Product name:	**Push sensor 2 plus, 2fold / 3fold / 6fold F-line**
Design:	Flush-mounted (uP)
Article-no.:	**2052 … 2fold** **2053 … 3fpld** **2056 … 6fold**
ETS search path:	Gira Giersiepen / Push button / Push button, xfold / push sensor 2 plus, xfold F-line Gira Giersiepen / Heating, air conditioning / Thermostat / push sensor 2 plus, xfold F-line Gira Giersiepen / Phys. sensors / Temperature / push sensor 2 plus, xfold F-line Gira Giersiepen / Push button / Push button, general/ push sensor 2 plus, xfold F-line

Functional description:

Push sensor functionality:
Depending on the parameterized software, the push sensor 2 plus will send telegrams to the instabus EIB if one of its keys is actuated. These can, for example, be telegrams for switching or touch control, dimming (also single-key dimming), or for shutter/blind control using various operating concepts. You can also program value transmitter functions such as dimming value transmitters, light scene extensions for recalling externally or internally stored light scenes, 2-byte analog value transmitters (temperature or brightness value transmitters) or 1-byte universal value transmitters (continuous run between start and target values). In this connection, you can freely assign the 4 keys/2 rockers of the dual-type push sensor 2 plus or the 10 keys/5 rockers of the quintuple-type push sensor 2 plus to the various functions. Distinction is made between key and rocker functions.
In addition, the push sensor 2 plus offers the possibility to disable specific individual keys or rockers or the entire push sensor. You can also operate the room temperature controller integrated in the push sensor 2 plus by actuating the latter.

Room temperature controller functionality:
You can use the push sensor 2 plus for single-room temperature control. In this connection, the controller can distinguish between up to two control loops which optionally have their own temperature setpoints and trigger them. Triggered by control loop 1, heating/cooling and the operating modes of the two control loops are switched over together. Thus, for example, you can use separate algorithms to control the radiators on the wall and the floor heating within one room.
Depending on the heating/cooling switch-over, the current temperature setpoint and on the room temperature, a variable for heating or cooling control can be sent to the instabus EIB for each of the two control loops.
In one control loop, the room temperature can be sensed either by the internal (in the push sensor enclosure) or by an optionally external temperature sensor. If the second control loop is activated the room temperature of the first loop will be sensed by the internal sensor, whereas the room temperature of the second loop will be determined by the external sensor.
If you use only one control loop you can activate another stage in addition to the heating or cooling basic stage to run an additional heater and/or cooling unit. In this connection, you can set the temperature setpoint difference between the basic and the additional stage by a parameter. For major deviations between the temperature setpoint and the actual temperature, you can activate this additional stage to heat up or cool down the room more quickly. You can assign different control algorithms to the basic and additional stages.

The controller has five different operating modes (comfort, standby, economy, frost/heat protection and disabled) with their separate temperature setpoints for heating or cooling. For heating and cooling functions, you can select continuous or switching PI or switching 2-point control algorithms.
A heating timer allows automatic operating mode control, depending on the time of the day and on the day of the week.

General functions:
Two independent 1-bit or 1-byte control functions permit the temperature-dependent or time-dependent transmission of control commands to the bus.
You can use an object to switch on or off the description field light. If the push sensor 2 plus is pulled off the bus coupling unit (BCU) an alarm message (of 1-bit or 1-byte type) can be sent.

instabus KNX/EIB System

Sensor

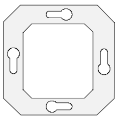
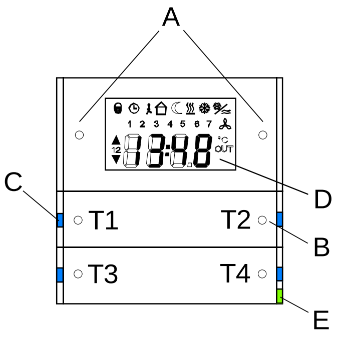


Illustration:

2fold



Dimensions:

Width: 70 mm

Height: 70 mm

Depth: 13 mm (without PEI)

Controls:

A: Rocker 1 (to control the room temperature controller functions).

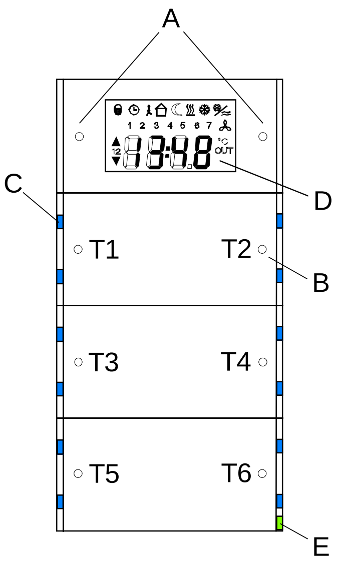
B: Rockers 2+3 (push sensor functions).

C: 4 status-LEDs (blue)
(to indicate the statuses of rockers 2+3).

D: Display with illumination (white).

E: Operation LED (green) (goes out automatically off, if the status LED lying over it lights up!)

3fold



Width: 70 mm

Height: 140 mm

Depth: 13 mm (without PEI)

A: Rocker 1 (to control the room temperature controller functions).

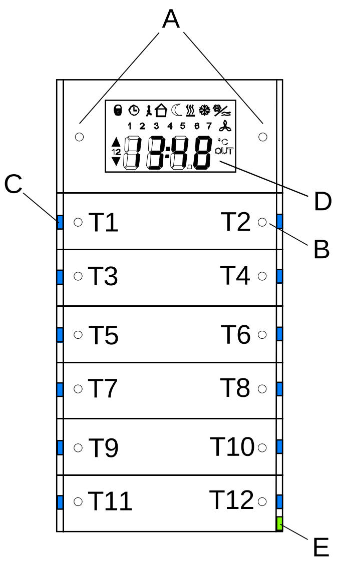
B: Rockers 2-4 (push sensor functions).

C: 12 status-LEDs (blue) (to indicate the statuses of rockers 2-6 / for each key 2 LED).

D: Display with illumination (white).

E: Operation LED (green) (goes out automatically off, if the status LED lying over it lights up!).

6fold



Width: 70 mm

Height: 140 mm

Depth: 13 mm (without PEI)

A: Rocker 1 (to control the room temperature controller functions).

B: Rockers 2-7 (push sensor functions).

C: 12 status-LEDs (blue) (to indicate the statuses of rockers 2-7).

D: Display with illumination (white).

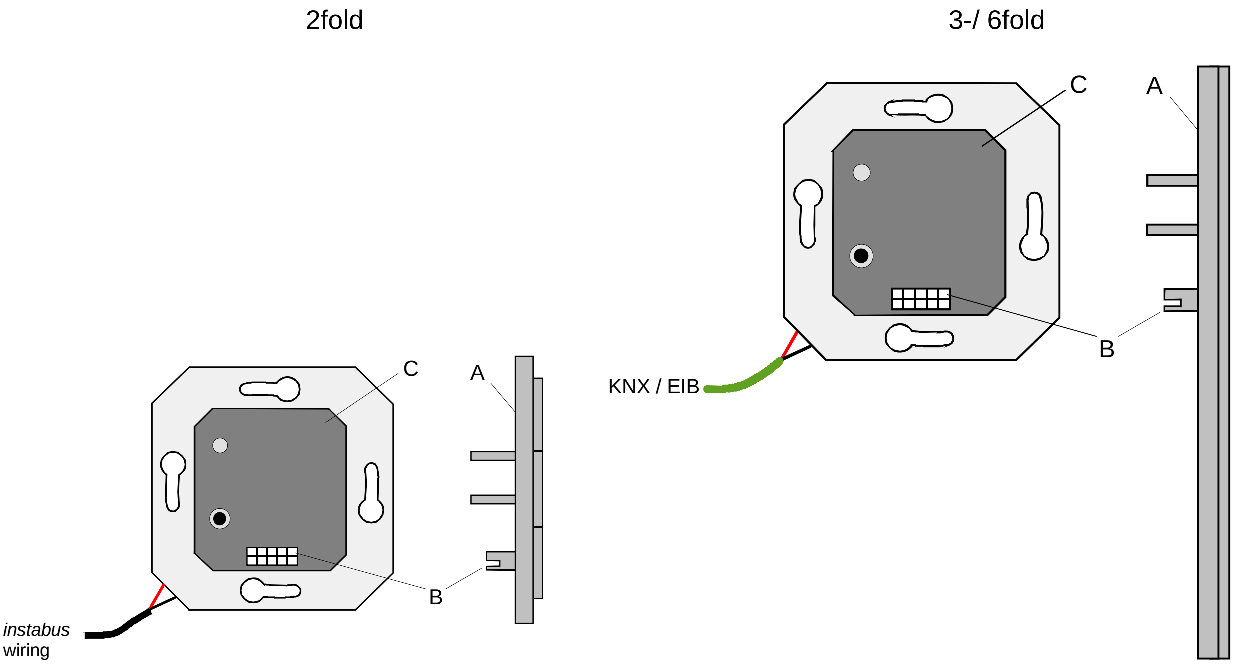
E: Operation LED (green) (goes out automatically off, if the status LED lying over it lights up!).

Technische Daten:

Protective system:	IP 20
Protection class:	III
Mark of conformity:	KNX / EIB
Ambient temperature:	-5 °C to +45 °C
Storage/transport temperature:	-20 °C to +60 °C (storage above +45 °C will shorten the life)
Fitting position:	Any (vertical to be preferred/display unit on top)
Minimum distances:	None
Type of fixing:	Plugging onto flush-mounted BCU (Please note: Remarks on the hardware.)
KNX / EIB supply	
Voltage:	21 - 32 VDC (through flush-mounted BCU)
Power consumption:	150 mW typ. (through flush-mounted BCU)
Connection:	2 x 5-pole male connector (PEI)
External supply	---
Room temperature controller (internal temperature sensor):	
Measuring range:	0 °C to + 40 °C ±1 %
Resolution:	0.1 K
Air humidity:	0 % to 95 % (no condensation)
Internal clock:	
Resolution:	1 minute
Time error:	8 minutes per day, max. To keep the time error low you should set and thus update the internal clock via the bus every hour.
Response to voltage failure	
Bus voltage only:	All object values will be deleted. Push sensor function: No response, LEDs will go out. Room temperature controller: No response, control OFF.
Mains voltage only:	---
Bus and mains voltages:	---
Response to recovery	
Bus voltage only:	Push sensor function: No response. Room temperature controller: The controller will initialize. Depending on the parameterization, various temperature values and the status will be sent and switch-over objects updated.
Mains voltage only:	---
Bus and mains voltages:	---
Input:	---
Output:	---

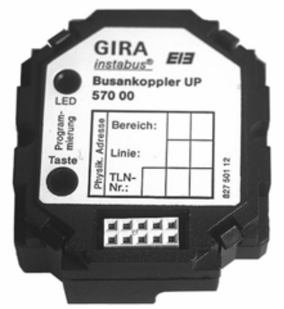
Wiring diagram: **Terminal assignment:**

The push sensor 2 plus must be plugged onto a flush-mounted bus coupling unit (UP BCU 1).


2fold

3-/ 6fold

A: push sensor 2 plus, dual/quintuple type
B: Physical external interface
C: BCU

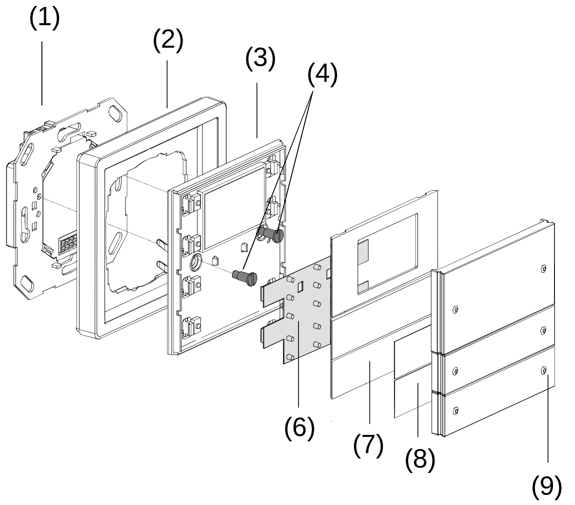


BCU without mounting plate

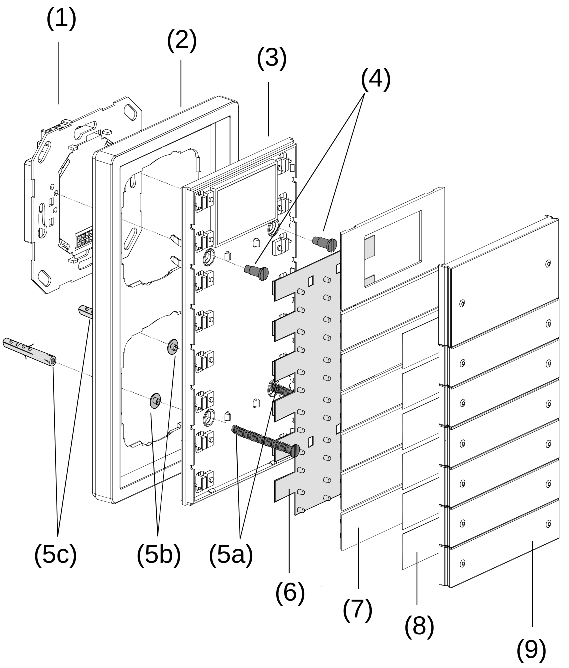
Remarks on the hardware:

- The push sensor 2 plus may only be plugged onto BCUs of the "new generation" which have a round programming key (see BCU shown above). Plugging the push sensor 2 plus onto earlier flush-mounted BCU models can cause malfunctioning.
- All variants are exclusively plugged onto a flush-mounted BCU. The pusbutton sensors 2 plus 3fold and 6fold can only be installed with a double design frame without middle strip.
- The operation LED (green) goes out automatically when the status-LED above lights up
- For programming of the application data by means of the ETS, the pushbutton sensor must be plugged into the BCU.

Assembly



2fold



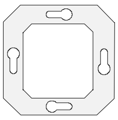
3-/ 6fold

Procedure:

1.) Assembly without anti-theft protection:
Place the cover frame (2) and the user module (3) on a flush-mounted BCU (1).

2.) Assembly with removal protection:
The device is protected against theft by fastening it with screws on the bus coupler insert.
- remove the cover frame (9),
- remove the rocker carrier (7) carefully with a screwdriver or with your fingernail,
- lift off the ESD protection mat (6),
- place the cover frame (2) and the user module (3) on the flush-mounted BCU already in place (1),
- screw the pushbutton sensor to the insert using only the screw set (4, 5a, 5b, 5c) supplied with the device,
- put the ESD protection mat (6) carefully back in place.
Important: proper functioning can only be guaranteed when the ESD protection mat is in place. Otherwise
risk of irreparable damage to the device in operation by electro-static discharge.
- Fit the rocker carrier (7), the inscription foil (8) and the rocker cover (9) by snap-fastening them on the device.

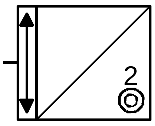
When the internal temperature sensor is used for room temperature measurements, it is recommended to install the pushbutton sensor at least 30 cm away from doors or windows and at least 1.5 m above the floor.



Software Description:

ETS search path:

ETS-symbol:



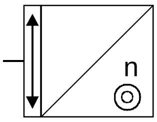
Push button / Push button, xfold / push sensor 2 plus, 2fold F-line
Heating, air conditioning / Thermostat / push sensor 2 plus, 2fold F-line
Phys. sensors / Temperature / push sensor 2 plus, 2fold F-line
Push button / Push button, general/ push sensor 2 plus, 2fold F-line

Application:

Brief description:	Name:
Multifunktion plus F-line Push sensor with room temperature controller	Multifunktion plus F-line 181201

ETS search path:

ETS-symbol:



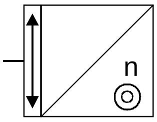
Push button / Push button, xfold / push sensor 2 plus, 3fold F-line
Heating, air conditioning / Thermostat / push sensor 2 plus, 3fold F-line
Phys. sensors / Temperature / push sensor 2 plus, 3fold F-line
Push button / Push button, general/ push sensor 2 plus, 3fold F-line

Application:

Brief description:	Name:
Multifunktion plus F-line Push sensor with room temperature controller	Multifunktion plus F-line 181301

ETS search path:

ETS-symbol:



Push button / Push button, xfold / push sensor 2 plus, 6fold F-line
Heating, air conditioning / Thermostat / push sensor 2 plus, 6fold F-line
Phys. sensors / Temperature / push sensor 2 plus, 6fold F-line
Push button / Push button, general/ push sensor 2 plus, 6fold F-line

Application:

Brief description:	Name:
Multifunktion plus F-line Push sensor with room temperature controller	Multifunktion plus F-line 181601

Push sensor functionality

General

- Free assignment of the switching/key actuation, dimming, shutter/blind, light scene extension/recall, dimming value transmitter, analog value transmitter and room temperature controller operation functions to the 4 keys (dual type) or 10 keys (quintuple type), respectively, when the key function is activated.
- Free assignment of the switching, dimming, shutter/blind, universal value transmitter and room temperature controller operation functions to the 2 rockers (dual type) or 5 rockers (quintuple type), respectively, when the rocker function is activated.
- Status indicator possible by 4 (dual type) or 10 (quintuple type) LEDs (status indicator for rocker function possible via objects, and status or actuation indicator possible when the key function is active).
- Even though certain keys or rockers have been given "no function", the status-LEDs can be triggered via objects.
- An inhibit object for disabling individual keys or rockers is available (the polarity of the inhibit object being selectable).

Switching/keying actuation function

- The command to be raised when a key is being pressed or released (ON, OFF, TOGGLE, no function) can be selected.
- Single-key operation is possible for the rocker function
(only for "command on pressing a rocker = left = TOGGLE, right = toggle").
- The function of the status-LED for the key function or the status indicator for the rocker function, respectively, can be parameterized.

Dimming function

- The time between dimming and switching and the dimming step width can be adjusted.
- Telegram repetition and sending stop telegrams are possible.
- Single-key operation is possible for the rocker function
(only for "command on pressing a key = left = TOGGLE, right = toggle").
- The function of the status-LED for the key function or the status indicator for the rocker function, respectively, can be parameterized.

Shutter/blind function

- You can select the key function (UP, DOWN).
- The operating concept (STEP - MOVE - STEP or MOVE - STEP, respectively) can be parameterized.
- Time between short-time mode and long-time mode is adjustable (for step - move - step only).
- The lamella adjustment (time during which a move command can be completed by releasing the key) can be preset.
- The function of the status-LED for the key function or the status indicator for the rocker function, respectively, can be parameterized.

Value transmitter/light scene extension function (for key function only)

- Dimming value transmitter key function (1 byte) or light scene recalling can be parameterized with/without storage function.
- Long-time key actuation will enable you to change values for the value transmitter function.
- The function of the status-LED can be parameterized.
- When the light scene function is active, internal scenes can also be recalled.

Analog value transmitter function (for key function only)

- The EIS 5 brightness value transmitter, EIS 5 temperature value transmitter and EIS 10 2-byte value transmitter functions can be parameterized.
- Long-time key actuation enables values to be changed.
- The function of the status-LED can be parameterized.

Universal value transmitter function (for rocker function only)

- The start, back-up and target values can be parameterized.
- The back-up value divides the value range into two sub-ranges. The time base and the time factor for a stage of the time ranges can be adjusted.
- The number of stages in the sub-ranges is selectable.
- The direction of control action and the response (start/stop) to key actuation can be parameterized.

Room temperature controller functionality:

General

- Five operating modes: Comfort, standby, economy, frost/heat protection and controller disable (e. g. dew-point operation).
- Switching over of the operating modes to KONNEX by a 1-byte object or by individual 1-bit objects.
- Reading of the room temperature controller functions through an integrated semi-graphical display unit.
- Several operator levels are possible. These can be activated or deactivated.
  - No operation: No local controller operation.
  - First operator level: Setpoint shift at level 0 is possible, switching over the operating mode at level 1 and activation or deactivation of the heating timer and the control function are also possible. In addition, you can access the "contrast adjustment" function from this level.
  - All operator levels: Full access to the device. It allows the user to access the functions of "setpoint adjustment" (if enabled under setpoints) and "setting the switching times" for the heating timer as well as to use the up to two control functions (if time-operated).

Heating/cooling system

- Heating/cooling switch-over: "Heating", "cooling", "heating and cooling", each of them with or without additional stage.
- Up to two control loops with optionally different temperature setpoints and operating mode common switch-over are possible.
  (For two control loops, only "heating" or "cooling" with no additional stage can be activated.)
- PI control (continuous or switching PWM) or 2-point control (switching) can be set as control algorithms.
- Output of continuous (1-byte) or switching (1-bit) variables.
- Control parameters for PI controller (if desired: proportional band, reset time) and 2-point controller (hysteresis) can be set.

Setpoints

- Each operating mode can get assigned its own temperature setpoints (for heating and/or cooling).
- The setpoints for the additional stage are derived from the values of the basic stage by a parameterizable difference between these stages.
- Temporary or permanent setpoint shift is possible by local operation on the device (parameterizable setpoint shift scaling).

Functionality

- Automatic or object-oriented switching over between "heating" and "cooling".
- Controller operation can be optionally disabled through an object.
- Parameterizable comfort mode prolongation period.
- Complete (1-byte) or partial (1-bit) status information can be parameterized and transmitted to the bus via an object.
- Deactivation of the control, the additional stage or of the second control loop via different objects is possible.

Room temperature measurement

- Internal and external room temperature sensors can be used.
- Creating of an internal vs. external measuring value for one control loop and an enabled external sensor can be parameterized.
- For two control loops, the actual temperature value of the second loop is determined by the external sensor.
- The sampling period of the external temperature sensor can be set.
- The actual and the setpoint temperatures can be sent to the bus (even cyclically) after a parameterizable deviation.
- The room temperature measurement (actual value) can be separately adjusted for the internal and the external sensor via certain parameters.

Output of variables

- Separate or common output of variables through one or two objects for "heating and cooling".
- Normal or inverted output of variables can be parameterized.
- Automatic transmitting and cycle time for the output of variables can be parameterized.

Heating timer

- Time- and day-dependent control of the operating modes.
- By local operation, it can be activated or deactivated from the first operator level.
- Besides, the heating timer can be disabled via the bus.

Scene functionality:

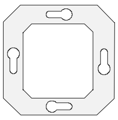
- Eight independent internal scenes.
- Up to eight objects per scene, i. e. eight different commands can be transmitted.
- Selectable data types are switching (ON/OFF), dimming value (0...255 / 0...100%) of shutter/blind long-time commands (UP/DOWN) which can be parameterized per scene and scene object.
- The scenes can be recalled or saved by an extension object.
- Internal scenes can also be recalled without any extension object by push sensor operation.

Control functionality:

- Up to two independent control functions.
- Switching (ON/OFF) or value telegrams (0…255) can be transmitted to the bus as control commands.
- Time- or temperature-dependent triggering of a control command.
- Both control functions can be disabled individually via the bus or by local operation.

Push sensor in general:

- Automatic switching-off of the description field light can be parameterized.
- The description field light can be switched via an object.
- An alarm to be raised after the device has been unplugged from the flush-mounted bus coupling unit possible (1-bit or 1-byte).
- The current time and/or the outside temperature can be indicated by the internal display unit. In this connection, the time can be set through a separate object.

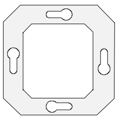


	Object		Object description
□←|	0 - 11	**Status:**	1-bit object for triggering the status-LED of a key or a rocker, resp.
□←|	0 - 11	**Switching:**	1-bit object for sending switching telegrams (ON, OFF).
□ |	12 - 23	**Dimming:**	4-bit object for relative brightness changing between 0 and 100 %.
□ |	0 - 11	**Short operation:**	1-bit object for the short-time mode (STEP) of a shutter/blind.
□ |	12 - 23	**Long operation:**	1-bit object for the long-time mode (MOVE) of a shutter/blind.
□ |	0 - 11	**Light scene extension:**	1-byte object for recalling or storing light scenes (1 - 64).
□ |	0 - 11	**Dimming value transm.:**	1-byte object for sending dimming value telegrams (0 - 255), for example.
□ |	0 - 11	**Analog signal:**	1-byte object for sending temperature values (0 - 40 °C), brightness values (0 - 1500 lux), or for transmitting 2-byte values (0 - 65535).
□ |	0/2/4/6/8 10	**Universal value: transmitter**	1-byte object for sending value telegrams (0 - 255) of the universal value transmitter.
□ |	24	**Switching:**	1-bit object for sending an alarm signal.
□ |	24	**Value:**	1-byte object for sending an alarm signal.
□←|	25	**Disabling function:**	1-bit object for disabling keys or rockers of the push sensor.
□←|	26	**Switching:**	1-bit object for switching the display illumination.
□→|	27	**Switching:**	1-bit object for switching the operation LED.
□←|	28	**Actual temperature:**	2-byte object for the output of the actual temperature. (Possible value range: -99.9 °C to +99.9 °C / internal temperature sensor measuring range: 0 °C to + 40 °C ±1 %.)
□←|	29	**External temp. sensor**	2-byte object for coupling an external room temperature sensor. (Possible value range: -99.9 °C to +99.9 °C.)
□←|	30	**External temp. sensor**	2-byte object for coupling an outside temperature sensor. (Possible value range: -99.9 °C to +99.9 °C.)
□←|	31	**Basic set value:**	2-byte object for basic setpoint external preselection. Depending on the heating/cooling switch-over, the possible value range is restricted by the parameterized frost protection and/or heat protection temperature.
	32	**Basic set value 2nd control loop:**	2-byte object for basic setpoint external preselection for the second control loop using its own setpoints. Depending on the heating/cooling switch-over, the possible value range is restricted by the parameterized frost protection and/or heat protection temperature.

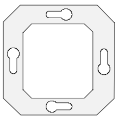
	Object		Object description
□←|	33	**Operating mode switch-over:**	1-byte object for changing the operating modes to konnex.
□←|	33	**Comfort mode:**	1-bit object for changing into the "comfort" mode.
□←|	34	**Standby mode:**	1-bit object for changing into the "standby" mode.
□←|	35	**Economy mode:**	1-bit object for changing into the "economy" mode.
□←|	36	**Frost/heat protection mode:**	1-bit object for changing into the "frost/heat protection" mode.
□←|	37	**Override-object operating mode:**	1-byte object for higher-order forced control of the operating modes to konnex.
□←|	38	**Presence object:**	1-bit object (bidirectional) which sends the status of the presence key to the bus (if the presence object has been enabled the presence key can be parameterized under the push sensor functionality), or through which a presence detector can, for example, be connected. (Presence = "1", no presence = "0".)
□←|	39	**Window status:**	1-bit object for the connection of window contacts. (Window open = "1", window closed = "0".)
□←|	40	**Heating / cooling:**	1-bit object for switching over between the "heating" and "cooling" modes if this is not done automatically by the controller (object value 1: heating; object value 0: cooling). During automatic switch-over, the active operating mode can be transmitted (depending on the parameter).
□ |	41	**Controller status:**	1-byte object for general status feedback or 1-bit object for individual status feedback of parameterized controller functions.
□ |	42	**Heating indication:**	1-bit object for indicating whether heating energy is requested (object value = "1": energy request; object value = "0": no energy request).
□ |	43	**Cooling indication:**	1-bit object for indicating whether cooling energy is requested (object value = "1": energy request; object value = "0": no energy request).
□←|	44	**Disabling controller operation:**	1-bit object for disabling controller local operation. (Controller operation disabled = "1", controller operation enabled = "0".)
□←|	45	**Disabling controller:**	1-bit object for deactivating the controller (activation of dew-point operation). (Controller deactivated = "1", controller activated = "0".)
□←|	46	**Disabling additional stage:**	1-bit object for disabling the additional stage of the controller. (Additional stage deactivated = "1", additional stage activated = "0".)
□←|	46	**Inhibit 2nd control loop:**	1-bit object for deactivating the second control loop. (Control loop 2 deactivated = "1", control loop 2 activated = "0".)

	Object		Object description
□↵	47	Heating (control loop 1):	1-byte object for the output of the heating mode continuous variable of the first control loop.
□↵	47	Heating (control loop 1):	1-bit object for the output of the heating mode switching or PWM variable of the first control loop.
□↵	47	Basic-heating (1st control loop):	1-byte object for the output of the basic heating mode continuous variable of the first control loop.
□↵	47	Basic-heating (1st control loop):	1-bit object for the output of the basic heating mode switching or PWM variable of the first control loop.
□↵	47	Heating/cooling (1st control loop):	1-byte object for the output of the alternative heating or cooling mode continuous variable of the first control loop. (For variable output through common object.)
□↵	47	Heating/cooling (1st control loop):	1-bit object for the output of the alternative heating or cooling mode switching or PWM variable of the first control loop. (For variable output through common object.)
□↵	47	Basic-heating and cooling (1st control loop):	1-byte object for the output of the alternative basic heating or basic cooling mode continuous variable of the first control loop. (For variable output through common object.)
□↵	47	Basic-heating and cooling (1st control loop):	1-bit object for the output of the alternative basic heating or basic cooling mode switching or PWM variable of the first control loop. (For variable output through common object.)
□↵	48	Additional-heating (1st control loop):	1-byte object for the output of a additional heating mode continuous variable of the first control loop.
□↵	48	Additional-heating (1st control loop):	1-bit object for the output of the additional heating mode switching or PWM variable of the first control loop.
□↵	48	Additional-heating and cooling (1st control loop):	1-byte object for the output of the alternative additional heating or additional cooling mode continuous variable of the first control loop. (For variable output through common object.)
□↵	48	Additional-heating and cooling (1st control loop):	1-bit object for the output of the alternative additional heating or additional cooling mode switching or PWM variable of the first control loop. (For variable output through common object.)
□↵	49	Cooling (1st control loop):	1-byte object for the output of the cooling mode continuous variable of the first control loop.
□↵	49	Cooling (1st control loop):	1-bit object for the output of the cooling mode switching or PWM variable of the first control loop.
□↵	49	Basic-cooling (1st control loop):	1-byte object for the output of the basic cooling mode continuous variable of the first control loop.
□↵	49	Basic-cooling (1st control loop):	1-bit object for the output of the basic cooling mode switching or PWM variable of the first control loop.

Object		Object description
50	**Additional-coolin (1st control loop):**	1-byte object for the output of the additional cooling mode continuous variable of the first control loop.
50	**Additional-cooling (1st control loop):**	1-bit object for the output of the additional cooling mode switching or PWM variable of the first control loop.
51	**Heating (2nd control loop):**	1-byte object for the output of the heating mode continuous variable of the second control loop.
51	**Heating (2nd control loop):**	1-bit object for the output of the heating mode switching or PWM variable of the second control loop.
51	**Heating (1st control loop):**	1-byte object for the PWM variable to feed back the continuous variable value for the heating mode. (Only for one control loop.)
51	**Basic-heating (1st control loop):**	1-byte object for the PWM variable to feed back the continuous variable value for the basic heating mode. (Only for one control loop.)
52	**Additional-heating (1st control loop):**	1-byte object for the PWM variable to feed back the continuous variable value for the additional heating mode. (Only for one control loop.)
53	**Cooling (2nd control loop):**	1-byte object for the output of the cooling mode continuous variable of the second control loop.
53	**Cooling (2nd control loop):**	1-bit object for the output of the cooling mode switching or PWM variable of the second control loop.
53	**Cooling (1st control loop):**	1-byte object for the PWM variable to feed back the continuous variable value for the cooling mode. (Only for one control loop.)
53	**Basic-cooling (1st control loop):**	1-byte object for the PWM variable to feed back the continuous variable value for the basic cooling mode. (Only for one control loop.)
54	**Additional-cooling (1st control loop):**	1-byte object for the PWM variable to feed back the continuous variable value for the additional cooling mode. (Only for one control loop.)
55	**Setpoint temperature:**	2-byte object for the output of the current temperature setpoint of the first control loop. Depending on the operating mode, the possible value range is restricted by the parameterized frost protection and/or heat protection temperature.
56	**Setpoint temperature 2nd control loop:**	2-byte object for the output of the current temperature setpoint of the second control loop. Depending on the operating mode, the possible value range is restricted by the parameterized frost protection and/or heat protection temperature.



	Object		Object description
□↵	**57**	**Time signal:**	3-byte object for receiving the current time via the bus.
□|	**58**	**Control function 1:**	1-bit object for transmitting the switching command of the first control function.
□|	**58**	**Control function 1:**	1-byte object for transmitting the switching command of the first control function.
□↵	**59**	**Disabling control function 1:**	1-bit object for inhibiting the first control function. (The polarity can be parameterized.)
□|	**60**	**Control function 2:**	1-bit object for transmitting the switching command of the second control function.
□|	**60**	**Control function 2:**	1-byte object for transmitting the switching command of the second control function.
□↵	**61**	**Disabling control function 2:**	1-bit object for inhibiting the second control function. (The polarity can be parameterized.)
□↵	**62**	**Disabling heating timer:**	1-bit object for deactivating the heating timer. (The polarity can be parameterized.)
□|	**63 - 70**	**Scene outputs 1 - 8:**	1-bit objects for transmitting the up to eight switching commands of a scene.
□|	**63 - 70**	**Scene outputs 1 - 8:**	1-byte objects for transmitting the up to eight value commands of a scene.
□|	**63 - 70**	**Scene outputs 1 - 8:**	1-bit objects for transmitting the up to eight shutter/blind long-time commands of a scene.
□↵	**71**	**Scene extension:**	1-byte object for externally recalling or saving the internally stored eight light scenes.



Number of addresses (max.):	77	Dynamic table management	Yes ☐	No ☒
Number of assignments (max.):	200	Maximum table length	277	
Communication objects:	66			

Push sensor functions:

The following objects only apply to "actuation of the rockers = key function":

Function: No function (for all keys) [2]

Object		Function	Name	Typ	Flag
🟨↵	0-11	Status	Key 3 - Key 14 [1]	1 Bit	C, W

Function: Switching(for all keys) [2]

Object		Function	Name	Typ	Flag
🟨↵	0-11	Switching	Key 3 – Key 14 [1]	1 Bit	C, W, T

Function: Dimming (for all keys) [2]

Object		Function	Name	Typ	Flag
🟨↵	0-11	Switching	Key 3 – Key 14 [1]	1 Bit	C, W, T
🟨	12-23	Dimming	Key 3 – Key 14 [1]	4 Bit	C, T

Function: Shutter/blind (for all keys) [2]

Object		Function	Name	Typ	Flag
🟨	0-11	Short operation	Key 3 – Key 14 [1]	1 Bit	C, T
🟨	12-23	Long operation	Key 3 – Key 14 [1]	1 Bit	C, T

Function: Value transmitter (for all keys) [2]

Object		Function	Name	Typ	Flag
🟨	0-11	Dimming value transmitter	Key 3 – Key 14 [1]	1 Byte	C, T

Function: Analog value transmitter (for all keys) [2]

Object		Function	Name	Typ	Flag
🟨	0-11	Analog signal	Key 3 – Key 14 [1]	2 Byte	C, T

Function: Light scene extension/recall (for all keys) [2, 3]

Object		Function	Name	Typ	Flag
🟨	0-11	Light scene extension	Key 3 – Key 14 [1]	1 Byte	C, T

Funktion: Room temperature controller operation/control function operation/heating timer operation (for all keys) [2]

Object	Function	Name	Typ	Flag

No further objects for the push sensor functionality.

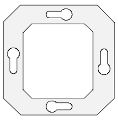
The following objects only apply to "actuation of the rockers = rocker function":

Function: No function (for all rockers) [4]

Object		Function	Name	Typ	Flag
🟨↵	1/3/5/7/9/11	Status	Rocker 2 – Rocker 7 [1]	1 Bit	C, W

Funktion: Switching(for all rockers) [4]

Object		Function	Name	Typ	Flag
🟨↵	0/2/4/6/8/10	Switching	Rocker 2 – Rocker 7 [1]	1 Bit	C, W, T
🟨↵	1/3/5/7/9/11	Status	Rocker 2 – Rocker 7 [1]	1 Bit	C, T



Function: Dimming (for all rockers) [4]

Object	Function	Name	Typ	Flag
🟨←| 0/2/4/6/8/10	Switching	Rocker 2 – Rocker 7 [1]	1 Bit	C, W, T
🟨←| 1/3/5/7/9/11	Status	Rocker 2 – Rocker 7 [1]	1 Bit	C, W
🟨 | 12/14/16/18/20/22	Dimming	Rocker 2 – Rocker 7 [1]	4 Bit	C, T

Function: Shutter/blind (for all rockers) [4]

Object	Function	Name	Typ	Flag
🟨 | 0/2/4/6/8/19	Short operation	Rocker 2 – Rocker 7 [1]	1 Bit	C, T
🟨←| 1/3/5/7/9/11	Status	Rocker 2 – Rocker 7 [1]	1 Bit	C, W
🟨 | 12/14/16/18/20/22	Long operation	Rocker 2 – Rocker 7 [1]	1 Bit	C, T

Function: Universal value transmitter (for all rockers) [4]

Object	Function	Name	Typ	Flag
🟨 | 0/2/4/6/8/10	Universal value transmitter	Rocker 2 – Rocker 7 [1]	1 Byte	C, T
🟨←| 1/3/5/7/9/11	Status	Rocker 2 – Rocker 7 [1]	1 Byte	C, W

Function: Room temperature controller operation (for all rockers) [4]

Object	Function	Name	Typ	Flag

No further objects for the push sensor functionality.

The following objects are provided for the inhibit function (push sensor functionality), the description field light or for the alarm function, respectively:

Function: "1-bit data format" alarm signal

Object	Function	Name	Typ	Flag
□| 24	Switching	Alarm signal	1 Bit	C, T [5]

Function: "1-byte data format" alarm signal

Object	Function	Name	Typ	Flag
□| 24	Value	Alarm signal	1 Byte	C, T [5]

Function: Inhibit function

Object	Function	Name	Typ	Flag
□←| 25	Disabling function	Disabling push-button sensor	1 Bit	C, W

Function: Switching the Display illumination

Object	Function	Name	Typ	Flag
□←| 26	Switching	Display illumination	1 Bit	K, S

Function: Switching the Operation LED

Object	Function	Name	Typ	Flag
□←| 27	Switching	Operation LED	1 Bit	K, S

Room temperature controller functions:

Function: Actual temperature

Object	Function	Name	Typ	Flag
□→| 28	Actual temperature	Measured / matched value	2 Byte	C, R, T

Function: Additional temperature sensor

Object	Function	Name	Typ	Flag
□←| 29	External temperature sensor	Temperature value	2 Byte	C, W, T
□←| 30	External temperature sensor (outside sensor)	Temperature value	2 Byte	C, W

Function: Basic setpoint selection

Object	Function	Name	Typ	Flag
□←| 31	Basic set value	Temperature default value	2 Byte	C, W
□←| 32	Basic set value 2nd control loop[6]	Temperature default value	2 Byte	C, W

Function: Operating mode switch-over

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

Object	Function	Name	Typ	Flag
□←| 33	Operating mode switch-over	KONNEX switch-over	1 Byte	C, W, (T)[7]
□←| 37	Override-object operating mode	KONNEX switch-over	1 Byte	C, W

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

Object	Function	Name	Typ	Flag
□←| 33	Comfort mode	Operating mode switch-over	1 Bit	C, W, (T)[7]
□←| 34	Standby mode	Operating mode switch-over	1 Bit	C, W, (T)[7]
□←| 35	Economy mode	Operating mode switch-over	1 Bit	C, W, (T)[7]
□←| 36	Frost/heat protection mode	Operating mode switch-over	1 Bit	C, W, (T)[7]

Presence object and window status:

Object	Function	Name	Typ	Flag
□← 38	Presence object	Presence-key / -detector	1 Bit	C, W, T
□← 39	Window status	Windowcontact	1 Bit	C, W

Function: Heating/cooling switch-over

Object	Function	Name	Typ	Flag
□← 40	Heating / cooling [8]	Heating / cooling switch-over	1 Bit	C, W, T

Function: Status information

Object	Function	Name	Typ	Flag
□ 41	Controller status	Statusindication general	1 Byte	C, T
□ 41	Controller status	Statusindication single	1 Bit	C, T
□ 42	Heating indication	Indication	1 Bit	C, T
□ 43	Cooling indication	Indication	1 Bit	C, T

Function: Disabling function (room temperature controller)

Object	Function	Name	Typ	Flag
□← 44	Disabling controller operation	Disabling function	1 Bit	C, W
□← 45	Disabling controller	Disabling function	1 Bit	C, W
□← 46	Disabling additional stage [9]	Disabling function	1 Bit	C, W
□← 47	Inhibit 2nd control loop [9]	Disabling function	1 Bit	C, W

Function: Heating variable

No additional stage activated/one control loop/

For mixed operation: Output of "heating" and "cooling" variables through separate objects:

Object	Function	Name	Typ	Flag
□← 47	Heating (control loop 1)	Continuous variable	1 Byte	C, W, T
□← 47	Heating (control loop 1)	PWM variable	1 Bit	C, W, T
□← 47	Heating (control loop 1)	Switching variable	1 Bit	C, W, T

No additional stage activated/one control loop/

For mixed operation: Output of "heating" and "cooling" variables through one common object:

Object	Function	Name	Typ	Flag
□← 47	Heating/cooling (1st control loop)	Continuous variable	1 Byte	C, W, T
□← 47	Heating/cooling (1st control loop)	PWM variable	1 Bit	C, W, T
□← 47	Heating/cooling (1st control loop)	Switching variable	1 Bit	C, W, T

Additional stage activated/one control loop/

For mixed operation: Output of "heating" and "cooling" variables through separate objects:

Object	Function	Name	Typ	Flag
□← 47	Basic-heating (1st control loop)	Continuous variable	1 Byte	C, W, T
□← 47	Basic-heating (1st control loop)	PWM variable	1 Bit	C, W, T
□← 47	Basic-heating (1st control loop)	Switching variable	1 Bit	C, W, T
□← 48	Additional-heating (1st control loop)	Continuous variable	1 Byte	C, W, T
□← 48	Additional-heating (1st control loop)	PWM variable	1 Bit	C, W, T
□← 48	Additional-heating (1st control loop)	Switching variable	1 Bit	C, W, T

Two control loops:

Object		Function	Name	Typ	Flag
□←	47	Heating (1st control loop)	Continuous variable	1 Byte	C, W, T
□←	47	Heating (1st control loop)	PWM variable	1 Bit	C, W, T
□←	47	Heating (1st control loop)	Switching variable	1 Bit	C, W, T
□←	51	Heating (2nd control loop)	Continuous variable	1 Byte	C, W, T
□←	51	Heating (2nd control loop)	PWM variable	1 Bit	C, W, T
□←	51	Heating (2nd control loop)	Switching variable	1 Bit	C, W, T

Additional stage activated/one control loop/

For mixed operation: Output of "heating" and "cooling" variables through one common object:

Object		Function	Name	Typ	Flag
□←	47	Basic-heating and -cooling (1st control loop)	Continuous variable	1 Byte	C, W, T
□←	47	Basic-heating and -cooling (1st control loop)	PWM variable	1 Bit	C, W, T
□←	47	Basic-heating and -cooling (1st control loop)	Switching variable	1 Bit	C, W, T
□←	48	Additional-heating and -cooling (1st control loop)	Continuous variable	1 Byte	C, W, T
□←	48	Additional-heating and -cooling (1st control loop)	PWM variable	1 Bit	C, W, T
□←	48	Additional-heating and -cooling (1st control loop)	Switching variable	1 Bit	C, W, T

Function: Cooling variable

No additional stage activated/one control loop/

For mixed operation: Output of "heating" and "cooling" variables through separate objects:

Object		Function	Name	Typ	Flag
□←	49	Cooling (1st control loop)	Continuous variable	1 Byte	C, W, T
□←	49	Cooling (1st control loop)	PWM variable	1 Bit	C, W, T
□←	49	Cooling (1st control loop)	Switching variable	1 Bit	C, W, T

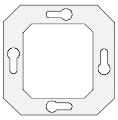
Additional stage activated/one control loop/

For mixed operation: Output of "heating" and "cooling" variables through separate objects:

Object		Function	Name	Typ	Flag
□←	49	Basic cooling (1st control loop)	Continuous variable	1 Byte	C, W, T
□←	49	Basic cooling (1st control loop)	PWM variable	1 Bit	C, W, T
□←	49	Basic cooling (1st control loop)	Switching variable	1 Bit	C, W, T
□←	50	Additional cooling (1st control loop)	Continuous variable	1 Byte	C, W, T
□←	50	Additional cooling (1st control loop)	PWM variable	1 Bit	C, W, T
□←	50	Additional cooling (1st control loop)	Switching variable	1 Bit	C, W, T

Two control loops:

Object		Function	Name	Typ	Flag
□←	49	Cooling (1st control loop)	Continuous variable	1 Byte	C, W, T
□←	49	Cooling (1st control loop)	PWM variable	1 Bit	C, W, T
□←	49	Cooling (1st control loop)	Switching variable	1 Bit	C, W, T
□←	50	Cooling (2nd control loop)	Continuous variable	1 Byte	C, W, T
□←	50	Cooling (2nd control loop)	PWM variable	1 Bit	C, W, T
□←	50	Cooling (2nd control loop)	Switching variable	1 Bit	C, W, T



Function: Heating status information variable [10]

Object		Function	Name	Typ	Flag
□↤	51	Heating (1st control loop)	PWM variable	1 Byte	C, W, T
□↤	51	Basic-heating (1st control loop)	PWM variable	1 Byte	C, W, T
□↤	52	Additional-heating (1st control loop)	PWM variable	1 Byte	C, W, T

Function: Cooling status information variable [10]

Object		Function	Name	Typ	Flag
□↤	53	Cooling (1st control loop)	PWM variable	1 Byte	C, W, T
□↤	53	Basic-cooling (1st control loop)	PWM variable	1 Byte	C, W, T
□↤	54	Additional-cooling (1st control loop)	PWM variable	1 Byte	C, W, T

Function: Setpoint temperature

Object		Function	Name	Typ	Flag
□→	55	Setpoint temperature	Temperature value	2 Byte	C, T, R
□→	56	Setpoint temperature 2nd control loop [11]	Temperature value	2 Byte	C, T, R

Function: Time signal

Object		Function	Name	Typ	Flag
□↤	57	Time signal	Time	3 Byte	C, W

Function: Heating timer

Object		Function	Name	Typ	Flag
□↤	62	Disabling heating timer	Disabling function	1 Bit	C, W

Control functions

Function: Switching (for both control functions [12]):

Object		Function	Name	Typ	Flag
□ |	58	Control function 1	Switching	1 Bit	C, T
□ |	60	Control function 2	Switching	1 Bit	C, T

Function: Value (for both control functions [12]):

Object		Function	Name	Typ	Flag
□ |	58	Control function 1	Value	1 Byte	C, T
□ |	60	Control function 2	Value	1 Byte	C, T

Inhibit functions of the control functions:

Object		Function	Name	Typ	Flag
□↤	59	Disabling control function 1	Disabling function	1 Bit	C, W
□↤	60	Disabling control function 2	Disabling function	1 Bit	C, W

Scene function

Function: Switching (for all 8 scene objects [13])

Object		Function	Name	Typ	Flag
□	63	Scene output 1	Switching	1 Bit	C, T
□	64	Scene output 2	Switching	1 Bit	C, T
□	65	Scene output 3	Switching	1 Bit	C, T
□	66	Scene output 4	Switching	1 Bit	C, T
□	67	Scene output 5	Switching	1 Bit	C, T
□	68	Scene output 6	Switching	1 Bit	C, T
□	69	Scene output 7	Switching	1 Bit	C, T
□	70	Scene output 8	Switching	1 Bit	C, T

Function: Value (for all 8 scene objects [13])

Object		Function	Name	Typ	Flag
□	63	Scene output 1	Value	1 Byte	C, T
□	64	Scene output 2	Value	1 Byte	C, T
□	65	Scene output 3	Value	1 Byte	C, T
□	66	Scene output 4	Value	1 Byte	C, T
□	67	Scene output 5	Value	1 Byte	C, T
□	68	Scene output 6	Value	1 Byte	C, T
□	69	Scene output 7	Value	1 Byte	C, T
□	70	Scene output 8	Value	1 Byte	C, T

Function: Shutter/blind position (for all 8 scene objects [13])

Object		Function	Name	Typ	Flag
□	63	Scene output 1	Shutter/blind position	1 Bit	C, T
□	64	Scene output 2	Shutter/blind position	1 Bit	C, T
□	65	Scene output 3	Shutter/blind position	1 Bit	C, T
□	66	Scene output 4	Shutter/blind position	1 Bit	C, T
□	67	Scene output 5	Shutter/blind position	1 Bit	C, T
□	68	Scene output 6	Shutter/blind position	1 Bit	C, T
□	69	Scene output 7	Shutter/blind position	1 Bit	C, T
□	70	Scene output 8	Shutter/blind position	1 Bit	C, T

Scene extension:

Object		Function	Name	Typ	Flag
□←	71	Scene extension:	Extension-input	1 Byte	C, W

Legende

1: The number of the available keys or rockers depends on the projected push sensor variant (2fold, 3fold or 6fold)

2: The switching, dimming, shutter/blind, value transmitter, analog value transmitter, light scene extension/recall, room temperature controller operation, heating timer operation and control function operation functions can be selected per key. The names of the communication objects and the object table (dynamic object structure) will change correspondingly. You can also combine key or rocker functions, respectively.

