B.IQ push button 3 - 5gang with room thermostat and display V2, flush-mounted 7566359x, 7566459x, 7566559x

Technical Documentation

Product name:	B.IQ push button 3-, 4-, 5gang with room thermostat and display V2
Design:	Flush-mounted
ETS search path:	Push button / B.IQ / B.IQ push button xgang with room thermostat and display V2 Push button / Push button xgang / B.IQ push button xgang with room thermostat and display V2 x = 3, 4, 5

Functional description:

Push button functionality:
Depending on the parameterized software, the B.IQ push button with room thermostat (RTR) and display (Room Temperature Regulator) will send telegrams to the instabus EIB if one of its buttons is actuated. These can, for example, be telegrams for switching or touch control, dimming (also single-button dimming), or for shutter control using various operating concepts. You can also program value transmitter functions such as value transmitter 1 bytes, light scene extensions for recalling externally or internally stored light scenes or value transmitter 2 bytes (for example, temperature or brightness value transmitters). Another function allows the transmission of different switching or value commands with only one key-press.
In this connection, you can freely assign the buttons/rockers to the various functions. Distinction can be made between button and rocker functions.
In addition, the B.IQ push button with room thermostat (RTR) and display offers the possibility to disable specific individual rockers or the entire push button. You can also operate the room temperature regulator integrated in the device by actuating the push button.

Room temperature regulator functionality:
You can use the B.IQ push button with room thermostat (RTR) and display for single-room temperature control. In this connection, the regulator can distinguish between and trigger up to two control circuits which optionally have their own temperature set values. Triggered by control circuit 1, the operating statuses and the operating modes of the two control circuits 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 operating mode, the current temperature set value 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 circuits. In a control circuit, the room temperature can be sensed either by the internal (in the push button enclosure) or by an optionally external temperature sensor. If the second control circuit is activated the room temperature of the first circuit will be sensed by the internal sensor, whereas the room temperature of the second circuit will be determined by the external sensor.
If you use only one control circuit 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 set value difference between the basic and the additional stage by a parameter. For major deviations between the temperature set value and the actual temperature, you can activate this additional stage to heat up or cool down the room faster. You can assign different control algorithms to the basic and additional stages.

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

Optionally, the stages of a fan coil actuator can be shown on the display of the B.IQ push button with room thermostat and display.

Functional description (continued):

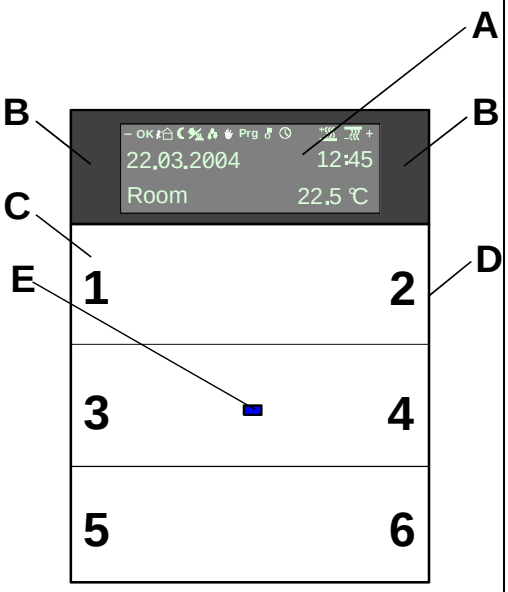
Controller extension functions:
Alternatively to its function as a room temperature regulator, the push button RTR can act as a controller extension. In this manner it can adequately control and operate via its keys or rockers one or several push buttons RTR which operate as controllers via KNX / EIB communication objects. The display of the controller extension can also be controlled via objects so that, among other things, the current temperature values and the active operating modes will also be displayed on all extensions.

Scene function:
The push button RTR can be programmed for up to 8 different scenes. Each scene can control up to 8 different scene objects via switching, value or shutter commands. The scene recall or the storing of new scene values takes place via an extension object or via touch control.

General functions:
Two independent 1-bit or 1-byte timer functions with up to 14 different switching time events permit the time-dependent transmission of commands to the bus.
If desired, you can activate a push button assistance function. In this connection, when you actuate a key its function can be briefly read on the display.
The display of a 14-byte text message (e. g. alarm message) received via the bus, or the display of up to three pre-parameterized text messages (triggered through 1-bit communication objects), is also possible.
The display background light can be optionally switched through a separate object.
If the B.IQ push button RTC is pulled off the bus coupling unit an alarm message (of 1-bit, 1-byte or of 2-byte type) can be sent.

Layout:

3gang type



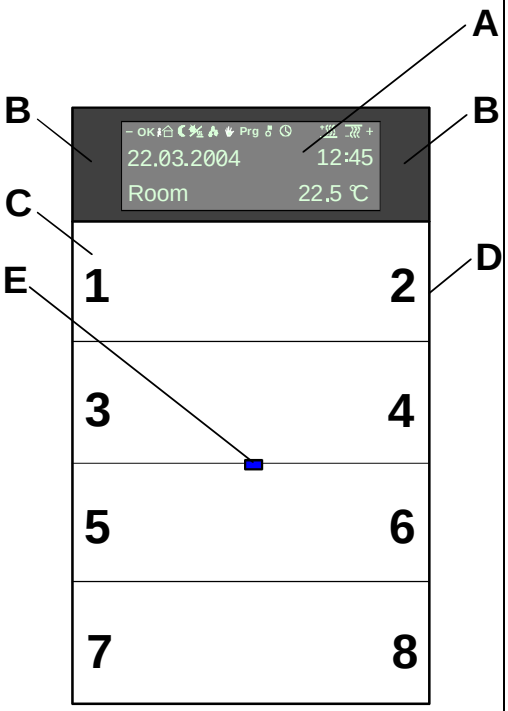
Dimensions:

Width: 88.5 mm
Height: 118 mm
Depth: 15 mm (without PEI)

Controls:

A: Display
B: Display buttons (on the right and left of the display)
C: Rockers 1 – 3 (push button function)
D: 3 x 2 status-LEDs (white) to indicate the statuses of the buttons or rockers
E: 1 device operation LED (blue)

4gang type



Dimensions:

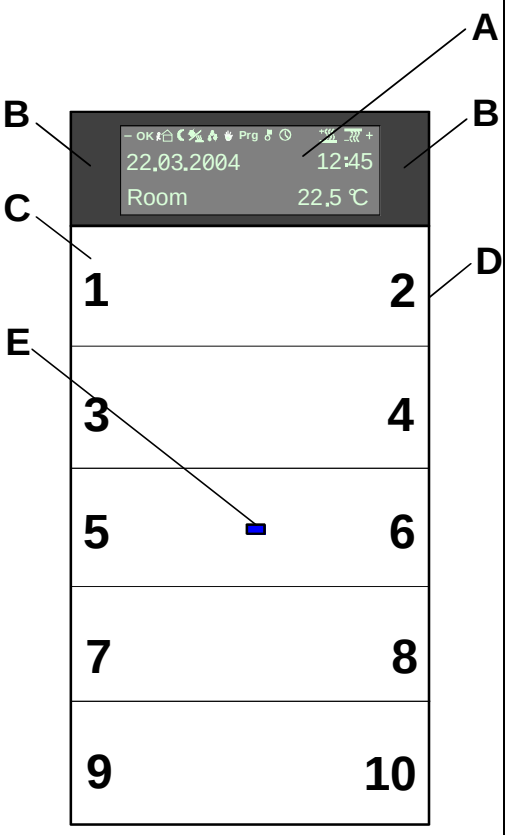
Width: 88.5 mm
Height: 147.5 mm
Depth: 15 mm (without PEI)

Controls:

A: Display
B: Display buttons (on the right and left of the display)
C: Rockers 1 – 4 (push button function)
D: 4 x 2 status-LEDs (white) to indicate the statuses of the buttons or rockers
E: 1 device operation LED (blue)

B.IQ push button 3 - 5gang with room thermostat and display V2, flush-mounted 7566359x, 7566459x, 7566559x

5gang type



Width: 88.5 mm
Height: 177 mm
Depth: 15 mm (without PEI)

A: Display
B: Display buttons (on the right and left of the display)
C: Rockers 1 – 5 (push button function)
D: 5 x 2 status-LEDs (white) to indicate the statuses of the buttons or rockers
E: 1 device operation LED (blue)

Item numbers:

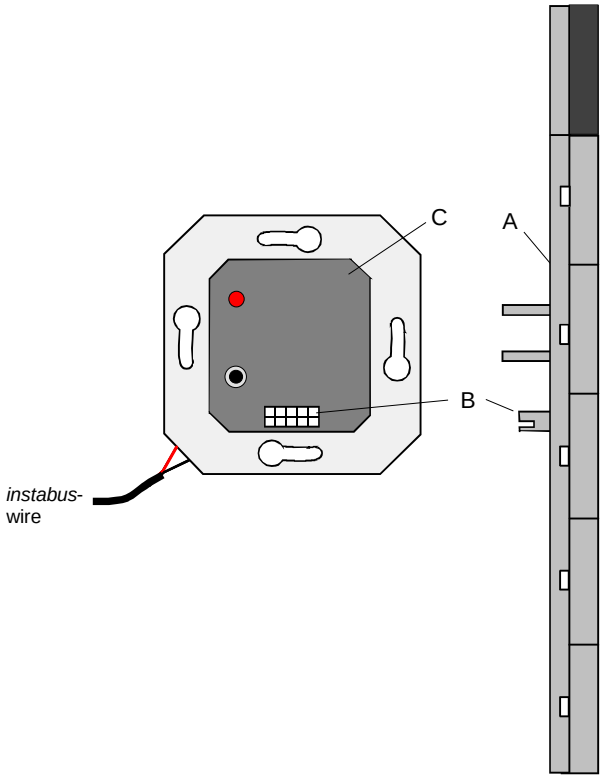
Push button	Berker no.	
B.IQ push button 3gang with RTR and display V2	7566 35 9x	
B.IQ push button 4gang with RTR and display V2	7566 45 9x	
B.IQ push button 5gang with RTR and display V2	7566 55 9x	

Technical data	
Type of protection:	IP 20
Safety class:	III
Mark of approval:	KNX / EIB
Ambient temperature:	-5 ℃ ... +45 ℃
Storage/ transport temperature:	-25 ℃ … +70 ℃ (storage over +45 ℃ will reduce pr oduct life)
Mounting position:	Any (preferably vertical/ display up)
Minimum distances:	None
Type of fastening:	Plugging onto flush-mounted BCU
Supply instabus EIB	
Voltage:	21 … 32 V DC (via flush-mounted BCU)
Power consumption:	Typically 150 mW (via flush-mounted BCU)
Wiring:	2 x 5 pole male connector strip (PEI)
Versorgung extern	---
Room temperature regulator (internal temperature sensor):	
Measuring range:	0 ℃ … + 40 ℃ ±1 %
Resolution:	0.1 K
Atmospheric humidity:	0 % to 95 % (no moisture condensation)
Internal clock:	
Resolution:	1 minute
Deviation:	Max. of 8 minutes a day In order to minimize the deviation, the internal clock should be set and thus be updated hourly via the bus.
Response to voltage failure	
Bus voltage only:	All object values will be deleted. Push button function: no response, LED's switch off Room temperature regulator: no response, control off
Mains voltage only:	---
Bus and mains voltage:	---
Response to return of voltage	
Bus voltage only:	Push button function: no response Room temperature regulator: The controller initializes. According to the parameterization, different temperature values and the status will be transmitted and the switch-over objects will be updated.
Mains voltage only:	---
Bus and mains voltage:	---
Input:	---
Output:	---

Wiring diagram: **Terminal assignment:**

The B.IQ push button with room thermostat (RTR) and display must be plugged onto a flush-mounted bus coupling unit (BCU 1).

For example, 5gang type



A: B.IQ RTR push button, 3gang/4gang/5gang type
B: User interface (PEI)
C: Bus coupling unit

model bus coupling unit <u>plus</u>

Berker ordering no.: 7504 00 03

Remarks on the hardware

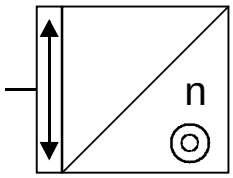
- The B.IQ push button with room thermostat (RTR) and display may only be plugged onto the model bus coupling unit <u>plus</u> (refer to the bus coupling unit illustration above and to its ordering number). The B.IQ push button with room thermostat (RTR) and display will not work when it is plugged onto different flush-mounted bus coupling units.

Software Description

ETS search path:

Push button / B.IQ / B.IQ push button 3gang with room thermostat and display V2
Push button / Push button 3gang / B.IQ push button 3gang with room thermostat and display V2

ETS symbol:



PEI type	00 Hex	0 Dez	No adapter used.

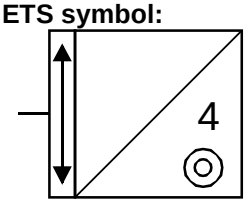
Application:

No.	Brief description:	Name:	Version:
1	B.IQ push button with room temperature regulator	B.IQ multifunktion RTR + display V2 161302	0.2

ETS search path:

Push button / B.IQ / B.IQ push button 4gang with room thermostat and display V2
Push button / Push button 4gang / B.IQ push button 4gang with room thermostat and display V2

ETS symbol:



PEI type	00 Hex	0 Dez	No adapter used.

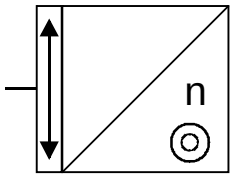
Application:

No.	Brief description:	Name:	Version:
1	B.IQ push button with room temperature regulator	B.IQ multifunktion RTR + display V2 161402	0.2

ETS search path:

Push button / B.IQ / B.IQ push button 5gang with room thermostat and display V2
Push button / Push button 5gang / B.IQ push button 5gang with room thermostat and display V2

ETS symbol:



PEI type	00 Hex	0 Dez	No adapter used.

Application:

No.	Brief description:	Name:	Version:
1	B.IQ push button with room temperature regulator	B.IQ multifunktion RTR + display V2 161502	0.2

Application:	1.	B.IQ multifunktion RTR + display V2 161302 B.IQ multifunktion RTR + display V2 161402 B.IQ multifunktion RTR + display V2 161502		
Runable from screen form version:	1.2			
Number of addresses (max.):	75	**Dynamic table management**	Yes ☐	No ☒
Number of assignments (max.):	200	**Maximum table length**	75-BCU + 200-application µC	
Communication objects:	91			

Push button functions:

The following objects apply solely for the "concept of operation = 2 push buttons (2 objects)":

Function: no function (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Function: Switching / pushing (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
0-9	Switching	Key 1 – key 10 [1]	1.001	1-bit	C, W, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Function: Dimming (for all 4 keys (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
0-9	Switching	Key 1 – key 10 [1]	1.001	1-bit	C, W, T
10-19	Dimming	Key 1 – key 10 [1]	3.007	4-bit	C, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Function: Shutter (for all 4 keys (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
0-9	STEP operation	Key 1 – key 10 [1]	1.007	1-bit	C, T
10-19	MOVE operation	Key 1 – key 10 [1]	1.008	1-bit	C, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Function: Value transmitter 1-byte (for all 4 keys (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
0-9	Value transmitter	Key 1 – key 10 [1]	5.001, 5.003, 5.004, 5.010	1-byte	C, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Function: Value transmitter 2 byte (for all 4 keys (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
0-9	Value transmitter [temp. / brightness]	Key 1 – key 10 [1]	7.001, 7.010, 8.001, 9.0xx	2-byte	C, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

[1]: The "no function", "switching/pushing", "dimming", "shutter", "value transmitter 1-byte", "value transmitter 2 byte", "light scene extension / light scene recall", "two telegrams", "operating mode switch-over", "timer operation" and "room temperature timer operation" functions can be selected per key. Accordingly, the names of the communication objects and the object table (dynamic object structure) change. It is also possible to combine key or rocker functions.

Function: Light-scene extension / light scene recall (for all 4 keys (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
0-9	Scene extension [2]	Key 1 – key 10 [1]	18.001	1-byte	C, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Function: Two telegrams (for all 4 keys (for all 6 / 8 / 10 keys [1])
Type = switching for both objects.

Object	Function	Name	DPT-ID	Type	Flag
0-9	Switching 2-stage A	Key 1 – key 10 [1]	1.001	1-bit	C, W, T
10-19	Switching 2-stage B	Key 1 – key 10 [1]	1.001	1-bit	C, W, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Type = Value for both objects.

Object	Function	Name	DPT-ID	Type	Flag
0-9	Value 2-stage A	Key 1 – key 10 [1]	5.001, 5.003, 5.004, 5.010	1-byte	C, T
10-19	Value 2-stage B	Key 1 – key 10 [1]	5.001, 5.003, 5.004, 5.010	1-byte	C, T
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

Function Operating mode switch-over / setpoint shifting / room temperature timer operation (for all 6 / 8 / 10 keys [1])

Object	Function	Name	DPT-ID	Type	Flag
81-90	Status	Key 1 – key 10 [1]	1.001	1-bit	C, W

[1]: The "no function", "switching/pushing", "dimming", "shutter", "value transmitter 1-byte", "value transmitter 2 byte", "light scene extension / -recall", "two telegrams", "operating mode switch-over", "setpoint shifting", "timer operation" and "room temperature timer operation" functions can be selected per key. Accordingly, the names of the communication objects and the object table (dynamic object structure) change. It is also possible to combine key or rocker functions.

[2]: The scene extension object is not visible when set to "function as = recall internal scene request".

The following objects apply solely for the "concept of operation = rocker (1 object)":

Function: no function (for all 2 / 3 / 5 rockers [3])

Object	Function	Name	DPT-ID	Type	Flag
81...90	Status left / right	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W

Function: Switching (for all 2 / 3 / 5 rockers [3])

Object	Function	Name	DPT-ID	Type	Flag
0/2/4/6/8	Switching	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W, T
81...90	Status left / right	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W

Function: Dimming (for all 2 / 3 / 5 rockers [3])

Object	Function	Name	DPT-ID	Type	Flag
0/2/4/6/8	Switching	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W, T
10/12/14/ 16/18	Dimming	Rocker 1 – rocker 5 [3]	3.007	4-bit	C, T
81...90	Status left / right	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W

Function: Shutter (for all 2 / 3 / 5 rockers [3])

Object	Function	Name	DPT-ID	Type	Flag
0/2/4/6/8	STEP operation	Rocker 1 – rocker 5 [3]	1.007	1-bit	C, T
10/12/14/ 16/18	MOVE operation	Rocker 1 – rocker 5 [3]	1.008	1-bit	C, T
81...90	Status left / right	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W

[3]: The "no function", "switching", "dimming", "shutter", "two telegrams" and "operating mode switch-over" functions can be selected per rocker. Accordingly, the names of the communication objects and the object table (dynamic object structure) change. It is also possible to combine key or rocker functions.

Function: Two telegrams (for all 2 / 3 / 5 rockers [3])
Type = switching for both objects.

Object	Function	Name	DPT-ID	Type	Flag
0-9	Switching 2-stage A	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W, T
10-19	Switching 2-stage B	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W, T
81...90	Status left / right	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W

Type = Value for both objects.

Object	Function	Name	DPT-ID	Type	Flag
0-9	Value 2-stage A	Rocker 1 – rocker 5 [3]	5.001, 5.003, 5.004, 5.010	1-byte	C, T
10-19	Value 2-stage B	Rocker 1 – rocker 5 [3]	5.001, 5.003, 5.004, 5.010	1-byte	C, T
81...90	Status left / right	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W

Function: Operating mode switch-over (for all 2 / 3 / 5 rockers [3])

Object	Function	Name	DPT-ID	Type	Flag
81...90	Status left	Rocker 1 – rocker 5 [3]	1.001	1-bit	C, W

[3]: The "no function", "switching", "dimming", "shutter", "two telegrams" and "operating mode switch-over" functions can be selected per rocker. Accordingly, the names of the communication objects and the object table (dynamic object structure) change. It is also possible to combine key or rocker functions.

With "concept of operation= without function" for all rockers there are no objects for keys or rockers available!

The following objects are provided for the disable function (push button functionality), the device operation LED or for the alarm and display functions:

Function: Alarm message (data format: switching telegram 1-bit)

Object	Function	Name	DPT-ID	Type	Flag
20	Switching	Alarm message	1.001	1-bit	C, T[4]

Function: Alarm message (data format: value telegram: 1-byte)

Object	Function	Name	DPT-ID	Type	Flag
20	Value transmitter	Alarm message	5.001, 5.003, 5.004, 5.010	1-byte	C, T

Funktion: Alarm message (data format: value telegram: 2-byte)

Object	Function	Name	DPT-ID	Type	Flag
20	Value transmitter	Alarm message	7.001	2-byte	C, T [4]

Function: Disable function

Object	Function	Name	DPT-ID	Type	Flag
21	Disable function	Disable push button	1.001	1-bit	C, W

Function: Switching the display backgroung lighting and the device operation LED

Object	Function	Name	DPT-ID	Type	Flag
22	Switching	Switching display	1.001	1-bit	C, W

Function: Display reading

Object	Function	Name	DPT-ID	Type	Flag
25	Value display	Display value 1 bit	1.001	1-bit	C, W
25	Value display	Display value 1 byte	5.0xx, 6.001, 6.010	1-byte	C, W
25	Value display	Display value 2 byte	7.00x, 8.001, 9.0xx	2-byte	C, W
25	Scene display	Reset display	1.001	1-bit	C, W
25	Info text	Read text	16.000	14-byte	C, W
25	Outside sensor	Temperature value	9.001	2-byte	C, W
57	Time	Time signal	10.001	3-byte	C, W
58	Date	Date	11.001	3-byte	C, W

Function: Text message

Object	Function	Name	DPT-ID	Type	Flag
73	Read alarm message	Text message	16.000	14-byte	C, W
74	Reset alarm message	Text message	1.001	1-bit	C, W

Function: Text display

Object	Function	Name	DPT-ID	Type	Flag
75	Read/reset text	Text 1 display	1.001	1-bit	C, W
76	Read/reset text	Text 2 display	1.001	1-bit	C, W
77	Read/reset text	Text 3 display	1.001	1-bit	C, W
78	Reset display	Text 1-3 display	1.001	1-bit	C, W

[4]: The "alarm message" object can only be assigned one group address!

R-flag: The objects can only be read out, if the push button is plugged on!

Room temperature regulator functions:

Function: Actual-temperature

Object	Function	Name	DPT-ID	Type	Flag
23	Actual-temperature	Measures adapted value	9.001	2-byte	C, R, T

Function: additional temperature sensor

Object	Function	Name	DPT-ID	Type	Flag
24	External temperature sensor	Temperature value	9.001	2-byte	C, W, T
25	Outside temperature sensor	Temperature value	9.001	2-byte	C, W

Function: Presetting basic setpoint

Object	Function	Name	DPT-ID	Type	Flag
26	Basic setpoint	Preset temperature	9.001	2-byte	C, W
27	Basic setpoint 2nd control circuit [5]	Preset temperature	9.001	2-byte	C, W

Function: Operating mode switch-over

Operating mode switch "via value (1-byte)":

Object	Function	Name	DPT-ID	Type	Flag
28	Operating mode switch-over	KONNEX operating mode switch-over	---	1-byte	C, W(, T)[6]
32	Operating mode forced-control	KONNEX operating mode switch-over	---	1-byte	C, W

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

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

Presence object and window status:

Object	Function	Name	DPT-ID	Type	Flag
33	Presence object	Presence key / presence detector	1.001	1-bit	C, W, T
34	Windows status	Window contact	1.001, 1.019	1-bit	C, W

Function: Control option switching

Object	Function	Name	DPT-ID	Type	Flag
35	Heating / cooling [7]	Control option switching	1.001	1-bit	C, W, (T)

[5]: This object is only active if the 2nd control circuit has been activated and if both circuits have their own setpoints.

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

[7]: This object is only active with one control circuit in the "heating and cooling" or "basic /additional - heating/ cooling" mixed-mode operation. The "T" flag is set for automatic heating / cooling switch-over.

Function: Status indication

Object	Function	Name	DPT-ID	Type	Flag
36	Controller status	Status indication general	---	1-byte	C, T
36	Controller status	Status indication individually	1.001	1-bit	C, T
37	Indication heating	Indication	1.001	1-bit	C, T
38	Indication cooling	Indication	1.001	1-bit	C, T

Function: Disable function (room temperature regulator)

Object	Function	Name	DPT-ID	Type	Flag
39	Controller operation disable	Disable function	1.001	1-bit	C, W
40	Controller disable	Disable function	1.001	1-bit	C, W
41	Disable additional stage [8]	Disable function	1.001	1-bit	C, W
41	2nd disable control circuit [8]	Disable function	1.001	1-bit	C, W

Function: Actuating variable heating

No additional stage activated / one control circuit /

For mixed-mode: Actuating variable output "heating" and "cooling" via separate objects:

Object	Function	Name	DPT-ID	Type	Flag
42	Heating (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
42	Heating (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
42	Heating (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T

No additional stage is activated / one control circuit /

For mixed-mode: Actuating variable output "heating" and "cooling" via a shared object:

Object	Function	Name	DPT-ID	Type	Flag
42	Heating/cooling (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
42	Heating/cooling (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
42	Heating/cooling (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T

Additional stage is activated / one control circuit /

For mixed-mode: Actuating variable output "heating" and "cooling" via separate objects:

Object	Function	Name	DPT-ID	Type	Flag
42	basic heating (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
42	basic heating (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
42	basic heating (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T
43	additional heating (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
43	additional heating (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
43	additional heating (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T

Two control circuits:					
Object	**Function**	**Name**	**DPT-ID**	**Type**	**Flag**
42	Heating (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
42	Heating (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
42	Heating (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T
46	Heating (control circuit 2)	Continuous actuating variable	5.001	1-byte	C, W, T
46	Heating (control circuit 2)	PWM actuating variable	1.001	1-bit	C, W, T
46	Heating (control circuit 2)	Switching actuating variable	1.001	1-bit	C, W, T

[8]: This object is only visible with activated additional stage or alternatively with two control circuits.

Additional stage is activated / one control circuit /
For mixed-mode operation: Actuating variable output "heating" and "cooling" via a shared object:

Object	Function	Name	DPT-ID	Type	Flag
42	Basic heating and basic cooling (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
42	Basic heating and basic cooling (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
42	Basic heating and basic cooling (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T
43	Additional heating and cooling (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
43	Additional heating and cooling (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
43	Additional heating and cooling (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T

Function: Actuating variable cooling
No additional stage is activated / one control circuit /
For mixed-mode: Actuating variable output "heating" and "cooling" via separate objects:

Object	Function	Name	DPT-ID	Type	Flag
44	Cooling (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
44	Cooling (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
44	Cooling (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T

Additional stage is activated / one control circuit /
For mixed-mode: Actuating variable output "heating" and "cooling" via separate objects:

Object	Function	Name	DPT-ID	Type	Flag
44	Basic cooling (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
44	Basic cooling (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
44	basic cooling (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T
45	Additional cooling (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
45	Additional cooling (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
45	Additional cooling (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T

Two control circuits:

Object	Function	Name	DPT-ID	Type	Flag
44	Cooling (control circuit 1)	Continuous actuating variable	5.001	1-byte	C, W, T
44	Cooling (control circuit 1)	PWM actuating variable	1.001	1-bit	C, W, T
44	Cooling (control circuit 1)	Switching actuating variable	1.001	1-bit	C, W, T
48	Cooling (control circuit 2)	Continuous actuating variable	5.001	1-byte	C, W, T
48	Cooling (control circuit 2)	PWM actuating variable	1.001	1-bit	C, W, T
48	Cooling (control circuit 2)	Switching actuating variable	1.001	1-bit	C, W, T

Function: Actuating variable status information heating [9]

Object	Function	Name	DPT-ID	Type	Flag
46	Heating (control circuit 1)	PWM actuating variable	5.001	1-byte	C, W, T
46	Basic heating (control circuit 1)	PWM actuating variable	5.001	1-byte	C, W, T
47	Additional heating (control circuit 1)	PWM actuating variable	5.001	1-byte	C, W, T

Function: Actuating variable status information cooling [9]

Object	Function	Name	DPT-ID	Type	Flag
48	Cooling (control circuit 1)	PWM actuating variable	5.001	1-byte	C, W, T
48	Basic cooling (control circuit 1)	PWM actuating variable	5.001	1-byte	C, W, T
49	Additional cooling (control circuit 1)	PWM actuating variable	5.001	1-byte	C, W, T

Function: Set temperature

Object	Function	Name	DPT-ID	Type	Flag
50	Set temperature	Temperature value	9.001	2-byte	C, T, R
51	Set temperature 2nd control circuit [10]	Temperature value	9.001	2-byte	C, T, R

Function: Controller extension:

Object	Function	Name	DPT-ID	Type	Flag
52	Current setpoint shifting	Feedback value	6.010	1-byte	C, T, R
53	Preset setpoint shifting	Value	6.010	1-byte	C, W

Function: Fan stage display

Object	Function	Name	DPT-ID	Type	Flag
54	Fan stage 1	Fan stage display	1.001	1-bit	C, W
55	Fan stage 2	Fan stage display	1.001	1-bit	C, W
56	Fan stage 3	Fan stage display	1.001	1-bit	C, W
54	Fan stages 0-3	Fan stages display	5.010	1-byte	C, W

[9]: The status information for the PWM actuating variable is only possible with one control circuit!

[10]: This object is only active if the 2nd control circuit has been activated and both circuits have their own setpoints.

Function: Switching for timer 1 or 2 [11]:

Object	Function	Name	DPT-ID	Type	Flag
59	Switching	Timer 1	1.001	1-bit	C, T
61	Switching	Timer 2	1.001	1-bit	C, T

Function: Value for timer 1 or 2 [11]:

Object	Function	Name	DPT-ID	Type	Flag
59	Value	Timer 1	5.001, 5.003, 5.004, 5.010	1-byte	C, T
61	Value	Timer 2	5.001, 5.003, 5.004, 5.010	1-byte	C, T

Function: Scene recall for timer 1 or 2 [11]:

Object	Function	Name	DPT-ID	Type	Flag
59	Scene extension	Timer 1	18.001	1-byte	C, T
61	Scene extension	Timer 2	18.001	1-byte	C, T

Disable functions for timer 1 or 2:

Object	Function	Name	DPT-ID	Type	Flag
60	Inhibit function	Disable timer 1	1.001	1-bit	C, W
62	Disable function	Disable timer 2	1.001	1-bit	C, W

Function: Room temperature timer

Object	Function	Name	DPT-ID	Type	Flag
63	Disable function	Disable room temperature timer	1.001	1-bit	C, W

[11]: The "switching", "value" and "scene recall" functions can be set separately per timer.

Controller extension functions:

Function: Actual-temperature

Object	Function	Name	DPT-ID	Type	Flag
23	Actual-temperature	Measures adapted value	9.001	2-byte	C, R, T

Function: Additional temperature sensor

Object	Function	Name	DPT-ID	Type	Flag
24	External temperature sensor	Temperature value	9.001	2-byte	C, W, T
25	External sensor	Temperature value	9.001	2-byte	C, W

Function: Operating mode switch-over

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

Object	Function	Name	DPT-ID	Type	Flag
28	Operating mode switch-over Controller extension:	KONNEX operating mode switch-over	---	1-byte	C, R, W, T
32	Operating mode forced-control Controller extension:	KONNEX operating mode switch-over	---	1-byte	C, R, W, T

Presence object:

Object	Function	Name	DPT-ID	Type	Flag
33	Presence object controller extension	Presence pushbutton	1.001	1-bit	C, W, T

Function: Status indication

Object	Function	Name	DPT-ID	Type	Flag
36	Controller status Controller extension	Status indication general	---	1-byte	C, W
37	Indication heating controller extension	Indication	1.001	1-bit	C, W
38	Indication cooling controller extension	Indication	1.001	1-bit	C, W

Function: Set temperature

Object	Function	Name	DPT-ID	Type	Flag
50	Set temperature controller extension	Temperature value	9.001	2-byte	C, W

Function: Controller extension:

Object	Function	Name	DPT-ID	Type	Flag
52	Current setpoint shifting controller extension	Feedback value	6.010	1-byte	C, W
53	Presetting setpoint shifting controller extension	Value	6.010	1-byte	C, T

Scene function

Function: Switching (for all 8 scene objects [12])

Object	Function	Name	DPT-ID	Type	Flag
64	Scene output 1	Switching	1.001	1-bit	C, W, T
65	Scene output 2	Switching	1.001	1-bit	C, W, T
66	Scene output 3	Switching	1.001	1-bit	C, W, T
67	Scene output 4	Switching	1.001	1-bit	C, W, T
68	Scene output 5	Switching	1.001	1-bit	C, W, T
69	Scene output 6	Switching	1.001	1-bit	C, W, T
70	Scene output 7	Switching	1.001	1-bit	C, W, T
71	Scene output 8	Switching	1.001	1-bit	C, W, T

Function: Value (for all 8 scene objects [12])

Object	Function	Name	DPT-ID	Type	Flag
64	Scene output 1	Value	5.001, 5.003, 5.004, 5.010	1-byte	C, W, T
65	Scene output 2	Value		1-byte	C, W, T
66	Scene output 3	Value		1-byte	C, W, T
67	Scene output 4	Value		1-byte	C, W, T
68	Scene output 5	Value		1-byte	C, W, T
69	Scene output 6	Value		1-byte	C, W, T
70	Scene output 7	Value		1-byte	C, W, T
71	Scene output 8	Value		1-byte	C, W, T

Function: Shutter position (for all 8 scene objects [12])

Object	Function	Name	DPT-ID	Type	Flag
64	Scene output 1	Shutter position	1.008	1-bit	C, W, T
65	Scene output 2	Shutter position	1.008	1-bit	C, W, T
66	Scene output 3	Shutter position	1.008	1-bit	C, W, T
67	Scene output 4	Shutter position	1.008	1-bit	C, W, T
68	Scene output 5	Shutter position	1.008	1-bit	C, W, T
69	Scene output 6	Shutter position	1.008	1-bit	C, W, T
70	Scene output 7	Shutter position	1.008	1-bit	C, W, T
71	Scene output 8	Shutter position	1.008	1-bit	C, W, T

Function: Scene extension:

Object	Function	Name	DPT-ID	Type	Flag
72	Scene extension	Extension input	18.001	1-byte	C, W

Temperature control

Function: Temperature alarm:

Object	Function		DPT-ID	Type	Flag
79	Switching	Temperature alarm 1	1.001	1-bit	C, T
80	Switching	Temperature alarm 2	1.001	1-bit	C, T

[12]: The "switching", "value" and "shutter position" functions can be set per scene object.

