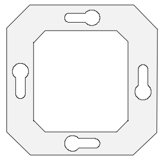
KNX Room temperature controller



Sensor

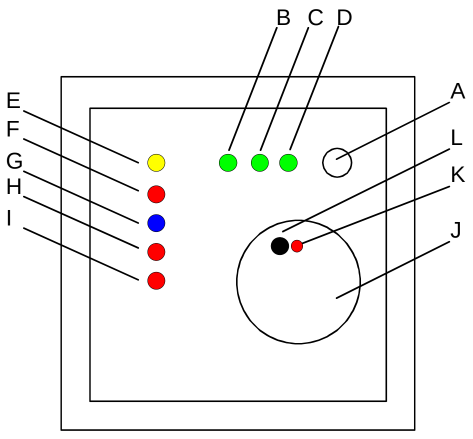
Product name:	KNX Room temperature controller
Design:	flush-mounting device (FMD)
Article. no.	2178
ETS search path:	Heating, ventilation, air conditioning / controller / continuous-action controller
Status:	08.11.2006

Functions:

Room temperature controller functions:
The room temperature controller can be used to control the temperature of individual rooms.
Depending on the control option of the current temperature-setpoint and the room temperature, an actuating variable for the heating or cooling control system can be transmitted to the KNX / EIB.

The room temperature can optionally be sensed by the internal temperature sensor or by an external temperature sensor connected to the terminal strip of the push-button interface.
As a supplement to basic heating or cooling, an additional stage can also be activated. The temperature setpoint difference between the basic and the additional stage can be set in a parameter. For larger deviations between the setpoint and actual temperature value, the room can therefore be heated up or cooled down more quickly by switching on the additional stage. The basic and the additional stage can have different control algorithms assigned to them.

The controller can operate in 5 operating modes (comfort, standby, night, frost/ heat protection and controller disable) each having their own temperature setpoints for the heating mode or cooling mode. For the heating and cooling functions continuous-action or switching PI or switching 2-point control characteristics can be selected.

Illustration:	Dimensions:	Controls:
 	Width: 60 mm Height: 60 mm Depth: 50 mm Overall device dimensions depending on design variant.	A: Presence key B: Status LED green, comfort mode C: Status LED green, standby mode D Status LED green, night mode E Status-LED yellow, energy supply F Status LED red, heating G: Status LED blue, cooling H: Status LED red, frost/heat protection I: Status LED red, dew-point J: Control knob for setpoint adjustment K: programming LED under control knob L: programming key, under control knob

Technical data	
Type of protection:	IP 20
Safety class:	III
Mark of approval:	KNX / EIB
Ambient temperature:	-5 °C ... +45 °C
Storage / transport temperature:	-25 °C ... +70 °C (storage above +45 °C reduces the lifetime)
Mounting position:	any
Minimum distances:	none
Type of fastening:	The connection insert module with its supporting ring is fastened with screws in the flush-mounting box. The electronic module is plugged into the insert module.
KNX supply	
Voltage:	21 – 32 V DC
Power consumption:	typically 150 mW
Connection:	bus connecting terminal (KNX type 5.1)
External supply	---
room temperature controller (internal temperature sensor):	
Measuring range:	0 °C … + 40 °C ±1 %
Resolution:	0.1 K
Air humidity:	0 % ... 95 % (no condensation)
Response to bus voltage failure	
Bus voltage only:	all object values will be deleted. room temperature controller: no response, control off pushbutton interface no response (outputs switched off)
Mains voltage only:	---
Bus and mains voltage:	---
Response on return of voltage	
Bus voltage only:	room temperature controller: the controller is initialized; depending on parameterization, different temperature values and the status will be transmitted and the switch-over objects will be updated. pushbutton interface the behaviour of the inputs and outputs can be parameterized.
Mains voltage only:	---
Bus and mains voltage:	---

Software description

ETS search path:
Heating, ventilation, air conditioning / controller / continuous-action controller Jung

BAU used:	ASIC 1065 + µC (mask version 0705 light)	
KNX/EIB type class:	3b - dev. with cert. PhL + stack	
Configuration:	S-mode standard	
PEI type:	- Hex	- Dec
PEI connector	no connector	

Application:

No.	Short description:	Name:	Version:
1	Room temperature controller	Continuous-action controller 705E10 (ETS 3) Continuous-action controller 705E01 (ETS 2)	1.0

Application:	Continuous-action controller with pushbutton interface, 4-fold 705C			
Executable from mask version:	7.1			
Number of addresses (max):	120	**dynamic table handling**	Yes	No
Number of assignments (max):	120	**maximum length of table**	120	
Communication objects:	59			

Room temperature controller functions:

Function: Actual temperature

Object	Function	Name	DPT-ID	Type	Flag
23	Actual temperature	RTC output	9.001	2 bytes	C, R, T

Function: additional temperature sensor

Object	Function	Name	DPT-ID	Type	Flag
24	External temperature sensor	RTC input	9.001	2 bytes	C, W, T

Function: basic setpoint preset

Object	Function	Name	DPT-ID	Type	Flag
26	Basic setpoint	RTC input	9.001	2 bytes	C, W

Function: Operating mode switch-over

With operating mode switch-over "via value (1 byte)":

Object	Function	Name	DPT-ID	Type	Flag
28	Operating mode switch-over	RTC input	20.102	1 byte	C, W(, T)[6]
32	Operating mode forcing object	RTC input	20.102	1 byte	C, W

With operating mode switch-over "via switching (4 x 1 bit)":

Object	Function	Name	DPT-ID	Type	Flag
28	Comfort mode	RTC input	1.001	1 bit	C, W(, T)6
29	Standby mode	RTC input	1.001	1 bit	C, W(, T)6
30	Night mode	RTC input	1.001	1 bit	C, W(, T)6
31	Frost/ heat protection	RTC input	1.001	1 bit	C, W(, T)6

Presence object and window status:

Object	Function	Name	DPT-ID	Type	Flag
33	Presence object	RTC input / output	1.001	1 bit	C, W, T
34	Window status	RTC input	1.019	1 bit	C, W

Function: Control option switch-over

Object	Function	Name	DPT-ID	Type	Flag
35	Heating / cooling switch-over [7]	RTC input	1.001	1 bit	C, W, (T)

[6]: Optionally, the "T" flags can be set for the operating mode switch-over. Once the flags are set, the object values which have changed according to the newly set operating mode will be actively transmitted to the bus.

[7]: This object is only visible in mixed operation "heating and cooling" or "basic / additional heating / cooling". The "T" flag is set for automatic heating / cooling switch-over.

Function: Status indication

Object	Function	Name	DPT-ID	Type	Flag
36	Controller status	RTC output	---	1 byte	C, T
36	Controller status, frost alarm	RTC output	1.001	1 bit	C, T
36	Controller status, heating / cooling	RTC output	1.001	1 bit	C, T
36	Controller status, comfort mode	RTC output	1.001	1 bit	C, T
36	Controller status, night mode	RTC output	1.001	1 bit	C, T
36	Controller status, controller disabled	RTC output	1.001	1 bit	C, T
36	Controller status, controller inactive	RTC output	1.001	1 bit	C, T
36	Controller status, frost / heat protection	RTC output	1.001	1 bit	C, T
36	Controller status, standby mode	RTC output	1.001	1 bit	C, T
37	Heating message	RTC output	1.001	1 bit	C, T
38	Cooling message	RTC output	1.001	1 bit	C, T

Function: Disabling function (room temperature controller)

Object	Function	Name	DPT-ID	Type	Flag
40	Disable controller	RTC input	1.001	1 bit	C, W
41	Disable additional stage [8]	RTC input	1.001	1 bit	C, W

Function: Actuating variable heating
no additional stage activated /
For mixed operation: Actuating variable output "heating" and "cooling" via separate objects:

Object	Function	Name	DPT-ID	Type	Flag
42	Actuating variable heating	RTC output	5.001	1 byte	C, W, T
42	Actuating variable heating (PWM)	RTC output	1.001	1 bit	C, W, T
42	Actuating variable heating	RTC output	1.001	1 bit	C, W, T

no additional stage activated /
For mixed operation: Actuating variable output "heating" and "cooling" via shared object:

Object	Function	Name	DPT-ID	Type	Flag
42	Actuating variable heating/cooling	RTC output	5.001	1 byte	C, W, T
42	Actuating variable heating/cooling (PWM)	RTC output	1.001	1 bit	C, W, T
42	Actuating variable heating/cooling	RTC output	1.001	1 bit	C, W, T

additional stage activated /
For mixed operation: Actuating variable output "heating" and "cooling" via separate objects:

Object	Function	Name	DPT-ID	Type	Flag
42	Actuating variable	RTC output	5.001	1 byte	C, W, T
42	Actuating variable basic heating (PWM)	RTC output	1.001	1 bit	C, W, T
42	Actuating variable basic heating	RTC output	1.001	1 bit	C, W, T
43	Actuating variable additional heating	RTC output	5.001	1 byte	C, W, T
43	Actuating variable additional heating (PWM)	RTC output	1.001	1 bit	C, W, T
43	Actuating variable additional heating	RTC output	1.001	1 bit	C, W, T

[8]: This object is only visible when additional stage is activated.

Additional stage activated /
For mixed operation: Actuating variable output "heating" and "cooling" via shared object:

Object		Function	Name	DPT-ID	Type	Flag
	42	Actuating variable basic stage	RTC output	5.001	1 byte	C, W, T
	42	Actuating variable basic stage (PWM)	RTC output	1.001	1 bit	C, W, T
	42	Actuating variable basic stage	RTC output	1.001	1 bit	C, W, T
	43	Actuating variable additional stage	RTC output	5.001	1 byte	C, W, T
	43	Actuating variable additional stage (PWM)	RTC output	1.001	1 bit	C, W, T
	43	Actuating variable additional stage	RTC output	1.001	1 bit	C, W, T

Function: Actuating variable cooling
no additional stage activated /
For mixed-mode: Actuating variable output "heating" and "cooling" via separate objects:

Object		Function	Name	DPT-ID	Type	Flag
	44	Actuating variable cooling	RTC output	5.001	1 byte	C, W, T
	44	Actuating variable cooling (PWM)	RTC output	1.001	1 bit	C, W, T
	44	Actuating variable cooling	RTC output	1.001	1 bit	C, W, T

additional stage activated /
For mixed-mode: Actuating variable output "heating" and "cooling" via separate objects:

Object		Function	Name	DPT-ID	Type	Flag
	44	Actuating variable basic cooling	RTC output	5.001	1 byte	C, W, T
	44	Actuating variable basic cooling (PWM)	RTC output	1.001	1 bit	C, W, T
	44	Actuating variable basic cooling	RTC output	1.001	1 bit	C, W, T
	45	Actuating variable additional cooling	RTC output	5.001	1 byte	C, W, T
	45	Actuating variable additional cooling (PWM)	RTC output	1.001	1 bit	C, W, T
	45	Actuating variable additional cooling	RTC output	1.001	1 bit	C, W, T

Function: Actuating variable status indication heating [9]

Object		Function	Name	DPT-ID	Type	Flag
	46	PWM actuating variable heating	RTC output	5.001	1 byte	C, W, T
	46	PWM actuating variable basic heating	RTC output	5.001	1 byte	C, W, T
	47	PWM actuating variable additional heating	RTC output	5.001	1 byte	C, W, T

Function: Actuating variable status information cooling [9]

Object		Function	Name	DPT-ID	Type	Flag
	48	PWM actuating variable cooling	RTC output	5.001	1 byte	C, W, T
	48	PWM actuating variable basic cooling	RTC output	5.001	1 byte	C, W, T
	49	PWM actuating variable additional cooling	RTC output	5.001	1 byte	C, W, T

Function: Setpoint temperature

Object		Function	Name	DPT-ID	Type	Flag
	50	Setpoint temperature	RTC output	9.001	2 byte	C, T, R

Function: Controller extension:

Object		Function	Name	DPT-ID	Type	Flag
	52	Setpoint shift feedback	RTC output	6.010	1 byte	C, T, R
	53	Setpoint shift preset	RTC input	6.010	1 byte	C, W

Function: Controller status indication additional stage

Object		Function	Name	DPT-ID	Type	Flag
	57	Status report additional stage	RTC output	---	1 byte	C, T

Function: Actual temperature not adjusted

Object		Function	Name	DPT-ID	Type	Flag
	58	Actual temperature not adjusted	RTC output	9.001	2 byte	C, T

Independent of the setting of the parameter "Temperature sensing" (internal sensor, external sensor or internal and external sensor), the function shows the non-adjusted (and non-weighted) actual value of the internal temperature sensor.

Object description

Objects for the room temperature controller:

	23	Actual temperature:	2-byte object for transmission of the actual temperature (room temperature) as measured and varied by a controller or a controller extension. (possible range of values: -99.9 °C ... +99.9 °C / Measuring range of internal temperature sensor: 0 °C ... + 40 °C (1 %)
	24 sensor	External temperature	2-byte object for connection of an external room temperature sensor or a controller extension (via "actual temperature" object). (possible range of values: -99.9 °C ... +99.9 °C)
	26	Basic setpoint:	2-byte object for external preset of basic setpoint. Depending on heating/cooling, the possible range of values is limited by the parameterized frost protection and/or heat protection temperature. The received value is mathematically rounded off to half °C!
	28	Operating mode switch-over:	1-byte object for switch-over of the controller’s operating modes acc. to KONNEX.
	28	Comfort operation:	1-bit object for switch-over into the "Comfort" operating mode.
	29	Standby operation:	1-bit object for switch-over into the "Standby" operating mode.
	30	Night-time operation:	1-bit object for switch-over into the "Night" operating mode.
	31	Frost/ heat protection:	1-bit object for switch-over into the "Frost/heat protection" operating mode.
	32	Forced-control object operating mode:	1-byte object for superordinated forced control of the controller’s operating modes acc. to KONNEX.

Object description (continued)

Objects:

No.	Name	Description
33	Presence object:	1-bit object (bi-directional) which transmits the status of the presence key - to the bus after pressing or which can be used for connection of a presence detector. (presence detected = "1", presence not detected = "0")
34	Window status:	1-bit object for the connection of window contacts. (window open = "1", window closed = "0")
35	Heating / cooling change-over:	1-bit object for switching over between control options "heating" and "cooling, if not done by the controller automatically (object value 1: heating; object value 0: cooling). In case of automatic switch-over the active control option can be transmitted (parameter-dependent).
36	Controller status:	1-byte object for general status feedback
36	Controller status:	1-bit object for individual status feedback of parameterizable functions of the controller (frost alarm, heating/cooling, comfort mode, night mode, standby mode, controller disabled, controller inactive, frost/heat protection).
37	Message heating:	1-bit object for the controller to indicate a request for heating energy (object value = "1": energy request, object value = "0": no energy request).
38	Message cooling:	1-bit object for the controller to indicate a cooling energy request (object value = "1": energy request, object value = "0": no energy request).
40	Controller disable:	1-bit object for deactivating the controller (activation of dew-point operation). (controller deactivated = "1", controller activated = "0")
41	Additional stage disable:	1-bit object for deactivating the additional stage of the controller. (additional stage deactivated = "1", additional stage activated = "0")
42	Actuating variable heating:	1-byte object for the output of the continuous actuating variable for heating operation.
42	Actuating variable heating:	1-bit object for the output of the continuous actuating variable for heating operation.
42	Actuating variable heating	1-bit object for the output of the continuous PWM actuating variable for heating operation.
42	Actuating variable basic heating:	1-byte object for the output of the continuous actuating variable for basic heating operation.
42	Actuating variable basic heating:	1-bit object for the output of the continuous actuating variable for basic heating operation.
42	Actuating variable basic heating (PWM) :	1-bit object for the output of the continuous PWM actuating variable for basic heating operation.
42	Actuating variable heating/cooling:	1-byte object for the output of the continuous actuating variable for heating or for cooling operation. (via shared object if actuating variables are output)
42	Actuating variable heating/cooling:	1-bit object for the output of the switching variable for heating or for cooling operation. (via shared object if actuating variables are output)
42	Actuating variable heating/cooling (PWM):	1-bit object for the output of the PWM actuating variable for heating or for cooling operation. (via shared object if actuating variables are output)

Object description (continued)

Objects:

	42	Actuating variable basic heating and cooling:	1-byte object for the output of the continuous actuating variable either for basic heating or for basic cooling operation (via shared object if actuating variables are output)
	42	Actuating variable basic heating and cooling:	1-bit object for the output of the switching variable either for basic heating or for basic cooling operation (via shared object if actuating variables are output)
	42	Actuating variable basic heating and cooling: (PWM):	1-bit object for the output of the PWM actuating variable either for basic heating or for basic cooling operation (via shared object if actuating variables are output)
	43	Actuating variable additional heating:	1-byte object for the output of the continuous actuating variable for additional heating operation.
	43	Actuating variable additional heating:	1-bit object for the output of the switching variable for heating operation.
	43	Actuating variable additional heating (PWM):	1-bit object for the output of the PWM actuating variable for additional heating operation
	43	Actuating variable additional heating and cooling:	1-byte object for the output of the continuous actuating variable either for additional heating or cooling operation (via shared object if actuating variables are output)
	43	Actuating variable additional heating and cooling:	1-bit object for the output of the switching variable either for additional heating or cooling operation (via shared object if actuating variables are output)
	43	Actuating variable additional heating and cooling: (PWM):	1-bit object for the output of the PWM actuating variable either for additional heating or cooling operation (via shared object if actuating variables are output)
	44	Actuating variable cooling:	1-byte object for the output of the continuous actuating variable for cooling operation.
	44	Actuating variable cooling:	1-bit object for the output of the switching variable for heating operation.
	44	Actuating variable cooling	1-bit object for the output of the PWM actuating variable for cooling operation
	44	Actuating variable basic cooling:	1-byte object for the output of the continuous actuating variable for basic cooling operation
	44	Actuating variable basic cooling:	1-bit object for the output of the switching variable for basic cooling operation
	44	Actuating variable basic cooling (PWM):	1-bit object for the output of the PWM actuating variable for basic cooling operation
	45	Actuating variable additional cooling:	1-byte object for the output of the continuous actuating variable for additional cooling operation
	45	Actuating variable additional cooling:	1-bit object for the output of the switching variable for additional heating operation.
	45	Actuating variable additional cooling (PWM):	1-bit object for the output of the PWM actuating variable for additional cooling operation.
	46	PWM actuating variable	1-byte object with PWM actuating variable for status feedback of the actuating variable value for heating operation
	46	PWM actuating variable Basic heating:	1-byte object with PWM actuating variable for status feedback of the continuous actuating variable value for basic heating operation
	47	PWM actuating variable Additional heating:	1-byte object with PWM actuating variable for status feedback of the continuous actuating variable value for additional heating..
	48	PWM actuating variable	1-byte object with PWM actuating variable for status feedback of the continuous actuating variable value for cooling operation
	48	PWM actuating variable Basic cooling:	1-byte object with PWM actuating variable for status feedback of the continuous actuating variable value for basic cooling operation
	49	PWM actuating variable Additional cooling:	1-byte object with PWM actuating variable for status feedback of the continuous actuating variable value for additional cooling operation

Object description (continued)

Objects:

⇥	50	Setpoint temperature:	2-byte object for the output of the current temperature. Depending on the control option, the possible range of values is limited by the parameterized frost protection and/or heat protection temperature.
⇤	50	Setpoint temperature:	2-byte object for receiving the current temperature setpoint of a controller.
⇥	52	Feedback setpoint shift:	1-byte object for current setpoint shift feedback x ≤ 0 ≤ y (0 = no active shifting); integers The possible range of values (x to y) is fixed by the setting of the upper and lower limits for the setpoint (parameterizable) in combination with the step value (0.5 °C).
⇤	53	Preset setpoint shift	1-byte object for presetting a basic setpoint shift, e.g. via a controller extension. x ≤ 0 ≤ y (0 = no active shifting); integers The possible range of values (x to y) is fixed by the setting of the upper and lower limits for the setpoint (parameterizable) in combination with the step value (0.5 °C). In case the limits of the value range are exceeded by the preset external value, the controller will automatically reset the received value to the minimum and maximum limits.
⇤	57	Additional status feedback	1-byte object for general additional status feedback
⇤	58	Actual temperature not adjusted	2-byte object for the output of the actual temperature (room temperature) as measured and not adjusted by the controller. (possible range of values: -99.9 °C ... +99.9 °C / Measuring range of internal temperature sensor: 0 °C to + 40 °C ±1 %)

Scope of functions

Room temperature controller functions:

General

- 5 operating modes: Comfort, standby, night, frost/heat protection and controller disable
- Operating modes switch-over via 1-byte object according to KONNEX or individual 1-bit objects.

