; 50 |

## Standby setpoint

Standby mode is usually used for daytime mode when people are absent.

***If setpoint values are entered separately:***

A starting setpoint value is defined as well as subsequently a temperature range in which the setpoint value may be changed.

| | |
|---|---|
| Starting heating/cooling setpoint (in 0.1 °C) valid until 1st communication | -300…800; 210 |

Minimum value that can be set via object.

| | |
|---|---|
| Min. object value heating/cooling (in 0.1 °C) | -300…800; 160 |

Maximum value that can be set via object.

| | |
|---|---|
| Max. object value heating/cooling (in 0.1 °C) | -300…800; 280 |

***If the comfort setpoint value is used as a basis:***

If the comfort setpoint value is used as a basis, the reduction of the value is set.

| | |
|---|---|
| Reduce heating setpoint (in 0.1°C) *(for heating)* | 0…200; 30 |

If the comfort setpoint value is used as a basis, the increment of the value is set.

| | |
|---|---|
| Increase cooling setpoint (in 0.1°C) *(for cooling)* | 0…200; 30 |

## Eco setpoint

Eco mode is usually used for night mode.

***If setpoint values are entered separately:***

A starting setpoint value is defined as well as subsequently a temperature range in which the setpoint value may be changed.

| | |
|---|---|
| Starting heating/cooling setpoint (in 0.1 °C) valid until 1st communication | -300…800; 210 |

Minimum value that can be set via object.

| | |
|---|---|
| Min. object value heating/cooling (in 0.1 °C) | -300…800; 160 |

Maximum value that can be set via object.

| | |
|---|---|
| Max. object value heating/cooling (in 0.1 °C) | -300…800; 280 |

***If the comfort setpoint value is used as a basis:***

If the comfort setpoint value is used as a basis, the reduction of the value is set.

| | |
|---|---|
| Reduce heating setpoint (in 0.1°C) *(for heating)* | 0…200; 50 |

If the comfort setpoint value is used as a basis, the increment of the value is set.

| | |
|---|---|
| Increase cooling setpoint (in 0.1°C) *(for cooling)* | 0…200; 60 |

## Setpoint values for frost/heat protection (building protection)

The building protection mode is for example used as long as windows are opened for ventilation or during longer absences (e.g. school buildings during holidays). Setpoints for frost protection (heating) and heat protection (cooling) are determined which may not be modified from outside (no access via operating devices etc.). The building protection mode may be activated with delay, which allows you to leave the building before the controls switch to frost/heat protection mode.

Predefine the setpoint for frost protection (heating).

| | |
|---|---|
| Setpoint frost protection (in 0.1°C) | -300…800; 70 |

Set the activation delay.

| | |
|---|---|
| Activation delay | less than • 5 s • ... • 5 min • ... • 2 h |

Predefine the setpoint for heat protection (cooling).

| | |
|---|---|
| Setpoint heat protection (in 0.1°C) | -300…800; 350 |

Set the activation delay.

| | |
|---|---|
| Activation delay | less than • 5 s • ... • 5 min • ... • 2 h |

## General control variables

This setting appears for the control types "Heating *and* Cooling" only.

Here, you can decide whether to use a common control variable for heating and cooling.

| | |
|---|---|
| For heating and cooling | • separate control variables are used<br>• common control variables are used for Level 1<br>• common control variables are used for Level 2<br>• common control variable are used for Level 1+2 |

Specify whether the control variable is used for a 4/6-way valve. Then applies:
0%...100% heating = 66%...100% control variable
OFF = 50% control variable
0%...100% cooling = 33%...0% control variable

| | |
|---|---|
| Use control variable for 4/6-way valve<br>*(only for common control variables in level 1)* | No • Yes |

Set the control mode of the 2nd level.

| | |
|---|---|
| Control type<br>*(for level 2 only)* | • 2-point-control<br>• PI control |

Select whether the control variable of the 2nd level is a 1 bit object (on/off) or an 8 bit object (on with percentage/off).

| | |
|---|---|
| Control variable of the 2nd Level is on<br>*(only for level 2 with 2 point controlling)* | • 1 bit object<br>• 8 bit object |

## 7.5.1. Heating control level 1/2

If a heating control mode is configured, one or two setting sections for the heating levels are displayed.

In the 1st level, heating is controlled by a PI control, which allows to either enter control parameters or select predetermined applications.