Object description

Objects:

☐→|	0 - 9	Switching:	1-bit object for transmission of switching telegrams (ON, OFF).
☐ |	10 - 19	Dimming:	4-bit object for change of relative brightness between 0 and 100 %.
☐ |	0 - 9	STEP operation:	1-bit object for short-time operation (Step) of a shutter.
☐ |	10 - 19	MOVE operation:	1-bit object for long-time operation (Move) of a shutter.
☐ |	0 - 9	Scene extension:	1-byte object for recalling or for storing (light-) scenes 1 – 64).
☐ |	0 - 9	Value transmitter:	1-byte object for transmission of, for example, dimming value telegrams (0 – 255).
☐ |	0 - 9	Value transmitter [temp. / brightness]:	1-byte object for transmitting temperature values (0 - 40 °C) of brightness values (0 – 1500 lux) or 2-byte values (0 – 65535).
☐ |	0 - 9	Switching 2-stage A:	1-bit object for transmission of switching telegrams (ON, OFF).
☐ |	10 - 19	Switching 2-stage B:	1-bit object for transmission of switching telegrams (ON, OFF).
☐ |	0 - 9	Value 2-stage A	1-byte object for transmission of, for example, dimming value telegrams (0 – 255).
☐ |	10 - 19	Value 2-stage B:	1-byte object for transmission of, for example, dimming value telegrams (0 – 255).
☐ |	20	Switching:	1-bit object for transmission of an alarm message.
☐ |	20	Value transmitter:	1-byte object for transmission of an alarm message.
☐ |	20	Value transmitter:	2-byte object for transmission of an alarm message.
☐→|	21	Disable function:	1-bit object for disabling the push button's keys or rockers.
☐→|	22	Switching:	1-bit object for switching the display background light.
☐|→	23	Actual-temperature:	2-byte object for the display of the actual-temperature (room temperature), which is determined by the controller or controller extension. (possible range of values: -99,9 °C ... +99,9 °C / Measuring range of internal temperature sensor: 0 °C ... + 40 °C ±1 %)
☐→|	24	External temperature sensor:	2-byte object for coupling an external room temperature sensor or a controller extension (via "actual-temperature" object). (possible range of values: -99,9 °C ... +99,9 °C)
☐→|	25	Outside temperature sensor:	2-byte object for coupling an outside temperature sensor to a controller or controller extension. The received value is used solely for the display. (possible range of values: -99,9 °C ... +99,9 °C)
☐ |	25	Value display:	1-bit object for receiving a display switching value.
☐ |	25	Value display:	1-byte object for receiving a 1-byte display value (display format can be parameterized).
☐ |	25	Value display:	2-byte object for receiving a 2-byte display value (display format can be parameterized).
☐ |	25	Scene display:	1-byte object for resetting a scene recall (1 = reset/0 = no response).
☐ |	25	Info text:	14-byte object for receiving a 14-character info text.

Object description

Objects continued:

↓	26	Basic setpoint:	2-byte object for external setting of basic setpoint. Depending on the operating mode, the possible range of values is limited by the parameterized frost protection and/or heat protection temperature. The received value is mathematically rounded off to half °C!
↓	27	Basic setpoint 2nd control circuit:	2-byte object for external setting of basic setpoint of the second control circuit with own setpoints. Depending on the control option, the possible range of values is limited by the parameterized frost protection and/or heat protection temperature. The received value is mathematically rounded off to half °C!
↓	28	Operating mode switch-over:	1-byte object for switch-over of the controller's operating modes acc. to KONNEX.
↓	28	Operating mode switch-over controller extension:	1-byte object for switch-over of the controller's operating modes acc. to KONNEX. The object is bi-directional so that the correct modes can also always be tracked on the extensions.
↓	28	Comfort mode:	1-bit object for switch-over into the "comfort" operating mode.
↓	29	Standby mode:	1-bit object for switch-over into the "standby" operating mode.
↓	30	Night mode:	1-bit object for switch-over into the "night" operating mode.
↓	31	Frost/ heat protection:	1-bit object for switch-over into the "frost/heat protection" operating mode.
↓	32	Forced-control object operating mode:	1-byte object for superordinated forced control of the controller's operating modes acc. to KONNEX.
↓	32	Forced-control operating mode:	1-byte object for superordinated forced control of a controller's operating modes acc. to KONNEX. The object is bi-directional so that the correct modes can also always be tracked on the extensions.
↓	33	Presence object:	1-bit object (bi-directional) which transmits the status of the presence key (if presence object is enabled, the presence key can be parameterized using the push button functions) on the bus or which allows to, for example, to couple a presence detector. (presence detected = "1", presence not detected = "0")
↓	33	Presence object controller extension:	1-bit object (bi-directional), which transmits the status of the presence key of the controller extension on the bus when activated. (presence detected = "1", presence not detected = "0")
↓	34	Windows status:	1-bit object for the coupling of window contacts. (window open = "1", window closed = "0")
↓	35	Heating / cooling:	1-bit object for the switch-over between the control options "heating" and "cooling" In case of automatic switch-over the active control option can be transmitted
↑	36	Controller status:	1-byte object for general status feedback or 1-bit object for individual status feedback of parameterizable controller functions.
↓	36	Controller status controller extension:	1-byte object for receiving the general status feedback of the main controller. This value is used to update the symbol and LED display of the extension.
↑	37	Indication heating:	1-bit object for the controller to indicate a request for heating energy (object value = "1": energy request, object value = "0": no energy request).

Object description

Objects continued:

↓	37	Indication heating controller extension:	1-bit object for feedback to the extension when the controller has requested heating energy (object value = "1": energy request, object value = "0": no energy request). This value is used to update the symbol displays of the extension.
	38	Indication cooling:	1-bit object for the controller to indicate a cooling energy request (object value = "1": energy request, object value = "0": no energy request).
↓	38	Indication cooling controller extension:	1-bit object for feedback to the extension if the controller has requested cooling energy (object value = "1": energy request, object value = "0": no energy request). This value is used to update the symbol displays of the extension.
↓	39	Disable controller operation:	1-bit object for disabling the local controller operation. (controller operation disabled = "1", controller operation enabled = "0")
↓	40	Disable controller:	1-bit object for deactivating the controller (activating dew point operation). (controller deactivated = "1", controller activated = "0")
↓	41	Disable additional stage:	1-bit object for deactivating the additional stage of the controller. (additional stage deactivated = "1", additional stage activated = "0")
↓	41	2. disable control circuit:	1-bit object for deactivating the second control circuit. (2^{nd} control circuit deactivated = "1", 2^{nd} control circuit activated = "0")
↓	42	Heating (control circuit 1):	1-byte object for the output of the continuous actuating variable used for the heating operation of the first control circuit.
↓	42	Heating (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable used for the heating operation of the first control circuit.
↓	42	Basic heating (control circuit 1):	1-byte object for the output of the continuous actuating variable for the basic heating operation of the first control circuit.
↓	42	Basic heating (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable for the basic heating operation of the first control circuit.
↓	42	Heating/cooling (control circuit 1):	1-byte object for the output of the continuous actuating variable for the heating or cooling operation of the first control circuit. (when actuating variables are output via a shared object)
↓	42	Heating/cooling (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable alternatively used for the heating or cooling operation of the first control circuit. (with actuating variables output via a shared object)
↓	42	Basic heating and cooling (control circuit 1):	1-byte object for the output of the continuous actuating variable either for the basic heating or basic cooling operation of the first control circuit. (if actuating variables are output via a shared object)
↓	42	Basic heating and cooling (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable either for the basic heating or basic cooling operation of the first control circuit. (if actuating variables are output via a shared object)
↓	43	Additional heating (control circuit 1):	1-byte object for the output of the continuous actuating variable for the additional heating operation of the first control circuit.
↓	43	Additional heating (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable used for the additional heating operation of the first control circuit.
↓	43	Additional heating and cooling (control circuit 1):	1-byte object for the output of the continuous actuating variable either for the additional heating or cooling operation of the first control circuit. (if actuating variables are output via a shared object)

Object description

Objects continued:

43	Additional heating and cooling (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable either for the heating or cooling operation of the first control circuit. (if actuating variables are output via a shared object)
44	Cooling (control circuit 1):	1-byte object for the output of the continuous actuating variable used for the cooling operation of the first control circuit.
44	Cooling (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable used for the cooling operation of the first control circuit.
44	Basic cooling (control circuit 1):	1-byte object for the output of the continuous actuating variable for the basic cooling operation of the first control circuit.
44	Basic cooling (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable for the basic cooling operation of the first control circuit.
45	Additional cooling (control circuit 1):	1-byte object for the output of the continuous actuating variable for the additional cooling operation of the first control circuit.
45	Additional cooling (control circuit 1):	1-bit object for the output of the switching actuating variable or PWM actuating variable for the additional cooling operation of the first control circuit.
46	Heating (control circuit 2):	1-byte object for the output of the continuous actuating variable for the heating operation of the second control circuit.
46	Heating (control circuit 2):	1-bit object for the output of the switching actuating variable or PWM actuating variable for the heating operation of the second control circuit.
46	Heating (control circuit 1):	1-byte object (pulse width modulated variable signal) for status feedback of the actuating variable value for heating. (only for one control circuit)
46	Basic heating (control circuit 1):	1-byte object (pulse width modulated variable signal) for status feedback of the continuous actuating variable value for basic heating. (only for one control circuit)
47	Additional heating (control circuit 1):	1-byte object (pulse width modulated variable signal) for status feedback of the actuating variable value for additional heating. (only for one control circuit)
48	Cooling (control circuit 2):	1-byte object for the output of the continuous actuating variable for the cooling operation of the second control circuit.
48	Cooling (control circuit 2):	1-bit object for the output of the switching actuating variable or PWM actuating variable for the cooling operation of the second control circuit.
48	Cooling (control circuit 1):	1-byte object (pulse width modulated variable signal) for status feedback of the continuous actuating variable value for heating. (only for one control circuit)
48	Basic cooling (control circuit 1):	1-byte object (pulse width modulated variable signal) for status feedback of the continuous actuating variable value for basic cooling. (only for one control circuit)
49	Additional cooling (control circuit 1):	1-byte object (pulse width modulated variable signal) for the status feedback of the actuating variable value for additional cooling. (only for one control circuit)
50	Set temperature:	2-byte object for the output of the current temperature-setpoint of the first control circuit. Depending on the control option, the possible range of values is limited by the parameterized frost protection and/or heat protection temperature.

Object description

Objects continued:

50	Set temperature:	2-byte object for receiving the current temperature-setpoint of a controller. The received value is, if parameterized, visible on the display of the extension.
51	Set temperature 2nd control circuit:	2-byte object for the output of the current temperature-setpoint of the first control circuit. Depending on the control option, the possible range of values is limited by the parameterized frost protection and/or heat protection temperature.
52	Current setpoint shifting:	1-byte object for giving feedback on the current setpoint shifting. x ≤ 0 ≤ y (0 = no active shifting); integral numbers The possible range of values (x to y) is fixed within the preset limits of the setpoint (parameterizable) in combination with the step value (0.5 °C)
52	Current setpoint shifting controller extension:	1-byte object for receiving the current basic setpoint shifting of a controller. x ≤ 0 ≤ y (0 = no active shifting); integral numbers The possible range of values (x to y) is fixed within the preset limits of the setpoint (parameterizable) in combination with the step value on the controller (0.5 °C)
53	Presetting setpoint shifting:	1-byte object for setting a basic setpoint shifting, e.g. via a controller extension. x ≤ 0 ≤ y (0 = no active shifting); integral numbers The possible range of values (x to y) is fixed within the preset limits of the setpoint (parameterizable) in combination with the step value (0.5 °C) In case the limits of the value range are exceeded by the preset external value, the controller inside the push button RTR will automatically reset the received value to the minimum and maximum limits.
53	Presetting setpoint shifting controller extension:	1-byte object for presetting a basic setpoint shifting, e.g. via a controller. x ≤ 0 ≤ y (0 = no active shifting); integral numbers Value object 52 + 1 (increase step value) Value object 52 + 1 (decrease step value) Thus the possible range of values (x to y) is fixed within the preset limits of the setpoint (parameterizable) in combination with the step value on the controller (0.5 °C)
54	Fan stage 1.	1-bit object for activating the fan stage display (1st stage).
54	Fan stages 0-3:	1-byte object for activating the fan stage display (no stage – 3rd stage / alternative to the 1-bit objects).
55	Fan stage 2:	1-bit object for activating the fan stage display (2nd stage).
56	Fan stage 3:	1-bit object for activating the fan stage display (3rd stage).
57	Time:	3-byte object for receiving the current time via the bus.
58	Date:	3-byte object for receiving the current time via the bus.
59	Switching:	1-bit object for transmitting the switching command of the first timer.
59	Value:	1-byte object for transmitting the value command of the first timer.
59	Scene extension:	1-byte object for transmitting the scene command of the first timer.

Object description

Objects continued:

☐→	60	Disable function:	1-bit object for disabling the first timer. (The polarity can be parameterized.)
☐ |	61	Switching:	1-bit object for transmitting the switching command of the second timer.
☐ |	61	Value:	1-byte object for transmitting the value command of the second timer.
☐ |	61	Scene extension:	1-byte object for transmitting the scene command of the second timer.
☐→	62	Disable function:	1-bit object for disabling the second timer. (The polarity can be parameterized.)
☐→	63	Disable function:	1-bit object for disabling the room temperature controller timer. (The polarity can be parameterized.)
☐ |	64 – 71	Scene output 1 – 8:	1-bit object for transmission of up to eight switching commands of one scene.
☐ |	64 – 71	Scene output 1 – 8:	1-byte object for transmission of up to eight value commands of one scene.
☐ |	64 – 71	Scene output 1 – 8:	1-bit object for transmission of up to eight shutter long-term commands of one scene.
☐→	72	Scene extension:	1-byte object for external recall or storing of the 8 internally stored scenes.
☐→	73	Read alarm message:	14-byte object for receiving a 14-character display text (e. g. alarm message).
☐→	74	Reset alarm message:	1-bit object for resetting a display alarm message. (The polarity can be parameterized.)
☐→	75	Read/reset text:	1-bit object for activating the reading of the given 1st display text. Optionally, you can also reset the display through this object (The polarity can be parameterized.)
☐→	76	Read/reset text:	1-bit object for activating the reading of the given 2nd display text. Optionally, you can also reset the display through this object. (The polarity can be parameterized.)
☐→	77	Read/reset text:	1-bit object for activating the reading of the given 3rd display text. Optionally, you can also reset the display through this object. (The polarity can be parameterized.)
☐→	78	Reset display:	1-bit object for resetting all three display texts. (The polarity can be parameterized.)
☐ |	79	Switching:	1-bit object for transmission of a switching telegram from the temperature control system (temperature alarm 1 / lower temperature value).
☐ |	80	Switching:	1-bit object for transmission of a switching telegram from the temperature control system (temperature alarm 2 / upper temperature value).
☐→	81 - 90	Status	1-bit object for separate status-LED control of a key or rocker.

Scope of functions

Push button functions:

- **General**
  - Free allocation of the switching/pushing, dimming, shutter, light-scene extension / light-scene recall, 1-byte value transmitter, 2-byte value transmitter, two telegrams, operating mode switch-over* and setpoint shifting functions to the keys.
  - Free allocation of the switching, dimming, shutter, two telegrams and operating mode switch-over functions to the rockers.
  - If the controller extension function is enabled, the controller extension function can be parameterized. This function allows the operating mode switch-over according to KONNEX, the control of the presence function as well as the basic setpoint adjustment of a room temperature regulator.
  - Status indication possible via 4 (2 gang), 6 (3 gang) or 10 (5 gang) red LED's. The status indication can also take place via separate status objects.
  - Even if "no function" is assigned to the rockers or keys, the status-LED's can still be controlled via objects
  - Disable object is available for disabling individual rockers (polarity of the disable object is adjustable)

  :Only if controller extension is deactivated!
- **"Switching/pushing" function**
  - Command by pushing or releasing the key is adjustable (ON, OFF, TOGGLE, no function)
  - Single-surface operation for rocker function possible (Only if "command when pressing the rocker = left = toggle, right = toggle")
  - Function of status-LED (key function) or status indication (rocker function) can be parameterized
- **"Dimming" function**
  - Time between dimming and switching and dimming increments is adjustable
  - Telegram repetition and stop telegram transmission possible
  - Single-surface operation with rocker functions possible (only if "command when pressing the rocker = left = toggle, right = toggle")
  - Function of status-LED for (key function) or status indication (rocker function) can be parameterized
- **"Shutter" function**
  - Adjustable key function (UP, DOWN, TOGGLE). Single-surface operation possible with shutter key function = TOGGLE.
  - Operation concept parameterizable (STEP – MOVE – STEP or MOVE – STEP)
  - Time adjustable between short-time and long-time operation (only for STEP – MOVE – STEP)
  - Adjustable Lamella adjustment time (time during which a MOVE command can be terminated by releasing the key).
  - Function of status-LED (key function) or status indication (rocker function) can be parameterized
- **"Value transmitter 1-byte/light-scene extension/ recall" function (only for key function!)**
  - Parameterizable value transmitter (1-byte) or light-scene recall with /without memory function
  - Value adjustment via long key-press possible
  - Parameterizable function of the status-LED
  - With light-scene recall, internal scenes can also be recalled.
- **"Value transmitter 2-byte" function (only for key function!)**
  - Parameterizable brightness value transmitter, temperature value transmitter and 2-byte value transmitter key functions
  - Value adjustment via long key press possible
  - Parameterizable status-LED function
- **"Two telegrams" function**
  - Transmission of two different switching or value telegrams via one key-press
  - Switching command (ON, OFF, TOGGLE) or value (0...255) parameterizable
  - Adjustable time delay between the telegrams
  - Status-LED function parameterizable

Room temperature regulator functions:

- **General**
  - 5 operating modes: comfort, standby, night, frost/heat protection and controller disable
  - Operating modes switch-over via 1-byte object according to KONNEX or individual 1-bit objects.
  - Display of room temperature regulator information via an integrated display
  - Local programming mode possible. This one can be activated or deactivated.
    - "No operation": 'normal operation' and local operation of room temperature regulator by pressing the display keys for shifting the basic setpoint.
    - "Limited operation": Switch-over into programming mode possible → 'normal operation' incl. setpoint shifting and operating mode switch-over and adjustments of different setpoints for heating and/or cooling.
    - "Complete operation": Full access to the device for local operation. In addition to limited operation, it grants the user access to the up to three timers (activation/deactivation of the timers and changing of the individual switching programs) and to the "settings" menu (activation/deactivation of the push button assistance function, the LCD background light and key disabling as well as of the display of the version data).
- **Heating/cooling system**
  - Control options: "heating", "cooling", "heating and cooling" each with or without additional stage.
  - Up to two control circuits with optional different temperature-setpoints and shared operating mode switch-over possible.
    (with two control circuits only "heating" or "cooling" can be activated, no additional stage!)
  - PI control (continuous or switching PWM) or 2-point control (switching) adjustable as control algorithms.
  - Continuous (1-byte) or switching (1-bit) actuating variable output.
  - Control parameter for PI controller (if desired: proportional range, reset-time) and 2-point controller (hystereses) adjustable.
- **Setpoints**
  - Each operating mode can be assigned its own temperature-setpoints (for heating and/or cooling).
  - The setpoints for the additional stage are derived via a parameterizable stage offset from the values of the basic stage.
  - Temporary setpoint shifting or permanent setpoint shifting via local control on the device or via communication objects possible (parameterizable scaling of setpoint shifting).
- **Functions**
  - Automatic or object oriented switch-over between "heating" and "cooling".
  - Optionally, the controller operation can be disabled via an object.
  - Parameterizable duration of the comfort mode extension.
  - Complete (1-byte) or partial (1-bit) status information parameterizable and transmissible on the bus via an object.
  - Deactivating the control, the additional stage or the second control circuit via different objects possible.
- **Room temperature measurement**
  - Internal and external room temperature sensor available.
  - Parameterizable internal to external determination of measured value with one control circuit and enabled external sensor.
  - If using two control circuits the actual-temperature value of the second circuit will be determined by the external sensor.
  - Measurement period of external temperature sensor is adjustable.
  - The actual and set temperature can be output on the bus if a parameterizable deviation is detected (also periodically).
  - The room temperature measurement (actual value) can be adjusted separately for the internal and external sensor via parameter.
  - Frost/heat protection switch-over via window status (delayed analysis possible) and automatic frost protection.
  - Temperature alarm within preset limits possible. Telegram activation via two separate objects.

- **Actuating variable output**
  - Separate or combined actuating variable output via one or two objects in "heating and cooling" mode
  - Normal or inverted actuating variable output parameterizable
  - Automatic transmission and cycle-time for actuating output parameterizable
- **Room temperature timer**
  - Time and day-of-week depending operating mode control with up to 28 different switching times.
  - Can be activated or deactivated via local control in programming mode.
  - In addition, the room temperature timer can be disabled via bus.

Timer functionality:

- Up to two separate timers, each of them having up to 14 different switching times.
- Switching (ON/OFF), value (0…255) or scene recall (1…8) telegrams can be transmitted to the bus as control commands.
- Both timers can be disabled individually via the bus or by local operation.

Controller extension functions:

- Alternatively to the function as room temperature regulator, the extension mode can be activated.
  → Activation of another push button RTR parameterized to function as a room temperature regulator.
- Full control (operating modes, presence functions and setpoint shifting)
- Full-featured display of the controller status on the display of the extension (heating /cooling indication, setpoint shifting, set or room temperature and current operating mode)
- Room temperature measurement on the extension possible.
- Local programming mode possible. This one can be activated or deactivated.
  - "No operation": Display keys without function
  - "Complete operation": Full access to the device for local operation. It grants the user access to the two timers (activation/deactivation of the timers and changing of the individual switching programs) and to the "settings" menu (activation/deactivation of the push button assistance function, the LCD background light and key disabling as well as of the display of the version data).

Scene functions:

- 8 independent internal scenes
- Each scene controls up to 8 objects, i.e. eight different commands are transmissible.
- Selectable data types include switching (On / Off), dimming value (0...255 / 0...100%) or long-time shutter commands (Up / Down) that are parameterizable per scene and scene object.
- The scenes can be recalled or stored via an extension object.
- Recall of internal scenes also possible without extension object via push button control.

Push button general:

The automatic deactivation of the display background light can be parameterized. Alternatively, the display background light can be switched through a separate object or is always ON. The blue device ON LED will always be triggered together with the display background light.

- The display can read an alarm message (text message with a maximum length of 14 characters) received via the bus. You can press any key or send an acknowledgement telegram to acknowledge such alarm message.
- Display of up to three predefined texts (with a maximum length of 20 characters). Recalling or resetting through separate 1-bit communication objects.
- Various display reading options in the push button normal mode: Time, date, room temperature, outside temperature, setpoint temperature, value display, info text display, scene number display.
- An alarm to be raised after the device has been unplugged from the flush-mounted bus coupling unit can be parameterized (1-bit, 1-byte, 2-byte).
- The push button assistance function can be parameterized. In this connection, when you actuate a push button, its function can be briefly read on the display as a help text.

Table of Contents I

Table of Contents II

Functional Description

1. General functions

1.1 Enabling the push button, scene, timer and room temperature regulator functions

In the B.IQ RTR push button, the push button, scene, timer and room temperature regulator functions must be considered as separate items. These individual components can be optionally activated, if necessary. For this purpose, set the "push button function", "scene function", "timer 1/2 function" or "room temperature regulator function" parameters in the "B.IQ RTR push button" parameter branch to "enabled". 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 of the push button 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 or value transmitter functions if you have deactivated the push button function. On the other hand, if the room temperature regulator function has been deactivated, you cannot set any room temperature control for the rockers or push buttons.

1.2 Alarm function

When you unplug the B.IQ push button with room thermostat (RTR) and display from the flush-mounted 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 application module = "disabled"" (default) ETS parameter setting.

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

When you unplug the push button from the bus coupling unit, an alarm telegram with an alarm value after removal corresponding to the parameterization (switching value "ON" or "OFF" for a switching telegram, or value "0…255" or value (0…655535) for a value telegram) will be sent.

After having been re-plugged, the push button will be ready for operation again after its initialization phase (display reading: "**Initializing**"). In this connection, the value of the alarm object will always be reset to "0" (for the 1 bit switching value and for the 2-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 button.

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

b) "Reset value" = "Yes" parameter

When you unplug the push button from the bus coupling unit, an alarm telegram with an alarm value after removal corresponding to the parameterization (switching value "ON" or "OFF" for a switching telegram, or value "1…255" or value (0…655535) for a value telegram) will be sent.

After having been re-plugged, the push button will be ready for operation again after its initialization phase (display reading: "**Initializing**"). 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 1byte or 2-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 button.

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 button has not been plugged on. If the push button has been plugged on when 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 1byte or 2-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 button plugged on (set the "R" flag).

1.3 Light duration of status LED at operation indication

For all push button functions associated with button operations, you can parameterize the status-LED of a button as operation indication. Only in this case, the LEDs will be lit for the period set by the "Light duration of status LED at operation indication" parameter in the "B.IQ RTR push button" parameter branch if the buttons are actuated. You will have the option to parameterize 1 s, 2 s or 3 s (default).

1.4 Programming mode/local operation

The B.IQ push button with room thermostat (RTR) and display has a programming mode directly on the device. By means of this option, you can locally configure various functions or even preset various temperature values or switching times in addition to the parameterization through the ETS plug-in. Switching over into the programming mode or navigating through the menu are facilitated by the two display buttons on the right and left of the display.

The "operation via display buttons" parameter in the "B.IQ RTR push button" parameter branch specifies to what extent local operation from the programming menu will be possible.

- "No operation":

  'Normal operation' and local operation of the regulator by actuation of the display buttons for shifting the basic set value possible.

- "Limited operation":

  Switching over into the programming mode is possible. → 'Normal operation' including set value shifting and switching over the operating mode and resetting of the different heating and/or cooling set values is possible.

- "Complete operation":

  Full access to the device for local operation. In addition to limited operation, it grants the user access to the up to three timers (activation/deactivation of the timers and changing of the individual switching programs) and to the "settings" menu (activation/deactivation of the push button assistance function, the LCD background light and key disabling as well as of the display of the version data).

Note:

- A few functions of the programming menu (e. g. set value resetting, setting the timers) will only be possible if access to these parts has been enabled in the ETS plug-in or if these functions are available at all, respectively.
  Moreover, the operation of the regulator (display buttons) can be disabled. From the ETS plug-in, you can parameterize whether disabling of the regulator operation shall always take place or shall be object-controlled (cf. "4.6.2 Disabling controller operation").
- In its function as a controller extension, control via the display keys is only limited. Access to the programming menu under the "complete operation" item will only facilitate access to the two timers (activation/deactivation of the timers and changing of the individual switching programs) and to the "settings" menu (activation/deactivation of the push button assistance function, the LCD background light and key disabling as well as of the display of the version data). The keys for displaying and shifting the basic setpoint in the 'normal mode' are deactivated.

You can use two display buttons on the left and right of the display and the rockers or buttons of the push button to operate the programming menu. During the time the programming menu is activated (display reading "**Prg**"), the functions of the display buttons will be indicated on the display directly opposite to the buttons. In the following, the programming mode operating functions are explained:

Functions of the display buttons:

OK (left button)	To call the menu item selected, or to change a setting. When this symbol is hidden you cannot call the menu item selected, or you cannot change the setting. In this connection, it may generally not be possible to call a function, or such function has been disabled by the ETS plug-in.
▲ (right button)	To change between the individual main menu items and between the submenus selected.
Enter (both buttons pressed at the same time)	To call the programming menu (keep the button pressed for at least 3 s) or to accept a value set, such as a set value or a switching time (keep the button pressed for at least 1 s).
- (left button)	To change the (blinking) value selected (e. g. temperature value or switching time) into the negative direction down to the adjusting limit. In 'normal operation' (no programming mode activated), you can use these buttons to shift the set value.
+ (right button)	To change the (blinking) value selected (e. g. temperature value or switching time) into the positive direction up to the adjusting limit. In 'normal operation' (no programming mode activated), you can use these buttons to shift the set value.

Functions of the push button rockers or buttons:

Escape (any push button button)	If you want to exit the programming mode or the menu selected (return to the higher-order menu) press any button of the push button (with the exception of the display buttons). If you do not wish to accept a value you have changed also execute the escape command. In such case, the value you have changed will not be stored in the device. The control will automatically execute an escape command if no further button operation takes place for about 20 s. Note: For settings you have made with the aid of the "**OK**" button, (e. g. activation/deactivation of functions) the last one will be stored in the device, even though you execute an escape command. A temperature set value shift made by means of the display buttons will also be accepted upon an escape command.

1.4.1 Local operation in the normal mode

The device will be in normal operation unless you have activated the programming mode. Depending on what you have parameterized (cf. "2.2 Display data in normal operation"), the display window will show as standard readings the current room temperature, the outside temperature, the setpoint temperature, an info text, a value display, a scene display, the time or the date.

If you actuate one of the display keys or one of the function keys parameterized in the ETS as "setpoint shifting" you will see the actual setpoint shifting on the display. Depending on the *"setpoint shifting display"* parameter in the *"display"* parameter branch, the setpoint shifting is shown on the display either as an absolute value or relative to the setpoint temperature.
The absolute reading shows the currently adjusted setpoint temperature of the active operating mode. For a new setpoint shift, the new temperature value will be calculated by the push button module and fully read on the display. Example: Displayed setpoint temperature without shifting: 21.0 °C → new setpoint shifting: +0.5 °C → new setpoint temperature displayed: 21.5 °C.
The relative display will only show the current setpoint shifting in °C without also reading the setpo int temperature derived thereof. Example: Setpoint temperature without shifting: 21.0 °C (reading: 0) → new setpoint shifting: +0.5 °C → reading: +0.5 °C.

You can press the right or left display key or the corresponding function key to shift the setpoint temperature in increments or decrements of 0.1 or 0.5 °C (depending on the ETS parameterization). This setpoint shifting (temperature offset of basic temperature) can be preset for each mode and optionally accepted when switching over to another operating mode (e.g. comfort mode → standby mode) affecting all of the controller's operating modes. More information on presetting or shifting the basic set-temperature can be found in chapter "4. Room temperature regulator functions".

The hand symbol " " on the display indicates that a basic setpoint shifting has been set. The set temperature value is instantly accepted as the new setpoint.
In case the setpoint shifting was set via the display keys, the display can be switched back to default by using the escape command (pressing any of the push button's keys / not a display key).
In case the setpoint shifting was preset via a function key, the actuation of another function key will cause the display to switch. In this case, however, the stored key function will also be executed.
If no other keys have been pressed for approx. 20 sec. The display will also switch back in both cases.

Notes:

- Following the return of bus voltage the controller always runs in normal operation!
- It has to be pointed out that the display keys or the function keys that are parameterized as „setpoint shifting" are influenced by a push button disable function or by disabling the controller. While disabling is activated (" " symbol is visible on display) any actuation of the keys, if applicable, will be ignored.
- If the push button functions as a controller extension, the display keys for the display and shifting of the basic setpoint will be deactivated.