- **Heating/cooling system**
  - Control options: heating", "cooling", "heating and cooling" each with or without additional stage.
  - PI control (continuous or switching PWM) or 2-state control (switching) adjustable as control algorithms.
  - Continuous (1-byte) or switching (1-bit) actuating variable output.
  - Control parameter for PI controller (if desired: proportional range, integral-action time) and 2-state controller (hysteresis) presettable.

- Setpoint values
  - Each operating mode can be assigned its own temperature setpoints (for heating and/or cooling).
  - The setpoints for the additional stage are derived via a parameterizable stage offset from the values of the basic stage.
  - Setpoint value shifting by local operation on device or via communication objects.

- **Functions**
  - Automatic or object-oriented switch-over between "heating" and "cooling".
  - The controller operation can optionally be disabled via an object.
  - Parameterizable duration of the comfort mode extension.
  - Complete (1-byte) or partial (1-bit) status information parameterizable and transmissible to the bus via an object.
  - Deactivation of the control or of the additional stage via different objects possible.

- Room temperature measurement
  - Internal and external room temperature sensor available.
  - Internal to external determination of measured value with enabled external sensor.
  - Request interval of external temperature sensor adjustable.
  - The actual and setpoint temperature can be output to the bus if a parameterizable deviation is detected (also cyclically).
  - The room temperature measurement (actual value) can be adjusted separately for the internal and external sensor via parameter.
  - Frost/heat protection switch-over depending on window state (delayed detection possible) and automatic frost protection.
  - Temperature alarm with upper and lower temperature limit possible. Telegram activation via two separate objects.

Actuating variable output

- Separate or combined actuating variable output via one or two objects in "heating and cooling" mode
- Normal or inverted actuating variable output parameterizable
- Automatic transmission and cycle time for actuating output parameterizable

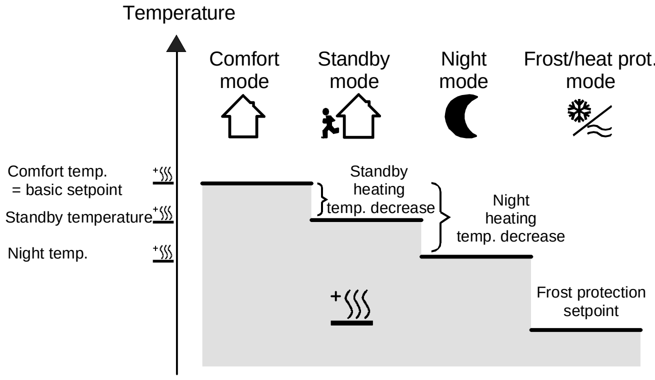
Table of contents

Functional description

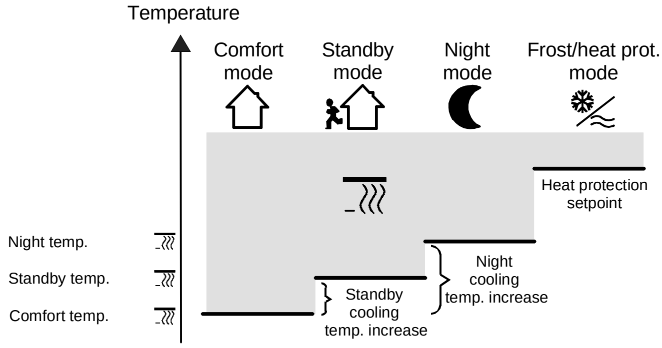
1 General room temperature controller functions

The room temperature controller supports the three control options "heating", "cooling" and "heating and cooling". In all three control options, the controller can work in different operating modes to which different setpoint temperatures are assigned. The following diagrams show these setpoint temperatures and their graduations.

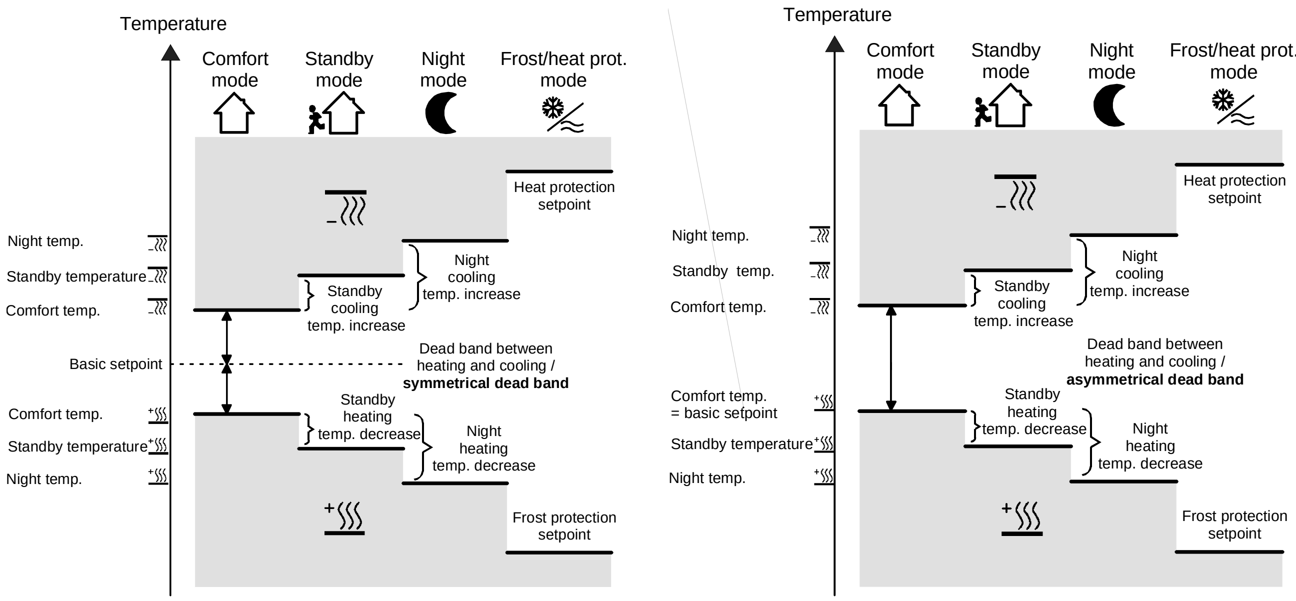
"Heating"



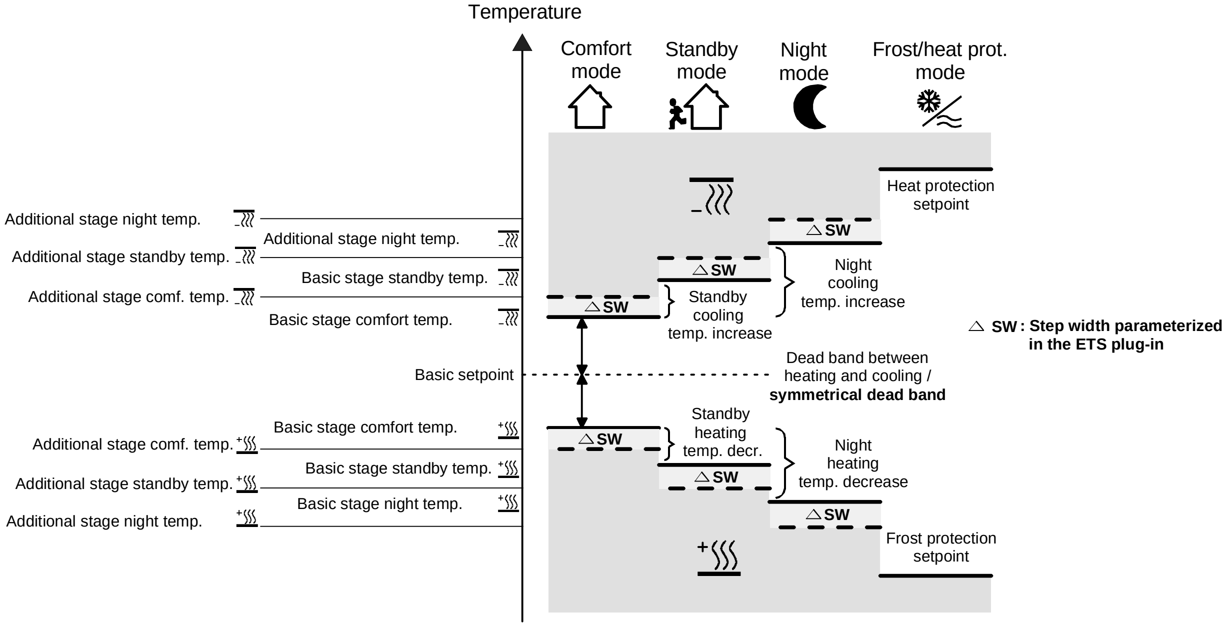
"Cooling"



"Heating and cooling"



Temperature control with additional stage considering as an example "heating and cooling" with symmetrical dead zone…



If enabled in the ETS, 6 temperature setpoints can be varied in "heating and cooling" control option. Depending on the temperature decrease, increase or dead zone parameterized in the ETS, all temperature setpoints are derived from the basic setpoint temperature.
It must be pointed out that changing the setpoint temperature for heating in the comfort mode will also change all other setpoint temperature values!

The dead zone (temperature zone for which there is neither heating nor cooling) is the difference between the setpoint temperatures for "heating" and "cooling" in the comfort mode. The following applies:

$$T_{\text{comfort setpoint cooling}} - T_{\text{comfort setpoint heating}} = T_{\text{dead zone}}; \; T_{\text{comfort setpoint cooling}} \geq T_{\text{comfort setpoint heating}}$$

Important notes:

- If the dead zone is symmetrical, the basic setpoint is indirectly set via the comfort temperature for heating.
- Changing the comfort setpoint temperature for cooling allows the adjustment of the dead zone. An adjustment of the dead zone with a symmetrical dead zone position will result in a shift of the comfort setpoint temperature for heating and thus of all other temperature setpoints. With an asymmetrical dead zone position, an adjustment of the comfort setpoint temperature for cooling will only shift the temperature setpoints for cooling. It is possible to shift the dead zone to 0 °C via local control ($T_{\text{comfort setpoint cooling}} = T_{\text{comfort setpoint heating}}$). In this case there is neither heating nor cooling, if the determined room temperature equals the comfort setpoint temperatures.

The setpoint temperatures for "Standby" and "Night" are derived from the comfort setpoint temperatures for heating or cooling. The temperature increase (for cooling) and the temperature decrease (for heating) of both operating modes can be preset in the ETS.

It is possible to adjust the setpoint temperatures for "Standby" and "Night" via local control in the programming mode independent of the values for the temperature increase/decrease which were originally parameterized in the ETS.

In this case, the standby or night setpoint temperatures will always shift together with the temperature increase/decrease resulting from the local control during the adjustment of the basic setpoint temperature or the dead zone. After the reprogramming with the ETS, the originally parameterized values can be accepted again.

The following applies:

$$T_{\text{standby setpoint heating}} \leq T_{\text{comfort setpoint heating}} \leq T_{\text{comfort setpoint cooling}} \leq T_{\text{standby setpoint cooling}}$$

or

$$T_{\text{night setpoint heating}} \leq T_{\text{comfort setpoint heating}} \leq T_{\text{comfort setpoint cooling}} \leq T_{\text{night setpoint cooling}}$$

In case of a two-stage control the setpoints of the additional stage are always derived dynamically from the setpoints of the basic stage. The temperature setpoints of the additional stage are predefined by the stage offset which is parameterized in the ETS. The stage offset cannot be adjusted in the local control mode.

As far as a change of the basic setpoint temperature is concerned (when a new comfort setpoint temperature value for heating is being received by communication object no. 26), there basically two cases to be distinguished:

- Case 1: The basic setpoint adjustment is permanently adopted,
- Case 2: The basic setpoint adjustment is only temporarily adopted (default).

Via the "Adopt basic temperature setpoint permanently" parameter on the "Room temperature controller function /setpoints" parameter page, it is possible to determine whether the changed basic temperature value shall be stored in memory permanently ("Yes") or only temporarily ("No").

Case 1:

If the basic temperature setpoint is changed, it will be permanently stored in the room temperature controller's EEPROM. The newly adjusted value will overwrite the basic temperature originally parameterized with the ETS!

It should be noted, however, that frequent adjustments of the basic temperature (e.g. several times a day) can affect the product life of the device as the non-volatile memory is designed only for less frequent write access.

Thus the basic setpoint received by the object remains in memory even after a bus voltage failure.

Case 2:

The basic setpoint received via the object stays only temporarily active in the current operating mode. In case of a bus voltage failure or following a switch-over into another operating mode (e.g. comfort followed by standby), the basic setpoint set via local control or received via the object will be discarded and replaced by the value which was originally parameterized in the ETS.

Notes:

- Since the setpoint temperatures for the "standby" and "night" operating modes or the setpoints for the "cooling" control option are derived - in consideration of the increase, decrease or dead zone values that are parameterized in the ETS - from the basic setpoint temperature for "heating", these setpoint temperatures will shift linearly by the change of the basic setpoint value.
  The temperature setpoints for the standby or night mode or "cooling" comfort mode (dead zone) will always be stored in the non-volatile EEPROM.
- It has to be pointed out that temperature setpoints can only be changed or stored via local control or via the "Basic setpoint" object, if it was enabled in the ETS. Any value preset via local control will not be adopted by the object.

2 Room temperature controller functions

2.1 Operating modes

The room temperature controller features several operating modes. By selecting theses modes it is possible to activate different temperature setpoints that, for example, depend on the presence of a person, the status of the heating or cooling system, the time of day or day of week.

- Comfort mode:

  The comfort mode should be activated if people are present in the room that requires the room temperature to be adjusted to a comfortable and appropriate value. The switch-over into this operating mode can also take place via presence control.
  The comfort mode when activated is signalled by LED B (🏠).

- Standby mode

  If a room is not in use during the day as people are absent, the standby mode may be activated. This will set the room temperature to a standby value thus saving heating or cooling energy in the process.
  The standby mode when activated is signalled by LED C (**Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.**).

- Night mode

  During the night hours or during a longer absence it is often best to adjust the room temperature to cooler temperatures for heating systems (e.g. in bedrooms). In this case cooling systems can be adjusted to higher temperature values, if climate control is not required (e.g. in offices). For this purpose the night mode can be activated.
  The night mode when activated is signalled by LED D (**Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.**).

- Frost / heat protection mode

  Frost protection is necessary, if, for example, the room temperature must not fall below critical values when the window is open. Heat protection might be necessary, if the temperature in a mostly warm environment becomes too high due to external influences.
  In these cases a freezing or overheating of the room can be prevented by activating the frost/heat protection depending on the adjusted "heating" or "cooling" control option by specifying an individual temperature setpoint.
  A frost/heat protection when activated is signalled by LED H (❄).

- Comfort mode extension (temporary comfort mode)

  The comfort mode extension is to be activated from the night mode or the frost/heat protection (not triggered by the "window state" object) and can be used to adjust the room temperature to the comfort temperature for a certain amount of time, if, for example the room 'is used' during the night as well. The extension is activated exclusively by a parameterized presence key. The comfort mode extension is automatically deactivated after a settable time has elapsed or by pressing the presence key again or via receiving a presence object value = "0". The extension cannot be retriggered.

An individual temperature setpoint can be preset for each "heating" or "cooling" control option.

2.1.1 Operating mode switch-over

There are several ways to activate or switch-over the operating modes. Activating or switching-over – interdependent in terms of priority – are possible via…

a) local operation of the presence key, if enabled,

c) the 1-bit objects that are available separately for each operating mode or alternatively via the KONNEX objects (1 byte).

Ad a):
If the presence key has been selected for presence detection on parameter page "Controller functions", the presence key can be used to switch from the night mode or from the frost/heat protection mode over to the comfort mode for the preset comfort extension time on provision that the above modes have not been activated by the "Window state" object. The comfort mode extension is deactivated after this time has elapsed, after a new press on the presence key or after receiving a presence object value = "0". If the duration of the comfort extension is set to "0", the presence function can be activated, but the operating mode is not changed. During the comfort extension period, the comfort LED is lit up together with the "night mode" or the "frost/heat protection" LED.

If the standby mode is active, it is possible to switch into the comfort mode by actuating the presence key or via a presence object value = "1".

Ad b):
One distinguishes whether the operating mode is to be switched-over via separate 1-bit objects or, alternatively, via the 1-byte KONNEX objects. The "*Operating mode switch-ove*r" parameter on the "*Room temperature controller function*" parameter page predefines how the switch-over will take place.

• Operating mode switch-over via "switching (4 x 1 bit):

There is a separate 1-bit switch-over object for each operating mode. Each one of these objects allows to switch-over or to preset the current operating mode by priority.

Taking into consideration the priority, the following switch-over hierarchy results from an operating mode switch-over via the objects. One distinguishes between presence detection by presence key (table 1 / figure 1) and by presence detector (table 2 / figure 2 on next page):

Table 1

"Operating mode switch-over" objects: Obj.-No. 31	Obj.-No. 28	Obj.-No. 29	Obj.-No. 30	Window status Obj.-No. 34	Presence key object Obj.-No. 33	activated operating mode
X	X	X	X	1	X	Frost /heat protection
1	X	X	X	0	0	Frost /heat protection
0	1	X	X	0	0	Comfort
0	0	1	X	0	0	Standby
0	0	0	1	0	0	Night
1	X	X	X	0	1	Comfort extension
0	1	X	X	0	1	Comfort
0	0	1	X	0	1	Comfort
0	0	0	1	0	1	Comfort extension
0	0	0	0	0	0	last available mode
0	0	0	0	0	1	Comfort / comfort mode extension *

X = irrelevant
*: depends on the last available operating mode.