3: The light scene extension object will not be visible when you have set "function as = internal light scene recall".

4: The switching, dimming, shutter/blind, universal value transmitter and room temperature controller operation functions can be selected per rocker. The names of the communication objects and the object table (dynamic object structure) will change correspondingly. You can also combine key or rocker functions, respectively.

5: You can only assign a group address to the "alarm signal" object.

6: This object will only be active if you have activated control loop 2 and if both loops have separate setpoints.

7: As an option, you can set the "T"-flags for the operating mode switch-over objects. After you have set the flags, the object values changed in accordance with the reset operating mode will be actively transmitted to the bus.

8: This object will only be active for the mixed "heating and cooling" or "basic/additional heating/cooling" mode if one control loop is used.

9: This object will only be visible when you have activated the additional stage or, alternatively, if you use two control loops.

10: The status information on the PWM variable is only possible for one control loop.

11: This object will only be active if you have activated control loop 2 and if both loops have separate setpoints.

12: You can set the "switching" and "value" functions for each control function.

13: You can set the "switching", "value" and "shutter/blind" functions for each scene object.

Contents

1. General Functions:

1.1 Enabling the push sensor, scene, control and room temperature controller functions

In the push sensor 2 plus, the push sensor, scene, control and room temperature controller functions must be considered as separate items. These individual components can be optionally activated, if necessary. For this purpose, set the "push-button sensor function", "scene function", "control function 1/2" or "function of room temperature controller" parameters in the "push sensor 2 plus" parameter branch to "switched on". Only in this case, the parameters and objects assigned to these functions will be activated and can be changed.
Please note that the parameterizable extent of the rockers or keys of the push sensor will be automatically adapted by the ETS plug-in if you deactivate individual functions.
Therefore, it will, for example, not be possible for you to parameterize switching, dimming, shutter/blind or value transmitter functions if you have deactivated the push sensor function. On the other hand, if the room temperature controller function has been deactivated, you cannot set any room temperature control for the rockers or keys.

1.2 Alarm function

When you unplug the push sensor 2 plus from the bus coupling unit, an ON or OFF telegram or a value telegram can be transmitted via the "alarm signal" object. Alternatively, such telegram triggering can be suppressed by the "alarm function after pulling off the user module = "disabled" (default) ETS parameter setting.

a) "Reset value" = "No" parameter (default)

When you unplug the push sensor from the bus coupling unit, an alarm telegram with an alarm value corresponding to the parameterization (switching value "1" or "0" for a switching telegram or "0…255" for a value telegram) will be sent.

After having been re-plugged, the push sensor will be ready for operation again after its initialization phase (display "Init"). In this connection, the value of the alarm object will always be reset to "0" (for the 1-bit switching value and for the 1-byte value) but not actively transmitted to the bus. You need not externally reset the alarm telegram. External write access to the alarm object will overwrite the object value. However, the latter will be overwritten by the parameterized or default values when you unplug or re-plug the push sensor.

In case of a bus voltage failure, an alarm message transmitted before will be permanently saved. An alarm message saved will be transmitted again upon bus voltage recovery if the push sensor is not plugged on while the bus voltage is reappearing.

b) "Reset value" = "Yes" parameter

When you unplug the push sensor from the bus coupling unit, an alarm telegram with an alarm value corresponding to the parameterization (switching value "1" or "0" for a switching telegram or "1…255" for a value telegram) will be sent.

After having been re-plugged, the push sensor will be ready for operation again after its initialization phase (display "Init"). During the initialization phase, the value of the alarm object will be reset to the inverted object value for the 1-bit switching value or to the "0" value for the 1-byte value and actively transmitted to the bus. You need not externally reset the alarm telegram. External write access to the alarm object will overwrite the object value. However, the latter will be overwritten by the parameterized or default values when you unplug or re-plug the push sensor.

In case of a bus voltage failure, an alarm message transmitted before will be permanently saved. An alarm message saved will be transmitted again upon bus voltage recovery if the push sensor has not been plugged on. If the push sensor is plugged on while the bus voltage is reappearing the alarm will be reset by the sending of the inverted object value for the 1-bit switching value or of the "0" value for the 1-byte value.

Note: The "alarm signal" object can only be linked to a group address. This object can only be read out with the push sensor plugged on (set the "R"-flag).

1.3 Lengh of operation display illumination

For all push sensor functions associated with key operations, you can parameterize the status-LED of a key as actuation indicator. Only in this case, the LEDs will be lit for the period set by the "lengh of operation display illumination" parameter in the "push sensor 2 plus" parameter branch if the keys are actuated. You will have the option to parameterize 1 s, 2 s or 3 s (default).

1.4 Display illumination and operation LED

The push sensor 2 plus has a common background light for the display unit. You can use the "display illumination" parameter in the "push sensor 2 plus" parameter branch to preset the function of this light.
You can permanently switch off the light (setting: "OFF"), permanently switch it on (setting: "ON"), or automatically switch it off (setting: "automatic switch-off"/default). If you select automatic switch-off the light will switch on if you press any key and will then automatically go out after the time you have set by the "automatic illumination switch-off, base/factor" parameters. In this connection, you can preset switch-on times between approx. 0.5 s and approx. 21 min./15 s.

Alternatively, the light can be switched via the "display illumination" object. For this purpose, set the "display illumination" parameter to "switching via object …" which will enable the object. You can preset the polarity of the object:

- For "…object = 1 : ON", the light will be switched on at an object value = "1", and switched off at "0".
- For "…object = 1 : OFF", the light will be switched on at an object value = "0", and switched off at "1".

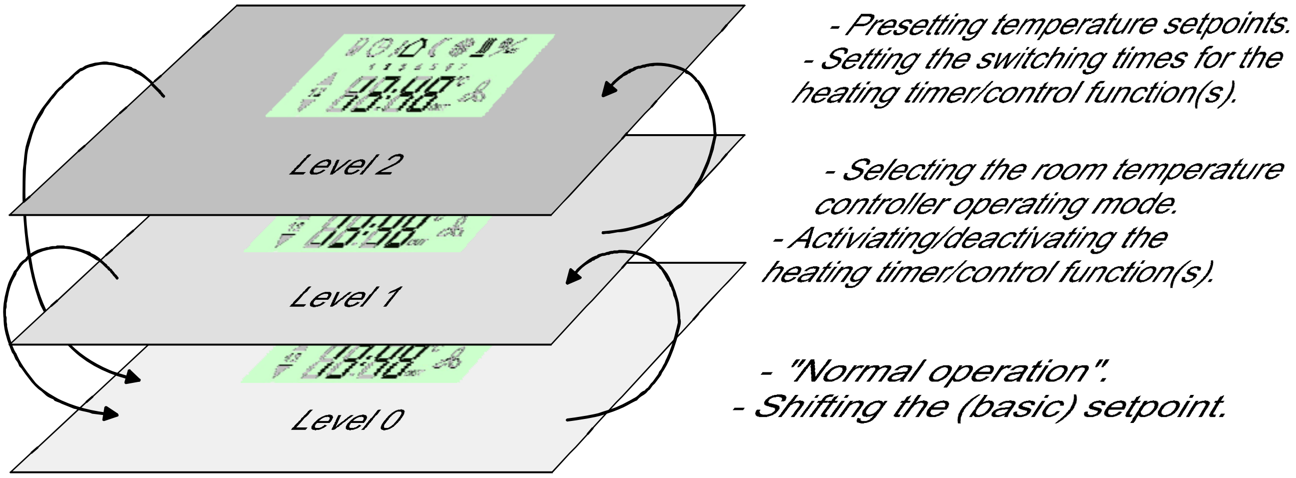
The behaviour of the green operation LED can additionally be defined with the "Operation LED" parameter. The settings and functions available are the same as those for the display illumination.

The operation LED goes out automatically when the status LED above in the last pushbutton lights up.

After bus voltage recovery, the "description field light" object value will always be "0".

1.5 Operator levels / local operation

The push sensor 2 plus has up to three different operator levels. By local operation, you can switch over these operator levels on the push sensor to facilitate the activation of different functions or the selection of various setpoints. You can actuate rocker 1 to switch over the operator levels and to navigate through the menus activated thereby (refer to "3.1 Rocker arrangement").



- Presetting temperature setpoints.
- Setting the switching times for the heating timer/control function(s).

- Selecting the room temperature controller operating mode.
- Activiating/deactivating the heating timer/control function(s).

- "Normal operation".
- Shifting the (basic) setpoint.

The "access to operator levels" parameter in the "push sensor 2 plus" parameter branch specifies which levels you can activate by local operation.

• "No operation":

Local operation of the controller by actuation rocker 1 is not possible. Rocker 1 is disabled. Only normal operation without shifting the basic setpoint is possible.

• "First operator level":

Switch-over to level 1 is possible. Level 0 (setpoint shift/normal operation) and level 1 (switch-over of the operating mode and activation/deactivation of the heating timer and the control functions as well as "contrast adjustment") can thus be activated.

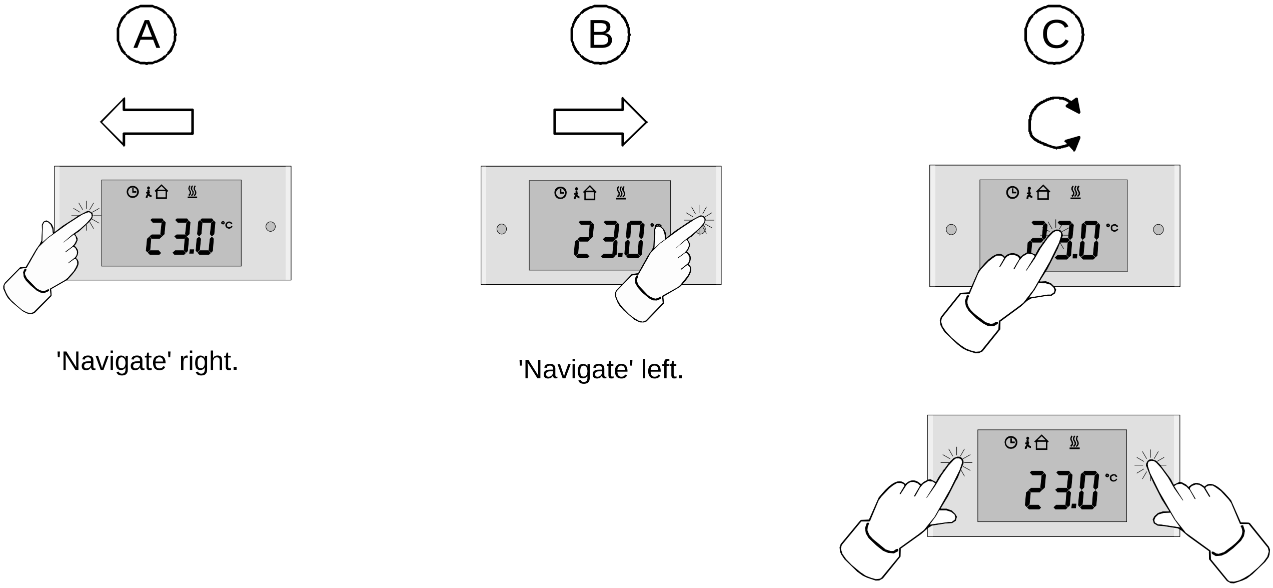
• "All operator levels":

Full access to the device with local operation of the controller. It additionally grants the user access to level 2, i. e. to the "setpoint assignment" function (if enabled in the ETS plug-in under "setpoints") and to the setting of the heating timer switching times as well as to the control functions.

Moreover, the operation of the controller (rocker 1) can be disabled. From the ETS plug-in, you can parameterize whether disabling of the controller operation shall always take place or shall be object-controlled (refer to "4.6.1 Disabling controller operation).

Ⓐ Ⓑ The menu items selectable in an operator level are indicated in the display by blinking symbols or values, respectively. You can actuate the left or right key of rocker 1 to change (navigate) between the menu functions.

Ⓒ You can actuate the middle of rocker 1 by at least 3 s or simultaneously press keys 1 and 2 of the same rocker for at least 3 s to switch over the operator levels. If you want to select individual menu items or accept settings you have made at an operator level actuate rocker 1 in the same way.



'Navigate' right.

'Navigate' left.

Switch over operator level or accept setting.

1.5.1 Level 0 / normal operation

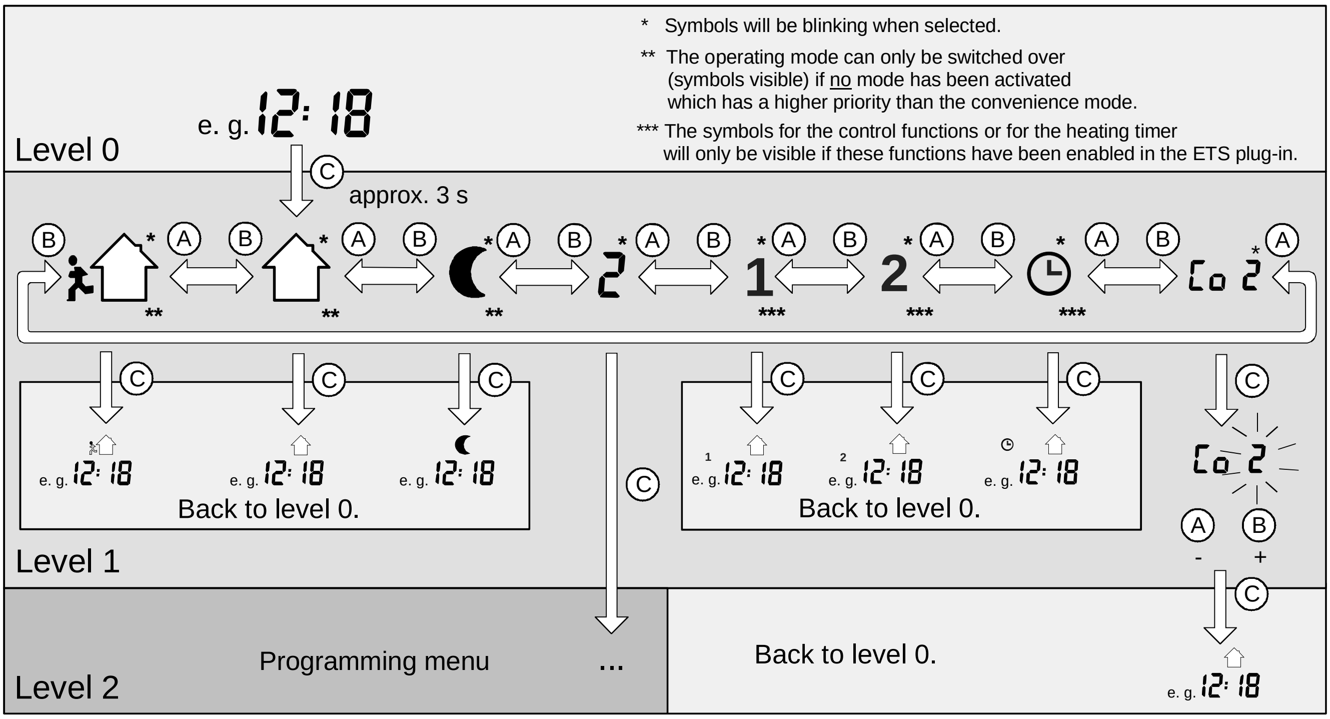
The device will be in the normal operation mode if you have activated operator level 0. Irrespective of what you have parameterized (refer to "2.2 Normal operation display data,"), the display window will show the current room temperature (default) and, in addition, or alternatively, the outside temperature, the setpoint temperature or the time (standard display).
If you actuate one of the keys of rocker 1 the display window will show the setpoint temperature of the activated operating mode, unless you are denied access to the operator levels. You can press the right or left key of the rocker to increment or decrement the setpoint temperature by 0.1 °C. You can set such setpoint shift (basic temperature offset) in any operating mode and, optionally, accept it when changing the mode (e. g. comfort → standby) so that this shift will act on all operating modes of the controller. For more information on the basic setpoint temperature assignment or shift, please refer to chapter "4. Room Temperature controller Functions".
If you have set a basic setpoint shift you can press any key of rockers 2 to 3 (2fold) or 2 to 5 (3fold or 6fold) to accept such value and return to the standard display. If you make no other entries within the next 2 min., the value you have set will also be accepted as the new setpoint, with the display being switched back.

Note: After bus voltage recovery, the controller will always be at operator level 0.

1.5.2 Operator level 1

At level 1, you can select the room temperature controller operating mode and activate or deactivate the heating timer or the control function(s). In addition, you can adjust the display contrast or change to operator level 2. For more information on how to set the operating modes and the heating timer or on their functions, please refer to chapter "4. Room Temperature Controller Functions". The control functions are described in chapter "5. Control Functions".

You can activate level 1 from level 0.



* Symbols will be blinking when selected.

** The operating mode can only be switched over (symbols visible) if <u>no</u> mode has been activated which has a higher priority than the convenience mode.

*** The symbols for the control functions or for the heating timer will only be visible if these functions have been enabled in the ETS plug-in.

If you have set the "access to operator levels" to "first operator level" or "all operator levels" you can actuate the middle of rocker 1 (ⓒ) for at least 3 s to change to operator level 1. At this position, you can select the controller operating mode and activate or deactivate the heating timer or the control function(s). You can also adjust the display contrast.
Use the left or right key (Ⓐ / Ⓑ) of rocker 1 to navigate, i. e. to switch among the menu functions. The symbol you select will blink in the display window. All the other display elements will be deactivated.
You can choose from the "comfort ⌂", "standby ⌂" and "economy ☾" modes, unless <u>no</u> higher-order mode (e. g. window contact/presence detector) or <u>no</u> KONNEX constraint object has been activated. Symbols **"1"** and **"2"** identify the two control functions, and the "◷" symbol stands for the heating timer. To activate the desired operating mode or the heating timer or the control function(s) select the corresponding symbol and actuate the middle of rocker 1 for a long time (approx. 1 s) (ⓒ). This will accep the operating mode, activate or deactivate the heating timer or the control function(s), and the display will return to standard (level 0).

If you select symbol " 2 " you can actuate the middle of rocker 1 (ⓒ) for at least 3 s to change to operator level 2.

Adjusting the display contrast:

To adjust the display contrast select corresponding symbol "Co 2" and actuate the middle of rocker 1 for a long time (approx. 1 s) (Ⓒ). Press the right or left key of rocker 1 (Ⓐ / Ⓑ) to shift the desired contrast stage up or down.
You can choose from contrast stages 0 (strong) and 5 (weak) (default: stage 2). The contrast will immediately response to what you select so that you can directly compare the change with the ambient light.
After you have set the desired stage press the middle of rocker 1 for a long time (approx. 1 s) to accept this setting and return to operator level 0.
After bus voltage recovery, or if the user module was unplugged and then re-plugged onto the bus coupling device, the contrast stage will be reset to its default value (stage 2).

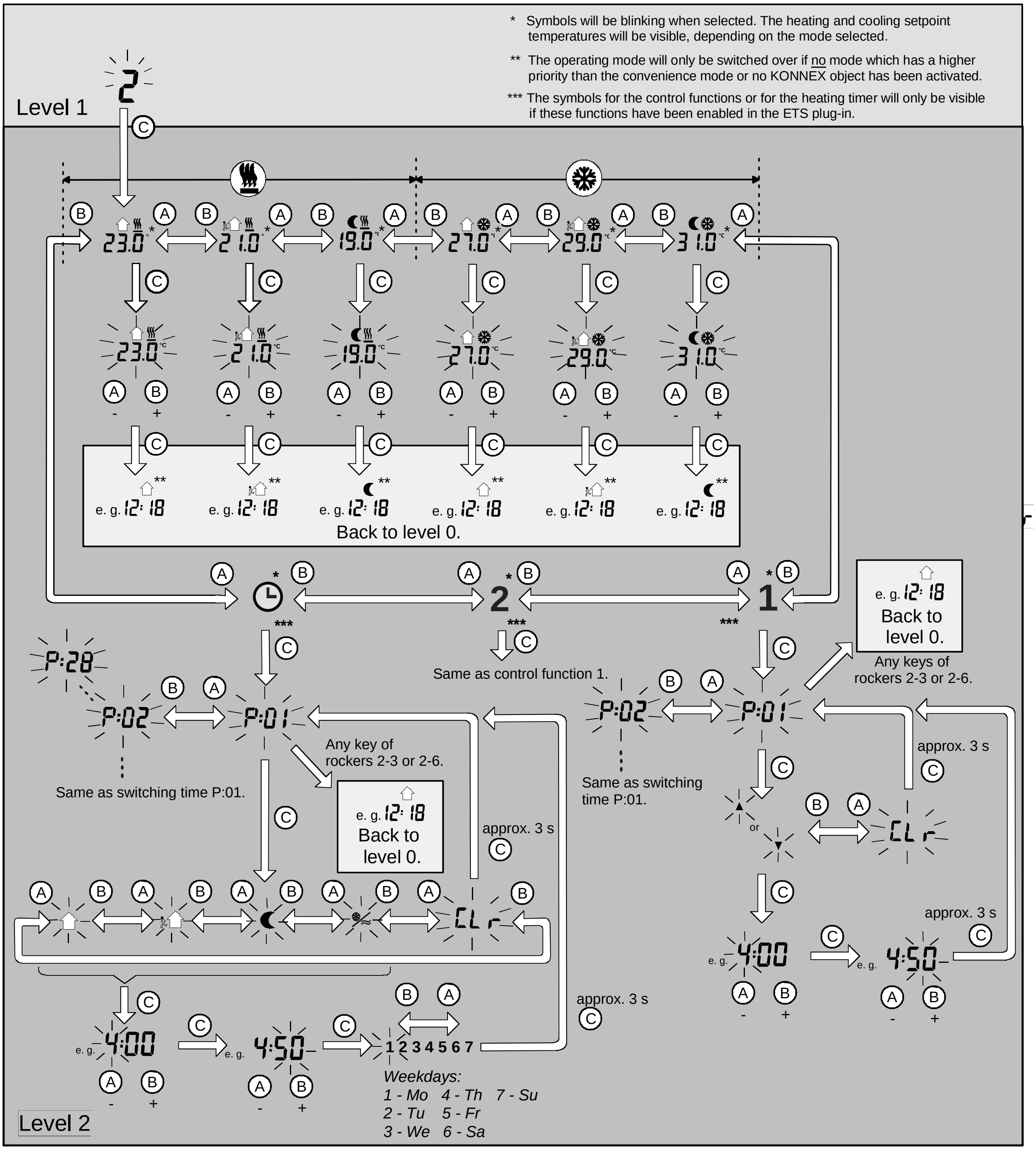
If you have selected operator level 1 and you make no more entries for the next 2 min. the display will automatically return to level 0. If any key of rockers 2 to 3 (dual type) or 2 to 5 (quintuple type) is actuated the display will also return to operator level 0 without changing the operating mode or changing to operator level 2.

Note: You can use the "operating mode after reset" parameter in the "room temperature controller function/functionality" parameter branch to preselect the operating mode to be activated after bus voltage recovery. After bus voltage recovery, operator level 0 will always be activated.

1.5.3 Operator level 2

At level 2, you can program the temperature setpoints of the different operating modes for control loop 1 of the room temperature controller and set the switching times of the heating timer and of the control function(s). For more information on how to assign the setpoints and on the controller behaviour, please refer to chapter "4. Room Temperature Controller Functions".

You can activate level 2 from level 1.


* Symbols will be blinking when selected. The heating and cooling setpoint temperatures will be visible, depending on the mode selected.
** The operating mode will only be switched over if no mode which has a higher priority than the convenience mode or no KONNEX object has been activated.
*** The symbols for the control functions or for the heating timer will only be visible if these functions have been enabled in the ETS plug-in.
Level 1
2
C
A
B
23.0
21.0
19.0
27.0
29.0
31.0
+
-
e. g. 12:18
Back to level 0.
2
1
Same as control function 1.
Back to level 0.
Any keys of rockers 2-3 or 2-6.
P:28
P:02
P:01
Any key of rockers 2-3 or 2-6.
Same as switching time P:01.
e. g. 12:18
Back to level 0.
approx. 3 s
CLr
or
e. g. 4:00
e. g. 4:50
1 2 3 4 5 6 7
Weekdays:
1 - Mo 4 - Th 7 - Su
2 - Tu 5 - Fr
3 - We 6 - Sa
Level 2

If you have set the "access to all operator levels" parameter to "all operator levels" you can actuate the middle of rocker 1 (Ⓒ) to change to operator level 2 after level 1 was activated and menu item " 2 " selected before.
At this position, you can now view and change the temperature setpoints of the different operating modes for control loop 1 and the switching times of the heating timer and of the control function(s). The switching times will only be visible if the heating timer or the control function(s) has (have) been enabled in the ETS plug-in.

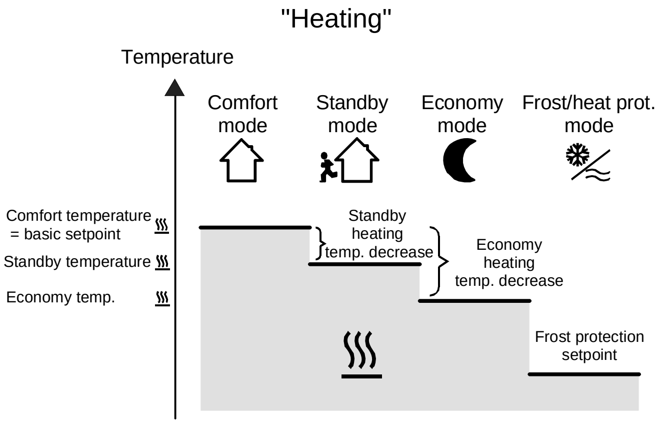
Use the left or right key of rocker 1 (Ⓐ / Ⓑ) to switch among the temperature setpoints and the switching times. The respective symbols you select will blink in the display window. All the other display elements will be deactivated.

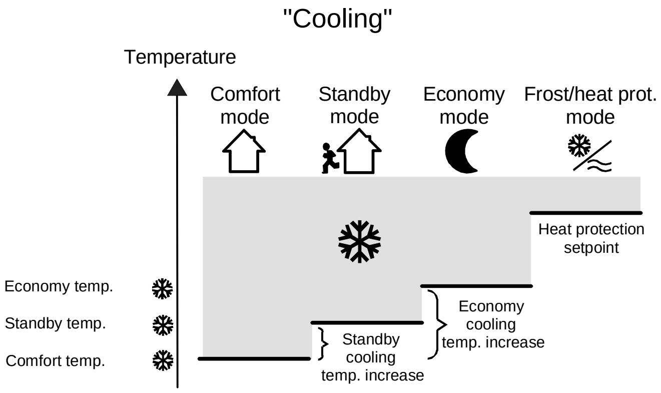
1.5.3.1 Assigning temperature setpoints

You can assign setpoints for the
- "comfort ⌂",
- "standby ⌂" and
- "economy ☾" modes.

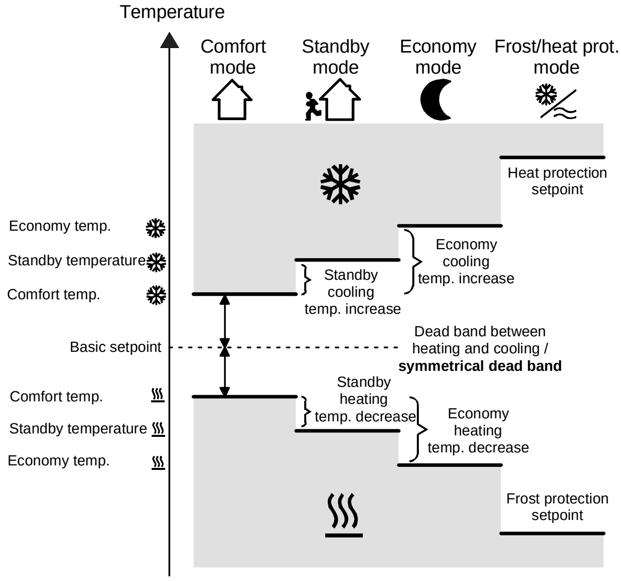
For the assignment of setpoints, up to six different values will be offered, depending on the operating mode enabled in the ETS plug-in. Please note that the individual setpoints have not been enabled in the ETS plug-in for local operation so that you can only view them in the display window without being able to change them (refer to "4.4. Temperature setpoints").
Furthermore, if the second control loop has its own setpoints, you can only the set the temperature values of the first control loop at operator level 2.
The following table shows the values you can set:

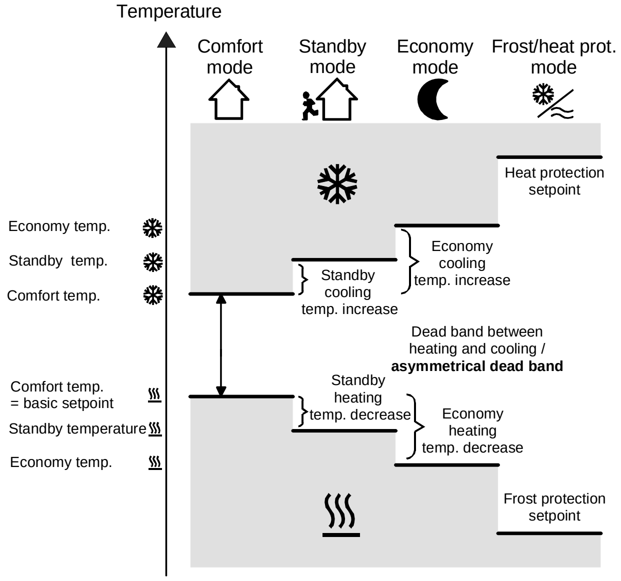
Activated Operating Mode	Parameterized Operating Mode			
			Heating and Cooling	
	Heating	Cooling	For Heating	For Cooling
Comfort ⌂	⌂ ♨ e. g. 23.0 °C Comfort setpoint temperature = basic setpoint	⌂ ❄ e. g. 27.0 °C Comfort setpoint temperature = basic setpoint	⌂ ♨ e. g. 23.0 °C Comfort setpoint temperature **= basic setpoint** **- ½ dead band** **for symm. dead band /** = basic setpoint for asymm. dead band	⌂ ❄ e. g. 27.0 °C Comfort setpoint temperature **= basic setpoint** **+ ½ dead band** **for symm. dead band /** = basic setpoint for asymm. dead band
Standby ⌂	⌂ ♨ e. g. 21.0 °C Standby setpoint temperature	⌂ ❄ e. g. 29.0 °C Standby setpoint temperature	⌂ ♨ e. g. 21.0 °C Standby setpoint temperature	⌂ ❄ e. g. 29.0 °C Standby setpoint temperature
Economy ☾	☾ ♨ e. g. 19.0 °C Economy setpoint temperature	☾ ❄ e. g. 31.0 °C Economy setpoint temperature	☾ ♨ e. g. 19.0 °C Economy setpoint temperature	☾ ❄ e. g. 31.0 °C Economy setpoint temperature


"Heating"
Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Comfort temperature = basic setpoint
Standby temperature
Economy temp.
Standby heating temp. decrease
Economy heating temp. decrease
Frost protection setpoint

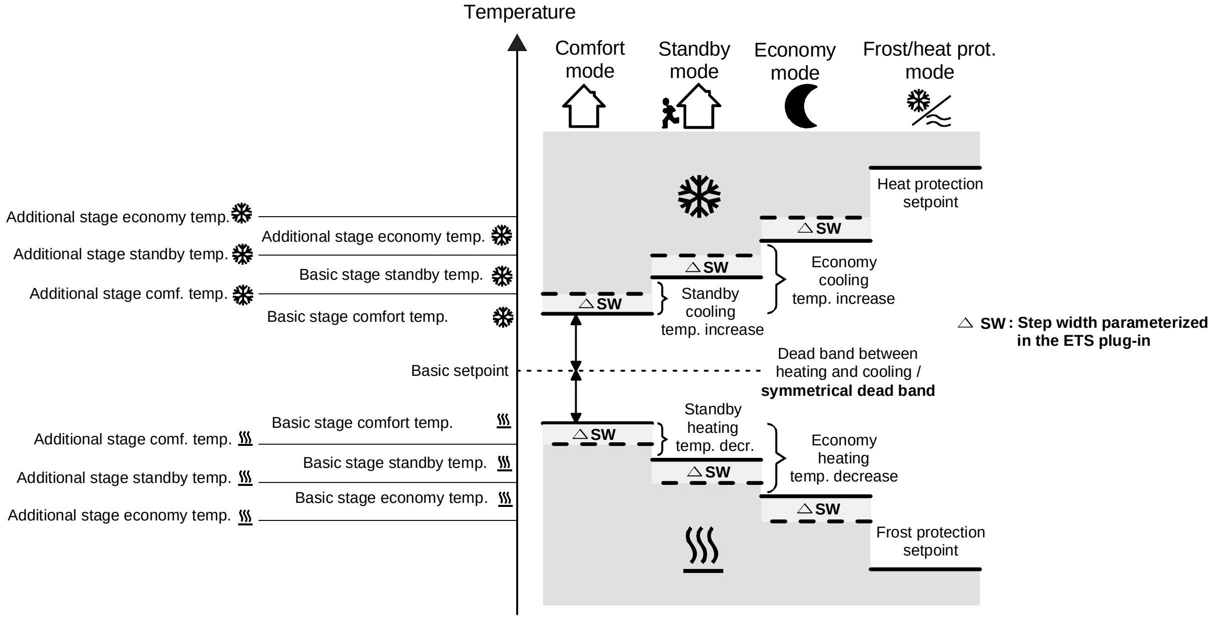


"Heating and Cooling"




Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Heat protection setpoint
Economy temp.
Standby temp.
Comfort temp.
Economy cooling temp. increase
Standby cooling temp. increase
Dead band between heating and cooling /
asymmetrical dead band
Comfort temp. = basic setpoint
Standby temperature
Economy temp.
Standby heating temp. decrease
Economy heating temp. decrease
Frost protection setpoint

Temperature Control with Additional Stage by the "Heating and Cooling" Example with Symmetrical Dead Band...


Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Heat protection setpoint
Additional stage economy temp.
Additional stage standby temp.
Additional stage comf. temp.
Additional stage economy temp.
Basic stage standby temp.
Basic stage comfort temp.
SW
Economy cooling temp. increase
Standby cooling temp. increase
SW : Step width parameterized in the ETS plug-in
Basic setpoint
Dead band between heating and cooling /
symmetrical dead band
Basic stage comfort temp.
Additional stage comf. temp.
Basic stage standby temp.
Additional stage standby temp.
Basic stage economy temp.
Additional stage economy temp.
Standby heating temp. decr.
Economy heating temp. decrease
Frost protection setpoint

In the *"heating and cooling"* mode, you can change six different temperature setpoints if they have been enabled in the ETS. In dependence on the temperature decrease, temperature increase or on the dead band parameterized in the ETS, all temperature setpoints are derived from the basic setpoint temperature.
If you change the comfort setpoint temperature please note in this connection that all the other setpoint temperatures values will also be changed.

The dead band (temperature zone where neither heating nor cooling takes place) is the difference between the comfort temperatures for "heating" and "cooling". Where:

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

Important notes:

- For a symmetrical dead band, the basic setpoint is indirectly set by the comfort heating temperature. The basic setpoint itself will not be read in the display for local operation.
- You can change the comfort cooling setpoint temperature to alter the dead band. When you alter the dead zone you can expect a shifting of the comfort heating setpoint temperature and thus of all the other temperature setpoints if the dead band position is symmetrical. For an asymmetrical dead band position, only the cooling temperature setpoints will be changed if you change the comfort cooling setpoint temperature. By local operation, you can shift the dead band to 0 °C ($T_{\text{comfort cooling setpoint}} = T_{\text{comfort heating setpoint}}$). In such case, neither heating nor cooling will take place if the room temperature determined is equal to the comfort setpoint temperatures.

The "standby" and "economy" setpoint temperatures are derived from the comfort heating or comfort cooling setpoint temperatures. In this connection, you can assign the temperature increase (for cooling) and the temperature decrease (for heating) of both operating modes in the ETS plug-in.
Through local operation at operator level 2, you can set the "standby" and "economy" setpoint temperatures independently of the values for the temperature increase or decrease originally parameterized in the ETS.
In such case, the "standby" or "economy" setpoint temperatures will always be shifted together with the temperature increase or decrease resulting from local operation if you change the basic setpoint temperature of the dead band. You can restore the originally parameterized values after reprogramming by the ETS.

Where:

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

or

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

For two-stage control, the setpoints for the additional stage are always derived dynamically from the setpoints of the basic stage. In this connection, the temperature setpoints of the additional stage are specified by the difference between the basic and additional stages parameterized in the ETS plug-in. Changing the difference between the basic and additional stages is not possible for local operation.

Change setpoints:

To be able to change setpoints select the corresponding symbol or value and actuate the middle of rocker 1 for a long time (approx. 1 s) at operator level 2 (Ⓒ). The symbol of the selected setpoint will no longer be blinking, and you can now actuate the left or right key of rocker 1 (Ⓐ / Ⓑ) to increment or decrement the setpoint by 0.1 °C. After you have set the desired setpoint, press the middle of rocker 1 for at least 3 s to accept this setting and return to operator level 0.
In this connection, the operating mode, whose setpoint you changed before, will be taken over as active mode. However, this will only happen if no higher-order mode (e. g. window contact/presence detector), or when no KONNEX constraint object has been activated.
If you want to set any further setpoints change to operator level 2 and proceed again as described.

For changing the temperature basic setpoint (e. g. by changing the comfort heating setpoint temperature at the second operator level), distinction must always be made between two different options:

- Option 1: The basic setpoint change will be accepred permanently.
- Option 2: The basic setpoint change will be accepted only temporarily (default).

In this connection, you can use the *"accept modification of the basic setpoint temperature permanently"* parameter in the *"room temperature controller function/setpoints"* parameter branch to determine whether you want to permanently (select *"yes"*) or only temporarily (select *"no"*) save the changed basic temperature value.

Option 1:

If you change the basic temperature setpoint it will be permanently saved in the EEPROM of the push sensor. In this connection, the newly set value will overwrite the basic temperature originally parameterized by the ETS.

Please take account of that:

- frequent changing of the basic temperature (e. g. several times a day) can adversely affect the life of the device as the non-volatile memory used has been designed for less frequent write access events only.
- you can alternatively assign this temperature via the bus through the "basic setpoint" object if it has been enabled in the ETS plug-in.

Thus, the basic setpoint selected on the push sensor or received through the object will remain stored even though the bus voltage may fail.

Option 2:

The basic setpoint selected on the push sensor or received through the object will remain stored in the currently set operating mode only in temporary form. In case of bus voltage failure or after you have changed the operating mode (e. g. from comfort to standby), the basic setpoint specified by local operation or received through the object will be discarded and replaced by the value originally parameterized in the ETS.

Important:

- As the setpoint temperatures for the "standby" and "economy" modes, or the setpoints for the "cooling" mode, respectively, are derived from the "heating" basic setpoint temperature, taking account of the decrease, increase or dead band values parameterized in the ETS plug-in or assigned by local operation, these setpoint temperatures will also linearly shift by the basic setpoint change made.
  The temperature setpoints for the standby or economy modes, or for the comfort "cooling" mode (dead band), respectively, will always be permanently saved in the EEPROM.
- Please note that you can only change or save any temperature setpoints by local operation or through the "basic set value" object, if the latter has been enabled in the ETS plug-in (refer to "4.4 Temperature setpoints"). Any value assigned by local operation will not be taken over into the object.

When you change a setpoint the operating mode, whose setpoint you changed before, will be taken over as active mode. However, this will only happen if no higher-order mode (e. g. window contact/presence detector), or no KONNEX constraint object has been activated.

Example 1:
1 - The comfort mode " ⌂ " is active as a result of a key actuation.
2 - Switch over to operator level 2.
3 - Change the setpoint for the economy mode " ☾ ".
4 - Accept the new setpoint - change over to normal (level 0).
5 - The economy mode " ☾ " will now be active.

Example 2:
1 - The presence detector is active (comfort mode " ⌂ ").
2 - Change to operator level 2.
3 - Change the setpoint for the economy mode " ☾ ".
4 - Accept the new setpoint - change over to normal (level 0).
5 - The comfort mode " ⌂ " will still be active.

You can only change the operating mode if the associated setpoint has been enabled in the ETS plug-in for local changing (refer to "4.4 Temperature setpoints").

If you have selected operator level 2 and you make no more entries for the next 2 min. level 0 will be automatically re-activated. If any key of rockers 2 to 3 (dual type) or 2 to 5 (quintuple type) is actuated the display will also return to operator level 0 without changing any setpoints.

Note: You can use the "operating mode after reset" parameter in the "room temperature controller function/ functionality" parameter branch to preselect the operating mode to be activated after bus voltage recovery. After bus voltage recovery, operator level 0 will always be activated.