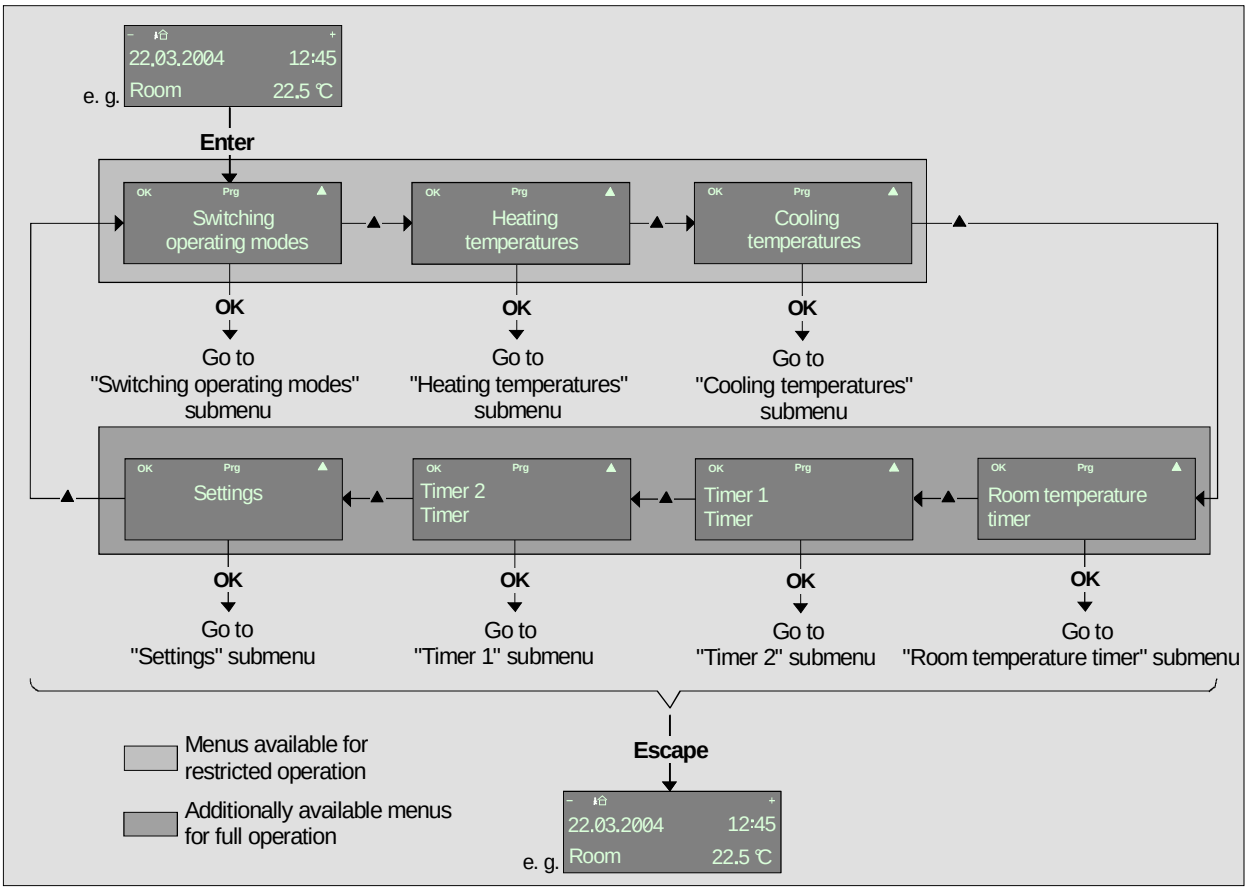
1.4.2 Local operation in the programming mode

In the programming mode, you can activate or deactivate various functions or change settings. So you can switch over the operating mode, change the temperature set values, set the timers or do any other basic settings.

Note:
In general, individual functions or settings may not be accessible, due to the parameterization of the device in the ETS plug-in.

For more information on how to set the operating modes and the room temperature timer or on their functions, respectively, please refer to chapter "4. Room Temperature Regulator Functions". The timers are described in chapter "5. Timers".

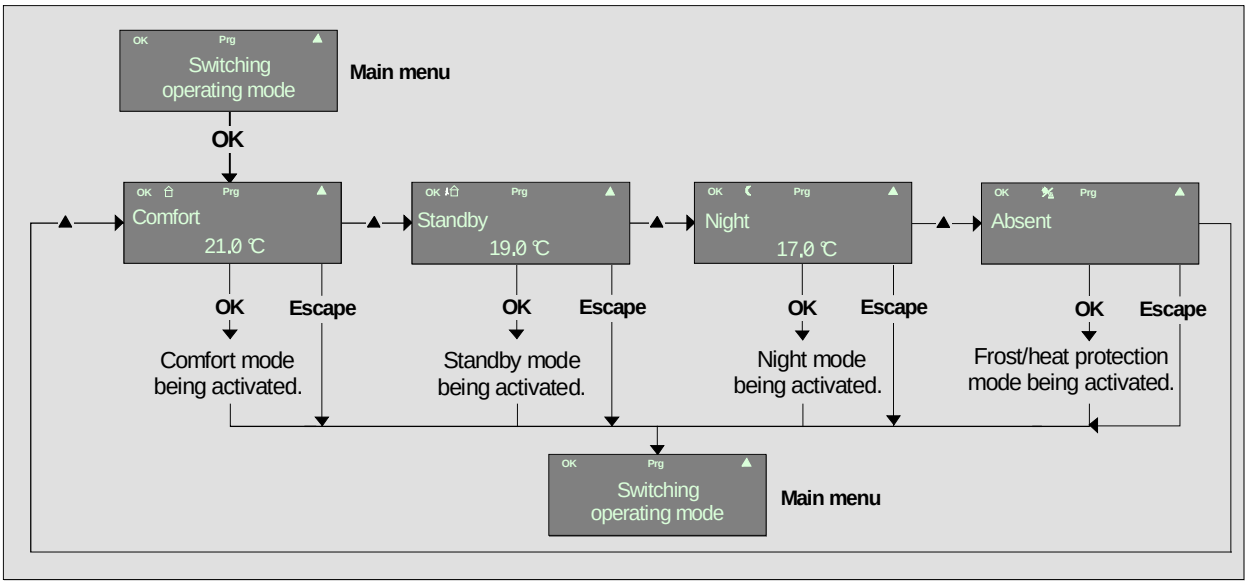
In the following, the individual menu items of the programming mode are explained:



Press both display buttons for about 3 s to call the programming mode. The display will read "**Prg**" and you will be in the main menu. You can use the " ▲ " right display button to change between the individual menu items from there. Press the "**OK**" left display button to call the corresponding submenus after you have selected the main menu.
The "heating temperatures" and "cooling temperatures" menus are visible separately or together, depending on the operating mode parameterized in the ETS plug-in. Also in the "heating and cooling" mixed mode, the cooling temperatures and thus the menu can be separately disablinged.
The submenus for the individual timers also depend on the parameterization and can be suppressed if you have not enabled these functions in the ETS plug-in. Please note that you can preset the designation of the first or second timer in the ETS plug-in. In this connection, the designation entered in the ETS plug-in will also be shown in the first line of the display (20 characters max.).
If you press any button of the push button (except the display buttons), or if you do not operate any button for about 20 s you will return to normal operation.

1.4.2.1 "Operating mode switch-over" submenu

Press the "**OK**" left display button from the main menu to call the "operating mode switch-over" submenu. This will not be possible if a higher-priority mode (e. g. window contact/presence detector) or the KONNEX override object has been activated. In such case, the "**OK**" symbol will be hidden when you select the operating mode switch-over option, with the operating mode remaining unchanged.

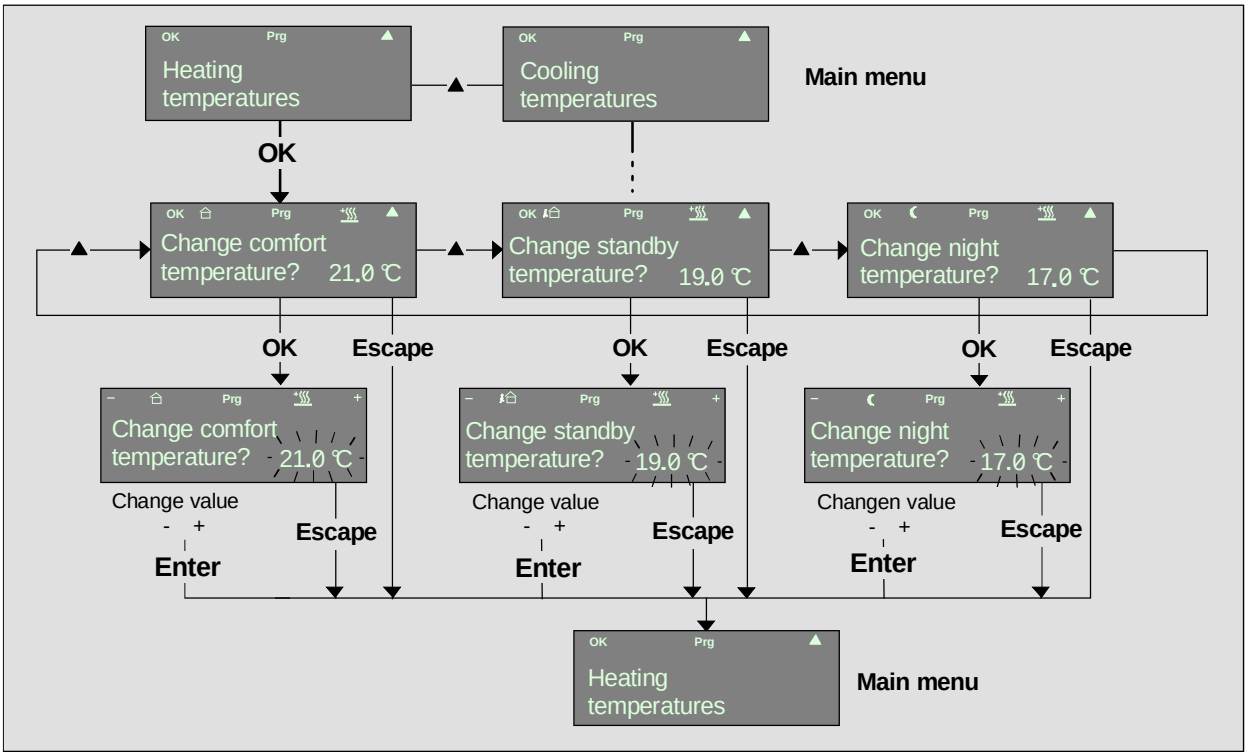


Press the "**OK**" left display button to accept the operating mode you have selected with the " ▲ " right display button. After this, the display will return to the main menu of the programming mode.
The temperature values shown on the display for the different operating modes represent the temperature set values of the selected mode to be temporarily expected. Taking account of the present operating mode of the room temperature regulator and the set value shift possibly made, those set values the regulator will take over as new ones when the operating mode is changed will be displayed here. Please note that you can change these temperature values at any later time during the 'running operation' of the regulator by local operation or by a basic set value change through the object, if enabled.

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

1.4.2.2 "Heating temperatures" or "cooling temperatures" submenu

Press the "**OK**" left display button from the main menu to call the "heating temperatures" or "cooling temperatures" submenu. Depending on the operating mode set in the ETS plug-in, these menus will be visible and can be available alternatively or together (mixed mode). In the mixed mode, you can disabling the cooling temperature set values during local operation by setting the *"modification of the setpoints 'cooling'"* parameter in the *"room temperature regulator function/set point values"* parameter branch to *"deactivated*.
The procedure is identical for the heating and cooling temperature values so that the menu structure will be described here only once. The procedure for the different temperatures does not differ either.



Once a setpoint has been set and the input has been confirmed by an **Enter** command, the operating mode associated with the adjusted setpoint will be accepted as the active mode. This, however, is only the case if no higher-priority mode (e.g. window contact / presence detector) is activated or if the KONNEX forced object is not activated!

Example 1:
1 – Comfort mode " ⌂ " is active via push button operation
2 – Switching into the programming mode
3 – Adjusting the setpoint for the night mode " ☾ "
4 – Confirming the new setpoint (**Enter**) – Switching into normal operation
5 – Night mode is activated " ☾ „!

Example 2:
1 - Presence detector is active (comfort mode " ⌂ ")
2 – Switching into programming mode
3 – Adjusting the setpoint for the night mode " ☾ "
4 – Confirming the new setpoint (**Enter**) – Switching into normal operation
5 – Comfort mode " ⌂ " is still active!!

The operating mode can only be switched if the associated setpoint in the ETS plug-in is enabled for local adjustment (cf. "4.4 Temperature setpoints").
If more setpoints are to be set, the steps described here have to be followed again.

Setting the temperature setpoints:

The setpoints for the
- "comfort ⌂",
- "standby ⌂" and
- "night ☾„ operating modes can be reset.

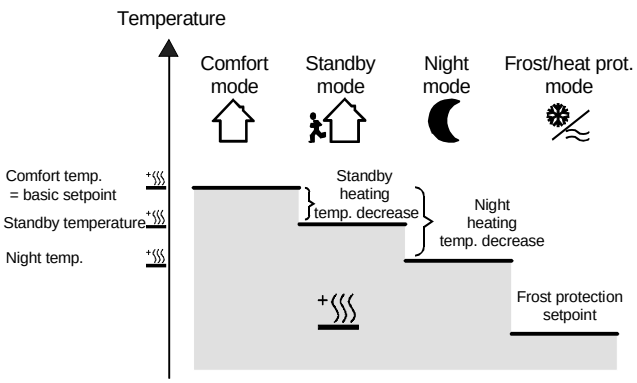
Depending on the control option enabled in the ETS plug-in, up to 6 different values are available for setting the setpoints.
It has to be pointed out that some setpoints in the ETS have not been freed for local control. Thus they can only be seen on the display window and cannot be changed (cf. "4.4. Temperature setpoints")!
Also, in case the second control circuit has its own setpoints, only the temperature values of the first control circuit are to be set in the programming mode.

The "setpoint shifting step value" parameter in the "room temperature regulator function/setpoint values" parameter branch defines the step width (0.1 K or 0.5 K) for each value readjustment key actuation in the programming menu.

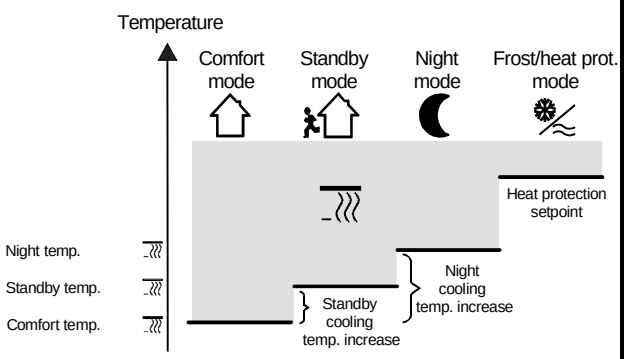
The following table shows the values to be set:

activated operating mode	parameterized control option			
			Heating and cooling	
	Heating	Cooling	for heating	for cooling
Comfort ⌂	e.g. 23.0 °C Comfort setpoint = Basic setpoint	e.g. 27.0 °C Comfort set-temperature = Basic setpoint	e.g. 23.0 °C Comfort set-temperature = Basic setpoint t – ½ Dead band symmetrical for Dead band / = Basic setpoint assymmetrical for Dead band	e.g. 27.0 °C Comfort set-temperature = Basic setpoint t + ½ Dead band symmetrical for Dead band / = basic setpoint + Dead band assymmetrical for Dead band
Standby ⌂	e.g. 21.0 °C Standby set-temperature	e.g. 29.0 °C Standby set-temperature	e.g. 21.0 °C Standby set-temperature	e.g. 29.0 °C Standby set-temperature
Night ☾	e.g. 19.0 °C Night set-temperature	e.g. 31.0 °C Night set-temperature	e.g. 19.0 °C Night set-temperature	e.g. 31.0 °C Night set-temperature

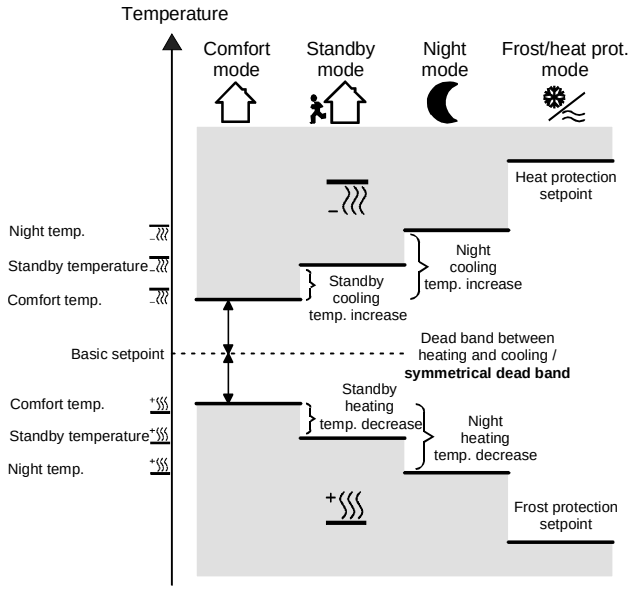
"Heating"

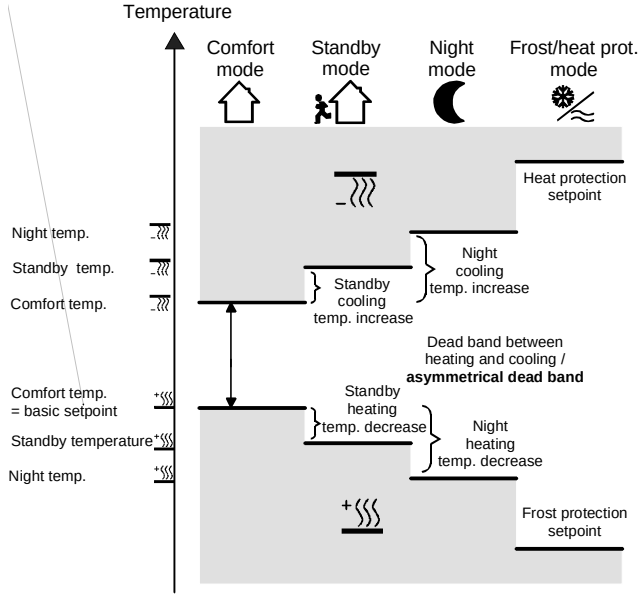


"Cooling"

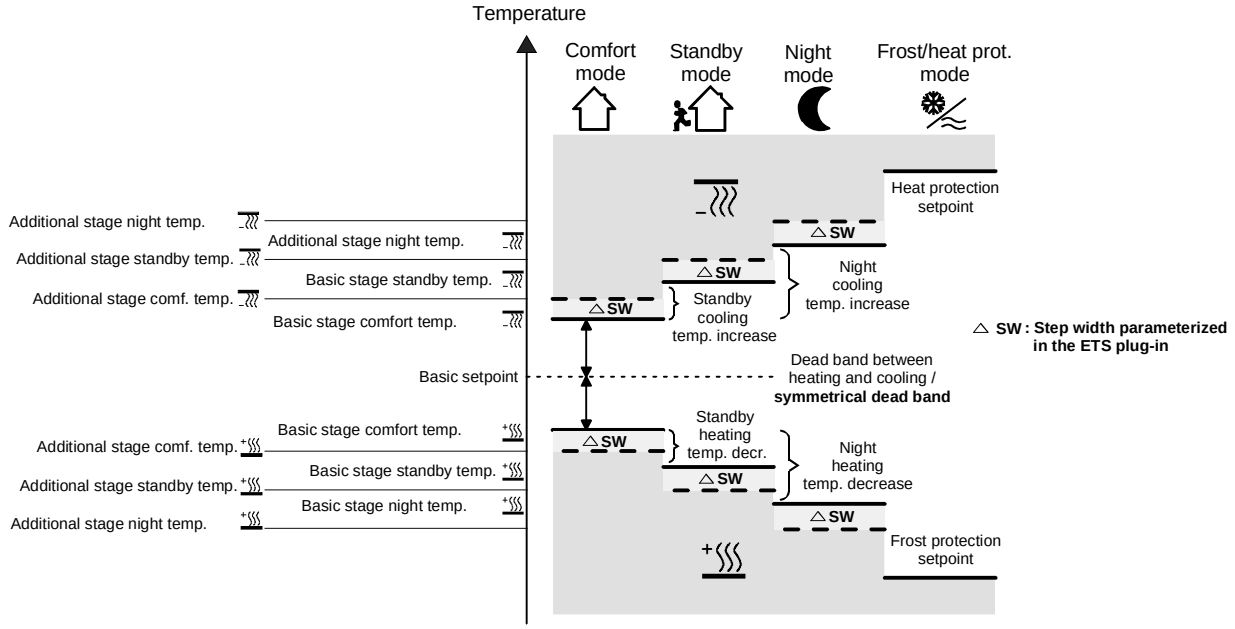


"Heating and Cooling"





Temperature control with additional stage considering as an example "heating and cooling" / symmetrical Dead band



6 temperature setpoints can be set in the *"heating and cooling"* control option if enabled in the ETS plug-in. Depending on the temperature decrease, increase or Dead band parameterized in the ETS, all temperature setpoints are derived from the basic set-temperature.
It must be pointed out that changing the set-temperature for heating in the comfort mode will also adjust all other set-temperature values!

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

$$T_{\text{Comfort Set Cooling}} - T_{\text{Comfort Set Heating}} = T_{\text{Dead band}};\ T_{\text{Comfort Set Cooling}} \geq T_{\text{Comfort Set Heating}}$$

Important:

- If the Dead band is symmetrical, the basic setpoint is indirectly set via the comfort temperature for heating. The basic setpoint is not shown on the display even with local control!
- Changing the comfort set-temperature for cooling allows the adjustment of the Dead band. An adjustment of the Dead band with a symmetrical dead band position will result in a shifting of the comfort set-temperature for heating and thus of all other temperature setpoints. With an asymmetrical dead band position, an adjustment of the comfort set-temperature for cooling will only shift the temperature setpoints for cooling. It is possible to shift the Dead band to 0 °C via local control ($T_{\text{Comfort Set Cooling}} = T_{\text{Comfort Set Heating}}$). In this case there will be neither heating nor cooling if the measured room temperature equals the comfort set-temperatures.

The set-temperatures for "standby" and "night" are derived from the comfort set-temperatures for heating or cooling. The temperature increase (for cooling) and the temperature decrease (for heating) of both operating modes can be preset in the ETS plug-in.
It is possible to adjust the set-temperatures for "standby" and "night" via local control in the programming mode independent of the values for the temperature increase/decrease, which were originally parameterized in the ETS.
In this case, the adjustment of the basic set-temperature or the Dead band, the standby or night set-temperatures will always shift together with the temperature increase/decrease resulting from the local control. After the reprogramming with the ETS, the originally parameterized values can be accepted again.

The following applies:

$$T_{\text{Standby Set Heating}} \leq T_{\text{Comfort Set Heating}} \leq T_{\text{Comfort Set Cooling}} \leq T_{\text{Standby Set Cooling}}$$

or

$$T_{\text{Night Set Heating}} \leq T_{\text{Comfort Set Heating}} \leq T_{\text{Comfort Set Cooling}} \leq T_{\text{Night Ste Cooling}}$$

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

Basically, one has to distinguish between two cases when adjusting the basic setpoint for the temperature (e.g. by adjusting the comfort set-temperature for heating in programming mode):

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

It is possible to determine via the *"accept modification of the basic temperature setpoint value permanently"* parameter in the *"room temperature regulator function /set point values"* parameter branch whether the set basic temperature value shall be stored in memory permanently (*"Yes") or only temporarily ("No"*).

Case 1:

If the basic temperature setpoint is adjusted, it will be permanently stored in the push button's EEPROM memory. The newly adjusted value will overwrite the basic temperature originally parameterized via the ETS!

It has to be taken into account that…:

- frequent adjustments of the basic temperature (e.g. several times a day) can affect the product life of the device as the non-volatile storage is designed for less frequent write access.
- alternatively to the local adjustment of the basic setpoint, this temperature can also be predetermined through the "basic setpoint" via the bus, if enabled in the ETS plug-in.

Thus the basic setpoint adjusted on the push button or received by the object remains in memory even after a bus voltage failure.

Case 2:

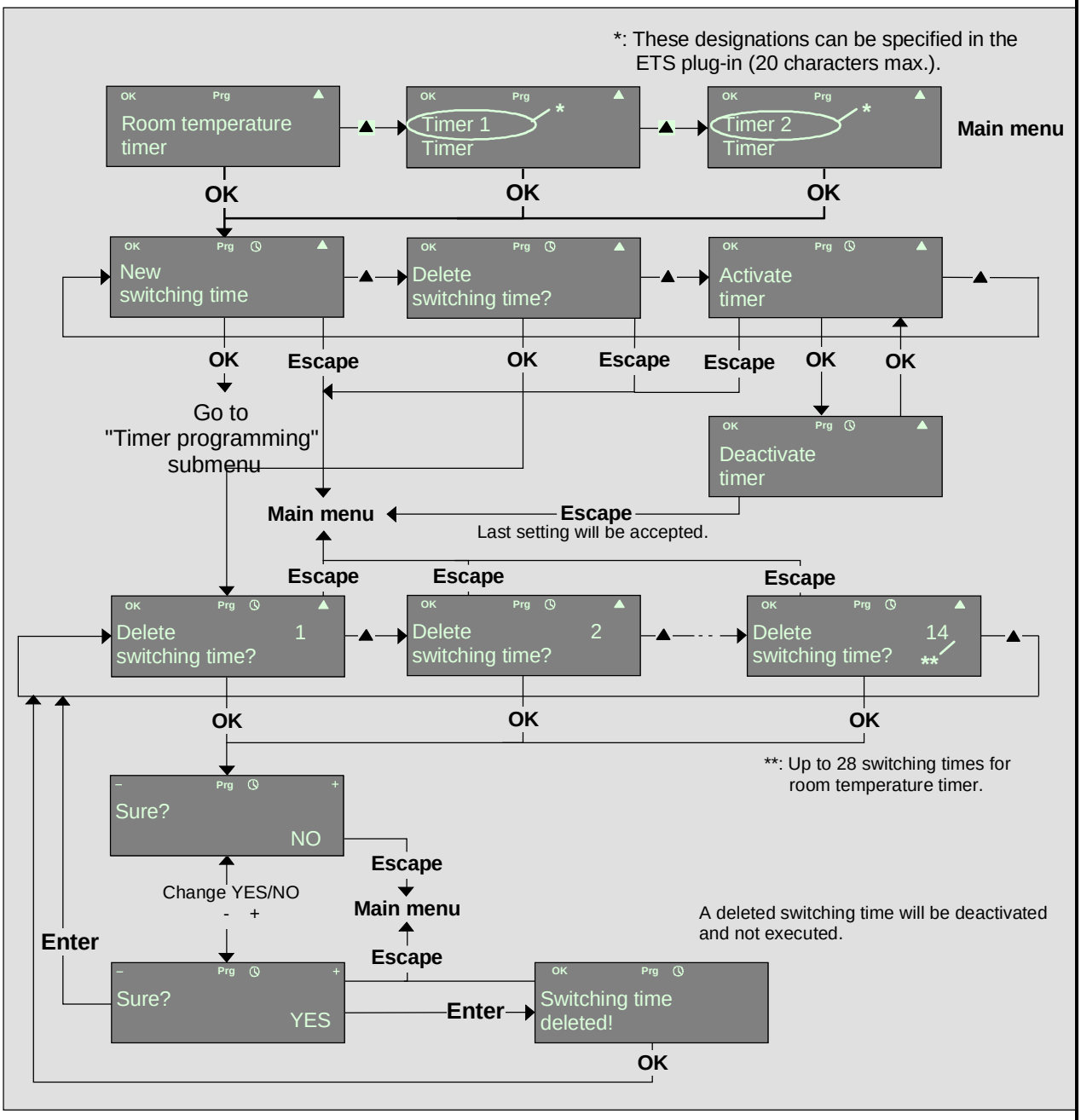
The basic setpoint, which was set on the push button or received via the object, stays only temporarily active in the current operating mode. In case of a bus voltage failure or following a switch-over into another operating mode (e.g. comfort after standby) the basic setpoint adjusted via local control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

Notes:

- Since the set-temperatures for the "standby" and "night" operating modes or the setpoints for the "cooling" control option are derived from the basic set-temperature for "heating" - in consideration of the increase, decrease or Dead band values that are parameterized in the ETS plug-in - these set-temperatures will shift linearly by the basic change of the setpoint value.
  The temperature setpoints for the standby or night mode or "cooling" comfort mode (Dead band) will always be stored non-volatile in the EEPROM memory.
- It has to be pointed out that temperature setpoints can only be changed or stored via local control or via the "basic setpoint" object, if it was enabled in the ETS plug-in (cf. "4.4 Temperature setpoints")!
  Any value preset via local control will not be accepted into the object.

1.4.2.3 "Timer" submenus

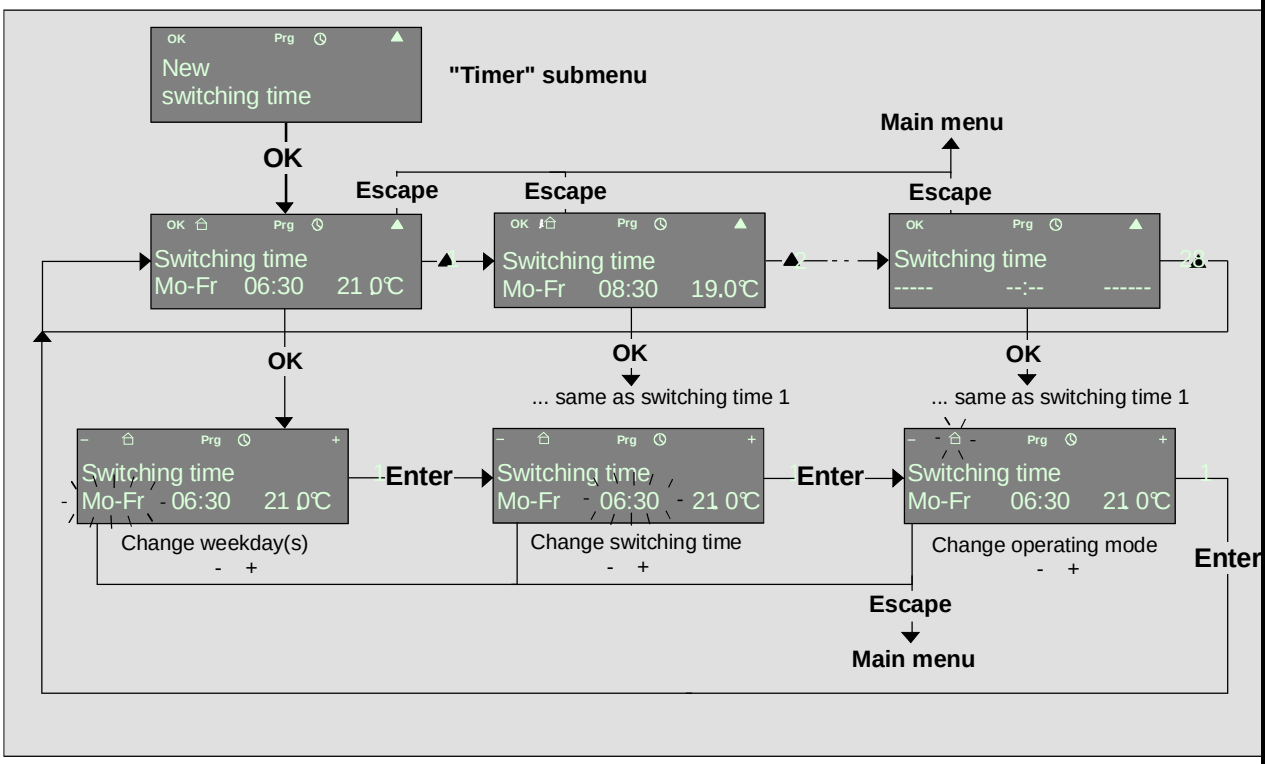
The B.IQ push button with room thermostat (RTR) and display knows up to three different timers. These are the "room temperature timer" with its up to 28 different switching times and "timer" 1 or 2, each of the holding up to 14 different switching times. In the programming mode, you can change, create or delete the individual switching times as well as activate or deactivate the timers.
The individual timers must always have been enabled in the ETS plug-in, and full operation must be accessible through the display buttons until local timer operation is possible in the programming mode. Press the "**OK**" left display button from the main menu to call the "room temperature timer" or "timer" submenu. Please note that you can give the two timers their own designations in the ETS plug-in (20 characters max.) which will also be shown in the first line of the display. Thus, easy and unambiguous assignment or identification of the timer will be possible. The selection menu shown below is identical for all timers:



The submenus for programming the switching times of the two timers differ from the room temperature timer as follows:

Room temperature timer "timer programming" submenu:

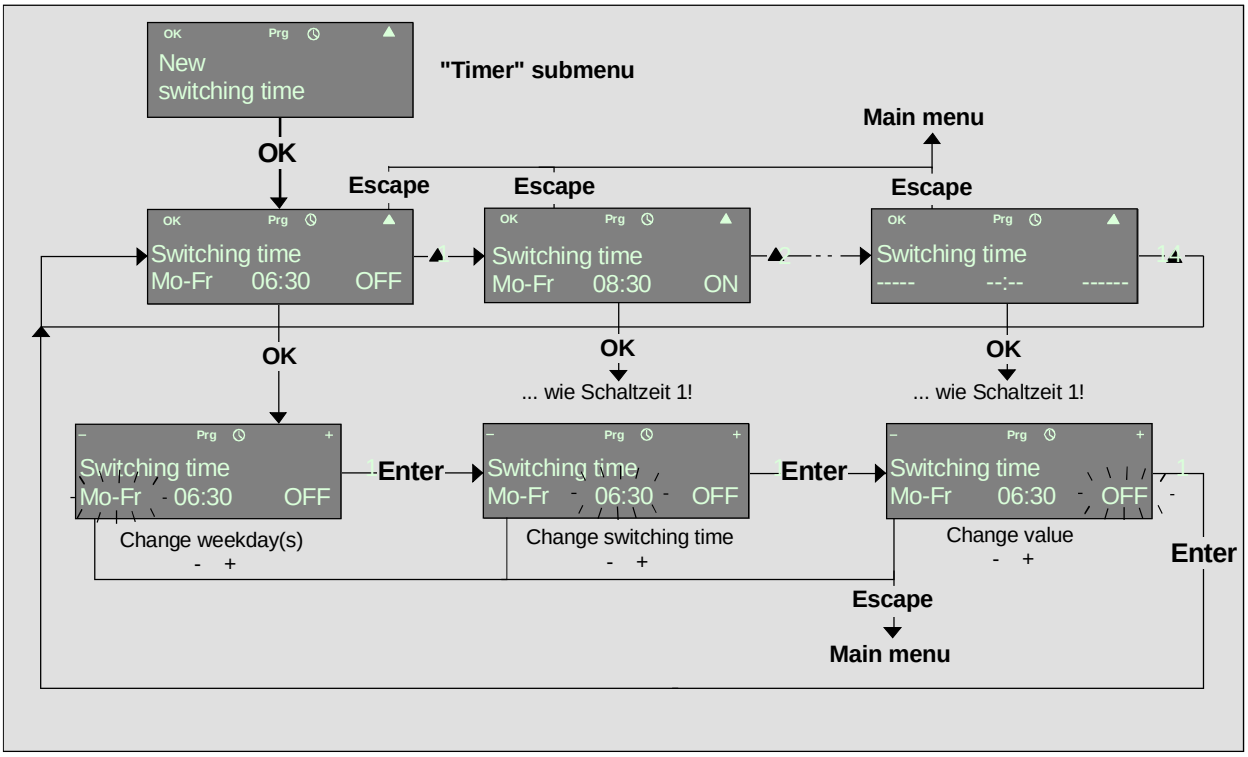
The room temperature timer can distinguish up to 28 different switching times and facilitates the switch-over of the room temperature regulator operating mode exactly to the minute, depending on the time and the day of the week. The switching times will be executed in chronological order.