Fig. 1:

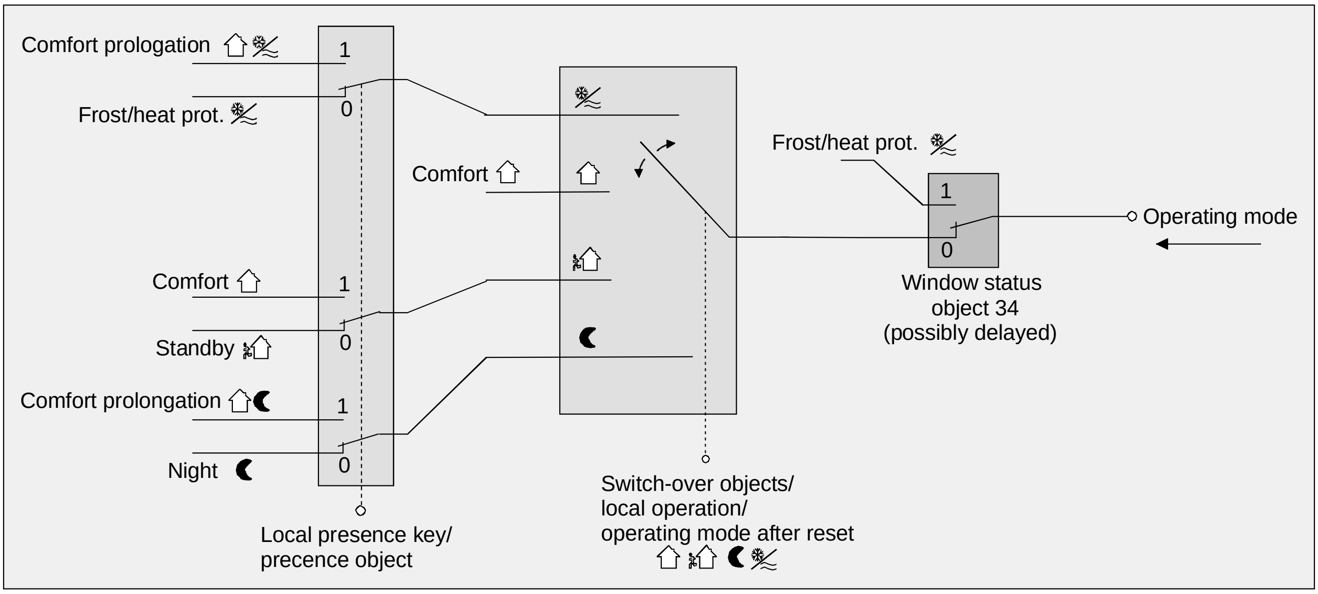
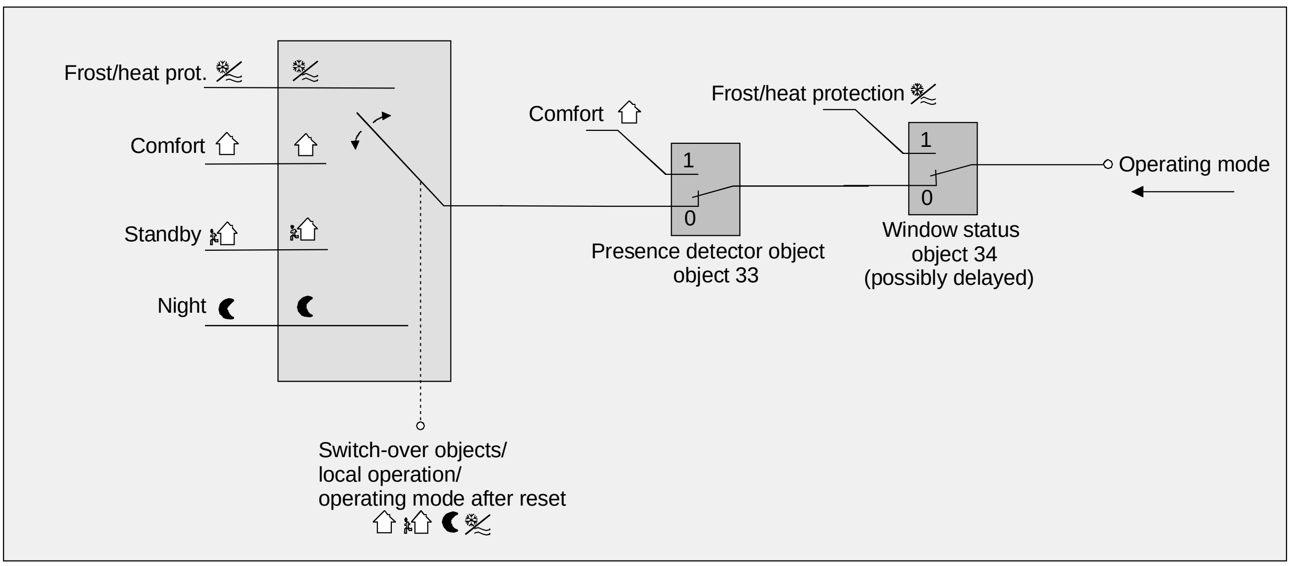


Table 2						
"Operating mode switch-over" objects: Obj.-No. 31	Obj.-No. 28	Obj.-No. 29	Obj.-No. 30	Window status Obj.-No. 34	Presence detector object Obj.-No. 33	activated operating mode
X	X	X	X	1	X	Frost /heat protection
X	X	X	X	0	1	Comfort
1	X	X	X	0	0	Frost /heat protection
0	1	X	X	0	0	Comfort
0	0	1	X	0	0	Standby
0	0	0	1	0	0	Night
0	0	0	0	0	0	last available mode

X = irrelevant

Fig. 2:



Notes on operating mode switch-over via "Switching" (4 x 1-bit):

- When the operating modes are switched-over, the objects, too, (comfort mode / standby mode / night mode / frost/heat protection) will always be updated and can, if applicable, be read out (set "read" flag!). Once the "transmission" flag is set for these objects, changed values will also be actively transmitted to the bus. Following a return of bus voltage or an initialization, the object corresponding to the set operating mode will be updated and its value actively transmitted to the bus when the "transmission" flag is set.
- When parameterizing a presence key:
  The presence object is active "1") for the duration of an activated comfort mode extension.
  The presence object will be automatically deleted ("0"), if the comfort mode extension is terminated after the extension time has elapsed or if the operating mode has been switched by a higher-priority control via the switch-over objects or via local operation.

- The operating mode switch-over via "value" (2 x 1-byte):

A shared 1-bit switch-over object exists for all operating modes. Via this value object, the operating mode can instantly be switched over after receiving only one telegram. The received value will determine the operating mode.
In addition, there is a second 1-byte object available which can (by forced control and higher ranking) set an operating mode independent of all other available switch-overs. Both 1-byte objects are implemented according to the KONNEX specification.

Taking into account the priorities there is the following switching hierarchy for an operating mode switch-over by objects, with a distinction being made between a presence detection via presence key (table 1 / figure 1) and by presence detector (table 2 / figure 2 on next page):

Table 1

"Operating mode switch-over" object Obj.-No. 28	"Forced object operating mode" object *** Obj.-No. 32	Window status Obj.-No. 34	Presence key object Obj.-No. 33	activated operating mode
X	01	X	X	Comfort
X	02	X	X	Standby
X	03	X	X	Night
X	04	X	X	Frost /heat protection
X	00	1	X	Frost /heat protection
01	00	0	0	Comfort
02	00	0	0	Standby
03	00	0	0	Night
04	00	0	0	Frost /heat protection
01	00	0	1	Comfort
02	00	0	1	Comfort
03	00	0	1	Comfort extension
04	00	0	1	Comfort extension
00	00	0	0	last available mode
00	00	0	1	Comfort / comfort mode extension *

*: depends on the last available operating mode. / X = irrelevant
**: Values greater than "04" will not be evaluated. A "00" value will leave the last available operating mode active.
***: Values greater than "04" will not be evaluated. A "00" value signifies a deactivated forced object.

Fig. 1:

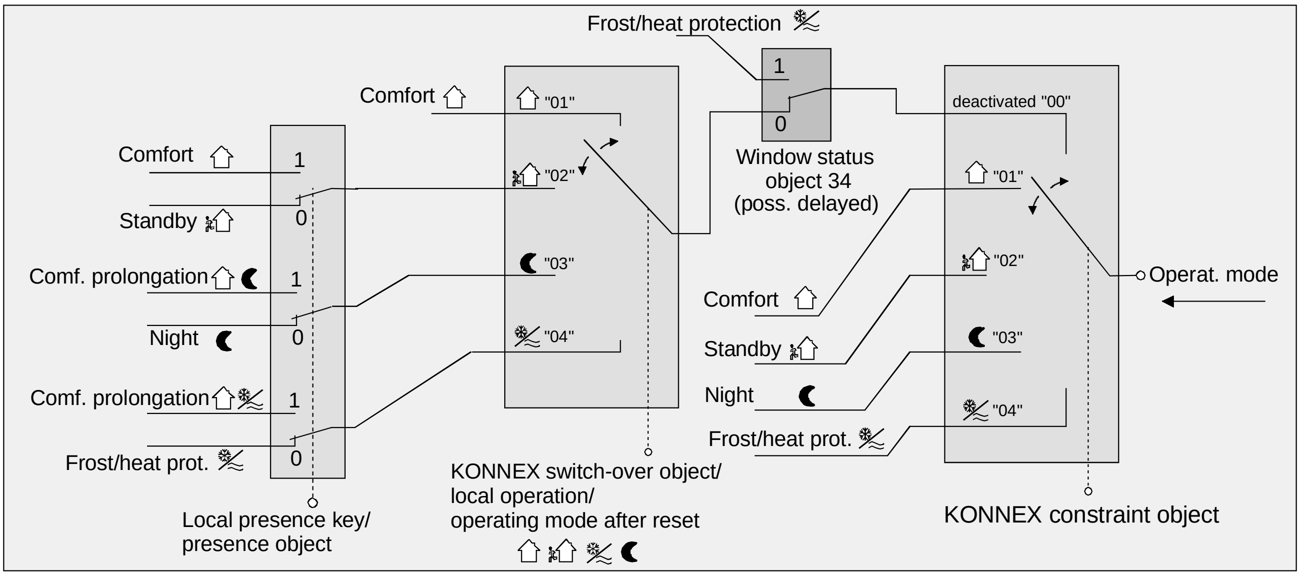


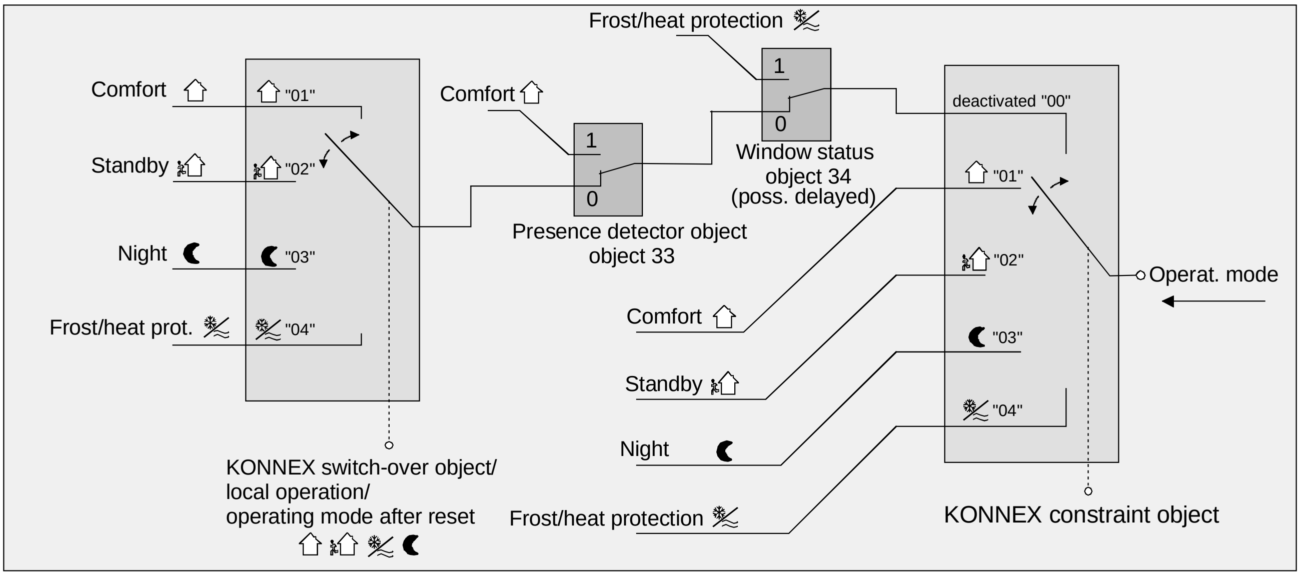
Table 2				
"Operating mode switch-over" object Obj.-No. 28	"Forced-control object operating mode" object *** Obj.-No. 32	Window status Obj.-No. 34	Presence detector object Obj.-No. 33	activated operating mode
X	01	X	X	Comfort
X	02	X	X	Standby
X	03	X	X	Night
X	04	X	X	Frost /heat protection
X	00	1	X	Frost /heat protection
X	00	0	1	Comfort
01	00	0	0	Comfort
02	00	0	0	Standby
03	00	0	0	Night
04	00	0	0	Frost /heat protection
00	00	0	0	last available mode

X = irrelevant

**: Values greater than "04" will not be evaluated. A "00" value will leave the last available operating mode active.

***: Values greater than "04" will not be evaluated. A "00" value signifies a deactivated forced object.

Fig. 2:



Notes for operating mode switch-over via "switching" (2 x 1-byte):

- Any operating mode switch-over will also update the KONNEX switch-over object and can be, if applicable, read out (set "read" flag!). If the "transmission" flag is set with this object, the current value will actively transmitted to the bus following a change. After a return of bus voltage or an initialization, the value corresponding to the adjusted operating mode will be actively transmitted on the bus if flag is set to "transmission". In case controller extensions are used, the "transmission" flag must also be set!
- When parameterizing a presence key:
  The presence object is active ("1") for the duration of an activated comfort mode extension.
  The presence object will automatically be deleted ("0"), if the comfort mode extension is terminated after the elapsed extension time, if the operating mode has been switched by a higher-priority control via the switch-over objects or local operation or if a forced operating mode has been deactivated via the KONNEX forced-control object (forced-control object → "00").

2.1.2 Notes on the operating modes

Presence function / comfort mode extension:

Via a presence detection the room temperature controller can switch into the comfort mode extension for a short time when a key is pressed or into the comfort mode if movement is detected. The "*Presence detection*" and "*Type of presence detection*" parameters on the "*Room temperature controller function – functions*" parameter page determine whether the presence detection is controlled by movement via the presence detector or manually by pressing the presence key:

- Presence detection via presence key:

  If the presence key is enabled as the type of presence detection, the setting "=*Presence key*" can be selected under key functions. In addition, the "*Presence object*" object 33 is enabled.

  That way, it is possible to switch into the comfort mode extension during activated night mode or frost/heat protection (not activated via the "window status" object) by actuating the presence key or via a presence object value = "1". The extension is automatically deactivated as soon as the parameterized "*Duration of comfort mode extension*" has elapsed. A comfort mode extension can be prematurely deactivated, if the presence key is pressed again or if a value = "0" is received by the object. Retriggering of the extension time is not possible.
  If the duration of the comfort mode extension is set to "0", it will not be possible to activate a comfort mode extension from the night mode or the frost/heat protection. In this case, the operating mode is not changed even though the presence function is activated.
  If the standby mode is active, it is possible to switch into the comfort mode by actuating the presence key or via a presence object value = "1". This will also be the case, if the duration of the comfort mode extension is parameterized to "0". The active mode remains active for as long as the presence function is activated or until there is another operating mode.

  The presence object or the presence function will always be deleted when switching over into another operating mode or after a forced-control operating mode has been deactivated (with KONNEX forced-control switch-over). The presence object is bi-directional ("W" and "T" flags set to default) so that an activation (= "1") or a deactivation (= "0") of the presence function will result in a transmission of telegrams with the corresponding object value. A presence function including the object that was activated before a reset will always be deleted after the reset.

- Presence detection by the presence detector:

  If a presence detector is enabled for presence detection purposes, only the "Presence object" object 33 will be visible. This object can be used to incorporate presence detectors in the room temperature control.
  If any movement is detected ("1" telegram), the controller will switch into the comfort mode. The presettings by the switch-over objects or via local control directly on the touch sensor itself are not relevant. Only a window contact or the automatic frost protection or the KONNEX forced-control object have a higher priority.
  After the delay time in the presence detector has elapsed ("0" telegram), the controller switches back into the mode which was active before the presence detection or it will track the telegrams of the switch-over objects received during the presence detection.
  Switching-over of the operating mode on the room temperature controller is not possible while the presence detection is active.
  A presence function that was activated before a reset will always be deleted after the reset. In this case the presence detector has to transmit a new "1" telegram in order to activate the presence function.

Window status / automatic frost protection:

The room temperature controller provides different ways of switching into the frost/heat protection . Besides switching by means of the corresponding operating mode switch-over object, the frost/heat protection can be activated by a window contact. Among these options, the window contact has the higher priority.

A telegram with the value = "1" (opened window) to object 34 will activate the frost/heat protection. In this case the operating mode cannot be deactivated by the operating mode switch-over objects (with the exception of the KONNEX forced-control object).

Only a telegram with the value = "0" (closed window) will reset the window status and deactivate the frost/heat protection. Subsequently, the operating mode that was set before the opening of the window or tracked via the bus during the time the window was open will be activated.

Operating mode after reset:

In the ETS it is possible to determine via the "*Operating mode after reset*" parameter on the "*room temperature controller function – functions*" parameter page which operating mode is to be activated following a return of bus voltage or a programming operation with the ETS. The following settings are possible:

- *"Comfort mode"*: After the initialization phase the comfort mode is activated.
- *"Standby mode"*: After the initialization phase the standby mode is activated.
- *"Night mode"*: After the initialization phase the night mode is activated.
- *"Frost/heat protection"*: After the initialization phase the frost/heat protection is activated.
- *Restore operating mode before reset":* The mode that was activated before a reset will be readjusted after the initialization phase of the device.

The objects associated with the activated operating mode will be updated after a reset.

Notes on the setting "*Restoring operating mode before reset*":

- Frequent adjustments of the operating mode during ongoing operation (e.g. several times a day) may affect the product life of the device as the non-volatile storage (EEPROM) is designed only for less frequent write access.
- A presence function including the object that was activated before a reset is deleted after the reset. The operating mode activated by the presence function, however, remains active after the reset.
  A comfort mode extension which was restarted by a reset is automatically deactivated after the extension time has elapsed.
  The "window status" object is deleted after a reset ("0"). In this case, too, the frost/heat protection which was previously activated via the window status remains activated even after a reset.