1.5.3.2 Setting the heating timer switching times or the control functions

Depending on the parameterization in the ETS plug-in, up to two time-dependent control functions and a heating timer are available to the user. Once these functions have been enabled in the ETS, you can locally change the switching times at operator level 2.

Heating timer:

The heating timer can distinguish up to 28 different switching times and facilitates the switch-over of the room temperature controller operating mode exactly to the minute, depending on the time and the day of the week. The switching times will be executed in chronological order.

You can set the switching times for the heating timer after you have selected the " 🕒 " symbol at operator level 2 and pressed the middle of rocker 1 (Ⓒ) for a long time (approx. 1 s) to call the edit mode. You can then use the left or the right key of rocker 1 (Ⓐ / Ⓑ) to select the switching program (1 to 28) you want to edit.
After you have selected the corresponding program, you can actuate the left or right key of rocker 1 to set the desired operating mode. From there, you can select the "comfort ⌂", "standby ⌂", "economy ☾" or the "frost/heat protection ❄" mode.
If a switching time has already been saved in the selected program an additional symbol " CLr " will appear in the display. Select this symbol and actuate the middle of rocker 1 (Ⓒ) for a long time to delete such saved switching time. This will deactivate the selected program.

After having selected the operating mode and accepted it by long-time actuation of the middle of rocker 1 (Ⓒ), you can now assign the switching time. Set the hours first, and then the minutes. To edit the switching time actuate the left or right key of rocker 1 (Ⓐ / Ⓑ). Long-time actuation of the middle of this rocker (Ⓒ) will accept and save the switching time.
After this, you can define the days of the week on which you want this switching time to be activated. You can use the left or right key of rocker 1 (Ⓐ / Ⓑ) to change among the settings "on weekdays" (Mo - Fr), "weekend" (Sa - Su), "daily" (Mo - Su), or "user-defined" (Mo, Tu, ..., Su). The display of the push sensor shows the days of the week in alternative form by digits 0 to 7.
Long-time actuation of the middle of rocker 1 (Ⓒ) will save your settings and complete the programming procedure. After this, you can set or change further switching programs.
As an alternative, you can actuate any key (rockers 2-3 or 2-6, respectively) to cancel programming and return to operator level 0.

Control function(s):

Each of the control functions can distinguish between up to two switching times which are executed in chronological order and exactly to the minute. At operator level 2, you can only edit time-based control functions.

You can set the switching times for the control functions after you have selected the "1" symbol (for control function 1) or "2" (for control function 2) at operator level 2 and pressed the middle of rocker 1 (Ⓒ) for a long time (approx. 1 s) to call the edit mode. You can then use the left or the right key of rocker 1 (Ⓐ / Ⓑ) to select the switching program (1 to 2) you want to edit.
After you have selected the corresponding program, the symbol of the control command (" ▲ " / " ▼ ") assigned to the switching program will be blinking in the display. However, the control command can only be configured in the ETS plug-in. In the local operation edit mode, you can only view but not change it.
If a switching time has already been saved in the selected program an additional symbol " CLr " will appear in the display after you actuate the left or right key of rocker 1. Select this symbol and actuate the middle of rocker 1 (Ⓒ) for a long time to delete such saved switching time. This will deactivate the selected program.

After having actuated the middle of rocker 1 (Ⓒ) for a long-time, you can now assign the switching time. Set the hours first, and then the minutes. To edit the switching time actuate the left or right key of rocker 1 (Ⓐ / Ⓑ).
Long-time actuation of the middle of this rocker (Ⓒ) will accept and save the switching time.
Long-time actuation of the middle of rocker 1 (Ⓒ) will save your settings and complete the programming procedure. Then you can set or change the second switching program if you wish to do so. As an alternative, you can actuate any key (rockers 2-3 or 2-6, respectively) to cancel programming and return to operator level 0.

2. Display Unit

2.1 Basic function

Behind transparent rocker 1, the display unit (semi-graphical LCD) is located. On this unit, various functions of the integrated room temperature controller or also of the push sensor can be displayed.
In the *"display"* parameter branch, the display function can be activated by *the "display = on"* parameter (default). Alternatively, you can permanently deactivate the display unit by setting *"off"* so that it will no longer read anything. Irrespective of the display unit, rocker 1 will always be in operation.
As described in "1.5 Operator levels/local operation - adjusting the display contrast", you can alter the display contrast.

2.2 Display data in normal operation

In the middle of the display unit, temperature values such as the current room inside temperature, the current outside temperature, the current setpoint temperature, or the time can be shown. By the *"display of"* parameter in the *"display"* parameter branch in the ETS plug-in, you can define which of such information will be displayed.
In this connection, is also possible to show more than only one piece of display information (for example, parameter setting *"room/outside/clock"*). In such case, the display will change at a time interval you can specify by the *"switch-over time"* parameter.

2.2.1 Temperature display

The display can read the room temperature (actual temperature of the first control loop) determined by the controller and, additionally or alternatively, the outside temperature received via the bus and/or the current setpoint temperature of the first control loop.
The room temperature reading has a resolution of 0.1 °C and covers a range from -99.9 °C to +99.9 °C.
The reading will refresh as soon as the determined temperature changes within the resolution interval. The outside temperature reading has a resolution of 0.1 °C and covers a range from -99.9 °C to +99.9 °C. The reading will refresh as soon as a temperature-value telegram is received via the *"external temperature sensor (outside sensor)"* object (no. 25). If parameterized, the outside temperature will only be read in the display and cannot be used for any further temperature or variable calculation.
The setpoint temperature reading has a resolution of 0.1 °C. Its possible temperature range depends on the parameterized heating/cooling switch-over and is given by the fixed values for the frost and/or heat protection temperature. The reading will refresh as soon as a new setpoint temperature for the controller results (e. g. from a change of the operating mode or of the basic setpoint, etc.). When two control loops with separate setpoints are used only the setpoint of the first control loop will be displayed.

2.2.2 Time

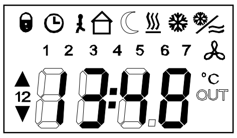
If parameterized, the time can be read in the middle of the display. In this connection, the time received via the *"time signal"* object, for example, transmitted by an EIB clock, will be displayed and the system clock implemented in the device initialized.
Once initialized, this clock will keep running internally, updating the display every minute. The "**:**" symbol between the hours and minutes will always be blinking at intervals of one second.
The current day of the week will be indicated by numbers (1-Mo to 7-Su) above the time. The information on the day of the week is derived from the time signal received.

The time signal should be transmitted at least every hour to keep the time error of the internal clock at a minimum. As long as no time signal has been received via the object, the display will read "--:--". The same reading will appear unless the internal clock has been updated via the bus at least once a day (updating check at 4:00 a. m.). However, the internal clock will keep running with the expected time error, with any possibly parameterized time-dependent control functions or the heating timer events still being executed.
Moreover, the internal time will get lost in case of a bus voltage failure so that the time signal should be transmitted after bus voltage recovery.

2.2.3 Symbols

Depending on the operating status of the device, the following symbols can be shown in the display in addition to the temperature or the time, respectively:

For example,



Symbol displayed	Display information at operator level 0	Display information at operator level 1	Display information at operator level 2
🔒	The operation of the controller or push sensor (incl. rocker 1), respectively, is disabled. If disabling is active this symbol will be blinking.		
⌂	"Comfort" mode active.	Activate "comfort" mode.	Comfort setpoint displayed./In the heating timer edit mode: "comfort" mode preselected.
🏃⌂	"Standby" mode active.	Activate "standby" mode.	Standby setpoint displayed./ In the heating timer edit mode: "standby" mode preselected.
☾	"Economy" mode active.	Activate "economy" mode.	Economy setpoint displayed./ In the heating timer edit mode: "economy" mode preselected.
❄≈	"Frost/heat protection" mode active.	---	In the heating timer edit mode: "Frost/heat protection" mode preselected.
♨	Room is being heated.	---	Display of heating setpoints.
❄	Room is being cooled.	---	Display of cooling setpoints.
°C	Display values for room, outside or setpoint temperature.	---	Display values for setpoint temperatures.
OUT	Outside temperature being displayed.	---	---
1...7	Display of day of the week Mo (1) - Su (7) together with time.	---	In the heating timer edit mode: programming the day of the week.
🕒	Heating timer active.	Activate/deactivate heating timer.	Heating timer edit mode active.
1	Control function 1 active.	Activate/deactivate control function 1.	Control function 1 edit mode active.
2	Control function 2 active.	Activate/deactivate control function 2.	Control function 2 edit mode active.
▲	Control command "0" or "UP", resp., being executed.	---	Executable command "0" or "UP", resp., of control function(s).
▼	Control command "1" or "DOWN", resp., being executed.	---	Executable command "1" or "DOWN", resp., of control function(s).
✇	This symbol has no function and is thus always deactivated.		

2.3 Data displayed during a programming process and initialization or in case of error

While the device is being programmed by the ETS, the display will show some status information.

When the ETS has built up a connection with the device via the bus, the display will read " PArA ". At this status, the device is being programmed with the project data. During this procedure, all push sensor and room temperature controller functions will be deactivated.

You can start a full program download, for example, when updating the firmware or in the event of an error (previously aborted programming process). For this purpose, select item *"with next download: transmit all"* under *"settings - options - hardware"* in the ETS plug-in. In this connection, the firmware matching the device will be automatically loaded and message " LoAd " shown in the display while programming is taking place. In parallel, the status-LEDs of rocker 2 will be blinking cyclically during this download variant. A firmware download can take several minutes.

After one programming process has been successfully completed, or after the push sensor has been re-plugged onto the bus coupling unit, the display will show for a short time the push sensor variant and the firmware version loaded.
For example, reading " 6F: 1.1 " indicates a 6fold push sensor 2 plus and a loaded firmware version 1.1.

After this, the device will initialize. At this status, the display will read " In It ". Then the device (both the push sensor and the room temperature controller) will be ready for operation.

As the bus coupling unit and the push sensor 2 plus form one unit and cannot be changed arbitrarily after having been programmed, the push sensor will check after bus voltage recovery or after having been re-plugged onto bus coupling unit whether the parameterizing information in the bus coupling unit agree with those in the memory of the push sensor.
If the parameterizations do not match as the unit was not previously placed into operation together, or as the push sensor or the bus coupling unit has been changed, the display of the push sensor 2 plus will read " noPA " (no parameter) to indicate that it has no valid parameters. In such case, the push sensor will not respond to any key actuation.

If the display reads " Err " the push sensor 2 plus is inoperable. In this case, the push sensor will not show any response and must be replaced.

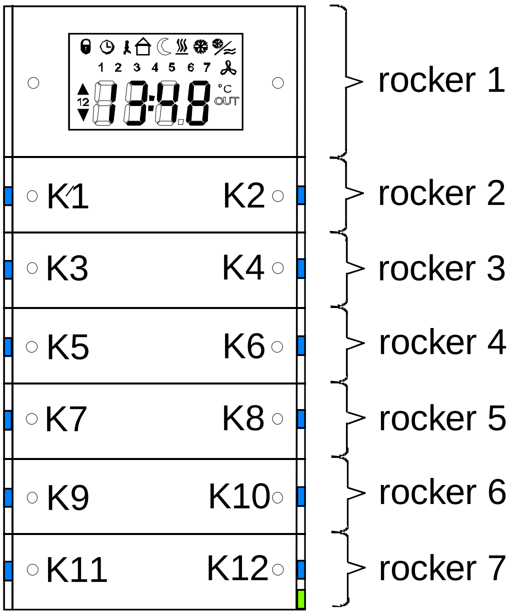
If the display reads no information and the status-LEDs of rocker 2 are blinking cyclically The device contains firmware which is invalid or does not run. Such condition can, for example, be caused if a previously made firmware download was too erroneous or was interrupted for any other reasons. In such case, the push sensor 2 plus will show no further response.
The remedy recommended for such erroneous situation is to perform a fresh firmware download (refer to "Remarks on the Software - Firmware" at the end of this documentation).

3. Push sensor Functions

3.1 Rocker arrangements

Depending on the variant projected, the pushbutton sensor 2 plus has 3, 4 or 7 rockers. The two display keys (rocker 1) cannot be parameterized as they programmed with predefined and fixed functions. The other rockers (function keys) can be parameterized without restrictions.

For example 6fold



Depending on the application (2fold, 3fold or 6fold) inserted into the ETS project, the required number of rockers or keys, respectively, will be created automatically in the ETS plug-in.
To make the configuration of the push sensor functions more clearly arranged and more efficient a preview window is optionally available in the ETS plug-in. This window can be activated from the *"preview"* menu item in the *"configuration"* menu. If the window is enabled you can mouse-click on one of rockers or keys, to go to the corresponding parameter branch and thus parameterize the selected rocker or key, respectively.
Within the corresponding parameter branch, you can assign a name to a rocker or a key. Such name just serves for better orientation in the plug-in and will neither be displayed in the preview window nor downloaded into the device.

Behind transparent rocker 1 there is the display unit (semi-graphical LCD) through which different statuses of the room temperature controller or of the push sensor are visualized (refer to "2. Display Unit").

3.2 Rocker functions

In the push sensor 2 plus, the push sensor, room temperature controller, scene and control functions must be considered as separate items.
For the function as push sensor, 2 rockers for the dual-type push sensor 2 plus, or 5 rockers for the quintuple-type push sensor 2 plus, which various functions can have been assigned to, are separately available.

Rocker 1 of the push sensor is always reserved to local operation of the integrated room temperature controller. The exact functionality of this rocker is discussed in more detail in the operator level switch-over description (refer to "1.5 Operator levels"). Rocker 1 has no status-LED.

On the other hand, rockers 2-3 or 2-5 - parameterizable in the ETS plug-in - can be given the following push sensor functions. In this connection, distinction is made between rocker and key actuation.

Function	Rocker Actuation	Key Actuation
No function	✓	✓
Switching/keying	---	✓
Switching	✓	---
Dimming	✓	✓
Shutter/blind	✓	✓
Universal value transmitter	✓	---
Analog value transmitter	---	✓
Light scene extension/recall	---	✓
Value transmitter	---	✓
Operation of room temperature controller *	✓	✓
Heating timer operation	---	✓
Control function operation	---	✓

*: The "operation of room temperature controller" function is a room temperature controller feature. The exact functionality of a rocker parameterized to this function is discussed in more detail in the description of the room temperature controller functions (refer to "4.1.1. Changing the operating modes").

In general, the push sensor function can be enabled. For this purpose, select option *"switched on"* under the *"push-button sensor function"* parameter in the *"push sensor 2 plus"* parameter branch in the ETS plug-in. If the push sensor function is *"switched off"* the selection of the push sensor functions will be matched so that only the functions of the enabled function elements ("operation of room temperature controller*", "heating timer operation", "control function operation" and/or "light scene extension recall"*) will be parameterizable in this case.

Through the *"operation"* parameter in the *"push-button sensor functions"* parameter branch, select the type of rocker actuation. Separately for each rocker, you can parameterize rocker actuation or key actuation. For rocker actuation, the left and right keys of a rocker form a pair of keys to which you can assign a joint function. For key actuation, the left and right keys of a rocker must be considered to be separate of each other so that two functions can be executed.
The same applies to the status-LEDs which form pairs or can be triggered separately, depending on the parameterization. In both cases, you can always parameterize how to trigger the status-LEDs.

You can separately parameterize the push sensor functions listed in the table to the different rockers or keys, respectively. This will dynamically change the parameter branch in the ETS plug-in and, consequently, the object table.

You can use the *"function of the rocker"* or *"function of the key"* parameter in the *"push-button sensor function/operation/ [key description]"* parameter branch to set the function to be executed upon the pressing of a key.

3.2.1 Rocker actuation

3.2.1.1 "No function"

If the *"function of the rocker"* parameter has been parameterized to *"no function"* the keys concerned and thus the associated object will be deactivated. Only the status-LEDs can be triggered through the status object. This triggering can be set by the *"show status object via"* parameter in the *"push sensor function/operation/[key description]/rocker X status"* parameter branch.

3.2.1.2 "Switching" function

If the function of the rocker has been parameterized to *"switching"* the *"command on pressing a rocker"* parameter and the *"switching"* object will appear. The *"command on pressing a rocker"* parameter defines the switching commands to be sent to the bus when the left and the right key are being pressed. Executable switching commands can be *"OFF"*, *"ON"* or *"TOGGLE"*. In case of *"TOGGLE"*, the value stored in the switching object will be changed over and sent. The commands are only set in combined form for the left and the right key and should be selected correspondingly. In addition, you can suppress the sending of a switching command when a key is being actuated (setting *"---"*).

The status-LEDs of the rocker (left and right) can be triggered through this status object. This triggering can be set by the *"show status object via"* parameter in the *"push sensor function/operation/[key description]/rocker X status"* parameter branch.

3.2.1.3 "Dimming" function

If the function of the rocker has been parameterized to *"dimming"* various parameters for the dimming function and the *"switching"* and *"dimming"* objects will appear. The *"command on pressing a rocker"* parameter defines the switching or dimming commands to be sent to the bus when the left and the right key are being pressed. Executable switching commands can be *"darker (OFF)"*, *"brighter (ON)"* or *"TOGGLE"*.
"Darker (OFF)" will release an OFF telegram upon short-time key actuation, whereas long-time actuation will cause a dimming telegram (darker). *"Brighter (ON)"* will release an ON telegram upon short-time key actuation, whereas long-time actuation will cause a dimming telegram (brighter). In case of *"TOGGLE"*, the switching status internally stored in the switching object will be changed over when the key is being pressed shortly. If the stored status is ON (OFF) an OFF (ON) telegram will be raised. By pressing the key for a long time a "darker" telegram will be sent after a "brighter" telegram, and vice versa.
The commands are only set in combined form for the left and the right key and should be selected correspondingly.

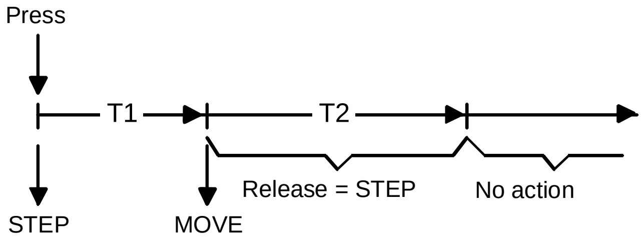
In addition, you can set *"reduce / increase brightness by"* and the *"time between switching and dimming"* options. A *"stop telegram"* at the end of a dimming process (telegram caused when the key is being released) can also be enabled. If you have set *"telegram repetition"* = *"yes"* dimming telegrams can be sent cyclically while a key is being pressed. In this connection, you can set the *"time between two dimming telegrams"* option. Each time when this period has elapsed, a new dimming telegram with the parameterized step width will be released.

The status-LEDs of the rocker (left and right) can be triggered through this status object. This triggering can be set by the *"show status object via"* parameter in the *"push-button sensor function/operation/[key description]/rocker X status"* parameter branch.

3.2.1.4 "Shutter/blind" function

If the function of the rocker has been parameterized to *"shutter/blind"* various parameters for the shutter/blind function and the *"short operation"* and *"long operation"* objects will appear. The *"operation concept"* parameter sets the short-time and long-time telegram sequence to be sent upon a key actuation or during a key actuation, respectively.

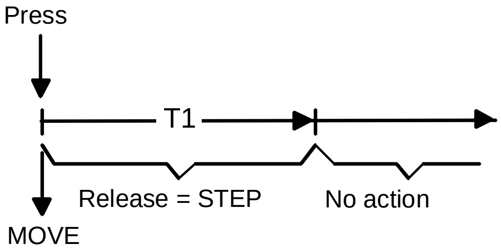
- *"STEP - MOVE - STEP"*:



When a key of the rocker is being pressed a STEP telegram will be sent, and start time T1 (*"time between short-time and long-time operation"*) will be started. If you release the key within time T1 no further telegram will be sent. This STEP serves for stopping an ongoing continuous run.

If the key remains pressed for longer than time T1 a MOVE telegram will be automatically sent after T1 has elapsed, and time T2 (*"lamella adjustment time"*) will be started. If you then release the key within time T2 the push sensor will send a STEP telegram. This function is used for lamella adjustment. T2 should correspond to a 180° lamella rotation.

- *"MOVE-STEP"*:



When a key of the rocker is being pressed a MOVE signal will be sent, and start time T1 ("lamella adjustment time") will be started. If you then release the key within time T1 the push sensor will send a STEP telegram. This function is used for lamella adjustment. T1 should correspond to a 180° lamella rotation.

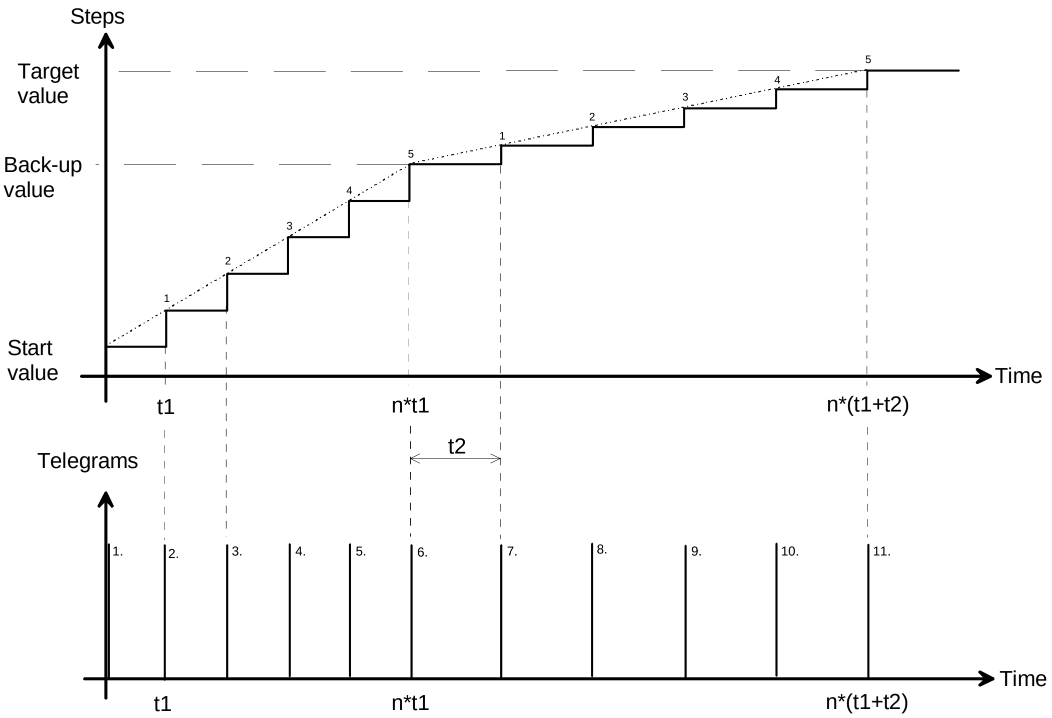
Which polarity the telegrams for long-time or short-time operation will have, i. e. which moving direction will be triggered in dependence on the actuated (left or right) key can be set by the *"command on pressing a rocker"* parameter. Executable commands can be *"UP"* or *"DOWN"*. The commands have only been preset in combined form for the left and the right key and should be selected correspondingly.

The status-LEDs of the rocker (left and right) can be triggered through the status object. This triggering can be set by the *"show status object via"* parameter in the *"push-button sensor function/operation/[key description]/rocker X status"* parameter branch.

3.2.1.5 "Universal value transmitter" function

The range of the universal value transmitter covers 8 bits. For this function, a start and a target value as well as a back-up value in between can be parameterized. The back-up value divides the value range into two sub-ranges. These sub-ranges are sub-divided into a preset number of steps which is equal for both sub-ranges. In addition, a time can be parameterized for each step of a sub-range. This time will specify at what time intervals the next step is to be reached or sent, respectively. The time consists of a common base for both sub-ranges and, in each case, of an own factor for each sub-range.

Example of the division of the value range and of the number of the steps for the universal value transmitter:



n: number of steps
t_1: time for a step from sub-range 1 consisting of the time base and time factor 1
t_2 time for a step from sub-range 2 consisting of the time base and time factor 2

Depending on the *"reaction on pressing a key"* parameter, you can either sweep through the entire value range as long as you keep the respective key pressed (*"continuous run as long as key is kept pressed"*), or you can start the continuous run with the first pressing and stop it with the second pressing (*"start or stop continuous run"/ default*). In the latter case, you can execute another key or rocker function in addition to the continuous run of the universal value transmitter, or you can allow several universal value transmitters of the individual rockers to run continuously at the same time. The direction of the continuous run can be defined by the *"left key effect"* parameter, i. e. you can set whether the start or the target value should be sent first when you actuate the left or the right key of a rocker.

Important:

- Ensure to keep the start value < back-up value < target value principle. If you do not parameterize these values as described the ETS plug-in will give an error message at this position and will automatically correct the values.
- In the worst case, the actual time of a step can slightly deviate from the parameterized time. This effect may occur when a high load is being imposed on the bus, in particular. In addition to time, deviations of the steps may also occur. As this function concerns an 8-bit value transmitter which does not provide any decimal positions in its telegram the calculation can only be in integer steps so that you will have to take into account slight deviations of the sent values. What can also happen is that the number of steps chosen will be bigger than the sub-range itself. In such case, the same values may be sent one after another several times.

3.2.2 Key actuation

3.2.2.1 "No function"

If the *"function of the key"* parameter has been set to *"no function"* the key will be deactivated. Only the status-LEDs can be triggered through the enabled status object. This triggering can be set by the *"function of the status-LED"* parameter in the *"push-button sensor function/operation/[key description]* parameter branch.

3.2.2.2 "Switching/key actuation" function

If the function of the key has been parameterized to *"switching/keying"* the *"command on pressing the key"* and *"command on releasing the key"* parameters and the *"switching"* object will appear.
The *"command on pressing/releasing the key"* parameter defines the switching commands to be sent to the bus when the key is being pressed or released, respectively. These two separate parameters even facilitate a single-key function (e. g. pressing = ON, releasing = OFF).
Executable switching commands can be *"OFF"*, *"ON"* or *"TOGGLE"*. In case of *"TOGGLE"*, the value stored in the switching object will be changed over and sent. In addition, the sending of a switching command upon the actuation of the key can be suppressed (*"no function"* setting).

You can use the *"function of the status-LED"* parameter in the *"push-button sensor function/operation/[key description]* parameter branch to set the function of the status-LED.

3.2.2.3 "Dimming" function

If the function of the key has been parameterized to *"dimming"* (single-key dimming) various parameters for the dimming function and the *"switching"* and *"dimming"* objects will appear. The *"command on pressing the key"* parameter defines the switching or dimming commands to be sent to the bus when the key is being pressed. Executable switching commands can be *"darker (OFF)"*, *"brighter (ON)"* or *"brighter/darker (TOGGLE)"*.
"Darker (OFF)" will release an OFF telegram upon short-time key actuation, whereas long-time actuation will cause a dimming telegram (darker). *"Brighter (ON)"* will release an ON telegram upon short-time key actuation, whereas long-time actuation will cause a dimming telegram (brighter). In case of *"brighter/darker (TOGGLE)"*, the switching status internally stored in the switching object will be switched over when the key is being pressed shortly. If the stored status is ON (OFF) an OFF (ON) telegram will be raised. By pressing the key for a long time a "darker" telegram will be sent after a "brighter" telegram, and vice versa.

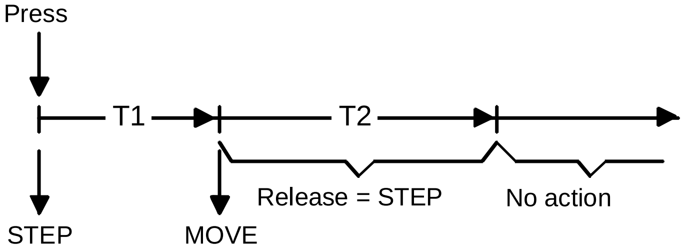
In addition, you can set *"reduce / increase brightness by"* and the *"time between switching and dimming"* options. A *"stop telegram"* at the end of a dimming process (telegram caused when the key is being released) can also be enabled. If you have set *"telegram repetition"* = *"yes"* dimming telegrams can be sent cyclically while a key is being pressed. In this connection, you can set the *"time between two dimming telegrams"* option. Each time when this period has elapsed, a new dimming telegram with the parameterized step width will be released.

You can use the *"function of the status-LED"* parameter in the *"push-button sensor function/operation/[key description]* parameter branch to set the function of the status-LED.

3.2.2.4 "Shutter/blind" function

If the function of the key has been parameterized to *"shutter/blind"* various parameters for the shutter/blind function and the *"short operation"* and *"long operation"* objects will appear. The *"operation concept (sequence of telegrams)"* parameter sets the short-time and long-time telegram sequence to be sent upon a key actuation or during a key actuation, respectively.

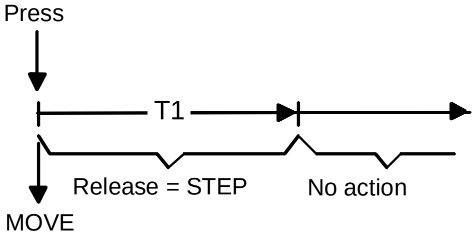
- *"Step - move - step"*:



When the key is being pressed a STEP message will be sent and start time T1 (*"time between short-time and long-time mode"*) started. If you release the key within time T1 no further telegram will be sent. This STEP serves for stopping an ongoing continuous run.

If the key remains pressed for longer than time T1 a MOVE telegram will be automatically sent after T1 has elapsed, and time T2 (*"lamella adjustment time"*) will be started. If you then release the key within time T2 the push sensor will send a STEP telegram. This function is used for lamella adjustment. T2 should correspond to a 180° lamella rotation.

- *"Move-step"*:



When the key is being pressed a MOVE telegram will be sent and start time T1 (*"lamella adjustment time"*) started. If you then release the key within time T1 the push sensor will send a STEP telegram. This function is used for lamella adjustment. T1 should correspond to a 180° lamella rotation.

Which polarity the telegrams for long-time or short-time operation will have, i. e. which moving direction will be triggered upon the actuation of the key can be set by the *"command on pressing the key"* parameter. Executable commands can be *"UP"*, or *"DOWN"*.

You can use the *"function of the status-LED"* parameter in the *"push-button sensor function/operation/[key description]* parameter branch to set the function of the status-LED.

3.2.2.5 "Value transmitter" and "analog value transmitter" functions

If the value transmitter has been parameterized (e. g. for dimming value transmitter applications) the push sensor will transmit an 8-bit value to the bus (according to EIS 6) when a key is being actuated. The value to be transmitted can be parameterized in the ETS plug-in and can be within the range from 0 to 255.

If the analog value transmitter option has been parameterized 2-byte values can be transmitted to the bus. In this connection, the *"function as"* parameter presets whether this value is a temperature value (according to EIS 5), a brightness value (according to EIS 5) or a dimensionless 2-byte counter value (according to EIS 10).

The range of the parameterizable temperature value is between 0 °C and 40 °C in steps of 1 °C. The brightness value can be within 0 lux and 1500 lux in 50-lux steps. If you parameterize brightness values which do not correspond to these 50-lux steps the plug-in will automatically correct the entered value by rounding up or down. The possible range for the 2-byte value transmitter is between 0 and 65535.

Value changing:

For value transmitter or analog value transmitter parameterization, you can change the value to be sent by pressing the key for a long time (> 5 s). In this connection, the set value will be decreased by the parameterized step width and sent. After the key is released, the value transmitted last will be kept stored. The next long-time key actuation will change the value setting direction.

You can parameterize the value changing step width for the 1-byte or the 2-byte value transmitter. The step width of the temperature value transmitter has been fixed to 1 °C and that of the brightness value transmitter to 50 lux.

The status-LED of the key actuated and that of the opposite key will blink (approx. 3 Hz) when you are changing a value (see next page). Do not press any other key while value changing is active.

You can use the *"function of the status-LED"* parameter in the *"push-button sensor function/operation/[key description]* parameter branch to set the function of the status-LED.

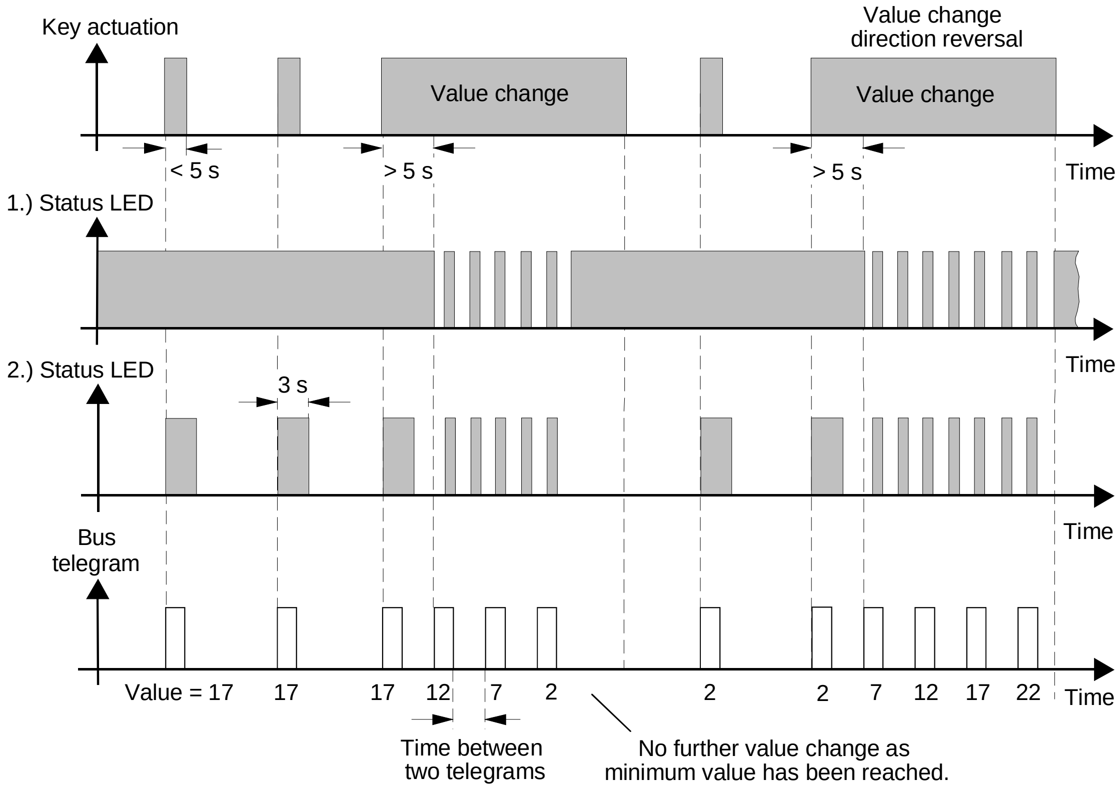
The following example shows how the value of the 1-byte value transmitter can be changed:

1.) *Function of the status-LED*	*Always ON*
Value (0...255)	*17*
Step width (1...10)	*5*

⇨ The status-LED will always be lit. When you are changing the value it will be blinking for the time of such value changing.

2.) *Function of the status-LED*	*Actuation indicator*
Lenghof operation display illumination	*3 s*
Value (0...255)	*17*
Step width (1...10)	*5*

⇨ When the key is being actuated the status-LED will be lit for the parameterized time. When you are changing the value it will be blinking for the time of such value changing.



Important:

- The value "0" may possibly never be attained by such changing in dependence on the parameterized step width (refer to the example above). This will ensure that the value originally parameterized by the ETS can be restored when another change is made.
- The values newly set by a change will only be saved in the RAM. After a bus voltage failure or after the application module has been detached and re-plugged, such values will thus be replaced by the preset values originally programmed through the ETS.

3.2.2.6 "Light scene extension/recall" function

For this function, you can distinguish between recalling an 'external' light scene through the light scene extension object or one of the internal scenes of the push sensor 2 plus. The *"function as"* parameter will set the mode of action.

If you have parameterized the function of the key to *"light scene extension"* the *"light scene extension"* object will be enabled. Upon short-time key actuation (< 1 s), you can use this object to recall via the bus light scenes stored with light scene functions in another bus device. In this connection, the light scene number (1 to 64) parameterized in the ETS plug-in will be transmitted.
If you have set the function to *"internal scene recall"* you can recall the scenes stored in the push sensor 2 plus by a short-time key operation (< 1 s). For this purpose, you must specify the corresponding scene number (1 to 8) in the ETS plug-in. An extension object will not be necessary for this function. Moreover, you can only recall an internal scene after you have enabled the scene function.
When working as actuation indicator, the status-LED will be lit for the parameterized time.

You can use the *"memory function"* parameter to specify whether you want to only recall the 'external' light scenes or the 'internal' ones, or if you want to save them, if required, upon a long-time key operation (> 5 s).

When parameterizing a *"light scene extension"* with the storage function, you can generate a memory telegram, depending on the parameterized light scene number. In this connection, long-time key operation (> 5 s) will cause the sending of such storage telegram.

When parameterizing an *"internal scene recall"* with the storage function, you can actuate the key for a long time (> 5 s) to save an internal scene in accordance with the parameterized scene number. In this connection, the scene control of the push sensor 2 plus will request the current values of the scene objects from the actuators via the bus and save them permanently.
Make absolutely sure that the read flags ("R"-flags) have been set for the corresponding actuators.

The status-LED of the key actuated and that of the opposite key will blink (approx. 3 Hz) during an active saving process. Do not actuate any other key at this state.
Short-time key actuation (< 1 s) will only recall the parameterized light scene. However, if you actuate the key for longer than 1 s, but shorter than 5 s, you will cause neither a recall nor a saving process.
When working as actuation indicator, the status-LED will be lit for the parameterized time.

Storage function examples:

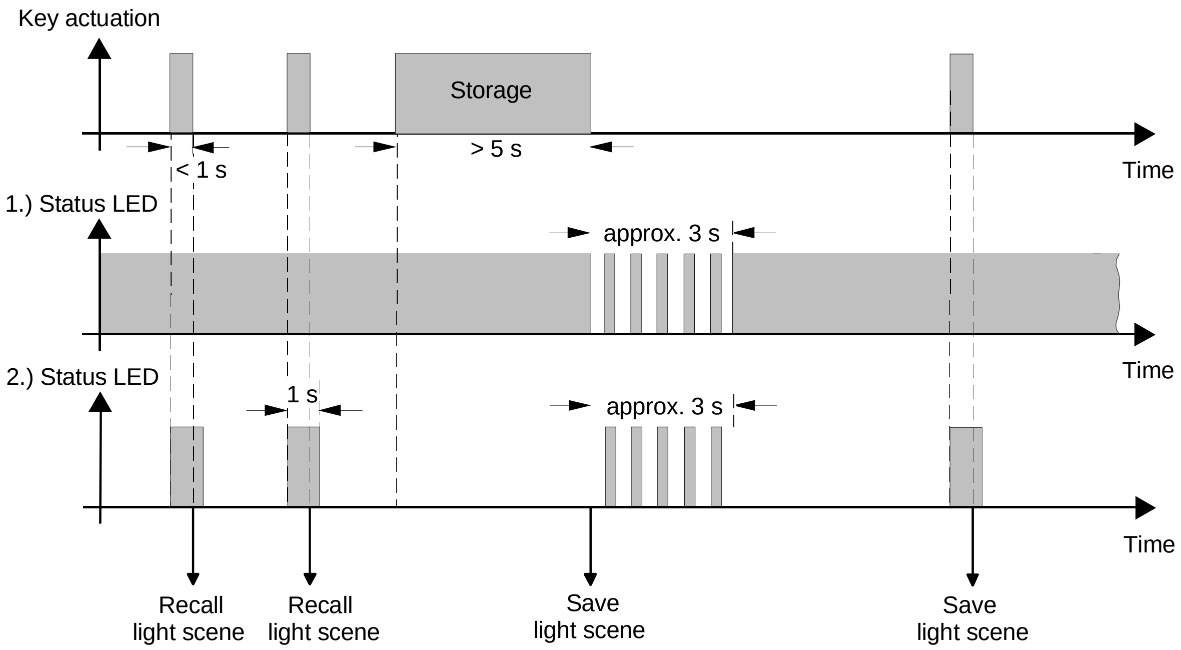
1.) *Function of the status-LED* *Always ON*

⇨ The status-LED will always be lit. It will begin to blink for some 3 s if storing takes place.

2.) *Function of the status-LED* *Actuation indicator*
Lenghof operation display illumination *1 s*

⇨ When the key is being actuated the status-LED will be lit for the parameterized time. It will begin to blink for some 3 s if storing takes place.

You can use the *"function of the status-LED"* parameter in the *"push-button sensor function/operation/[key description]* parameter branch to set the function of the status-LED.



3.2.2.7 "Heating timer operation" or "control function operation" function

Once the heating timer and/or one of the control functions has/have been enabled in the ETS plug-in, you can additionally set the following two operating functions for the key functions.

For heating timer operation, you can activate or deactivate the heating timer in dependence on the parameterized response to a key actuation. Toggling (between the activated and the deactivated status) will also be possible.
For a control function operation, you must specify in the ETS plug-in which of the two control functions you want to operate. For this purpose, the *"function"* parameter will define the type of action. Depending on the enabled control function(s), the choice for this parameter is automatically restricted. You can activate or deactivate the control function, depending on what response to a key actuation you have parameterized. Toggling (between the activated and the deactivated status) will also be possible.

You can parameterize the function of the status-LED. In addition to the *"OFF"*, *"ON"* and *"actuation indicator"* standard settings, you can select the *"key function indication active"* and *"key function indication inactive"* options. Thus, the status-LEDs can indicate whether a function linked with the associated key is activated or not activated, respectively. Such indication will also be made if the corresponding function has been activated or deactivated at operator level 1.
It should be noted that the operating function of the heating timer and of the control function(s) at operator level 1 will always be possible, independently of a function parameterized for the keys.

3.3 Push sensor inhibit function

The push sensor has an inhibit function, by means of which you can disable all keys or rockers, respectively. In addition, you can parameterize that all keys or rockers should behave in the same was as an explicitly preset rocker. You can use the *"inhibit behaviour"* parameter in the *"push-button sensor functions - disable"* parameter branch to enable the inhibit function. The following settings can be explained as follows:

Settings:

"Single rocker disabled" → - Rockers 2-3 or 2-5, respectively, can be separately disabled.
- In this connection, rocker 1 (room temperature controller operation) will always function.

"Push-button sensor disabled" → - The entire push sensor including rocker 1 will be disabled. In this case, you can only operate the room temperature controller via the bus if enabled for this purpose.

"Push-button sensor not disabled" → - No inhibit function is enabled (default). The push sensor will work in its normal mode.

"Performance of all rockers like rocker…" → - All rockers of the device (also rocker 1) will behave in the same way as in the the one parameterized here. In this case, the key or rocker functions assigned to the parameterized rocker will always be executed when you actuate any other rocker. The status-LEDs of the rockers will be triggered in the same way as in the 'normal mode'. In this case, you can only operate the room temperature controller via the bus if enabled for this purpose.

When the inhibit function is active the " 🔒 " symbol will be blinking in the display.

Please note that the room temperature controller operation can be additionally influenced by the controller inhibit function (refer to "4.6 Room temperature controller inhibit functions"). Thus, keys or rockers which have been assigned to room temperature controller operation must be disabled by the push sensor or controller inhibit function.

You can parameterize the polarity of the inhibit object.
If the polarity of the inhibit object is set to be *"inverted (disable = 0)"* the push sensor will not be immediately disabled upon bus voltage recovery or after a download. In this case, the inhibit function will only be activated when the inhibit object is updated (value = 0).

An inhibit function activated before a bus voltage failure will be permanently stored and therefore re-activated after a reset (bus voltage return).
After programming with the ETS, the inhibit function is always de-activated.

4. Room Temperature Controller Functions

4.1 Operating modes

The room temperature controller has various operating modes. The selection of these modes will, for example, facilitate the activation of different temperature setpoints, depending on the presence of a person, on the status of the heating or cooling system, on the time of the day, or on the day of the week.

- Comfort mode:

The comfort mode should be activated if persons are in a room and the room temperature should be adjusted to an adequately convenient value. The switch-over into this mode can also be controlled by the presence of persons.
The activated comfort mode will be indicated in the display by the " ⌂ " symbol.

- Standby mode

If a room is not used during the day because persons are absent, for example, you can activate the standby mode. Thereby, you can adjust the room temperature on a standby value thus to save heating or cooling energy, respectively.
The activated standby mode will be indicated in the display by the " 🚶⌂ " symbol.

- Economy mode

During the night hours, it mostly makes sense to adjust the room temperature to lower values for heating systems (e. g. in bedrooms) or to higher values for cooling systems (e. g. in office rooms). For this purpose, you can activate the economy mode.
The activated economy mode will be indicated in the display by the " ☾ " symbol.

- Frost/heat protection mode

Frost protection will be required if, for example, the room temperature must not fall below critical values while the window is open. Heat protection can be required where the temperature rises too much in an environment which is always warm, mainly due to external influences.
In such cases, you can activate the frost/heat protection mode and specify some temperature setpoint of its own for either option, depending on whether "heating" or "cooling" has been selected, to prevent freezing or overheating of the room.
The activated frost/heat protection mode will be indicated in the display by the " ❄≈ " symbol.