In the 2nd level (therefore only in case of 2-level heating), heating is controlled via a PI or a 2-point-control.
In level 2, the setpoint difference between the two levels must also be specified, i.e. below which setpoint deviation the second level is added.

| | |
|---|---|
| Setpoint difference between 1st and 2nd level (in 0.1°C)<br>*(for level 2)* | 0...100; 40 |
| Control type<br>*(for level 2, no common control variables)* | • 2-point-control<br>• PI control |

| | |
|---|---|
| Control variable is a<br>*(for level 2 with 2-point controlling, no common control variables)* | • 1 bit object<br>• 8 bit object |

Select whether the parameters for the PI control are entered individually or predefined by fixed parameters for frequent applications.

| | |
|---|---|
| Control type | • **PI control** |
| Setting of the controller by | • **Controller parameter**<br>• specified applications |

***PI control with control parameters:***

This setting allows individual input of the parameters for PI control.

| | |
|---|---|
| Control type | • **PI control** |
| Setting of the controller by | • **Controller parameter**<br>• specified applications |

Specify the deviation from the setpoint value at which the maximum control variable value is reached, i.e. the point at which maximum heating power is activated.

| | |
|---|---|
| Maximum control variable is reached at setpoint/actual difference of (in °C) | 1...5 |

The reset time shows how quickly the controller responds to deviations from the setpoint value. In case of a short reset time, the control responds with a fast increase of the control variable. In case of a long reset time, the control responds somewhat less urgently and needs longer until the necessary control variable for the setpoint value deviation is reached.
You should set the time appropriate to the heating system at this point (observe manufacturer's instructions).

| | |
|---|---|
| Reset time (in min.) | 1...255; 30 |

***PI control with predetermined application:***

This setting provides fixed parameters for frequent applications.

| | |
|---|---|
| Control type | • **PI control** |
| Setting of the controller by | • Controller parameter<br>• **specified applications** |

Select the appropriate application.

| | |
|---|---|
| Application | • Warm water heating<br>• Floor heating<br>• Convection unit<br>• Electric heating |
| Maximum control variable is reached at setpoint/actual difference of (in °C) | Warm water heating: 5<br>Floor heating: 5<br>Convection unit: 4<br>Electric heating: 4 |
| Reset time (in min.) | Warm water heating: 150<br>Floor heating: 240<br>Convection unit: 90<br>Electric heating: 100 |

***2-point-control (only level 2):***

2-point-control is used for systems which are only set to ON or OFF.

| | |
|---|---|
| Control type<br>*(is determined at a higher level for common control variables)* | • **2-point-control** |

Enter the switching distance that prevents frequent on/off switching of temperatures in the threshold range.

| | |
|---|---|
| Switching distance (in 0.1°C) | 0...100; 20 |

If separate control variables are used, select whether the control variable of the 2nd level is a 1 bit object (on/off) or an 8 bit object (on with percentage/off).

| | |
|---|---|
| Control variable is a | • 1 bit object<br>• 8 bit object |

Specify the value of the 8 bit object.

| | |
|---|---|
| Value (in %)<br>*(for 8 bit object)* | 0...100 |

## 7.5.2. Cooling control level 1/2

If a cooling control mode is configured, one or two setting sections for the cooling levels are displayed.

In the 1st level, cooling is controlled by a PI control in which either control parameters can be entered or predetermined applications can be selected.

In the 2nd level (therefore only for 2-level cooling), cooling is controlled via a PI or a 2-point-control.

In level 2, the setpoint deviation between the two levels must also be specified, i.e. above which setpoint value deviation the second level is added.

| | |
|---|---|
| Setpoint difference between 1st and 2nd level (in 0.1°C)<br>*(for level 2)* | 0...100; 40 |
| Control type<br>*(for level 2, no common control variables)* | • 2-point-control<br>• PI control |
| Control variable is a<br>*(for level 2 with 2-point controlling, no common control variables)* | • 1 bit object<br>• 8 bit object |

Select whether the parameters for the PI control are entered individually or predefined by fixed parameters for frequent applications.

| | |
|---|---|
| Control type | • **PI control** |
| Setting of the controller by | • **Controller parameter**<br>• specified applications |

***PI control with control parameters:***

This setting allows individual input of the parameters for PI control.

| | |
|---|---|
| Control type | • **PI control** |
| Setting of the controller by | • **Controller parameter**<br>• specified applications |

Specify the deviation from the setpoint value which reaches maximum variable value, i.e. the point at which maximum cooling power is activated.

| | |
|---|---|
| Maximum control variable is reached at setpoint/actual difference of (in °C) | 1...5 |