2.1.3 Controller status

The room temperature controller is able to transmit its status. Available is either a general collective status report (1-byte) or alternatively one of up to 8 individual status reports (1-bit).
The "*Status controller*" parameter on the "*Room temperature controller function – actuating variable and status output*" parameter page releases the status report and determines the status format:

- *"Status controller" = "controller general":*

The 1-byte status object 36 includes the complete status information. The status - controlled by the control algorithm – is actively transmitted (cyclically every 30 seconds) to the bus (pre-condition: "T" flag is set!)The setting of the "R" flag allows the read-out of the status.

Settings	**Relevance of data**	
Controller general 1-byte	Bit 0: 1: comfort mode active Bit 1: 1: standby mode active Bit 2: 1: night mode active Bit 3: 1: frost/heat protection active	Bit 4: 1: controller disabled Bit 5: 1: heating; 0: cooling Bit 6: 1: controller inactive (dead zone) Bit 7: 1: frost alarm ($T_{room} \leq + 5$ °C)

- *"Status controller" = "Transmit individual status":*

The 1-bit status object 36 includes the status information selected by the "Individual status" parameter. The status - controlled by the control algorithm – is actively transmitted (cyclically every 30 seconds) to the bus (pre-condition: "T" flag is set!)). The setting of the "R" flag allows the read-out of the status.

Parameterization for "Individual status"	**Relevance of data**	
Comfort mode active	1: comfort mode / extension active	0: no comfort mode
Standby mode active	1: standby mode active	0: no standby mode
Night mode active	1: Night mode active	0: no night mode
Frost/ heat protection active	1: frost/heat protection active	0: no frost/heat protection
Controller disabled	1: controller disabled (dew-point mode)	0: controller not disabled
Heating/cooling	1: heating operation	0: cooling operation
Controller inactive	1: controller inactive (dead zone)	0: controller active
Frost alarm	1: frost alarm (T room ≤ + 5 °C)	0: no frost alarm (T room > + 5 °C)

Meaning of status reports:

- Comfort operation: Active if the operating mode "comfort ' '" or a comfort mode extension " " or " " is activated.
- Standby operation: active if the "standby ' '" operating mode is activated.
- Night-time operation: active if the "night-time ' '" operating mode is activated.
- Frost/ heat protection: active if the "frost/heat protection ' '" operating mode is activated.
- Controller disabled: active if controller disable is activated (dew-point mode).
- Heating/cooling: active if the heating mode is activated and inactive if cooling mode is activated. (inactive with controller disabled.)
- Controller inactive: active with the "heating and cooling" control option when the measured room temperature lies within the dead zone. This status information is always "0" for the individual "*heating*" or "*cooling*" control options! (inactive if controller is disabled.)
- Frost alarm: active if the detected room temperature reaches or drops below + 5 °C. The status report has no significant influence on the controller behaviour.

The status object 36 will be updated following a reset after the initialization phase. Afterwards the status is updated every 30 seconds parallel to the calculation of the controller's actuating variable.

2.1.4 Additional controller status

The additional controller status is an object in which information already available on the bus is to be collected so that it can be displayed with a suitable device. This 1-byte object is a mere visualization object which does not allow any write access. This status object possesses a KNX-certified (but not standardized) datapoint type.

Settings for "Controller general"	Relevance of data for "Additional controller status report"	
Bit 0	1: normal operating mode	0: forced-control operating mode
Bit 1	1: comfort extension active	0: no comfort extension
Bit 2	1: presence (presence detector)	0: no presence (presence detector)
Bit 3	1: presence (presence key)	0: no presence (presence key)
Bit 4	1: window contact active	0: no window opened
Bit 5	1: additional stage active	0: additional stage not active
Bit 6	1: heat protection active (heat protection temp. < actual temp.)	0: no heat protection (heat protection temp. > actual temp.)
Bit 7	1: controller disabled (dew-point mode)	0: controller not disabled

The status object 57 will be updated following a reset after the initialization phase. Thereafter, the additional controller status is updated every 30 seconds parallel to the computation of the actuating variable of the controller. In the event of a change, the new status determined will be transmitted to the EIB.

2.2 Control options and control option switch-over

The room temperature controller features up to two control options. These control options determine whether the controller shall address heating systems (individual control option "*heating*") or cooling systems (individual control option "*cooling*") via its actuating variable. It is also possible to activate a mixed-mode in which the controller can switch-over automatically or, alternatively, object-controlled between "*heating*" and "*cooling*".
Moreover, the control operation can be carried out in two stages for addressing an additional heating and cooling device. If controlled in two stages, actuating variables will be calculated separately for the basic and additional stage and transmitted on the bus depending on the deviation between actual and setpoint temperature.
The "*Control option*" parameter on the "*Room temperature controller functions*" parameter page determines the control option to be carried out and activates, if applicable, the additional stage(s).

For the individual control options "*heating*" or "*cooling*" without additional stage, the controller runs with only one actuating variable. Alternatively, it runs with two actuating variables for the parameterized control option, if the additional stage is activated. Depending on the determined room temperature and the preset setpoint temperatures of the operating modes, the room temperature controller decides independently whether heating or cooling energy is required and calculates the actuating variable for the heating or cooling system.
Following a reset (return of bus voltage or new ETS programming) in the "*heating*" or "*cooling*" mode, the controller will always operate in the control option that was set in the ETS.

In the "*heating and cooling*" mixed-mode the controller is able to address heating and cooling systems. In this case, the switch-over behaviour of the control options can be preset:

- The *"Switching-over between heating and cooling"* parameter on parameter page "*Room temperature controller-functions*" is set to "automatic":

  Depending on the determined room temperature and the preset basic temperature setpoint value or the dead zone, the heating or cooling mode is automatically activated . If the room temperature lies within the preset dead zone, neither heating nor cooling will take place (both actuating variables = "0"). The room will be cooled down if the room temperature is higher than the temperature setpoint for cooling. The room will be heated up, if the room temperature is lower than the temperature setpoint for heating.

  With an automatic switch-over of the control option, the information can be actively transmitted to the bus via the "*Control option switch-over*" object 35. The *"Automatic transmission heating/cooling switch-over"* parameter determines when a control option switch-over is transmitted.

  - Setting "*Changing the control option*":
    In this case, a telegram is transmitted solely when switching-over from heating to cooling (object value = "0") or from cooling to heating (object value = "1).
  - Setting *"Changing the output variable"*:
    With this setting, the current control option will always be transmitted if the output variable has changed. With the actuating variable = "0" the control option that was last active will be transmitted.

  If the determined room temperature lies within the dead zone, the control option last activated will be retained in the object value until the controller is switched-over into the other control option.

  With an automatic switch-over, the object value can also be transmitted cyclically. The *"Cyclical heating/cooling switch-over"* parameter enables cyclical transmission (factor setting > "0") and determines the cycle time.

  Notes on the automatic switch-over of the control option:

  A dead zone that is too narrow might result in continuous switching over between heating and cooling. For this reason the dead zone (temperature difference between the set-temperatures for heating and cooling comfort mode) should preferably not be adjusted below the default value.

- The "*Switching-over between heating and cooling*" parameter on parameter page *"Room temperature controller-functions"* is set to *"via object"*:

  Independent of the dead zone, the control option is then controlled via the *"Control option switch-over"* object 35. This type of switch-over may be required, for example, in case of heating and cooling via a single-duct system (combined heating and cooling system). For this purpose, the temperature of the medium in the single-duct system must be changed beforehand by the system control. Afterwards, the control option is set via the object (often the single-duct system uses cold water for cooling during the summer, hot water for heating during the winter).
  The *"Control option switch-over"* object has the following polarity: "1": heating; "0": cooling. After a reset, the object value "0" and the *"Control option heating/ cooling after reset"* parameter will be activated.

  The *"Control option heating / cooling after reset"* parameter determines which control option will be activated after a reset. The *"heating"* or *"cooling"* setting causes the controller to activate the parameterized control option directly after the initialization phase. Setting the *"Control option before reset"* parameter will activate the control option that was selected before the reset.
  If a switch-over takes place via the control option object, the controller will first switch-over into a control option that was set after the reset. The controller will switch-over, if applicable, into the other control option only after the device has received an object update.

  Notes on the *"control option before reset"* setting:

  - Frequent adjustments of the control options during ongoing operation (e.g. several times a day) may affect the product life of the device as in this case the non-volatile storage (EEPROM) is only designed for less frequent write access.

Heating / cooling message:

Depending on the selected control option it is possible to output the information via separate objects whether heating or cooling energy is currently required, i.e. whether heating or cooling operation is in progress.
As long as the actuating variable for heating (cooling) is > "0", a "1" telegram is transmitted via the *"heating"* (*"cooling"*) signalling object. The message telegrams will be reset only if the actuating variables = "0" ("0" telegram will be transmitted).

Exception: In case of a 2-state control, the LEDs "heating" or "cooling" will light up or the signalling objects for heating and cooling will already become active as soon as the temperature falls short of the temperature setpoint in case of heating or exceeds the temperature setpoint in case of cooling. In this case, the parameterized hysteresis is not being considered.

Heating and cooling at the same time is not possible!

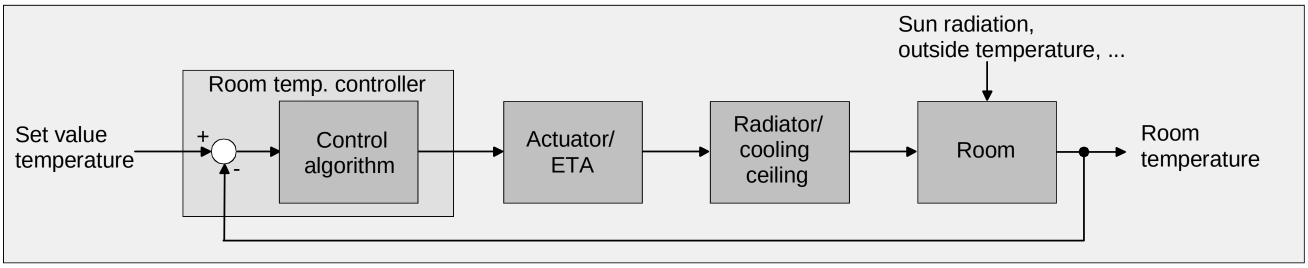
The signalling objects may be enabled via the *"Heating message"* or the *"Cooling message"* parameter on the *"Actuating variable and status output"* parameter page.

The signalling objects are controlled by the control algorithm. It has to be considered that the actuating variable is calculated only every 30 seconds (thus updating the signalling objects every 30 seconds).

2.3 Room temperature control and actuating variables

2.3.1 Control algorithms, control circuits and calculation of actuating variable

A comfortable temperature control for a living space requires a special control algorithm which controls the installed heating or cooling systems. By considering the setpoint temperatures as well as the actual room temperature, the controller determines the actuating variables controlling the heating or cooling system.
The control system (control circuit) consists of the room temperature controller, the servo drive or the switching actuator (for the use of electro-thermal drives), the actual heating or cooling element (e.g. heaters or cooling ceiling) and the room. This results in the following control process:



The controller measures the actual temperature (determined room temperature) and compares it with the given setpoint temperature. The control algorithm calculates the actuating variable from the difference between actual and setpoint temperature. This enables the controller to compensate for actual/setpoint temperature differences in the control circuit caused by external influences (e.g. strong exposure to sun or varying outside temperatures) by regularly readjusting the actuating variable. In addition, the flow temperature of the heating and cooling circuit affects the control process making it necessary to adapt the actuating variables.

The room temperature controller allows the option between a continuous or switching proportional/integral control (PI) or a switching 2-state control.

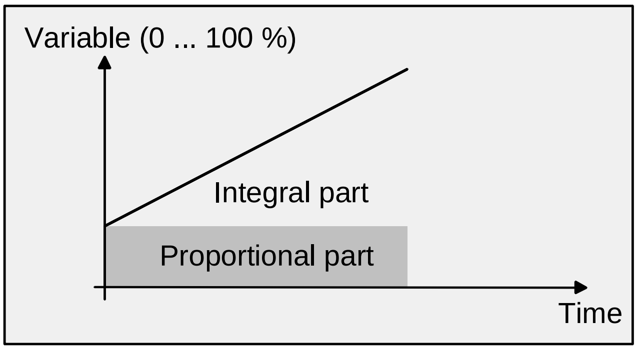
The actuating variables calculated by the control algorithm are output via the *"Actuating variable heating"* or *"Actuating variable cooling"* communication objects. The control algorithm selected for the heating and/or cooling operation determines, among other things, also the format of the actuating variable object. Thus, it is possible to create 1-bit or 1-byte actuating variable objects.

The control algorithm is determined via the *"Type of heating control"* or *"Type of cooling control"* parameters on the *"Room temperature controller function"* parameter page, possibly also for the additional stages.
Each of the following three algorithms can be selected:

1. Continuous PI control:

A PI control is an algorithm consisting of a proportional and an integral part. A combination of these control characteristics allows to accurately adjust the room temperature as fast as possible without or with only small control deviations.

This algorithm lets the room temperature controller calculate a new continuous actuating variable periodically every 30 seconds. This one will be output to the bus via a 1-byte value object, if the calculated actuating variable has changed by a predetermined percentage. The *"Automatic transmission if value changes by..."* parameter on the *"Room temperature controller function – actuating variable and status output"* parameter page determines the change interval in percent.



An additional PI controlled heating or cooling stage works exactly as the PI control of the basic stage. The difference is that the setpoint will shift by taking into account the parameterized stage offset.

Characteristic feature of the PI control:

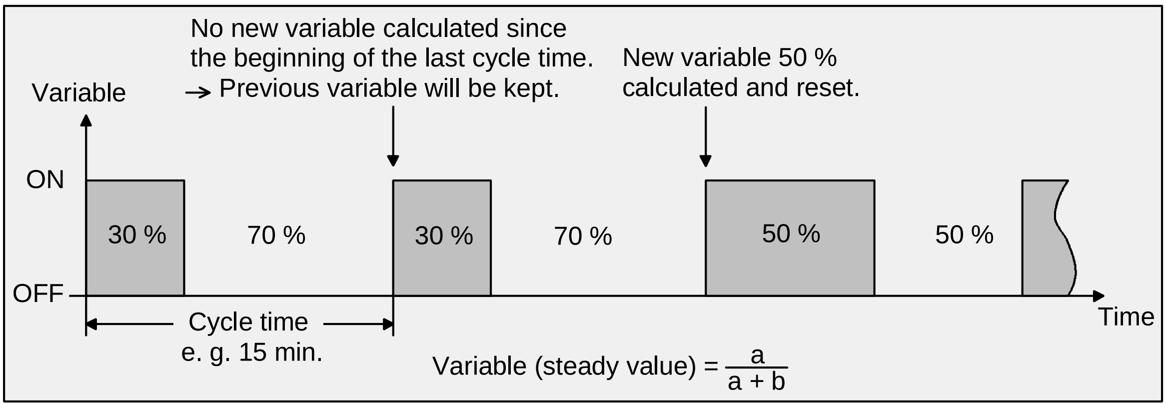
If the setpoint/actual value deviation of the room temperature results in an actuating variable of 100%, the room temperature controller uses the maximum actuating variable until the determined room temperature reaches the setpoint. This special control behaviour is known as 'clipping'.
This will quickly heat up or cool down the temperature in chilled or overheated rooms. In two stage heating or cooling systems, this control behaviour also applies to the actuating variables of the additional stages.

2. Switching PI control:

This parameterization will also keep the room temperature constant via the PI control algorithm. Averaged over time, the control system will behave the same as with a continuous controller. The only difference compared to a continuous control is the actuating variable output.
The actuating variable calculated periodically every 30 seconds by the algorithm is internally converted into an equivalent pulse width modulated (PWM) actuating variable signal and output to the bus via a 1-bit switching object after the cycle time.
Allowing for the cycle time which is adjustable via the *"Cycle time of the switching actuating variable..."* parameter on the *"Room temperature controller function – actuating variable and status output"* parameter page, the average value of the actuating variable signal resulting from this modulation is a measure for the averaged valve position of the control valve thus making it a reference value for the adjusted room temperature.
Shifting of the average value and thus changing of the heating output is achieved by changing the pulse/pause ratio of the actuating variable signal.

Depending on the calculated actuating variable, the controller adapts the pulse/pause ratio only at the end of a time period! Each change in the actuating variable is realized no matter what the relative change of the actuating variable is (the *"Automatic transmission if value changes by..."* and *"Cycle time for automatic transmission..."* parameters have no function here). The actuating variable last calculated within an active time period will be realized. If the setpoint temperature changes, for example, due to switching-over into another operating mode, the actuating variable will also be adapted only at the end of an active cycle time.

The following figure shows the output actuating variable switching signal depending on the actuating variable internally calculated (first 30 %, then 50 % actuating variable; actuating variable output not inverted).



With an actuating variable of 0 % (continuously switched-off) or 100 % (continuously switched-on) an actuating variable telegram according to the actuating variable value ("0" or "1") is output after a cycle time has elapsed. With this type of control, the 'clipping' behaviour (cf. "continuous PI control") is active as well.

Even in case of a switching PI control, the controller always calculates internally with continuous actuating variable values. These continuous values can be additionally output to the bus via a separate 1-byte value object, for example, to display it as status information.
With a switching PI control (PWM), the value object 46 is created for heating and the value object 48 for cooling. If additional stages are used, the value object 47 will be additionally enabled for the additional heating and the value object 49 for the additional cooling.
If the actuating variable for heating and cooling is to be output via a shared object, the continuous value for the activated control option will be transmitted via the object 46 and, if applicable, for the additional stages via the object 47.
The status value objects are only updated after the elapse of the parameterized cycle time together with the actuating output. The "*Automatic transmission if value changes by...*" and *"cycle time for automatic transmission..."* parameters have no function here.

An additional PI controlled heating or cooling stage works exactly the same as the switching PI control of the basic stage. The difference is that the setpoint will shift by taking into account the parameterized stage offset. All PWM controls use the same cycle time.

Cycle time:

In most cases, the pulse width modulated actuating variables are used to control electro-thermal drives. The room temperature controller transmits the switching actuating variable telegrams to a switching actuator (preferably with semi-conductor switching elements) which is connected to the drives.

Setting the cycle time for the PWM signal allows to adapt the control to the drives that are being used. The cycle time determines the switching frequency of the pulse modulated signal and allows the adaptation to the adjusting cycle times of the servo drive (traversing time the drive requires to adjust the valve from a completely closed position to a completely opened position). In addition to the adjusting cycle time the dead time (time during which the servo drives show no response when switching on or off) has to be considered. If different drives with different adjusting cycle times are used, the longer time is to be considered. The manufacturer's specifications for the drive have to be observed.

Two cases for the adjustment of the cycle time can be considered:

I. Cycle time 2 x adjusting cycle time of the drives being used, for example 15 minutes (default)

In this case, the pulse/pause times of the PWM signal are long enough for the drives to completely open or close within a time period.