The temperature values shown on the display for the different switching times represent the temperature set values of each preselected operating mode to be temporarily expected. Taking account of the present operating mode of the room temperature regulator and of the set value shift possibly made, those set values the regulator will take over as new ones as soon as the switching time is executed will be displayed here. Please note that you can change these temperature values at any later time during the 'running operation' of the regulator by local operation or by a basic set value change through the object, if enabled.

Timer 1 to 2 "timer programming" submenu:

Each of these up to two timers (seven-day timers) can distinguish between 14 different switching times and facilitates the transmission of bus commands precisely down to the minute, depending on the time and the day of the week. The switching times will be executed in chronological order.



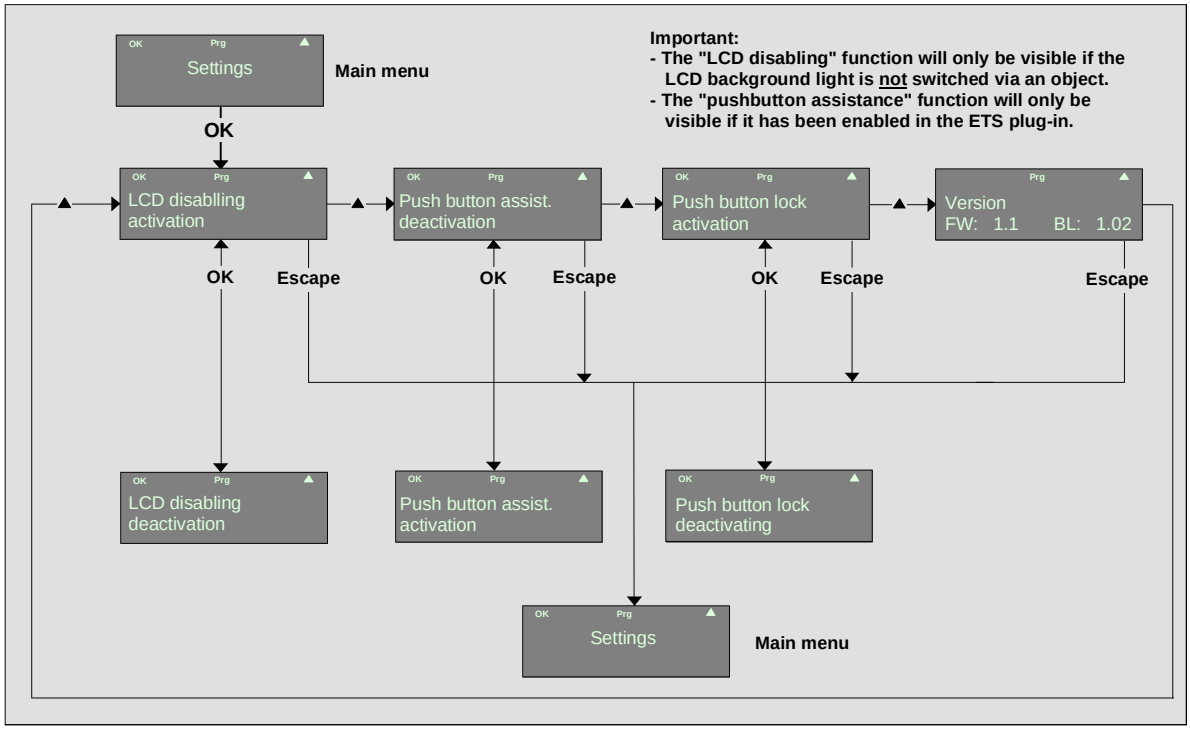
You can send switching commands (ON, OFF), value commands (0...255 / 0...100 %) or scene recall commands (1...8) to the bus and edit them from the programming menu. The data format (switching, value, scene) of the command to be transmitted is parameterized in the ETS plug-in and cannot be changed locally. Depending on the selection made in the ETS plug-in, the value range and the representation of the command are adapted in the programming menu.

Data Format	Value Type	Display Reading
Switching	---	OFF ("0") ON ("1")
Value	0...255 0...100 %	0...255 (Step width: 1) 0...100 % (Step width: 1 %)
Scene recall	---	S01 ... S08

The two timers can have different data formats. However, the up to 14 switching times of a timer have all one common data format.

1.4.2.4 "Settings" submenu

Press the "**OK**" left display button from the main menu to call the "settings" submenu. From this submenu, you can activate the automatic LCD background (LCD illumination) light switch-off or the button lock functions, depending on the parameterization in the ETS. In addition, the current software version data can be displayed. The "settings" submenu will only be visible when the "complete operation" option of the programming menu is active.



LCD background light switch-off:
In the ETS plug-in, you can specify how you want to trigger the LCD background light in general. In this connection, the "permanently ON", "automatic switch-off" and "switching via object" settings are possible. If the light is not switched through the object you can activate automatic switch-off from the programming menu.
For an LCD background light switch-off activated in the programming mode, either the switch-off time preset in the ETS plug-in (for "automatic switch-off" parameterization) or the default time of 10 s (for "permanently ON" parameterization) will be used as illumination period.

Push button assistance function:
A push button assistance function enabled and configured in the ETS plug-in can be deactivated in the programming mode at any time. The button assistance function deactivated from the menu will always be activated after initialization (after a reset). Please note that the button assistance function will be suppressed if the button lock function is active.

Push button lock:
If you have activated the push button lock function (e. g. protection of children) in the programming mode the rockers of the push button will be disabled. In such case, rocker actuation will show no response. This type of disabling is independent of a push button disabling function initiated via the bus. If the button lock function has been activated in the programming mode the " " symbol will not appear on the display.
The display buttons will still be in operation after an disabling function has been activated from the programming menu.
The disabling function activated from the menu will always be deactivated after initialization (after a reset).

Display of the software version:
The version of the firmware loaded (e. g. **FW: 2.0**) and the software version of the bootloader (e.g. **BL: 1.02**) will be displayed. Together with the ETS plug-in, the B.IQ push button with room thermostat (RTR) and display facilitates firmware updating so that future function extensions will be possible without replacing the device.

2. Display

2.1 Basic function and display-illumination

The display unit is located behind the black transparent window between the two display buttons. On this unit, various functions of the integrated room temperature regulator or also of the push button can be displayed.
You can use the *"Illumination"* parameter in the *"display"* parameter branch to preset the function of the display background light. You can permanently switch on the light (*"ON"* setting), allow it to switch off automatically (setting: *"automatic switch-off"*/default) or switch it through a separate object (*"switching via object"* setting).

If you select automatic switch-off the light will switch on if you press any button and will then automatically go out after the time you have set by the *"automatic illumination switch-off"* parameters has elapsed. In this connection, you can preset switch-on times between approx. 1 s and approx. 20 min.

Alternatively, the light can be switched via object *"switching display"*. The polarity of this object is fixed. So at an object value = "1", the light will be on, whereas it will be off at "0". After bus voltage recovery, the object value will always be "0".
In addition, the *"kind of switching"* parameter defines whether the light will remain permanently on at an object value = "1" (*"ON"* setting), or whether it will be automatically disabled after a parameterized time (*"automatic switch off"* setting). In the latter case, the display background light will only be switched back on if another "1" telegram is received via object. A "0" telegram will always immediately switch off the light.

The blue device ON LED will always be triggered together with the display background light.

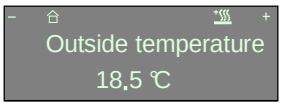
2.2 Display data in normal operation

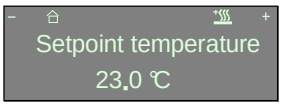
In the normal operating mode of the push button module (standard display), the following information can be read on the display: the current room temperature, the outside temperature, the setpoint temperature, an info text, a value display reading, a scene display, the date or the time.
By the *"display of"* parameter in the *"display"* parameter branch in the ETS plug-in, you can define which of such information will be actually displayed.
In this connection, you can also have more than one piece of information displayed at the same time (e. g. *"date/time/room temperature"* parameter setting). In such case, all three pieces of information will be displayed at the same time.
In the preview of the ETS plug-in (*"configuration"* menu → *"preview"* item) you can already see the configured display of the push button module prior to a download.
The operation of the push button module in its normal mode is described in chapter "1.4.1 Local operation in the normal mode". The subsequent chapters will describe the possible display information in more detail.

2.2.1 Temperature display

The display can read the room temperature (actual temperature of the first control circuit) determined by the regulator and, additionally or alternatively, the outside temperature (exterior) received via the bus and/or the current set value temperature of the first control circuit.
The room temperature reading has a resolution of 0.1 ℃ and covers a range from -99.9 ℃ to +99.9 ℃.
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 ℃ and covers a range from -99.9 ℃ to +99.9 ° C.
The reading will refresh as soon as a temperature-value telegram is received via the object *"external temperature sensor"*. If parameterized, the outside temperature will only be read on the display and cannot be used for any further temperature or variable calculation.
The setpoint temperature display depends on the parameterized *"setpoint shifting display"*.
Absolute display:
The absolute reading shows the currently adjusted setpoint temperature of the active operating mode. The resolution of the setpoint temperature display depends on the parameterized setpoint shifting incrementation (*"setpoint shifting step value"* parameter in the *"room temperature regulator function/setpoint values"* parameter branch). For a *"0.1 K"* incrementation, the display will refresh upon each change by 0.1 ℃. If you select the *"0.5 K"* step width the push button module will always round the setpoint temperature to half a degree and read this rounded temperature on the display. Its possible temperature range depends on the parameterized mode and is given by the fixed values for the frost and/or heat protection temperature. The reading will refresh once a new setpoint temperature for the regulator 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.
Relative display:
The relative display of the setpoint will only show the current setpoint shifting in ℃, taking into consideration the incrementation, without also reading the setpoint temperature derived thereof. Example: Setpoint temperature without shifting: 21.0 ℃ (rea ding: 0) → (new setpoint shifting: +0.5 ℃ → reading: +0.5 ℃. This representation corresponds to the adj usting wheel of a conventional room temperature regulator.

Temperature display examples:



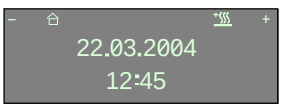


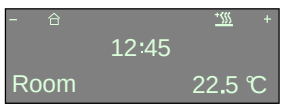
2.2.2 Date and time display

If parameterized, the date and the time can be read on the display. In this connection, the time received via the *"time signal"* object and the date received via the *"date"* object 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 weekday information is obtained from the time signal received and is used for the control of the timers. The weekday is not displayed. The time is always displayed in the 24-hour format.

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 or date signal has been received via the objects, 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 at the expected time error, with any possibly parameterized time-dependent control functions or the room temperature timer still being executed. Moreover, the internal time and date will get lost after bus voltage failure so that the time and date signals of a KNX/EIB system master clock, for example, should be transmitted to the bus after bus voltage recovery

Date and time display examples:





2.2.3 Value display

The display allows the reading of an independent value. The value display has a separate communication object through which display values are received. Depending on the *"data type"* parameter, the reading of 1-bit switching values ("1" / "0") or the display of 1-byte or 2-byte value telegrams is possible.
For each type of display, you can specify a 20-character descriptive text in the ETS plug-in to be read on the display of the push button module over the value description (e. g. "outside temperature" or "bath light" or "corridor brightness value".
After the initialization of the push button module the object value of the display will always be "0".

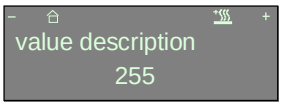
Value display for the "switching" data type:
You can specify display texts for the two possible telegram polarities. The *"text for '0' telegram"* and *"text for '1' telegram"* parameters characterize the text read on the display when a corresponding switching telegram is being received.

Switching value display examples (two display texts):

Value display for "value (1 byte)" data type:
You can use the *"sign"* parameter to define whether the values should evaluated and displayed with or without a sign. For a display without sign, you can also set the *"display format"* in the ETS plug-in, whereby the value display can be matched to the data point type used. Possible display formats are… *"0…255"*: DPT-ID 5.004 or 5.010 / *"0…100"*: DPT-ID 5.001 / *"0…360"*: DPT-ID 5.003.

For display including the sign, the display format is fixed to "-128…127" (DPT-ID: 6.001 or 6.010).
In addition, a unit for the display value can be firmly specified for any display format. The *"unit"* parameter enables the display (setting *"yes"*). When the display is enabled, you can parameterize in the ETS plug-in the *"unit text"* with a maximum of 10 characters. This unit text will be shown on the display of the push button module, directly behind the display value.

1-byte value display examples:



Value display for "value (2-byte)" data type:
For a 2-byte display, you should specify the *"number format"* first. What can be displayed are integers or floating-point values. For integers, value display with or without sign is possible. Possible display formats are…
without sign: 0…65535 (DPT-ID 7.00x) / with sign: -32768…32767 (DPT-ID 8.001).
For the display of floating-point values (e. g. temperature or brightness values / DPT-ID 9.0xx), you can use the *"display format"* parameter to specify the number of visible digits behind the decimal point.
In addition, a unit for the display value can be firmly specified for any number and display format. The *"unit"* parameter enables the display (setting *"yes"*). When the display is enabled, you can parameterize in the ETS plug-in the *"unit text"* with a maximum of 10 characters. This unit text will be shown on the display of the push button module, directly behind the display value.

2-byte value display examples:

2.2.4 Displaying an information text

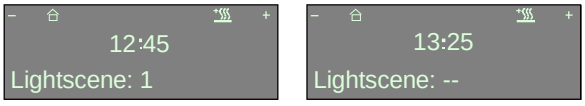
On the display of the push button module, it is possible to read an information text with a maximum length of 14 characters. What is possible to display is, for example, general information from the house management or text messages from alarm centres.
To activate such text display set the *"display of"* parameter in the *"display"* parameter branch to *"setpoint temperature/text"*. So, a combined display including the current temperature setpoint value is exclusively possible.
In this display configuration, the first line reads the setpoint temperature of the currently active operating mode (depending on the *"setpoint shifting display"*, maybe only the relative shift in ℃). In the seco nd line, the text string received via the *"info text"* object is displayed.
The information text will remain visible on the display as a standard reading until a new text is received. After initialization, some text will only be displayed if a text string has been received via the object. The other text display messages of the push button module (cf. "2.3 Text display (alarm text)" and "2.4 Text display via 3 x 1 bit") will overwrite the standard display, thus deleting the information text.

Info text display examples:

2.2.5 Displaying a scene number

On the display of the push button module, it is possible to read the number of the internal scene 1…8 of the B.IQ push button with room thermostat (RTR) and display recalled or saved last if the scene function has been enabled.
To activate the display of the scene number set the *"display of"* parameter in the *"display"* parameter branch to *"time/light scene"*. So, a combined display including the current time is exclusively possible.
In this display configuration, the first display line reads the time. In the second line, the number of the internal scene recalled last will appear. You can reset the scene number display via the *"scene display"* object. If a "1" telegram is received through this object the display will be reset, thus hiding the scene number. Instead of the scene number, the display will show dashes ("--") to symbolize that 'no scene is active'. The reception of a "0" telegram by this object will show no response.
The scene number will remain visible as standard reading until a new internal scene is recalled. After initialization, a scene number will only be displayed if a scene is recalled (or the display will read "--"). The text display messages of the push button module (cf. "2.3 Text display (alarm text)" and "2.4 Text display via 3 x 1 bit") will overwrite the standard display, thus deleting the information text.

Scene display examples:



Important:
Internal scene numbers 1…8 of the B.IQ push button with room thermostat (RTR) and display will solely be displayed, i. e. that these internal scenes must have been recalled or saved. This means that no numbers of any scenes will be displayed that were recalled exclusively via the scene extension of the push button sensor function in a different bus device (e. g. light scene push button sensor). An internal scene of the B.IQ push button with room thermostat (RTR) and display can be recalled either through scene extension object 72 of the scene function, or via a "light scene extension/internal scene recall" push button sensor function
The B.IQ push button with room thermostat (RTR) and display will not automatically check whether a scene recalled before (e. g. light scenario) is still set or has already been 'subsequently' changed. Therefore, the push button module will not automatically reset the scene display.
The display of the scene number can only be parameterized after the scene function has been enabled.

2.2.6 Symbols

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

Symbol displayed	Description
-	To change the (blinking) value (e. g. temperature value or switching time) selected in the programming mode by means of the left display button into the negative direction down to the adjusting limit. In 'normal operation' you can use the left display button to shift the set value.
+	To change the (blinking) value (e. g. temperature value or switching time) selected by means of the right display button into the positive direction up to the adjusting limit. In 'normal operation' you can use the right display button to shift the set value.
OK	To call the menu item selected in the programming mode or to change a setting by means of the left display button. When this symbol is hidden you cannot call the menu item selected, or you cannot change the setting. In this connection, it may generally not be possible to call a function, or such function has been disabled by the ETS plug-in.
	"Comfort" mode active or "comfort" mode preselected from the programming menu, or to display the comfort set value temperature.
	"Standby" mode active or "standby" mode preselected from the programming menu, or to display the standby set value temperature.
	"Night" mode active or "night" mode preselected from the programming menu, or to display the night set value temperature.
	"Frost/heat protection" mode active or "frost/heat protection" mode preselected from the programming menu.
	Room temperature regulator disabled (dew-point operation) is active.
	A Shift of basic set value is active. In this case, the set value has been shifted in normal operation by one of the display buttons.
Prg	The programming menu is active.
	A push button disabling function has been activated via the bus. This symbol will also appear even though regulator operation is disabled.
	The room temperature timer, timer 1 or timer 2 are activated.
	The room is being heated or a heating set value temperature is shown in the programming mode.
	The room is being cooled or a cooling set value temperature is shown in the programming mode.
	To change between the individual main menu items and, if necessary, between the selected submenus by means of the right display button.
	Display of the stage of an integrated fan coil actuator. The display proceeds in four stages: Stage 1 shows one blade, stage 2 shows two blades, and stage 3 shows three blades. When the fan coil actuator is off this symbol will not be shown.

2.3 Text display (alarm text)

The B.IQ push button RTR can display a text message received via the bus and having a maximum length of 14 characters, according to KNX data point type 16.000. Such text can, for example, be the alarm or status message from an alarm centre.
Once a character string is received via object 73, *"read text/alarm message"*, the push button module will show in the bottom line of the display the text contained therein. Any other display elements, also the standard display (cf. "2.2 Display data in normal operation") or the key help display (cf. "3.4 Push button assistance function") will be overwritten by a new text display. Any operation on the push button module (e. g. long key actuation for a dimming process, or programming menu) will also be cancelled in a defined way upon the receipt of a new text message. The reception of a text message will not affect the functions of the room temperature regulator.

The text received last will always be displayed. Only after the text reading has been reset, the display will return to standard. The *"reset text message"* parameter in the *"B.IQ push button xgang"* parameter branch defines how an alarm text message can be reset…

"Via key actuation" setting:
The alarm text message can be acknowledged and reset by pressing any key. This will not trigger the function of the key actuated. Pressing the same key again will only activate its function.

"Via object" setting:
An alarm text message can solely be reset via separate communication object 74, "reset text/alarm message". The polarity of this object can be parameterized in the ETS plug-in. A key actuation will be evaluated in the normal way, i. e. the corresponding key function will be executed together with its assistance display. You can also call the programming menu. However, when you exit the programming menu or hide the push button assistance texts the alarm text will be shown again as long as the same has not been reset through the object yet.

"Via key actuation and object" setting:
An alarm text message can be reset either by pressing any key on the push button module, or via separate communication object 74, *"reset text/alarm message"*. Any key actuation will not trigger the function of the key actuated. Pressing the same key again will only activate its function.

In addition to the alarm text message received via the object, the B.IQ push button with room thermostat (RTR) and display can still display some other text messages predefined in the ETS plug-in (cf. "2.4 Text display via 3 x 1 bit"). You can use the *"text display priority"* parameter in the *"display"* parameter branch to set the display priority when using predefined text messages. When being called, a predefined text message can thus overwrite an alarm text message. However, this will not reset the alarm text message. As described, resetting will only be possible by a key actuation or through object 74, *"reset text/alarm message"* – even though a higher-priority predefined text message is being displayed.

The display of an alarm text will get lost when the push button module is reset (bus voltage failure, pulling off the bus coupling unit). After initialization of the push button module, some text will only be displayed if a new text string has been received via object 73, *"read text/alarm message"*.

2.4 Text display via 3 x 1 bit

The B.IQ push button with room thermostat (RTR) and display offers the option of calling up to three different text messages via three separate 1-bit communication objects. The texts themselves are predefined in the ETS plug-in and can have a maximum length of 20 characters.
To use this option the text display function in the *"display"* parameter branch must be enabled by the "text display" parameter (setting *"yes"*). When this function is enabled the associated parameters and communication objects will become visible.
Each predefined text message (*"text 1"*, *"text 2"*, *"text 2"* parameter) is called by its associated communication object *"text 1 display"*, *"text 2 display"* or *"text 3 display"*. The polarity of the call telegrams can be parameterized. Once such a call telegram is received, the push button module will show the predefined text in the bottom line of the display. Any other display elements, also the standard display (cf. "2.2 Display data in normal operation") or the push button assistance display (cf. "3.4 Push button assistance function") will be overwritten by a new text display. Any operation on the push button module (e. g. long key actuation for a dimming process, or programming menu) will also be cancelled in a defined way upon the receipt of a text message. The reception of a text message will not affect the functions of the room temperature regulator.
The text received last will always be displayed.

Only after the text reading has been reset, the display will return to standard. The *"reset text display"* parameter in the *"display"* parameter branch defines how the predefined text messages can be reset…

"Via separate object" setting:
In addition to the three objects for calling the predefined text messages, another 1-bit communication object for resetting all predefined text messages is available. Once a telegram that conforms with the parameterized polarity is received via object 78, *"text 1-3 display/reset display"*, the push button module will hide all predefined text displays.

"Via display objects" setting:
The predefined texts will be called separately via correspondingly associated objects *"text 1 display"*, *"text 2 display"* or *"text 3 display"* . Resetting proceeds through the same objects with inverted polarity. Hence, the display of predefined texts 1-3 will only be entirely hidden when all three calling objects have the corresponding polarity value for resetting. The polarity can, in general, be parameterized for all calling objects under the *"display object polarity"* parameter.

"Time-controlled" setting:
The predefined display texts will be reset altogether once the *"display period"* parameterized in the ETS plug-in has elapsed. This period will be restarted with each call of a predefined text.

A key actuation performed while a predefined text message is being displayed will be evaluated in the normal way, i. e. the corresponding key function will be executed together with its assistance display. You can also call the programming menu. However, when you exit the programming menu or hide the push button assistance texts the text message will be shown again as long as the same has not been reset yet.

In addition to calling the predefined texts, the B.IQ push button with room thermostat (RTR) and display can display an alarm text message (cf. "2.3 Text display (alarm text)"). The display priority for the additional use of the alarm text display can be set by the *"text display priority"* parameter in the *"display"* parameter branch. When being called, a predefined text message can thus overwrite an alarm text message, or vice versa. However, overwriting will not be reset by such overwriting. Once a higher-priority text display is reset, i. e. hidden, the push button module will show the lowest-priority message if the same is still active.

The display of the predefined texts will get lost when the push button module is reset (bus voltage failure, pulling off the bus coupling unit). After initialization of the push button module, a text will only be displayed when it is called again through objects *"text 1 display"*, *"text 2 display"* or *"text 3 display"* .

2.5 Data displayed during a programming process and during 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 "**Download parameters**". In this state, the device is being programmed with the project data. During this procedure, all push button and room temperature regulator 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 "**Firmware download loading.....**" shown on the display while programming is taking place. A firmware download can take several minutes.

After one programming process has been successfully completed, or after the push button has been re-plugged onto the bus coupling unit, the display will show for a short time the push button variant and the software version loaded.
Reading "**Berker B.IQ 5fach FW: 2.0 BL: 1.02**", for example, indicates a 5gang B.IQ push button with room thermostat (RTR) and display with firmware version 1.0 loaded and with a version 1.02 bootloader.

After this, the device will initialize. In this state, the display will read "**Initializing**". Then the device (both the push button and the room temperature regulator) will be ready for operation.

As the bus coupling unit and the B.IQ push button with room thermostat (RTR) and display form one unit and cannot be changed arbitrarily after having been programmed, the push button 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 button.
If the parameterizations do not match as the unit was not previously placed into operation as a whole, or as the push button or the bus coupling unit has been changed, the display of the B.IQ push button with room thermostat (RTR) and display will read "**No parameters**" to indicate that it has no valid parameters. In such case, the push button will not respond to any button actuation.

If the display reads "**Error**" the B.IQ push button with room thermostat (RTR) and display will be inoperable. In this case, the push button will not show any response and must be replaced.
If the display reads "**Programming-Mode**" the push button has been plugged onto a flush-mounted bus coupling unit which cannot be used. In such case, replace the bus coupling unit by a suitable UP BCU 1 (Berker ordering no.: 75040003).

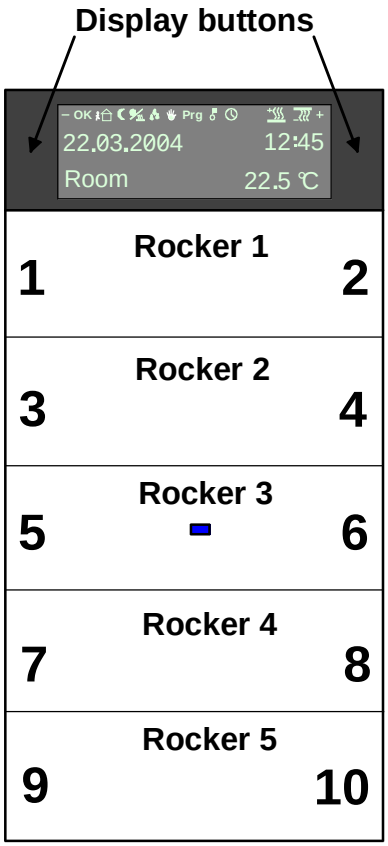
If the display shows "---" while it is reading the software version data and, in addition, if the hourglass symbol appears, there is no valid or runable firmware in the device. Such condition can, for example, be caused if a previously made firmware download was too erroneous or was interrupted for any other reasons. In such case, the B.IQ push button with room thermostat (RTR) and display 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 button functions

3.1 Rocker arrangements

Depending on the variant configured, the B.IQ push button with room thermostat (RTR) and display has up to 5 or 10 rockers, to which certain functions have been firmly assigned, or to which parameterizable functions can be assigned in the ETS plug-in.

For example, 5gang type



Depending on the application (3gang, 4gang or 5gang type) inserted into the ETS project, the required number of rockers or buttons, respectively, will be created automatically in the ETS plug-in.

To make the configuration of the push button functions more clearly arranged 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 buttons to go to the corresponding parameter branch and thus parameterize the selected rocker or button, respectively.

Within the corresponding parameter branch, you can assign a name to a rocker or a button. Such name just serves for better orientation in the plug-in and will neither be displayed in the preview window nor be downloaded into the device.

3.2 Functions of the rockers

The push button, room temperature regulator, controller extension, scene functions and timer functions have to be considered separately.
For the function as push button, up to 5 rockers, to which various functions can have been assigned, are separately available, depending on the variant configured.

The two display buttons next to the display are always reserved for local operation (basic set point shift / programming mode). The exact functionality of this rocker is discussed in more detail in the operator level switch-over description (refer to "1.4 Programming mode/local operation"). The display buttons have no status-LEDs.

Parameterizable in the ETS plug-in, the rockers can be given the following push button functions: In this connection, distinction is made between rocker and button actuation.

Function	Rocker Actuation	Button Actuation
No function	✓	✓
Switching/touch control	---	✓
Switching	✓	---
Dimming	✓	✓
Shutter	✓	✓
Light scene extension/recall	---	✓
value transmitter 1 byte	---	✓
value transmitter 2-byte	---	✓
Two telegrams	✓	✓
Betriebsmodus-Umschaltung *	✓	✓
Operating mode switch-over *	---	✓
Room temperature timer operation	---	✓
Timer operation	---	✓
Controller extension **	✓	✓

*: The "operating mode switch-over" and "setpoint shifting" functions actuate the internal room temperature regulator. The exact function of a rocker parameterized to these functions is found in the description of the room temperature regulator functions (chapter 4.).

**: The "controller extension" function can be parameterized with enabled controller extension functions. The exact function of a rocker parameterized to these functions is found in the description of the controller extension functions (chapter 5).

In general, the push button function can be enabled. For this purpose, set the *"push button function"* parameter in the *"B.IQ RTR push button"* parameter branch in the ETS plug-in to *"enabled"*. If the push button function is *"disabled"* the selection of the push button functions will be matched so that only the functions of the enabled function elements (*"room temperature regulator operation", "room temperature timer operation", "timer operation" and/or "light scene extension recall"*) will be parameterizable in this case.

Through the *"concept of operation"* parameter in the *"push button functions / general"* parameter branch, select the type of rocker actuation. Separately for each rocker, you can parameterize rocker actuation or button actuation. For rocker actuation, the left and right buttons of a rocker form a pair of buttons to which you can assign a joint function. For button actuation, the left and right buttons 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 button functions listed in the table to the different rockers or buttons, 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 push button"* parameter in the *"push button function / general / [button description]"* parameter branch to set the function to be executed upon the pressing of a button.

3.2.1 Rocker actuation

3.2.1.1 "No function" function

If the rocker's *"rocker function"* parameter is parameterized to *"no function"*, the affected actuation keys and thus the associated objects are deactivated. Only the status-LED's can be controlled via the status object. The control is predefined by the *"function of the left/right status-LED"* parameter in the *"push button function / general / [key designation] / status of rocker X"* parameter branch.

3.2.1.2 "Switching" function

If the rocker's function is parameterized to *"switching"* the "command on pressing a rocker" parameter and the "*switching*" object will become visible. The *"*command on pressing a rocker*"* parameter determines the switching commands which are to be output on the bus when pressing the left and the right key. Executable switching commands may include *"ON"*, *"OFF"* or *"TOGGLE"*. The value, which is stored in the switching object, is switched and transmitted when executing the *"TOGGLE"* command. The commands are to be preset and preselected in combination only for the left and right key. In addition, the transmission of a switching command may be suppressed with one key-press (setting *"---"*).

The two status-LED's of the rocker (left and right) can be parameterized and controlled irrespective of the rocker function. The *"function of left / right status-LED"* parameter in the *"push button function / general / [key designation] / status of rocker X"* parameter branch determines whether the LED is permanently switched on or off or whether it is controlled via the "rocker X *[status left / right]*" object. In the latter case the polarity of the status objects is definable.

3.2.1.3 "Dimming" function

If the function of the rocker is parameterized to "dimming", several parameters for the dimming function as well as the *"switching"* and *"dimming" objects* will become visible. The *"*command on pressing a rocker*"* parameter determines the switching or dimming commands, which are to be output on the bus when pressing the left and the right key. Executable switching commands may include *"darker (OFF)"*, *"Brighter (ON)"* or *"TOGGLE"*.
With the *"darker (OFF)"* command a short-time key-press will trigger an OFF telegram; a long key-press will trigger a dimming telegram (darker). With the "brighter *(ON)*" command a short key-press will trigger an ON telegram; a long key-press will trigger a dimming telegram (brighter). With the *"TOGGLE"* command a short key-press will switch-over the switching state which is internally saved in the switching object. If the saved state is ON (OFF), an OFF (ON) telegram will be triggered. A long key-press will transmit a "brighter" telegram followed by a "darker" telegram and vice versa.
The commands are to be preset and preselected in combination for the left and right key.

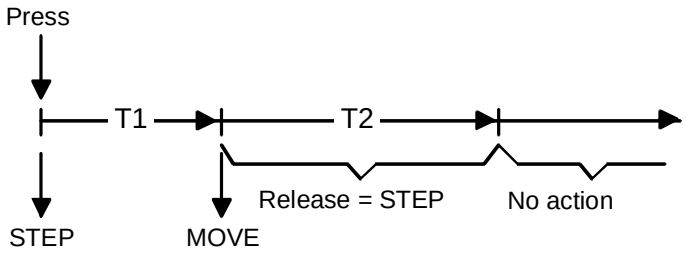
In addition, the dimming increments can be adjusted for "increasing and decreasing brightness*"* and the *"time between switching and dimming ".* A *"stop telegram"* can be enabled at the end of the dimming operation (telegram transmission on releasing the push button). If the *"telegram repetition"* parameter is set to *"YES"*, dimming telegrams can be periodically transmitted during a key-press. The *"time between two dimming telegrams"* can be adjusted. In each case this time has elapsed, a new dimming telegram will be issued with the parameterized dimming increment.