The reset time shows how quickly the controller responds to deviations from the set-point value. In case of a short reset time, the control responds with a fast increase of the control variable. In case of a long reset time, the control responds somewhat less urgently and needs longer until the necessary control variable for the setpoint value deviation is reached. You should set the time appropriate to the cooling system at this point (observe manufacturer's instructions).

| | |
|---|---|
| Reset time (in min.) | 1...255; 30 |

***PI control with predetermined application:***

This setting provides fixed parameters for a cooling ceiling

| | |
|---|---|
| Control type | • **PI control** |
| Setting of the controller by | • Controller parameter<br>• **specified applications** |
| Application | • Cooling ceiling |
| Maximum control variable is reached at setpoint/actual difference of (in °C) | Cooling ceiling: 5 |
| Reset time (in min.) | Cooling ceiling: 30 |

***2-point-control (only level 2):***

2-point-control is used for systems which are only set to ON or OFF.

| | |
|---|---|
| Control type<br>*is determined at a higher level for common variables* | • **2-point-control** |

Enter the switching distance that prevents frequent on/off switching of temperatures in the threshold range.

| | |
|---|---|
| Switching distance (in 0.1°C) | 0...100; 20 |

If separate control variables are used, select whether the control variable of the 2nd level is a 1 bit object (on/off) or an 8 bit object (on with percentage/off).

| | |
|---|---|
| Control variable is a | • 1 bit object<br>• 8 bit object |

Specify the value of the 8 bit object.

| | |
|---|---|
| Value (in %)<br>*(for 8 bit object)* | 0...100 |

## 7.5.3. Fan Coil Control

The fan coil control enables the regulation of the fan of convector heating/cooling systems.

Activate the fan coil control.

| | |
|---|---|
| Use fan coil control | No • **Yes** |

In fan coil control, the fan is automatically controlled by one or, in multi-level systems, several control variables for heating or cooling. Select which actuating variable(s) are to control the output. The available options depend on the type of heating/cooling control and the settings made for the actuating variables.

| | |
|---|---|
| Output is controlled via actuating variable | • Heating 1<br>• Heating 2<br>• Cooling 1<br>• Cooling 2<br>• Heating 1 and cooling 1<br>• Heating 2 and cooling 1<br>• Heating 1 and cooling 2<br>• Heating 2 and cooling 2 |

Select whether the first fan level should also be on when the second and third level are running.

| | |
|---|---|
| Switch Level 1 on also with Level 2 and 3 | No • Yes |

Select whether the second fan level should also be on when the third level is running.

| | |
|---|---|
| Switch Level 2 on also with Level 3 | No • Yes |

Set which mode is to be active after a reset.
In automatic mode, the fan coil level depends on the controller actuating variable:
Controller actuating variable 0 % ≙ Fan coil level 0
Controller actuating variable 1...33 % ≙ Fan coil level 1
Controller actuating variable 33...66 % ≙ Fan coil level 2
Controller actuating variable 66...100 % ≙ Fan coil level 3

| | |
|---|---|
| Mode after reset | • Manual<br>• Automatic (e.g. controller actuating variable) |

## 7.6. Temperature PI control – Controller extension unit

Activate the control

| | |
|---|---|
| Use control | No • Yes |

Determine if this device should take over the temperature control (stand-alone controller), or if the device should act as an extension for operating another controller.

| | |
|---|---|
| Intended as a | • Stand-alone controller<br>• **Controller extension** (for operating a stand-alone controller only) |

The settings for the 'controller extension' option are described below. For configuration as a stand-alone controller, please see Chapter *Temperature PI control – Independent controller*, page16.

The **mode** may be switched with two 8 bit objects of different priority. The objects are:
„... HVAC mode (Prio 2)“ for switching in everyday operation and
„... HVAC mode (Prio 1)“ for central switching with higher priority.
The objects are coded as follows:
0 = Auto
1 = Comfort
2 = Standby
3 = Eco
4 = Building Protection

Alternatively, you can use three objects, with one object switching between eco and standby mode and the two others activating comfort mode and frost/heat protection mode respectively. The comfort object blocks the eco/standby object, and the frost/heat protection object has the highest priority. The objects are:
„... Mode (1: Eco, 0: Standby)“,
„... comfort activation mode" and
„... frost/heat protection activation mode"

| | |
|---|---|
| Switch mode via | • two 8-bit objects (HVAC modes)<br>• three 1-bit objects |

Set the behaviour of the on/off object.

| | |
|---|---|
| Behaviour of the on/off object with value | • 1 = On \| 0 = Off<br>• 0 = On \| 1 = Off |