- Comfort prolongation (temporary comfort mode)

You can activate the comfort prolongation option from the economy or frost/heat protection mode (not triggered by the *"window status"* object) and use it to adjust the room temperature to a comfort value for some time if, for example, the room is also 'used' during the night hours. This mode can only be activated by a parameterized presence key or, in this case, also by the presence object, respectively. The comfort prolongation option will be automatically deactivated after a definable time has elapsed, or by pressing the presence key once more, or by receiving a presence object value = 0, respectively. You cannot re-trigger this prolongation.
The activated comfort prolongation option will be indicated in the display by the " ⌂☾ " or " ⌂❄≈ " symbols.

You can assign an own temperature setpoint to each operating mode and to each "heating"/"cooling" switch-over (refer to "4.4 Temperature setpoints").

Only one operating mode can be activated at a time so that both control loops will always be in the same mode if two control loops are used.

4.1.1 Changing the operating modes

You can activate or switch over the operating modes in various ways. Depending on one another in priority, activation or switching over is possible by…

a) local operation on the push sensor (rocker 1) at operator level 1 or 2, if enabled;

b) local operation on the push sensor (rockers 2-3 or 2-5, respectively) and parameterized room temperature controller operation;

c) the objects separately available for each operating mode, or, alternatively, by the KONNEX objects.

About a):
You can activate operator level 1 (refer to "1.5 Operator levels/Local operation) to select the "comfort", "standby" or "economy" operating modes.
At operator level 1 or 2, you cannot change to the comfort prolongation or frost/heat protection options.

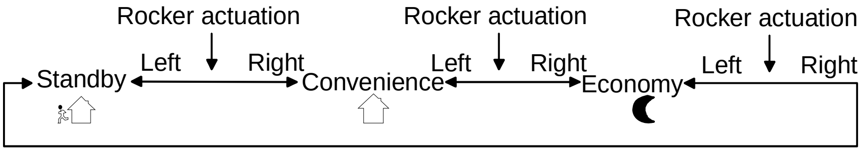
About b):
In addition to operating rocker 1, you can parameterize the "operation of room temperature controller" function to rockers 2-3 or 2-5 of the push sensor (refer to "3.2 Rocker functions"). In this connection, distinction is made between key and rocker functions:

- Key function:
  The function of a key has been set to "operation of room temperature controller". In this case, you can set in the ETS plug-in which operating mode you want to activate by pressing this key. For this purpose, the "comfort", "standby", "economy" and "frost/heat protection" modes are available.

  To be able to activate the comfort prolongation mode you can use the *"presence detection"* parameter in the *"room temperature controller function - functionality"* parameter branch to additionally enable the presence key in the ETS plug-in. If enabled, the *"presence object"* will appear, and you can select the *"presence key"* setting from the key functions. In this way, you can actuate the presence key to change to the comfort prolongation mode or to deactivate the latter earlier when the economy or frost/heat protection mode has been activated. Also, you can switch over from the standby to the comfort mode when you actuate the presence key.

  You can parameterize the function of the status-LED. In addition to the *"OFF"*, *"ON"* and *"actuation indicator"* standard settings, you can select the *"key function indication active"* and *"key function indication inactive"* options. Thus, the status-LEDs can indicate whether a function linked with the associated key is activated or not activated, respectively. For this purpose, the corresponding operating mode need not have been activated or deactivated by a key operation.

- Rocker function:
  The function of a rocker has been set to "operation of room temperature controller". In such case, you can use the left or right key of the rocker to change the operating mode. The switch-over sequence is from the "comfort" to the "standby" and then to the "economy" mode.



  For a rocker function, it will not be possible to activate the frost/heat protection mode or the comfort prolongation mode. If you have activated frost/heat protection or comfort prolongation from the frost/heat protection mode you cannot change these operating modes by a local rocker function.

  Via the status object, you can trigger the status-LEDs of the rocker in the same way as for a push sensor rocker function, irrespective of the room temperature controller operation.

If the room temperature controller operation function is disabled local operation can be disabled by rockers 2-3 and 2-5, respectively (refer to "4.6.2 Disabling controller operation").

About c):
Distinction is made whether the operating modes should be switched over via separate 1-bit objects or, alternatively, by the 1-byte KONNEX objects. You can use the *"operating mode switch-over"* parameter in the *"room temperature controller function"* parameter branch to set the way how to switch over.

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

There is a separate 1-bit switch-over object for each operating mode. Each of these objects allows the current operating mode to be switched over or to be set, depending on the priority.

Taking account of the priority, the following hierarchy will result from the operating mode switch-over by the objects, distinction being made between presence detection by the presence key (Table 1/Fig. 1) and by the presence detector (Table 2/Fig. 2 on the next page):

Table 1

"Operating mode switch-over" objects				Window status	Presence key object	Activated operating mode
❄ Obj. no. 36	⌂ Obj. no. 33	🚶⌂ Obj. no. 34	☾ Obj. no. 35	Obj. no. 39	Obj. no. 38	
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	Economy ☾
1	X	X	X	0	1	Comfort prolongation ⌂❄
0	1	X	X	0	1	Comfort ⌂
0	0	1	X	0	1	Comfort ⌂
0	0	0	1	0	1	Comfort prolongation ⌂☾
0	0	0	0	0	0	Last valid mode set
0	0	0	0	0	1	Comfort/ Comfort prolongation *

X = irrelevant
*: Depending on the last valid mode set.

Fig. 1:

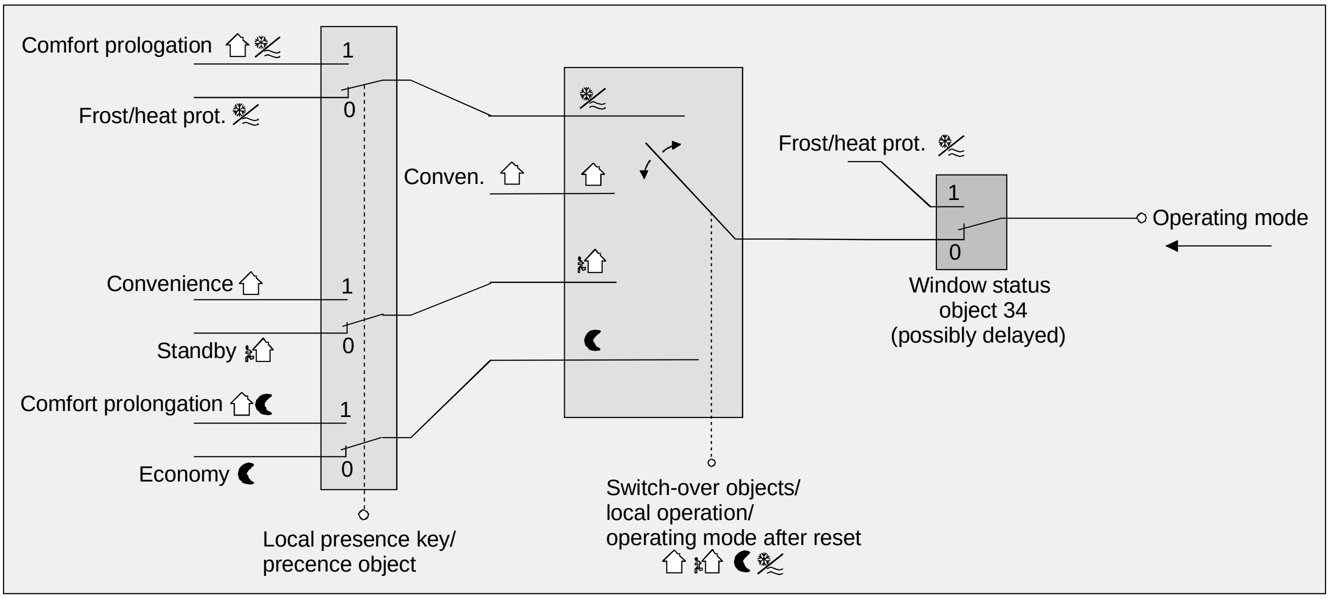
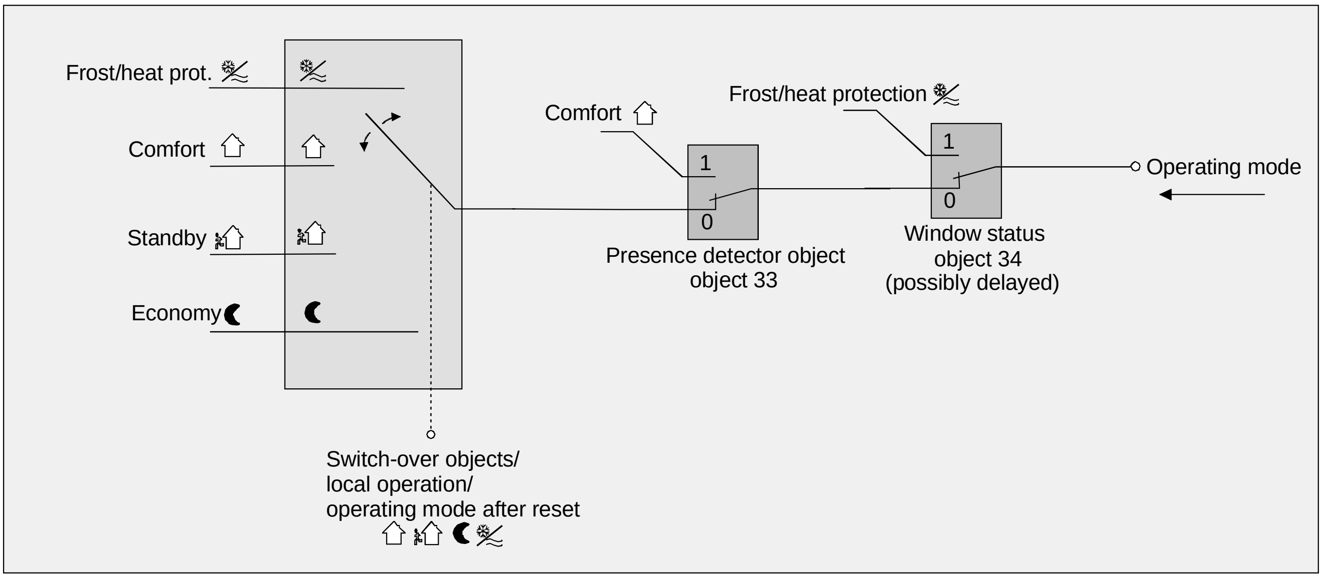


Table 2

"Operating mode switch-over" objects				Window status	Presence detector object	Activated operating mode
❄	⌂	⌂	☾			
Obj. no. 36	Obj. no. 33	Obj. no. 34	Obj. no. 35	Obj. no. 39	Obj. no. 38	
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	Economy ☾
0	0	0	0	0	0	Last valid mode set

X = irrelevant

Fig. 2:



Notes on operating mode switch-over through "switching" (4 x 1 bit):

- When the operating modes are switched over the objects (comfort / standby / economy / frost/heat protection mode) will always be updated at the same time and can be read out, if necessary (set "R"-flag). If the "T"-flag has been set for these objects the current values will, in addition, be actively transmitted to the bus when they are changed. After bus voltage recovery or after initialization, respectively, the object which corresponds to the selected operating mode will be updated and its value actively transmitted to the bus if the "T"- flag has been set.
- A switch-over via the objects is equal to a local switch-over made on the push sensor, taking account of the priorities of the operating modes. You can operate the room temperature controller on the device by a key or rocker function to switch over an operating mode set by an object.

  Restriction: If you have activated frost/heat protection or comfort prolongation from the frost/heat protection mode you cannot change these operating modes by a local rocker function or by the objects for the "comfort", "standby" and "economy" modes. However, you can change the operating modes by a key function (e. g. comfort mode, standby mode, etc.) or through operator level 1 if no higher-order mode (e. g. window contact/presence detector) has been activated.
- When a presence key has been parameterized:
  The presence object will be active ("1") for the period of active comfort prolongation.
  The presence object will be automatically deleted ("0") if comfort prolongation is stopped after the prolongation time has elapsed, or if the operating mode has been changed by a higher-order operation through the switch-over objects or by local operation.
- If you use further push sensor 2 pluss as extensions to switch over the operating modes such switch-over should only be effected by keys or rockers (push sensor functionality) which have been parameterized with the "switching" function. Otherwise (for example, for extension parameterization as "room temperature controller operation") the priority evaluation of the incoming telegrams can activate an undesired operating mode at the main unit (push sensor 2 plus acting as room temperature controller).
  If extensions are used, it is recommended to work with KONNEX opmode switch-over function (see next chapter).
- Only one operating mode can be activated at a time so that both control loops will always be in the same mode if two control loops are used.

The operating mode switch-over of the second control loop always proceeds in parallel with the first control loop.

- Operating mode switch-over through "value" (2 x 1 byte):

There is a common 1-byte switch-over object for all operating modes. During the running time, the operating mode can be switched over through this value object immediately after the receipt of only one telegram. In this connection, the value received will set the operating mode.
In addition, a second 1-byte object is available which, by forced control and of higher order, can set an operating mode, irrespective of any other switch-over options. According to the KONNEX specification, both 1-byte objects have been implemented.

Taking account of the priorities, the following hierarchy will result from the operating mode switch-over by the objects, distinction being made between presence detection by the presence key (Table 1/Fig. 1) and by the presence detector (Table 2/Fig. 2 on the next page):

Table 1

"Operating mode switch-over" object ** Obj. no. 33	"Forced operating mode" object *** Obj. no. 37	Window status Obj. no. 39	Presence key object Obj. no. 38	Activated operating mode
X	01	X	X	Comfort ⌂
X	02	X	X	Standby ⌂
X	03	X	X	Economy ☾
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	Economy ☾
04	00	0	0	Frost/heat protection ❄
01	00	0	1	Comfort ⌂
02	00	0	1	Comfort ⌂
03	00	0	1	Comfort prolongation ⌂☾
04	00	0	1	Comfort prolongation ⌂❄
00	00	0	0	Last valid mode set
00	00	0	1	Comfort/comfort prolongation *

*: Depending on the last valid mode set. / X = irrelevant

**: Values higher than "04" will not be evaluated. A value of "00" will keep active the last valid operating mode that has been set.

***: Values higher than "04" will not be evaluated. A value of "00" stands for a deactivated constraint object.

Fig. 1:

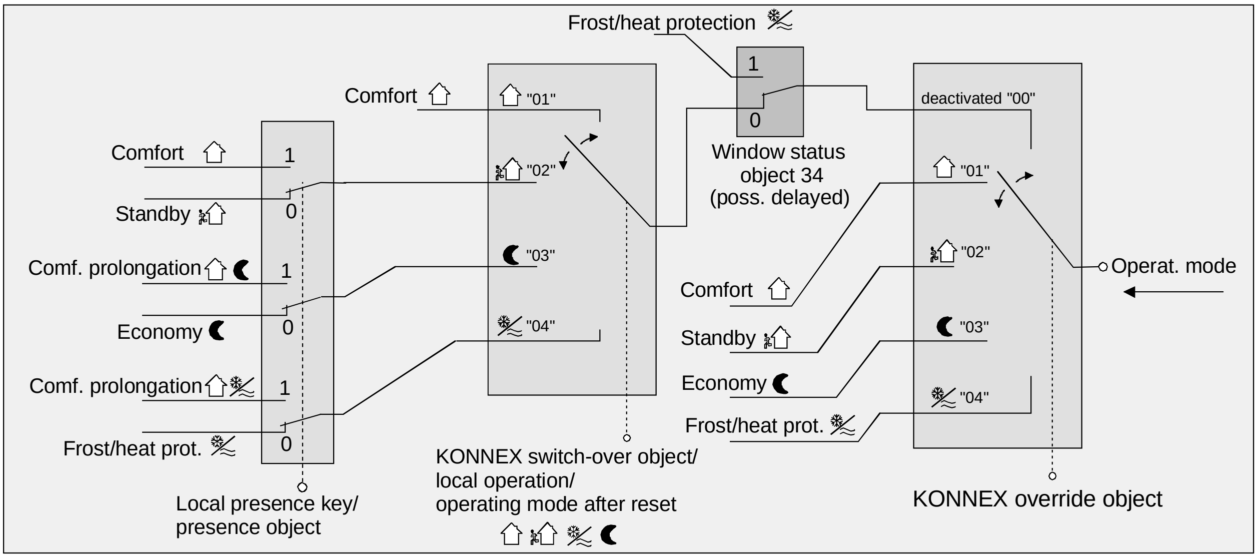


Table 2

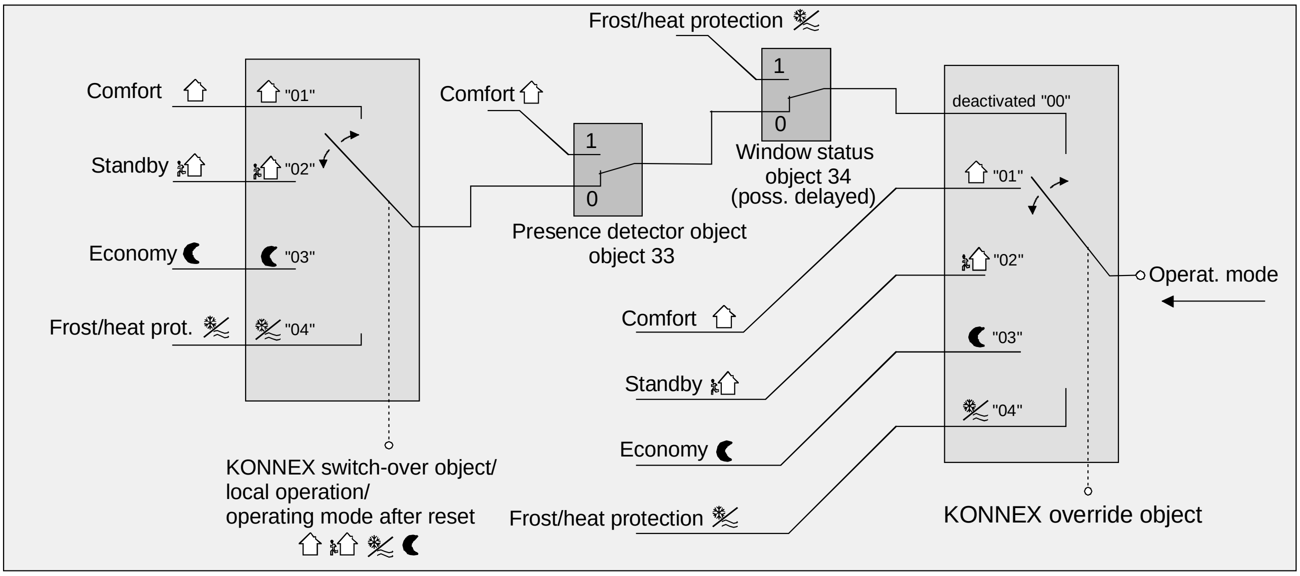
"Operating mode switch-over" object ** Obj. no. 33	"Override" object *** Obj. no. 37	Window status Obj. no. 3	Presence detector object Obj. no. 38	Activated operating mode
X	01	X	X	Comfort ⌂
X	02	X	X	Standby ⌂
X	03	X	X	Economy ☾
X	04	X	X	Frost/heat protection ❄
X	00	1	X	Frost/heat protection ❄
X	00	0	1	Comfort ⌂
01	00	0	0	Comfort ⌂
02	00	0	0	Standby ⌂
03	00	0	0	Economy ☾
04	00	0	0	Frost/heat protection ❄
00	00	0	0	Last valid mode set

X = irrelevant

**: Values higher than "04" will not be evaluated. A value of "00" will keep active the last valid operating mode that has been set.

***: Values higher than "04" will not be evaluated. A value of "00" stands for a deactivated constraint object.

Fig. 2:



Notes on operating mode switch-over through "value" (2 x 1 byte):

- When the operating modes are switched over the KONNEX switch-over object will always be updated at the same time and can be read out, if necessary (set "R"-flag). If the "T"-flag has been set for this object the current value will, in addition, be actively transmitted to the bus when it is changed. After bus voltage recovery or after initialization, respectively, the object which corresponds to the selected operating mode will be actively transmitted to the bus if the "T"-flag has been set.
- A switch-over via the KONNEX switch-over object is equal to a local switch-over made on the push sensor, taking account of the priorities of the operating modes. You can operate the room temperature controller on the device by a key or rocker function to switch over an operating mode set by an object.

  Restriction: If you have activated frost/heat protection or comfort prolongation from the frost/heat protection mode you cannot change these operating modes by a local rocker function. However, you can change the operating modes by a key function (e. g. comfort mode, standby mode, etc.) or through operator level 1 if no higher-order mode (e. g. window contact/presence detector) and no KONNEX constraint object has been activated.

  The KONNEX constraint object will always have the highest priority.
- When a presence key has been parameterized:
  The presence object will be active ("1") for the period of active comfort prolongation.
  The presence object will be automatically deleted ("0") if comfort prolongation is stopped after the prolongation time has elapsed, or if the operating mode has been changed by a higher-order operation through the switch-over objects or by local operation, or if an operating mode forced by the KONNEX constraint object is being deactivated (constraint object → "00").
- Only one operating mode can be activated at a time so that both control loops will always be in the same mode if two control loops are used.

The operating mode switch-over of the second control loop always proceeds in parallel with the first control loop.

4.1.2 Notes on the operating modes

Presence function/comfort prolongation:

By a presence detection, the room temperature controller can quickly switch over to comfort prolongation upon a key actuation or go into the comfort mode when a movement is being detected. In this connection, you can use the *"presence detection"* parameter in the *"room temperature controller function - functionality"* parameter branch to set whether presence detection should be movement-controlled by a presence detector or manual through key actuation:

• Presence detection by the presence key:

If you enable the presence key for presence detection the *"presence object"* will appear, and you can select the *"presence key"* setting from the key functions.

In this way, you can actuate the presence key or use a presence object value = "1" to switch over to comfort prolongation when the economy or the frost/heat protection mode is active (not activated by the *"window status"* object or by the automatic frost protection). The prolongation will be automatically deactivated as soon as the parameterized *"length of comfort prolongation"* time has elapsed. If you press the presence key once more, or if the object receives a value ="0", you can deactivate the comfort prolongation earlier. You cannot re-trigger such prolongation time.
If you have set the length of comfort prolongation to "0" you cannot activate a comfort prolongation from the economy or frost/heat protection mode. In this case, the operating mode will not be changed, although the presence function has been activated.
If the standby mode is active you can actuate the presence key or use a presence object value = "1" to switch over to the comfort mode. This will also be the case if you have parameterized the length of comfort prolongation to "0". The comfort mode will remain active as long as the presence function remains active, or until another operating mode comes into effect.

The presence object or the presence function, respectively, will always be deleted whenever a switch-over to a different operating mode takes place, or after a forced mode has been deactivated (associated with KONNEX forced switch-over). The presence object is bidirectional ("W"- and "T"-flags set by default) so that telegrams with the corresponding object values will be released upon activation (= "1") or deactivation (= "0"), respectively. A presence function including its object activated prior to a reset will always be deleted after the reset.

• Presence detection by the presence detector:

If you enable a presence detector for presence detection the *"presence object"* will appear. Via this object, you can integrate presence detectors into room temperature control.
If a movement is detected ("1" telegram) the controller will switch over into the comfort mode. In this connection, it will not be relevant what has been set by the switch-over objects or by local operation directly on the push sensor. Only the window contact or the automatic frost protection or the KONNEX constraint object are of higher priority.
After some delay time has elapsed in the presence detector ("0" telegram), the controller will return to the mode which was active before the presence detection, or it will compensate the telegrams of the switch-over objects received during presence detection, respectively.
During active presence detection, you cannot change the operating mode on the push sensor.
A presence function activated prior to a reset will always be deleted after the reset. In this case, the presence detector must transmit a new "1" telegram to activate the presence function.

Window status/automatic frost protection:

The push sensor 2 plus offers various options to switch over into the frost/heat protection mode. In addition to the switch-over by the corresponding switch-over object or by room temperature controller operation on the push sensor (key function), the frost/heat protection mode can by activated by a window contact or, alternatively, frost protection can be activated by an automatic temperature control option. In this connection, the window contact or the automatic control is given a higher priority among these options (refer to "4.1.1 Changing the operating modes"). You can use the *"frost/heat protection"* parameter in the *"room temperature controller function"* parameter branch to set the way how such higher-priority switch-over will take place:

• Frost/heat protection switch-over *"via window status"*:
Object 39, *"window status"* is enabled. A telegram having the value of = "1" (open window) and sent to this object will activate the frost/heat protection mode. If this is the case, this operating mode cannot be deactivated, neither by local operation nor by the switch-over objects (with the exception of the KONNEX constraint object).
Only a telegram with the value of = "0" (closed window) will reset the window status and deactivate the frost/heat protection mode. After this, the operating mode set before the opening of the window or that mode followed up via the bus while the window was open will be activated.

You can optionally parameterize a window status delay. Such delay can make sense if short ventilation of the room by opening the window is not supposed to change the operating mode. You can use the *"window status delay"* parameter to set this delay time between 1 and 255 minutes. The window status will only be changed and thus the frost/heat protection mode activated after this parameterized time has elapsed. A setting of "0" will effect the immediate activation of the frost/heat protection mode when the window is open. The window status will be in effect in the heating and in the cooling mode. The *"window status"* object will be deleted after a reset.

• Frost protection mode switch-over by *"automatic frost protection"*:
For this setting, a switch-over to the frost protection mode can be made from time to time, depending on the room temperature determined. If there are no window contacts, this setting can prevent unnecessary heating up of the room when windows or external doors are open.
In connection with this function, a quick temperature drop can be detected by measuring the actual temperature every minute as, for example, is the case when a window is open.
If the temperature drop detected reaches a parameterized value the room temperature controller will automatically switch over to the frost protection mode. The *"automatic frost protection"* parameter sets the maximum temperature drop in K/min. for switching over to the frost protection mode.
After the time preset by the *"automatic frost protection"* parameter has elapsed, the controller will return into the mode which was set before frost protection. Re-triggering will not be possible.
If a new operating mode was received by the KONNEX switch-over object during frost protection activated by 1 byte this followed-up mode will be set after automatic frost protection.
If a new operating mode was received by the switch-over objects during frost protection activated by 4 x 1 bit this newly received operating mode will be discarded after the end of the automatic frost protection period so that the controller will further stay in the frost protection mode. Only now, you can change the operating mode through the objects or locally on the push sensor.
The KONNEX constraint object has a higher priority than the automatic frost protection mode and can interrupt the latter.

The automatic frost protection mode only acts on heating for temperatures below the setpoint temperature of the operating mode selected. Thus, no automatic switch-over to frost protection can take place at room temperatures in the dead band or in the active cooling mode if the "heating and cooling" mode is on. Automatic heat protection activation is not intended with this parameterization.

Compared with the alternative setting of the frost/heat protection detection by the window contact, the automatic frost protection mode will have the same priority when the operating mode is being changed.

Note:
Frequent draughts in a room can cause unintentional activation/deactivation of frost protection when the automatic frost protection mode is active, and if the parameterized temperature drop is not low enough. Switching into the frost/heat protection mode by window contacts should be preferred to the automatic option.

Operating mode after reset:

In the ETS plug-in, you can use the *"operating mode after reset"* parameter in the *"room temperature controller function - functionality"* parameter branch to set which operating mode you want to be activated after bus voltage recovery, after a programming process by the ETS, or after re-plugging the application module to the bus coupling unit. In this connection, the following settings will be possible:

- *"Comfort mode"*: The comfort mode will be activated after the initializing phase.
- *"Standby mode"*: The standby mode will be activated after the initializing phase.
- *"Economy mode"*: The economy mode will be activated after the initializing phase.
- *"Frost/heat protection mode"*: The frost/heat protection mode will be activated after the initializing phase.

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

4.1.3 Controller status

The room temperature controller can send out its status. For this purpose, an optional centralized status signal (1-byte type) or, alternatively, one up to eight single status signals (1-bit type) are available.
The *"status controller"* parameter in the *"room temperature controller function - variable and status out put"* parameter branch will enable the status signal and set the status format.

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

One-byte status object 41 contains the entire status information. Controlled by the control algorithm, the status will be actively transmitted to the bus in cycles every 30 seconds (provided that the "T"-flag has been set).If you set the "R"-flag you can read out the status.

Setting	Data Description	
Controller in general 1-byte type	Bit 0: 1: Comfort mode active Bit 1: 1: Standby mode active Bit 2: 1: Economy mode active Bit 3: 1: Frost/heat prot. mode active	Bit 4: 1: Controller disabled Bit 5: 1: Heating; 0 cooling Bit 6: 1: Controller inactive (dead band) Bit 7: 1: Frost alarm ($T_{room} \leq + 5$ °C)

- *Status controller" = "transmit individual state"*:

One-bit status object 36 contains the status information selected by the *"individual state"* parameter. Controlled by the control algorithm, the status will be actively transmitted to the bus in cycles every 30 seconds (provided that the "T"-flag has been set). If you set the "R"- flag you can read out the status.

Parameterization for "Individual State"	Data Description	
Comfort mode active	1: Conv. mode/prolongation active	0: No comfort mode
Standby mode active	1: Standby mode active	0: No standby mode
Economy mode active	1: Economy mode active	0: No economy mode
Frost/heat prot. mode active	1: Frost/heat protection mode active	0: No frost/heat protection mode
Controller disabled	1: Contr. disabled (dew-point operation)	0: Controller not disabled
Heating/cooling mode	1: Heating mode	0: Cooling mode
Controller inactive	1: Controller inactive (dead band)	0: Controller active
Frost alarm	1: Frost alarm ($T_{room} \leq + 5$ °C)	0: No frost alarm ($T_{room} > + 5$ °C)

Explanation of the status signals:

- Comfort mode: It will be active if the "comfort '⌂'" mode or some comfort prolongation " ⌂ ☾ " or " ⌂ ❄ ", respectively, has been activated.
- Standby mode: It will be active if the "standby '⌂'" mode has been activated.
- Economy mode: It will be active if the "economy '☾'" mode has been activated.
- Frost/heat protection mode: It will be active if the "frost/heat protection '❄'" mode has been activated.
- Controller disabled: It will be active if controller disabling has been activated (dew-point operation).
- Heating/cooling mode: It will be active if the heating mode has been activated, and will be inactive when the cooling mode has been activated. (It will be inactive if the controller has been disabled.)
- Controller inactive: It will be active in the *"heating and cooling"* mode, if the room temperature determined is within the dead band. In the individual "heating" or "cooling" mode, this status information will always be "0". (It will be inactive if the controller has been disabled.)
- Frost alarm: It will be active if the room temperature determined has reached +5 °C or is below this value. This status signal will have no special influence on the control behaviour.

Upon a reset, status object 41 will be updated after the initializing phase. After this, the status will be updated every 30 seconds in parallel with the variable calculation of the controller.

4.2 Operating modes and operating mode switch-over

The room temperature controller has up to two different operating modes. These operating modes specify whether you want the controller to use its variable to trigger heating systems *("heating"* single mode) or cooling systems *("cooling"* single mode). You can also activate mixed operation, with the controller being capable of switching over between *"heating"* and *"cooling"* automatically or alternatively, controlled by an object.
In addition, you can establish two-stage control operation for triggering an additional heating or cooling unit. For two-stage control, separate variables will be calculated as a function of the temperature deviation between the setpoint and the actual value and transmitted to the bus for the basic and additional stages.
In this connection, the *"operating mode switch-over"* parameter in the *"room temperature controller functions"* parameter branch sets the operating mode to be executed and, if necessary, enables the additional stage(s).

In the individual *"heating"* or *"cooling"* modes without any additional stage, the controller will always work with one variable and, alternatively, when the additional stage is enabled it will use two variables in the parameterized mode. Depending on the room temperature determined and on the specified setpoint temperatures of the operating modes (refer to "4.4 Temperature setpoints"), the room temperature controller will automatically decide whether heating or cooling energy will be required and will calculate the variable for the heating or cooling system (refer to "4.3 Room temperature control and variables).
For *"heating"* or *"cooling"*, the controller will always be in the operating mode set in the ETS plug-in after a reset (bus voltage recovery, re-programming by the ETS, or re-plugging of the application module).

In the *"heating and cooling"* mode, the controller is capable of triggering heating and cooling systems. In this connection, you can set the switch-over behaviour of the operating modes.

- *"Switch over between heating and cooling"* parameter in the *"room temperature controller functions"* parameter branch set to *"automatically"*:

  In this case, a heating or cooling mode will be automatically activated, depending on the room temperature determined and on the given temperature basic setpoint, or on the dead band, respectively. If the room temperature is within the preset dead band neither heating nor cooling will take place (both variables = "0") In this connection, the display will read the heating temperature setpoint of the activated operating mode when you actuate rocker 1. If the room temperature is higher than the cooling temperature setpoint cooling will take place. If the room temperature is lower than the cooling temperature setpoint heating will take place.

  When automatic heating/cooling switch-over takes place information can be actively sent to the bus through object 40, *"heating/cooling switch-over"* to indicate whether the controller is working in the heating mode ("1" telegram) or in the cooling mode ("0" telegram). In this connection, the *"automatic heating/cooling switch-over transmission"* parameter specifies when an operating mode switch-over will be transmitted.

  - *"On changing the heating / cooling"* setting:
    In this case, a telegram will only be transmitted on changing from heating to cooling (object value = "0") or from cooling to heating (object value = "1"), respectively.
  - *"On changing the output value"* setting:
    By this setting, the current mode will be transmitted whenever the output variable changes. If the variable = "0" the mode which was active last will be transmitted.

  If the room temperature determined is within the dead band the mode activated last will be retained in the object until a switch-over into the other operating mode takes place, if necessary.

  In addition, the object value can be output in cycles when automatic switch-over is being made. The *"cyclic transmission heating/cooling switch-over"* parameter enables cyclic transmission (factor > "0" setting) and specifies the cycle time.

  Note on automatic mode switch-over:

  Selecting too narrow a dead band may possibly result in permanent changing between heating and cooling. For this reason, you should, if possible, not set the dead band (temperature difference between the setpoint temperatures for the heating and cooling convenient modes) below the default value.

- *"Switch-over between heating and cooling"* parameter in the *"room temperature controller functions"* parameter branch set to *"via object"*:

  In this case, heating/cooling switch-over will be controlled via object 35, *"heating/cooling switch-over"*, independently of the dead band. This type of switch-over can, for example, become necessary if both heating and cooling should be effected through a one-pipe system (heating and cooling system). For this purpose, the temperature of the medium in the one-pipe system must, first of all, be changed by the control system of the installation. Subsequently, you can select the heating/cooling switch-over (often, cold water is used in the one-pipe system for cooling n summer, while hot water is used for heating in winter).
  The *"heating/cooling switch-over"* object has the following polarities: "1": heating; "0": cooling. After a reset, the object value will be "0", with *"heating/cooling switch-over after reset"* being activated.

  You can use the *"heating/cooling switch-over after reset"* parameter to set which operating mode you want to activate after a reset. For the *"heating"* or *"cooling"* settings, the controller will activate the parameterized heating/cooling switch-over immediately after the initializing phase. If you have parameterized *"heating/cooling switch-over before reset"* the mode which was selected before the reset will be activated.
  If a switch-over is made through the operating modes object the mode will first be changed into the one specified to be activated after a reset. Only after the device receives an object update a switch-over into the other mode will take place, if necessary.

  Notes on the *"heating/cooling switch-over before reset"* setting:

  - Frequent changing of the operating mode (e. g. several times a day) during running operation can adversely affect the life of the device as the read-only memory (EEPROM) used has been designed for less frequent write access events only.

Heating and cooling in the mixed mode at the same time (variables > "0") is, in general, not possible.
Only if heating or cooling energy is required in one of the modes and, consequently, the variable is > "0" the " ♨ " or " ❄ " symbol will appear in the display.

Heating/cooling signal:

Depending on the operating mode set, you can output information through separate objects whether heating or cooling energy is required at the moment, i. e. whether heating " ♨ " or cooling " ❄ " takes place.
As long as the heating (cooling) variable is > "0", a "1" telegram will be transmitted through the *"heating"* (*"cooling"*) signal object. Only after the variables have become = "0", the signal telegrams will be reset ("0" telegram being transmitted).

Exception: For two-point control it should be noted that the " ♨ " or " ❄ " symbol will appear in the display, or that the heating or cooling signal objects will already become active once the temperature falls below the setpoint of the active operating mode for heating or exceeds that for cooling. In this connection, the parameterized hysteresis will be disregarded (refer to "4.3.1 Control algorithms, control loops and calculation of variables").

Heating and cooling at the same time will not be not possible. The signals are only referred to the control loop.

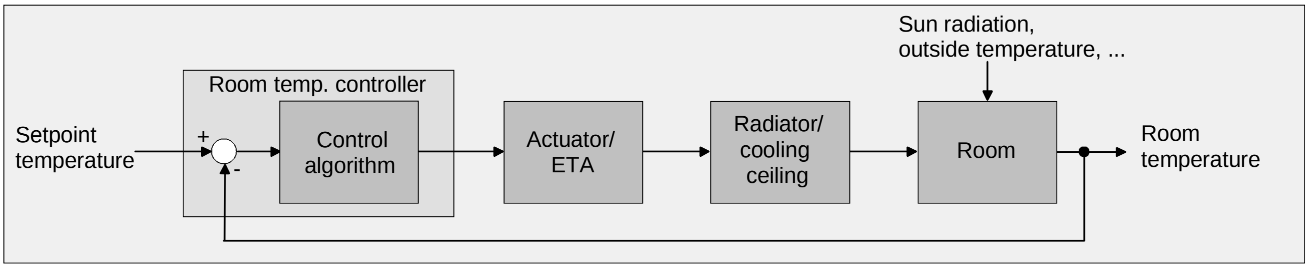
The signal objects can be enabled by the *"heating signal"* or *"cooling signal"* parameters in the *"variable and status output"* parameter branch.

The control algorithm (refer to "4.3 Room temperature control and variables") controls the signal objects. Please note that the variable is re-calculated exclusively every 30 s , followed by an updating of the signal objects.

4.3 Room temperature control and variables

4.3.1 Control algorithms, control loops and calculation of variables

To facilitate convenient temperature control in a living room a specific algorithm which controls the installed heating or cooling systems is required. Taking account of the preset temperature setpoints and the actual room temperature, the controller thus determines variables which trigger the heating or cooling system.
The control system (control loop) consists of a room temperature controller, an actuator or switching actuator (when ETA electrothermal actuators are used), the actual heating or cooling element (e. g. radiator or cooling ceiling) and of the room. This results in the following control system:



The controller measures the actual temperature (room temperature determined) and compares it with the preselected temperature setpoint. With the aid of the selected control algorithm, the variable is then calculated from the difference between the actual and the setpoint temperature. By resetting the variable at regular intervals, the controller is thus capable of compensating in the control loop temperature differences between the actual and the desired values caused by external influences (e. g. intensive sun radiation or varying outside temperatures). In addition, the flow temperature of the heating or cooling circuit influences the control system, which necessitates adaptations of the variable.

The room temperature controller in the push sensor 2 plus facilitates either proportional/integral (PI) control as a continuously working or switching option, or switching 2-point control.

In some practical cases, it can become necessary to use more than one control algorithm. For example, in bigger systems using floor heating, one control loop which only triggers the floor heating can be used to keep the latter at a constant temperature. The radiators on the wall, and possibly even in a side area of the room, will be controlled separately by another algorithm.
In such cases, distinction must be made between the different types of control, as floor heating systems, in most cases, require control parameters which are different to those faster-response radiators will need. Moreover, there are cases where the different control systems require different variables with different object widths (1-bit or 1-byte type). This would justify the use of a second controller.

The push sensor 2 plus offers the option to activate one or, alternatively, two control loops. The *"control loops"* parameter in the *"room temperature controller function"* parameter branch sets the number of control loops.

- Using one control loop:
If you use only one control loop you can parameterize the *"heating"*, *"cooling"* or, as an alternative, the mixed *"heating and cooling"* modes. You can also use additional stages in any cases.
In this connection, you can set different control algorithms for the heating and/or cooling system. Thus, you can use up to four separate algorithms for two-stage heating or cooling operation.

- Using two control loops:
If you use two control loops you can only choose between the *"heating"* or *"cooling"* mode. In this connection, both control loops will always work in the same operating mode (comfort, standby, etc.). however, you can set different control algorithms for both control loops. For this type of parameterization, the use of two-stage control is not intended. Both control loops can alternatively work with joint or with separate setpoints (refer to "4.4 Temperature setpoints").

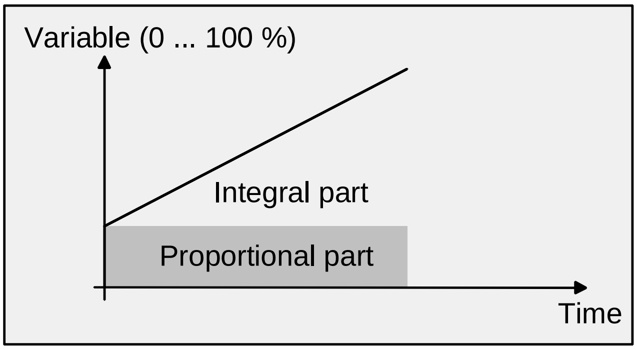
The variables calculated by the control algorithm are output via the *"heating variable"* or *"cooling variable"* communication objects. Depending on the control algorithm selected for the heating and/or cooling mode, the format of the variable objects is, among other things, also specified. So you can create 1-bit or 1-byte variable objects (refer to "4.3.3 Output of variables").

You can use the *"type of heating control"* or *"type of cooling control"* parameter in the *"room temperature controller function"* parameter branch to specify the control algorithm, if necessary, also for the additional stages or for both control loops, respectively.
In this connection, you can select each of the three following algorithms:

1. Continuous PI control

PI control is an algorithm which consists of a proportional and of an integral part. By the combination of these control properties, you can obtain room temperature control as fast and precise as possible without or only with low deviations.

When you use this algorithm, the room temperature controller will calculate a new continuous variable in cycles of 30 seconds and send it to the bus via a 1-byte value object if the calculated variable value has changed by a specified percentage. You can use the *"automatic transmission at modification by…"* parameter in the *"room temperature controller function - variable and status output"* parameter branch to set the change interval in per cent.



An additional heating or cooling stage as PI control works in the same way as the PI control of the basic stage, with the exception that the setpoint will shift, taking account of the parameterized difference between the basic and additional stages.

Special features of the PI control:

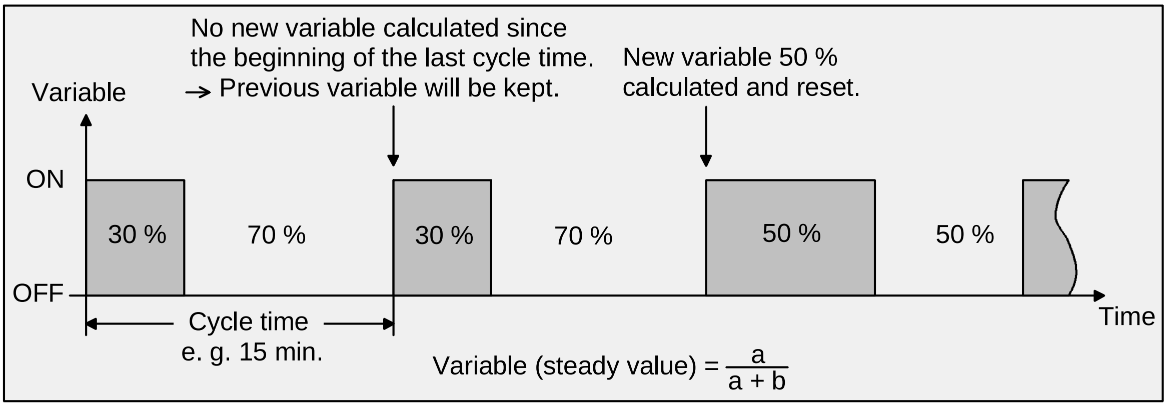
If the room temperature deviation between the actual value and the setpoint is high enough to have a 100 % variable the room temperature controller in the push sensor 2 plus will work with this maximum variable until the room temperature measured has reached its setpoint. This particular behaviour is known as 'clipping'.
This way, rapid heating up of undercooled rooms or quick cooling in overheated rooms will be achieved. In two-stage heating or cooling systems, this control behaviour also applies to the variables of the additional stages.

2. Switching PI control

For this parameterization the room temperature will also be kept constant by the PI control algorithm. Taking the mean value for a given time, the same behaviour of the control system will results a you would obtain with a continuous controller. The difference compared with continuous control is only the way how the variable is output. The variable calculated by the algorithm in cycles of every 30 seconds is internally converted into a pulse-width-modulated (PWM) variable signal and sent to the bus via a 1-bit switching object after the cycle time has elapsed. The mean value of the variable signal resulting from this modulation is a measure for the averaged position of the control valve, thus being a reference to the room temperature set, taking account of the cycle time which you can set through the *"cycle time of the switching variable …"* parameter in the *"room temperature controller function - variable and status output"* parameter branch.
A shift of the mean value, and thus a change in the heating capacity, can be obtained by changing the duty factor of the switch-on and switch-off pulses of the variable signal.