The two status-LED's of the rocker (left and right) can be parameterized and controlled irrespectively of the rocker function. The *"function of left / right status-LED"* parameter in the "push button function / general / [key designation] / status of rocker X" parameter branch determines whether the LED is permanently switched on or off or whether it is controlled via the "*rocker X [status left / right]*" object. In the latter case the polarity of the status objects is definable.

3.2.1.4 "Shutter" function

If the rocker function is parameterized to "*shutter*", several parameters for the shutter function as well as the "*STEP operation*" and "*MOVE operation*" objects will be visible. The "*operation concept*" parameter predefines the telegram sequence for short and long-time telegrams that are transmitted with or during key actuation.

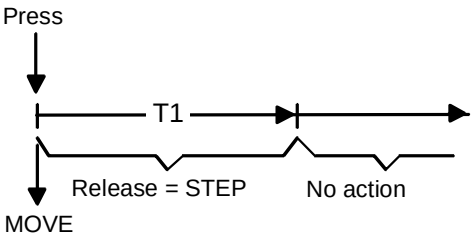
- *"Step - move - step"*:



Pressing a rocker's key will transmit a step command and time T1 is started (*"time between step and move operation"*) No other telegram is transmitted, if the key is released within T1. This step serves to stop a continuous run in progress.

If the key is pressed for more than T1, a MOVE command is automatically transmitted after T1 has elapsed and time T2 is started (*"Lamella adjustment time"*) If the key is released again within T2, the push button will transmit a step-telegram. This function is used to adjust the Lamellas. T2 should correspond to the time required for a 180° rotation.

- *"Move - step":*



Pressing a rocker's key will transmit a Move command and time T1 is started ("*Lamella adjustment time*") If the key is released again within T1, the push button will transmit a step-telegram. This function is used to adjust the Lamellas. T1 should correspond to the time required for a 180° rotation.

The *"command on pressing the rocker"* parameter determines the polarity of the telegrams for long or short-time operation, i.e. their activated direction depending on the actuated (left or right) key. Executable commands may include "UP", "DOWN" or "TOGGLE". The commands are to be preset and preselected in combination only for the left and right key.

The two status-LED's of the rocker (left and right) can be parameterized and controlled irrespectively of the rocker function. The *"function of left / right status-LED"* parameter in the "push button function / general / [key designation] / status of rocker X" parameter branch determines whether the LED is permanently switched on or off or whether it is controlled via the "*rocker X [status left / right]*" object. In the latter case the polarity of the status objects is definable.

3.2.1.5 "Two telegrams" function

This function provides the option of sending to the bus two different telegrams via different communisation objects upon only one key actuation. For both channels, you can use parameters *"type of 1st object"* and *"type of 2nd object"* parameters to define what communication object types should be used. You can choose between…

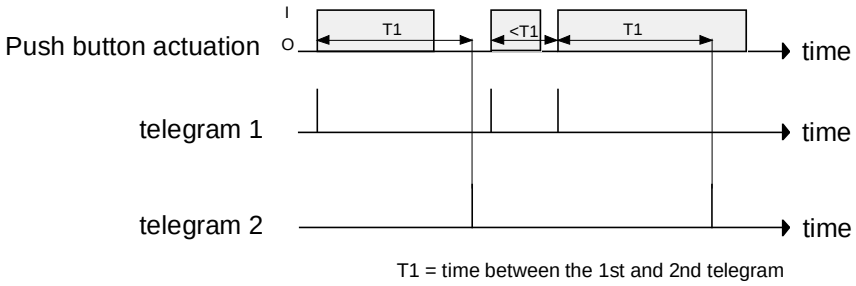
- switching (1-bit – DPT-ID 1.001)
- 0 … 255 value transmitter (1-byte – DPT-ID 5.001, 5.003, 5.004, 5.010)

In this way, you can, for example switch on or off different groups of lamps or set them to a brightness value without having to use scenes.
The switching commands (ON, OFF, TOGGLE, no function) or value commands (0…255) to be executed can be parameterized for each key of a rocker in a different way.

This push button function distinguishes between two operating concepts. The *"operating concept"* parameter defines the key evaluation and specifies the telegram transmission. Upon one key actuation, you can send to the bus either *"always two telegrams"* or alternatively *"either telegram 1 or 2"*.

"Always two telegrams" operating concept:

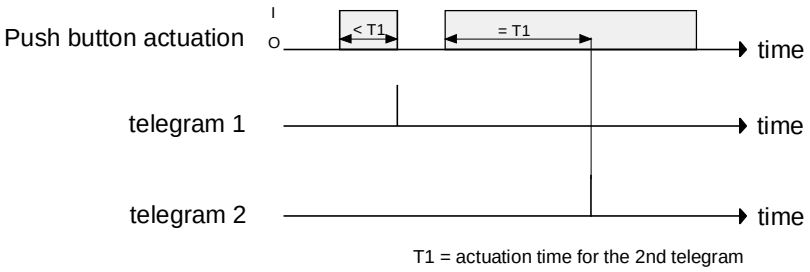
The first telegram is always sent at the same time as the key is being actuated. The presetting effects the sending of the second telegram after a certain delay time started upon the key actuation has elapsed. Thus, you need not hold the key pressed to send the second telegram. The delay time is activated by the *"delay between the 1st and 2nd telegram"* parameter (*"yes"*) and defined by the *"time between the 1st and 2nd telegram"* parameter. You can parameter periods between one second and 30 minutes. The delay time can be re-triggered by actuating the key several times.
Alternatively, you can deactivate the delay time so that the two telegrams will be sent to the bus one by one upon one key actuation. In this case, you cannot predetermine the time sequence of the telegrams. Releasing the key will show no response.



T1 = time between the 1st and 2nd telegram

"Either telegram 1 or 2" operating concept:

For this operating concept, always only one of the two parameterized telegrams will be set to the bus upon a key actuation. The time how long the key is being actuated determines which of the two telegrams will be sent.
You can use the *"actuation period for the 2nd telegram"* parameter to define the time period for distinguishing between short-time and long-time key actuation. If you actuate the key for a time shorter that the parameterized period only the first telegram will be sent when you release the key. If you exceed the parameterized actuation period only the second telegram will be transmitted. You can parameter periods between 0.5 and 60 seconds



T1 = actuation time for the 2nd telegram

The two status LEDs of the rocker (left and right) can be parameterized and triggered independently of the rocker function. The *"function of the left/right status-LED"* parameter in the *"push button function operation/[key designation]/status of rocker X"* parameter branch defines whether the LED will be permanently ON or OFF or will be triggered by the *"rocker X [status left/right]"* object. In the latter case, the polarity of the status objects can be parameterized.

3.2.2 Key actuation

3.2.2.1 "No function" function

The key is deactivated, if the "*key function"* parameter is set to "*no function*". Only the status-LED can be used. The control is defined by the "*function of the status-LED*" parameter in the "*push button function / general / [key designation]*" parameter branch. It is possible to permanently switch on or off the LED or to switch the LED via a separate communication object. In the latter case the polarity of the status objects in the *"push button function / general / [key designation] /state of push button"* parameter branch can be parameterized.

3.2.2.2 "Switching/pushing" function

If the key's function is parameterized to "*switching / pushing*", the *"command on pressing the push button" and "command on releasing the push button"* parameters and the "switching" object will be visible.
The "command on pressing a rocker/releasing the key" parameters determine the switching commands which are to be output on the bus by releasing the key. With these two parameters, which are independent of each other, it is also possible to execute a momentary contact function (e.g. pressing = on, releasing = off).
Executable switching commands may include "ON", "OFF" or "TOGGLE". *"TOGGLE"* will switch and transmit the value, which is stored in the switching object. In addition, the transmission of a switching command can be suppressed with one key-press (setting "no function").

The status-LED function is defined by the "*function of the status-LED*" parameter in the "*push button function / general / [key designation]* " parameter branch. It is possible to permanently switch on or off the LED, to signal a key actuation, to indicate the status of the switching object or to switch the LED via a separate communication object. In the latter case the polarity of the status objects in the "*push button function / general / [key designation] /state of push button*" parameter branch can be parameterized.

3.2.2.3 "Dimming" function

If the function of the key is parameterized to "*dimming*" (one-touch dimming), several parameters for the dimming function as well as the *"switching"* and *"dimming"* objects will be visible. The "*command on pressing the push button, push button function*" parameter determines the switching or dimming commands, which shall be output on the bus when pressing a key. Executable commands may include *"darker (OFF)"*, *"brighter (OFF)"* or "brighter / darker (*TOGGLE)*".
With the *"darker (OFF)"* command a short key-press will trigger an OFF telegram; a long key-press will trigger a dimming telegram (darker). With the *"brighter (ON)"* command a short key-press will trigger an ON telegram; a long key-press will trigger a dimming telegram (brighter). With a "brighter / darker (*TOGGLE)*" command a short key-press will switch-over the switching state that is internally stored in the switching object. If the saved state is ON (OFF), an OFF- (ON) telegram will be triggered. A long key-press will transmit a "darker" telegram followed by a "brighter" telegram and vice versa.

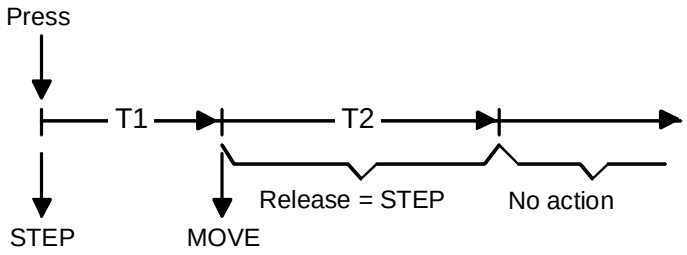
In addition, the dimming increments can be adjusted for "*increasing and decreasing brightness*" and the "*time between switching and dimming* ". A "*stop telegram*" can also be released at the end of the dimming operation (telegram transmission on releasing the push button). If the "*telegram repetition*" parameter is set to "YES", dimming telegrams can be periodically transmitted during a key-press. The "*time between two dimming telegrams*" is adjustable. In each case this time has elapsed, a new dimming telegram is issued with the parameterized dimming increment.

The status-LED function is defined by the "*function of the status-LED*" parameter in the "*push button function / general / [key designation]* " parameter branch. It is possible to permanently switch on or off the LED, to signal a key actuation, to indicate the status of the switching object or to switch the LED via a separate communication object. In the latter case the polarity of the status objects in the "*push button function / general / [key designation] /state of push button*" parameter branch can be parameterized.

3.2.2.4 "Shutter" function

Different parameters for the shutter function and the *"STEP operation"* und *"MOVE operation" objects* will be visible when parameterizing the *"shutter" function*. The "operation concept" parameter (telegram sequence) predefines the telegram sequence for short and long telegrams that are transmitted when actuating a key.

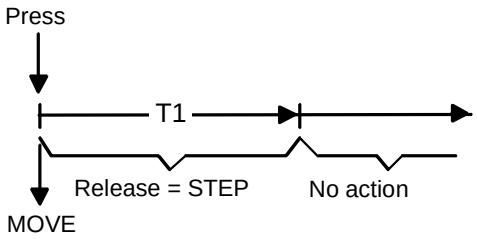
- *"Step - move - step":*



Pressing a key will transmit a step command and time T1 is started ("*time between step and move operation*") No other telegram will be transmitted, if the key is released within T1. This step serves to stop a continuous run in progress.

If the key is pressed for more than T1, a MOVE command is automatically transmitted after T1 has elapsed while time T2 ("Lamella adjustment time") is started. If the key is released again within T2, the push button will transmit a step-telegram. This function is used to adjust the Lamellas. T2 should correspond to the time required for a 180° rotation .

- *"Move - step":*



Pressing a key will transmit a Move command and time T1 is started ("Lamella adjustment time"). If the key is released again within T1, the push button will transmit a step-telegram. This function is used to adjust the Lamellas. T1 should correspond to the time required for a 180° rotation.

The "*function of shutter push button*" parameter determines the polarity of the telegrams for long or short-time operation, i.e. their direction when the key is actuated. Executable commands may include "UP", "DOWN" or "TOGGLE".

The status-LED function is defined by the "*function of the status-LED*" parameter in the "*push button function / general / [key designation]* " parameter branch. It is possible to permanently switch on or off the LED, to signal a key actuation or to switch the LED via a separate communication object. In the latter case the polarity of the status object in the "*push button function / general / [key designation] /state of push button*" parameter branch can be parameterized.

3.2.2.5 "Value transmitter 1-byte" and "value transmitter 2-byte" function

When parameterized as a 1-byte value transmitter (e.g. for dimming value transmitter applications), the push button will transmit an 8-bit value on the bus when pressing a key-press. The value to be transmitted is parameterized in the ETS plug-in and can lie within the value range of 0 to 255.

When parameterized as a 2-byte value transmitter, 2-byte values can be transmitted on the bus. The *"function as"* parameter determines whether the value is a temperature value, a brightness value or a dimensionless 2-byte reading.

The range of values of the parameterizable temperature value lies between 0 °C and 40 °C in 1 °C increments. The brightness value may lie between 0 lux and 1500 lux in 50 lux increments. In case that brightness values are parameterized which do not correspond to the 50-lux increments, the plug-in will automatically adjust the value by rounding it up or down. The range of values possible for the 2-byte value transmitter lies between 0 and 65535.

Value readjustment:

When parameterizing a value transmitter is parameterized it is possible to readjust the value to be transmitted via a long key-press (> 5 sec). This will lower the predefined value by the parameterized increment before transmitting it. The previously transmitted value is stored after releasing the key. The next long key-press will change the direction of the value readjustment.

The adjustment increments for the 1-byte or 2-byte value transmitter are parameterizable. The increments for the temperature value transmitter are set to 1 °C while the increments for the brightness value transmitter are set to 50 lux.

The status-LED function is defined by the "function of the status-LED" parameter in the "*push button function / general / [key designation]* " parameter branch. It is possible to permanently switch on or off the LED, to signal a key actuation or to switch the LED via a separate communication object. In the latter case the polarity of the status object in the "*push button function / general / [key designation] /state of push button*" parameter branch can be parameterized.

The status-LED of the actuated key and the one of the opposite key will blink during a value readjustment with approx. 3 Hz no matter what functions the LED is parameterized for (cf. example on the next page). No other key may be actuated during a value readjustment!

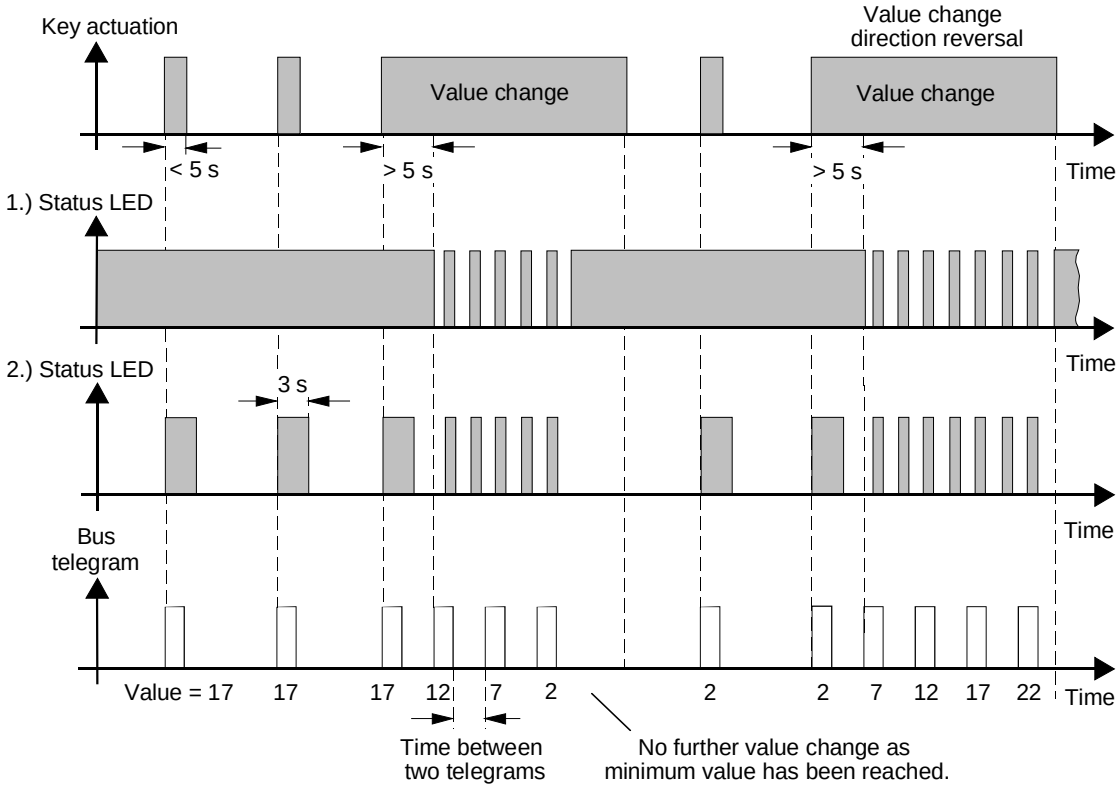
The following demonstrates a value readjustment considering as example a 1-byte value transmitter:

1.Function of the status-LED	*always ON*
Value (0...255)	*17*
Increments (1...10)	*5*

⇨ The status-LED is always on. The status-LED will blink during the value readjustment.

2.Function of the status-LED	*Actuation indicator*
Light-emitting period of actuation indicator	*3 sec*
Value (0...255)	*17*
Increments (1...10)	*5*

⇨ The status-LED lights up for the duration of the parameterized time when pressing a key. The status-LED will blink during the value readjustment.



Notes:

- Depending on the parameterized increments it might be possible that the value "0" never reaches „0". This will ensure that the value, which was originally parameterized by the ETS, can be reset with a new value readjustment.
- The newly adjusted values are only stored in RAM. Thus, following a bus voltage failure or a plugging/unplugging of the user module, these values will be replaced by the preset values, which were originally programmed via the ETS.

3.2.2.6 "Light-scene extension/recall" function

This function determines whether an ‘external’ light-scene is to be addressed via the light-scene extension object or via one of the push button’s internal scenes. The *"function as"* parameter determines the mode of action.

If the key function is parameterized to *"light-scene extension", the "light-scene extension"* will be enabled. Light-scenes, which are stored in another bus device that features a light-scene function, may be recalled via this object via a short key-press (< 1 sec). The light-scene number parameterized in the ETS plug-in will be transmitted (1 to 64).
If the function is parameterized to *"internal scene request"*, a short key-press (< 1 sec) will recall the scene stored in the push button RTR. In doing so, the corresponding scene number (1 to 8) must be predetermined in the ETS plug-in. An extension object is not required for this function. In addition, an internal scene can only be recalled with enabled scene function!
The status-LED of the actuated key lights up for the duration of the parameterized time.

The "*memory function* " parameter can be used to determine whether the ‘external’ light-scenes or the ‘internal’ scenes are only to be recalled or also to be stored following a long key-press (> 5 sec).

It is possible to create a memory telegram depending on the parameterized light-scene number with a parameterization as *"light-scene extension"* with memory function. A long key-press > 5 sec will transmit the corresponding memory telegram.

A parameterization as *"internal scene request"* with memory function allows to store an internal scene acc. to the parameterized scene number via a long key-press > 5 sec. The scene control of the push button RTR requests the current values of the scene objects from the actuators via the bus and stores them in non-volatile memory.
Attention should be paid to the fact that these read flags ("R" flags) are set with the corresponding actuator objects!

The status-LED function is defined by the "*function of the status-LED*" parameter in the "*push button function / general / [key designation]* " parameter branch. It is possible to permanently switch on or off the LED, to signal a key actuation or to switch the LED via a separate communication object. In the latter case the polarity of the status object in the "*push button function / general / [key designation] /state of push button*" parameter branch can be parameterized.

While information is actively stored the status-LED of the actuated key and the one of the opposite key will blink (approx. 3 Hz). No other key may be pressed during this time!
A short key-press < 1 sec will simply recall the parameterized light-scene. If the key is pressed for more than 1 sec but no longer than 5 sec, neither a recall nor a saving procedure will be triggered.
The status-LED of the actuated key lights up for the duration of the parameterized time.

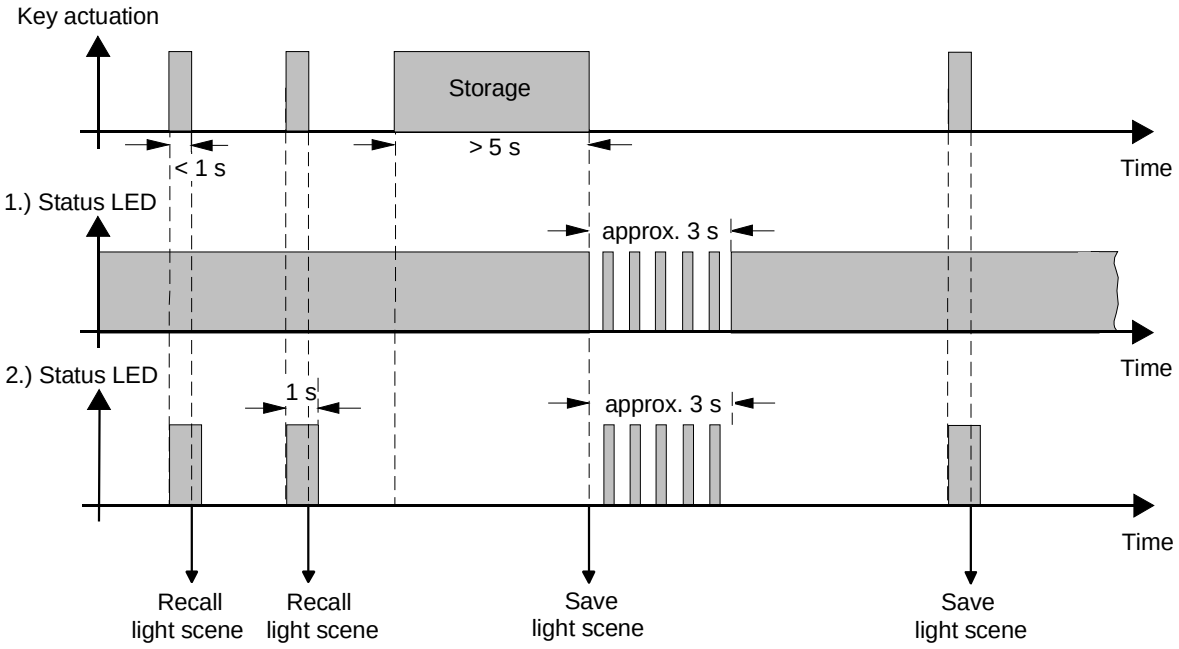
Examples for the memory function:

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

⇨ The status-LED is always on. During saving it will blink for approx. 3 sec.

2.Function of the status-LED *Actuation indicator*
Light-emitting period of actuation indicator *1 sec*

⇨ The status-LED lights up for the duration of the parameterized time when pressing a key. During saving it will blink for approx. 3 sec.



3.2.2.7 "Room temperature timer operation" or "timer operation" function

Once the room temperature timer and/or one of the timers has/have been enabled in the ETS plug-in you can additionally set the following two operating functions for the button functions.

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

The function of the status-LED can be parameterized. That way the *"display timer active"* and *"display timer inactive"* settings can be selected in addition to the *"off"*, *"on"* and *"operation indication"* standard settings. In this manner the status-LED's can signal whether the function linked with an associated key is activated or deactivated. This signaling will take place even if the corresponding function has been locally activated or deactivated in programming mode.
Alternatively, the LED can be switched via a separate communication object. The polarity of the status object in the "*push button function / general / [key designation] /state of push button*" parameter branch can be parameterized.

It should be noted that the operating function of the room temperature timer and of the timer(s) in the programming mode will always be possible, independently of any function parameterized for the buttons.

3.2.2.8 "Two telegrams" function

This function provides the option of sending to the bus two different telegrams via different communisation objects upon only one key actuation. For both channels, you can use parameters *"type of 1st object" and "type of 2nd object"* parameters to define what communication object types should be used. You can choose between…

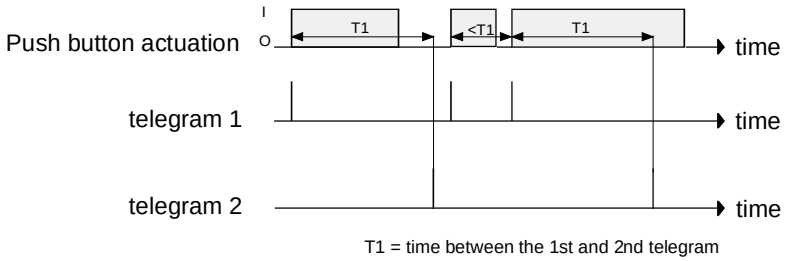
- switching (1 bit – DPT-ID 1.001)
- 0 … 255 value transmitter (1-byte – DPT-ID 5.001, 5.003, 5.004, 5.010)

In this way, you can, for example switch on or off different groups of lamps or set them to a brightness value without having to use scenes.
The switching commands (ON, OFF, TOGGLE, no function) or value commands (0…255) to be executed can be parameterized.

This push button function distinguishes between two operating concepts. The *"operating concept"* parameter defines the key evaluation and specifies the telegram transmission. Upon one key actuation, you can send to the bus either *"always two telegrams"* or alternatively *"either telegram 1 or 2"*.

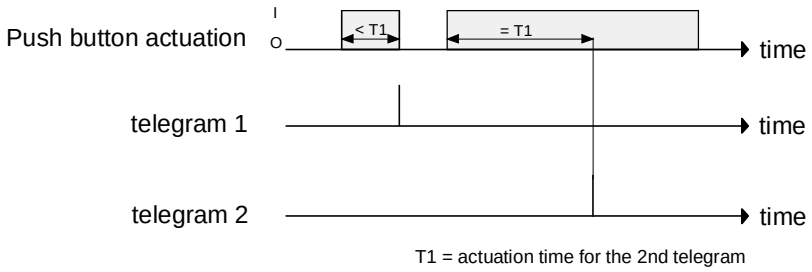
"Always two telegrams" operating concept:

The first telegram is always sent at the same time as the key is being actuated. The presetting effects the sending of the second telegram after a certain delay time started upon the key actuation has elapsed. Thus, you need not hold the key pressed to send the second telegram. The delay time is activated by the *"delay between the 1st and 2nd telegram"* parameter (*"yes"*) and defined by the *"time between the 1st and 2nd telegram"* parameter. You can parameter periods between one second and 30 minutes. The delay time can be re-triggered by actuating the key several times.
Alternatively, you can deactivate the delay time so that the two telegrams will be sent to the bus one by one upon one key actuation. In this case, you cannot predetermine the time sequence of the telegrams. Releasing the key will show no response.



"Either telegram 1 or 2" operating concept:

For this operating concept, always only one of the two parameterized telegrams will be set to the bus upon a key actuation. The time how long the key is being actuated determines which of the two telegrams will be sent.
You can use the *"actuation period for the 2nd telegram"* parameter to define the time period for distinguishing between short-time and long-time key actuation. If you actuate the key for a time shorter that the parameterized period only the first telegram will be sent when you release the key. If you exceed the parameterized actuation period only the second telegram will be transmitted. You can parameter periods between 0.5 and 60 seconds



The function of the status LED can be defined by the *"function of the status LED"* parameter in the *"pushbutton sensor function/operation/[key description]"* parameter branch. In this case, it is possible to permanently switch the LED ON or OFF, to signalize some key actuation, or to switch the LED via a separate communication object. In the latter case, the polarity of the status objects can be parameterized in the *"push button function/operation/[key designation]/state of key X"*.

Alternatively, you can configure the status LED so that is can indicate the status of the first object (*"status indication (switching object)"* or *"inverted status indication (switching object)"* settings). However, status indication will only be possible if the first object is a *"switching"* object. Status indication for the *"value"* object type is, in this case, not possible for the first object.

3.3 Push button disabling function

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

Settings:

"Single rocker disabled" → - Rockers 1-3 (3gang type), 1-4 (4gang type) or 1-5 (5gang type) can be separately disabled.
- In this connection the display buttons (room temperature regulator operation/programming menu) will always be in operation.

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

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

"Function of all rockers like…" → - All rockers of the device will behave in the same was as the one parameterized here. In this case, the button or rocker or button 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 'normal operation'.
- The display buttons are not affected by this disabling function and will show their 'normal' behaviour.

When the disabling function is active the " ". symbol will appear on the display.
In addition, you can locally disable the push buttons of the device from the programming menu, for example as children protection (refer to the "settings" menu). This type of disabling is independent of an disabling function initiated via the bus. If the push button disabling function has been activated in the programming mode the " " symbol will not appear on the display. In addition, the push button assistance function will be suppressed when the button disabling function is active.

Please note that the room temperature regulator operation can be additionally influenced by the regulator disabling function (refer to "4.6 Room temperature regulator disabling functions"). Thus, buttons or rockers which have been assigned to room temperature regulator operation must be disabled by the push button or regulator disabling function. When regulator operation has been disabled the " " symbol will also appear on the display.

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

3.4 Push button assistance function

If desired, you can activate a push button assistance function for the function keys (rockers 1 to 5 max.) of the push button sensor. This assistance function is to indicate to the push button operator what function will be executed when a key is actuated. This push button assistance function can be centrally enabled by the *"push button assistance function"* parameter in the "*B.IQ push button xgang"* parameter branch.

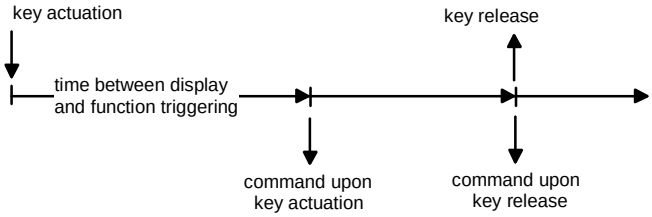
The push button assistance is a display text with a maximum length of 20 characters and can be independently parameterized in the ETS plug-in for each function key when the push button assistance function has been enabled. This text will be shown on the display of the bush button RTR, once a function key is pressed. In this connection, the stored key function will either be triggered in a time-controlled way after a definable actuation time has elapsed or, alternatively, when the same key is pressed for the second time, rather than being executed immediately. The *"push button assistance function"* parameter defines the following behaviour.
The functions of the two display keys next to the display are fixed so that any assistance for them will not be necessary and is thus not implemented.

Time-controlled push button assistance function:

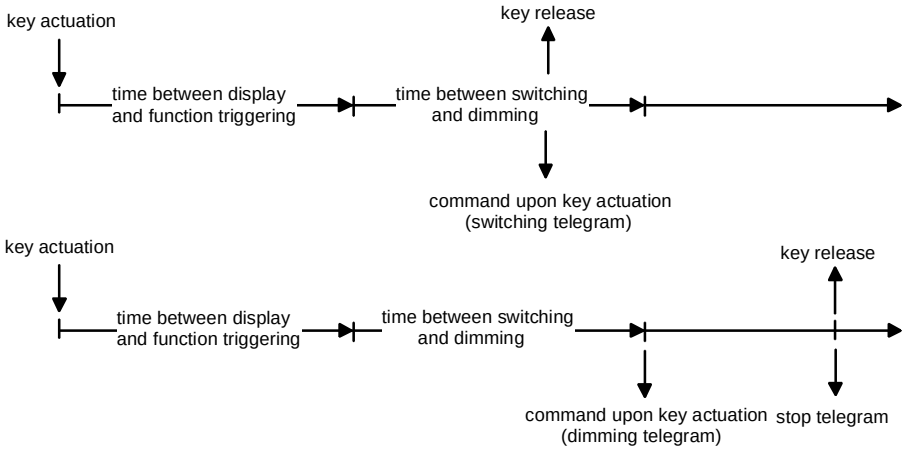
Upon a key actuation, the push button assistance text will be displayed, with the *"time between display and function triggering"* parameterized in the ETS plug-in being started. Within this time, the operator can decide whether to execute this function or not. Only if the key remains pressed for more than the parameterized time the push button module will fully execute the key function stored. After the key is released, the push button assistance text will remain on the display for about another 3 s until the display returns to normal.
A status LED parameterized as an actuation indicator will only shine after the *"time between display and function triggering"* has elapsed.
The default setting of the time between the display and the triggering of the function is "0.0" so that the push button assistance text will be displayed and the corresponding key function executed immediately upon the key actuation.