Set the type of control.

| | |
|---|---|
| Type of control | • Single stage heating<br>• Dual-stage heating<br>• Single-stage cooling<br>• Single-stage heating + single-stage cooling<br>• Dual-stage heating + single-stage cooling<br>• Dual-stage heating + dual-stage cooling |

You may enter separate set point values for each mode or use the comfort set point as a basic value. If you are using the basic value, only the deviation from the comfort set point value is listed for the other modes (e. g., 2 °C less for standby mode).

| | |
|---|---|
| Setting the nominal values | • separately<br>• with comfort set point as a basis |

If heating or cooling, specify how the status object should be analysed.

| | |
|---|---|
| Analysis of the status object | • 0 = Heating 1 = Cooling<br>• 1 = Heating \| 0 = Cooling |

Set in which mode the set point values can be changed.

| | |
|---|---|
| Comfort | No • Yes |
| Standby | No • Yes |
| Eco | No • Yes |
| Protection | No |

Activate fan coil control if a fan is used for heating/cooling.

| | |
|---|---|
| Use fan coil control | No • Yes |

Select whether the first fan level should also be on when the second and third level are running.

| | |
|---|---|
| Switch Level 1 on also with Level 2 and 3 | No • Yes |

Select whether the second fan level should also be on when the third level is running.

| | |
|---|---|
| Switch Level 2 on also with Level 3 | No • Yes |

Select when the controller information is to be read at the latest.

| | |
|---|---|
| Read controller information after | 5 ... 60; 10 sec. at the latest |

Select whether the nominal values are to be changed when the +/- keys are operated.

| | |
|---|---|
| Changing and transmitting nominal values when operating the +/- keys | No • Yes |

Set the grading for the set point changes.

| | |
|---|---|
| Grading for set point changes *(when nominal values are changed during operation)* | 1 ... 50; 5 in 0.1 °C |

## 7.7. LEDs

Set the brightness of the LEDs.

| | |
|---|---|
| Brightness | 0 ... 100%; 30% |

Set whether the two objects no. 86 (All LEDs on/off) and no. 87 (All LEDs brightness) are used. This allows the LEDs to be switched on or off via the bus and their brightness to be set.

| | |
|---|---|
| Use objects | No • Yes |

Depending on the object value, the display is switched on (1) or off (0) after a reset.

| | |
|---|---|
| Object value on/off after reset (*if objects are used*) | 0 • 1 |

The LEDs can switch off after a touch surface has been touched. This means that the device does not light up permanently, but only when being operated.
Activate the function if required:
Set whether the LEDs switch off automatically when a touch surface is touched.

| | |
|---|---|
| Use automatic switch-off after pressing a button | No • Yes |

Set how long it takes for the display to turn off after operation.

| | |
|---|---|
| Switch off after *(is automatic switch off is used)* | 1 ... 255; 30 secs. after pressing |

Select what the temperature display shows here.

| | |
|---|---|
| Temperature display | • display actual value only<br>• display only target value/base shift<br>• display actual value and target value/base shift |

## 7.8. Logic

The device has 8 logic inputs, 2 AND and 2 OR logic gates.

| | |
|---|---|
| Use logic inputs | Yes • No |

For each logic input, the object value can be assigned before the first communication, which is used for the initial commissioning and when the voltage returns.

| | |
|---|---|
| Object value prior to first communication for | |
| - Logic input 1... 8 | 0 • 1 |

Select which logic gate should be used.

**AND logic**

| | |
|---|---|
| AND Logic 1 / 2 | not active • active |

**OR logic**

| | |
|---|---|
| OR Logic 1 / 2 | not active • active |

## 7.8.1. AND logic 1/2 and OR logic outputs 1/2

Select a switch event.

| | |
|---|---|
| 1 / 2 / 3 / 4 Input | • Do not use<br>• Logic inputs 1...8<br>• Logic inputs 1...8 inverted<br>• Temperature sensor malfunction = ON<br>• Temperature sensor malfunction = OFF<br>*only with OR logic:*<br>• Switching output AND logic 1/2<br>• Switching output AND logic 1/2 inverted |

Each logic output can transmit one 1-bit or two 8-bit objects.

| | |
|---|---|
| Output type | • a 1-bit-object<br>• sends two 8-bit objects |

If the **output type is a 1-bit-object**, set the output values.

| | |
|---|---|
| Output value<br>if logic = 1 | 1 • 0 |
| Output value<br>if logic = 0 | 1 • 0 |
| Output value<br>if block active | 1 • 0 |
| Output value if<br>monitoring time exceeded | 1 • 0 |