Advantages:
The desired average value for the actuating variable and thus the required room temperature is relatively accurately adjusted even with several drives that are simultaneously activated.

Disadvantages:
It has to be considered that - due to the constantly 'travelled' full valve lift - the product life of the drives may decrease. The heat transfer to the room in the vicinity of the heater may be non-uniform and may be felt bothering to some people if the cycle times are very long (> 15 minutes) and if the system has a slower response (for example in case of smaller hot water heaters).

Notes:

- This setting for the cycle time is recommended for slower, more inert heating systems (for example underfloor heating).
- This setting is also recommended for a larger number of possibly different drives making it easier to average the valve travel.

II. Cycle time > adjusting cycle time of the drives being used, for example 2 minutes (default)

In this case, the short pulse/pause times of the PWM signal are not sufficient for the drives to completely open or close within a time period.

Advantages:
This setting ensures a constant water flow through the heaters with a uniform heat transfer to the room.
If only one servo drive is controlled, the controller is able to compensate for the shift of the average value caused by the short cycle time by continuously adapting the actuating variable and is thus able to adjust the desired room temperature.

Disadvantages:
If more than one drive is controlled at the same time, the desired average for the actuating variable and thus the required room temperature is adjusted only very poorly or with larger deviations.

Important:

- This setting for the cycle time is recommended for fast-responding heating systems (for example hot water heaters with higher flow temperature).

3. Switching 2-state control:

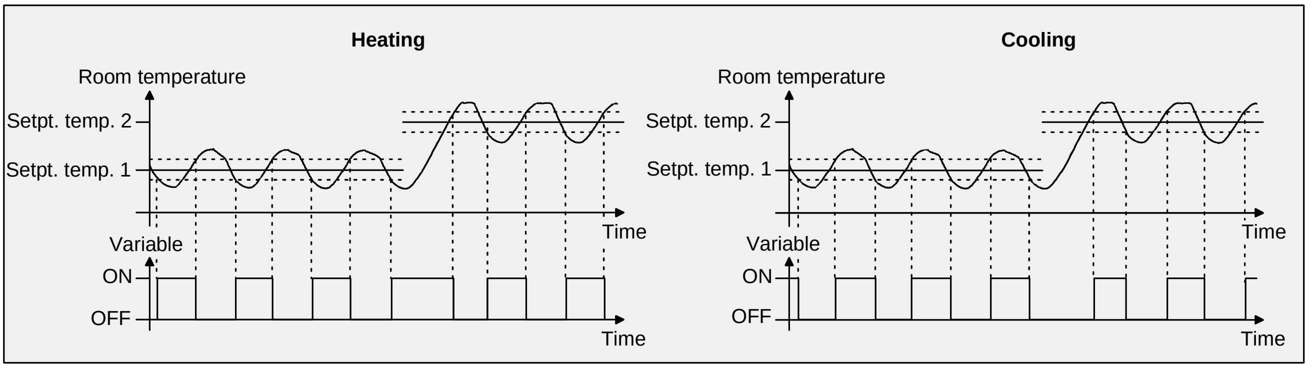
The 2-state control represents a very simple temperature control. For this type of control, two hysteresis temperature values are predefined. The controller addresses the actuating elements via switch-on and switch-off actuating variable commands (1-bit). A continuous actuating variable will not be calculated with this type of control. The room temperature is evaluated periodically every 30 seconds as well, i.e. the actuating variables, if required, will change only during these times.
While the 2-state temperature control is very simple, the fluctuating temperature is a disadvantage. For this reason, no fast responding heating or cooling systems should be controlled via a 2-state control as it may result in very strong temperature overshoot and thus in a loss of comfort.

When defining the hysteresis limit settings, one has to distinguish between the control options:

• Individual control options *"heating"* or *"cooling"*:

While in heating mode the controller will switch-on the heater , if the temperature falls below a preset limit. In the heating mode the controller will switch-off the heater only, if an adjusted temperature limit has been exceeded.
In the cooling mode the controller will switch-on the cooling, if the room temperature has exceeded a preset limit. While in the cooling mode, the controller will switch-off the cooling only if the temperature has fallen below an adjusted temperature limit.
Depending on the switching state of the actuating variable, a "1" or "0" will be output, if the value exceeds or remains under the hysteresis limits.
In case of a 2-step control, the LEDs "heating" or "cooling" will light up or the signalling objects for heating and cooling will already become active as soon as the temperature falls short of the temperature setpoint in case of heating or exceeds the temperature setpoint in case of cooling. In this case, the hysteresis is not being considered.
The upper and lower hysteresis limits are to be parameterized in the ETS for both control options.

The following figure shows a 2-state control for the individual control options "*heating*" or "*cooling*" (heating on the left, cooling on the right; two temperature setpoints; one-stage heating or cooling; non-inverted actuating variable output):

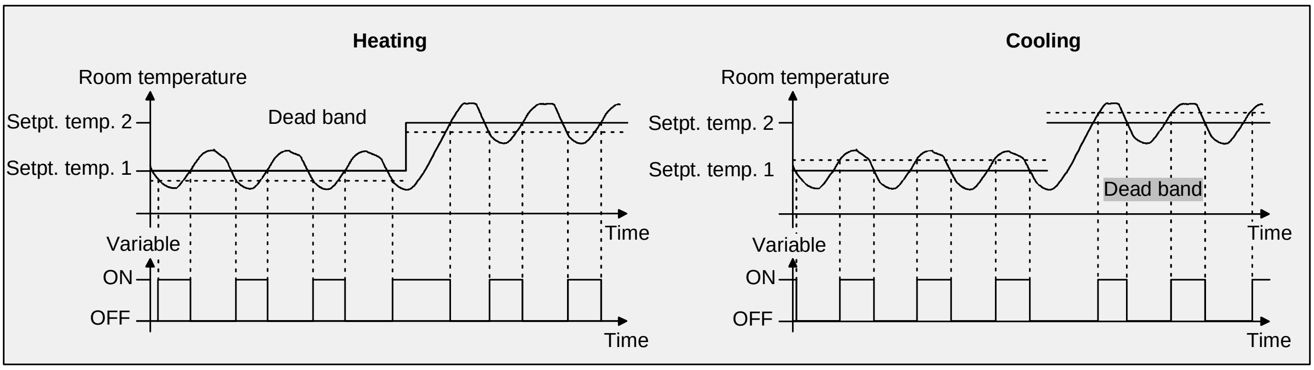


An additional 2-state control heating or cooling stage works in exactly the same way as the 2-state control of the basic stage. The difference is that the setpoint and the hysteresis values will shift by taking into account the parameterized stage offset.

• *"Heating"* and *"cooling"* mixed mode:

While in heating mode the controller will switch-on the heater , if the temperature falls below a preset limit. The control will switch-off the heater as soon as the room temperature exceeds the temperature setpoint of the active operating mode.
In the cooling mode the controller will switch-on the cooling, if the room temperature has exceeded a preset limit. While in the cooling mode the controller will switch-off the cooling as soon as the room temperature falls below the temperature setpoint of the active operating mode.
Thus, in mixed-mode operation there is no upper hysteresis limit value for heating or no lower hysteresis limit value for cooling as these values would lie in the dead zone. There is neither heating nor cooling within the dead zone.
Depending on the switching state the actuating variable "1" or "0" will be output, if the values exceed or remain under the hysteresis limits.
In case of a 2-state control, the LEDs "heating" or "cooling" will light up or the signalling objects for heating and cooling will already become active as soon as the temperature falls short of the temperature setpoint in case of heating or exceeds the temperature setpoint in case of cooling. In this case, the hysteresis is not being considered.
The upper and lower hysteresis limits are to be parameterized in the ETS for both control options.

The following figure shows a 2-state control for the "heating" or "cooling" mixed-mode (activated heating on the left, activated cooling on the right; two temperature setpoints; non-inverted actuating variable output):



An additional 2-state control heating or cooling stage works in exactly the same way as the 2-state control of the basic stage. The difference is that the setpoint and the hysteresis values will shift by taking into account the parameterized stage offset.

2.3.2 Adapting the control algorithms

2.3.2.1 Adapting the PI control

There are several systems available which may heat or cool a room. Thus, it is possible to uniformly heat or cool the surroundings with heat transfer media (preferably water or oil) in combination with room air convection. Such systems are used, for example, with wall mounted heaters, underfloor heating or cooling ceilings.
Alternatively, rooms can be heated or cooled by blower units. In most cases, such systems are electrical fan heaters, fan coolers or refrigerating compressors with fans. Due to the direct heating of the room air, such heating and cooling systems work quite fast.

The control parameters need to be adjusted so that the PI control algorithm may efficiently control all common heating and cooling systems thus making the room temperature control work as fast as possible and without deviation.
Certain factors can be adjusted in a PI control system that can influence the control behaviour quite significantly at times. For this reason, the room temperature controller can be set to predefined 'experience values' for the most common heating and cooling systems. In case the selection of a corresponding heating or cooling system does not yield a satisfactory result with the default values, the adaptation can optionally be optimized via control parameters.

Predefined control parameters for the heating or cooling stage and, if applicable, also for the additional stages are adjusted via the *"Type of heating*" or *"Type of cooling"* parameters These fixed values correspond to the practical values of a properly planned and executed air conditioning system and will result in an ideal behaviour of the temperature control. The following types can be set for heating or cooling.

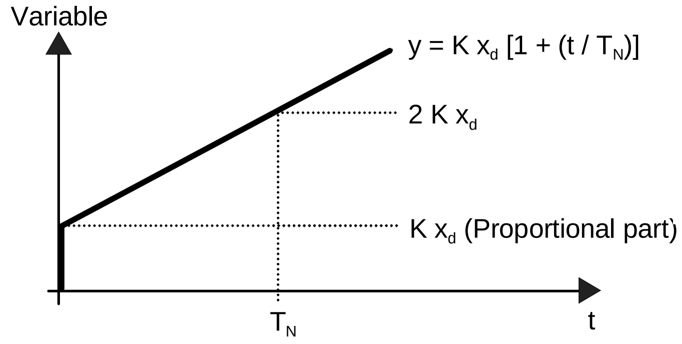
for heating control **Type of heating**	**default values** **Proportional range**	 **Reset-time**	**recommended type of PI control:**	**recommended PWM cycle time**
• Hot-water heating	5 Kelvin	150 minutes	continuous / PWM	15 minutes **
• Underfloor heating	5 Kelvin	240 minutes	PWM	15 – 20 min.
• Electrical heating	4 Kelvin	100 minutes	PWM	10 – 15 min.
• Blower convector	4 Kelvin	90 minutes	continuous	---
• Split-unit *	4 Kelvin	90 minutes	PWM	10 – 15 min.
for cooling control **Type of cooling**	**default values** **Proportional range**	 **Reset-time**	**recommended type of PI control:**	**recommended PWM cycle time**
• Cooling ceiling	5 Kelvin	240 minutes	PWM	15 – 20 min.
• Blower convector	4 Kelvin	90 minutes	continuous	---
• Split-unit *	4 Kelvin	90 minutes	PWM	10 – 15 min.

*: split mobile climate control unit,
**: for smaller, fast-working heaters (e.g. higher flow temperature) PWM cycle time 2 – 3 minutes.

If the *"Type of heating"* or *"Type of cooling"* parameters are set to *"via control parameters"* it will be possible to adjust the control parameters manually. The control may be considerably influenced by presetting the proportional range for heating or for cooling ('P' part) and the reset-time for heating or for cooling ('I' part).

Notes:

- Even minor adjustments of the control parameters will lead to a noticeably different control behaviour.
- The adaptation should start with the control parameter setting for the corresponding heating or cooling system according to the fixed values mentioned above.



x_d : Regeldifferenz $x_d = x_{soll} - x_{ist}$
P = 1/K : parametrierbarer Proportionalbereich
K = 1/P : Verstärkungsfaktor
T_N : parametrierbare Nachstellzeit

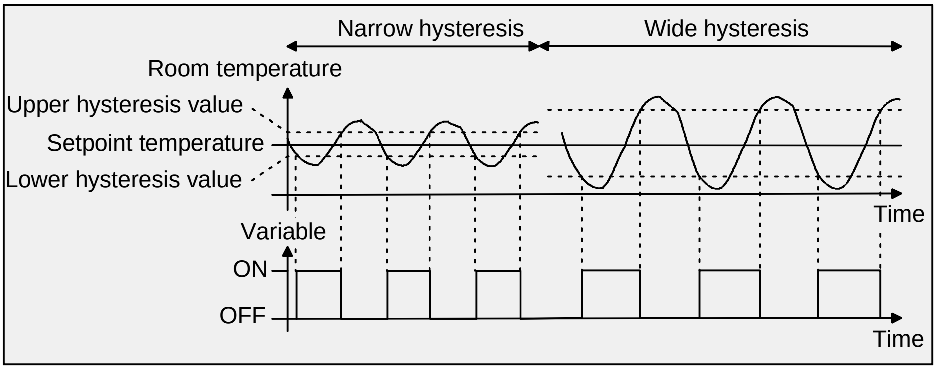
PI control algorithm: Actuating variable $y = K x_d [1 + (t / T_N)]$; By deactivating the reset-time (setting = "0"):
P control algorithm: Actuating variable $y = K x_d$

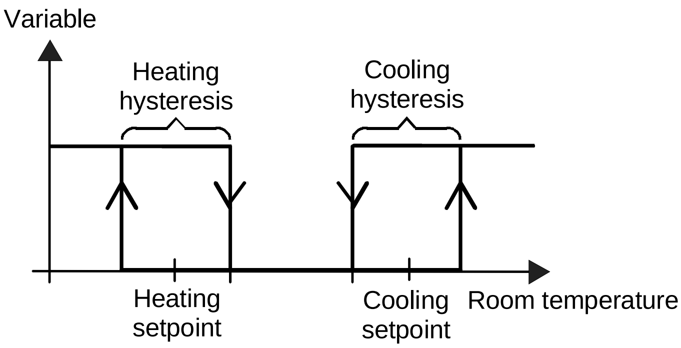
Parameter setting		**Effect**
P	small proportional range	large overshoot in case of setpoint changes (possibly permanently), quick adjustment to the setpoint
P	large proportional range	no (or small) overshoot but slow adjustment
T_N	short reset-time	fast compensation of control deviations (ambient conditions), risk of permanent oscillations
T_N	long reset-time	slow compensation of control deviations

2.3.2.2 Adapting the 2-state control

The 2-state control represents a very simple temperature control. For this type of control, two hysteresis temperature values are predefined.
The upper and lower temperature hysteresis limits can be adjusted via parameters. It has to be considered that…:

- a small Hysteresis will lead to small temperature variations but to a higher bus load.
- a large hysteresis switches less frequently but will cause uncomfortable temperature variations.





2.3.3 Actuating variable output

2.3.3.1 Actuating variable objects

The format of the actuating variable objects is determined depending on the control algorithm selected for heating and / or cooling and, if applicable, also for the additional stages. 1-bit or 1-byte actuating variable objects can be created. The control algorithm calculates and outputs the actuating variables in intervals of 30 seconds. With the pulse-width modulated PI control (PWM), the actuating variable is updated, if required, only at the end of a time cycle.

Possible object data formats for the actuating variables separately for both control options, for the basic and the additional stage, are…

- the continuous PI control: 1 byte,
- the switching PI control: 1 bit + additionally 1 byte (e.g. for status indication in visualization systems),
- the switching 2-state control: 1 bit.

Depending on the selected control option, the controller is able to address heating and / or cooling systems, to determine actuating variables and to output them via separate objects. One distinguishes between two cases for the mixed mode "heating and cooling:

Case 1: Heating and cooling system are two separate systems.
In this case, the *"Transmit actuating variable heating and cooling to a shared object"* parameter should be set to *"no"* (default) on the *"Room temperature controller functions"* parameter page. Thus there are separate objects available for each actuating variable which can be separately addressed via the individual systems. This setting allows to define separate types of control for heating and cooling.

Case 2: Heating and cooling system are a combined system.
In this case the *"Transmit actuating variable heating and cooling to a shared object"* parameter may be set, if required, to *"yes"* on the *"room temperature controller functions"* parameter page. This will transmit the actuating variables for heating and cooling to the same object. In case of a two-stage control, another shared object will be enabled for the additional stages for heating and cooling.
With this setting, it is only possible to define the same type of control for heating and for cooling as the control and the data format must be identical. The (*"Type of heating / cooling*") control parameter for cooling and heating still have to be defined separately.
A combined actuating variable object may be required, for example, if heating as well as cooling shall take place via a single-duct system (combined heating and cooling system). For this purpose, the temperature of the medium in the single-duct system must be changed beforehand by the system control. Afterwards the control option is set via the object (often the single-duct system uses cold water for cooling during the summer, hot water for heating during the winter).

Important:

Basically, it is not possible to heat and cool at the same time (actuating variables > "0")!

If required, the actuating variable can be inverted before the transmission. The actuating variable value will be output in inverted form according to the object data format via the *"Output of actuating variable heating"* or *"Output of actuating variable cooling"* parameters or via a combined *"Output of actuating variable"* object . The parameter for inverting the additional stage(s) are additionally available in the two-stage controlled operation.
The following applies…

for continuous actuating variables:	not inverted:	Actuating variable 0 % ... 100 %,	Value 0 … 255,
	inverted:	Actuating variable 0 % ... 100 %,	Value 255 … 0,
for switching actuating variables:	not inverted:	Actuating variable on / off,	Value 0 / 1,
	inverted:	Actuating variable on / off,	Value 1 / 0.

2.3.3.2 Automatic transmission

- Continuous PI control:
  In case of a continuous PI control the room temperature controller calculates a new actuating variable periodically every 30 seconds and outputs them to the bus via a 1-byte value object. The change interval of the actuating variable can be determined in percent according to which a new actuating variable is to be output on the bus via the *"Automatic transmission if value changes by..."* parameter on the *"Room temperature controller function – actuating variables and status output"* parameter page . The change interval can be parameterized to "0" so that a change in the actuating variable will not result in an automatic transmission.

  In addition to the actuating variable output following a change, the current actuating variable value can be periodically transmitted to the bus. In addition to the times when changes are to be expected, other actuating variable telegrams will be output according to the active value after a parameterizable cycle time.
  This ensures that telegrams can be received within the monitoring interval during periodic safety monitoring of the actuating variable in the servo drive or in the addressed switching actuator. The time interval predetermined by the *"Cycle time for automatic transmission..."* parameter should correspond to the monitoring interval in the actuator (cycle time in the controller is preferably to be parameterized smaller).
  The "0" setting will deactivate the periodic transmission of the actuating variable.

  With the continuous PI control it should be noted that - if periodic and automatic transmission are both deactivated - no more actuating variable telegrams will be transmitted in case of a change!

- Switching PI control PWM):
  In case of a switching PI control (PWM), the room temperature controller calculates a new actuating variable internally every 30 seconds. With this type of control, however, updating of the actuating variable takes place, if required, only at the end of a cycle. The *"Automatic transmission if value changes by..."* and *"Cycle time for automatic transmission..."* parameters are not enabled with this control algorithm.