The duty factor will be adapted by the controller only at the end of a time period, depending on the variable calculated. This applies to any change of the variable, regardless of what the ratio is by which the variable changes (the *"automatic transmission at modification by …"* and *"cycle time for automatic transmission…"* parameters will have no function in this case). Each variable value calculated last during an active time period will be converted. Even after you have changed the setpoint temperature, for example, by changing the operating mode, the variable will still be adapted after the end of an active cycle time.

The illustration below shows the variable switching signal output in dependence on the internally calculated variable value (first of all, a variable of 30 %, then of 50 %, with the variable output not being inverted).


No new variable calculated since
the beginning of the last cycle time.
→ Previous variable will be kept.
New variable 50 %
calculated and reset.
Variable
ON
OFF
30 %
70 %
30 %
70 %
50 %
50 %
Time
Cycle time
e. g. 15 min.
Variable (steady value) = a / (a + b)

For a variable of 0 % (permanently OFF) or of 100 % (permanently ON), a variable telegram corresponding to the variable value ("0" or "1") will always be sent after a cycle time has elapsed. 'Clipping' (refer to "continuous PI control") will also be active for this type of control.

Also for switching PI control, the controller will always use continuous variable values for internal calculation. Such continuous values can additionally be sent to the bus via a separate 1-byte value object, for example, as status information for visualization purposes.
Therefore, when you use switching PI control (RWM), value object 51 will be sent for the heating mode, and value object 53 for the cooling mode. If you use additional stages value object 52 for additional heating operation and value object 49 for additional cooling operation will be additionally enabled. For the use of two control loops, the separate 1-byte value object will not be available.
If you want to output the heating and cooling variables through a common object (refer to "4.3.3 Output of variables") the continuous value of the activated operating mode will be transmitted via object 46 and, if necessary, via object 51 for the additional stages.
The status value object will be updated at the same time as the variable is output and will only take place after the parameterized cycle time has elapsed. The *"automatic transmission at modification by …"* and *"cycle time for automatic transmission…"* parameters will have no function in this case.

An additional heating or cooling stage as switching PI control works in the same way as the PI control of the basic stage, with the exception that the setpoint will shift, taking account of the parameterized difference between the basic and additional stages. All PWM control options will use the same cycle time.

Cycle time:

The pulse-width-modulated variables are mainly used for triggering electrothermal actuators (ETA). In this connection, the room temperature controller sends the switching variable telegrams to a switching actuator which the actuators are connected to.

By setting the cycle time of the PWM signal, you can adapt the control to the actuators used. The cycle time sets the switching frequency of the PWM signal and allows adaptation to the adjusting cycle times (the adjusting time it takes the actuator to bring the valve from its completely closed to its completely opened position) of the actuators used. In addition to the adjusting cycle time, take account of the dead time (the time in which the actuators do not show any response when being switched on or off). If different actuators with different adjusting cycle times are used take account of the longest of the times. Always note the information given by the manufacturers of the actuators.

Two different options of how to set the cycle time can be identified:

I. Cycle time > 2 x adjusting cycle time of the actuators used (ETA), e. g. 15 minutes (default)

In this case, the switch-on or switch-off times of the PWM signal are long enough for the actuators to have enough time to fully open or fully close within a given time period.

Advantages:
The desired mean value for the variable and thus for the required room temperature will be set relatively precisely, even for several actuators triggered at the same time.

Disadvantages:
It should be noted, that, due to the full valve lift to be continuously 'swept', the life expectancy of the actuators can diminish. For very long cycle times (> 15 minutes) with less sluggishness in the system, the heat emission into the room, for example, in the vicinity of the radiators, can possibly be non-uniform and be found disturbing.

Important:

- Such setting is recommended for slower, more sluggish heating systems (such as floor heating).
- Even for a bigger number of triggered actuators, maybe of different types, this setting can be recommended to be able to obtain a better mean value of the adjusting travels of the valves.

II. Cycle time > adjusting cycle time of the actuators used (ETA), e. g. 2 minutes

In this case, the switch-on or switch-off times of the PWM signal are too short for the actuators to have enough time to fully open or fully close within a given time period.

Advantages:
This setting ensures continuous water flow through the radiators, for example, thus facilitating uniform heat emission into the room.
If only one actuator is triggered the controller can continuously adapt the variable to compensate the mean value shift caused by the short cycle time, thus setting the desired room temperature.

Disadvantages:
If more than one actuator is triggered at the same time the desired mean value will become the variable, which will result in a very poor adjustment of the required room temperature, or to adjustment of the latter with major deviations, respectively.

Note:

- Such cycle time setting is recommended for less sluggish heating systems (such as radiators).

3. Switching 2-point control:

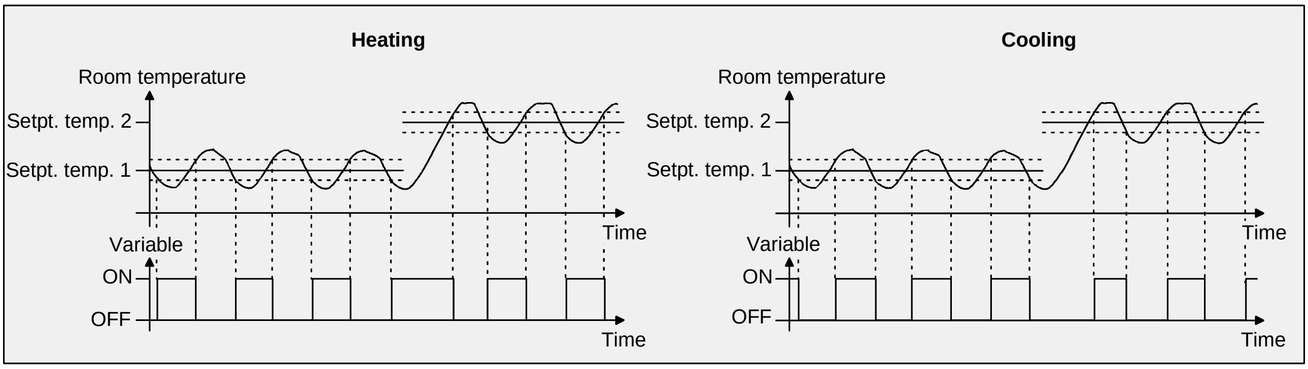
The 2-point control represents a very simple type of temperature control. For this type of control, two hysteresis temperature values are preset. The actuators are triggered by the controller via switch-on and switch-off variable commands (1-bit type). A continuous variable is not calculated for this type of control. The room temperature is also evaluated by this type of control in cycles every 30 seconds, i. e. the variables will only change at these moments, if required.
The disadvantage of a continuously varying temperature as a result of this option is in contrast with the advantage of this very simple 2-point room temperature control. For this reason, sluggish heating or cooling systems should not be triggered by a 2-point control system, for this can lead to very high overshooting of the temperature, thus resulting in loss of comfort.

When presetting the hysteresis limits, you should distinguish between the following operation modes:

- *"Heating"* or *"cooling"* single modes.

In the heating mode, the controller will turn on the heating system when the room temperature has fallen below a preset limit. The control system will only turn off the heating once a preset temperature limit has been exceeded.
In the cooling mode, the controller will turn on the cooling system when the room temperature has exceeded a preset limit. The control system will only turn off the cooling system once the temperature has fallen below a preset limit.
In this connection, variable "1" or "0" will be output, depending on the switching status, if the temperature exceeds or falls below the hysteresis limits.
Please note that the " ♨ " or " ❄ " symbol will appear in the display, or that the heating or cooling signal objects will already become active once the temperature is falling below the setpoint of the active operating mode for heating or is exceeding that for cooling. The hysteresis will be disregarded in this case.
You can parameterize the upper or lower hysteresis limit of the two operating modes in the ETS plug-in.

The following illustration shows a 2-point control example for the *"heating"* or *"cooling"* single modes (heating on the left and cooling on the right; two temperature setpoints; single-stage heating or cooling; non-inverted variable output).

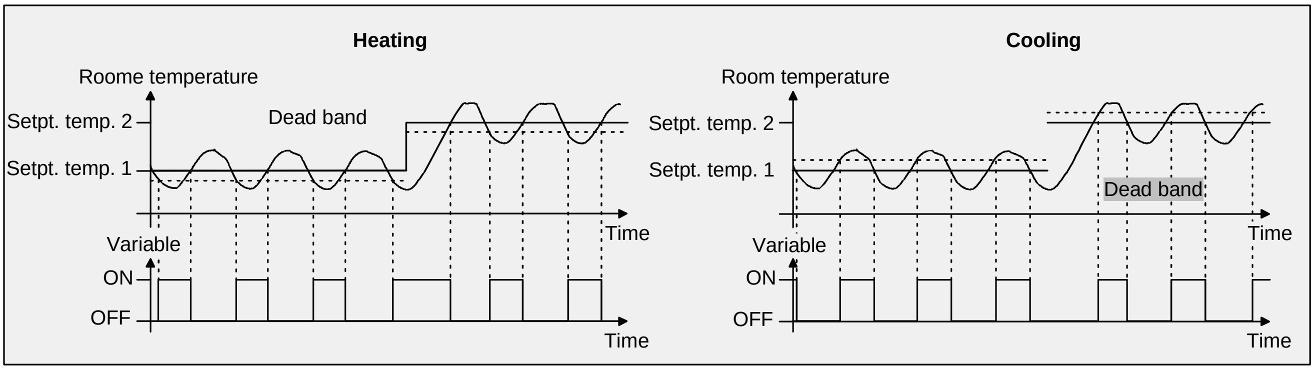


An additional heating or cooling stage as 2-point control works in the same way as the 2-point control of the basic stage, with the exception that the setpoint and the hysteresis values will shift, taking account of the parameterized difference between the basic and additional stages.

- *"Heating and cooling"* mixed operating mode:

  In the heating mode, the controller will turn on the heating when the room temperature has fallen below a preset limit. As soon as the room temperature is exceeding the setpoint of the current operating mode, the control will turn off the heating in the heating mode.
  In the cooling mode, the controller will turn on the cooling system when the room temperature has exceeded a preset limit. As soon as the room temperature is falling below the setpoint of the current operating mode, the control will turn off the cooling system in the cooling mode.
  Thus, there is no longer any upper hysteresis limit for heating or no lower one for cooling, respectively, for these values would be in the dead band. Within the dead band, neither heating nor cooling will take place.
  In this connection, variable "1" or "0" will be output, depending on the switching status, if the temperature exceeds or falls below the hysteresis limits or the setpoints, respectively.
  Please note that the " ♨ " or " ❄ " symbol will appear in the display, or that the heating or cooling signal objects will already become active once the temperature is falling below the setpoint of the active operating mode for heating or is exceeding that for cooling. The hysteresis will be disregarded in this case.
  You can parameterize the upper or lower hysteresis limit of the two operating modes in the ETS plug-in.

The following illustration shows a 2-point control example for the *"heating"* and *"cooling"* mixed mode (activated heating on the left and activated cooling on the right; two temperature setpoints; single-stage heating or cooling; non-inverted variable output).



An additional heating or cooling stage as 2-point control works in the same way as the 2-point control of the basic stage, with the exception that the setpoint and the hysteresis values will shift, taking account of the parameterized difference between the basic and additional stages.

4.3.2 Adapting the control algorithms

Adapting PI control
There are various installation systems which can heat or cool a room. By the use of heat transfer media (preferably water or oil) in conjunction with room air convection, it is possible to uniformly heat or cool the environment. Such systems are, for example, used in heating panels, floor heating systems or cooling ceilings.
As an alternative, or in addition, fan systems can heat or cool rooms. In most cases, these are electric fan heating or cooling systems or refrigeration compressors with fans. Due to the direct heating or cooling of the room air, such heating or cooling systems are rather quick.

To enable the PI control algorithm to efficiently control any conventional heating or cooling systems so that room temperature control works as fast as possible and without any deviations it will be necessary to adjust the control parameters.
For this purpose, some factors which sometimes influence the control behaviour to a considerable extent can be set for a PI control system. For this reason, you can set the room temperature controller to pre-defined 'experience values' for the most common heating or cooling systems. If the preset values do not give you any satisfactory control result for the corresponding heating or cooling system you have chosen, you can optionally use some control parameters to optimize adaptation.

You can use the *"type of heating"* or *"type of cooling"* parameters to set pre-defined control parameters for the heating or cooling stage and, if necessary, for the additional stages. These fixed values correspond to field-proven values of a properly planned and installed air conditioning system and will result in an optimum temperature control behaviour. The following can be set for heating or cooling operation, respectively:

For heating control				
	Preset Values		**Recommended**	**Recommended**
Type of Heating	**Proportional band**	**Reset Time**	**PI Control Type**	**PWM Cycle Time**
• Warm water heating	5 Kelvin	150 minutes	Contin./PWM	15 min.
• Floor heating	5 Kelvin	240 minutes	PWM	15 - 20 min.
• Electric 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				
	Preset Values		**Recommended**	**Recommended**
Type of Cooling	**Proportional band**	**Reset Time**	**PI Control Type**	**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.

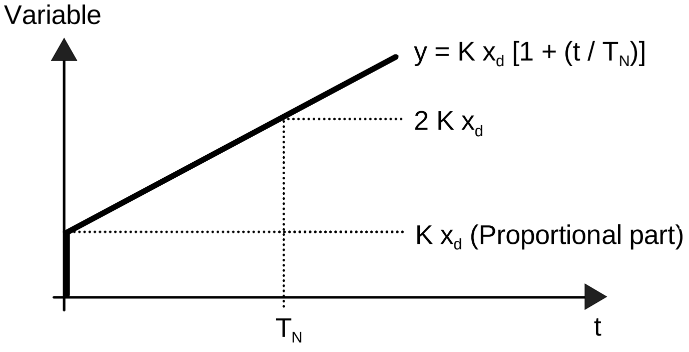
*: split, mobile air conditioning unit

**: For smaller, quicker radiators, PWM cycle time 2 - 3 minutes.

If you have set the *"type of heating"* or *"type of cooling"* parameters to *"via control parameter"* you can 'manually' match the control parameters. You can preset the proportional band for heating or cooling (P part) and the reset time for heating or cooling (I part) to influence the control to a considerable extent.

Important:

- Even varying the control parameters by low amounts will lead to a clearly different control behaviour.
- Setting the control parameters of the corresponding heating or cooling system in accordance with the above-mentioned fixed values should be the starting point for adaptation.



x_d : control difference $x_d = x_{set} - x_{act}$
$P = 1/K$: parameterizable proportional band
$K = 1/P$: gain
T_N : parameterizable reset time

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

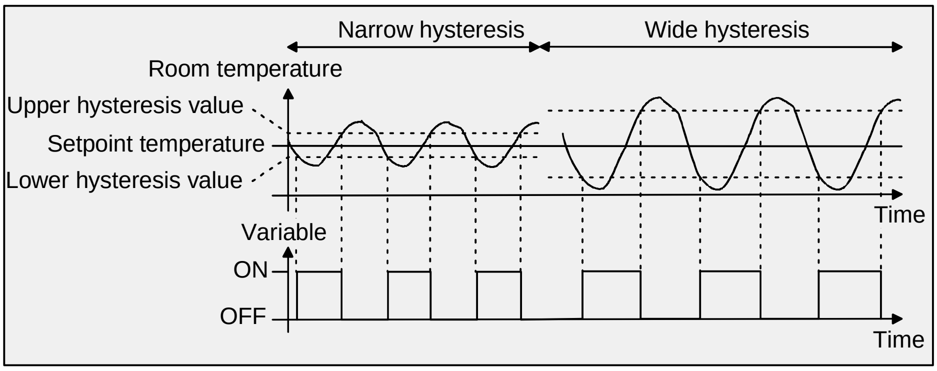
P control algorithm: variable $y = K x_d$

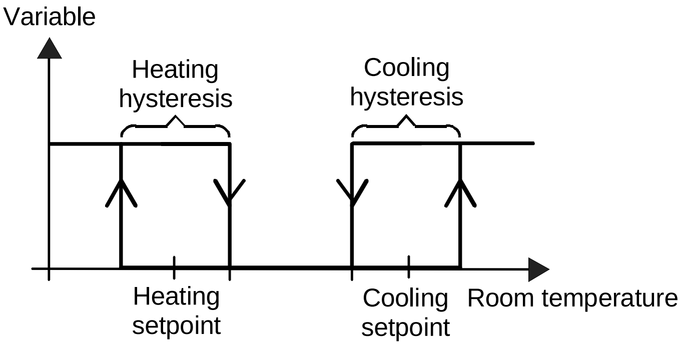
Parameter Setting		**Effect**
P	Narrow proportional band	High overshooting upon setpoint changes (possibly even permanent oscillation), quick adjustment on setpoint.
P	Wide proportional band	No (or low) overshooting, but slowly adjusting.
T_N	Short reset time	Quick compensation of deviations (ambient conditions), risk of permanent oscillation.
T_N	Long reset time	Slow compensation of deviations.

4.3.2.1 Adapting the 2-point control:

The 2-point control represents a very simple type of temperature control. For this type of control, two hysteresis temperature values are preset.
You can use parameters to set the upper and lower temperature hysteresis limits. Please take into consideration that:

- a narrow hysteresis will lead to low temperature fluctuations but to a higher bus load.
- a wide hysteresis will switch less frequently but will cause more inconvenient temperature fluctuations.





4.3.3 Output of variables

4.3.3.1 Variables objects

Depending on the control algorithm selected for the heating and/or cooling mode and, if necessary, also for the additional stages, the format of the variables objects is set. Therefore, you can create 1-bit or 1-byte variables objects. The control algorithm calculates the variables at time intervals of 30 seconds and outputs them. For pulse-width-modulated PI control (PWM), the variable is only updated at the end of a time cycle, if required.

Possible object data formats for the variables, separately for both operating modes, for the basic and additional stages, or for both control loops, are…

- continuous PI control: 1-byte type,
- switching PI control: 1-bit type + additionally 1 byte (e. g. for status indication in connection with visualization),
- switching 2-point control: 1-bit type.

Depending on the operating mode selected, the controller can trigger heating and/or cooling systems and determine variables and output them via separate objects. In the *"heating and cooling"* mixed mode, distinction is made between two different options:

Option 1: The heating and the cooling system are two separate systems.
In this case, you should set the *"send variable heating and cooling to one common object"* parameter in the *"room temperature controller functions"* parameter branch to *"no"* (default). Thus, separate objects per variable will be available through which the individual systems can be triggered separately. This setting will enable you to define separate types of control for heating or cooling.

Option 2: The heating and cooling systems are one combined unit.
In this case, you can set the *"send variable heating and cooling to one common object"* parameter in the *"room temperature controller functions"* parameter branch to *"yes"*, if required. Thus, the variables for heating and cooling will be sent to the same object. For two-stage control, another common object will be enabled for the heating and cooling additional stages.
If you use this setting you can only define the same type of control for heating and cooling as the type of control and the data format must be identical in this case. The control parameters (*"type of heating/ cooling"*) must still be parameterized separately for heating or cooling operation.
A combined variables object can, for example, become necessary if both heating and cooling should be effected through a one-pipe system (combined heating and cooling system). For this purpose, the temperature of the medium in the one-pipe system must, first of all, be changed by the control system of the installation. Subsequently, you can select heating/cooling switch-over (often, cold water is used in the one-pipe system for cooling in summer, while hot water is used for heating in winter).

Note:

Heating and cooling at the same time (variables > "0") is, in general, not possible.

If required, you can invert the variable prior to transmission. By the *"output of the heating variable"* or *"output of the cooling variable"* parameter, or in case of output through a combined *"variable output"* object, the variable value will be output in inverted form in accordance with the object data format. When two-stage control is used the parameters for inverting the additional stage(s) will also be available.
Where:

for continuous variables:	not inverted:	variable 0 % ... 100 %,	value 0 .. 255,
	inverted:	variable 0 % ... 100 %,	value 255 ... 0,
for switching variables:	non-inverted:	variable OFF/ON,	value 0 / 1,
	inverted:	variable OFF/ON,	value 1 / 0.

4.3.3.2 Automatic transmission

- Continuous PI control:

For continuous PI control, the room temperature controller cyclically calculates a new variable every 30 seconds and transmits it to the bus through a 1-byte value object. In this connection, you can use the *"automatic transmission at modification by…"* parameter in the *"room temperature function - variable and status output"* parameter branch to set the variable change interval after which a new variable should be sent to the bus. You can parameterize the change interval to "0" so that no automatic transmission will take place whenever the variable changes.

In addition to the output of the variable upon a change of the latter, the current variable value can be cyclically sent to the bus. In this connection, further variable telegrams will be released after a parameterized cycle time in addition to the expected times of change, according to the active value.
This is to ensure that, if the variable undergoes cyclic safety monitoring, telegrams will be received in the actuator or in the triggered switching actuator within the monitoring time. The time interval set by the *"cycle time for automatic transmission…"* parameter should correspond to the monitoring time in the actuator (prefer to parameterize a shorter cycle time in the controller).
Setting "0" will deactivate the cyclic transmission of the variable.

Please note for continuous PI control that no more variable telegrams will be sent upon a change if you have deactivated cyclic transmission and switched off automatic transmission.

- Switching PI control (PWM):

For switching PI control (PWM), the room temperature controller will also internally calculate a new variable every 30 seconds. However, updating the variable with this type of control will only take place at the end of a time cycle, if required. The *"automatic transmission at modification by…"* and *"cycle time for automatic transmission…"* parameters will have no function in connection with this control algorithm.

- 2-point control:

For 2-point control, the room temperature and thus the hysteresis values will be evaluated cyclically every 30 seconds so that the variable, if required, will only change at these intervals. As no continuous variables will be calculated for this control algorithm the *"automatic transmission at modification by…"* parameter will show no effect in connection with this control algorithm.

In addition to the output of the variable upon a change of the latter, the current variable value can be cyclically sent to the bus. In this connection, further variable telegrams will be released after a parameterized cycle time in addition to the expected times of change, according to the active value.
This is to ensure that, if the variable undergoes cyclic safety monitoring, telegrams will be received in the actuator or in the triggered switching actuator within the monitoring time. The time interval set by the *"cycle time for automatic transmission…"* parameter should correspond to the monitoring time in the actuator (prefer to parameterize a shorter cycle time in the controller).
Setting "0" will deactivate the cyclic transmission of the variable.

4.4 Temperature setpoints

4.4.1 Setpoint assignment in the ETS

You can preset temperature setpoints for each operating mode. In the ETS plug-in, you can parameterize the setpoints for the "comfort ⌂", "standby ⌂" and "economy ☾" modes. If desired, you can adapt the setpoint temperatures later during running operation by local actuation of the controller at operator level 2 or by control via objects. For the "frost/heat protection ❄" mode, you can separately parameterize two temperature setpoints only in the ETS, one for heating (frost protection) and one for cooling operation (heat protection).

When presetting the setpoint temperatures for the comfort, standby and economy modes, always make sure that all setpoints have a fixed relation with one another, for all values are derived from the basic temperature (basic setpoint). The *"basic temperature after reset"* parameter in the *"setpoints"* parameter branch sets the basic setpoint which will be loaded as presetting when the device is being programmed by the ETS.
From this value, the temperature setpoints for the standby and economy modes will be derived, taking account of the *"lower/raise the setpoint temperature during standby operation"* or *"lower/raise the setpoint temperature during economy operation"*, depending on the heating or cooling mode. For the "heating and cooling" mode, the dead band will also be taken into account.

In two-stage control operation, all setpoint temperatures of the additional stage will be derived from the setpoint temperatures of the basic stage. In this connection, to determine the setpoint temperatures of the additional stage the *"difference between basic and additional stages"* firmly parameterized in the ETS plug-in will be deducted from the setpoints of the basic stage for heating operation or added to them for cooling operation. If the temperature setpoints of the basic stage are, for example, changed on the push sensor at operator level 2 or by presetting a new basic setpoint, the setpoint temperatures of the additional stage will automatically change indirectly at the same time. At a setpoint difference of "0", both stages will be heating or cooling at the same time, using the same variable.

If you use two control loops you can preset common setpoints or, alternatively, separate values for both control loops. In this connection, the *"own setpoints for second control loop"* parameter in the *"room temperature controller function - setpoints"* parameter branch will preset the setpoints:

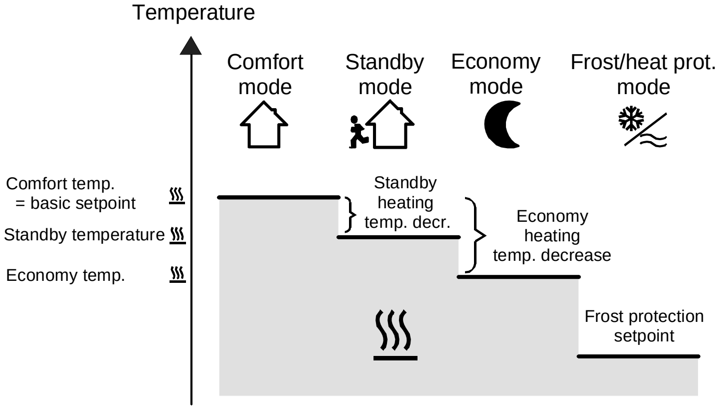
- Setting *"no"* (default):
  Both control loops have the same setpoints for the comfort, standby and economy modes. The frost and heat protection temperatures are also identical. If enabled, this setting has one common object for presetting the basic setpoint and one object for the transmission of the setpoint temperature to the bus.

- Setting *"yes"*:
  Both control loops have their own separate setpoints for the comfort, standby and economy modes. Only the frost and heat protection temperatures are identical. If enabled, this setting has separate objects per control loop for presetting the basic setpoint or for the transmission of the setpoint temperature to the bus, respectively. You can only change the setpoint temperature for the first control loop at operator level 2 on the push sensor.

The operating mode switch-over of the second control loop always proceeds in parallel with the first control loop switch-over. If you use two control loops two-stage control and mixed heating and cooling operation will not be possible.

Depending on the heating/cooling switch-over, please note the relations shown on the next pages when presetting the setpoints.
If you use two control loops you can set the heating/cooling switch-over for both control loops either to *"heating"* or to *"cooling".* Mixed *"heating and cooling"* operation will not be possible in this connection.

4.4.1.1 *"Heating"* mode setpoints:


Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Comfort temp. = basic setpoint
Standby temperature
Economy temp.
Standby heating temp. decr.
Economy heating temp. decrease
Frost protection setpoint

(Recommended setpoint temperature value presettings)

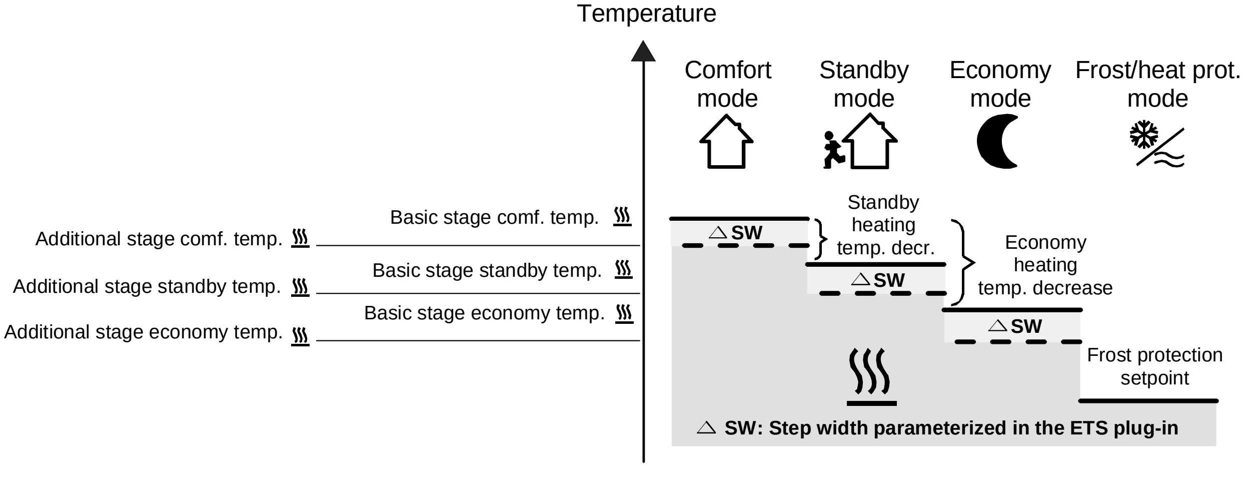
In this mode, the comfort, standby and economy setpoint temperatures are given, and you can preset the frost protection temperature. Where:

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

The standby and economy setpoint temperatures are derived from the comfort setpoint temperature (basic setpoint) in accordance with the parameterized decreasing temperatures. On the controller at operator level 2, if enabled, you can locally set even other decrease temperatures by changing the economy and standby setpoint temperature values (refer to "1.5 Operator levels/Local operation). Such local operation will only be possible for control loop 1.

Frost protection is intended to prevent the heating system from freezing. For this reason, you should set the frost protection temperature lower than the economy temperature (default: +7 °C).However, you can, in general, select values between +7 °C and +40 °C as frost protection temperature.
For "heating", the possible value range is between +7.0 °C and +99.9 °C and is restricted by the frost protection temperature in the lower range.

For two-stage heating operation, the difference between the basic and additional stages parameterized in the ETS plug-in will also be taken into account.

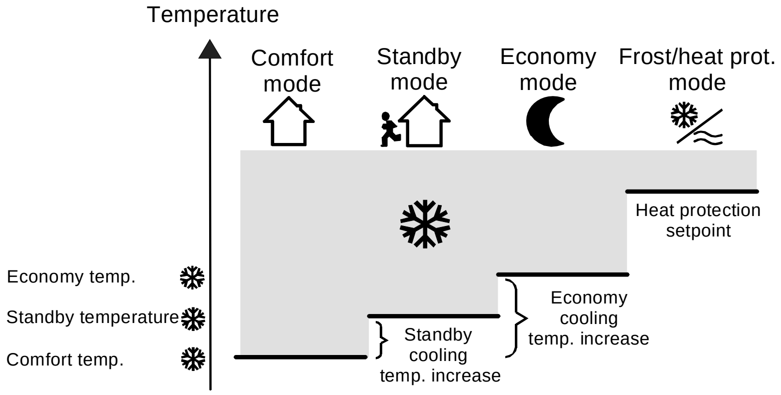

Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Basic stage comf. temp.
Additional stage comf. temp.
Basic stage standby temp.
Additional stage standby temp.
Basic stage economy temp.
Additional stage economy temp.
△SW
Standby heating temp. decr.
△SW
Economy heating temp. decrease
△SW
Frost protection setpoint
△ SW: Step width parameterized in the ETS plug-in

$$T_{\text{comfort heating add. st. setpt.}} \leq T_{\text{comfort heating basic st. setpt.}} \;/\; T_{\text{standby heating add. st. setpt.}} \leq T_{\text{standby heating basic st. setpt.}}$$
$$T_{\text{standby heating setpoint}} \leq T_{\text{comfort heating setpoint}}$$

or

$$T_{\text{comfort heating add. st. setpt.}} \leq T_{\text{comfort heating basic st. setpt.}} \;/\; T_{\text{economy heating add. st. setpt.}} \leq T_{\text{economy heating basic st. setpt.}}$$
$$T_{\text{economy heating setpoint}} \leq T_{\text{comfort heating setpoint}}$$

4.4.1.2 *"Cooling"* mode setpoints:



(Recommended setpoint temperature value presettings)

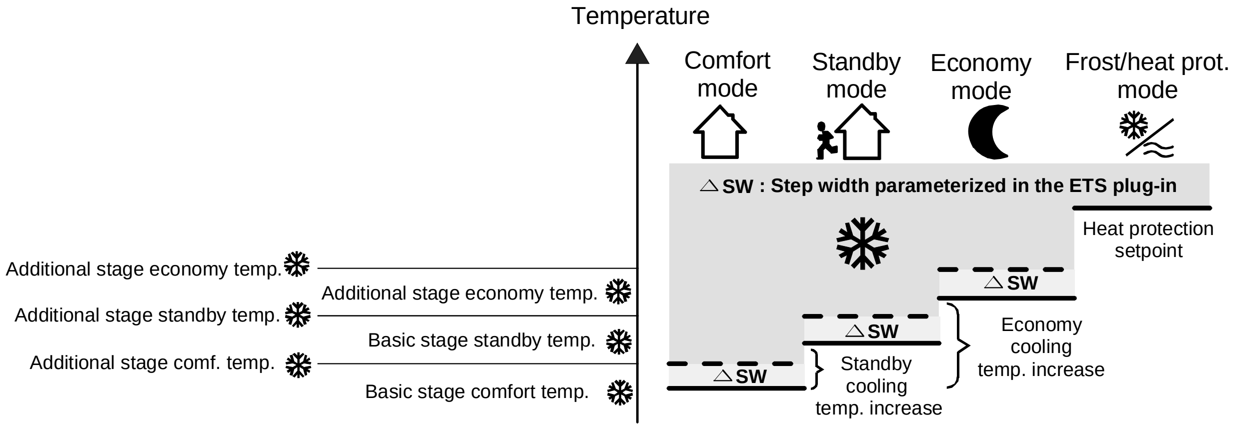
In this mode, the comfort, standby and economy setpoint temperatures are given, and you can preset the heat protection temperature. Where:

$$T_{\text{comfort cooling setpoint}} \leq T_{\text{standby cooling setpoint}} \quad \text{or} \quad T_{\text{comfort cooling setpoint}} \leq T_{\text{economy cooling setpoint}}$$

The standby and economy setpoint temperatures are derived from the comfort setpoint temperature (basic setpoint) in accordance with the parameterized increasing temperatures. On the controller at operator level 2, if enabled, you can locally set even other increasing temperatures by changing the economy and standby setpoint temperature values (refer to "1.5 Operator levels/Local operation). Such local operation will only be possible for control loop 1.

Heat protection is intended to prevent a maximum permissible room temperature from being exceeded in order to protect parts of the installation, if necessary. For this reason, you should set the heat protection temperature higher than the economy temperature (default: +35 °C).However, you can, in general, select values between +7 °C and +45 °C as heat protection temperature.
For "cooling", the possible value range is between -99.9 °C and +45.0 °C and is restricted by the heat protection temperature in the upper range.

For two-stage cooling operation, the difference between the basic and additional stages parameterized in the ETS plug-in will also be taken into account.


Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
△SW : Step width parameterized in the ETS plug-in
Heat protection setpoint
△SW
△SW
△SW
Economy cooling temp. increase
Standby cooling temp. increase
Additional stage economy temp.
Additional stage standby temp.
Additional stage comf. temp.
Additional stage economy temp.
Basic stage standby temp.
Basic stage comfort temp.

$$T_{\text{comfort cooling basic. st. setpt.}} \leq T_{\text{comfort cooling add. st. setpt.}} \;/\; T_{\text{standby cooling basic . st. setpt.}} \leq T_{\text{standby cooling add. st. setpt.}}$$

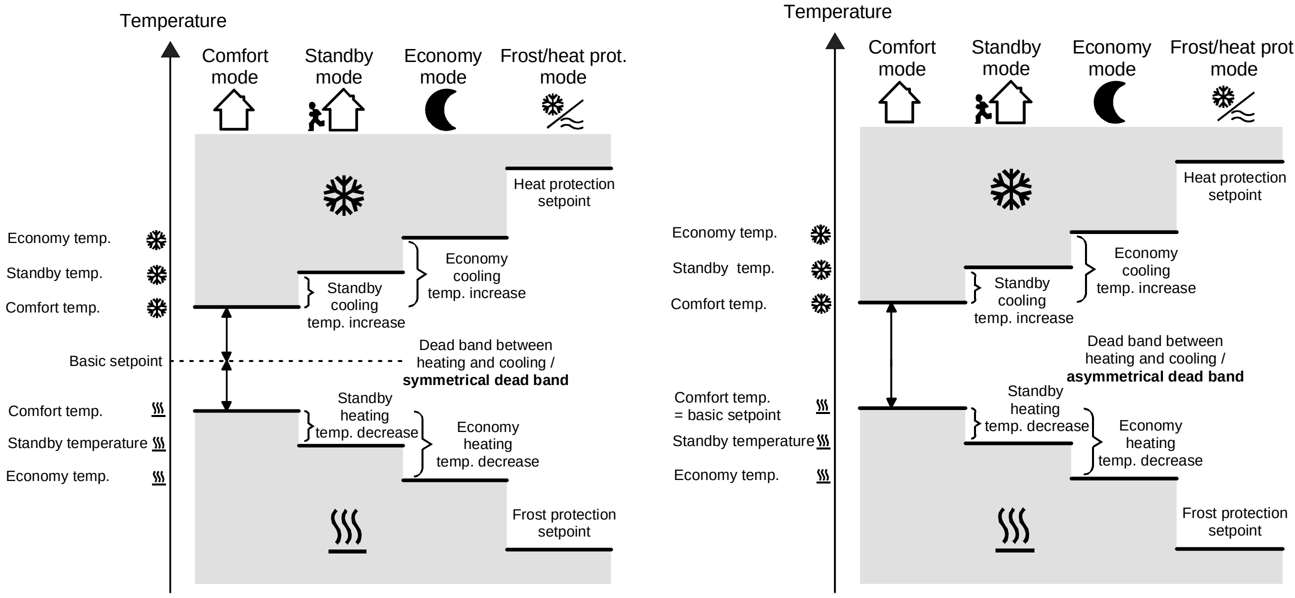
$$T_{\text{comfort cooling setpoint}} \leq T_{\text{standby cooling setpoint}}$$

or

$$T_{\text{comfort cooling basic. st. setpt.}} \leq T_{\text{comfort cooling add. st. setpt.}} \;/\; T_{\text{economy cooling basic . st. setpt.}} \leq T_{\text{economy cooling add. st. setpt.}}$$

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

4.4.1.3 *"Heating and Cooling"* mode setpoints:


Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Heat protection setpoint
Economy temp.
Standby temp.
Comfort temp.
Economy cooling temp. increase
Standby cooling temp. increase
Basic setpoint
Dead band between heating and cooling /
symmetrical dead band
Comfort temp.
Standby temperature
Economy temp.
Standby heating temp. decrease
Economy heating temp. decrease
Frost protection setpoint
Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Heat protection setpoint
Economy temp.
Standby temp.
Comfort temp.
Economy cooling temp. increase
Standby cooling temp. increase
Dead band between heating and cooling /
asymmetrical dead band
Comfort temp. = basic setpoint
Standby temperature
Economy temp.
Standby heating temp. decrease
Economy heating temp. decrease
Frost protection setpoint

(Recommended setpoint temperature value presettings)

In this mode, the comfort, standby and economy setpoint temperatures of both operating modes and of the dead band are given. In addition, you can preset the frost and the heat protection temperatures. Where:

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

or

$$T_{\text{economy heating setpoint}} \le T_{\text{comfort heating setpoint}} \le T_{\text{comfort cooling setpoint}} \le T_{\text{economy cooling setpoint}}$$

The standby and economy setpoint temperatures are derived from the comfort heating or comfort cooling setpoint temperatures. In this connection, you can assign the temperature increase (for cooling) and the temperature decrease (for heating) of both operating modes in the ETS plug-in. The comfort temperatures themselves are derived from the dead band and the basic setpoint (refer to "dead band" on the next page).

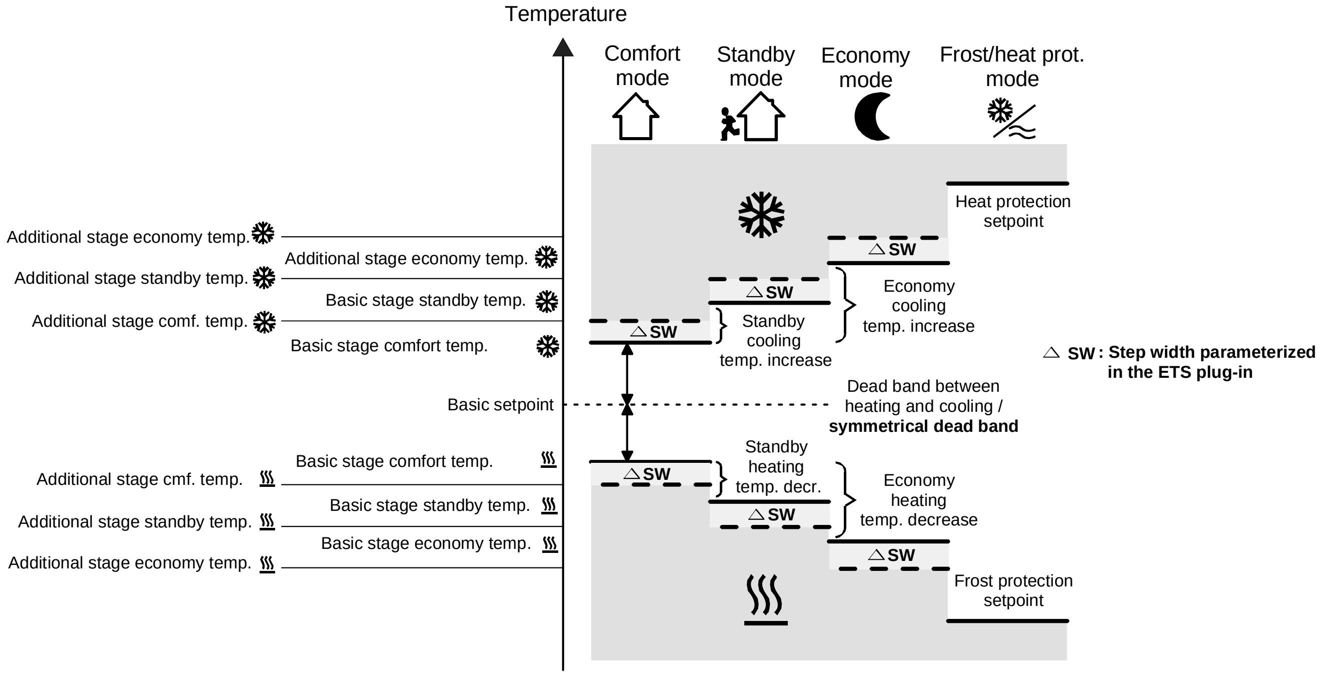
Note:

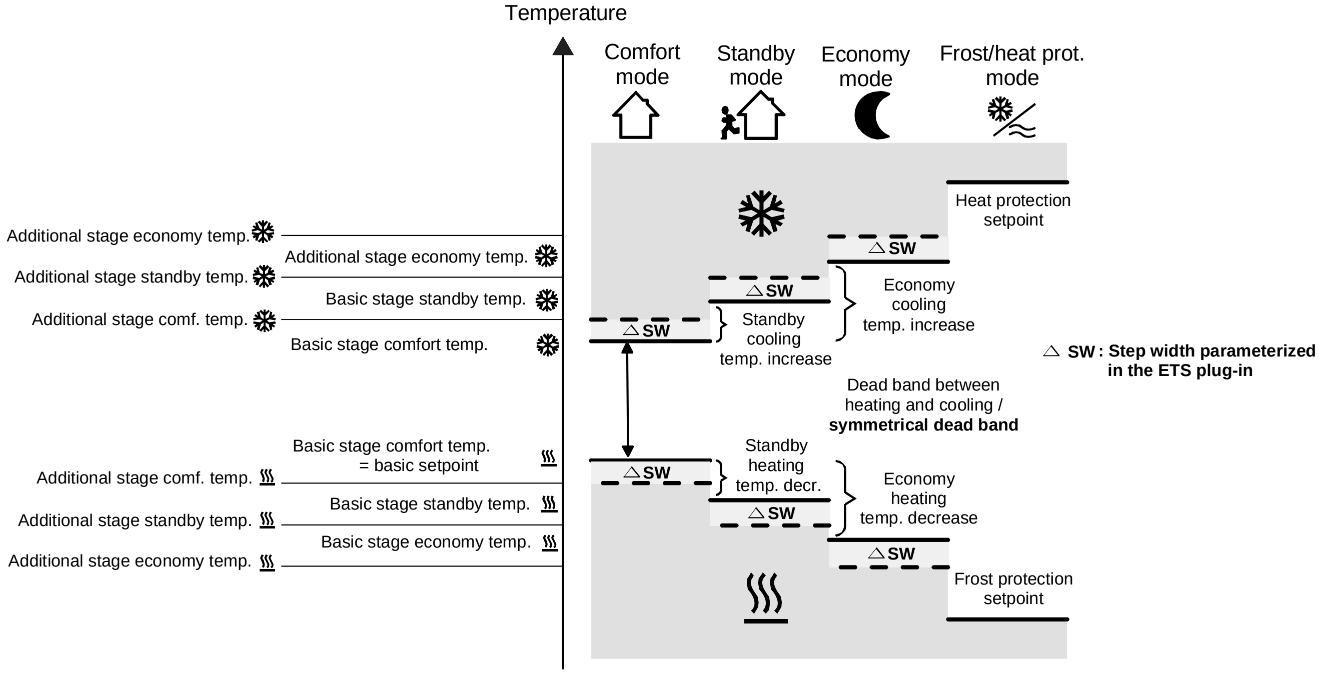
On the controller at operator level 2, if enabled, you can locally set even other increase or decrease temperatures by changing the economy and standby setpoint temperature values (refer to "1.5 Operator levels/Local operation). Such local operation will only be possible for control loop 1.