Time-controlled push button assistance examples:

"Switching" function:



"Dimming" function:

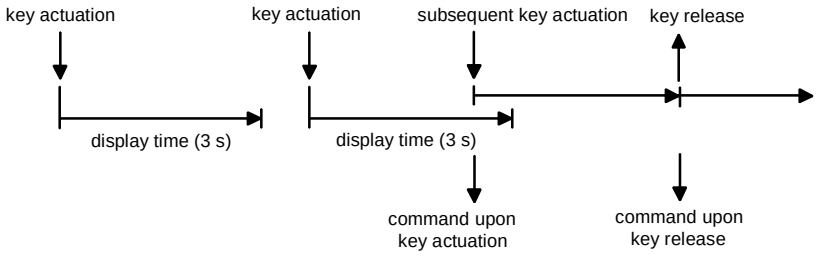


Key-controlled push button assistance function

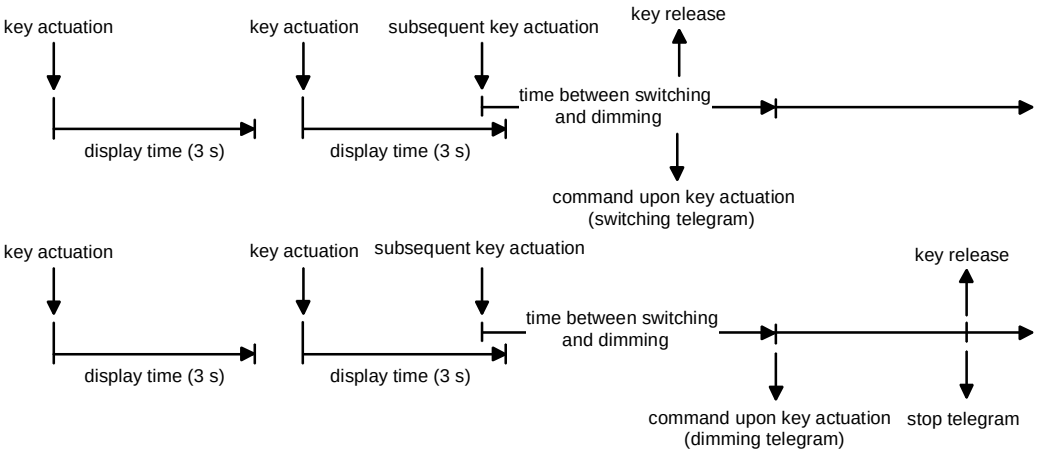
Upon a key actuation, the push button assistance text will be displayed. Irrespective of the time how long the key is being pressed, the push button module will not trigger the key function. After the key is released, the push button assistance text will remain on the display for about another 3 s until the display returns to normal.
Within this 3 s display time, the operator can decide whether to execute this function or not.
Only if the same key is pressed once more within the text display period the push button module will execute the stored key function as parameterized.
In this connection, the push button assistance text will be displayed once more and will again remain shown on the display for some 3 s after the key has been released. Within this time, the same key can be pressed again and again, with the key function again being immediately executed (this can, for example, be helpful for changing a shutter moving direction or a dimming process).
A status LED parameterized as an actuation indicator will only be lit upon a next key actuation, i. e. when the key function is being executed.

Key-controlled push button assistance examples:

"Switching" function:



"Dimming" function:



Information on the push button assistance function:

- The pus button assistance function is a useful means for beginners. If the push button assistance function is no longer necessary after some time of operation or is to be temporarily switched off it can be deactivated or subsequently re-activated from the programming menu (cf. "1.4.2.4 "Settings" submenu).
- Please note that the push button assistance function will be suppressed if a key is disabled.
- A push button assistance display can be overwritten, i. e. cancelled, by a text message (cf. "2.3 Text display (alarm text)" and "2.4 Text display via 3 x 1 bit").

Functional Description

4. Room temperature regulator functions

4.1 Operating modes

The room temperature regulator has various operating modes. The selection of these modes will, for example, facilitate the activation of different temperature set values, depending on the presence of a person, on the state 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, for this reason, 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 on 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 on the display by the " ⌂ " symbol.

- Night mode

  During the night hours or during the absence of persons for a longer time, 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 night mode.
  The activated night mode will be indicated on 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 prescribe some temperature set value 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 on the display by the " ❄ " symbol.

- Comfort mode prolongation (temporary comfort mode)

  You can activate the comfort prolongation option from the night 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 exclusively be activated by a parameterized presence button 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 button 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 on the display by the " ⌂☾ " or " ⌂❄ " symbol.

You can assign an own temperature set value to each *"heating"* or *"cooling"* operating mode (refer to "4.4 Temperature set values",).

Only one operating mode can be activated at a time so that both control circuits will always be in the same mode if two control circuits 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 button (programming mode), if enabled;

b) local operation on the push button (rockers 1 to 5 max.) and parameterized operating mode switch-over;

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

About a):
You can activate the programming mode (refer to "1.4 Programming mode/local operation") to select the "comfort", "standby", "night" or "frost/heat protection (absence)" operating modes from the "operating mode switch-over" menu.
Switch-over to comfort prolongation will not be possible in the programming mode.

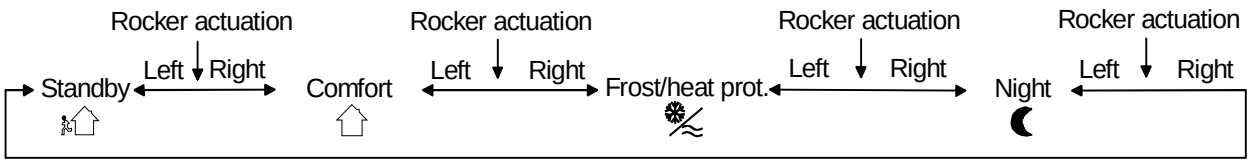
About b):
In addition to operation in the programming mode, you can parameterize the "operating mode switch-over" function to rockers 1 to max. 5 (depending on the variant configured) of the push button (refer to "3.2 Rocker functions"). In this connection, distinction is made between button and rocker functions:

- Button function:
The function of a button has been set to "operating mode switch-over". In this case, you can set in the ETS plug-in which operating mode you want to activate by pressing this button. For this purpose, the "comfort", "standby", "night" and "frost/heat protection" modes are available.

  To be able to activate the comfort prolongation option you can use the *"presence detection"* and the *"type of presence detection"* parameters in the *"room temperature regulator function / functionality"* parameter branch to additionally enable the presence button in the ETS plug-in. If enabled, the *"presence object"* will appear, and you can select the *"presence button"* setting from the button functions. In this way, you can actuate the presence button to change to the comfort prolongation option or to deactivate the latter prematurely when the night or frost/heat protection mode (not activated by the *"window status"* object) has been activated. Also, you can switch over from the standby to the comfort mode when you actuate the presence button.

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

- Rocker function:
The function of a rocker has been set to "operating mode switch-over". In such case, you can use the left or right button of the rocker to change the operating mode. The switch-over sequence will always be from the "comfort" to the "standby" and then to the "night" mode, followed by the "frost/heat protection" mode.



The activation of the comfort prolongation option (presence function) will not be possible for a rocker function.

Via the status object, you can trigger the status-LEDs of the rocker in the same way as for a push button rocker function, irrespective of the room temperature regulator 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 regulator 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 button (Table 1/Fig. 1) and presence detector (Table 2/Fig. 2 on the next page):

Table 1

"Operating mode switch over" objects				Window status	Presence button object	Activated operating mode
Obj. no. 31	Obj. no. 28	Obj. no. 29	Obj. no. 30	Obj. no. 34	Obj. no. 33	
X	X	X	X	1	X	Frost/heat protection
1	X	X	X	0	0	Frost/heat protection
0	1	X	X	0	0	Comfort
0	0	1	X	0	0	Standby
0	0	0	1	0	0	Night
1	X	X	X	0	1	Comfort 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:

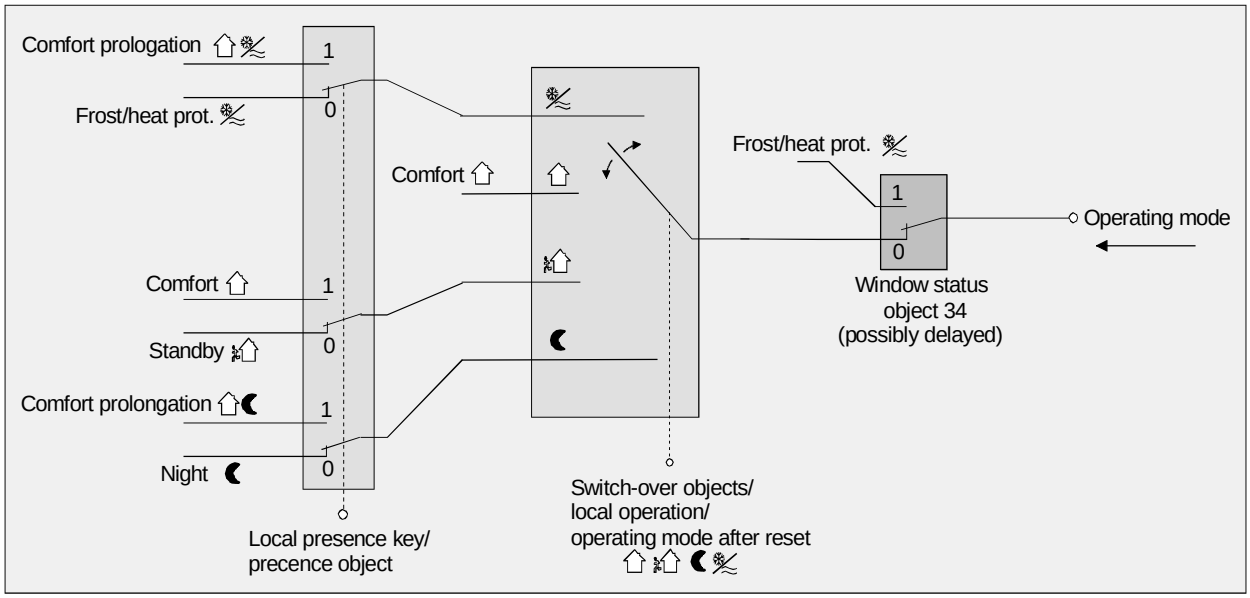
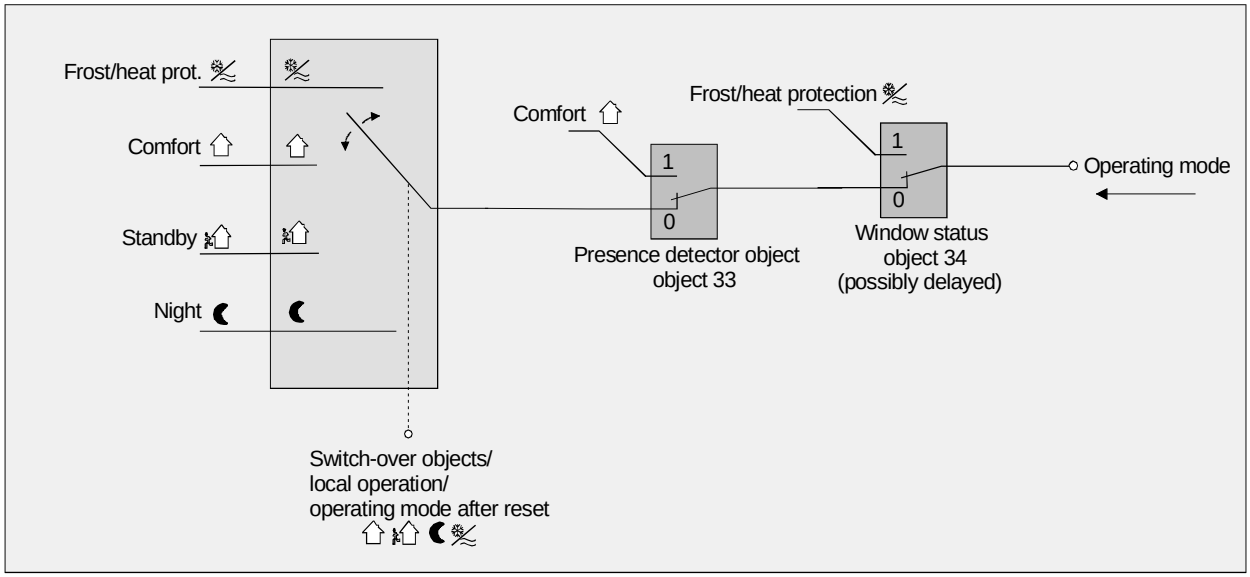


Table 2

"Operating mode switch over" objects				Window status	Presence detector object	Activated operating mode
Obj. no. 31	Obj. no. 28	Obj. no. 29	Obj. no. 30	Obj. no. 34	Obj. no. 33	
X	X	X	X	1	X	Frost/heat protection
X	X	X	X	0	1	Comfort
1	X	X	X	0	0	Frost/heat protection
0	1	X	X	0	0	Comfort
0	0	1	X	0	0	Standby
0	0	0	1	0	0	Night
0	0	0	0	0	0	Last 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 / night / 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 button, taking account of the priorities of the operating modes. If no higher-priority mode (e. g. window contact/presence detector) has been activated you can use the button or rocker function to switch over on the device an operating mode preset by an object.
- When a presence button 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 the comfort prolongation is stopped after the prolongation time has elapsed, or if the operating mode has been changed by a higher-priority operation through the switch-over objects or by local operation.
- If you use further B.IQ push button with room thermostat (RTR) and display as extensions to switch over the operating modes such switch-over should be solely effected by push buttons or rockers (push button functionality) which have been parameterized with the "switching" function. Otherwise (for example, for extension parameterization as "operating mode switch-over") the priority evaluation of the incoming telegrams can activate an undesired operating mode at the main unit (B.IQ push button with room thermostat (RTR) and display acting as room temperature regulator).
- Only one operating mode can be activated at a time so that both control circuits will always be in the same mode if two control circuits are used.
  The operating mode switch-over of the second control circuit always proceeds in parallel with the first control circuit.

• 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 button (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. 28	"Override object mode" object *** Obj. no. 32	Window status Obj. no. 34	Presence button object Obj. no. 33	Activated operating mode
X	01	X	X	Comfort
X	02	X	X	Standby
X	03	X	X	Night
X	04	X	X	Frost/heat protection
X	00	1	X	Frost/heat protection
01	00	0	0	Comfort
02	00	0	0	Standby
03	00	0	0	Night
04	00	0	0	Frost/heat protection
01	00	0	1	Comfort
02	00	0	1	Comfort
03	00	0	1	Comfort 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 val of "00" stands for a deactivated override object.

Fig. 1:

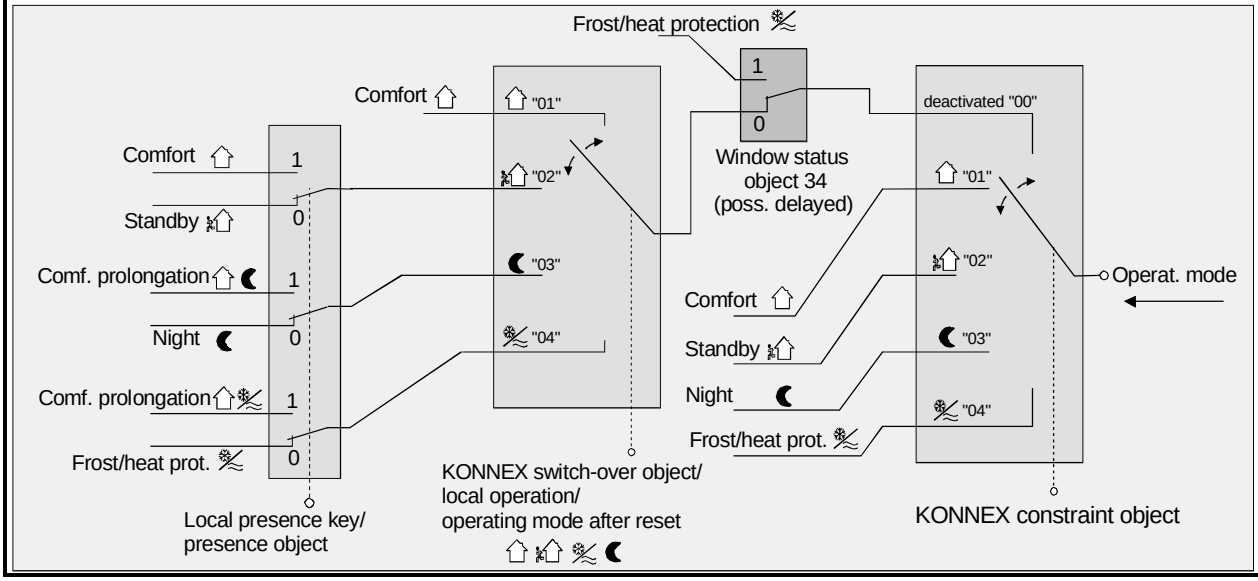


Table 2

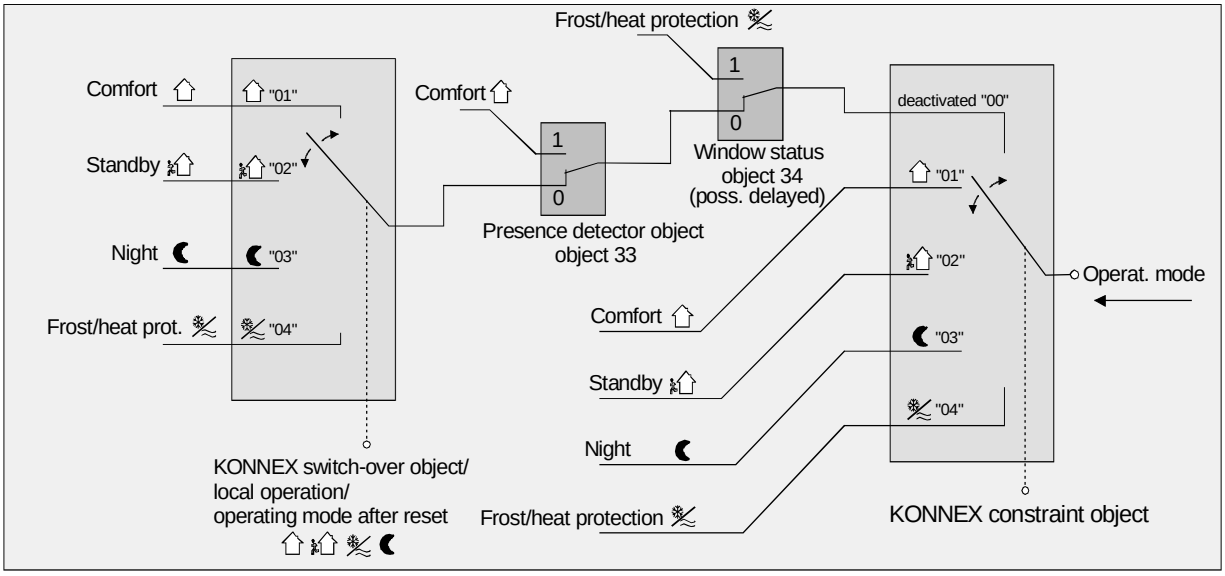
"Operating mode switch-over" object ** Obj. no. 28	"Override object mode" object *** Obj. no. 32	Window status Obj. no. 34	Presence detector object Obj. no. 33	Activated operating mode
X	01	X	X	Comfort
X	02	X	X	Standby
X	03	X	X	Night
X	04	X	X	Frost/heat protection
X	00	1	X	Frost/heat protection
X	00	0	1	Comfort
01	00	0	0	Comfort
02	00	0	0	Standby
03	00	0	0	Night
04	00	0	0	Frost/heat protection
00	00	0	0	Last 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 override 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 the "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 button, taking account of the priorities of the operating modes. If no higher-priority mode (e. g. window contact/ presence detector) and no KONNEX override object has been activated you can use the button or rocker function to switch over by temperature regulator operation on the device an operating mode preset by an object.

  The KONNEX override object will always have the highest priority.
- When a presence button 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 the comfort prolongation is stopped after the prolongation time has elapsed, or if the operating mode has been changed by a higher-priority operation through the switch-over objects or by local operation, or if an operating mode forced by the KONNEX override object is being deactivated (override object → "00").
- Only one operating mode can be activated at a time so that both control circuits will always be in the same mode if two control circuits are used.
  The operating mode switch-over of the second control circuit always proceeds in parallel with the first control circuit.

4.1.2 Notes on the operating modes

Presence function / comfort prolongation:

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

- Presence detection by the presence button:

  If you enable the presence button for presence detection you can select the *"presence button"* setting from the push button functions. In addition, object 33, *"presence object"*, will be enabled.

  In this way, you can actuate the presence button or use a presence object value = "1" to switch over to comfort prolongation when the night or the frost/heat protection mode is active (not activated by the *"window status"* object or by 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 button 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 night 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 button 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 as type of presence detection object 33, *"presence object"*, will only appear. Via this object, you can integrate presence detectors into room temperature control.
  If a movement is detected ("1" telegram) the regulator 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 button. Only the window contact or the automatic frost protection, or the KONNEX override object are of higher priority.
  After the delay time has elapsed in the presence detector ("0" telegram), the regulator will return to the mode which was active before 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 button.
  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 B.IQ push button with room thermostat (RTR) and display offers various options to switch over into the frost/heat protection mode. In addition to the switch-over by the corresponding operating mode switch-over object or by an operating mode switch-over on the push button (button 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 regulator 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 34, *"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 override 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 carried by 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, automatic switch-over to the frost protection mode can be made at times, 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 decrease detected reaches a parameterized value the room temperature regulator will automatically switch over to the frost protection mode. You can use the *"automatic frost protection"* parameter to set the maximum temperature drop in K/min for switching over to the frost protection mode. After the time preset by the *"frost protection period in automatic mode"* parameter has elapsed, the regulator will return into the mode which was set before frost protection. Re-triggering will not be possible.
  If a switch-over was made by the objects (4 x 1 bit or 1 byte) during frost protection and new operating mode was received this followed-up mode will be set after automatic frost protection.
  The KONNEX override 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 set value 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 decrease is not low enough. Switching into the frost/heat protection mode by window contacts should generally 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 regulator 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 operation"*:	The comfort mode will be activated after the initializing phase.
- *"Standby operation"*:	The standby mode will be activated after the initializing phase.
- *"Night peration"*:	The night mode will be activated after the initializing phase.
- *"Frost/heat protection operation"*:	The frost/heat protection mode will be activated after the initializing phase.
- *"Restore operation mode before reset"*:	The mode activated before a reset will be restored after the initializing phase of the device.

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

Notes on the *"restore operation mode 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.
- A presence function including its object activated prior to a reset will be deleted after the reset. The operating mode caused by the presence function, however, will remain active after the reset.
  A comfort prolongation restarted by a reset will be automatically deactivated after its delay time has elapsed.
  The *"window status"* object will be deleted after a reset. Also in this case, the frost/heat protection mode activated before by a window status will remain activated even after a reset.

4.1.3 Controller status

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

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

One-byte status object 36 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	
Regulator in general 1 byte type	Bit 0: 1: Comfort operation active Bit 1: 1: Standby operation active Bit 2: 1: Night operation active Bit 3: 1: Frost/heat protection active	Bit 4: 1: Controller disabled Bit 5: 1: Heating; 0: cooling Bit 6: 1: Controller inactive (dead band) Bit 7: 1: Frost alarm ($T_{room} \leq +5$ °C)

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

One-bit status object 36 contains the status information selected by the *"single 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 Status"	Data Description	
Comfort operation activated	1: Comfort mode/prolongation active	0: No comfort mode
Standby operation activated	1: Standby mode active	0: No standby mode
Night operation activated	1: Night mode active	0: No night mode
Frost/heat protection active	1: Frost/heat protection mode active	0: No frost/heat protection mode
Controller disabled	1: Controller disabled (dew-point operation)	0: Controller not disabled
Heating / Cooling	1: Heating mode	0: Cooling mode
Controller inactivated	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 operation activated: It will be active if the "comfort ' ⌂ '" mode or some comfort prolongation " ⌂ ☾ " or " ⌂ ❄ ", respectively, has been activated.
- Standby operation activated: It will be active if the "standby ' ⌂ '" mode has been activated.
- Night operation activated: It will be active if the "night " ☾ " mode has been activated.
- Frost/heat protection active: It will be active if the "frost/heat protection ' ❄ '" mode has been activated.
- Controller disabled: It will be active if regulator disabling has been activated (dew-point operation).
- Heating / Cooling: 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 regulator has been disabled.)
- Controller inactivated: 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 regulator 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 36 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 regulator variables.

4.2 Heating/cooling modes and heating/cooling mode switch-over

The room temperature regulator has up to two different modes. These modes specify whether you want the regulator 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 regulator 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 set value and the actual value and transmitted to the bus for the basic and additional stages. In this connection, the *"heating/cooling mode"* parameter in the *"room temperature regulator 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 regulator 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 set value temperatures of the operating modes (refer to "4.4 Temperature set values"), the room temperature regulator 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 regulator will always be in the 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 regulator is capable of triggering heating and cooling systems. In this connection, you can set the switch-over behaviour of the modes.

- *"Switch-over between heating and cooling"* parameter in the *"room temperature regulator 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 set value, 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 set value of the activated operating mode when you actuate the display buttons. If the room temperature is higher than the cooling temperature set value cooling will take place. If the room temperature is lower than the cooling temperature set value heating will take place.

  When the heating/cooling mode is changed automatically information can be actively sent to the bus through object 35, *"heating/cooling switch-over",* to indicate whether the regulator 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 be transmitted solely 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 mode takes place, if necessary.

  In addition, the object value can be output in cycles when automatic switch-over is being made. The *"cyclical 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 set value temperatures for the comfort heating and cooling modes) below the default value.

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

  In this case, the heating/cooling mode 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 through the object (often, cold water is used in the one-pipe system for cooling in 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 mode you want to activate after a reset. For the *"heating"* or *"cooling"* settings, the regulator 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 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 for PWM, a short-time 'variable overlapping' could occur during the transition between heating and cooling, due to the matching of the variable at the end of a time cycle. However, such overlapping will be corrected at the end of a PWM time cycle.
Only if heating or cooling energy is required in one of the modes and, consequently, the variable is > "0" the " $^{+}$ " or " $^{-}$ " symbol will appear on the display.

Heating/cooling indication:

Depending on the 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 on the display, or that the heating or cooling signal objects will already become active once the temperature falls below the set value 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 and calculation of variables").

Heating and cooling at the same time will not be possible. The signals are exclusively referred to control circuit 1.

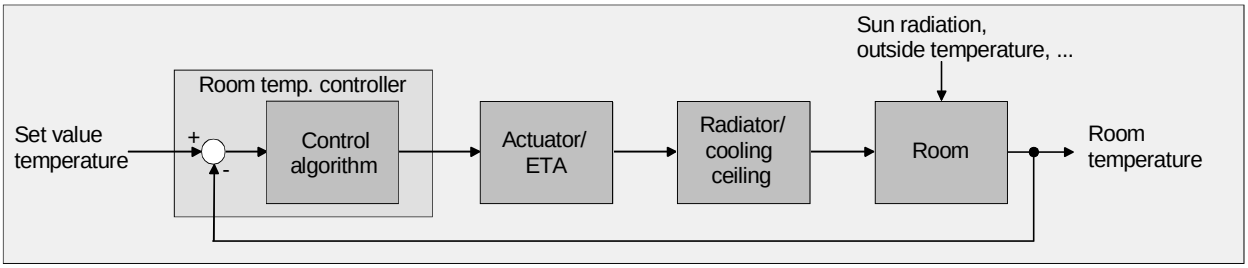
The indication objects can be enabled by the *"heating indication"* or *"cooling indication"* parameters in the *"variable and status output"* parameter branch.

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

4.3 Room temperature control and variables

4.3.1 Control algorithms, control circuits 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 set values and the actual room temperature, the regulator thus determines variables which trigger the heating or the cooling system.
The control system (control circuit) consists of a room temperature regulator, an actuator or switching actuator (when ETA electrothermal drives are used), the actual heating or cooling element (e. g. radiator or cooling ceiling) and of the room. This results in the following controlled system:



The regulator measures the actual temperature (room temperature determined) and compares it with the preselected set value temperature. With the aid of the selected control algorithm, the variable is then calculated from the difference between the actual and the set value temperature. By resetting the variable at regular intervals, the regulator is thus capable of compensating in the control circuit 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 regulator in the B.IQ push button with room thermostat (RTR) and display 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 circuit which solely 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 regulator.

The B.IQ push button with room thermostat (RTR) and display offers the option to activate one or, alternatively, two control circuits. The *"control circuits"* parameter in the *"room temperature regulator function"* parameter branch sets the number of control circuits.

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

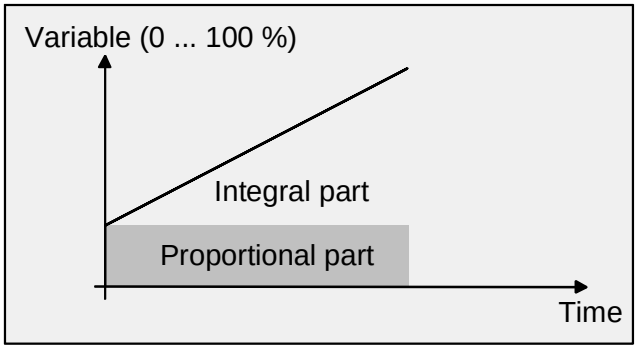
The variables calculated by the control algorithm are output via the *"heating variable"* or *"cooling variable"* communication objects. Depending on the control algorithm selected for the heating and/or cooling mode, the format of the variables objects is, among other things, also specified. So you 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 regulator function"* parameter branch to specify the control algorithm, if necessary, also for the additional stages or for both control circuits, 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 regulator 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 regulator 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 set value will shift, taking account of the parameterized step width.

Special features of the PI control:

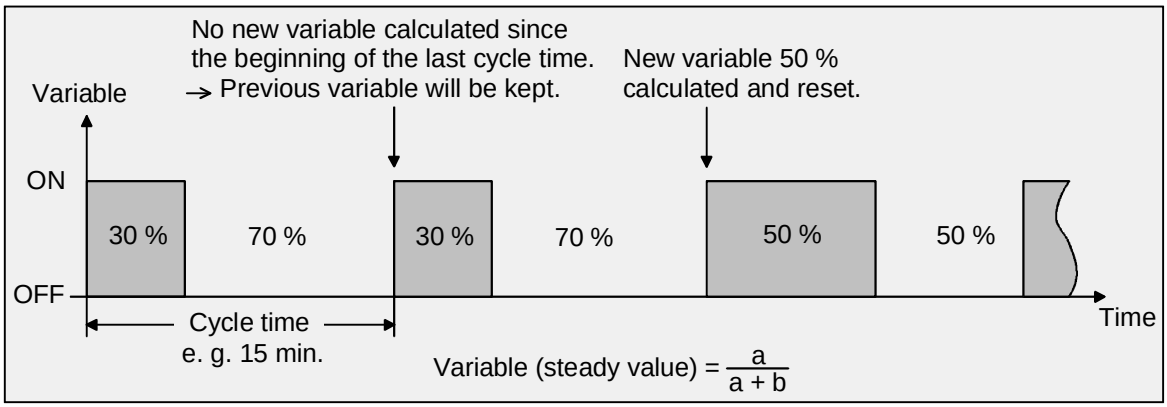
If the room temperature deviation between the actual value and the set value is high enough to have a 100 % variable the room temperature regulator in the B.IQ push button with room thermostat (RTR) and display will work with this maximum variable until the room temperature measured has reached its set value. 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 result as you would obtain with a continuous regulator. 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 regulator 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 regulator 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 set value 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).



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 regulator 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 46 will be sent for the heating mode, and value object 48 for the cooling mode. If you use additional stages, value object 47 for additional heating operation and value object 49 for additional cooling operation will be additionally enabled. For the use of two control circuits, the separate 1 byte value object will not be available.
If you want to output the heating and cooling variables through a joint 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 47 for the additional stages.
The status value objects 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 set value will shift, taking account of the parameterized step width. All PWM control options will use the same cycle time.

Cycle time:

The pulse-width-modulated variables are mainly used for triggering electrothermal drives (ETA). In this connection, the room temperature regulator sends the switching variable telegrams to a switching actuator preferably equipped with semiconductor switching elements which the drives are connected to.

By setting the cycle time of the PWM signal, you can adapt the control to the drives 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 drive 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 enabled 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.

In common practice, 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 sufficient 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 (e. g. for smaller warm-water radiators), 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 regulator 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 in adjustment of the latter with major deviations, respectively.

Note:

- Such cycle time setting is recommended for less sluggish heating systems (such as warm-water radiators with a higher flow temperature).

3. Switching 2-point control:

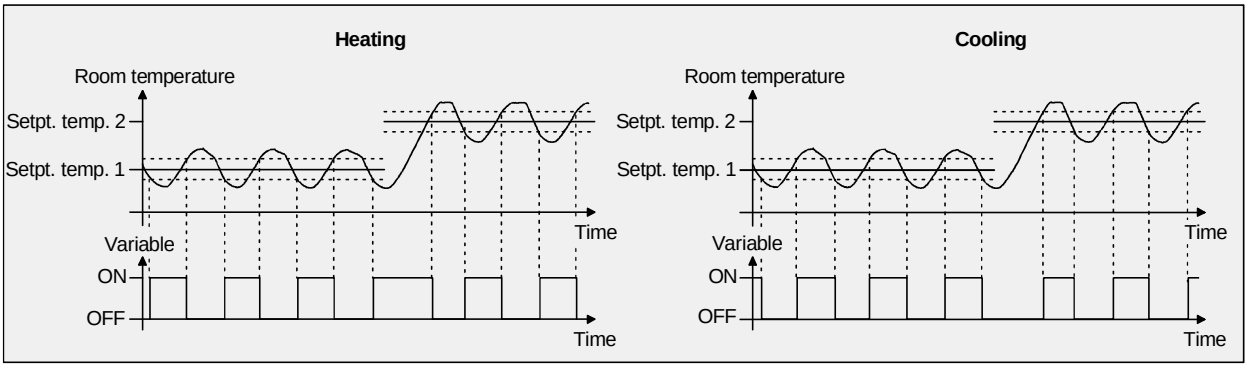
The 2-point control represents a very simple type of temperature control. For this type of control, two hysteresis temperature values are preset. The actuators are triggered by the regulator via switch-on and switch-off variable commands (1 bit type). A continuous variable is not calculated for this 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 regulator will turn on the heating 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 regulator 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 on the display, or that the heating or cooling signal objects will already become active once the temperature is falling below the set value 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 set values; 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 set value and the hysteresis values will shift, taking account of the parameterized step width.