If the **output type is two 8-bit-objects**, first set the object type.

| | |
|---|---|
| Object type | • value (0…255)<br>• Percent (0…100%)<br>• Angle (0…360°)<br>• Scenario call-up (0…63) |

Then set the output values.

| | |
|---|---|
| Output value object A<br>if logic = 1 | 0 ... 255 / 100% / 360° / 63; 1 |
| Output value object B<br>if logic = 1 | 0 ... 255 / 100% / 360° / 63; 1 |
| Output value object A<br>if logic = 0 | 0 ... 255 / 100% / 360° / 63; 0 |
| Output value object B<br>if logic = 0 | 0 ... 255 / 100% / 360° / 63; 0 |
| Output value object A<br>if block active | 0 ... 255 / 100% / 360° / 63; 0 |
| Output value object B<br>if block active | 0 ... 255 / 100% / 360° / 63; 0 |
| Output value object A if<br>monitoring time exceeded | 0 ... 255 / 100% / 360° / 63; 0 |
| Output value object B if<br>monitoring time exceeded | 0 ... 255 / 100% / 360° / 63; 0 |

Set the cases in which the logic output is to be sent to the bus.

| | |
|---|---|
| Send behaviour | • upon a change of logic<br>• upon a change of logic to 1<br>• upon a change of logic to 0<br>• upon a change of logic and periodically<br>• upon a change of logic to 1 and periodically<br>• upon a change of logic to 0 and periodically<br>• upon a change of logic+object receipt<br>• upon a change of logic+object receipt and cyclically |

When sending periodically, the logic object is sent to the bus in a fixed cycle that can be set.

| | |
|---|---|
| Send cycle<br>*(if sent periodically)* | 5 s • 10 s • ... • 2 h |

## Blocking

With the help of the blocking object, the output can be blocked, e.g. by a manual command (push button).

| | |
|---|---|
| Use block | No • Yes |

The block can take effect at value 0 or 1, depending on the intended use.

| | |
|---|---|
| Assessment of the block object | • At value 1: block \| At value 0: release<br>• At value 0: block \| At value 1: release |

Specify an object value until first communication.

| | |
|---|---|
| Blocking object value before first communication | 0 • 1 |

The behaviour of the output during blocking can be set.

| | |
|---|---|
| Output behaviour on blocking | • Do not send message<br>• Send block value [see above, output value if block active] |

The behaviour of the output on release, i.e. when the lock is removed, can be set.

| | |
|---|---|
| on release (with 2 second release delay) | • Do not send message<br>• transmit value for current logic status |

## Monitoring

If necessary, activate the input monitoring.
The input monitoring is a safety function that periodically needs a live message at input. A ratio of 1:3 is recommended here as a monitoring period.
Example: Monitoring period 30 min, input communication object(s) should receive a message every 10 min.

| | |
|---|---|
| Use input monitoring | No • Yes |

Set the inputs to be monitored.

| | |
|---|---|
| Input monitoring | • 1 • 2 • 3 • 4<br>• 1 + 2 • 1 + 3 • 1 + 4 • 2 + 3 • 2 + 4 • 3 + 4<br>• 1 + 2 + 3 • 1 + 2 + 4 • 1 + 3 + 4 • 2 + 3 + 4<br>• 1 + 2 + 3 + 4 |

Set the monitoring period.

| | |
|---|---|
| Monitoring period | 5 s • ... • 2 h; 1 min |

The behaviour of the output can be set if the monitoring period is exceeded.

| | |
|---|---|
| Output behaviour on exceeding the monitoring time | • Do not send message<br>• Send value exceeding [= value of the parameter "Output value if monitoring time exceeded"] |

## Questions about the product?

You can reach the technical service of Elsner Elektronik under
**Tel. +49 (0) 70 33 / 30 945-250** or
**service@elsner-elektronik.de**

We need the following information to process your service request:

- Type of appliance (model name or item number)
- Description of the problem
- Serial number or software version
- Source of supply (dealer/installer who bought the device from Elsner Elektronik)

For questions about KNX functions:

- Version of the device application
- ETS version used for the project

elsner

**Elsner Elektronik GmbH** Control and Automation Engineering

Sohlengrund 16
75395 Ostelsheim
Germany

Phone +49 (0) 70 33 / 30 945-0 info@elsner-elektronik.de
Fax +49 (0) 70 33 / 30 945-20 www.elsner-elektronik.de