Frost protection is intended to prevent the heating system from freezing. For this reason, you should set the frost protection temperature lower than the economy heating temperature (default: +7 °C).However, you can, in general, select values between +7 °C and +40 °C as frost protection temperature.
Heat protection is intended to prevent a maximum permissible room temperature from being exceeded in order to protect parts of the installation, if necessary. For this reason, you should set the heat protection temperature higher than the economy cooling temperature (default: +35 °C). However, you can, in general, select values between +7 °C and +45 °C as heat protection temperature.

For "heating and cooling", the possible value range of a setpoint temperature is between +7 °C and +45.0 °C and is restricted by the frost protection temperature in the lower range as well as by the heat protection temperature in the upper range.

For two-stage heating or cooling operation, the difference between the basic and additional stages parameterized in the ETS plug-in will also be taken into account.


Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Heat protection setpoint
Additional stage economy temp.
Additional stage economy temp.
Additional stage standby temp.
Basic stage standby temp.
Additional stage comf. temp.
Basic stage comfort temp.
△SW
△SW
△SW
Standby cooling temp. increase
Economy cooling temp. increase
△ SW : Step width parameterized in the ETS plug-in
Basic setpoint
Dead band between heating and cooling /
symmetrical dead band
Basic stage comfort temp.
Additional stage cmf. temp.
Basic stage standby temp.
Additional stage standby temp.
Basic stage economy temp.
Additional stage economy temp.
△SW
△SW
△SW
Standby heating temp. decr.
Economy heating temp. decrease
Frost protection setpoint


Temperature
Comfort mode
Standby mode
Economy mode
Frost/heat prot. mode
Heat protection setpoint
Additional stage economy temp.
Additional stage economy temp.
Additional stage standby temp.
Basic stage standby temp.
Additional stage comf. temp.
Basic stage comfort temp.
△SW
△SW
△SW
Standby cooling temp. increase
Economy cooling temp. increase
△ SW : Step width parameterized in the ETS plug-in
Dead band between heating and cooling /
symmetrical dead band
Basic stage comfort temp. = basic setpoint
Additional stage comf. temp.
Basic stage standby temp.
Additional stage standby temp.
Basic stage economy temp.
Additional stage economy temp.
△SW
△SW
△SW
Standby heating temp. decr.
Economy heating temp. decrease
Frost protection setpoint

$$T_{\text{comf. heating add. st. setpt.}} \leq T_{\text{comf. heating basic st. setpt.}} \leq T_{\text{comf. cooling basic. st. setpt.}} \leq T_{\text{comf. cooling add. st. setpt.}}$$
$$T_{\text{standby heating add. st. setpt.}} \leq T_{\text{standby heating basic st. setpt.}} \leq T_{\text{standby cooling basic. st. setpt.}} \leq T_{\text{standby cooling add. st. setpt.}}$$
$$T_{\text{standby heating setpoint}} \leq T_{\text{comfort heating setpoint}} \leq T_{\text{comfort cooling setpoint}} \leq T_{\text{standby cooling setpoint}}$$

or

$$T_{\text{comf. heating add. st. setpt.}} \leq T_{\text{comf. heating basic st. setpt.}} \leq T_{\text{comf. cooling basic. st. setpt.}} \leq T_{\text{comf. cooling add. st. setpt.}}$$
$$T_{\text{economy heating add. st. setpt.}} \leq T_{\text{economy heating basic st. setpt.}} \leq T_{\text{economy cooling basic. st. setpt.}} \leq T_{\text{economy cooling add. st. setpt.}}$$
$$T_{\text{economy heating setpoint}} \leq T_{\text{comfort heating setpoint}} \leq T_{\text{comfort cooling setpoint}} \leq T_{\text{economy cooling setpoint}}$$

Dead band:

The comfort heating and cooling setpoint temperatures are derived from the basic setpoint, taking account of the preset dead band. The dead band (temperature zone where neither heating nor cooling takes place) is the difference between the comfort setpoint temperatures.
In the ETS plug-in, you can set the *"dead band between heating and cooling"*, *"dead band position"* and *"basic temperature after reset"* parameters. In this connection, distinction is made between the following settings:

Dead band position = *"symmetrical"* (default):
The dead band preset in the ETS plug-in is divided into two parts at the basic setpoint. From the resulting half of the dead band, the comfort setpoint temperatures are directly derived from the basic setpoint. Where:

$T_{\text{basic setpoint}} - \tfrac{1}{2}T_{\text{dead band}} = T_{\text{comfort heating setpoint}}$ or $T_{\text{basic setpoint}} + \tfrac{1}{2}T_{\text{dead band}} = T_{\text{comfort cooling setpoint}}$

➔ $T_{\text{comfort cooling setpt.}} - T_{\text{comfort heating setpt.}} = T_{\text{dead band}}$; $T_{\text{comfort cooling setpt.}} \geq T_{\text{comfort heating setpt.}}$

Important for a symmetrical dead band:

- For a symmetrical dead band, local operation at operator level 2 will indirectly set the basic setpoint through the comfort heating temperature. For this reason, the basic setpoint will not be shown in the display.
- By changing the comfort cooling setpoint temperature, you can alter the dead band through local operation, if enabled (dead band shift). When you alter the dead zone you can expect a shifting of the comfort heating setpoint temperature and thus of all the other temperature setpoints if the dead band position is symmetrical. You can set the dead band to 0 °C (result: $T_{\text{comfort cooling setpoint}} = T_{\text{comfort heating setpoint}}$). In such case, neither heating nor cooling will take place if the room temperature determined is equal to the comfort setpoint temperatures.

Dead band position = *"asymmetrical"*:
For this setting, the comfort heating setpoint temperature is equal to the basic setpoint.
The dead band set in the ETS plug-in will only become effective from the basic setpoint towards the comfort cooling temperature. Thus, the comfort mode cooling setpoint temperature is directly derived from the comfort heating setpoint. Where:

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

➔ $T_{\text{comfort cooling setpt.}} - T_{\text{comfort heating setpt.}} = T_{\text{dead band}}$; $T_{\text{comfort cooling setpt.}} \geq T_{\text{comfort heating setpt.}}$

Important for an asymmetrical dead band:

- By changing the comfort cooling setpoint temperature, you can alter the dead band through local operation, if enabled (dead band shift). For an asymmetrical dead band position, only the cooling temperature setpoints will be changed if you change the comfort cooling setpoint temperature. You can set the dead band to 0 °C (result: $T_{\text{comfort cooling setpoint}} = T_{\text{comfort heating setpoint}}$). In such case, neither heating nor cooling will take place if the room temperature determined is equal to the comfort setpoint temperatures.

4.4.2 Changing the setpoints

4.4.2.1 Changing the basic temperature setpoint temperatures for the comfort, standby and economy modes

When presetting the setpoint temperatures for the comfort, standby and economy modes, always make sure that all setpoints have a fixed relation with one another, for all values are derived from the basic temperature (basic setpoint). The *"basic temperature after reset"* parameter in the *"setpoints"* parameter branch sets the basic setpoint which will be loaded as presetting when the device is being programmed by the ETS.

By local operation on the controller at operator level 2, or controlled by the *"basic set value"* object, it is possible to 'subsequently' change or reset the setpoint temperatures. After programming, the setpoint temperatures of the second control loop can only be adapted in an object-controlled way.

Any change must always be enabled in the ETS plug-in in the *"setpoints"* parameter branch. In this connection, it is possible…

- to allow the *"modification of the basic temperature setpoint"* by directly changing the comfort heating temperature on the device (locally; only for control loop 1) and/or by setting a new basic setpoint via the bus (object 31 for control loop 1/object 32 for control loop 2).
- to facilitate the *"standby temperature change"* by directly changing the standby heating or cooling temperatures of the first control loop only on the device (locally).
- to allow the *"economy temperature change"* by directly changing the economy heating or cooling temperatures of the first control loop only on the device (locally), and
- to enable the *"dead band shift"* by changing the comfort cooling setpoint temperature only on the device (locally; control loop 1 for "heating and cooling").

If changing is not enabled (setting: *"deactivated"*), 'subsequent' resetting of the value preset by the ETS will not be possible. In this case, local setting of the corresponding temperature value will be 'rejected' (you cannot go to the setting mode of the selected value). In case of an unadmitted basic setpoint resetting via the bus, object 26 or 27 will be hidden.

Changing the basic setpoint/the convenient heating temperature:

Only when the basic setpoint is changed (by local operation and/or by the object) you can distinguish between two options:

- Option 1: The basic setpoint change will be accepted <u>permanently</u>.
- Option 2: The basic setpoint change will be accepted <u>only temporarily</u> (default).

In this connection, you can use the *"accept modification of the basic setpoint permanently"* parameter in the *"room temperature controller function/setpoints"* parameter branch to determine whether you want to permanently (select *"yes"*) or only temporarily (select *"no"*) save the changed basic temperature value.

Option 1:

If you change the basic temperature setpoint of the first or second control loop it will be permanently saved in the EEPROM of the push sensor. In this connection, the newly set value will overwrite the basic setpoint temperature originally parameterized by the ETS. This is the only way to keep the changed basic setpoint, even after changing the operating mode or after a reset.

Important:

- Frequent changing of the basic temperature (e. g. several times a day) can adversely affect the life of the device as the non-volatile memory (EEPROM) used has been designed for less frequent write access events only.
- Any value assigned by local operation will not be taken over into object 31.
- After bus voltage recovery, the saved basic setpoint will still remain active. However, the value of object 31 or 32 will be "0". You can only read out the current basic setpoint (set the "R"-flag) after an external object update.

Option 2:

The basic setpoint selected on the push sensor or received through the object will remain stored in the currently set operating mode only in temporary form. In case of bus voltage failure or after you have changed the operating mode (e. g. from comfort to standby), the basic setpoint given by local operation or received through the object will be discarded and replaced by the value originally parameterized in the ETS.

Changing the setpoints for the standby and economy modes as well as for the dead band (comfort cooling temperature):

As the setpoint temperatures for the "standby" and "economy" modes, or the setpoints for the "cooling" mode, respectively, are derived from the basic setpoint temperature, taking account of the decrease, increase or dead band values parameterized in the ETS plug-in, these setpoint temperatures will also linearly shift by the basic setpoint change made.

Only by local operation on the controller at operator level 2, you can, in addition, set temperature values for the standby and economy modes as well as for the dead band other than those parameterized in the ETS for the first control loop. In such case, the originally parameterized decrease, increase and dead-band temperatures will be replaced by the newly resulting values caused by the locally changed temperature setpoints. In this connection, the temperature setpoints for the standby and economy modes or for comfort "cooling" (dead band) will always be permanently stored in the EEPROM, no matter what the setting of the "*accept modification of the basic setpoint permanently"* parameter is.

4.4.2.2 Basic setpoint shift

In addition to the fixed presetting of individual temperature setpoints by the ETS, by local operation at operator level 2, or by the basic setpoint object, it is possible to allow the user to shift the basic setpoint within a selectable range. By actuating one of the keys of rocker 1 at operator level 0 (normal operation), you can make visible in the display window the setpoint temperature of the activated operating mode of the first control loop if access to the operator levels is enabled.

In addition, you can press the right or left key of rocker 1 to shift the displayed setpoint temperature in increments or decrements of 0.1 °C. Please note that such shift of the displayed setpoint temperature (basic temperature offset) will directly act on the basic setpoint, thus also shifting all the other temperature setpoints. If you use two control loops with separate setpoints the setpoint temperatures of <u>both</u> control loops will be shifted.

Whether a basic setpoint shift will only act on the temporarily activated operating mode or on all the other setpoint temperatures of the remaining operating modes can be preset by the *"accept modification of shift of the basic setpoint permanently"* parameter in the *"setpoints"* parameter branch.

Setting *"no"* (default):
The basic setpoint shift made will only be active until the operating mode is changed. After a change, the setpoint shift will be reset to "0".

Setting *"yes"*:
The basic setpoint shift made will, in general, act on all operating modes. The shift will be maintained even after a change.

You can use the *"upward adjustment of basic setpoint temperature"* or *"downward adjustment of basic setpoint temperature"* parameter to define the temperature range selectable for a basic setpoint shift. In this connection, you can shift the current setpoint by a maximum of +/- 10 °C or +/- 10 K, respectively.

Notes on the basic setpoint shift:

- As the value for the basic setpoint shift is only saved in a volatile memory (RAM) the shift will get lost after a reset (e. g. bus voltage failure).
- A setpoint shift will have no effect on the frost or heat protection temperature setpoints.
- If controller operation is disabled rocker 1 will show no response when being actuated.

4.4.3 Transmitting the setpoint temperature

The setpoint temperature specified by the active operating mode or, if necessary, readjusted subsequently can be actively transmitted to the bus via object 55 or, if both control loops are used with separate setpoints, additionally via object 56.
The *"transmission at setpoint temperature modification by…"* parameter in the *"room temperature controller functions - setpoints"* parameter branch sets the temperature value by which the setpoint must change until the setpoint temperature value is automatically transmitted via the object. In this connection, temperature value changes between 0.1 °C and 25.5 °C or 0.1 K and 25.5 K are possible. Setting "0", in this case, will deactivate automatic transmission of the setpoint temperature.
In addition, the setpoint can be transmitted cyclically. The *"cyclic transmission of setpoint temperature"* parameter sets the cycle time (1 to 255 minutes). The value "0" (default) will deactivate the cyclic transmission of the setpoint temperature value.
Please note that no more telegrams regarding the setpoint temperature will be sent if cyclic transmission and automatic transmission upon a change have been deactivated.

You can set the "R"-flag on the "setpoint temperature" object to read out the current setpoint. After bus voltage recovery, re-programming by the ETS, or after re-plugging the application module, the object value will be updated and actively transmitted to the bus in accordance with the current setpoint temperature value.

4.5 Room temperature measurement

The room temperature controller measures the actual temperature and compares it with the preselected setpoint temperature. With the aid of the selected control algorithm, the variable is then calculated from the difference between the actual and the setpoint temperature.
In order to always ensure error-free and efficient room temperature control it is of utmost importance that the actual temperature is measured accurately.

The push sensor 2 plus has an integrated temperature sensor. Alternatively (e. g. if the push sensor has been installed at an unfavourable place, or under difficult application conditions such as in damp rooms) or additionally (e. g. in large rooms or halls), you can externally connect a second EIB temperature sensor via the bus and use it for actual-value measurements if you only have one control loop.
If you use both control loops, the actual temperature value of the second loop will be determined by the external sensor. In this connection, the actual-temperature measurement of the first control loop can exclusively be made by the internal sensor.

When choosing the place where you want to install the push sensor 2 plus or the second sensor, please take account of the following points:

- Avoid integration of the push sensor in multiple combinations, particularly if flush-mounted dimmers are involved.
- Do not install the sensors in the vicinity of major electric consumers (thermal radiation).
- Do not install the sensors in the vicinity of radiators or cooling systems.
- Avoid direct sun radiation onto the temperature sensors.
- The installation of sensors on the inside of an outer wall can adversely affect the temperature measurement.
- Install temperature sensors at least 30 cm away from doors or windows and at least 1.5 m above the floor.

4.5.1 Sensing the temperature and creating measured values

Temperature sensing in a control loop depends upon the parameterization. If you use both control loops, the actual temperature value of the second loop will be determined by the external sensor.

One control loop:
In only one control loop, the *"temperature sensing"* parameter in the *"room temperature controller function - room temperature measurement"* parameter branch specifies by which sensors you want to determine the actual temperature. In this connection, the following settings are possible:

- *"Internal sensor"*:
  The temperature sensor integrated in the push sensor 2 plus is activated. Thus, the actual temperature value is determined only locally in the device.
  This parameterization will initiate control immediately after a reset.
- *"External sensor"*:
  The actual temperature is determined only by the external sensor. The internal sensor is deactivated. In this connection, the external sensor must send the detected temperature value to the *"ext. temperature sensor"* 2-byte object 29 (DPT-ID 9.001) of the push sensor 2 plus. Alternatively or additionally, the push sensor can request the current temperature value in cycles (set the "R"-flag on the external sensor). For this purpose, set the *"scanning time for external sensor…"* parameter to a value of > "0". You can set the scanning interval from 1 minute to 255 minutes. This parameterization will make the room temperature controller wait for a temperature value telegram from the external sensor after a reset until control starts and a variable is output, if necessary.
- *"Internal and external sensor"*:
  In this case, both the internal and the external temperature sensor is active. In this connection, the external sensor must send the detected temperature value to the *"ext. temperature sensor"* 2-byte object 29 (DPT-ID 9.001) of the push sensor 2 plus. Alternatively or additionally, the push sensor can request the current temperature value in cycles (set the "R"-flag on the external sensor). For this purpose, set the *"scanning time for external sensor…"* parameter to a value of > "0". You can set the scanning interval from 1 minute to 255 minutes. This parameterization will make the room temperature controller wait for a temperature value telegram from the external sensor after a reset until control starts and a variable is output, if necessary.

  This setting results in creating the actual temperature from the two temperature values measured. In this connection, the *"creating of measuring value internal against external"* parameter specifies the weighting of the temperature values. Thus, you can balance the actual temperature measurement in dependence on the various places of installation of the sensors or on a possibly different heat distribution within the room, respectively. Often, temperature sensors which are under negative external influences (for example, unfavourable place of installation due to sun radiation, or in the vicinity of a radiator or of a door/window) are weighted less intensively.

  Example:
  push sensor 2 plus installed next to the room entrance door (internal sensor). An additional external temperature sensor has been installed on an internal wall in the middle of the room below the ceiling.

Internal sensor:	21.5 °C (internal sensor measuring range: 0 °C to + 40 °C ±1 %)
External sensor:	22.3 °C
Creating measured value:	30 % vs. 70 %

Result: $T_{\text{internal result}} = T_{\text{internal}} \cdot 0.3 = 6.45\ °C$, $T_{\text{external result}} = T_{\text{external}} = 22.3\ °C \cdot 0.7 = 15.61\ °C \rightarrow$

$T_{\text{actual result}} = T_{\text{internal result}} + T_{\text{external result}} = \underline{\underline{22.06\ °C}}$

Two control loops:
The actual temperature measurement is exclusively made by the internal sensor. The external sensor detects the actual temperature of the second control loop and must, in this connection, send this temperature value to the *"external temperature sensor"* 2-byte object 29 (DPT-ID 9.001) of the push sensor 2 plus. Alternatively or additionally, the push sensor can request the current temperature value in cycles (set the "R"-flag on the external sensor). For this purpose, set the *" scanning time for external sensor…"* parameter to a value of > "0". You can set the scanning interval from 1 minute to 255 minutes. This parameterization will make the room temperature controller wait for a temperature value telegram from the external sensor after a reset until the control process of the second loop starts and a variable is output, if necessary.

4.5.2 Adjusting the measured values

In some cases, it can become necessary to adjust the temperature measurements between the internal and external sensors. Such adjustment will, for example, be necessary if the temperature measured by the sensors is permanently below or above the room temperature actually prevailing in the vicinity of the sensor. In this connection, the actual room temperature should be determined by a reference measurement made with the aid of a calibrated thermometer.

You can use the *"adjustment internal sensor…"* or *"adjustment external sensor…"* parameter in the *"room temperature controller function - room temperature measurement"* parameter branch to parameterize the positive (temperature increase, factors: 1 … 127) or the negative (temperature decrease, factors: -128 … -1) temperature adjustment in steps of 0.1 °C. Thus, you will only have to make this adjustment once, and it will be the same for all operating statuses.

Important:

- Increase the measured value if the value measured by the sensor is below the actual room temperature.
  Decrease the measured value if the value measured by the sensor is above the actual room temperature.
- When the measured value is created by means of the internal and external sensors only for one control loop the adjusted value will be used for the calculation of the actual value.

4.5.3 Transmitting the actual temperature

The measured actual temperature of the first control loop can be actively transmitted to the bus via object 28, *"actual temperature".*
The *"transmission at room temperature modification by…"* parameter in the *"room temperature controller functions - room temperature measurement"* parameter branch sets the temperature value by which the actual value must change until the actual temperature value is automatically transmitted via object 28. In this connection, temperature value changes between 0.1 °C and 25.5 °C or 0.1 K and 25.5 K are possible. Setting "0" in this case, will deactivate automatic transmission of the actual temperature.
In addition, the actual value can be transmitted cyclically. The *"cyclical transmission of room temperature"* parameter sets the cycle time (1 to 255 minutes). The value "0" (default) will deactivate the cyclic transmission of the actual temperature value.
You can set the "R"-flag on the "actual temperature" object to read out the current actual value.
Please note that no more telegrams regarding the actual temperature will be sent if cyclic transmission and automatic transmission upon a change have been deactivated.

After bus voltage recovery, re-programming by the ETS, or after re-plugging the application module, the object value will be updated and actively transmitted to the bus in accordance with the current actual temperature value. If no temperature value telegram has been received from the external sensor yet when the external sensor is used in one control loop only the value created by the internal sensor will only be sent. If you only use the external sensor the object will carry a value of "0" after a reset. For this reason, the external sensor should always transmit the current value after a reset.

4.6 Room temperature controller inhibit functions

4.6.1 Disabling the controller

At some operating statuses, it can become necessary to disable room temperature control. For example, you can switch off control in the dew-point mode of a cooling system, or during maintenance work done on the heating or cooling system.

The *"switch off controller (dew-point operation)"* parameter in the *"room temperature controller functions"* parameter branch will enable object 45, *"disabling controller"*, when you set it to *"via bus"*. Moreover, you can use setting *"no"* (default) to permanently deactivate the controller inhibit function.
If a "1" telegram is received through the enabled inhibit object the room temperature control of both control loops will be entirely deactivated. In this case, all variables will be = "0". However, you can still operate the controller.

In the two-stage heating or cooling mode, you can separately disable the additional stage. The *"additional stage inhibit object"* parameter in the *"room temperature controller functions"* parameter branch will enable object 41, *"disabling additional stage"*, when you set it to *"yes"*. Moreover, you can use setting *"no"* (default) to permanently deactivate the additional stage inhibit function.
If a "1" telegram is received through the enabled inhibit object room temperature control by the additional stage will be deactivated. The variable of the additional stage will be "0", with the basic stage incessantly continuing working.

If you use both control loops you can separately disable the second control loop. If a "1" telegram is received via inhibit object 46, *"inhibit 2nd control loop"* the room temperature control of the second control loop will be deactivated, the variable of this loop being "0". In this case, the first control loop will incessantly continue working.

Disabling will always be cancelled after a reset.

4.6.2 Disabling controller operation

You can disable room temperature controller local operation (all rockers). Activated disabling of operation will be indicated in the display by the blinking " 🔒 " symbol. Please note that this symbol will also light up when a push sensor inhibit function is activated.

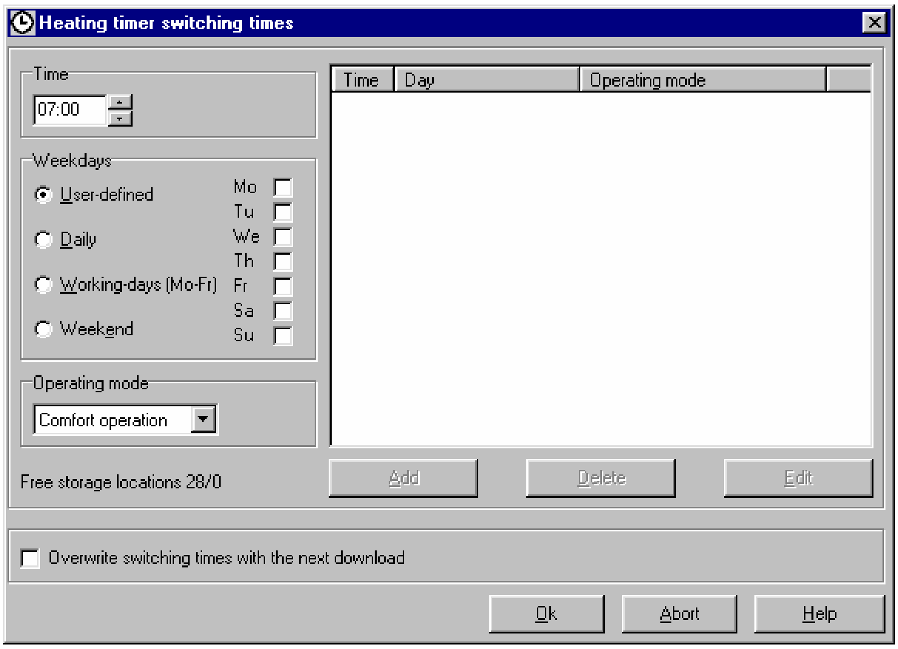
You can use the *"operation controller inhibitable"* parameter in the *"room temperature controller functions"* parameter branch to preset whether local operation shall always be impossible (setting: "*always disabled*") or can be disabled by object 44, *"disabling controller operation"* (setting *"via bus"*).
If you select *"always disabled"* you cannot parameterize room temperature controller operation under the key or rocker functions when the push sensor functionality is involved. In addition, rocker 1 will have no function if you select this setting.
The *"via bus"* setting will deactivate local operation when the object receives a "1" telegram. Consequently, local operation will be re-enabled upon the receipt of a "0" telegram. Actuating a key assigned as room temperature controller operation, or actuating rocker 1, will show no response if disabling is active.

Activated disabling of the controller operation will neither affect the operation of the controller via the objects nor room temperature control itself, i. e. the control algorithm will work and create variables and status signals.

Controller operation disabling will always be cancelled after a reset.

4.7 Heating timer

The heating timer can distinguish up to 28 different switching programs and facilitates the switch-over of the room temperature controller operating mode, depending on the time and the day of the week. Use the *"heating timer = ON"* parameter in the *"room temperature controller function - heating timer"* parameter branch to enable the heating timer. Alternatively, you can use the *"heating timer = OFF"* setting to permanently disable this function (default). When the heating timer function is enabled you can parameterize the switching programs in the ETS plug-in, and they will then be executed in chronological order. The *"heating timer"* menu item in the *"timer editor"* menu will call the "heating timer switching events":



Heating timer switching times

Time: 07:00

Weekdays:
- ● User-defined
- ○ Daily
- ○ Working-days (Mo-Fr)
- ○ Weekend

Mo □ Tu □ We □ Th □ Fr □ Sa □ Su □

Operating mode: Comfort operation

Free storage locations 28/0

□ Overwrite switching times with the next download

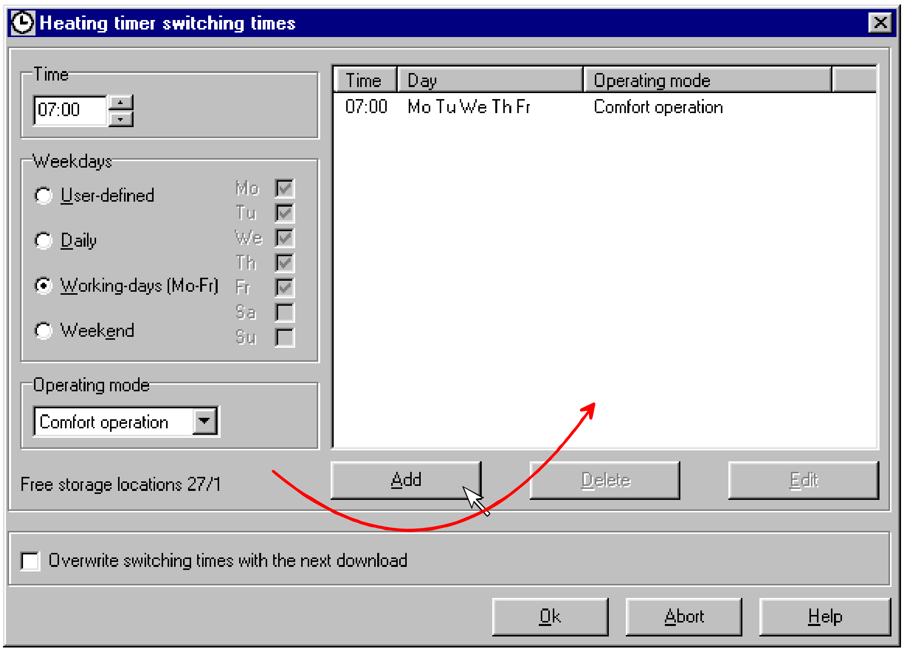
In the left part of the window, you can define the time of the switching program down to the minute. After this, you can define the days of the week on which you want this switching time to be activated. You can choose from the "user-defined" (Mo, Tu, … Su), "daily" (Mo - Su), "work days" (Mo - Fr) or "weekend" (Sa - Su) options.

Each switching program will occupy one memory location in the push sensor 2 plus. For the "daily", "work days" or "weekend" settings, only one memory location will be required. For the "user-defined" option, however, sometimes up to five different switching programs can be created, depending on the parameterized days. The "free memory locations" information parameter in the editor indicates how many memory locations are available (the number in front of the slash) or how many memory locations will have to be occupied by the selected setting (the number behind the slash).

In addition, you must specify the operating mode which you want to activate when a switching program is called. For this purpose, the "comfort", "standby", "economy" and "frost/heat protection" modes are available.
Please note that an operating mode set by the heating timer will have the same priority as a local condition on the push sensor or caused by the switch-over objects (4 x 1-bit or 1-byte KONNEX switch-over object) and can thus be changed.

The switching times can be configured down to the minute. When the heating timer is activated the switching times will be checked every second by the time control of the push sensor so that, for example, a program configured for 7:00 a. m. will be executed exactly at 7:00 a. m. and 00 seconds. The internal clock of the push sensor should be set by an external time control signal at least every hour to keep the time error as low as possible. If such updating from the bus takes place at the moment of a switching time it can be possible that this switching time concerned will be 'skipped' due to a time shift, thus not being executed. For this reason, ensure that updating the time and executing a switching program will not coincide.

After you have defined the switching program (switching time, week days, operating mode), you can click on the "add" button to take the program into the program list. This list will appear on the right in the editor window.



Heading timer switching times

Time: 07:00

Weekdays:
- ○ User-defined
- ○ Daily
- ● Working-days (Mo-Fr)
- ○ Weekend

Mo ☑ Tu ☑ We ☑ Th ☑ Fr ☑ Sa □ Su □

Operating mode: Comfort operation

Time	Day	Operating mode
07:00	Mo Tu We Th Fr	Comfort operation

Free storage locations 27/1

□ Overwrite switching times with the next download

Thus, you can preset up to 28 switching programs. You can edit a program created in the right part of the window if you highlight it and click on the "edit" button. Highlighting a program and clicking on the "delete" button will delete the program and remove it from the list.
Click on "OK" to accept the settings into the configuration of the push sensor.

If you check the "overwrite switching times with the next download" option in the editor window the switching programs of the heating timer will be loaded into the device when the entire application is being programmed, or when you are partially programming the parameters.

In addition to presetting the heating timer programs in the ETS plug-in, you can locally edit on the push sensor at operator level 2 the switching times, the week days and the operating modes even after programming. A precondition to do so is that access to all operator levels on the push sensor has been enabled (depending on the parameters).
To avoid overwriting of the locally changed data by a subsequent programming process by the ETS you can uncheck the "overwrite switching times with next download" option.

If it is enabled, you can activate or deactivate the heating timer by local operation at operator level 1 (refer to "1.5 Operator levels/Local operation) and/or by push sensor actuation (key function). When this function has been activated the " 🕒 " symbol will light up in the display, and the switching programs will be executed in chronological order at the "00" second marks in accordance with the parameterized switching times.
Info: If you activate the heating timer exactly at the moment of a parameterized switching time, and the "00" second mark of the parameterized minute has already been passed, the switching program concerned will be executed afterwards.

After switching programs have been fed into the device the heating timer will be activated immediately after the initializing phase, and the programs will be executed. A valid time and a valid week day received are a prerequisite to this. If no programs have been stored in the device and the function itself has been enabled no switching programs will be executed, although the symbol will be lit after a reset.

An operating mode switch-over by the heating timer can additionally and temporarily be suppressed via a separate inhibit object. To facilitate the inhibit function set the *"disable heating timer through bus"* parameter in the *"room temperature controller function - heating timer"* parameter branch to *"yes"*. In this case, inhibit object 62, *"disabling heating timer"*, will be enabled. You can parameterize its polarity.
When the heating timer has been disabled through the bus, the " 🕒 " symbol will be blinking in the display. During an active inhibit function, the operating mode will not be changed by the heating timer. If you re-enable the heating timer exactly at the moment of a parameterized switching time the switching program concerned will be executed afterwards. Activations or deactivations of the heating timer during the disabling phase will be stored and followed-up after the end of disabling.

Notes on the heating timer:

- The internal clock of the push sensor should be set by an external time control signal at least every hour to keep the time error as low as possible. Unless the internal clock has been updated through the bus at least once per day (update check at 4:00 a. m.) the display of the push sensor will read "--:--" if the time is indicated in the display (depending on the corresponding parameter). However, the internal clock will keep running with the expected time error, and the switching programs of the heating timer will still be executed.
- The week day information is derived from the time signal. The heating timer will only execute the given switching programs after it has received a valid week day. The week days will be shown in the display of the push sensor in alternative form by digits 0 to 7 if the time is indicated by the display (depending on the corresponding parameter).

4.8 Valve protection

To avoid calcification or seizing of the triggered radiator or cooling system control valves you can perform cyclic valve protection. The *"valve protection"* parameter in the *"room temperature controller function"* parameter branch will activate valve protection if set to *"yes"*.

This protective function will, in general, only be started for variable outputs which are not active, i. e. for outputs which have not requested any heating or cooling energy during the past 24 hours.
For these outputs, the controller will cyclically set the variable to its maximum value once per day for a period of some 5 minutes, taking account of the following parameterization:

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

Thus, even such valves which are closed for longer periods will be shortly opened at regular intervals.

Valve protection is controlled by the internal clock and activated for the output variables concerned at 8:00 a. m. If the internal clock was not yet set after a reset, valve protection will be done for the first time some 32 hours after the reset, at the earliest.

5. Control functions

5.1 Function

The push sensor 2 plus has up to two separate control functions. Each of these functions allow time-operated or temperature-dependent transmission of switching commands (ON/OFF) or value telegrams (0…255) to the bus through one object, depending on the data type.
You can preferably use such commands for triggering long-time or position objects of blinds or roller shutters. However, any other bus control options are also possible by such commands.
Depending on the control command transmitted, the display of the push sensor will show the " ▲ " and " ▼ " symbols:

Data type	Control command	Symbol displayed
Switching	1 0	▼ ▲
Value	1...255 0	▼ ▲

The display symbols will light up as soon as the control command has been transmitted and will go out 20 seconds later.

You can use the *"control function 1"* or *"control function 2"* parameters in the *"push sensor 2 plus"* parameter branch to separately enable the control functions. When the functions are in the deactivated state (default) the parameters and the objects of the control functions will be hidden.

I. Temperature-dependent control function:

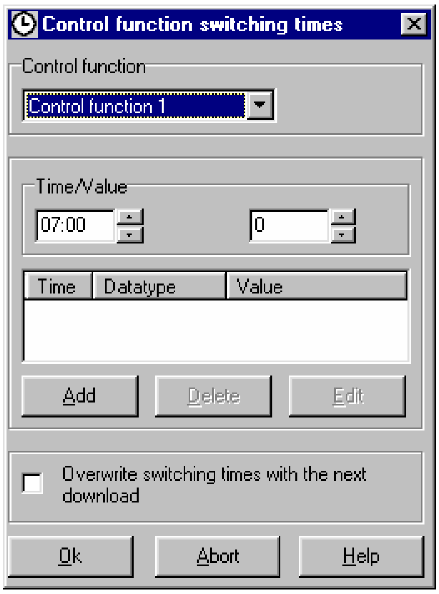
The *"function"* parameter in the *"control function X"* (X = 1 or 2) parameter branch has been set to *"temperature-dependent"*. In this case, the measured room temperature is compared with the basic setpoint of the first control loop every minute. If the temperature difference is higher or lower than a preset value the parameterized command will be sent to the bus.

Optionally, the control command can be delayed when the preset temperature is exceeding its value or falling below it. The *"exeed telegram delay time"* or *"remain telegram delay time"* parameters set the two delay times. The delay time can suppress short-time room temperature changes which would raise a control command. Setting "0" will deactivate the delay time (default).
If you have parameterized telegram delay please note that the control command will be transmitted to the bus after the delay has elapsed, at the earliest. As the temperature difference is checked only once per minute a short time prolongation before telegram transmission must be expected in most cases.

For the exceeding of or falling below the preset temperature, different control commands can be sent. You can use the *"value when exceeding"* or *"value when remaining under"* parameters to set these commands. Depending on the parameterized data type, you can set switching or value commands.
For the temperature-dependent control function, the commands will be transmitted only once each time if the room temperature is equal to or less than the parameterized temperature when exceeding the latter and equal to or greater than the preset temperature limit when falling below it.

II. Time-operated control function

The *"function"* parameter in the *"control function X"* (X = 1 or 2) parameter branch has been set to *"heating timer"*. Once you have thus assigned one of the two control functions to the timer, the *"control functions"* menu item in the *"heating timer editor"* main menu will be enabled. When you select this menu item the "control function switching times" window will appear. In dependence on the parameterization of the two control functions, you can parameterize from this window each of the maximum of two switching times down to the minute, together with the associated control, depending on the parameterized data type.



Control function switching times

Control function: Control function 1

Time/Value: 07:00 | 0

Time	Datatype	Value

□ Overwrite switching times with the next download

After you have set the switching time and the control command, you can click on the "add" button to create the switching program. You can create the maximum of the two switching programs per control function from the program list in the middle of the window. If you highlight a program in the list and click on the "edit" button you can edit this program. Highlighting a program and clicking on the "delete" button will delete the program and remove it from the list.
Click on "OK" to accept the settings into the configuration of the push sensor.

If you check the "overwrite switching times with the next download" option in the editor window the switching programs of the control function(s) will be loaded into the device when the entire application is being programmed, or when you are partially programming the parameters.

By local operation at operator level 2, you can edit the switching times of possibly both control functions. A precondition to do so is that access to all operator levels on the push sensor has been enabled (depending on the parameters). Thus, you can subsequently change the times programmed by the ETS plug-in. With the next download, you can have the ETS replace the locally changed data by the originally parameterized times. For this purpose, you must have checked the "overwrite switching times with the next download" option in the "control function switching times" window. If this option is not checked no switching timer data configured in the ETS plug-in will be loaded into the device, neither will any changed switching or value commands. The locally set switching times will thus remain unchanged.
You cannot change the control commands assigned in the ETS plug-in.

Notes on the time-operated control function:

- The switching times can be configured down to the minute. When the control function is activated the switching times are checked every second by the time control of the push sensor so that, for example, a program configured for 7:00 a. m. will be executed exactly at 7:00 a. m. and 00 seconds. The internal clock of the push sensor should be set by an external time control signal at least every hour to keep the time error as low as possible. If such updating from the bus takes place at the moment of a switching time it can be possible that this switching time concerned will be 'skipped' due to a time shift, thus not being executed. For this reason, ensure that updating the time and executing a switching program will not coincide.
  <u>Unless</u> the internal clock has been updated through the bus at least once per day (update check at 4:00 a. m.) the display of the push sensor will read "--:--" if the time is indicated in the display (depending on the corresponding parameter). However, the internal clock will keep running with the expected time error, and the switching programs of the control function(s) will still be executed.
- If both switching times have been paramterized to the same time only the command of the second switching time will be transmitted to the bus, the symbols in the display, however, being possibly lit both if different control commands have been set. Distinction of the switching times by the week day or date will not be possible.
- The control function(s) will only execute the given switching programs after it (they) has (have) received a valid time.

5.2 Activating and deactivating the control functions

I. Temperature-controlled control function:

If it is enabled, you can activate or deactivate the control function by local operation at operator level 1 (refer to "1.5 Operator levels/Local operation) and/or by push sensor actuation (key function). When a function has been activated the "1" or "2" symbol will be lit in the display, and temperature monitoring will be activated.

Temperature-based control functions will only be active after the initializing phase of the push sensor if the room temperature controller was able to determine a valid room temperature. This will be the case immediately after initialization when the internal temperature sensor is used. For the alternative or additional use of the external temperature sensor, a temperature value must first be received before the room temperature can be determined. Also, activation from operator level 1 or by push sensor operation will only be possible when a valid room temperature is available. Moreover, you can only activate the control function if you have parameterized at least one control command in the ETS plug-in.

In addition, you can temporarily suppress the transmission of a control command by the control function(s) via separate inhibit objects. To facilitate such inhibit function set the *"inhibit object"* parameter in the *"control function X"* (X = 1 or 2) parameter branch to *"yes"*. In this case, inhibit object 59, *"disabling control function 1"*, or inhibit object 61, *"disabling control function 2"* will be enabled. You can parameterize their polarities.
When the control function has been disabled through the bus, the " 1" or "2" symbol will be blinking in the display. During an active inhibit function, no control command will be transmitted. After the inhibit function has been completed, the temperature will be evaluated and, if necessary, a control command sent to the bus.
Activations or deactivations of the control function(s) during the disabling phase will be stored and followed up after the end of disabling.

II. Time-operated control function

If it is enabled, you can activate or deactivate the control function by local operation at operator level 1 (refer to "1.5 Operator levels/Local operation) and/or by push sensor actuation (key function). When this function has been activated the "1" or "2" symbol will light up in the display, and the switching programs will be executed in chronological order at the "00" second marks in accordance with the parameterized switching times.

After switching programs have been fed into the device, the control function will be activated immediately after the initializing phase, and the programs will be executed. A prerequisite to this is a valid time received. Time-based control functions without a defined switching program will not be activated after initialization. Although activation from operator level 1 or by push sensor operation is possible in such case, no control commands will be transmitted to the bus.

In addition, you can temporarily suppress the transmission of a control command by the control function(s) via separate inhibit objects. To facilitate such inhibit function set the *"inhibit object"* parameter in the *"control function X"* (X = 1 or 2) parameter branch to *"yes"*. In this case, inhibit object 59, *"disabling control function 1"*, or inhibit object 61, *"disabling control function 2"* will be enabled. You can parameterize their polarities.
When the control function has been disabled through the bus, the " 1" or "2" symbol will be blinking in the display. During an active inhibit function, no control command will be transmitted. After the inhibit function has been completed, time control will be re-activated, and all further switching programs will be executed. Switching programs which have elapsed during an active inhibit function will not be followed up.
Activations or deactivations of the control function(s) during the disabling phase will be stored and followed up after the end of disabling.

6. Scene Function

6.1 Scene definition

Similar to a light scene push-button sensor, the push sensor 2 plus has a scene function. Under this function, you can save in the push sensor up to eight different scenarios. Each scene can trigger up to eight bus outputs (scene objects). You can configure switching, value or shutter/blind commands.

You can use the *"scene function"* parameter in the *"push sensor 2 plus"* parameter branch to enable the scene function. When the functions are in the deactivated state (default) the parameters and the objects of the scene functions will be hidden.

Depending on the respective scene recalled, the scene commands are transmitted to the bus via the scene outputs. In the *"scene function - scene X"* (X = 1 to 8) parameter branch, you can separately define the scene command for each output.
In the ETS plug-in, the data type for each scene object can be parameterized in the *"scene function"* parameter branch. Possible types and, thus, available commands are:

Data type	Scene command
Switching (1 bit)	ON ("1")
	OFF ("0")
Value (1 byte)	0...255
	alternative *
	0...100 %
Shutter/blind (1 bit)	UP ("0")
	DOWN ("1")

*: The *"value type"* parameter in the *"scene function"* parameter branch defines whether dimensionless values (0…255) or percentage values (0…100 %) shall be preset for the *"value"* data type.

Up to eight scene commands per scene can be transmitted via the output objects. For each scene output, you can parameterize whether a command should, in general, be sent when a scene is being recalled. The *"send output = yes"* parameter setting in the *"scene function - scene X"* (X = 1 to 8) parameter branch enables the scene command. Consequently, you can use setting *"no"* to suppress scene commands for the output concerned.

The scene commands are permanently saved in the push sensor so that they will not get lost after a bus voltage failure.

6.2 Scene recall/scene saving

You can recall a scene by:

- the scene extension object (object 71):
  A scene number received via the scene extension object will recall an internally stored scene. This way of recalling is frequently used by external bus components such as push sensors, display panels or complex scene control systems.
- a local key function on the push sensor:
  In addition, you can recall a stored scene by locally actuating a key on the push sensor. If you have parameterized the key function to *"light scene extension/recall"*, and if you want the key to act as *"internal scene recall"*, you can recall the scenes stored in the push sensor 2 plus by a short-time key operation (< 1 s). For this purpose, you must specify the corresponding scene number (1 to 8) in the ETS plug-in (refer to "3. Push sensor Functions").
  For this function, the extension object will only be required if triggering by external bus components is involved.