- 2-state control:
  In case of a 2-state control, the room temperature and thus the hysteresis values are evaluated periodically every 30 seconds, so that the actuating variables, if required, will change only during these times. The *"Automatic transmission if value changes by..."* parameter is not enabled as this control algorithm does not calculate continuous actuating variables.

  In addition to the actuating variable output following a change, the current actuating variable value can be periodically transmitted to the bus. In addition to the times when changes are to be expected, other actuating variable telegrams will be output according to the active value after a parameterizable cycle time.
  This ensures that telegrams can be received within the monitoring interval during periodic safety monitoring of the actuating variable in the servo drive or in the addressed switching actuator. The time interval predetermined by the *"Cycle time for automatic transmission..."* parameter should correspond to the monitoring interval in the actuator (cycle time in the controller is preferably to be parameterized smaller).
  The "0" setting will deactivate the periodic transmission of the actuating variable.

2.4 Temperature setpoints

2.4.1 Setpoint presettings in the ETS

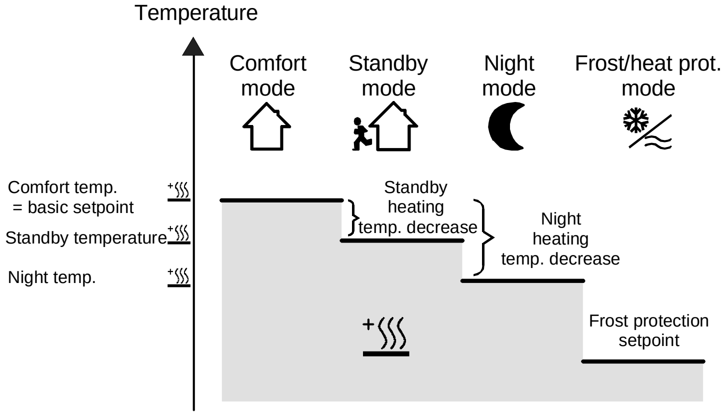
Temperature setpoints can be preset independently for each operating mode. The setpoints for the "comfort ", "standby " und "night " modes can be parameterized in the ETS. If desired, the setpoint temperatures can be subsequently adjusted via local operation of the controller in the programming mode or via object control. The "frost/heat protection " operating mode allows the separate parameterization of two temperature setpoints for heating (frost protection) and cooling (heat protection) only in the ETS.

When presetting the setpoint temperatures for comfort, standby and night mode, attention has to be paid to the fact that all setpoints depend on each other as all values are derived from the basic temperature (basic setpoint). The *"Basic temperature after reset"* parameter on the *"Setpoints"* parameter page determines the basic setpoint which is loaded when the device is programmed with the ETS.
Taking into account the *"Decreasing /increasing the setpoint temperature in standby mode"* or *"Decreasing/ increasing the setpoint temperature in night mode"* parameters, the temperature setpoints for the standby and night mode are derived from this value depending on the heating or cooling control option. The dead zone will be additionally considered for the *"heating and cooling"* mode.

In two stage control mode, all setpoint temperatures of the additional stage are derived from the setpoint temperatures of the basic stage. The setpoint temperatures of the additional stage are determined by subtracting the *"stage offset"* , which is parameterized in the ETS from the setpoints of the basic stage in heating mode or by adding the setpoints in cooling mode. If the temperature setpoints of the basic stage are changed by setting a new basic setpoint, the setpoint temperatures of the additional stage will be indirectly and automatically changed as well. Both stages will heat or cool with the same actuating variable at the same time when the setpoint difference is "0".

Depending on the control option, the relationships described on the following pages have to be considered for the setpoint temperatures.

2.4.1.1 Setpoints for the "heating" control option



(Recommended setpoint temperatures)

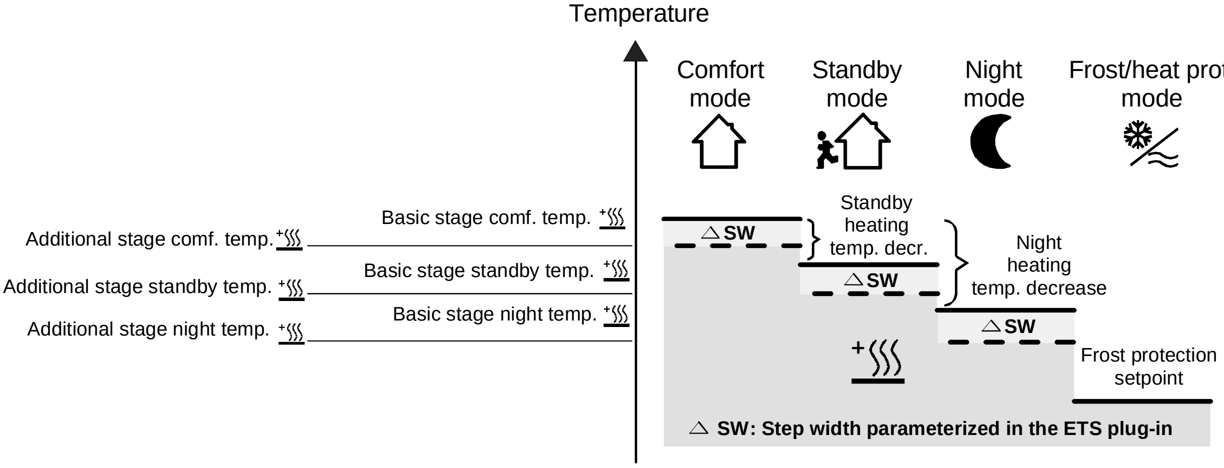
The setpoint temperatures for comfort, standby and night mode exist for this control option. The frost protection temperature can be preset. The following applies:

$$T_{\text{standby setpoint heating}} \leq T_{\text{comfort setpoint heating}} \text{ or } T_{\text{night setpoint heating}} \leq T_{\text{comfort setpoint heating}}$$

The standby and night setpoint temperatures are derived from the comfort setpoint temperature (basic setpoint) in line with the parameterized decrease-temperatures. If enabled, it is also possible to adjust also other reduction temperatures by means of a local operation on the controller itself in the programming mode by changing the setpoint temperature values for night and standby mode.

The frost protection is supposed to prevent the heating system from freezing. For this reason the frost protection temperature should be set to a smaller value than the night temperature for heating (default: +7 °C). In principle, however, it is possible to select frost protection temperature values between +7 °C and +40 °C.
The possible range of values for a setpoint temperature lies between + 7.0 °C and + 99.9 °C for "heating" and is limited by the frost protection temperature in the lower range.

The stage offset parameterized in the ETS will be additionally considered in a two stage heating mode.



$$T_{\text{comfort setpoint additional stage heating}} \leq T_{\text{comfort setpoint basic stage heating}} \,/\, T_{\text{standby setpoint additional stage heating}} \leq T_{\text{standby setpoint basic stage heating}}$$

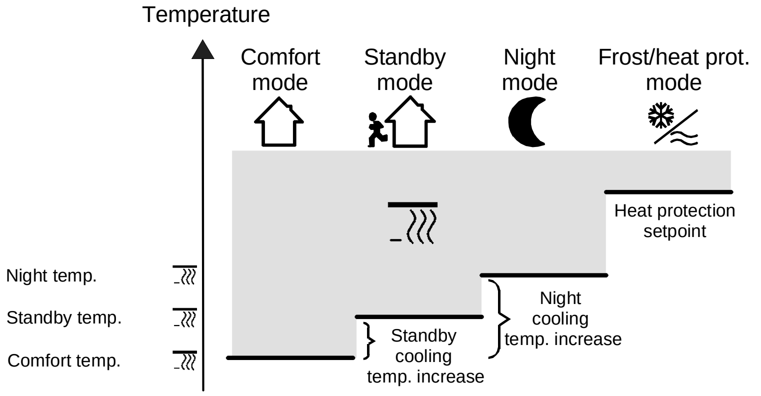
$$T_{\text{standby setpoint heating}} \leq T_{\text{comfort setpoint heating}}$$

or

$$T_{\text{comfort setpoint additional stage heating}} \leq T_{\text{comfort setpoint basic stage heating}} \,/\, T_{\text{night setpoint additional stage heating}} \leq T_{\text{night setpoint basic stage heating}}$$

$$T_{\text{night setpoint value heating}} \leq T_{\text{comfort setpoint heating}}$$

2.4.1.2 Setpoints for the "cooling" control option



(Recommended setpoint temperatures)

The setpoint temperatures for comfort, standby and night mode exist for this control option. The heat protection temperature can be preset. The following applies:

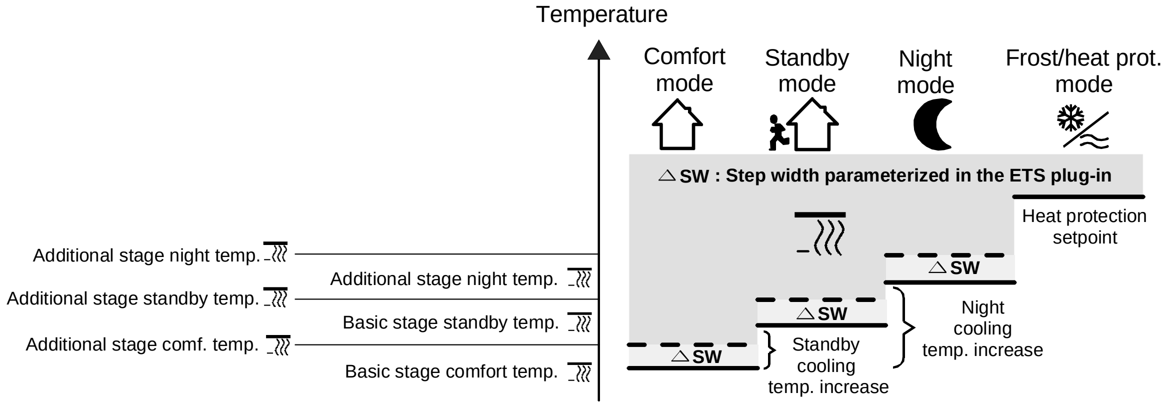
$$T_{\text{comfort setpoint value cooling}} \leq T_{\text{standby setpoint cooling}} \text{ or } T_{\text{comfort setpoint cooling}} \leq T_{\text{night setpoint cooling}}$$

The standby and night setpoint temperatures are derived from the comfort setpoint temperature (basic setpoint) in line with the parameterized increase-temperatures.

The heat protection is supposed to prevent the temperature from exceeding the maximum permissible room temperature in order to protect system components.

For this reason, the heat protection temperature should be set to a larger value than the night temperature (default: +35 °C). In principle, however, it is possible to select heat protection temperature values between +7 °C and +45 °C.
The possible range of values for a set-temperature lies between - 99.9 °C and + 45.0 °C for "cooling" and is limited by the heat protection temperature in the upper range.

The stage offset parameterized in the ETS will be additionally considered in a two stage heating mode.



$$T_{\text{comfort setpoint basic stage cooling}} \leq T_{\text{comfort setpoint additional stage cooling}} / T_{\text{standby setpoint basic stage cooling}} \leq T_{\text{standby setpoint additional stage cooling}}$$

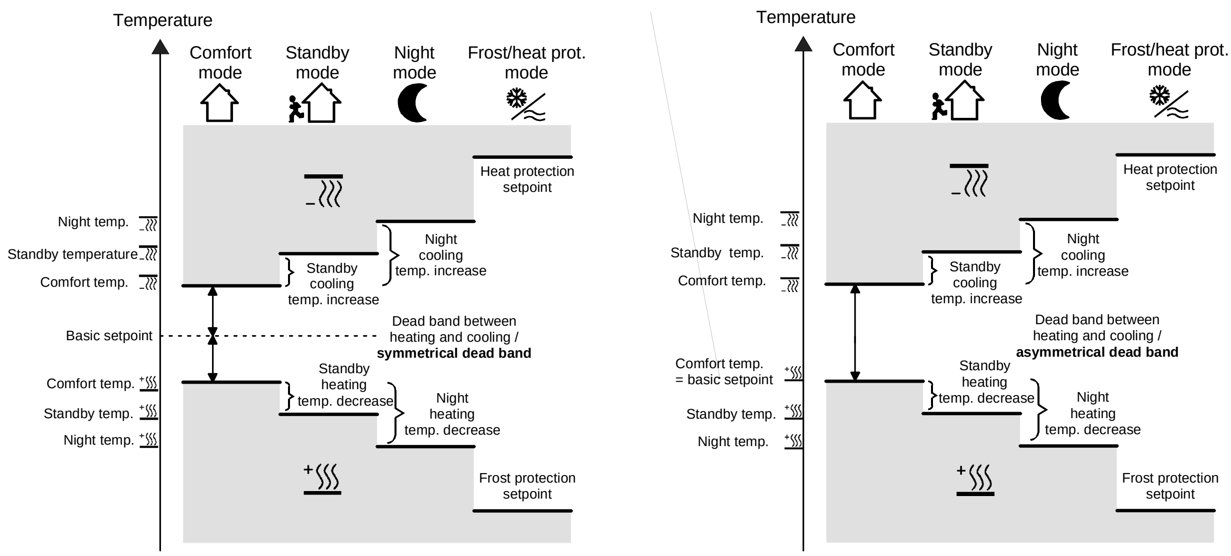
$$T_{\text{comfort setpoint cooling}} \leq T_{\text{standby setpoint cooling}}$$

or

$$T_{\text{comfort setpoint basic stage cooling}} \leq T_{\text{comfort setpoint additional stage cooling}} / T_{\text{night setpoint basic stage cooling}} \leq T_{\text{night setpoint additional stage cooling}}$$

$$T_{\text{comfort setpoint cooling}} \leq T_{\text{night setpoint cooling}}$$

2.4.1.3 Setpoint for the "heating and cooling" control option:



(Recommended setpoint temperatures)

For this control option, the setpoint temperatures of both control options exist for comfort, standby and night mode as well as the dead zone. In addition, the frost protection and the heat protection temperatures can be preset. The following applies:

$$T_{\text{standby setpoint heating}} \leq T_{\text{comfort setpoint heating}} \leq T_{\text{comfort setpoint cooling}} \leq T_{\text{standby setpoint cooling}}$$

or

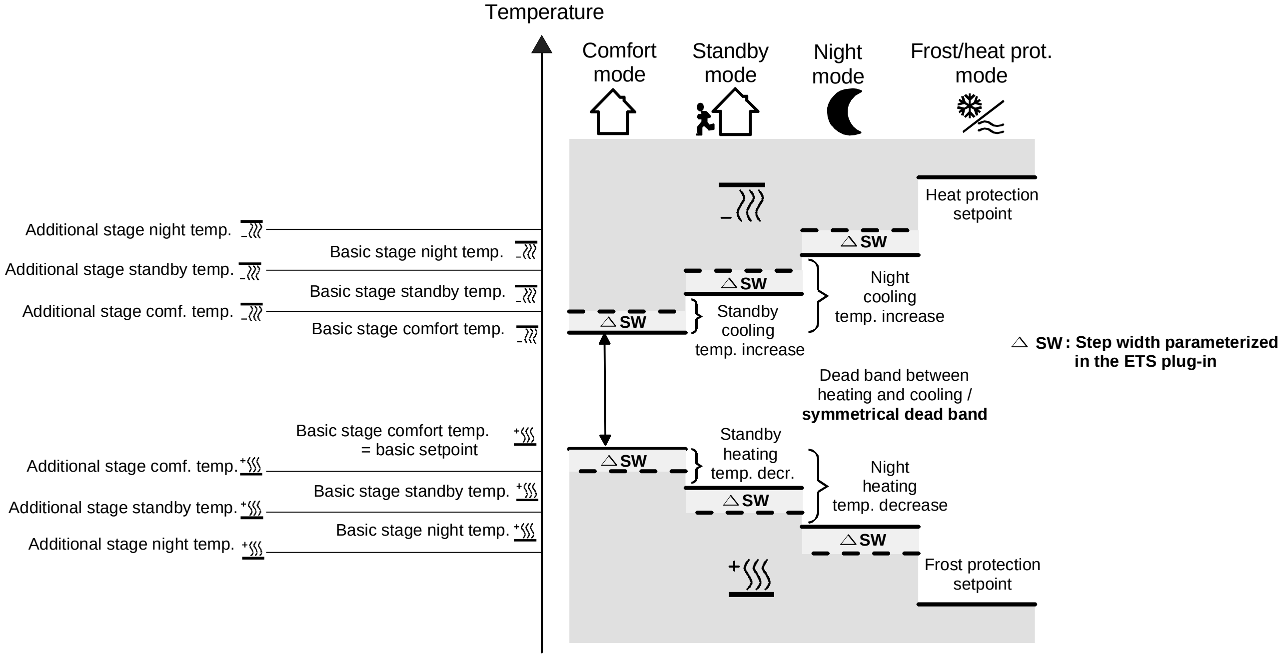
$$T_{\text{night setpoint heating}} \leq T_{\text{comfort setpoint heating}} \leq T_{\text{comfort setpoint cooling}} \leq T_{\text{night setpoint cooling}}$$

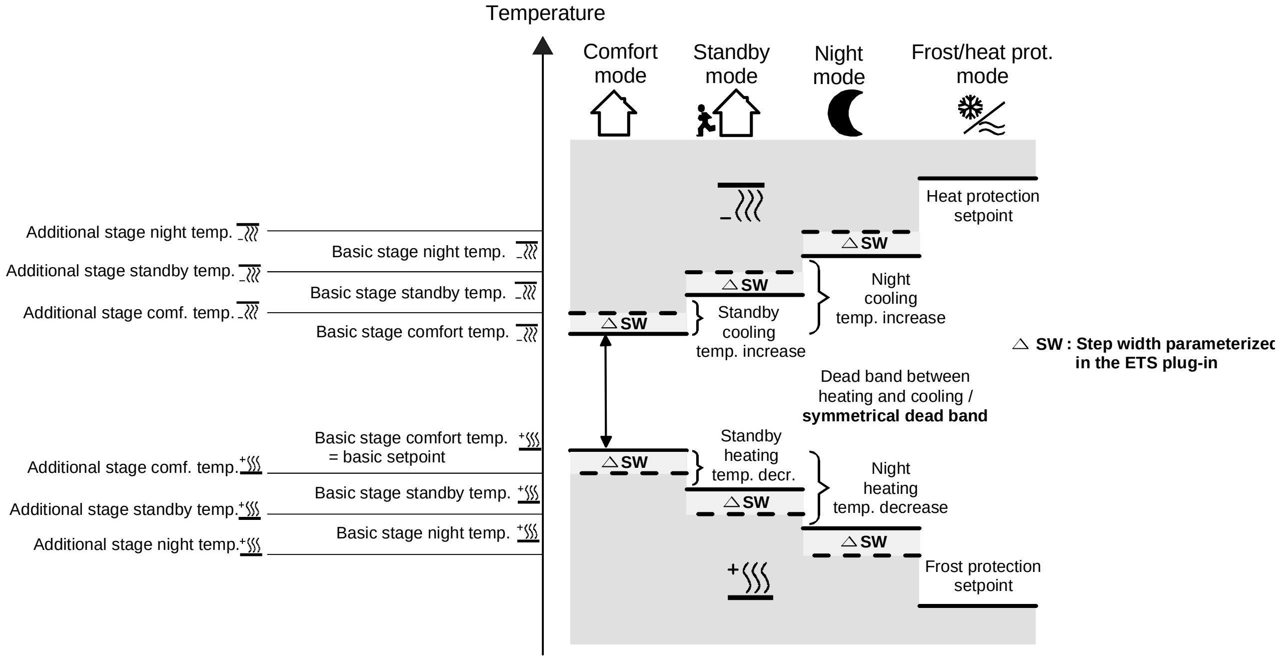
The setpoint temperatures for "Standby" and "Night" are derived from the comfort setpoint temperatures for heating or cooling. The temperature increase (for cooling) and the temperature decrease (for heating) of both operating modes can be preset in the ETS. The comfort temperatures themselves are derived from the dead zone and the basic setpoint.