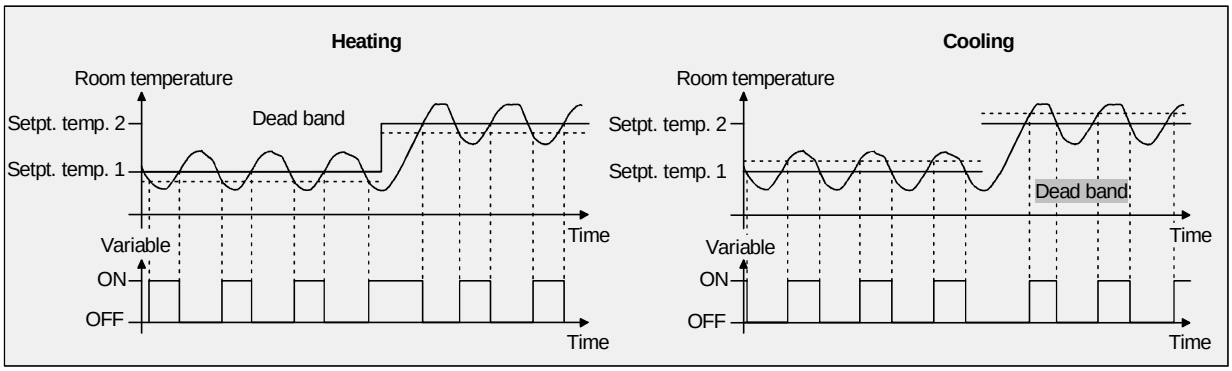
- *"Heating and cooling"* mixed mode:

In the mixed mode, distinction is made whether the switch-over between heating and cooling is to be effected automatically or in a controlled way through the object.

For automatic switch-over:
In the heating mode, the controller will turn on the heating when the room temperature has fallen below a preset hysteresis 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 hysteresis 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 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.

For mode switch-over through the object:
In the heating mode, the controller will turn on the heating when the room temperature has fallen below a preset hysteresis limit. The control system will only turn off the heating in the heating mode once the preset upper hysteresis limit has been exceeded. In the cooling mode, the controller will turn on the cooling system when the room temperature has exceeded a preset hysteresis limit. The control system will only turn off the cooling system in the cooling mode once the temperature has fallen below the preset lower hysteresis limit.
Same as for the individual modes of heating or cooling, there are two hysteresis limits per mode. Although there is a dead band for the calculation of the temperature setpoints for cooling, it has no influence of the calculation of the two-point variable, as the operating mode is switched over "manually" through the corresponding object. Within the hysteresis spans, it thus will be possible to request heating or cooling energy for temperature values that are located within the dead band.

The following illustration shows a two-point control system for the *"heating and cooling"* mixed mode (activated heating mode on the left, activated cooling mode on the right, two temperature setpoints, non-inverted variable output, automatic mode switch-over – for mode switch-over through the object, an upper hysteresis for heating and a lower hysteresis for cooling can, in addition, be parameterized):



Depending on the switching state, the actuating variable "1" or "0" will be output, if the values exceed or remain under the hystereses limits. It has to be pointed out that the " + " or " - " symbols will light up on the display or that the message objects for heating and cooling will already become active as soon as the temperature falls short of the temperature setpoint in case of heating or exceeds the temperature setpoint in case of cooling. In this case the hystereses is not being considered.
The upper and lower hystereses limit is to be parameterized in the ETS plug-in for both heating/cooling modes.

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

4.3.2 Adapting the control algorithms

4.3.2.1 Adapting the PI control

There are several systems available, which may heat or cool a room. One option is to uniformly heat or cool the surroundings via heat transfer media (preferably water or oil) in combination with room air convection. Such systems are used, for example, with wall mounted heaters, underfloor heating or cooling ceilings.
Alternatively or additionally forced air systems may heat or cool rooms. In most cases such systems are electrical forced hot air systems, forced cool air systems or refrigerating compressors with fan. Due to the direct heating of the room air such heating and cooling systems work quite swiftly.

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

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

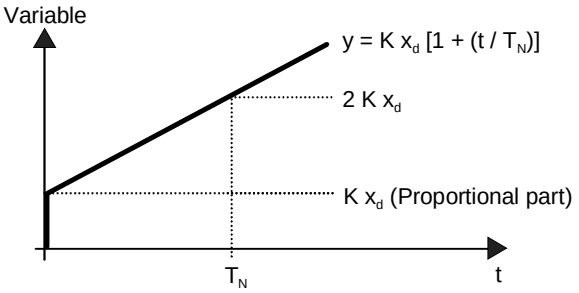
For heating control				
Type of heating	**Default values**		**Recommended**	**Recommended**
	Proportional range	**Reset-time**	**Type of PI control:**	**PWM cycle time**
• Hot water heating	5 Kelvin	150 minutes	continuous / PWM	15 minutes **
• Underfloor heating	5 Kelvin	240 minutes	PWM	15 – 20 min.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

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

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

Notes:

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



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

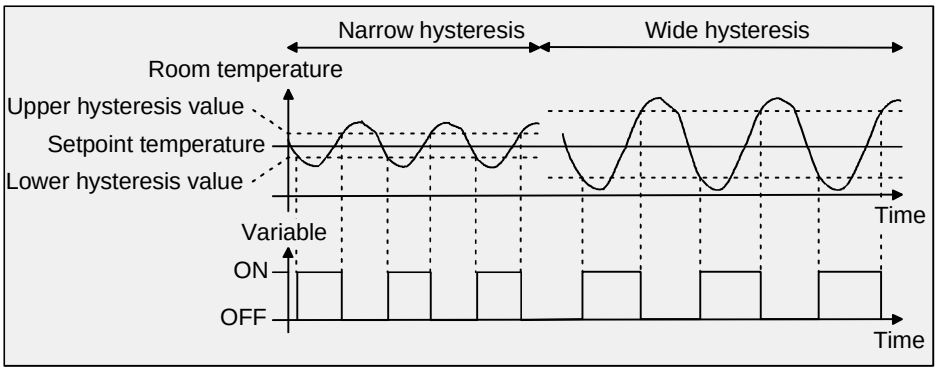
P control algorithm: Actuating variable $y = K x_d$

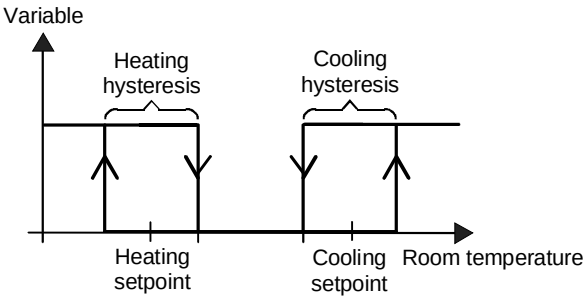
Parameter setting		Effect
P	small proportional range	large overshooting in case of setpoint changes (possibly permanently), quick adjustment to the setpoint
P	large proportional range	no (or small) overshooting but slow adjustment
T_N	short reset-time	fast compensation of control deviations (ambient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

4.3.2.2 Adapting the 2-point control

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

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





4.3.3 Actuating variable output

4.3.3.1 Actuating variable objects

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

Possible object data formats for the actuating variables separately for both heating/cooling modes, for the basic and the additional stage or for both control circuits are…

- continuous PI control: 1-byte,
- switching PI control: 1-bit + additional 1-byte (for example for the status indication with visualizations),
- switching 2-point control: 1-bit.

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

Case 1: Heating and cooling system are two separate systems.
In this case the *"send variable heating and cooling to one common object"* parameter should be set to *"no"* (default) in the *"room temperature regulator functions"* parameter branch. Thus, there are separate objects available for each actuating variable, which can be separately addressed via the individual systems. This setting allows to define separate types of control for heating and cooling.

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

Note:

Basically, it is not possible to heat and cool at the same time (actuating variable > "0"). Only for PWM, a short-time 'variable overlapping' could occur during the transition between heating and cooling, due to the matching of the variable at the end of a time cycle. However, such overlapping will be corrected at the end of a PWM time cycle.

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

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

4.3.3.2 Automatic transmission

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

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

  If the periodic and the automatic transmission are both deactivated, no actuating telegrams will be transmitted in case of a change!

- Switching PI control (PWM):
  In case of a switching PI control (PWM), the room temperature regulator calculates a new actuating variable internally every 30 seconds. The update of the actuating variable, however, takes place, if required, solely at the end of a cycle. The "automatic transmission at modification by..." and "cycle time for automatic transmission..." parameters are not enabled with this control algorithm.

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

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

4.4 Temperature setpoints

4.4.1 Setpoint presettings in the ETS

Temperature setpoints can be preset for each operating mode. It is possible to parameterize the setpoints for the "comfort", "standby" und "night" modes in the ETS plug-in. If desired, the set-temperatures can be subsequently adjusted via local control in programming mode or via object control. The "frost/heat protection" operating mode allows the separate parameterization of two temperature setpoints for heating (frost protection) and cooling (heat protection) solely in the ETS.

When presetting the set-temperatures for comfort, standby and night mode attention has to be paid to the fact that all setpoints depend on each other as all values are derived from the basic temperature (basic setpoint). The *"basic temperature after reset"* parameter in the *"set point values"* parameter branch determines the basic setpoint, which is loaded when the device is programmed via the ETS.
Taking into account the *"lower/raise the setpoint temperature in standby mode"* or *"lower / raise the setpoint temperature in night mode"* parameters the temperature setpoints for the standby and night mode are derived from this value depending on the heating or cooling heating/cooling mode. The Dead band will be additionally considered for the *"Heating and cooling"* mode.

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

When using two control circuits it is possible to set shared setpoints or alternatively separate values for both circuits. The *"own setpoints for the 2nd control circuit"* parameter in the *"room temperature regulator function – set point values"* parameter branch determine the setpoints:

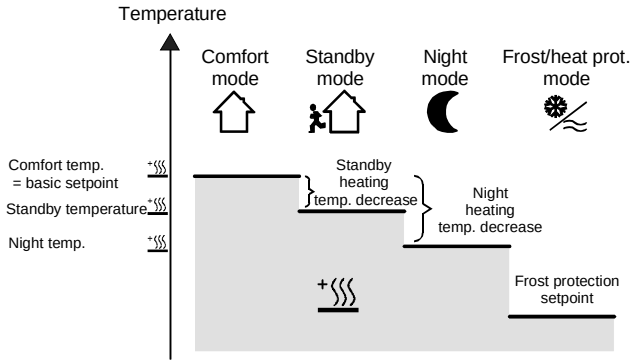
- Settings: *"no"* (default):
  Both control circuits have the same setpoints for the comfort, standby and night mode. The frost or heat protection temperatures are identical as well. This setting features, if enabled, a shared object for setting the basic setpoints and an object for transmitting the set-temperature on the bus.

- Setting *"yes"*:
  Independent of each, both control circuits have other their own setpoints for the comfort, standby and night mode. Only the frost or heat protection temperatures are identical. With this setting there are separate objects per control circuit available for setting the basic setpoint or transmitting the set-temperature, if enabled. A change of the set-temperature in programming mode on the push button is only possible for the first control circuit.

The operating mode switch-over of the second control circuit always takes place parallel to the switch-over of the first control circuit. It is not possible to have a two-stage control and a mixed-mode of heating and cooling with two control circuits.

Depending on the heating/cooling mode, the relationships described on the following pages have to be considered for the set-temperatures.
When using two control circuits the heating/cooling mode for both circuits can be set to either "heating" or "cooling". In this case, it is not possible to have a "heating and cooling" mixed-mode!

4.4.1.1 Setpoints for the "heating" heating/cooling mode



(Recommended set temperatures)

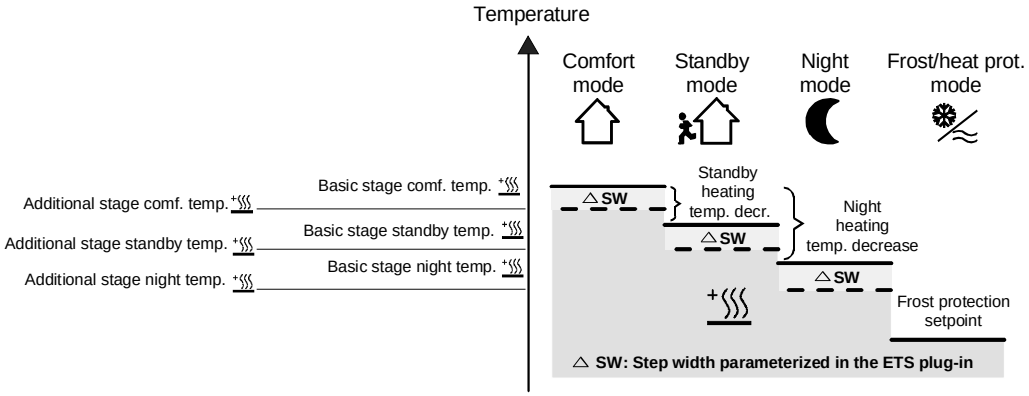
The set-temperatures for comfort, standby and night mode exists for this heating/cooling mode. The frost protection temperature can be preset. The following applies:

$$T_{\text{Standby set value heating}} \leq T_{\text{Comfort set value heating}} \quad \text{or} \quad T_{\text{Night set value heating}} \leq T_{\text{Comfort set value heating}}$$

The standby and night set-temperatures are derived after the parameterized decrease-temperatures from the comfort set-temperature (basic setpoint). It is also possible to adjust other decrease-temperatures via local control in programming mode on the controller, if enabled, by changing the set-temperature values for night and standby mode (cf. "1.4 Programming mode / local control"). This local control is only possible for control circuit 1!

The frost protection is supposed to prevent the heating system from freezing. For this reason the frost protection temperature should be set to a smaller value than the night temperature for heating (default: +7 ℃). In principle, however, it is possible to se lect frost protection temperature values between +7 ℃ and +40 ℃.
The possible range of values for a set-temperature lies between + 7,0 ℃ and + 99,9 ℃ for "heating" and is bounded by the frost protection temperature in the lower range.

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



$$T_{\text{Comfort set value additional stage heating}} \leq T_{\text{Comfort set value basic stage heating}} \;/\; T_{\text{Standby set value additional stage heating}} \leq T_{\text{Standby set value basic stage heating}}$$

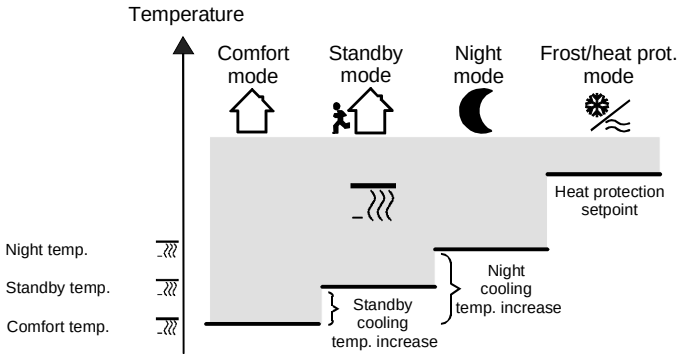
$$T_{\text{Standby set value heating}} \leq T_{\text{Komfort set value heating}}$$

or

$$T_{\text{Comfort set value additional stage heating}} \leq T_{\text{Comfort set value basic stage heating}} \;/\; T_{\text{Night set value additional stage heating}} \leq T_{\text{Night set value basic stage heating}}$$

$$T_{\text{Night set value heating}} \leq T_{\text{Comfort set value heating}}$$

4.4.1.2 Setpoints for the "cooling" heating/cooling mode



(Recommended set temperatures)

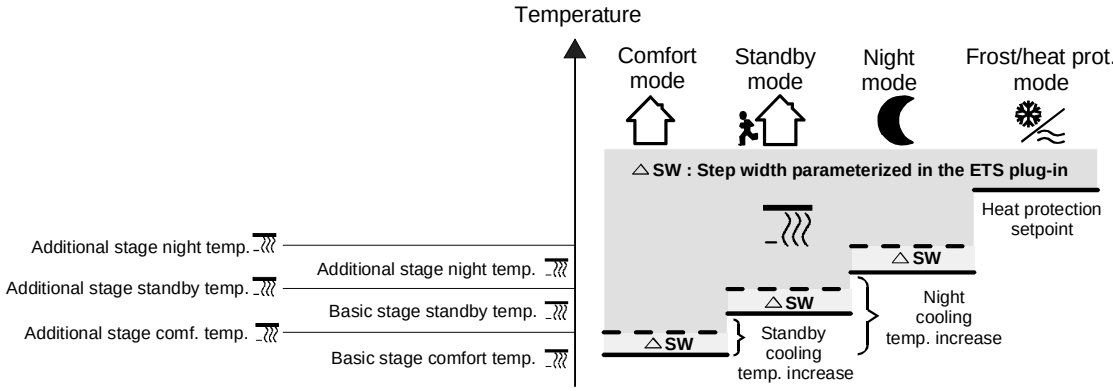
The set-temperatures for comfort, standby and night mode exists for this heating/cooling mode. The heat protection temperature can be preset. The following applies:

$$T_{\text{Comfort set value cooling}} \leq T_{\text{Standby set value cooling}} \quad \text{or} \quad T_{\text{Comfort set value cooling}} \leq T_{\text{Night set value cooling}}$$

The standby and night set-temperatures are derived after the parameterized increase-temperatures from the comfort set-temperature (basic setpoint). It is also possible to adjust other increase- temperatures via local control in programming mode on the controller, if enabled, by changing the set-temperature values for night and standby mode (cf. "1.4 programming mode / local control"). This local control is only possible for control circuit 1!

The heat protection is supposed to prevent the temperature from exceeding the maximum permissible room temperature in order to protect system components. For this reason, the heat protection temperature should be set to a larger value than the night temperature (default: +35 °C). In principle , however, it is possible to select heat protection temperature values between +7 °C and +45 °C.
The possible range of values for a set-temperature lies between - 99,9 °C and + 45,0 °C for "cooling" and is bounded by the heat protection temperature in the upper range.

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



$$T_{\text{Comfort set value basic stage cooling}} \leq T_{\text{Comfort set value additional stage cooling}} \;/\; T_{\text{Standby set value basic stage cooling}} \leq T_{\text{Standby set value additional stage cooling}}$$

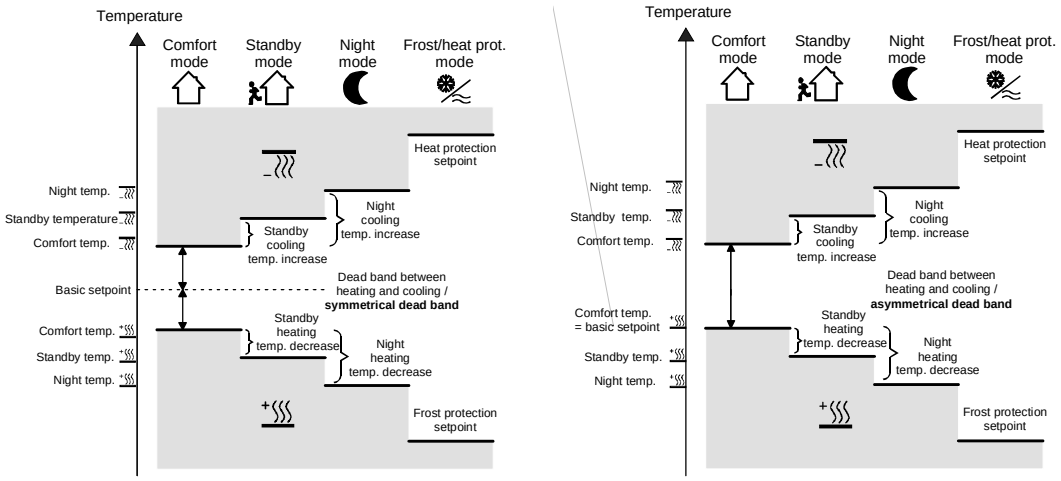
$$T_{\text{Comfort set value cooling}} \leq T_{\text{Standby set value cooling}}$$

or

$$T_{\text{Comfort set value basic stage cooling}} \leq T_{\text{Comfort set value additional stage cooling}} \;/\; T_{\text{Nacht set value basic stage cooling}} \leq T_{\text{Nacht set value additional stage cooling}}$$

$$T_{\text{Comfort set value cooling}} \leq T_{\text{Night set value cooling}}$$

4.4.1.3 Setpoint for the "heating and cooling" heating/cooling mode:



(Recommended set temperatures)

For this heating/cooling mode, the set-temperatures of both heating/cooling modes exist for comfort, standby and night mode as well as the Dead band. In addition, the frost protection and the heat protection temperatures can be preset. The following applies:

$$T_{\text{Standby set value heating}} \leq T_{\text{Comfort set value heating}} \leq T_{\text{Comfort set value cooling}} \leq T_{\text{Standby set value cooling}}$$

or

$$T_{\text{Night set value heating}} \leq T_{\text{Comfort set value heating}} \leq T_{\text{Comfort set value cooling}} \leq T_{\text{Night set value cooling}}$$

The set-temperatures for "Standby" and "Night" are derived from the comfort set-temperatures for heating or cooling. The temperature increase (for cooling) and the temperature decrease (for heating) of both operating modes can be preset in the ETS plug-in. The comfort temperatures itself are derived from the Dead band and the basic setpoint (cf."Dead band" on the following page).

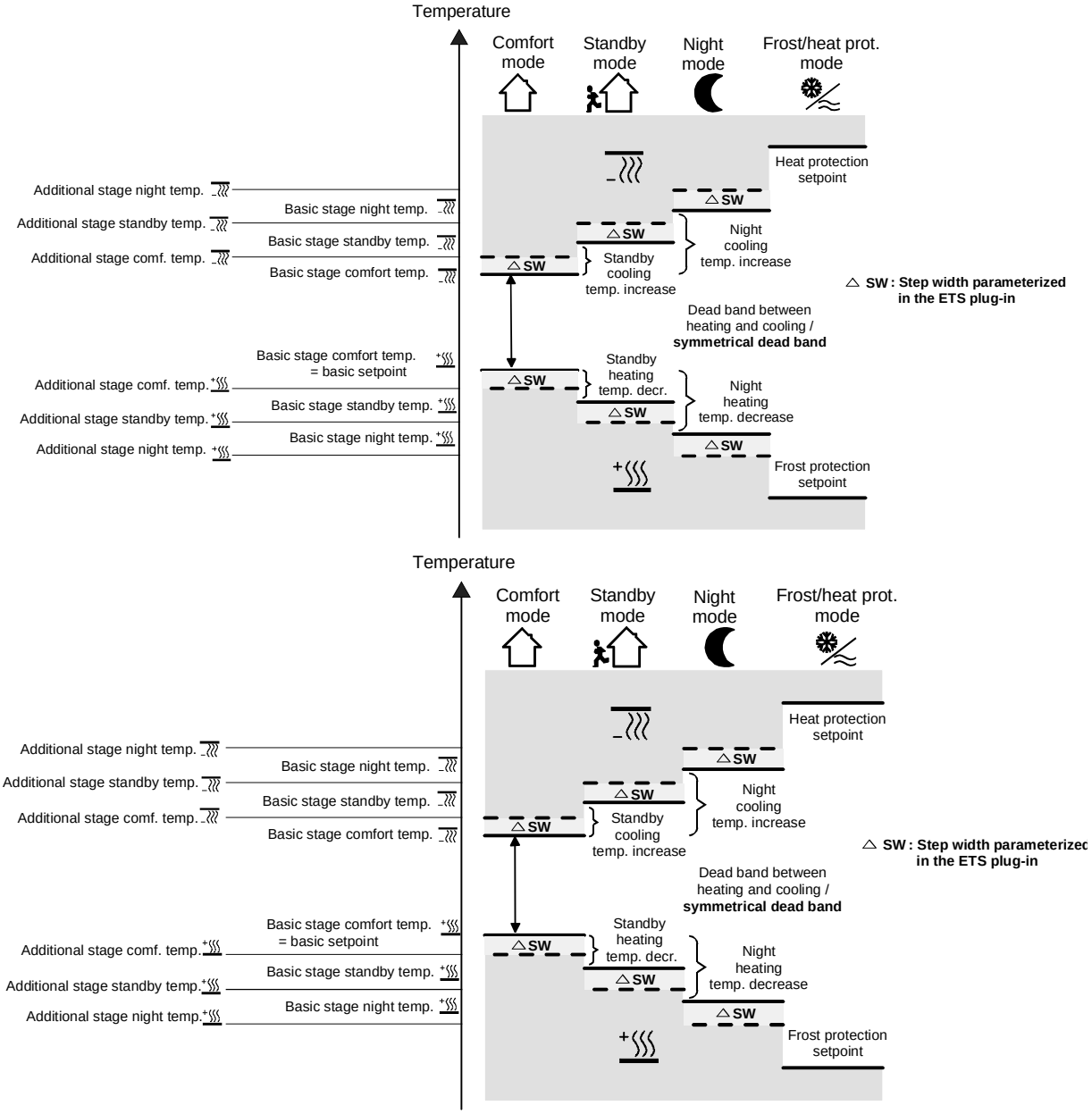
Note:

It is also possible to adjust other decrease or increase-temperatures via local control in programming mode on the controller, if enabled, by changing the set-temperature values for night and standby mode (cf. "1.4 Programming mode / local control"). This local control is only possible for control circuit 1!

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

The possible range of values for a set-temperature ("heating and cooling") lies between + 7 °C and + 45,0 °C and is bounded by the frost protection temp erature in the lower range and by the heat protection temperature in the upper range.

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



$T_{\text{Comfort set value additional stage heating}} \leq T_{\text{Comfort set value basic stage heating}} \leq T_{\text{Comfort set value basic stage cooling}} \leq T_{\text{Comfort set value additional stage cooling}}$

$T_{\text{Standby set value additional stage heating}} \leq T_{\text{Standby set value basic stage heating}} \leq T_{\text{Standby set value basic stage cooling}} \leq T_{\text{Standby set value additional stage cooling}}$

$T_{\text{standby set value heating}} \leq T_{\text{Comfort set value heating}} \leq T_{\text{Comfort set value cooling}} \leq T_{\text{Standby set value cooling}}$

or

$T_{\text{Comfort set value Zusatzstufe Heizen}} \leq T_{\text{Comfort-set value basic stage heating}} \leq T_{\text{Comfort set value set value cooling}} \leq T_{\text{Comfort set value additional stage cooling}}$

$T_{\text{Night set value additional stage heating}} \leq T_{\text{Night set value basic stage heating}} \leq T_{\text{Night set value basic stage cooling}} \leq T_{\text{Night set value additional stage cooling}}$

$T_{\text{Night set value heating}} \leq T_{\text{Comfort set value heating}} \leq T_{\text{Comfort set value cooling}} \leq T_{\text{Night set value cooling}}$

Dead band:

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

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

$T_{Basic\ set\ value} - ½T_{Dead\ band} = T_{Comfort\ set\ value\ heating}$ or $T_{Basic\ set\ value} + ½T_{Dead\ band} = T_{Comfort\ set\ value\ cooling}$

➔ $T_{Comfort\ set\ value\ cooling} - T_{Comfort\ set\ value\ heating} = T_{Dead\ band}$; $T_{Comfort\ set\ value\ cooling} \geq T_{Comfort\ set\ value\ heating}$

Important notes on the symmetrical Dead band:

- In case of a symmetrical Dead band, the basic setpoint is indirectly set via the comfort temperature for heating (local control in programming mode). For this reason the basic setpoint is not shown on the display!
- Changing the comfort set-temperature for cooling allows the adjustment of the Dead band with local control, if enabled (Dead band shifting). An adjustment of the Dead band with a symmetrical dead band position will result in a shifting of the comfort set-temperature for heating and thus of all other temperature setpoints. It is possible to preset the Dead band to 0 °C (result: $T_{Comfort\ set\ value\ cooling} = T_{Comfort\ set\ value\ heating}$). In this case there is neither heating nor cooling, if the determined room temperature equals the comfort set-temperatures.

Dead band position = "asymmetrical":
With this setting the comfort set-temperature for heating equals the basic setpoint!
The Dead band preset in the ETS plug-in is effective only from the basic setpoint in the direction of comfort temperature for cooling. Thus the comfort set-temperature for cooling is derived directly from the comfort setpoint for heating. The following applies:

$T_{Basic\ set\ value} = T_{Comfort\ set\ value\ heating}$ ➔ $T_{Basic\ set\ value} + T_{Dead\ band} = T_{Comfort\ set\ value\ cooling}$

➔ $T_{Comfort\ set\ value\ cooling} - T_{Comfort\ set\ value\ heating} = T_{Dead\ band}$; $T_{Comfort\ set\ value\ cooling} \geq T_{Comfort\ set\ value\ heating}$

Important notes on the asymmetrical Dead band:

- Changing the comfort set-temperature for cooling allows the adjustment of the Dead band with local control, if enabled (Dead band shifting). With an asymmetrical dead band position, an adjustment of the comfort set-temperature for cooling will only shift the temperature setpoints for cooling. It is possible to preset the Dead band to 0 °C (result: $T_{Comfort\ set\ value\ cooling} = T_{Comfort\ set\ value\ heating}$). In this case there is neither heating nor cooling, if the determined room temperature equals the comfort set-temperatures.

Setpoint temperature feedback within the dead band:
If the room temperature is within the preset dead band and you actuate the display keys the display will read the heating and cooling temperature setpoints which were active last. Likewise, the corresponding temperature value will be sent to the bus via the *"setpoint temperature"* object.

4.4.2 Adjusting the setpoints

4.4.2.1 Adjusting basic temperature and set temperatures for comfort, standby and night mode

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

It is possible to change or adjust the set-temperatures 'later' via local control in programming mode on the controller or via object control (*"basic setpoint"* object). After the programming, the set-temperatures of the second control circuit can only be adjusted via object control.

Any change must always be enabled in the ETS plug-in in the *"setpoints"* parameter branch. It is possible…

- to permit the *"modification of the basic temperature setpoint value" by* directly changing the comfort temperature for heating on the device (local; only control circuit 1) and/or by setting a new basic setpoint via the bus (object 26 for control circuit 1 / object 27 for control circuit 2,
- to allow the "*1st control circuit standby temperature change"* by directly changing the standby temperatures for heating or cooling of the first control circuit only on the device (local).
- to permit the "1st control circuit night temperature change" by directly 1st control circuit night temperature changes for heating or cooling of the first control circuit only on the device (local).
- to enable the *"Dead band shifting"* by changing the comfort set-temperature for cooling only on the device (local; control circuit 1 when "heating and cooling") and
- to prevent the *"modification of the setpoints 'cooling'"* in programming mode on the device when in mixed-mode.

If a change is not enabled (setting: "deactivated"), a ‘subsequent’ adjustment of the value predetermined by the ETS cannot take place and setting the corresponding temperature values locally will not be possible.
In case the basic setpoint adjustment via the bus is disabled, the object 26 or 27 will be hidden.

Note:
The incrementation of a change of temperature values via the basic setpoint object or by local operation from the programming menu is defined by the setpoint shifting incrementation. The *"setpoint shifting step value"* parameter in the *"room temperature regulator function/setpoint values"* parameter branch defines the maximum shifting step width. The push button module rounds the temperature values received via the *"basic setpoint"* object and matches the values to the parameterized step value. In this connection, there will be no feedback of the corrected temperature value through the object.

Adjusting the basic setpoint / comfort temperature for heating:

One has to distinguish between two cases only if the basic setpoint has been adjusted, (via local control and/or via the object):

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

Via the "accept modification of the basic temperature setpoint value permanently" parameter in the "room temperature regulator function / set point values" parameter branch it is possible to determine whether the set basic temperature value shall be stored in memory permanently ("Yes") or only temporarily ("No").

Case 1:

If the basic temperature setpoint of the first or second control circuit is adjusted, it will be permanently stored in the push button's EEPROM memory. The newly adjusted value will overwrite the basic set-temperature originally parameterized via the ETS! This is the only way to keep the adjusted basic setpoint even after switching-over the operating mode or after a reset.

Notes:

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

Case 2:

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

Adjusting the setpoints for standby mode, night mode and Dead band (comfort temperature for cooling):

Since the set-temperatures for the "standby" and "night" operating modes or the setpoints for the "cooling" heating/cooling mode are derived from the basic set-temperature - in consideration of the increase, decrease or Dead band values that are parameterized in the ETS plug-in - these set-temperatures will shift linearly by the change of the basic setpoint value.

In addition, it is possible adjust set-temperatures exclusively via local control in the programming mode for "standby" and "night" other than parameterized in the ETS for the first control circuit. In this case, the originally parameterized decrease, increase temperatures or Dead band temperatures will be replaced by the new values resulting from the locally adjusted temperature setpoints. Independent of the "*accept modification of the basic temperature setpoint value permanently*" parameter, the temperature setpoints for the standby or night mode or "cooling" comfort mode (Dead band) will always be stored in the non-volatile EEPROM memory.

4.4.2.2 Basic setpoint shifting

In addition to the setting of individual temperature setpoints via the ETS, the user is able to shift the basic setpoint within a settable range anytime via local control in programming mode or via the basic setpoint object.

During normal operation (no programming mode is active), the set-temperature for the activated operating mode of the first control circuit can be shown on the display by actuating one of the two display keys. In addition, the "setpoint shifting" push button function can be adjusted (cf. "3.3 Rocker functions") if the room temperature function is enabled. ("3.3 Rocker functions"). An actuated key parameterized to function as such will activate – just like the display keys – the temperature display of the setpoint.