Even after they have been programmed by the ETS, you can still change the scenes stored in the push sensor 2 plus. You can save a scene by:

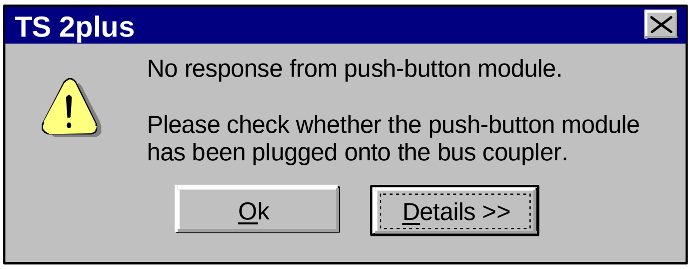
- the scene extension object (object 71):
  Via the extension object, a storage telegram is received. According to the scene number, the scene control of the push sensor 2 plus will request the current values of the scene objects from the actuators via the bus and save them permanently.
- a local key function on the push sensor:
  When parameterizing an *"internal scene recall"* with the storage function enabled, you can actuate the key for a long time (> 5 s) to save an internal scene in accordance with the parameterized scene number. In this connection, the scene control of the 2 plus push-button sensor will request the current values of the scene objects from the actuators via the bus and save them permanently.

In a saving process, the scene commands of the scene concerned and originally configured by the ETS will be replaced by the new values.
If the push sensor 2 plus does not receive any acknowledgement of a read request no new command will be saved. Non-transmitting scene objects of a scene cannot be changed.
You can, in general, save new switching, value or shutter/blind commands.

To enable the actuators on the bus to respond to the read request from the push sensor 2 plus set the read flag ("R"-flag) on the actuator objects concerned.
In order to avoid communication problems when recalling or saving scenes do not change the communication flags ("C"-flags) of the scene objects on the push sensor 2 plus.

7. Messages in a Programming Process

After you have programmed the push sensor 2 plus with the aid of the ETS plug-in, you can program it in the ETS start-up environment.
During a programming process, the following messages can appear:

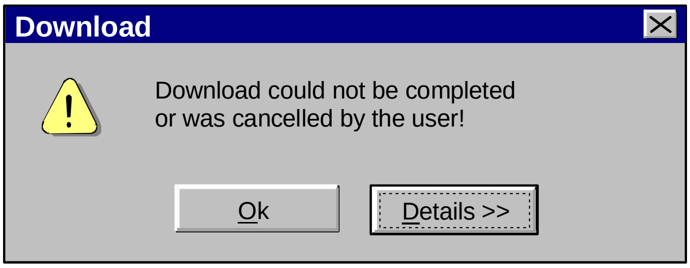


TS 2plus

No response from push-button module.

Please check whether the push-button module has been plugged onto the bus coupler.

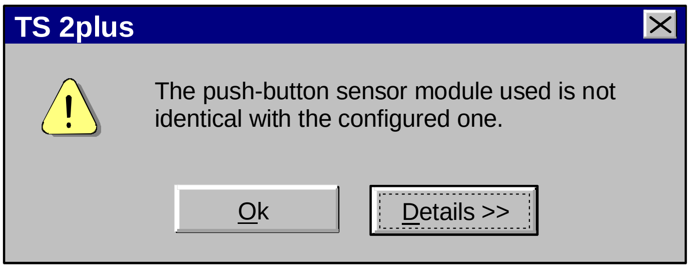
Reason: Attempt to load the application data into the device.
Cause: No push sensor 2 plus has been plugged onto the bus coupling unit.
Remedy: Plug the push sensor 2 plus onto the bus coupling unit. When doing so, keep in mind the correct physical address of the bus coupling unit.
Note: You can also program the physical address of the device, even though you have not plugged on the push sensor. Even for application data partial programming, you must have plugged on the push sensor.



Download

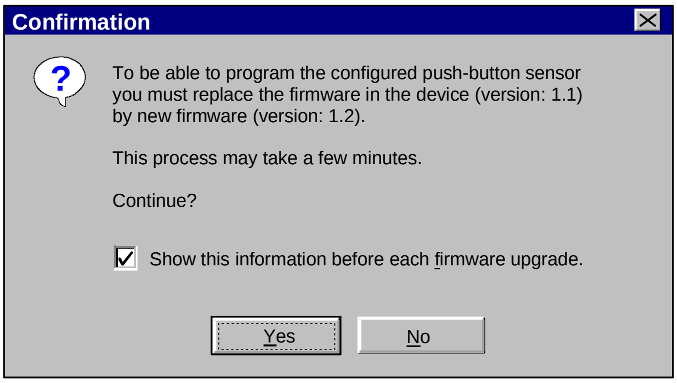
Download could not be completed or was cancelled by the user!

Reason: Attempt to load the application data into the device.
Cause: The programming process was cancelled through the "cancel" button, or there was a communication error.
Remedy: Start a new programming process.
Note: During a programming process, especially when the firmware is being programmed, larger data quantities are being sent to the device via the bus. In this connection, the intelligent programming algorithm of the push sensor 2 plus can recognize communication errors itself and re-transfer the erroneous data. Very rarely, errors can occur which cannot even be avoided by repeating the data transfer. In such cases, changing the data interface, the PC or the serial data connection to the data interface can be a remedy.
In the ETS plug-in of the push sensor 2 plus, you can specify the number of download attempts in the event of a problem from the "settings - options" menu on the "hardware" tab. The default setting of three attempts should only be changed in exceptional cases.
Please note that updating the firmware will only be necessary for special exceptions.


TS 2plus

The push-button sensor module used is not identical with the configured one.

Reason: Attempt to load the application data into the device.
Cause: A push sensor 2 plus variant other than the configured one has been plugged onto the bus coupling unit (e. g. 6fold configured and 2fold plugged on).
Remedy: Plug on the variant which conforms with the configuration.


Confirmation

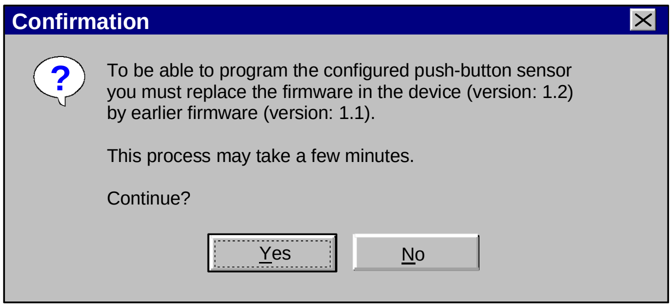
To be able to program the configured push-button sensor you must replace the firmware in the device (version: 1.1) by new firmware (version: 1.2).

This process may take a few minutes.

Continue?

☑ Show this information before each firmware upgrade.

Reason: Attempt to load the application data into the device.
Cause: A push sensor 2 plus containing earlier firmware (e. g. V 1.2) is being programmed with some later push sensor 2 plus software version.
Remedy: This message does not represent an error. If you click on "yes" to accept, different firmware which corresponds to the current configuration will be automatically loaded into the device. If you select "no" the earlier device will not be programmed as the parameters and functions preset by the new software are not backward compatible.
Note: If you uncheck the "show this information before each firmware upgrade" box this message will no longer appear, even though you are programming further earlier push sensor 2 plus versions.
You can re-check this box later in the ETS plug-in of the push sensor 2 plus from the "settings - options" menu on the "hardware" tab.



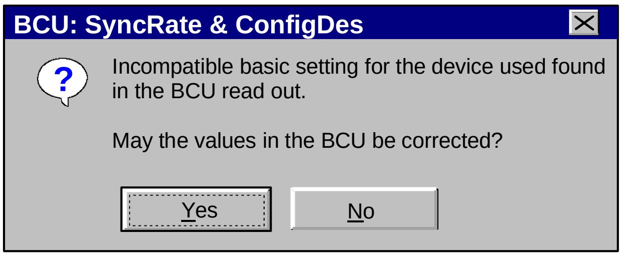
Confirmation

To be able to program the configured push-button sensor you must replace the firmware in the device (version: 1.2) by earlier firmware (version: 1.1).

This process may take a few minutes.

Continue?

Reason: Attempt to load the application data into the device.
Cause: You want to program a push sensor 2 plus containing new firmware (e. g. V 1.2). In this case, the version in the device is later than that specified by the push sensor 2 plus software used (e. g. V 1.1).
Remedy: This message does not represent an error. If you click on "yes" to accept, the later firmware contained in the device will be replaced by the earlier firmware version specified by the ETS plug-in. If you select "no" the later device will not be programmed as the parameters and functions preset by the earlier software are not upward compatible. In such case, you should reinstall current push sensor 2 plus software. Depending on the changes resulting therefrom, it can possibly become necessary to configure a new device in the ETS.



BCU: SyncRate & ConfigDes

Incompatible basic setting for the device used found in the BCU read out.

May the values in the BCU be corrected?

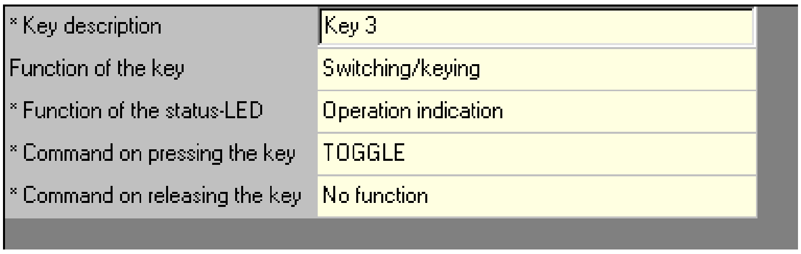
Reason: Attempt to load the application data into the device.
Cause: The push sensor 2 plus has been plugged onto a bus coupling unit which does not go with the configuration of the push sensor. It is probably a bus coupling unit which was originally used for a different purpose or a new device which has not been used yet in the existing push sensor 2 plus configuration.
Remedy: This message does not represent an error. If you click on "yes" to confirm, the data in the BCU will be overwritten. If you select "no" the later device will not be programmed as the data in the BCU does not go with the configuration of the push sensor.

8. Global parameters

It is often the case that devices of the same type are used in EIB installations and commissioned by an ETS plug-in. It may happen therefore that a certain number of push sensors with the same parameters and the same application must be entered and configured in the ETS project. To avoid multiple projecting and thus increased costs, the so-called global parameters feature has been implemented.

With the help of global parameters, selected settings of a project device can be transferred to several other devices of the same type in the actual ETS project without the need for individual editing of the devices. The devices to be adapted are selected during an adaptation procedure. Only devices of the same type and with the same number of channels can be globally adapted (e.g. 5-channel push sensors or 2-channel push sensors). After global adaptation, the procedure can be verified by means of a status report permitting detection of adaptation errors.

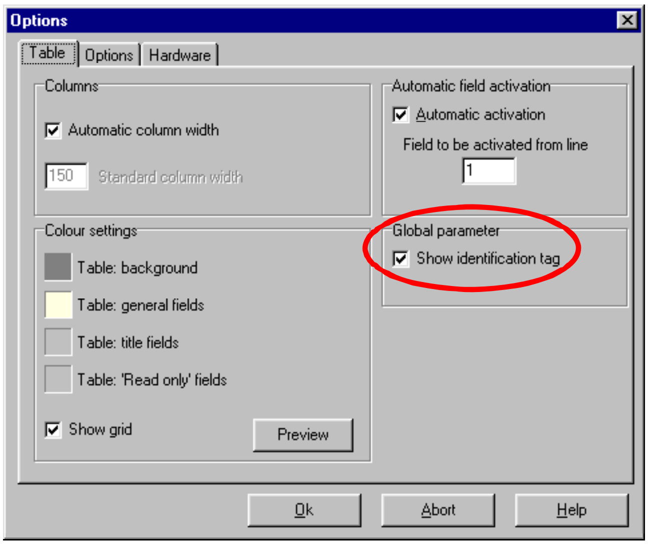
Only parameters without parent-child relations can be globally adapted. Parameters permitting global adaptation are identified by an "*" (asterisk symbol) at the beginning of the text line before the parameter concerned
(see fig. 1).



* Key description	Key 3
Function of the key	Switching/keying
* Function of the status-LED	Operation indication
* Command on pressing the key	TOGGLE
* Command on releasing the key	No function

Fig. 1: Example for global parameters marked by the asterisk

The global parameter identification can be activated in the menu under *"Setting – Options"* on the filecard *"Table"* in the plug-in software. To show the identification, tick the box *"Show identification tag"* (see fig. 2).



Columns

☑ Automatic column width

150 Standard column width

Colour settings

Table: background

Table: general fields

Table: title fields

Table: 'Read only' fields

☑ Show grid

Automatic field activation

☑ Automatic activation

Field to be activated from line 1

Global parameter

☑ Show identification tag

Fig. 2: Checkbox for activation of global parameter identification

Adaptation procedure

Before a global adaptation of parameters can be made, all devices to be adapted must first be existing in the ETS project. Thereafter, the parameter window or the device plug-in can be opened.

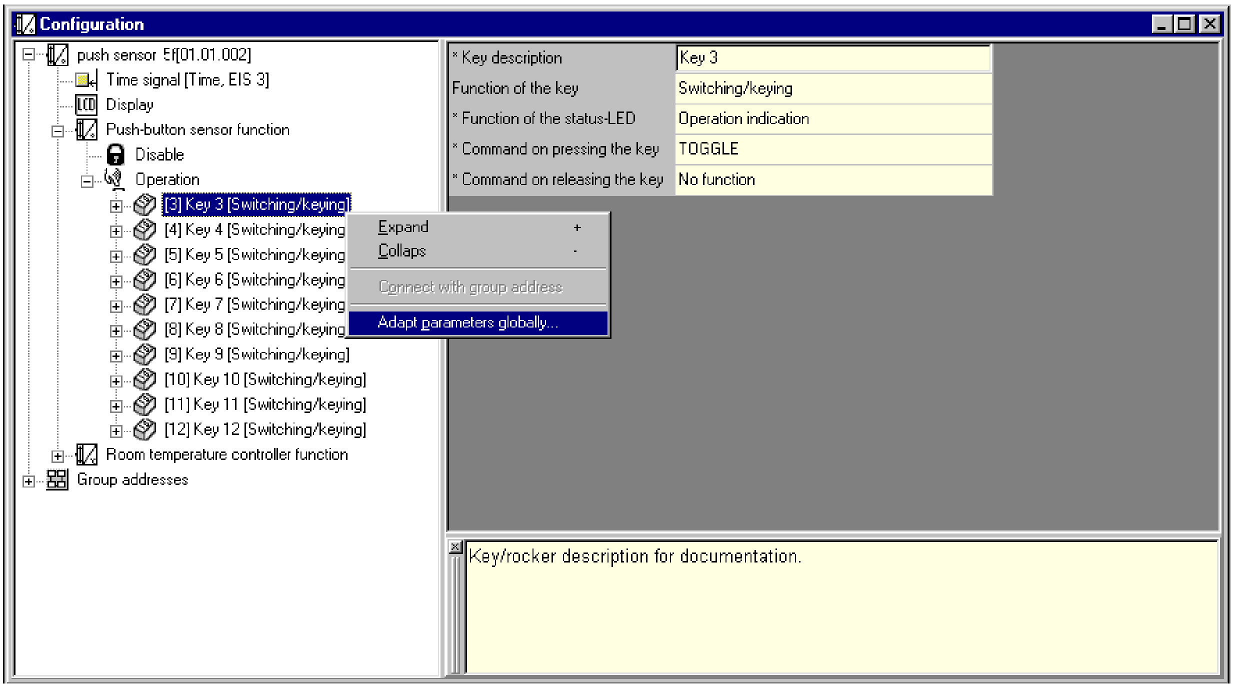
All parameters of a device are assigned to different parameter branches that can be selected on the left-hand side of the plug-in window (tree-view). Each branch can be adapted separately provided it has global settings.

For the adaptation, the following steps are necessary:

1. The (global) parameters must be selected at the user's discretion.

2. The parameter branch containing the parameters to be adapted must be selected.

3. In the context menu (click on right mouse button), the command *"Adapt parameters globally"* must be selected (see fig. 3). This command is only available if the parameter branch has global parameters.

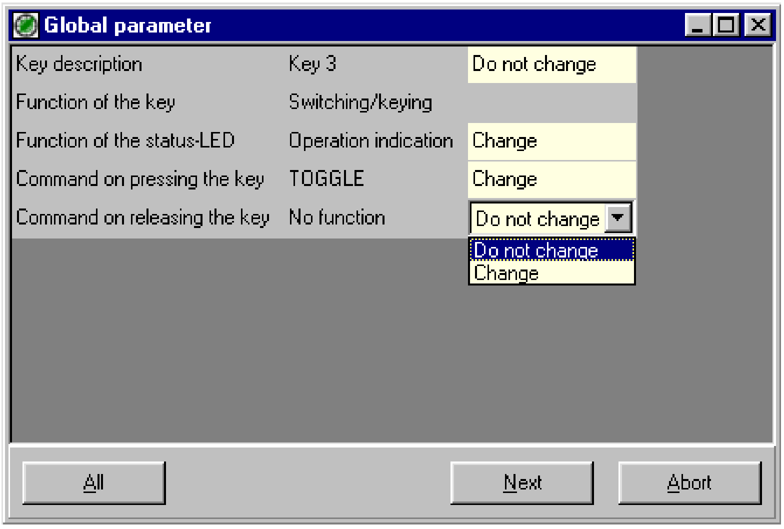

Configuration
push sensor 5f[01.01.002]
Time signal [Time, EIS 3]
Display
Push-button sensor function
Disable
Operation
[3] Key 3 [Switching/keying]
[4] Key 4 [Switching/keying
[5] Key 5 [Switching/keying
[6] Key 6 [Switching/keying
[7] Key 7 [Switching/keying
[8] Key 8 [Switching/keying
[9] Key 9 [Switching/keying]
[10] Key 10 [Switching/keying]
[11] Key 11 [Switching/keying]
[12] Key 12 [Switching/keying]
Room temperature controller function
Group addresses
Expand +
Collaps -
Connect with group address
Adapt parameters globally...

* Key description	Key 3
Function of the key	Switching/keying
* Function of the status-LED	Operation indication
* Command on pressing the key	TOGGLE
* Command on releasing the key	No function

Key/rocker description for documentation.

Fig. 3: Activating global adaptation of parameters in the selected branch by means of the context menu

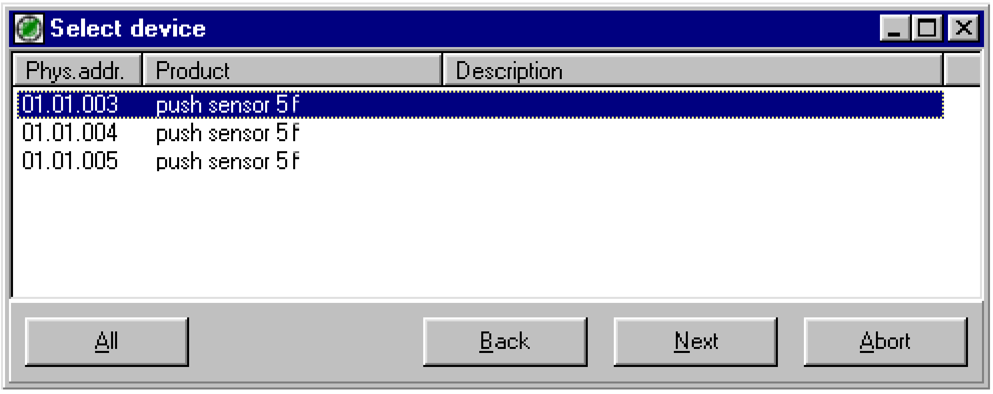
4. The *"Global parameter"* dialog (see fig. 4) is started. The dialog field lists the global parameters of the selected branch that can be adapted. The parameters to be adapted can now be selected. Only the settings marked *"Change"* will be adapted in the further course of action.
   If the *"All"* button is selected, all global parameters will be changed together (all parameters either changed or not changed).
   After specifying the desired items, select *"Next"*.



Key description	Key 3	Do not change
Function of the key	Switching/keying	
Function of the status-LED	Operation indication	Change
Command on pressing the key	TOGGLE	Change
Command on releasing the key	No function	Do not change

Fig. 4: Display and selection of global parameters to be adapted

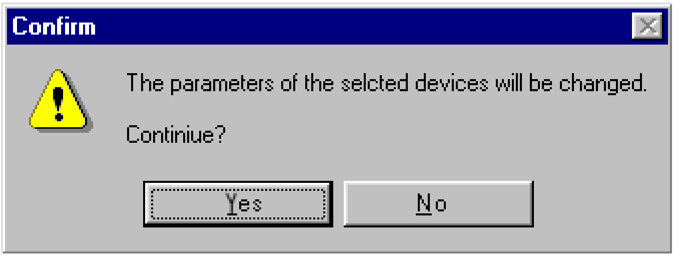
5. The list of all adaptable devices of the ETS project (see fig. 5) is displayed. Note that only devices of the same type and with the same number of channels are displayed. The devices to be adapted can then be selected (select multiples items by keeping the "Ctrl" key depressed).
   Selecting the *"All"* button now either activates or deactivates all devices together.
   After specifying the desired items, select *"Next"*.



Phys.addr.	Product	Description
01.01.003	push sensor 5 f	
01.01.004	push sensor 5 f	
01.01.005	push sensor 5 f	

Fig. 5: Display and selection of the ETS project devices to be adapted

6. The confirmation dialogue is diplayed (see fig. 6). When the button *"Yes"* is selected, the adaptation procedure in the ETS database is continued. *"No"* returns to the selection dialog (step 5).



Confirm

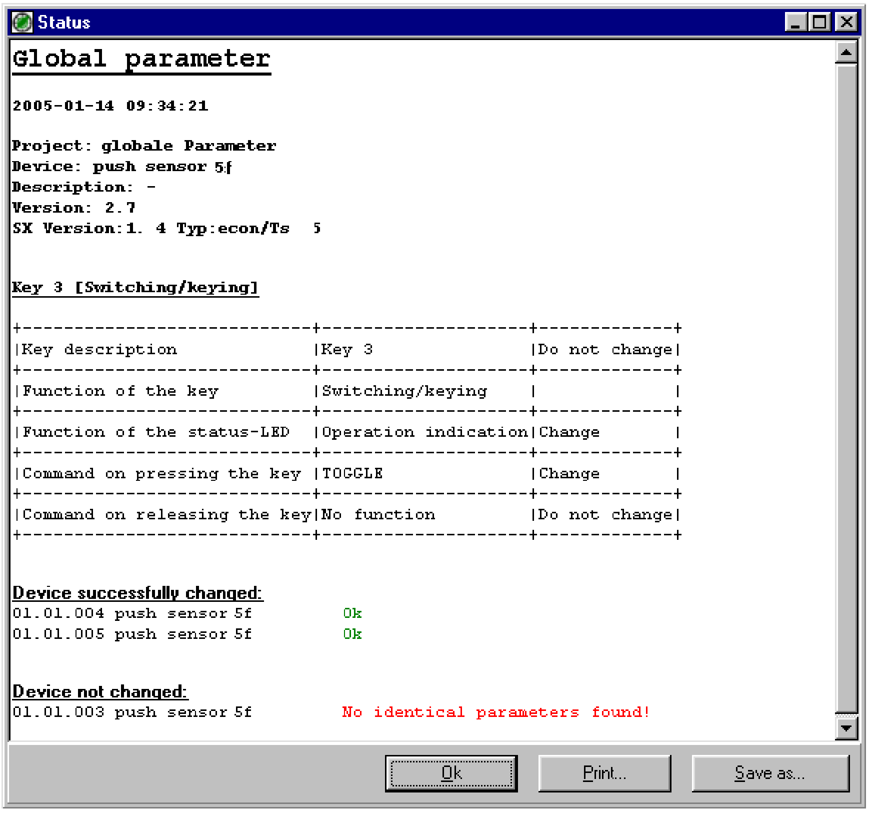
The parameters of the selcted devices will be changed.

Continiue?

Fig. 6: Confirmation dialog before changing the selected devices

7. The selected devices are automatically adapted one after another in the ETS database. Depending on the number of devices and parameters, this procedure may take some time. The progress of the adaptation procedure is indicated by a bargraph.
   By selecting the button *"Abort"*, the adaptation procedure can be stopped before completion (cf. step 8 'error messages in the status report').

8. At the end of the adaptation procedure, a status report gives an overview of all products adapted (see fig. 7) with a list of all devices successfully adapted and of those not adapted. For failed adaption attempts, the reason of the failure is displayed. Possible failure causes are shown in table 1. The status report can also be printed out or saved as an RTF file.
   "Ok" concludes the adaptation procedure and returns to the project window of the plug-in.



Status

```
Global parameter

2005-01-14 09:34:21

Project: globale Parameter
Device: push sensor 5f
Description: -
Version: 2.7
SX Version:1. 4 Typ:econ/Ts   5


Key 3 [Switching/keying]

+----------------------------+--------------------+-------------+
|Key description             |Key 3               |Do not change|
+----------------------------+--------------------+-------------+
|Function of the key         |Switching/keying    |             |
+----------------------------+--------------------+-------------+
|Function of the status-LED  |Operation indication|Change       |
+----------------------------+--------------------+-------------+
|Command on pressing the key |TOGGLE              |Change       |
+----------------------------+--------------------+-------------+
|Command on releasing the key|No function         |Do not change|
+----------------------------+--------------------+-------------+


Device successfully changed:
01.01.004 push sensor 5f          Ok
01.01.005 push sensor 5f          Ok


Device not changed:
01.01.003 push sensor 5f          No identical parameters found!
```

Fig. 7: Example of a status report after global parameter adaptation

Table 1: Possible error causes and remedial action

Error message	Error cause	Remedial action
No identical parameters found	The device to be changed does not have the parameters to be globally adapted.	Basically, the same parameters must be visible in all devices which are to be adapted. The parameter setting in the devices to be adapted must correspond to the basic parameterization in the reference device. Example: In push sensor A (reference product) key 1 is parameterized with the "switching" function. Push sensor B (target product) must then have the same function parameterized for key 1, i.e. "switching".
Device not processed	The device could not be processed because the user activated the *"Abort"* button while adapting of the device was in progress.	The adaptation procedure can be aborted at any time. Only the devices completely adapted by the time of abortion will remain adapted and stored in the ETS database. All other devices devices in the device list not yet modified remain unchanged.
Device not compatible	The device to be changed was parameterized or created by means of a plug-in software version not compatible with the current version.	All devices to be adapted must be created and edited with the same plug-in software. If this is not the case, an older version can often be adapted to the current one simply by opening and by closing the plug-in of the device concerned in the ETS. It is basically recommended to install the latest software compatible with the devices.
Error	An error occurred during ETS database access. The parameter field of the device concerned was not found.	In these cases, the ETS database has uncorrectible errors independent of the plug-in, preventing a modification of the device. Often, this problem can only be remedied by creating a new device in the ETS project.

9. Reconstruction / Reverse engineering

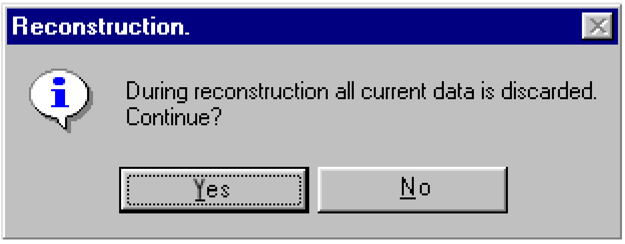
The reconstruction or reverse engineering item permits reading out data from programmed devices in an EIB installation. The parameter and object settings of the devices concerned can then be analyzed and stored in the ETS database. Non existing or lost projecting information can thus be restored in a simple way.
During the reconstruction procedure, an on-line bus link is established with the device. To ensure troublefree operation, a local connection to the bus must be existing and the device must be in place on the BCU.

By selecting the menu item *"Reconstruction – reconstruct the device"* in the plug-in of the device to be reconstructed, the device-related reconstruction procedure is started. Reconstruction presupposes that the device concerned is existing as such in the project of the ETS and that the physical address is known and has been entered.
As an alternative, an auxiliary module in the ETS 3 permits starting the reconstruction of a complete and possibly unknown project in the projecting window of the ETS. This auxiliary software module is not part of the basic ETS and must be bought from the ETS software distributor. For more details, please refer to the documentation of the respective supplier.

Reconstruction procedure (device-related reconstruction started in the plug-in)

The reconstruction requires the following two easy steps.

1. After initiating the reconstruction procedure, a confirmation request is displayed (see fig. 1). Confirming with *"Yes"* continues the procedure. Note that all device data existing in the database will be replaced by the data read out in the following procedure. Selecting *"No"* aborts the reconstruction without affecting the data stored in the database.

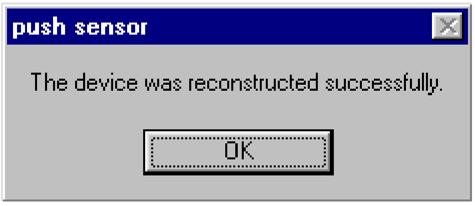


Reconstruction.

During reconstruction all current data is discarded.
Continue?

Fig. 1: Confirmation request before reconstruction

2. The reconstruction procedure is then started. Projecting data are read out step by step from the target device: target system identification, read-out of parameters, group address table, object settings and taking over of the values read out into the ETS database. The read-out progress is displayed by a bargraph.
When the device reconstruction is completed without errors, a confirmation dialog is displayed at the end of the procedure (see fig. 2).



push sensor

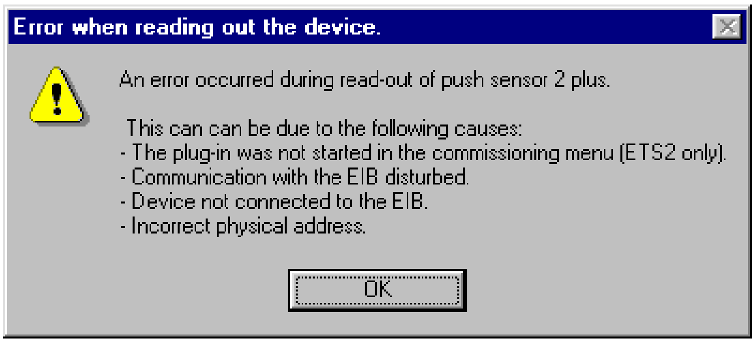
The device was reconstructed successfully.

Fig. 2: Confirmation dialog after successful reconstruction

Reconstruction info:

- Only parameters and other settings programmed directly into the device memory of the EIB component can be restored by a reconstruction procedure. For this reason, some of the plug-in settings such as text markers or function names cannot be reconstructed and will therefore be replaced during a reconstruction by the default values in the device database.
  The same applies to functions which were deactivated in the device at the time of reconstruction and which will be enabled in the plug-in only in the further course of the projecting task. Enabled parameters will also be set with their default values.
  It is always advisable to check all parameters, group addresses and settings for correctness after a reconstruction.

- During the reconstruction, the bus connection must not be interrupted. If the bus connection is disturbed, the reconstruction will be immediately aborted with an error message. The original values in the ETS database will then remain unchanged.
  If no communication with the device can be established at the beginning of the procedure, the plug-in reports a general communication error (see fig. 3).
  Remedial action: in the ETS 2, the reconstruction must be activated out of the commissioning menu in order to have access to the bus. With the help of the ETS diagnosis functions (line scan / press on programming button and reading out of devices in the programming mode) the user should be able to find out whether the device is connected to the bus and whether it can be contacted via its physical address.



Error when reading out the device.

An error occurred during read-out of push sensor 2 plus.

This can can be due to the following causes:
- The plug-in was not started in the commissioning menu (ETS2 only).
- Communication with the EIB disturbed.
- Device not connected to the EIB.
- Incorrect physical address.

Fig 3: General communication error during a reconstruction

Parameters

Description:	Values:	Comment:
🗁	push sensor 2 plus, (2fold, 3fold or 6fold)	
Push-button sensor function	Switched off **Switched on**	This parameter specifies whether the push-button sensor function is activated or deactivated.
Function of room temperature controller	Switched off **Switched on**	This parameter specifies whether the room temperature controller function is activated or deactivated.
Scene function	Switched off **Switched on**	This parameter specifies whether the scene function is activated or deactivated.
Control function 1	**Switched off** Switched on	This parameter specifies whether control function 1 is activated or deactivated.
Control function 2	**Switched off** Switched on	This parameter specifies whether control function 2 is activated or deactivated.
Alarm function after pulling off the user module	**Disabled** Enabled	When the push sensor 2 plus is pulled off the flush-mounted bus coupling unit an alarm message can be sent to the bus. This parameter specifies whether the alarm function is enabled or disabled.
Data format	**Switching telegram, 1-bit** Value telegram, 1-byte	To specify the data format of the alarm message.
Switching value	OFF **ON**	To specify the value of the switching telegram sent when an alarm message is being raised. Only for data format = "1 bit".
Value	0 to 255, **255**	To specify the value of the value telegram sent when an alarm message is being raised. Only for data format = "1 byte".
Reset value	**No** Yes	To specify whether the alarm value should be automatically reset to its inverse value (no alarm) after the application module has been re-plugged.
Lengh of operation display illumination	1 s 2 s **3 s**	It defines the period for which the status-LED will be lit to indicate actuation.

Display illumination		It specifies the function of the display illumination.
	OFF	The display illumination will always be OFF.
	ON	The display illumination will always be ON.
	Automatic switch-off	The display illumination switches on upon the pressing of a key and goes out after the parameterized time.
	Switching via object (object = 1 : ON)	The display illumination can be switched via an object. If a "1" is written into the object the description field light will be on (and vice versa).
	Switching via object (object = 1 : OFF)	The display illumination can be switched via an object. If a "0" is written into the object the description field light will be on (and vice versa).
Automatic illumination switch-off, base	0.5 s **1 s** 2 s 5 s	To specify the time base which defines the display illumination period for automatic switch-off after a key actuation. Time = base · factor
Automatic illumination switch-off, factor (1…255)	1 to 255, **10**	To specify the time factor which defines the display illumination period for automatic switch-off after a key actuation. Time = base · factor Presetting: 10 · 1 = 10 s
Operation LED		It specifies the function of the operation LED.
	OFF	The operation LED will always be OFF.
	ON	The display illumination will always be ON.
	Automatic switch-off	The operation LED switches on upon the pressing of a key and goes out after the parameterized time.
	Switching via object (object = 1 : ON)	The operation LED can be switched via an object. If a "1" is written into the object the description field light will be on (and vice versa).
	Switching via object (object = 1 : OFF)	The operation LED can be switched via an object. If a "0" is written into the object the description field light will be on (and vice versa).

Automatic illumination switch-off, operation LED base	0.5 s **1 s** 2 s 5 s	To specify the time base which defines the display illumination period for automatic switch-off after a key actuation. Time = base · factor
Automatic illumination switch-off, operation LED factor (1…255)	1 to 255, **10**	To specify the time factor which defines the operation LED period for automatic switch-off after a key actuation. Time = base · factor Presetting: 10 · 1 = 10 s
Access to operator levels		The push sensor 2 plus has several local operator levels:
	No operation	Local operation of the controller by rocker 1 actuation is not possible. Rocker 1 is disabled. Only normal operation without shifting the basic setpoint possible.
	First operator level	Switch-over to level 1 is possible. Thus, level 0 (setpoint shift/normal operation) and level 1 (operating mode switch-over) can be activated.
	All operator levels	Full access to the device with local operation of the controller. It allows the user additional access to the "contrast adjustment" and "setpoint assignment" (if enabled under "setpoints" in the ETS plug-in) functions at level 2.
📂	Display (2fold, 3fold and 6fold)	
Display	OFF **ON**	It activates or deactivates the display. A deactivated display will not read anything.
Display of	Outside temperature **Room temperature** Setpoint temperature Time Outside/clock Room/outside Room/outside/clock Room/clock Setpoint/outside Setpoint/outside/clock Setpoint/clock	Various pieces of information can be read by the display. You can select them from here. If you select several pieces of display information you can separately set the switch-over time.
Switch-over time (0.5 ... 127.5) * 1 sec	0.5 to 127.5, **10**	To specify the time interval after which the display information set by "display of" are switched over. Presetting: 10 · 1 = 10 s

📂	Push-button sensor function / Disable (2fold, 3fold and 6fold)	
Inhibit behaviour		This parameter defines the behaviour of the push sensor 2 plus when the inhibit function is active.
	Push-button sensor not disabled	The inhibit function is deactivated.
	Performance of all rockers like rocker 2…n	When the inhibit function is active all rockers of the push sensor 2 plus will behave in the same way as the parameterized one.
	Single rocker disabled	When the inhibit function is active specific individual rockers of the push sensor 2 plus can be disabled.
	Push-button sensor disabled	When the inhibit function is active the entire push sensor will be disabled.
Inhibit object polarity	**Not inverted (inhibit = 1)** Inverted (inhibit = 0)	To specify the inhibit object polarity.
Performance like rocker	**Rocker 2** (2fold, 3fold, 6fold) Rocker 3 (2fold, 3fold, 6fold) Rocker 4 (3fold, 6fold) Rocker 5 (6fold) Rocker 6 (6fold)	When the inhibit function is active all rockers of the push sensor 2 plus will behave in the same way as the parameterized one. Only for inhibit behaviour = " Performance of all rockers like rocker 2…n ".

Rocker X disabled?
X = 2 to 3 (2fold)
X = 2 to 6 (6fold)
	No
	Yes
		To specify whether rocker X will be disabled when the inhibit function is active i. e. key actuation (left and right) of this rocker will show no function.
		Only for inhibit behaviour = " Single rocker disabled "

📂	Push-button sensor function / Operation (2fold, 3fold and 6fold)	

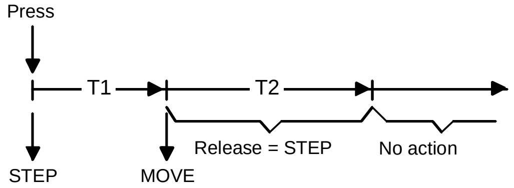
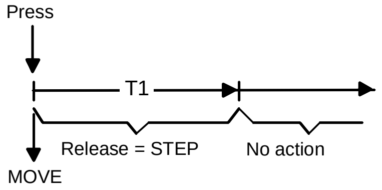
Rocker X
X = 2 to 3 (2fold)
X = 2 to 6 (6fold)
Rocker or key function
		For the push sensor 2 plus, two key functions or one rocker function can be assigned to each of the individual rockers.

	Keys	Two independent key functions are assigned to rocker X.
	Rocker	A rocker function is assigned to rocker X.
	Without function	Rocker X has no function, i. e. key actuation (left or right) will have no effect, and the status-LEDs of this rocker cannot be triggered.

🗁	Push-button sensor function - operation - Key 3 (2fold, 3fold and 6fold)	
Functions of key 3	No function Switching/keying Dimming Shutter/blind Value transmitter Analog value transmitter Operation of room temperature controller Light scene extension/recall Heating timer operation * Control function operation **	To specify the function of key 3. *: Heating timer operation can only be parameterized when the heating timer has been enabled. **: Control function operation can only be parameterized when at least one control function has been enabled.
Key 3 function = "no function"		
Function of the status-LED		For key 3 function = "no function", only the status-LED of the key can be triggered via the corresponding object. Key actuation will show no response.
	Always OFF	The status-LED will always be OFF.
	Always ON	The status-LED will always be ON.
	Status indication (switching object)	The status-LED will indicate the object status.
	Inverted status indication (switching object)	The status-LED will indicate the inverted object status.
Key 3 function = "switching/keying"		
Function of the status-LED		To specify the function of the status-LED.
	Always OFF	The status-LED will always be OFF.
	Always ON	The status-LED will always be ON.
	Status indication (switching object)	The status-LED will indicate the object status.
	Inverted status indivation (switching object)	The status-LED will indicate the inverted object status.
	Operation indication	When the key is actuated the status-LED will light up for the parameterized time.
Command on pressing the key	No function ON **TOGGLE** OFF	To specify the command sent when the key is being pressed.
Command on releasing the key	**No function** ON OFF TOGGLE	To specify the command sent when the key is being released.

Key 3 function = "dimming"		
Function of the status-LED	 Always OFF Always ON Status indication (switching object) Inverted status indication (switching object) **Operation indication**	To specify the function of the status-LED. The status-LED will always be OFF. The status-LED will always be ON. The status-LED will indicate the object status. The status-LED will indicate the inverted object status. When the key is actuated the status-LED will light up for the parameterized time.
Command on pressing the key, key function	 Darker (OFF) Brighter (ON) **Brighter/darker (TOGGLE)**	To specify the response to key actuation. Short-time key actuation will release an OFF telegram, whereas long-time key actuation will cause a dimming telegram (darker). Short-time key actuation will release an ON telegram, whereas long-time key actuation will cause a dimming telegram (brighter). The internally stored switching status will be toggled when the key is pressed shortly. If the stored status is ON (OFF) an OFF (ON) telegram will be raised. Pressing the key for a long time a "darker" telegram will be sent after a "brighter" telegram, and vice versa.
Increase brightness by	**100 %** 6 % 50 % 3 % 25 % 1.5 % 12.5 %	To specify the maximum dimming step width of a dimming telegram. With a dimming telegram, you can increase the brightness by a maximum of X %. This parameter depends on the key function selected.
Reduce brightness by	**100 %** 6 % 50 % 3 % 25 % 1.5 % 12.5 %	To specify the maximum dimming step width of a dimming telegram. With a dimming telegram, you can reduce the brightness by a maximum of X %. This parameter depends on the key function selected.
Stop telegram	**Yes** No	Releasing the key will send or not send a stop telegram.

Time between switching and dimming, base	**100 ms** 200 ms	Time from which on the function of a long-time actuation (dimming) will be executed. Time = base · factor
Time between switching and dimming, factor (1...255)	1 to 255, **4**	Time from which on the function of a long-time actuation (dimming) will be executed. Time = base · factor Presetting: 100 ms ·4 = 400 ms
Telegram repetition	**No** Yes	Cyclic dimming telegram repetition during key actuation.
Time between two dimming telegrams	**200 ms** 750 ms 300 ms 1 s 400 ms 2 s 500 ms	Time between two telegrams when telegram repetition has been set. Each time this period has elapsed, a new dimming telegram will be sent. Only for telegram repetition = "yes".
Key 3 function = "shutter/blind"		
Function of the status-LED		To specify the function of the status-LED.
	Always OFF	The status-LED will always be OFF.
	Always ON	The status-LED will always be ON.
	Operation indication	When the key is actuated the status-LED will light up for the parameterized time.

Operation concept (sequence of telegrams)	**Step - move - step** Move - step	To specify the telegram sequence after key actuation. Step - move - step:  Press T1 T2 Release = STEP No action STEP MOVE When the key is pressed a STEP telegram will be sent and start time T1 (time between short-time and long-time actuation) started. If you release the key within time T1 no further telegram will be sent. This STEP telegram serves for stopping an ongoing continuous run. If the key remains pressed for longer than time T1 a MOVE telegram will be automatically sent after T1 has elapsed, and time T2 (lamella adjustment time) will be started. If you then release the key within T2 a STEP telegram will be sent. This function is used for lamella adjustment. T2 should correspond to a 180° lamella rotation. Move - step:  When the key is pressed a MOVE telegram will be sent and start time T1 (lamella adjustment time) started. If you then release the key within T1 a STEP telegram will be sent. This function is used for lamella adjustment. T1 should correspond to a 180° lamella rotation.
Time between short- and long-time operation, base	**100 ms** 200 ms 500 ms	Time from which on the function of a long-time key actuation be executed. Only for operation concept = "step - move - step". Time = base · factor
Time between short- and long-time operation, factor (1…255)	1 to 255, **3**	Time from which on the function of a long-time key actuation be executed. Only for operation concept = "step - move - step". Time = base · factor Presetting: 100 ms · 3 = 300 ms

Shutter key function	**UP** DOWN	Short-time key actuation will release a STEP telegram (UP), whereas long-time key actuation will cause a MOVE telegram (UP). Short-time key actuation will release a STEP telegram (DOWN), whereas long-time key actuation will cause a MOVE telegram (DOWN).
Lamella adjustment time, base	**100 ms** 200 ms 500 ms	Time during which a lamella-adjusting move telegram can be stopped by releasing the key. Time = base · factor
Lamella adjustment time, factor (0...255)	0 to 255, **6**	Time during which a lamella-adjusting move telegram can be stopped by releasing the key. Time = base · factor Presetting: 100 ms · 6 = 600 ms
Key 3 function = "value transmitter"		
Function of the status-LED	 Always OFF Always ON **Operation indication**	To specify the function of the status-LED. The status-LED will always be OFF. The status-LED will always be ON. When the key is actuated the status-LED will light up for the parameterized time.
Value (0...255)	0 to 255, **255**	To specify the value to be sent.
Adjustment of values by long pressing of a key	**Disabled** Enabled	If the key remains pressed for at least 5 s the current value will be cyclically reduced by the parameterized step width and sent. After the key is released, the value sent last will be kept stored. This parameter specifies whether value changing will be possible.
Time between two telegrams	0,5 s; **1 s**; 2 s; 3 s	Time between two cyclic telegrams when the key is pressed for a long time.
Step width (1…10)	1 to 10, **10**	Step width by which the value set will be decreased upon long-time key actuation.