The frost protection is supposed to prevent the heating system from freezing. For this reason, the frost protection temperature should be to a set smaller value than the night temperature for heating (default: +7 °C). In principle, however, it is possible to select frost protection temperature values between +7 °C and +40 °C.
The heat protection is supposed to prevent the temperature from exceeding the maximum permissible room temperature in order to protect system components. For this reason the heat protection temperature should be set to a larger value than the night temperature for cooling (default: +35 °C). In principle, however, it is possible to select heat protection temperature values between +7 °C and +45 °C.

The possible range of values for a setpoint temperature ("heating and cooling") lies between + 7 °C and + 45.0 °C and is limited by the frost protection temperature in the lower range and by the heat protection temperature in the upper range..

The stage offset parameterized in the ETS will be additionally considered in a two stage heating or cooling mode.





$$T_{\text{comfort setpoint additional heating}} \leq T_{\text{comfort setpoint basic heating}} \leq T_{\text{comfort setpoint basic cooling}} \leq T_{\text{comfort setpoint additional cooling}}$$
$$T_{\text{standby setpoint additional heating}} \leq T_{\text{standby setpoint basic heating}} \leq T_{\text{standby setpoint basic cooling}} \leq T_{\text{standby setpoint additional cooling}}$$
$$T_{\text{standby setpoint heating}} \leq T_{\text{comfort setpoint heating}} \leq T_{\text{comfort setpoint cooling}} \leq T_{\text{standby setpoint cooling}}$$

or

$$T_{\text{comfort setpoint additional heating}} \leq T_{\text{comfort setpoint basic heating}} \leq T_{\text{comfort setpoint basic cooling}} \leq T_{\text{comfort setpoint additional cooling}}$$
$$T_{\text{night setpoint additional heating}} \leq T_{\text{night setpoint basic heating}} \leq T_{\text{night setpoint basic cooling}} \leq T_{\text{night setpoint additional cooling}}$$
$$T_{\text{night setpoint heating}} \leq T_{\text{comfort setpoint heating}} \leq T_{\text{comfort setpoint cooling}} \leq T_{\text{night setpoint cooling}}$$

Dead zone:

The comfort setpoint temperatures for heating and cooling are derived from the basic setpoint in consideration of the adjusted dead zone. The dead zone (temperature zone for which there is neither heating nor cooling) is the difference between the comfort setpoint temperatures.
The *"Dead zone between heating and cooling"*, *"Dead zone position"* parameters as well as the *"Basic temperature after reset"* parameter are preset in the ETS. The following settings must be distinguished:

Dead zone = *"symmetrical"* (default):
The dead zone preset in the ETS is divided into two parts at the basic setpoint. The comfort setpoint temperatures are derived directly from the basic setpoint resulting from the half dead zone. The following applies:

$T_{\text{basic setpoint}} - ½T_{\text{dead zone}} = T_{\text{comfort setpoint heating}}$ or $T_{\text{basic setpoint}} + ½T_{\text{dead zone}} = T_{\text{comfort setpoint cooling}}$

$T_{\text{comfort setpoint cooling}} - T_{\text{comfort setpoint heating}} = T_{\text{dead zone}}$; $T_{\text{comfort setpoint cooling}} \geq T_{\text{comfort setpoint heating}}$

Dead zone position = *"asymmetrical"*:
With this setting, the comfort setpoint temperature for heating equals the basic setpoint!
The dead zone preset in the ETS takes only effect from the basic setpoint in the direction of comfort temperature for cooling. Thus the comfort setpoint temperature for cooling is derived directly from the comfort setpoint for heating. The following applies:

$T_{\text{basic setpoint}} = T_{\text{comfort setpoint heating}}$ ➔ $T_{\text{basic setpoint}} + T_{\text{dead zone}} = T_{\text{comfort setpoint cooling}}$

➔ $T_{\text{comfort setpoint cooling}} - T_{\text{comfort setpoint heating}} = T_{\text{dead zone}}$; $T_{\text{comfort setpoint cooling}} \geq T_{\text{comfort setpoint heating}}$

2.4.2 Adjusting the setpoints

2.4.2.1 Adjusting basic temperature and setpoint temperatures for comfort, standby and night mode

When presetting the setpoint temperatures for comfort, standby and night mode, attention has to be paid to the fact that all setpoints depend on each other as all values are derived from the basic temperature (basic setpoint). The *"Basic temperature after reset"* parameter on the *"Setpoints"* parameter page determines the basic setpoint which is loaded when the device is programmed with the ETS.

It is possible to change or adjust the setpoint temperatures 'later' via the *"Basic setpoint"* object).

Any change must always be enabled in the ETS on the *"Setpoints"* parameter page. It is possible to permit the *"Change of basic temperature setpoint*" by directly changing the comfort temperature for heating on the device and/or by presetting a new basic setpoint via the bus.

In case the basic setpoint adjustment via the bus is disabled, object 26 will be hidden.

Adjusting the basic setpoint / comfort temperature for heating:

It is only in the event of a basic setpoint change that two cases must be distinguished:

- Case 1: The basic setpoint change is permanently adopted,
- Case 2: The basic setpoint change is adopted only temporarily (default).

Via the *"Adopt basic temperature setpoint change permanently"* parameter on the *"Room temperature controller function /setpoints"* parameter page, it is possible to determine whether the changed basic temperature value shall be stored in memory permanently ("yes") or only temporarily ("no").

Case 1:

If the basic temperature setpoint is changed, it will be permanently stored in the room temperature controller's EEPROM. The newly adjusted value will overwrite the basic setpoint temperature originally parameterized in the ETS! This is the only way to keep the changed basic setpoint even after switching-over the operating mode or after a reset.

Notes:

- Frequent adjustments of the basic temperature (e.g. several times a day) can affect the product life of the device as the non-volatile storage (EEPROM) is designed only for less frequent write access.
- Any value that is preset via local operation will not be adopted by object 26.
- The stored basic setpoint will still be active after the return of bus voltage. The value of the object 26 or 27 is then however "0". The current basic setpoint can be read out only after an external object update (set "R" flag!).

Case 2:

The basic setpoint adjusted on the room temperature controller itself or received via the object stays only temporarily active in the current operating mode. In case of a bus voltage failure or following a switch-over into another operating mode (e.g. comfort followed by standby), the basic setpoint adjusted via local operation or received via the object will be discarded and replaced by the value which was originally parameterized in the ETS.

Adjusting the setpoints for standby mode, night mode and dead zone (comfort temperature for cooling):

Since the setpoint temperatures for the "standby" and "night" operating modes or the setpoints for the "cooling" control option are derived - in consideration of the increase, decrease or dead zone values that are parameterized in the ETS or preset locally - from the basic setpoint temperature, these setpoint temperatures will shift linearly by the change of the basic setpoint value.

In addition, it is possible to adjust setpoint temperatures other than those parameterized in the ETS for "standby", "night" or "dead zone" only via local control in the programming mode. In this case, the originally parameterized decrease, increase temperatures or dead zone temperatures will be replaced by the new values resulting from the locally adjusted temperature setpoints. Independent of the *"Adopt basic temperature setpoint change permanently"* parameter, the temperature setpoints for the standby or night mode or "cooling" comfort mode (dead zone) will always be stored in the non-volatile EEPROM memory.

2.4.2.2 Basic setpoint shifting

In addition to the setting of individual temperature setpoints by the ETS, by local operation in the programming mode or by the basic setpoint object, the user is able to shift the basic setpoint anytime with the control wheel within the parameterized limits.

It has to be considered that a shift of the displayed setpoint temperature (temperature offset of the basic temperature) will directly affect the basic setpoint and as a result shift all other temperature setpoints.

Whether a basic setpoint shift only affects the currently active operating mode or whether it influences all other setpoint temperatures of the remaining operating modes is determined by the *"Adopt change of basic setpoint shift permanently*" parameter on the *"Setpoint"* parameter page.

Settings: "no" (default):
Shifting of the basic setpoint is effective only as long as the operating mode or control option is not changed or the basic setpoint is maintained. Otherwise the setpoint shift will be reset to "0".

Setting "yes":
In general, shifting of the basic setpoint affects all operating modes. The shift is maintained even after switching-over the operating mode or the control option or after readjusting the basic setpoint.

The adjustable temperature range for a basic setpoint shift is defined via the *"Adjusting the basic setpoint temperature upwards"* or *"Adjusting the basic setpoint temperature downwards"* parameters. It is possible to shift the current setpoint by a maximum of +/- 10 K. The stage offset is set to a non-adjustable 0.5 °C.

Notes on the basic setpoint shifting:

- Since the value for the basic setpoint shifting is stored exclusively in volatile memory (RAM), the shift will get lost in case of a reset (e.g. bus voltage failure).
- A setpoint shift does not affect the temperature setpoints for frost or heat protection.

Communication objects for the basic setpoint shifting:

The controller tracks the current setpoint shift in the communication object 52 (*"Current setpoint shifting"*) with a 1-byte counting value (acc. to KNX DPT 6.010 – representation of positive and negative values in a 2's complement). By connecting to this object the controller extensions are also able to display the current setpoint shift.

As soon as there is an adjustment by one temperature step (0.5 °C) in positive direction, the controller counts up the value by one digit. The counting value will be counted down by one digit, if there is a negative adjustment of the temperature.
Thus the possible range of values for the object is determined by the setpoint's adjustment options. A value of "0" means that no setpoint shift has been adjusted.

Example:

Initial situation:
The temperature step for the setpoint shift is set to 0.5 °K.
Current setpoint temperature = 21.0°C / counting value in object 52 = "0" (no setpoint shift active)

After shifting the setpoint:
- A setpoint shift by one temperature step in positive direction will count up the value in object 52 by one = "1". Current setpoint temperature = 21.5°C.
- Another setpoint shift by one temperature step in positive direction will count up the value in object 52 again by one = "2.. Current setpoint temperature = 22.0°C.
- A setpoint shift by one temperature step in negative direction will count down the value in object 52 by one = "1". Current setpoint temperature = 21.5°C.
- Another setpoint shift by one temperature step in negative direction will count down the value in object 52 again by one = "0". Current setpoint temperature = 21.0°C.
- Another setpoint shift by one temperature step in negative direction will count down the value in object 52 again by one = "-1". Current setpoint temperature = 20.5°C.

etc.

The maximum possible range of values for the *"Current setpoint shift"* communication object depends on the *"Adjustment options of the basic setpoint temperature upwards/downwards"* parameter. A parameterization of ± 10 K at this point will have the value of the object move within the limits –20 ... +20.

In addition, the setpoint shift of the controller can be externally adjusted via communication object 53 (*"Setpoint shift preset"*). This object has the same datapoint type and range of values as object 52 (see above). By connecting to object 53, the controller extensions are able to directly adjust the current setpoint shift of the controller.

As soon as the controller receives a value, it will adjust the setpoint shift accordingly. Each value increment corresponds to a temperature step of 0.5°C (cf. example above). Values that lie within the possible range of values can be approached directly.

The controller monitors the received value independently. As soon as the external preset value exceeds the limits of the adjustment options for the setpoint shift in positive or negative direction, the controller will correct the received value and adjust the setpoint shift to maximum shift. Depending on the direction of the shift, the value feedback is set to the maximum value via communication object 52 (*"Current setpoint shift"*).

2.4.3 Transmitting the setpoint temperature

The setpoint temperature determined by the current operating mode or subsequently readjusted can be transmitted actively to the bus by means of object 50 *"Setpoint temperature".*
The *"Transmission when setpoint temperature changes by..."* parameter on the *"Room temperature controller functions – setpoints"* parameter page determines the temperature value by which the setpoint has to change in order to have the setpoint temperature value transmitted automatically via the object. Temperature value changes between 0.1 C and 25.5°C or 0.1 K and 25.5 K are possible. A setting of "0" at this point will deactivate the automatic transmission of the set temperature.
In addition, the setpoint can be transmitted cyclically. The *"Cyclical transmission of room temperature"* parameter determines the cycle time (1 to 255 minutes). The value "0" (default) will deactivate the cyclical transmission of the setpoint temperature.
It should be noted that in case of deactivated cyclical transmission and deactivated automatic transmission, no setpoint temperature telegrams will be transmitted anymore.

Setting the "R" flag on the *"Setpoint temperature"* object makes it possible to read out the current setpoint. After return of bus voltage, new programming via the ETS or replugging of the user module, the object value will be updated according to the current setpoint temperature value and actively transmitted on the bus.

2.5 Room temperature measurement

The room temperature controller measures the actual temperature cyclically and compares it with the given setpoint temperature. The control algorithm calculates the adjusted actuating variable from the difference between actual and set-temperature.
In order to always ensure a fault-free and effective room temperature control, it is very important to determine the exact actual temperature.

The room temperature controller features an intergrated temperature sensor. Alternatively (e.g. if the room temperature controller has been installed in an unfavourable location or operates in difficult conditions, for example, in a moist atmosphere) or additionally (e.g. in large rooms or halls), a second KNX/EIB temperature sensor externally coupled via the bus or an external sensor at channel 4 of the pushbutton interface can be used to determine the actual value.

When choosing the installation location of the controller or the external sensor, the following points should be considered:

- The controller sensor should not be used in multiple combinations, especially together with flush-mounted dimmers in the same combination.
- The sensors should not be installed in the vicinity of large electrical consumers (heat radiation).
- The sensor should not be installed in the vicinity of heaters or cooling systems.
- The temperature sensor should not be exposed to direct sun.
- The installation of sensors on the inside of an outside wall might have a negative impact on the temperature measurement.
- Temperature sensors should be installed at least 30 cm away from doors or windows and at least 1.5 m above the floor.

2.5.1 Temperature detection and determination of measured value

The *"Temperature detection"* parameter on the *"Room temperature controller function – room temperature measurement"* parameter page will determine which one of the sensors is used for sensing the actual temperature. The following settings are possible:

- *"Internal sensor":*
  The temperature sensor integrated in the room temperature controller is activated. Thus, the actual temperature is determined only locally on the device.
  When parameterized as such, the control will start directly after a reset.

- *"External sensor":*
  The actual temperature is determined only by the external temperature sensor. The internal sensor is deactivated. The external sensor must transmit the detected temperature value to the room temperature controller's 2-byte *"External temperature sensor"* (DPT-ID 9.001) object 24 . Alternatively or additionally, the room temperature controller can cyclically request the current temperature value (set "R" flag for the external sensor). For this purpose, the *"Interrogation interval for external sensor..."* parameter has to be set to a value > “0”. The interrogation interval can be set from 1 minute to 255 minutes. This parameterization will cause the room temperature controller to wait for a temperature value telegram from the external temperature sensor after a reset until the control starts and an actuating variable, if applicable, is output.

- *"Internal and external sensor":*
  With this setting, the internal as well as the external temperature sensor is active. The external sensor must transmit the detected temperature value to the room temperature controller's 2-byte *"External temperature sensor"* (DPT-ID 9.001) object 24 . Alternatively or additionally, the room temperature controller can cyclically request the current temperature value (set "R" flag for the external sensor). For this purpose, the *"Interrogation interval for external sensor..."* parameter has to be set to a value > “0”. The interrogation interval can be set from 1 minute to 255 minutes. This parameterization will cause the room temperature controller to wait for a temperature value telegram from the external temperature sensor after a reset until the control starts and an actuating variable, if applicable, is output.

  The actual-temperature is made up from the two measured temperature values. The weighting of the temperature values is determined by the *"Calculation of temperature based on internal / external measurement"* parameter. Depending on the different locations of the sensors or a possible non-uniform heat distribution inside the room, it is thus possible to adjust the actual temperature measurement. Often, those temperature sensors that are subject to negative external influences (for example, unfavourable location because of exposure to sun or heater or door / window directly next to it) are weighted less heavily.

  Example:
  Room temperature controller installed next to the entrance door (internal sensor). An additional external temperature sensor is installed on an inner wall in the middle of the room below the ceiling.

  Internal sensor: 21.5°C (measuring range of internal sensor: 0 C … + 40°C ±1%)
  External sensor: 22.3 °C
  Calculation of temperature: 30% to 70 %

  Result: $T_{result\ internal} = T_{intern} \cdot 0.3 = 6.45°C$, $T_{result\ external} = T_{external} = 22.3\ °C \cdot 0.7 = 15.61\ C \rightarrow$
  $T_{result\ actual} = T_{result\ intern} + T_{result\ extern} = \underline{22.06\ C}$

If a temperature sensor is connected to channel 4 of the pushbutton interface and if the function *"External temperature sensor for the controller"* is selected on parameter page *"Pushbutton interface – General"*, the measured values of this sensor will be written automatically into the controller's communication object. To prevent malfunctions, no other values must be written into this communication object via a group address. The parameter *"Interrogation interval for external sensor"* is then without function.

2.5.2 Calibrating the measured values

In some cases it may be required to calibrate the temperature measurements of the internal and external sensor. A calibration becomes necessary, if the temperature measured by the sensors stays permanently below or above the actual room temperature in the vicinity of the sensor. The actual room temperature should be determined by a reference measurement with a calibrated temperature measuring device.

The *"Calibration of internal sensor..."* or *"Calibration of external sensor..."* parameter on the *"Room temperature controller function –room temperature measurement"* parameter page allows to parameterize the positive temperature adjustment (increase, factors: 1…127) or the negative adjustment (temperature decrease: factors – 128…-1) in 0.1°C increments. Thus, the calibration is made only once and is the same for all operating modes.

Important:

- The measured value has to be increased, if the value measured by the sensor lies below the actual room temperature. The measured value has to be decreased, if the value measured by the sensor lies above the actual room temperature.
- When the measured values from internal and external sensor are used, the actual value is calculated on the basis of the adjusted value.

2.5.3 Transmitting the actual temperature

The measured actual temperature can be actively transmitted to the bus via the *"Actual temperature"* object 23. The *"Transmission when setpoint temperature changes by..."* parameter on the *"Room temperature controller functions –room temperature measurement"* parameter page determines the temperature value by which the actual value has to change in order to have the actual temperature value transmitted automatically via the object 23. Temperature value changes between 0.1 C and 25.5°C or 0.1 K and 25.5 K are possible. A setting of "0" at this point will deactivate the automatic transmission of the actual temperature.
In addition, the actual value can be transmitted cyclically. The *"Cyclical transmission of room temperature"* parameter determines the cycle time (1 to 255 minutes). The value "0" (default) deactivates the cyclical transmission of the actual temperature value.
Setting the "R" flag on the *"Actual temperature"* object makes it possible to read out the current actual value.
It should be noted that in case of deactivated cyclical transmission and deactivated automatic transmission, no setpoint temperature telegrams will be transmitted anymore.

After return of bus voltage or after new programming with the ETS, the object value will be updated according to the actual temperature value and transmitted to the bus.
If no temperature value telegram has as yet been received from the external sensor when such an external sensor is used, only the value provided by the internal sensor will be transmitted. If only the external sensor is used, the value "0" will be in the object after a reset. For this reason, the external temperature sensor should always transmit the current value after a reset.

2.6 Disable functions of the room temperature controller

2.6.1 Disabling the control function

In certain operating conditions it may be required to deactivate the room temperature control. For example, the control can be switched-off during the dew-point mode of a cooling system or during maintenance work on the heating or cooling system.

The *"Switch-off controller (dew-point mode)"* parameter on the *"Room temperature controller functions"* parameter page enables the *"Disable controller"* object 40 when set to *"via object"*. In addition, the controller disable function can be permanently deactivated when set to "no" (default).
If a "1" telegram is received via the enabled disable object, the room temperature control is completely deactivated. In this case all actuating variables = "0" and the "dew-point operation" LED is lit up (wait for 30 s actuating variable update interval). The controller, however, can be operated in this case.

In the two-stage heating or cooling mode, the additional stage can be separately disabled. The *"Disable object additional stage"* parameter on the *"Room temperature controller functions*" parameter page will enable the *"Disable additional stage"* object 41 when set to "yes". In addition, the disable function of the additional stage can be permanently deactivated when set to "no" (default).
In case a "1" telegram is received via the enabled disable object, the room temperature control is completely deactivated by the additional stage. The actuating variable of the additional stage is "0" while the basic stage continues to operate.

A disable is always deleted after a reset (return of bus voltage, ETS programming operation).

2.7 Valve protection

A valve protection may be carried out periodically in order to prevent the addressed control valves of the heater or cooling system to become calcified or stuck. The *"Valve protection"* parameter on the *"Room temperature controller function"* parameter page will activate the valve protection when set to "yes".

This type of protection is generally started not only for non-active actuating variable outputs, i.e. for outputs which have not requested any heating or cooling energy over the past 24 hours.
For these outputs, the controller will periodically set the actuating variable to the maximum value once a day for a duration of approx. 5 minutes based on the following parameterization:

Actuating variable output not inverted: 1-bit actuating variable: "1", 1-byte actuating variable: "255",
Actuating variable inverted: 1-bit actuating variable: "0", 1-byte actuating variable: "0".

Thus, even valves closed for prolonged periods will be shortly opened on a regular basis.

Parameters		
Description:	**Values:**	**Remarks:**
Controller general		
Control option	**heating** cooling heating and cooling basic and additional heating basic and additional cooling basic and additional heating and cooling	Setting of the control option
Only in two-stage heating or cooling mode		
Disable object additional stage		The additional stages can be separately disabled via the bus. The parameter enables the disable object.
	no	The additional stages cannot be separately disabled.
	yes	The additional stages cannot be separately disabled via the disable object.
Only with control options "heating and cooling" or "basic and additional heating and cooling"		
Transmit actuating variables for heating and cooling to shared object	**no** yes	If the parameter is set to "yes", the actuating variable will be transmitted to a shared object during heating or cooling. This function is used, if the same heating system is used to cool the room in the summer and used to heat the room in the winter.
Type of heating control (if applicable, for basic and additional stage)	**continuous PI control** switching PI control PWM) switching 2-state control (ON/OFF)	Selects a control algorithm (PI or 2-point) with data format (1-byte or 1-bit) for the heating system.
Only with continuous or switching PI control:		
Type of heating (if applicable, for basic and additional stage)	**hot-water heating (5 K / 150 min)** underfloor heating (5 K / 240 min) electric heating (4 K / 100 min) fan convector (4 K / 90 min) split unit (4 K / 90 min)	Adapts the PI algorithm to different heating systems using experience values for the proportional range and reset time control parameters.
	via control parameter	Separate input of control parameter.
Only with separate input of control parameter:		
Proportional range heating (10 … 127) * 0.1 K	10...127, **50**	Separate setting of the "proportional range" control parameter. Only if "type of heating" = "via control parameter".
Reset time heating (0 ... 255) * 1 min; 0 = inactive	0...250, **150**	Separate setting of the "reset time" control parameter. Only if "type of heating" = "via control parameter".

Only with switching 2-state control:		
Upper hysteresis of the 2-state controller heating (5 ... 127) * 0.1 K	5...127, **5**	Definition of switch-on and switch-off temperatures for heating. Only if "type of heating control" = "2-state".
Lower hysteresis of the 2-state controller heating (-128 ... –5) * 0.1 K	-128...-5, **-5**	Definition of switch-on and switch-off temperatures for heating. Only if "type of heating control" = "2-state".
Type of cooling control (if applicable, for basic and additional stage)	**Continuous PI control** Switching PI control PWM) Switching 2-point control (ON/OFF)	Selects a control algorithm (PI or 2-state) with data format (1 byte or 1 bit) for the cooling system. Only if "Transmit actuating variables for heating and cooling to shared object" = "no". If "Transmit actuating variables for heating and cooling to shared object" = "yes", the parameter settings for "Type of heating control" will be accepted.
Only with continuous or switching PI control:		
Type of cooling (if applicable, for basic and additional stage)	**Cooling ceiling (5 K / 240 min)** Fan convector (4 K / 90 min) Split unit (4 K / 90 min)	Adapts the PI algorithm to different cooling systems using experience values for the proportional range and reset time control parameters.
	Via control parameter	Separate input of control parameter.
Only with separate input of control parameter:		
Proportional range cooling (10 ... 127) * 0.1 K	10...127, **50**	Separate setting of the "proportional range" control parameter. Only if "Type of cooling" = "via control parameter".
Reset time cooling (0 ... 255) * 1 min; 0 = inactive	0...255, **240**	Separate setting of the "reset time" control parameter. Only if "Type of cooling" = "via control parameter".
Only with switching 2-state control:		
Upper hysteresis of the 2-state controller cooling (5 ... 127) * 0.1 K	5...127, **5**	Defines the switch-on and switch-off temperatures for cooling. Only if "Type of cooling control" = "2-state".
Lower hysteresis of the 2-state controller cooling (-128 ... –5) * 0.1 K	-128...-5, **-5**	Defines the switch-on and switch-off temperatures for cooling Only if "Type of cooling control" = "2-state".

Operating mode switch-over	**via value (1-byte)**	The switch-over of the operating modes via the bus takes place according to the KONNEX specification via a 1-byte value object. In addition, a higher-ranking forced-control object is available for this setting.
	Via switching (4 x 1-bit)	The 'classic' switch-over of the operating modes via the bus is via separate 1-bit objects.
Operating mode after reset	**Comfort mode** standby mode Night mode Frost /heat protection	Defines the operating mode which is set, for instance after a reset caused by bus voltage return or by new programming.
Only with operating mode switch-over via 1-bit objects:		
Operating mode when all bit objects = 0 (preferred state)	Comfort mode standby mode Night mode Frost /heat protection **last state before change to 0**	Defines the operating mode which is activated when all 1-bit operating mode objects have value "0".
Setpoint values		
Basic temperature after reset (7 ... 4) * 1 °C	7.0 °C … + 40 °C, **21 °C**	Defines the basic setpoint value after the initialization.
Basic temperature setpoint change	deactivated **permit via the bus**	Determines whether an adjustment of the basic temperature possible via the bus.
Adopt basic temperature setpoint permanently	**NO** YES	This parameter determines whether the basic temperature value which has been adjusted via the bus is to be permanently (setting "yes") or only temporarily (setting "no") stored in memory. When set to "yes" the changed basic value will be retained even after a switch-over of the operating mode and after a reset. Only if "Basic temperature setpoint change" = "permit via the bus"
Frost protection setpoint temperature (7...40) * 1 °C	7 °C … + 40 °C, **7 °C**	Defines the setpoint temperature with activated frost protection. Only if "Control option" = "heating" or "heating and cooling", if applicable, with additional stage
Heat protection setpoint temperature (7...45) * 1 °C	7 °C … + 45 °C, **35 °C**	Defines the setpoint temperature with activated heat protection. Only if "Control option" = "cooling" or "heating and cooling", if applicable, with additional stage.

Dead zone position:		The comfort setpoint temperatures for heating and cooling are derived from the basic setpoint in consideration of the adjusted dead zone. The dead zone (temperature zone for which there is neither heating nor cooling) is the difference between the comfort setpoint temperatures.
	Symmetrical	Symmetrical: The dead zone can be equally positioned above and below the basic setpoint (e.g. +/- 1K). The comfort setpoint temperatures are derived directly from the basic setpoint resulting from the half dead zone.
	Asymmetrical	Asymmetrical: With this setting, the comfort setpoint temperature for heating equals the basic setpoint! The preset dead zone takes only effect from the basic setpoint in the direction of comfort temperature for cooling. Thus the comfort setpoint temperature for cooling is derived directly from the comfort setpoint for heating. Only with the "heating and cooling" or "basic / additional heating/cooling" mixed modes !
Dead zone between heating and cooling (0...127) * 0,1 K	0 ... 127, **20**	The comfort setpoint temperatures for heating and cooling are derived from the basic setpoint in consideration of the adjusted dead zone. The dead zone (temperature zone for which there is neither heating nor cooling) is the difference between the comfort setpoint temperatures. Only with the "heating and cooling" or "basic / additional heating/cooling" mixed modes
Stage offset from the basic to the additional stage (0...127) * 0.,1 K	0 ... 127, **20**	In a two stage control mode it is necessary to determine the temperature difference to the basic stage with which the additional stage is to be incorporated into the control Only in two stage control operation!
Transmit when setpoint temperature changes by (0...255) * 0,1 K	0 ... 255, **1**	Determines the size of the value change required for automatic transmission of the current value via the "Setpoint temperature" object. 0 = no automatic transmission
Cyclical transmission of setpoint temperature (0...255) * 1 min; 0 = inactive	0 ... 255, **0**	Determines whether the setpoint temperature is to be cyclically output via the "Setpoint temperature" object.
Adjustment of setpoint to higher temperatures	0 K +1 K +2 K **+3 K** +4 K +5 K +6 K +7 K +8 K +9 K +10 K	Determines the maximum adjustment range for the upward adjustment of the basic setpoint temperature. (cf. "Basic temperature setpoint change" parameter)

Adjustment of setpoint to lower temperatures (-10...0) * 1 K	0 K -1 K -2 K **-3 K** -4 K -5 K -6 K -7 K -8 K -9 K -10 K	Determines the maximum adjustment range for the downward adjustment of the basic setpoint temperature. (cf. "Basic temperature setpoint change" parameter)
Decreasing the setpoint temperature in standby mode (heating) (-128...0) * 0.1 K	-128 ... 0, **-20**	The value by which the standby setpoint temperature for heating is lowered compared to the basic setpoint. Only if "Control option" = "heating" or "heating and cooling", if applicable, with additional stages.
Increasing the setpoint temperature in night mode (heating) (-128...0) * 0,1 K	-128 ... 0, **-40**	The value by which the night setpoint temperature for heating is lowered compared to the basic setpoint. Only if "Control option" = "heating" or "heating and cooling", if applicable, with additional stages.
Decreasing the setpoint temperature in standby mode (cooling) (0...127) * 0.1 K	0 ... 127, **20**	The value by which the standby setpoint temperature for cooling is lowered compared to the basic setpoint. Only if "Control option" = "cooling" or "heating and cooling", if applicable, with additional stages.
Increasing the setpoint temperature in night mode (cooling) (0...127) * 0.1 K	0 ... 127, **40**	The value by which the night set-temperature for cooling is lowered compared to the basic setpoint. Only if "Control option" = "cooling" or "heating and cooling", if applicable, with additional stages.
Controller functions		
Presence detection	**none**	
	presence key	The presence detection takes place via a presence key on the controller or via the presence object (e.g. presence detector). A press on the presence key activates the comfort mode extension.
	presence detector	The presence detection takes place via an external presence detector. The detector will be coupled via the presence object. If presence is detected, the comfort mode will be activated as long as the presence detector detects movement. The presence key is without function.

Only with presence key:		
Duration of comfort mode extension (0 .. 255) * 1 min; 0 = OFF	0 ... 255, 3**0**	During a presence detection the controller can temporarily switch into the comfort extension – depending on the active operating mode. The parameter determines the time after which the comfort mode extension is automatically terminated. Only if "Presence detection" = "presence key".
Switching-over between heating and cooling		In parameterized mixed mode it is possible to switch-over between heating and cooling.
	automatically	Depending on the operating mode and the room temperature, the switch-over takes place automatically.
	via object (heating/cooling switch-over)	Switch-over takes place only via the *"Heating / cooling switch-over"* object 35. Only with the "heating and cooling" or "basic / additional heating/cooling" mixed modes.
Automatic transmission heating/cooling switch-over	**when control option changes** when output variable changes	Determines when an operating mode switch-over telegram will be transmitted automatically to the bus via the "*Heating / cooling switch-over*" object 35. Only if "Switch-over between heating and cooling" = "automatic"!
Switch-off controller (dew-point)	**no** via the bus	This parameter enables the "Disable controller" object 40. There is no control until enabled (actuating variables = 0). When controller disable is active (dew-point operation), the " 💧 " LED lights up.
Valve protection	**no** yes	The valve is periodically opened (every 24 hours). Works against calcification and thus prevents the valve from getting stuck.
Room temperature measurement		
Temperature detection		Determines which sensor will be used for room temperature measurement.
	internal sensor	Internal sensor: built-in sensor of the controller
	external sensor	External sensor: an external sensor coupled via the bus, e.g. for severe measuring conditions (swimming pools or similar).
	internal and external sensor	Internal and external sensor: Both sensors are used, for example, in large rooms.

Determination of measured value from internal / external ratio	10% to 90 % 20% to 80 % 30% to 70 % 40% to 60 % **50% to 50 %** 60% to 40 % 70% to 30 % 80% to 20 % 90% to 10 %	Determines the weighting of the measured temperature value for the internal and external sensors. That results in an overall value which will be used for the further evaluation of the room temperature. Only if "Temperature detection" = "internal and external sensor"
Calibration of internal sensor (-128...127) * 0.1 K	-128 ... 127, **0**	Determines the value by which the internal sensor's room temperature value is calibrated. Only if "Temperature detection" = "internal sensor" or "internal and external sensor".
Calibration of external sensor (-128...127) * 0.1 K	-128 ... 127, **0**	Determines the value by which the external sensor's room temperature value is calibrated. Only if "Temperature detection" = "external sensor" or "internal and external sensor".
Interrogation interval for external sensor (0...255) * 1 min; 0 = inactive	0 ... 255, **0**	Determines the interrogation interval for the external sensor's temperature value . 0" = sensor transmits its temperature value automatically. Only if "Temperature detection" = "external sensor" or "internal and external sensor".
Transmission when room temperature changes by (0..255) * 0,1 K; 0 = no automatic transmission	0 ... 255, **3**	Determines the size of the value change of the room temperature after which the current values are automatically transmitted to the bus via "Actual temperature" object 23.
Cyclical transmission of room temperature (0...255) * 1 min; 0 = inactive	0 to 255, **15**	Determines whether or when the determined room temperature is to be cyclically output via "Actual temperature" object 23.
Actuating variable and status output		
Automatic transmission if value changes by... (0...100) * 1 %; 0 = inactive	0 to 100, **3**	Determines the size of the actuating variable change that will cause the automatic transmission of the continuous actuating variables via the actuating variable objects. Only if at least one type of control is parameterized to "continuous PI control"!
Cycle time of switching variable (1...255) * 1 min	1 to 255, **15**	Determines the cycle time for the pulse-width-modulated actuating variable (PWM). Only if at least one type of control is parameterized to "switching PI control (PWM)".

Cycle time for automatic transmission (0...255) * 1 min; 0 = inactive	0 to 255, **10**	Time interval for the cyclical transmission of the actuating variable via the actuating variable objects. Only if at least one type of control is parameterized to "continuous PI control" or "switching 2-state control".
Output of actuating variable heating	inverted (closed when sourcing current) **normal (opened when sourcing current)**	continuous: Act. var.= 100 % - normal act. var. switching: Act. var. = 1 - normal act. var. normal actuating variable output heating Only if "Control option" = "heating" or "heating and cooling".
Output of actuating variable basic stage heating	inverted (closed when sourcing current) **normal (opened when sourcing current)**	continuous: Act. var. = 100 % - normal act. var. switching: Act. var. = 1 - normal act. var. normal actuating variable output basic stage heating Only if "control option= basic and additional heating" or "basic/additional heating/cooling".
Output of actuating variable additional stage heating	inverted (closed when sourcing current) **normal (opened when sourcing current)**	continuous: Act. var. = 100 % - normal act.var. switching: Act. var. = 1 - normal act. var. normal actuating variable output additional stage heating Only if "Control option" = "basic and additional heating" or "basic/additional heating/cooling".
Output of actuating variable cooling	inverted (closed when sourcing current) **normal (opened when sourcing current)**	continuous: Act. var. = 100 % - normal act. var. switching: Act. var. = 1 - normal act. var. normal actuating variable output cooling Only if "Control option" = "cooling" or "heating and cooling".
Output of actuating variable basic stage cooling	inverted (closed when sourcing current) **normal (opened when sourcing current)**	continuous: Act. var. = 100 % - normal act. var. switching: Act. var. = 1 - normal act. var. normal actuating variable output basic stage cooling Only if "Control option" = "basic and additional cooling" or "basic/additional heating/cooling".
Output of actuating variable additional stage cooling	inverted (closed when sourcing current) **normal (opened when sourcing current)**	continuous: Act. var. = 100 % - normal act. var. switching: Act. var. = 1 – normal act. var. normal actuating variable output additional stage cooling Only if "Control option" = "basic and additional cooling" or "basic/additional heating/cooling".
Heating message	**no** yes	Enables the "heating" message function and thus the "Heating message" object 37.
Cooling message	**no** yes	Enables the "cooling" message function and thus the "message cooling" object 38.

Controller status		The controller can output its current operating status.
	no status	No status will be output.
	controller general	The controller status is generally output via the 1-byte object (object 36 *"Controller status"*).
	transmitting individual status	The controller status preset by the "Individual status" parameter will be output via the 1-bit object (Object 36 *"Controller status"*).
Individual status	**comfort mode active** standby mode active night mode active frost/heat protection active controller disabled heating/cooling controller inactive frost alarm	Defines the controller status to be transmitted. Only if "Controller status" = "transmit individual status".

Software information