Key 3 function = "analog value transmitter"		
Function of the status-LED	 Always OFF Always ON **Operation indication**	To specify the function of the status-LED. The status-LED will always be OFF. The status-LED will always be ON. When the key is actuated the status-LED will light up for the parameterized time.
Function as	**Temperature value transmitter** Brightness value transmitter Value transmitter	To specify the function to be executed.
Temperature value (0...40) * 1 °C	0 to 40 °C in 1 °C steps, **25 °C**	To set the temperature value to be sent. Only for function as = "temperature value transmitter".
Brightness value (0...1500) * 1 lux	0 to 1500 lux in 50 lux steps, **500 lux**	To set the brightness value to be sent. Only for function as = "brightness value transmitter".
Value (0...0.65535)	0 to 65535, **0**	To set the EIS 10 value to be sent. Only for function as = "value transmitter".
Adjustment of values by long pressing of a key	**Disabled** Enabled	If the key remains pressed for at least 5 s the current value will be cyclically reduced by the parameterized step width and sent. After the key is released, the value sent last will be kept stored. This parameter specifies whether value changing will be possible.
Time between two telegrams	0.5 s; **1 s**; 2 s; 3 s	Time between two cyclic telegrams when the key is pressed for a long time.
Step width	Temperature value transmitter: **1 °C** Brightness value transmitter: **50 lux** Value transmitter 1 2 5 **10** 20 50 75 100 200 500 750 1000	Step width by which the value set will be decreased upon long-time key actuation.

Key 3 function = "operation of room temperature controller"		
Function of the status-LED		To specify the function of the status-LED.
	Always OFF	The status-LED will always be OFF.
	Always ON	The status-LED will always be ON.
	Key function indication active	The status-LED will shine as soon as the function assigned to it has been executed or activated by the actuation of the key.
	Key function indication inactive	The status-LED will shine as soon as the function assigned to it has been deactivated by the actuation of the key.
	Actuation indicator	When the key is actuated the status-LED will light up for the parameterized time.
Reaction on pressing a key	**Comfort operation** Standby operation Economy operation Frost/heat protection operation Presence key *	To specify the room temperature controller function to be activated by the actuation of the key. *: The presence key can only be parameterized when the "presence detection by presence key" option has been enabled.
Key 3 function = "light scene extension/recall"		
Function of the status-LED		To specify the function of the status-LED.
	Always OFF	The status-LED will always be OFF.
	Always ON	The status-LED will always be ON.
	Operation indication	When the key is actuated the status-LED will light up for the parameterized time.
Function as		To define scene recalling.
	Light scene extension	To recall an 'external' light scene via the light scene extension object.
	Internal scene recall	To recall an 'internal' scene. This setting can only be parameterized when the light scene function of the push sensor 2 plus has been enabled.
Light scene (1…64)	1 to 64, **1**	To specify the light scene number to be sent via the object. Only for "function as = light scene extension".
Scene (1…8)	1 to 8, **1**	To specify the number of the internal scene to be recalled. Only for "function as = internal scene recall".

Memory function	**No** Yes	This parameter will enable the storage function. When this function is enabled, long-time key actuation (> 5 s) will transmit a storage telegram or save the internal scene according to the parameterized number, respectively.
Key 3 function = "heating timer operation"		
Function of the status-LED		To specify the function of the status-LED.
	Always OFF	The status-LED will always be OFF.
	Always ON	The status-LED will always be ON.
	Key function indication active	The status-LED will shine as soon as the function assigned to it has been activated by the actuation of the key.
	Key function indication inactive	The status-LED will shine as soon as the function assigned to it has been deactivated by the actuation of the key.
	Operation indication	When the key is actuated the status-LED will light up for the parameterized time.
Reaction on pressing a key	Activate heating timer (ON)	Actuating the key will activate the heating timer. The parameterized switching programs will be executed.
	Deactivate heating timer (OFF)	Actuating the key will deactivate the heating timer. The parameterized switching programs will not be executed or will be suppressed, respectively.
	Deactivate / activate heating timer (TOGGLE)	Actuating the key will activate or deactivate the heating timer. Switch-over between executing and suppressing the switching programs.
Key 3 function = "control function operation"		
Function of the status-LED		To specify the function of the status-LED.
	Always OFF	The status-LED will always be OFF.
	Always ON	The status-LED will always be ON.
	Key function indication active	The status-LED will shine as soon as the function assigned to it has been activated by the actuation of the key.
	Key function indication inactive	The status-LED will shine as soon as the function assigned to it has been deactivated by the actuation of the key.
	Operation indication	When the key is actuated the status-LED will light up for the parameterized time.

Function	Both control functions enabled: **Control function 1** Control function 2 Only control function 1 enabled: **Control function 1** Only control function 2 enabled: **Control function 2**	Depending on which control function(s) has (have) been enabled, this parameter specifies which of the control function(s) should be triggered by the key function.
Reaction on pressing a key	Activate control function (ON)	Actuating the key will activate the corresponding control function. The parameterized switching programs or temperature control, respectively, will be executed.
	Deactivate control function (OFF)	Actuating the key will deactivate the corresponding control function. The parameterized switching programs or temperature control, respectively, will not be executed, or will be suppressed.
	Deactivate / activate control function (TOGGLE)	Actuating the key will activate or deactivate the corresponding control function. Switch-over between executing and suppressing the switching programs or of temperature control, respectively.
📂	For push-button sensor function - operation - key 4, refer to key 3 (dual and quintuple type).	
📂	For push-button sensor function - operation - key 5, refer to key 3 (dual and quintuple type).	
📂	For push-button sensor function - operation - key 6, refer to key 3 (dual and quintuple type).	
📂	For push-button sensor function - operation - key 7, refer to key 3 (quintuple type only).	
📂	For push-button sensor function - operation - key 8, refer to key 3 (quintuple type only).	
📂	For push-button sensor function - operation - key 9, refer to key 3 (quintuple type only).	
📂	For push-button sensor function - operation - key 10, refer to key 3 (quintuple type only).	
📂	Push sensor function - operation - rocker 2 (dual and quintuple type)	
Function of the rocker	No function **Switching** Dimming Shutter/blind Universal value transmitter Operation of room temperature controller	To specify the function of rocker 2.
Rocker 2 function = "no function"		
For rocker 2 function = "no function", only the status-LED of the rocker can be triggered via the corresponding status object. Rocker or key actuation will show no response. Status-LED parameter only (refer to "rocker 2 status").		

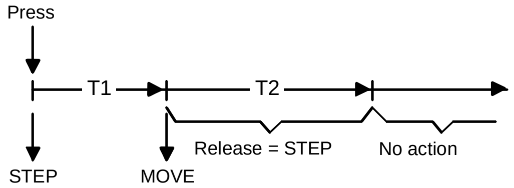
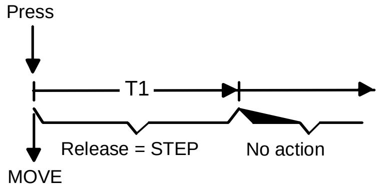
Rocker 2 function = "switching"		

Command on pressing a rocker	Left = ---, right = ---
	Left = OFF, right = ON
	Left = ON, right = OFF
	Left = TOGGLE, right = TOGGLE
		To specify the commands to be sent when both keys are pressed.

Rocker 2 function = "dimming"		
Command on pressing a rocker		To specify the response to rocker key actuation.
	Left = brighter (ON), **Right = darker (OFF)**	Short-time key actuation (left key) will release an ON telegram, while long-time actuation (left key) will cause a dimming telegram (brighter). Short-time key actuation (right key) will release an OFF telegram, while long-time actuation (right key) will cause a dimming telegram (darker).

	Left = darker (OFF),
	Right = brighter (ON)
		Short-time key actuation (left key) will release an OFF telegram, while long-time actuation (left key) will cause a dimming telegram (darker). Short-time key actuation (right key) will release an ON telegram, while long-time actuation (right key) will cause a dimming telegram (brighter).

	Left = TOGGLE, right = TOGGLE	The internally stored switching status will be toggled when the key is pressed shortly. If the stored status is ON (OFF) an OFF (ON) telegram will be raised. Pressing the key for a long time a "darker" telegram will be sent after a "brighter" telegram, and vice versa.
Increase brightness by	**100 %** 6 % 50 % 3 % 25 % 1.5 % 12.5 %	To specify the maximum dimming step width of a dimming telegram. With a dimming telegram, you can increase the brightness by a maximum of X %.
Reduce brightness by	**100 %** 6 % 50 % 3 % 25 % 1.5 % 12.5 %	To specify the maximum dimming step width of a dimming telegram. With a dimming telegram, you can reduce the brightness by a maximum of X %.
Stop telegram	**Yes** No	Releasing one of the keys (left or right) will send or not send a stop telegram.
Time between switching and dimming, base	**100 ms** 200 ms	Time from which on the function of a long-time actuation (dimming) will be executed. Time = base · factor
Time between switching and dimming, factor (1...255)	1 to 255, **4**	Time from which on the function of a long-time actuation (dimming) will be executed. Time = base · factor Presetting: 100 ms ·4 = 400 ms

Telegram repetition	**No** Yes	Cyclic dimming telegram repetition during key actuation.
Time between two dimming telegrams	**200 ms** 750 ms 300 ms 1 s 400 ms 2 s 500 ms	Time between two telegrams when telegram repetition has been set. Each time this period has elapsed, a new dimming telegram will be sent. Only for telegram repetition = "yes".
Rocker 2 function = "shutter/blind"		
Operation concept (sequence of telegrams)	**Step - move - step** Move - step	To specify the telegram sequence after key actuation. Step - move - step:  Press T1 T2 Release = STEP No action STEP MOVE When the key is pressed a STEP telegram will be sent and start time T1 (time between short-time and long-time actuation) started. If you release the key within time T1 no further telegram will be sent. This STEP telegram serves for stopping an ongoing continuous run. If the key remains pressed for longer than time T1 a MOVE telegram will be automatically sent after T1 has elapsed, and time T2 (lamella adjustment time) will be started. If you then release the key within T2 a STEP telegram will be sent. This function is used for lamella adjustment. T2 should correspond to a 180° lamella rotation. Move - step:  When the key is pressed a MOVE telegram will be sent and start time T1 (lamella adjustment time) started. If you then release the key within T1 a STEP telegram will be sent. This function is used for lamella adjustment. T1 should correspond to a 180° lamella rotation.

Time between short-time and long-time actuation, base	**100 ms** 200 ms 500 ms	Time from which on the function of a long-time key actuation will be executed. Only for operation concept = "step - move - step". Time = base · factor
Time between short-time and long-time actuation, factor (1…255)	1 to 255, **3**	Time from which on the function of a long-time key actuation will be executed. Only for operation concept = "step - move - step". Time = base · factor Presetting: 100 ms · 3 = 300 ms
Command on pressing a rocker	**Left - shutter/blind UP / right - shutter/blind down**	Short-time key actuation (left key) will release a STEP telegram (UP), whereas long-time actuation (left key) will cause a MOVE telegram (UP). Short-time key actuation (right key) will release a STEP telegram (DOWN), whereas long-time actuation (right key) will cause a MOVE telegram (DOWN).
	Left - shutter/blind DOWN / right - shutter/blind UP	Short-time key actuation (left key) will release a STEP telegram (DOWN), whereas long-time actuation (left key) will cause a MOVE telegram (DOWN). Short-time key actuation (right key) will release a STEP telegram (UP), whereas long-time actuation (right key) will cause a MOVE telegram (UP).
Lamella adjustment time, base	**100 ms** 200 ms 500 ms	Time during which a lamella-adjusting move telegram can be stopped by releasing the key. Time = base · factor
Lamella adjustment time, factor (0…255)	0 to 255, **6**	Time during which a lamella-adjusting move telegram can be stopped by releasing the key. Time = base · factor Presetting: 100 ms · 6 = 600 ms

Rocker 1 function = "universal value transmitter"		
Left key effect	**Run from start to target value (right key opposite)** Run from target to start value (right key opposite)	This parameter defines the direction of control action, i. e. it can be specified whether the start or the target value should be sent first when the left key of the rocker is being actuated. The right key always acts in the opposite way to the left one.
Reaction on pressing a key		To specify how the universal value transmitter can be started or stopped, respectively.
	Continuous run as long as key is kept pressed	As long as the left or the right key is pressed the universal value transmitter will send values to the bus. After both sub-ranges have been swept and the key is still held pressed no more values will be transmitted.
	Start / stop continuous run	If the left or right key is pressed the universal value transmitter will send values to the bus, even though the key is released. Only the next key actuation will stop the universal value transmitter. After both sub-ranges have been swept no more values will be sent. In such case, actuating the key again will restart the continuous run.
Number of steps (per sub-range) (1..15)	1 to 15, **10**	To define the number of steps in sub-ranges 1 and 2.
Continuous running time, base	**approx. 0.5 s** approx. 1 s approx. 2 s approx. 3 s approx. 5 s	You can parameterize a time for each step in the sub-ranges. This time defines at what time intervals the next step will be reached or transmitted, respectively. To specify the common time base for both sub-ranges. Time = base · factor
Time factor for sub-range 1 (1..255)	1 to 255, **10**	You can parameterize a time for each step in the sub-ranges. This time defines at what time intervals the next step will be reached or transmitted, respectively. To specify the time factor of sub-range 1. Time = base · factor Presetting: 10 · 0.5 s = 5 s
Time factor for sub-range 2 (1..255)	1 to 255, **10**	You can parameterize a time for each step in the sub-ranges. This time defines at what time intervals the next step will be reached or transmitted, respectively. To specify the time factor of sub-range 2. Time = base · factor Presetting: 10 · 0.5 s = 5 s

Start value (0…253)	0 to 253, **0**	To specify the start value.
Back-up value (1…254)	1 to 254, **127**	To specify the back-up value.
Target value (2...255)	2 to 255, **255**	To specify the target value.
Rocker 2 function = "Operation of room temperature controller"		
Command on pressing a rocker	**Switch-over between the operating modes comfort, standby, economy-operation and frost/heat protection** Nor other settings possible.	To specify the function of rocker 2.
📂 For push-button sensor function - operation - rocker 3, refer to rocker 2 (2fold, 3fold and 6fold).		
📂 For push-button sensor function - operation - rocker 4, refer to rocker 2 (3fold and 6fold).		
📂 For push-button sensor function - operation - ocker 5, refer to rocker 2 (6fold only).		
📂 For push-button sensor function - operation - ocker 6, refer to rocker 2 (6fold only).		
📂 For push-button sensor function - operation - ocker 7, refer to rocker 2 (6fold only).		
📂 Push-button sensor function - operation - rocker 2 - rocker 2 status (2fold, 3fold and 6fold)		
Rocker 2 function = "no function", "switching", "dimming", "shutter/blind" and "operation of room temperature controller"		
Show status object via	**Left and right status-LEDs** Left status-LED Right status-LED Inverted left and right status-LEDs Inverted left status-LED Inverted right status-LED Left and right LED always ON Left and right LED always OFF	To define the function of the status-LED.
📂 For push-button sensor function - operation - rocker 3 - rocker 3 status, refer to rocker 2 (2fold, 3fold and 6fold).		
📂 For push-button sensor function - operation - rocker 4 - rocker 4 status, refer to rocker 2 (3fold and 6fold).		
📂 For push-button sensor function - operation - rocker 5 - rocker 5 status, refer to rocker 2 (6fold only).		
📂 For push-button sensor function - operation - rocker 6 - rocker 6 status, refer to rocker 2 (6fold only).		
📂 For push-button sensor function - operation - rocker 7 - rocker 7 status, refer to rocker 2 (6fold only).		
📂 Room temperature controller function		
Operating mode switch-over	**Via value (byte)** Via switching (4 x 1 bit)	Switching over the operating modes via the bus proceeds by a 1-byte object in accordance with the KONNEX specification. In addition, a higher-order constraint object is available for this setting. The operating modes will be switched over via the bus in the 'classical' way through separate 1-bit objects.
Control loops (HA (high access))	**One control loop** Two control loops	The room temperature controller will trigger one control loop only. The room temperature controller can trigger up to two control loops.

Heating/cooling mode (HA)	**Heating** Cooling Heating and cooling * Basic and additional heating * Basic and additional cooling * Basic/additional heating/cooling *	To set the heating/cooling switch-over. *: "The mixed "heating and cooling" mode and two-stage control operation will not be possible when two control loops are used.
Additional stage inhibit object (HA)		The additional stages can be separately disabled via the bus. The parameter enables the inhibit object.
	No	The additional stages cannot be separately disabled.
	Yes	The additional stages can be disabled via the inhibit object. Only for two-stage heating or cooling operation.
Send variable heating and cooling to one common object (HA)	**No** Yes	If the parameter is set to "yes" the heating or cooling variable will be sent to a common object. This function can be used if the same heating system in the room is used for cooling in summer and for heating in winter. Only for mixed "heating and cooling" mode using additional stages, if necessary.
Type of heating control (for basic and additional stage, if necessary) (HA)	**Continuous PI control** Switching PI control (PWM) Switching 2-point control (ON/OFF)	To select a control algorithm (PI or 2-point) with data format (1 byte or 1 bit) for the heating system.
Type of heating (for basic and additional stage, if necessary) (HA)	**Warm water heating (5 K / 150 min.)** Floor heating (5 K / 240 min.) Floor heating (4 K / 100 min.) Blower convector (4 K / 90 min.) Split unit (4 K / 90 min.)	To match the PI algorithm to various heating systems with experience values for the proportional band and reset time control parameters
	Via control parameter	Separate control parameter input. Only for "type of heating control" = "PI".
Heating proportional band (10 ... 127) * 0.1 K (HA)	10...127, **50**	To separately set the "proportional band" control parameter. Only for "type of heating control" = "via control parameter".

Heating reset time (0 … 255) * 1 min.; 0 = inactive (HA)	0...255, **150**	To separately set the "reset time" control parameter. Only for "type of heating control" = "via control parameter".
Heating 2-point-controller hysteresis upper limit (5 ... 127) * 0.1 K (HA)	5...127, **5**	To define the heating switch-on and switch-off times. Only for "type of heating control" = "2-point".
Heating 2-point-controller hysteresis lower limit (-128 ... -5) * 0.1 K (HA)	-128...-5, **-5**	To define the heating switch-on and switch-off times. Only for "type of heating control" = "2-point".
Type of cooling control (for basic and additional stage, if necessary) (HA)	**Continuous PI control** Switching PI control (PWM) Switching 2-point control (ON/OFF)	To select a control algorithm (PI or 2-point) with data format (1 byte or 1 bit) for the cooling system. Only for "send heating and cooling variables to a common object" = "no". If "send heating and cooling variables to a common object" = "yes" the "type of heating control" parameter settings will be taken over for cooling operation.
Type of cooling (for basic and additional stage, if necessary) (HA)	**Cooling ceiling (5 K / 240 min.)** Blower convector (4 K / 90 min.) Split unit (4 K / 90 min.) Via control parameter	To match the PI algorithm to various cooling systems with experience values for the proportional range and reset time control parameters. Separate control parameter input. Only for "type of cooling control" = "PI".
Cooling proportional band (10 ... 127) * 0.1 K (HA)	10...127, **50**	To separately set the "proportional band" control parameter. Only for "type of cooling" = "via control parameter".
Cooling reset time (0 ... 255) * 1 min.; 0 = inactive (HA)	0...255, **240**	To separately set the "reset time" control parameter. Only for "type of cooling" = "via control parameter".
Cooling 2-point-controller hysteresis upper limit (5 ... 127) * 0.1 K (HA)	5...127, **5**	To define the cooling switch-on and switch-off times. Only for "type of cooling control" = "2-point".
Cooling 2-point-controller hysteresis lower limit (-128 ... –5) * 0.1 K (HA)	-128...-5, **-5**	To define the cooling switch-on and switch-off times. Only for "type of cooling control" = "2-point".

Operation controller inhibitable	 **No** Always disabled Via bus	You can disable room temperature controller local operation (all rockers). Activated disabling of operation will be indicated in the display by the blinking " 🔒 " symbol. Inhibit function deactivated. To always disable the operation of the controller. To enable object 44, "disable controller operation".
Switch off controller (dew point operation)	**No** Via bus	This parameter enables object 45, "disable controller". There will be no more control action until enabling (variables = 0).
Frost/heat protection	 Automatic frost protection **Via window status**	You can specify the way how the controller will switch over into the frost/heat protection mode. Automatic frost protection in activated. This can effect automatic switch-over into the frost protection mode, depending on the room temperature. The switch-over into the frost/heat protection mode proceeds via the "window status" object.
Automatic frost protection	**OFF** 0.2 K / min. 0.3 K / min. 0.4 K / min. 0.5 K / min. 0.6 K / min.	To specify the temperature decrease the room temperature must reduce within one minute before the controller switches over into the frost protection mode. When "off" is selected automatic frost protection will be deactivated. Only for "frost/heat protection = automatic frost protection".
Frost protection period in automatic mode (1..255) * 1 min.	1 to 255, **20**	To define the time after which the controller will automatically deactivate frost protection when it is in the automatic frost protection mode. Only for enabled automatic frost protection.
Window status delay (0..255) * 1 min.; 0 = inactive	0 to 255, **0**	To define the delay time after which the frost/heat protection mode will be activated by the window status. Only for "frost/heat protection = via window status".

Switch-over between heating and cooling	 **Automatically** Via object	If heating/cooling switch-over has been parameterized you can change between heating and cooling. Depending on the operating mode and the room temperature, this switch-over is automatic. The switch-over takes proceeds solely via the "heating/cooling switch-over" object 40. Only for the "heating and cooling" or "basic/ additional heating/cooling" mixed mode.
Heating/cooling switch-over after reset	**Heating** Cooling Heating/cooling switch-over before reset	To specify the preset heating/cooling switch-over after reset. Only for "switch-over between heating and cooling = via object".
Automatic heating/cooling switch-over transmission	**On heating/cooling switch-over change** On changing the output value	To specify when a heating/cooling switch-over telegram will be automatically transmitted to the bus via the "heating/cooling switch-over" object 40. Only for "switch-over between heating and cooling = automatically".
Cyclicyt transmission heating/cooling switch-over (0…255) * 1 min.; 0 = inactive	0 to 255, **0**	To specify whether the current status of object 40, "heating/cooling switch-over", should be cyclically transmitted to the bus when switching over is automatic. You can set the cycle time. Setting "0" will deactivate the cyclic transmission of the object value. Only for "switch-over between heating and cooling = automatically".
Valve protection	**No** Yes	The valve will be cyclically opened (every 24 hours). This is a precaution against deposits and prevents choking of the valve.

🗁 Room temperature controller function setpoints		
Own setpoints for control loop 2	**No** Yes	If you use both control loops the second control loop can have separate setpoints. Setting "yes" will enable setpoint presetting for the second control loop. Only for "Control loops = 2 control loops".
Basic temperature after reset (7.0 ... 40.0) * 1 °C	7.0 °C to 40 °C, **21 °C**	To specify the basic setpoint after initialization.
Basic temperature in the 2nd control loop after reset (7.0 ... 40.0) * 1 °C	7.0 °C to 40 °C, **21 °C**	To specify the basic setpoint of the second control loop after initialization. Only for "Own setpoints for second control loop = yes".
Accept modification of shift of basic setpoint permanently	**No** Yes	By the basic setpoint shift (through rocker 1), the temperature of the current operating mode of both control loops can be adapted. You can shift the temperature up or down within a preset value range. Setting "no" will delete the temperature shift when a change to a different operating mode takes place. Setting "yes" will keep the temperature shift when a change to a different operating mode takes place.
Modification of the basic temperature setpoint	**Deactivated** Permit at device Admit via bus Permit at device and via bus	To specify whether a basic temperature change of the first control loop will be possible via the bus or locally on the device.
Modification of the basic temperature setpoint in the 2nd control loop	**Deactivated** Admit via bus	To specify whether a basic temperature change of the second control loop will be possible via the bus or locally on the device. Only for "Own setpoints for second control loop = yes".
Accept modification of the basic temperature setpoint permanently	**No** Yes	This parameter specifies whether the basic temperature readjusted via the bus or locally on the device shall be stored in the memory permanently (setting "yes") or only temporarily setting "no"). If you select "yes" the changed basic value will be kept, even after an operating mode switch-over and a reset. Only for " Modification of the basic temperature setpoint = Permit at device", "Admit via bus" or "Permit at device and via bus".

1st control loop standby temperature change	**Deactivated** Permit at device	To facilitate the changing of the basic standby temperature of the first control loop on the device.
1st control loop economy temperature change	**Deactivated** Permit at device	To facilitate the changing of the economy temperature of the first control loop on the device.
Frost protection setpoint temperature (7...40) * 1 °C	7 °C to 40 °C, 7 °C	To specify the setpoint temperature when frost protection is active. Only for "Heating/cooling mode" = "heating" or "heating and cooling", with additional stage, if necessary.
Heat protection setpoint temperature (7...45) * 1 °C	7 °C to 45 °C, **35 °C**	To specify the setpoint temperature when heat protection is active. Only for "Heating/cooling mode" = "cooling" or "heating and cooling", with additional stage, if necessary.
Dead band position		The comfort heating and cooling setpoint temperatures are derived from the basic setpoint, taking account of the preset dead band. The dead band (temperature zone where neither heating nor cooling takes place) is the difference between the comfort setpoint temperatures.
	Symmetrical	Symmetrical: The preset dead band is divided into two ranges at the basic setpoint. From the resulting half of the dead band, the comfort setpoint temperatures are directly derived from the basic setpoint.
	Asymmetrical	Asymmetrical: For this setting, the comfort heating setpoint temperature is equal to the basic setpoint. The preset dead band will solely become effective from the basic setpoint towards the comfort cooling temperature. Thus, the comfort mode cooling setpoint temperature is directly derived from the comfort heating setpoint. Only for the "heating and cooling" or "basic/ additional heating/cooling" mixed mode.
Dead band between heating and cooling (0…127) * 0.1 K	0 to 127, **20**	The comfort heating and cooling setpoint temperatures are derived from the basic setpoint, taking account of the preset dead band. The dead band (temperature zone where neither heating nor cooling takes place) is the difference between the comfort setpoint temperatures. Only for the "heating and cooling" or "basic/ additional heating/cooling" mixed mode.

Dead band shift	**Deactivated** Permit at device	To specify whether the dead band and thus the comfort cooling temperature can be set on the device at operator level 2. Only for the "heating and cooling" or "basic/ additional heating/cooling" mixed mode.
Difference between basic and additional stages (0…127) * 0.1 K	0 to 127, **20**	For two-stage control operation, you must specify at what temperature difference compared to the basic stage you want to include the additional stage into control. Only for two-stage control operation.
Transmission at setpoint temperature modification by (0...255) * 0.1 K; 0 = no automatic transmission	0 to 255, **1**	To specify the setpoint change value by what the current value will be automatically sent to the bus via the "setpoint temperature" object. If you use both control loops with separate setpoints both setpoints can be transmitted.
Cyclical transmission of setpoint temperature (0…255) * 1 min.; 0 = inactive	0 to 255, **0**	To specify whether the setpoint temperature shall be output via the "setpoint temperature" object. If you use both control loops with separate setpoints both setpoints can be transmitted.
Upward adjustment of basic setpoint temperature (0...10) * 1 K	0 to 10, **3**	To specify the maximum adjusting value range you can set for shifting the basic setpoint temperature upward. (Refer to "Basic temperature setpoint change".)
Downward adjustment of basic setpoint temperature (-10...0) * 1 K	-10 to 0, **-3**	To specify the maximum adjusting value range you can set for shifting the basic setpoint temperature downward. (Refer to "Basic temperature setpoint change".)
Lower the setpoint temperature during standby operation (heating) (-128...0) * 0.1 K	-128 to 0, **-20**	To decrease the standby heating setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = heating" or "heating and cooling", with additional stages, if necessary.
Lower the setpoint temperature during economy operation (heating) (-128...0) * 0.1 K	-128 ... 0, **-40**	To decrease the economy heating setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = heating" or "heating and cooling", with additional stages, if necessary.

Lower the setpoint temperature during standby operation (heating - control loop 2) (-128...0) * 0.1 K	-128 to 0, **-20**	To decrease the standby heating setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = heating" and two control loops with separate setpoints.
Lower the setpoint temperature during economy operation (heating - control loop 2) (-128...0) * 0.1 K	-128 to 0, **-40**	To decrease the economy heating setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = heating" and two control loops with separate setpoints.
Raise the setpoint temperature during standby operation (cooling) (0..0.127) * 0.1 K	0 to 127, **20**	To increase the standby cooling setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = cooling" or "heating and cooling", with additional stages, if necessary.
Raise the setpoint temperature during economy operation (cooling) (0...127) * 0.1 K	0 to 127, **40**	To increase the economy cooling setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = cooling" or "heating and cooling", with additional stages, if necessary.
Raise the setpoint temperature during standby operation (cooling - control loop 2) (0...127) * 0.1 K	0 to 127, **20**	To increase the standby cooling setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = cooling" and two control loops with separate setpoints.
Raise the setpoint temperature during economy operation (cooling - control loop 2) (0...127) * 0.1 K	0 to 127, **40**	To increase the economy cooling setpoint temperature by this value, compared with the basic setpoint. Only for "Heating/cooling mode = cooling" and two control loops with separate setpoints.
🗁	Room temperature controller function - functionality	
Operating mode after reset	**Comfort operation** Standby operation Economy operation Frost/heat protection operation	Parameter to set the operating mode to be activated after the push sensor initialization phase (e. g. after bus voltage recovery).

Presence detection		When a person is present in the room it makes sense that the controller switches over into the comfort mode or into the comfort prolongation mode, respectively. To specify though which 'detectors' detection will be made.
	None	No presence detection.
	Presence key	Presence will be detected by a presence key on the push sensor (key function) or via the presence object (e. g. external push sensor).
	Presence detector	Presence will be detected via an external presence detector. The detector will be connected via the presence object.
Length of comfort prolongation (0...255) * 1 min.; 0 = off	0 to 255, **30**	During presence detection, the controller can temporarily switch over to comfort prolongation, depending on the active operating mode. The parameter specifies the time after which the comfort prolongation will be automatically terminated. Only for "presence detection = presence key".
🗁 Room temperature controller function - room temperature measuring (HA)		
Temperature detection		To specify which sensor will be used for room temperature measuring in the first control loop.
	Internal sensor	Internal sensor: Built-in sensor of the push sensor 2 plus.
	External sensor	External sensor: An external sensor connected via the bus, e. g. under difficult measuring conditions (in swimming baths or similar facilities).
	Internal and external sensor	Internal and external sensors: Both sensors are used, e. g. in large rooms. Only for one control loop.
Creating of measuring value internal against external	10 % vs. 90 % 20 % vs. 80 % 30 % vs. 70 % 40 % vs. 60 % **50 % vs. 50 %** 60 % vs. 40 % 70 % vs. 30 % 80 % vs. 20 % 90 % vs. 10 %	To specify the weighting of the measured temperature value of the internal and external sensor. This will form a resulting overall measured value used for the evaluation of the room temperature. Only for one control loop and "Temperature detection = internal and external sensor".

Adjustment internal sensor (-128...127) * 0.1 K	-128 to 127, **0**	To specify the value by which the measured room temperature value of the internal sensor is adjusted. Only for "Temperature detection = internal sensor" or "internal and external sensor", or for two control loops.
Adjustment external sensor (-128...127) * 0.1 K	-128 to 127, **0**	To specify the value by which the measured room temperature value of the external sensor is adjusted. Only for "Temperature detection = external sensor" or "internal and external sensor", or for two control loops.
Scanning time for external sensor (0…255) * 1 min.; 0 = inactive	0 to 255, **0**	To specify the temperature value scanning time of the external sensor. "0" = the sensor will automatically send its temperature value. Only for "Temperature detection = external sensor" or "internal and external sensor", or for two control loops.
Transmission at room temperature modification by (0..255) * 0.1 K; 0 = no automatic transmission	0 to 255, **3**	To specify the room temperature value change extent after which the current values will be automatically sent to the bus via object 28, "actual temperature".
Cyclical transmission of temperature (0…255) * 1 min.; 0 = inactive	0 to 255, **15**	To specify whether or at what intervals the measured room temperature of the first control loop shall be output via object 28, "actual temperature".
📂 Room temperature controller function - variable and status output (HA)		
Automatic transmission at modification by (0...100) * 1 %; 0 = inactive	0 to 100, **3**	To specify the variable change extent after which the continuous variables will be automatically transmitted via the variables objects. Only if at least one type of control has been parameterized to "continuous PI control".
Cycle time of the switching variable (1...255) * 1 min.	1 to 255, **15**	To specify the cycle time for the pulse-width-modulated (PWM) variable. Only if at least one type of control has been 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 cyclic transmission of the variable via the variables objects. Only if at least one type of control has been parameterized to "continuous PI control" or to "switching 2-point control".

Output of the heating variable	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Heating normal variable output. Only for "Heating/cooling mode = heating" or "heating and cooling".
Output of the heating variable in the 2nd control loop	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Control loop 2 heating normal variable output. Only for "Heating/cooling mode = heating" with two control loops.
Output of the heating basic stage variable	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Heating basic stage normal variable output. Only for "Heating/cooling mode = basic and additional heating" or "basic/additional heating/cooling".
Output of the heating additional stage variable	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Heating additional stage normal variable output. Only for "Heating/cooling mode = basic and additional heating" or "basic/additional heating/cooling".
Output of the cooling variable	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Cooling normal variable output. Only for "Heating/cooling mode = cooling" or "heating and cooling".
Output of the cooling variable in the 2nd control loop	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Control loop 2 cooling normal variable output. Only for "Heating/cooling mode = cooling" with two control loops.
Output of the cooling basic stage variable	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Cooling basic stage normal variable output. Only for "Heating/cooling mode = basic and additional cooling" or "basic/additional heating/cooling".

Output of the cooling additional stage variable	Inverted **Normal**	Continuous: variable = 100 % - normal variable. Switching: variable = 1 - normal variable. Cooling additional stage normal variable output. Only for "Heating/cooling mode = basic and additional cooling" or "basic/additional heating/ cooling".
Heating indication	**No** Yes	To enable the "heating indication" function and thus object 42, "heating indication". The signal is exclusively referred to the first control loop.
Cooling indication	**No** Yes	To enable the "cooling indication" function and thus object 43, "cooling indication". The signal is exclusively referred to the first control loop.
Satus controller	 **No status** Controller general Transmit individual state	The controller can output its current status. No status output. The controller status will generally be output via the 1-byte object (object 41, "controller status"). The controller status preset by the "individual status" parameter will be output via the 1-bit object (object 41, "controller status").
Single state	**Comfort operation activated** Standby operation activated Economy operation activated Frost/heat protection activeted Controller disabled Heating/cooling Controller inactivated Frost alarm	To specify the controller status to be transmitted. Only for "Status controller = transmit individual state".
📂 Room temperature controller function - heating timer		
Heating timer	on **off**	To enable the heating timer.
Disable heating timer through bus	 Yes **No**	The execution of the heating timer switching programs can be suppressed via the bus by the inhibit function. To enable the inhibit function and object 57, "disabling heating timer". The heating timer inhibit function is not deactivated.
Inhibit object polarity	Inverted (inhibit = 0) **Not inverted (inhibit = 1)**	To specify the heating timer inhibit object polarity. Only for "Disable heating timer through bus = yes".

📂 Scene function		
Data type Output 1	**Switching** Value Shutter/blind	To specify the data type of the scene output.
Value type	**0...100 %** 0...255	To define the value type for a 1-byte scene object. Depending on this setting, you can either set percentages or dimensionless values for the scene commands. Only for "data type = value".
Data type Outputs 2 to 8	Refer to output 1 data type.	
📂 Scene function - [1] scene 1		
Name	[Text], **scene 1**	Here, you can assign a name to the internal scene. This text will only be used in the ETS plug-in for better orientation and will not be downloaded into the device.
Send output	Yes **No**	To specify whether a scene command shall be transmitted via the selected scene output when the scene is being recalled.
Value	**ON** OFF	To define the switching value which will be sent to the bus when a scene is being recalled. Only for "send output = yes" and "data type = switching".
Value (0…100) * 1 %	0 to 100 %, **0 %**	To define the value which will be sent to the bus when a scene is being recalled. Only for "send output = yes", "data type = value" and "value type = 0...100 %".
Value (0…255)	0 to 255, **0**	To define the value which will be sent to the bus when a scene is being recalled. Only for "send output = yes", "data type = value" and "value type = 0...255".
Shutter/blind position	Up **Down**	To define the shutter/blind long-time command which will be sent to the bus when a scene is being recalled. Only for "send output = yes" and "data type = shutter/blind".
📂 Scene function - [X] scene X, for X = 2 to 8, refer to scene 1.		

🗀 Control function 1		
Function	**Temperature-dependent** Timer	To specify whether the control function shall be temperature-dependent or time-controlled.
Data type	**Switching** Value	To specify whether switching or value commands shall be sent to the bus when the control function is active.
Exceed basic temperature value (1...255) * 0.1 K	0.1 to 25.5 K, **1 K**	To specify the temperature deviation which must be higher than the basic setpoint of the first control loop before the parameterized control command will be sent to the bus. Only for "function = temperature-dependent".
Remain under basic temperature value (1…255) * 0.1 K	0.1 to 25.5 K, **1 K**	To specify the temperature deviation which must be lower than the basic setpoint of the first control loop before the parameterized control command will be sent to the bus. Only for "function = temperature-dependent".
Exceed telegram delay time (0…255) * 1 min.; 0 = inactive	0 to 255 minutes, **0 minutes**	Telegram transmission after exceeding the preset temperature value can be delayed. To specify the delay time after which the parameterized control command will be sent to the bus. Only for "function = temperature-dependent".
Remain undertelegram delay time (0…255) * 1 min.; 0 = inactive	0 to 255 minutes, **0 minutes**	Telegram transmission after falling below the preset temperature value can be delayed. To specify the delay time after which the parameterized control command will be sent to the bus. Only for "function = temperature-dependent".
Value when exceeding		To specify the control command which will be sent to the bus when the parameterized temperature value is exceeded.
	Send no value	To transmit no control command.
	0	To transmit an "OFF" switching command.
	1	To transmit an "ON" switching command. Only for "data type = switching" and "function = temperature-dependent".

Value when exceeding	 **Send no value** Value	To specify the control command which will be sent to the bus when the temperature exceeds its parameterized value. To transmit no control command. To transmit a value command. Only for "data type = value" and "function = temperature-dependent".
Value (0…255)	0 to 255, **0**	To specify the value command which will be sent to the bus. Only for "Value when exceeding = value".
Value when remaining under	 **Send no value** 0 1	To specify the control command which will be sent to the bus when the temperature falls below its parameterized value. To transmit no control command. To transmit an "OFF" switching command. To transmit an "ON" switching command. Only for "data type = switching" and "function = temperature-dependent".
Value when remaining under	 **Send no value** Value	To specify the control command which will be sent to the bus when the temperature falls below its parameterized value. To transmit no control command. To transmit a value command. Only for "data type = value" and "function = temperature-dependent".
Value (0…255)	0 to 255, **0**	To specify the value command which will be sent to the bus. Only for " Value when remaining under = value".
Inhibit object (e. g. sun sensor)	 Yes **No**	The execution of the control function can be suppressed via the bus by the inhibit function. To enable the inhibit function and the "disabling control function" object. The inhibit function of the control function is not enabled.
Inhibit object polarity	Inverted (inhibit = 0) **Not inverted (inhibit = 1)**	To specify the inhibit object polarity. Only for "inhibit object = "yes".

📂 For control function 2, refer to control function 1.

Remarks on the Software

› Parameter access

To be able to set all parameters of the push sensor 2 plus you must have set the access option in the ETS plug-in to "high access" (HA). To set the type of access select or deselect the "full access" menu item in the "configuration" menu.

› Dimming function (push sensor functionality)

For the correct functioning of the status-LEDs in the status indicator mode, the dimming actuator connected must send back its status to the switching object when the key function is active or to the status object when the rocker function is active (set the Ü flag on the actuator).
For correct functioning in the key function mode (brighter/darker (TOGGLE)), the dimming actuator connected must also send back its status to the switching object.
For the key or rocker function, only the switching object will be followed up internally and externally. The dimming object (dimming direction) will be followed up only internally so that the dimming direction will not always be reversed upon another pressing of the key if extensions are used (two or more push sensors to dim one lamp).
For dual-key operation in connection with the key function, you must assign the same group address to the objects of the associated keys.

› Shutter/blind function (push sensor functionality)

To implement a "complete" shutter/blind function (UP and DOWN) in connection with the key function you must assign the same group address to the STEP and MOVE objects of the associated keys.

› Value transmitter/analog value transmitter (push sensor functionality)

When you change values by long-time key actuation the newly set values will only be saved in the RAM, i. e. these values will be replaced by those presettings as originally programmed through the ETS after a voltage failure or a bus reset.

› Status indication (push sensor functionality)

The status-LEDs (in the status indicator mode) will indicate the instantaneous status of the switching object when the key function is active. If you actuate a key (e. g. ON), and the push sensor does not receive a positive acknowledgement (IACK) from an actuator addressed, the object status will be updated, with the corresponding status-LED lighting up.

• ETS plug-in system requirements

Operating system:	Windows 9x, ME, NT 4.0, 2000, XP
ETS:	ETS 2 v 1.2 a or later, ETS 3.0c recommended.
PC:	Pentium I processor (or similar), 166 MHz, 32 MB or better recommended.

• ETS functions

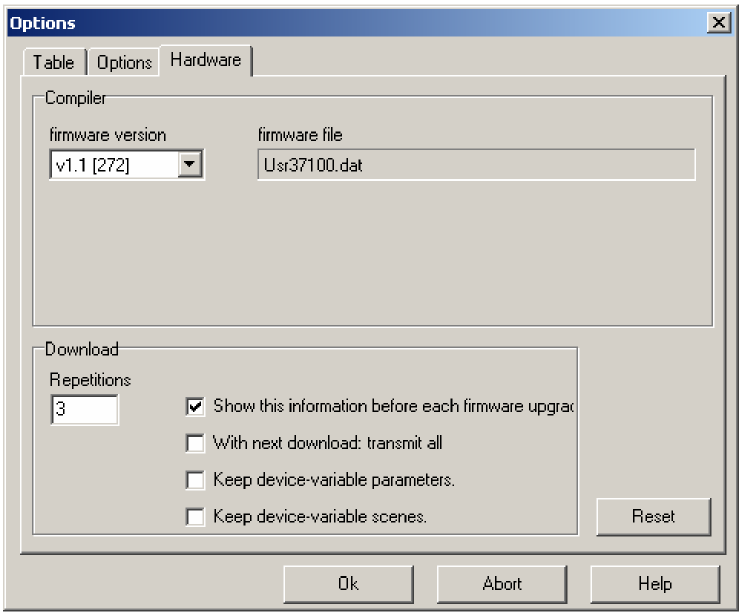
The "reading out the information into the device" or "reading out the device memory" ETS functions are not possible for the push sensor 2 plus.

Even executing the "reduce database" ETS function will lead to project data corruption for the push sensor 2 plus up to including version v 1.3 and should be avoided under any circumstances.

- **Firmware**

The push sensor 2 plus offers the possibility to update the application software in the device. The ETS plug-in facilitates this firmware download by loading the data into the application module via the bus.
In this way, even earlier push sensor 2 pluss can always be brought to the latest standard without having to replace the device. Only the push sensor 2 plus software in the ETS must be up-to-date. Normally, a firmware download will only be necessary if you want to update an earlier device.
For 'normal' application data programming procedures, it will not be necessary to transfer the firmware.
Even for the first start-up, the firmware is pre-programmed at the factory and already contained in the push sensor.
During the Downloads the message appears in the announcement
A firmware download takes a few minutes and is indicated by the message " LoAd " at the display and a rhythmic blinking of both status-LEDs of the rocker 2.

If a firmware download should become necessary you must call the "options" menu item from the "settings" menu in the ETS plug-in. This will open the options dialog. On the "hardware" tab, you can preset the following parameters:



firmware version: v1.1 [272]

firmware file: Usr37100.dat

Repetitions: 3

☑ Show this information before each firmware upgrac

□ With next download: transmit all

□ Keep device-variable parameters.

□ Keep device-variable scenes.

- Compiler firmware version: This selection box lists the firmware versions known to the software. Here, you should always select the most current version (highest number). New firmware versions will, in the future, be provided by a separate software update.
- The firmware download will be started together with the application download. To ensure the loading of the firmware into the device during the next programming process check the "with next download: transmit all" box.
- Prior to a programming process, the software will detect automatically whether the firmware existing in the device corresponds to the version specified by the software. If this is not the case the software will offer a firmware upgrade or downgrade in the form of a dialog. If you uncheck the "show this information before each firmware upgrade" box this message will no longer appear, even though you are programming further push sensor 2 plus versions containing 'inappropriate firmware.
  Later, you can reactivate this check box in the ETS plug-in from the options dialog on the "hardware" tab.
- If you have checked the "keep device-variable parameters" box the temperature setpoints (decrease/increase/standby, economy, dead band, basic setpoint) of the first control loop locally changeable on the device or via the bus will not be replaced by the values parameterized in the ETS plug-in during a download. If the "Keep device-variable scene parameters" checkbox is selected, the scene values, which are stored locally on the device, will not be replaced – during a download- by the values parameterized in the ETS plug-in. If the push button is programmed for the first time after including it into the ETS project, all parameters and scene values are definitely loaded into the target device even if the flags are set.