Furthermore, you can press the right display key to shift downward or press the left key to shift upward the displayed setpoint temperature in parameterized increments/decrements. Analogous to the shifting via the display keys, a push button's function key parameterized to a setpoint shifting can decrease or increase the shifting value. The direction of the value adjustment is determined via the "command on pressing the push button" parameter in the "push button function / general / [key designation]" parameter branch.

A long key-press will continue the adjustment. An adjustment takes place every 0.5 seconds.
The hand symbol " " on the display indicates that a basic setpoint shifting has been set. The adjusted temperature value is instantly accepted as the new setpoint.

It has to be considered that a shifting of the displayed set-temperature (temperature offset of the basic temperature) will directly affect the basic setpoint and as a result shift all other temperature setpoints.
In case of two control circuits with separate setpoints, the set-temperatures of both circuits will be shifted.

Whether a basic setpoint shifting only affects the currently active operating mode or whether it influences all other set-temperatures of the remaining operating modes is determined by the *"accept modification of shift of basic setpoint value permanently"* parameter in the *"setpoint"* parameter branch.

Settings: "no" (default):
The basic setpoint shifting carried out is in effect for only as long as the operating mode or heating/cooling mode has not changed or the basic setpoint is maintained. Otherwise the setpoint shifting will be reset to "0".

Setting "yes":
In general, the shifting of the basic setpoint carried out affects all operating modes. The shifting is maintained even after switching-over the operating mode or the heating/cooling mode or readjusting the basic setpoint.

The adjustable temperature range for a basic setpoint shifting is defined via the "upward adjustment of basic setpoint temperature" or *"downward adjustment of basic setpoint temperature"* parameters. It is possible to shift the current setpoint by a maximum of +/- 10 K. The incrementation of a setpoint shift can be parameterized by the *"setpoint shifting step value"* parameter in the "*room temperature regulator function/setpoint values"* parameter branch. Each setpoint shifting key actuation changes the setpoint temperature value, depending on the incrementation. The push button module rounds the temperature values received via the *"basic setpoint"* object and matches the values to the parameterized step value.

Notes on the basic setpoint shifting:

- Since the value for the basic setpoint shifting is stored exclusively in volatile memory (RAM), the shifting will get lost in case of a reset (e.g. bus voltage failure).
- A setpoint shifting does not affect the temperature setpoints for frost or heat protection!
- If the control is actively disabled, there will be no response to an actuation of the left or right display key or a "setpoint shifting" function key.
- The basic setpoint shifting has no influence on the "basic setpoint" object.

Communication objects for the basic setpoint shifting:

The controller tracks the current setpoint shifting in the communication object 52 (*"current setpoint shifting"*) via the controller with a 1-byte counter value (acc. to KNX DPT 6.010 – representation of positive and negative values in a 2's complement) This object is found in the *"room temperature regulator function / set point values"* parameter branch. By connecting to this object the controller extensions are also able to display the current setpoint shifting.

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

Example:

Initial situation:
The temperature increment for the setpoint shifting is set to 0.5 K.
Current set-temperature = 21.0 °C / counter value i n Object 52 = "0" (no active setpoint shifting)

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

etc.

The maximum possible value range of the *"current setpoint shifting"* communication object depends on the preset setpoint shifting incrementation (0.1 K or 0.5 K) and on the *"upward/downward adjustment of basic setpoint temperature"* parameter. With a parameterization of ± 10 K at this position and an incrementation of "0.5 K", the value of the object ranks withinin a maximum span of -20 to +20.

In addition, the setpoint shifting of the controller can be externally adjusted via the communication object 53 (*"preset setpoint shifting"*). This object can also be found in the *"room temperature regulator function / set point values"* parameter branch and has the same data point type and range of values as the object 52 (see above). By connecting to the object 53 the controller extensions are also able to directly adjust the current setpoint shifting of the controller.

As soon as the controller receives a value, it will adjust the setpoint shifting correspondingly. In this connection, each value step corresponds to the parameterized temperature step (0.1 K oder 0.5 K – cf. example above).Values that lie within the possible range can be directly jumped to.

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

4.4.3 Transmitting the set-temperature

The set-temperature, which is given by the active operating mode or has been subsequently adjusted, can be actively transmitted on the bus via the object 50, or in case of two control circuits with separate setpoints additionally via the *"set temperature"* object 51.
The *"transmission at setpoint temperature modification by..."* parameter in the *"room temperature regulator functions – set point values"* parameter branch determines the temperature value by which the setpoint has to change in order to have the set-temperature value transmitted automatically via the object. Temperature value changes between 0.1 C and 25.5 C or 0.1 K and 25.5 K are possible. The setting "0" at this point will deactivate the automatic transmission of the set temperature.
In addition, the setpoint can be transmitted periodically. The "cyclical transmission of setpoint temperature" parameter determines the cycle time (1 to 255 minutes). The value "0" (default) will deactivate the periodical transmission of the set-temperature.
It has to be pointed out that in case of a deactivated periodical transmission and a deactivated automatic transmission, no more set-temperature telegrams will be transmitted".

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

4.5 Room temperature measurement

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

The push button RTR features an integrated temperature sensor. Alternatively (e.g. if the push button has been installed in an unfavourable location or operates in difficult conditions, for example, in a moist atmosphere) or additionally (e.g. in large rooms or halls), a second KNX/EIB temperature sensor externally coupled via the bus may be used to determine the actual value in case of only one control circuit.
When using two control circuits, the actual-temperature of the second circuit is determined by the external sensor. The first control circuit can measure the actual-temperature only via the external sensor!

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

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

4.5.1 Temperature detection and determination of measured value

The temperature detection with one control circuit depends on the following parameterization. When using both control circuits, the actual-temperature of the second circuit is determined by the external sensor.

1 control circuit:
When using only one control circuit the *"temperature detection"* parameter in the *"room temperature regulator function – room temperature measuring"* parameter branch will determine which one of the sensors shall determine the actual-temperature. The following settings are possible:

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

- *"External sensor"* (not with controller extension):
  The actual-temperature is determined solely via the external temperature sensor. The internal sensor is deactivated. The external sensor must transmit the detected temperature value on the push button's 2-byte object 24 *"external temperature sensor"* (DPT-ID 9.001). Alternatively or additionally, the push button can periodically request the current temperature value (set "R" flag for the external sensor!). The *"scanning time for external sensor..."* parameter has to be set to a value > "0". The measurement interval can be set from 1 minute to 255 minutes. When parameterized as such, the room temperature regulator will wait for a temperature value telegram from the external temperature sensor after a reset until the control starts and an actuating variable, if applicable, is output.

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

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

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

  Internal sensor: 21.5 °C (measuring range of inte rnal sensor: 0 °C … + 40 °C ±1%)
  External sensor: 22.3 °C
  Determination of measured value: 30% to 70 %

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

2 control circuits:
The first control circuit measures the actual-temperature exclusively via the internal sensor. The external sensor measures the actual-temperature of the second control circuit and must transmit the temperature value on the *"external temperature sensor"* (DPT-ID 9.001) 2-byte object 24. Alternatively or additionally, the push button can periodically request the current temperature value (set "R" flag for the external sensor!). The "scanning time for external sensor..." parameter has to be set to a value > "0". The measurement interval can be set from 1 minute to 255 minutes. This parameterization causes the room temperature regulator to wait for a temperature value telegram from the external temperature sensor after a reset until the control of the second circuit starts and an actuating variable, if applicable, is output.

4.5.2 Calibrating the measured values

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

The *"adjustment internal sensor..."* or *"adjustment external sensor..."* parameter in the *"room temperature regulator function –room temperature measuring"* parameter branch allows to parameterize the positive temperature increase, factors 1…127) or negative (temperature decrease, factors – 128…-1) temperature adjustment in 0.1℃ increments. Thus the calibratio n is adjusted only once and is the same for all operating modes.

Notes:

- The measured value has to be increased, if the value measured by the sensor lies below the actual room temperature. The measured value has to be decreased, if the value measured by the sensor lies above the actual room temperature.
- When determining the measured value between the internal and external sensor with only one control circuit, the calibrated value is used to calculate the actual value.

4.5.3 Transmitting the actual-temperature

The determined actual-temperature of the first control circuit can be actively transmitted on the bus via the *"actual-temperature"* object 23.
The "transmission at rom temperature modification by..." parameter in the "room temperature regulator functions –room temperature measuring" parameter branch determines the temperature value by which the actual value has to change in order to have the actual-temperature value transmitted automatically via the object 23. Temperature value changes between 0.1 C and 25.5℃ or 0.1 K and 25.5 K are possible.
The setting to "0" at this point will deactivate the automatic transmission of the actual-temperature.
In addition, the actual value can be transmitted periodically. The "cyclical transmission of room temperature" parameter determines the cycle time (1 to 255 minutes). The value "0" (default) will deactivate the periodical transmission of the actual-temperature value.
Setting the „R" flag on the "actual-temperature" object makes it possible to read out the current actual value.
It has to be pointed out that with deactivated periodical transmission and deactivated automatic transmission, no more actual-temperature telegrams will be transmitted".

Following the return of bus voltage, new programming via the ETS or a replugging of the user module, the object value will be updated according to the actual-temperature value and transmitted on the bus.
In case a temperature value telegram has not been received from the external sensor when using the external sensor with only one control circuit, only the value provided by the internal sensor will be transmitted. If only the external sensor is used, the value "0" will be in the object after a reset. For this reason, the external temperature sensor should always transmit the current value after a reset.

4.6 Disable functions of the room temperature regulator

4.6.1 Disabling controller

Certain operation conditions may require the deactivation of the room temperature control. For example, the control can be switched-off during the dew point mode of a cooling system or during maintenance work on the heating or cooling system.

The *"switch-off controller (dew point mode)"* parameter in the *"room temperature regulator functions"* parameter branch enables the *"controller disable"* object 40 when set to *"via object".* In addition, the controller disable function can be permanently deactivated when set to *"no"* (default).
In case a „1" telegram is received via the enabled disable object, the room temperature control of both control circuits will be completely deactivated. In this case all actuating variables = "0" and the " " symbol lights up on the display (wait for 30 sec actuating variable update interval!). The controller, however, can be operated in this case.

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

In case both control circuits are used, the second control circuit can be separately disabled. If a "1" telegram is received via the "2nd disable *control circuit"* disable object 41, the room temperature control of the second control circuit will be deactivated, the actuating variable of this circuit will be "0". In this case the first control circuit will continue to run.

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

4.6.2 Disabling controller operation

It is possible to disable the local control of the room temperature regulator (all keys associated with the room temperature regulator). An active disable of the control will be indicated by the " " symbol on the display. It has to be pointed out that this symbol is not exclusive and that it will also light up if the push button disable function is activated.

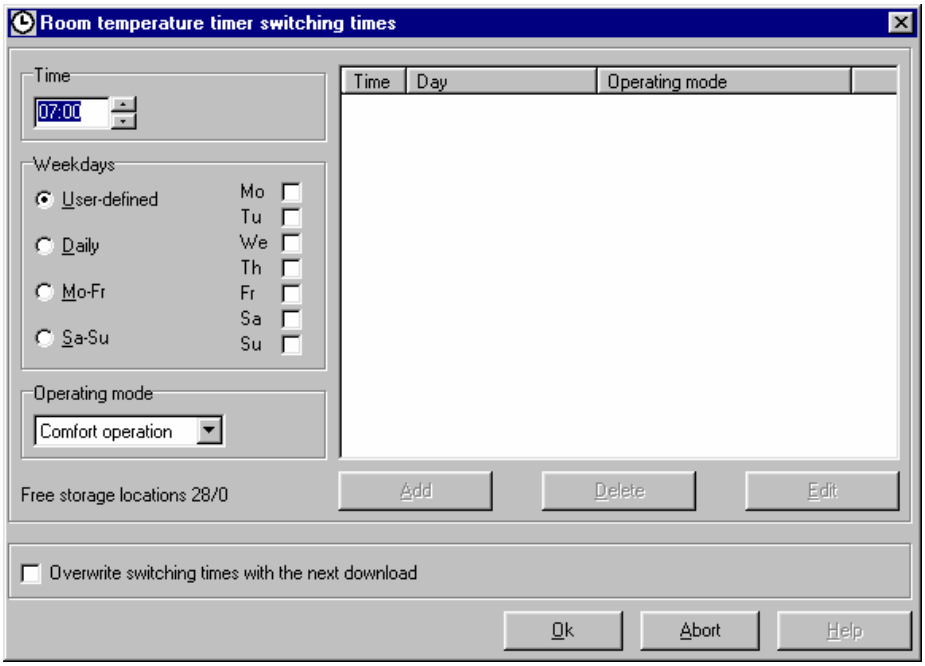
The "operation of controller inhibitable" parameter in the "room temperature functions" parameter branch can be used to determine whether the local control is never possible (setting: "always disabled" or whether it can be disabled via the *"controller operation disable"* object 39 (setting: *"via object"*).
The *"always disabled"* setting does not allow to parameterize the operating mode switch-over among the key or rocker functions. In addition, the two display keys for the shifting of the basic setpoint have no function with this setting.
When set to *"via object"* the local control will be deactivated, if a "1" telegram is received on the object. Hence, the local control will be activated again after receiving a „0" telegram. Actuating a key assigned to an operating mode switch-over or the display keys for the shifting of the basic setpoint will show no response during an active disable.

If the control is activated, it will neither influence the operation of the controller via the objects nor the room temperature control itself, i.e. the control algorithm is running and creating actuating variables and status reports.

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

4.7 Room temperature timer

The room temperature timer can differentiate up to 28 different switching programs and allows to switch-over the room temperature regulator's operating mode depending on the time and day of week. The room temperature timer must be enabled via the *"room temperature timer = on"* parameter in the "*room temperature regulator function – room temperature timer"* parameter branch. Alternatively, the function is permanently disabled via the *"room temperature timer = off"* setting (default). In case the push button functions as a controller extension, a room temperature timer cannot be projected.
The switching programs will be parameterized in the ETS plug-in and chronologically processed, if the timer function is enabled. The *"room temperature timer"* menu item in the *"timer editors"* menu will open the "room temperature timer switching times" editor:



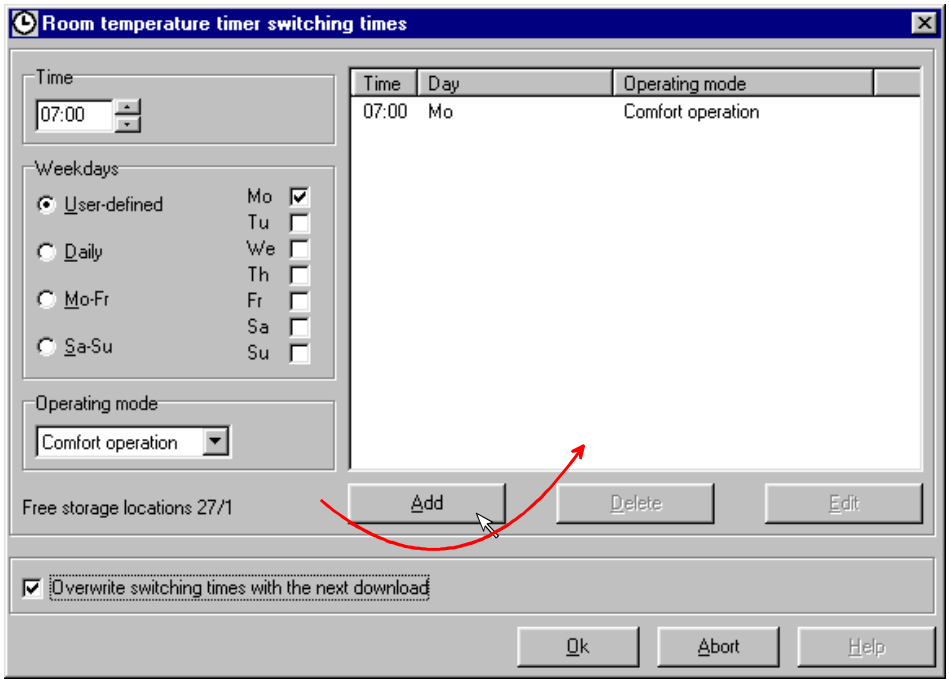
In the left part of the window the time is shown exact to the minute. In addition, the weekdays can be defined on which the switching time is to be executed. Available are the selections: "User-defined" (Mo, Tu, ..., Su), "Daily" (Mo – Su), "weekdays" (Mo – Fr) or "weekend" (Sa – Su).

Each switching program allocates a memory location in the push button RTR. Only one memory location is required for the "weekdays" or "weekend" settings. When set to "User-defined", however, sometimes up to 5 different switching programs are created depending on the parameterized days. The "free storage locations" info parameter in the editor indicates how many memory locations are available (number before the slash) or how many memory locations can be allocated for the selected setting (number behind the slash).

In addition, the operating mode that is to be activated when calling a switching programme has to be selected. The "comfort mode", "standby mode" or "night mode" modes can be selected.
It has to be pointed out that an operating mode set via the room temperature timer is on equal terms with a local control on the push button or via the switch-over objects (4 x 1-bit or 1-byte KONNEX switching object) and can be changed accordingly.

The switching times are projected by the minute. With an activated timer, checking the switching times takes place via the time control of the push button by the minute.
The internal clock of the push button should be set at least once an hour by an external time signal via the bus in order to keep the deviation as small as possible.
In rare cases it is possible that switching times will not be executed (skipped switching minute), if there are larger time differences between the tracked time in the push button and the one received via the bus. For this reason, the bus synchronization should not take place at predetermined switching times!

Once the switching programme (switching time, weekdays, operating mode) has been selected, the programme will be accepted into the programme list by clicking the "Add" button. This list is displayed on the right side of the editor window:



This way it is possible to pre-program up to 28 switching programmes. A programme created in the right part of the window may be edited by highlighting it and clicking the "Edit" button. Highlighting the programme and clicking the "Delete" button will delete the programme and remove it from the list. Clicking the "OK" button will accept the settings into the projection of the push button.

The temperature timer's switching programmes will be downloaded into the device when programming the complete application or partially programming the parameters, if the *"overwrite switching times with the next download"* check box in the editor window has been selected.

In addition to setting the timer programmes in the ETS plug-in, it is possible to edit the switching times, the weekdays and the operating modes even after a local programming operation on the push button in programming mode. For this, the "Complete operation" via the display keys must be enabled (depending on the parameter).
In order to avoid overwriting the locally changed data by an ETS programming operation, it is possible to deselect the *"overwrite switching times with the next download"* check box in the editor window.
Editing the switching programmes in programming mode is not possible, if the room temperature timer has been deactivated (cf. next page).

The room temperature timer may be activated or deactivated via local control in programming mode, if enabled, (cf."1.4 Programming mode / local control") and/or via a push button control (key function). The " ⏲ " symbol will light up on the display if the function is activated. The switching programmes will be chronologically processed according to the parameterized switching times.

Info: In case the room temperature is activated at exactly the same time as a parameterized switching time, the concerned switching programme will be executed later.

Once switching programmes have been programmed into the device, the room temperature timer is activated directly after the initialisation phase and the programmes are being processed. For this, a received valid time and weekday is required. If no programmes are stored in the device and the function itself enabled, the symbol will light up after a reset but no switching programmes will be executed.

A switch-over into another operating mode triggered by the room temperature timer can also momentarily be suppressed via a separate disable object. In order to enable the disable function, the *"lock room temperature timer via object"* parameter in the "*room temperature regulator function – room temperature timer"* parameter branch must be set to „yes". In this case the *"disabling room-temperature-timer"* disable object 55 will be enabled. The polarity can be parameterized.
If the room temperature timer is disabled via the bus, the " ⏲ " symbol will disappear.

During an active disable function, the operating mode will not be switched-over by the room temperature timer. In case the room temperature timer is enabled again at the exact time as a parameterized switching time, the concerned switching programme will be executed later.
Activations or deactivations of the room temperature timer during the disable phase will be memorized and tracked once the disable has ended.

Important notes on the room temperature timer:

- The room temperature timer will be automatically deactivated, if the frost/heat protection operating mode is activated! This will prevent an unwanted operating mode switch-over, for example, during a longer absence that accidentally heats up or cools down the room.
  A switch-over into the frost/heat protection can take place directly on the device (e.g. in programming mode or via push button control) or via the switch-over objects (4 x 1-bit or KONNEX). Activating the frost/heat protection via the window status or the frost protection automatic will not deactivate the room temperature timer!
  Even when setting the operating mode via the KONNEX forced-object the room temperature timer will be deactivated until enabled (value "0" / Auto).
  A room temperature timer automatically deactivated may be activated again anytime via push button control (room temperature timer control). An automatic re-activation will take place as soon as the frost/heat protection mode is terminated.
  Even a regular operating mode switch-over (comfort, standby, night) will re-activate the room temperature timer!
- The internal clock of the push button should be set at least once an hour by an external time signal via the bus in order to keep the deviation as small as possible. If the internal clock has not been updated at least once a day via the bus. (update check at 4:00 at night), the "--:--" symbol will appear on the push button's display, if the time of day is shown on the display (depends on the parameter). In this case, however, the internal clock continues to run with the expected deviation while the parameterized switching programmes of the room temperature timer will continue to be processed.
- The weekday information is provided by the time signal. The room temperature timer will process the programmed switching programmes only after receiving a valid weekday. In normal operation, the weekdays are displayed on the push button as numerical characters (1...7).
- Receiving a date is not required for the function of the room temperature timer.

4.8 Temperature alarm

The push button RTR is able to monitor two temperature values, if desired. When exceeding or falling short of these parameterizable values, switching telegrams, e.g. an alarm value, can be transmitted on the bus.

The temperature monitoring can be activated in the *"room temperature regulator function / room temperature measuring"* parameter branch via the *"send temperature alarm via object = yes".* Alternatively, the "no" setting (default) deactivates the temperature monitoring.
When this function is active objects 79 and 80, *"temperature alarm 1"* for the lower temperature value and *"temperature alarm 2"* for the upper temperature value will appear.

If the lower temperature value ($T_{actual} < T_{lower\ value}$) is exceeded, a "1" telegram will be transmitted on the bus via the "temperature alarm 1" object. As soon as the room temperature reaches or exceeds the lower limit ($T_{actual} \geq T_{lower\ value}$), a "0" telegram will be transmitted on the bus via the "temperature alarm 1" object.
If the upper temperature value ($T_{actual} > T_{upper\ value}$) is exceeded, a "1" telegram will be transmitted on the bus via the "temperature alarm 2" object. As soon as the room temperature reaches or exceeds the upper limit ($T_{actual} \leq T_{upper\ value}$), a "0" telegram will be transmitted on the bus via the "temperature alarm 2" object.

Every minute the temperature values are compared with the determined actual-temperature (room temperature). Hence, a temperature alarm telegram is transmitted no more than once a minute. In addition, the telegrams will be transmitted only if the switching value has changed.
It has to be pointed out that the temperature monitoring requires a valid actual-temperature value in order to operate. Before using an external temperature sensor a valid telegram must have been received.
The lower temperature value must be below the upper limit. In case the values are parameterized differently than described, the ETS plug-in will report an error and request to change the data.

Disabling the room temperature regulator (dew point operation) will not affect the temperature alarm.

4.9 Valve protection

A valve protection may be carried out periodically in order to prevent the addressed control valves of the heater or cooling system to become calcified or stuck. The "valve protection" parameter in the *"room temperature regulator function"* parameter branch will activate the valve protection by setting it to "yes".

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

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

Thus even long closed valves will be shortly opened on a regular basis.

The valve protection is controlled by the internal clock and activated for the relevant actuating variables at 8:00 o'clock. In case the clock has not been set following a reset, the valve protection will be carried out approx. 32 hours after the reset at the earliest.

4.10 Fan coil actuator stages display

Acting as a room temperature regulator, the B.IQ push button with room thermostat (RTR) and display can also trigger fan coil actuators via the variables. Normally, heating or cooling fans are connected to fan coil actuators. Such actors mainly convert continuous room temperature regulator variables into equivalent switching stages and can thus regulate the intensity of the fans connected. The bigger the variable is the more switching stages will be activated in the actuator.

Fan coil actuators are normally also capable of sending to the bus the current fan stage in the form of a feedback message. The display of the B.IQ push button with room thermostat (RTR) and display can show the fan stages of a fan coil actuator.

The display of the fan stages can be enabled by the *"use fan stages display"* parameter in the *"room temperature regulator function"* parameter branch (setting *"yes"*). When this function is enabled the associated communication objects will become visible.

The fan stages are displayed in three steps. Once a stage of the integrated fan coil actuator is activated, the " " fan symbol will be lit on the display of the B.IQ. push button RTR. At the first fan stage, only one wing of the symbol will be highlighted. At the second stage, two wings will be highlighted, while all three wings will appear at the third stage. Thus, you can recognize the fan intensity.
When a fan coil actuator is deactivated (no fan stage active) the fan symbol will be hidden.

You can configure the B.IQ push button with room thermostat (RTR) and display for two different feedback variants. The fan coil actuator either returns the stage feedback message via up to three different 1-bit communication objects, or it uses a 1-byte object as a combined feedback for all fan stages. In this connection, the *"fan stage display object type"* parameter defines the data type.

"3 x 1 bit" setting:
For each fan stage, an own 1-bit object is available. Once one fan stage is activated (object value "1"), the corresponding wing of the fan symbol will light up on the display. In this connection, the display of the highest fan stage will always get through. he fan symbol will go out when all objects are "0".

"Fan stage" objects			Active stage	Symbol
1 Obj. no. 54	2 Obj. no. 55	3 Obj. no. 56		
1	0	0	1	
X	1	0	2	
X	X	1	3	
0	0	0	---	hidden

"1-byte" setting:
The fan stages are triggered by a common 1-byte object. The value received will decide which stage is displayed by the B.IQ push button RTR.

"Fan stages 1-3" object Obj. no. 54	Active stage	Symbol
"01"	1	
"02"	2	
"03" or "04" … "255"	3	
"00"	---	hidden

At a reset of the push button module, the fan stage displayed will get lost. After initialization, the symbol will only reappear when an update of the fan stage objects (values > "0") takes place.

5. Controller extension function

5.1 General

Alternatively to the function as a room temperature regulator, the extension mode can be activated. This way the B.IQ push button with room thermostat (RTR) and display can address other push buttons that are parameterized as controllers. For example, it is possible to fully control the controller by changing the operating mode or the presence function or the setpoint shifting. In addition, the controller extension can indicate the status of the room temperature regulator on the display and the status-LED. So, the set or room temperature and the setpoint shifting on the extension can be displayed as well as the current operating mode and the message indicating whether the system is heating or cooling.

The controller extension function is switched-on via the *"controller extension function"* parameter in the *"push button xgang"* parameter branch. If the room temperature regulator function is activated, it will not be possible to parameterize the extension.
Several controller extensions may affect a main controller.

5.2 Push button functions of the controller extension

While the controller extension functions are activated, the push button's function keys can be parameterized in the "*push button function / general / [key-/rocker designation]"* to "*controller extension"*(cf. "3.3 Rocker functions") parameter branch. In this case the controller operating functions will be activated. The controller extension may be set for touch control as well as for rocker control.
The *"function"* parameter defines the control function, which is assigned to the key or the rocker. The following chapters describe the settable controller extension functions.

Depending on the controller extension the status-LED's for touch control can be parameterized (cf. following chapter). With rocker control the two status-LED's of the rocker (left and right) can be parameterized and controlled irrespectively of the rocker function. The "function of left / right status-LED" parameter in the "push button function / general / [key designation] / state of rocker x" parameter branch determines whether the LED is permanently switched on or off or whether it is controlled via the "rocker X [status left / right]" object. In the latter case the polarity of the status objects is definable.

5.2.1 "Normal operating mode switch-over" controller extension function

If these extension functions are set, the controller extension will be able to switch-over the operating mode of the room temperature regulator. This requires that the operating mode switch-over on the controller takes place via a communication object that conforms to KONNEX (1-byte switch-over – cf. 4.1.1 Operating mode switch-over). In this case the *"operating mode switch-over controller extension"* object in the *"controller extension function"* parameter branch must be connected with the main controller. In order to transmit a change of the operating mode on the main controller to the extensions as well, the "T" flag has to be set on the main controller's object!

The *"operating mode on pressing a push button"* parameter in the parameter branch of the function key defines the operating mode which will be transmitted on the bus when actuating a key and which should be activated on the controller. With touch control either one of the "comfort", "standby", "night" and "frost/heat protection" operating modes can be parameterized.
With rocker control the modes are basically changed one after another depending on the key pressed left or right.

The function of the status-LED can be parameterized. Touch control allows to select the *"display push button function active"* and *"display push button function inactive"* settings in addition to the *"always off"*, *"always on"* and *"operation indication"* default settings. Thus the status-LED can signal whether an operating mode recalled by the key is active or not.
Alternatively, the status-LED can be switched via a separate communication object. The polarity of the status object in the "push button function / general / [key designation] /state of rocker x" parameter branch can be parameterized.

In order for the LED status indication for the key function to work, the "*controller status indication of controller extension*" object in the "controller extension function" parameter branch must be connected to the main controller!

5.2.2 "Forced operating mode switch-over" controller extension function

If these extension functions are set, the controller extension will be able to switch-over the operating mode of the room temperature regulator via forced-control. This requires that the operating mode switch-over on the controller takes place via a forced communication object that conforms to KONNEX (1-byte switch-over – cf. 4.1.1 Operating mode switch-over). In this case the "*forced object operating mode controller extension*" object in the "controller extension function" parameter branch must be connected to the main controller. In order to transmit a change of the forced operating mode on the main controller to the extensions as well, the "T" flag has to be set on the main controller's object!

The "operating mode *on pressing a push button* " parameter in the parameter branch of the function key defines the operating mode which will be transmitted on the bus when actuating a key and which should be activated on the controller via forced control. With touch control either one of the "comfort", "standby", "night" and "frost/heat protection" operating modes can be parameterized. The "auto" setting will deactivate the forced setting.
With rocker control the modes are basically changed one after another depending on the key pressed left or right.

The function of the status-LED can be parameterized. Touch control allows to select the "display key function active" and "display key function inactive" settings in addition to the "always off", "always on" and "operation indication" default settings. Thus the status-LED can signal whether an operating mode called by the key is active or not. Only the „auto" mode cannot be indicated by a status-LED. The LED display can also not distinguish whether the operating mode has been set via a forced object or via the 'normal' operating switch-over'.
Alternatively, the status-LED can be switched via a separate communication object. The polarity of the status object in the "push button function / general / [key designation] / state of rocker x" parameter branch can be parameterized.

In order for the LED status indication for the key function to work, the "controller status indication of controller extension " object in the "controller extension function" parameter branch must be connected to the main controller!

5.2.3 "Presence button" controller extension function

This extension function causes the transmission of a presence telegram when pressing a key and thus the activation of the presence mode on the controller (cf. 4.1.2 Notes on the operating modes – presence function/comfort mode extension). This requires that the presence function is activated on the controller *("type of presence detection = presence button"*). In this case the "*presence object* controller extension" object in the "controller extension function" parameter branch must be connected to the "*presence object" object of* the main controller.

When actuating a function key, which is parameterized to this function, the presence function will be activated or deactivated. Actuating the rocker will also always result in a switch-over of the presence status regardless whether the left or right key was actuated.

The function of the status-LED can be parameterized. Touch control allows to select the "display key function active" and "display key function inactive" settings in addition to the "always off", "always on" and "operation indication" default settings. Thus the status-LED can signal whether an operating mode called by the key is active or not.
Alternatively, the status-LED can be switched via a separate communication object. The polarity of the status object in the "push button function / general / [key designation] / state of rocker x " parameter branch can be parameterized.

In order for the LED status indication for the key function to work, the "*presence object controller extension*" object in the "controller extension function" parameter branch must be connected to the main controller!

5.2.4 "Setpoint shifting" controller extension function

This extension function allows to shift the basic setpoint for the temperature on the room temperature regulator via a function key. The control on the extension is the same as a control on the main controller (cf. "4.4.2.2 Basic setpoint shifting").

A function key parameterized for a setpoint shifting will decrease or increase the value of the setpoint shifting when pressing a key. The direction of the value adjustment is set via the "command on pressing the push button" parameter in the "push button function / general / [key designation]" parameter branch.

A long key-press will continue the adjustment. An adjustment takes place every 0.5 seconds.

The function of the status-LED can be parameterized. With touch control the *"always off"*, *"always on"* and *"operation indication"* settings can be selected. Alternatively, the status-LED can be switched via a separate communication object. The polarity of the status object in the "push button function / general / [key designation] /state of push button" parameter branch can be parameterized.

Communication objects for the basic setpoint shifting:

In order to be able to use the setpoint shifting on the extension, the communication objects in the *"controller extension function"* parameter branch must be connected to the objects of the controller or other extensions as follows:

Controller extension objects		Controller objects
"Current setpoint shifting controller extension" object 52	→	"Current setpoint shifting" object 52
"Preset setpoint shifting controller extension" object 53	→	"Preset setpoint shifting" object 53

The setpoint shifting currently adjusted on the controller is transmitted to the extensions. For this the controller updates the shifting in the *"current setpoint shifting" communication* object 52 with a 1-byte counter value (cf. "4.4.2.2 Basic setpoint shifting"). By connecting to this object the controller extensions are able to determine the position of the setpoint shifting and to display the current setpoint shifting.

In addition, the controller's setpoint shifting can be externally adjusted via the communication object 53 ("preset setpoint shifting"). This object has the same data point types and range of values as the object 52 (see above). By connecting to the object 53 the controller extensions are able to directly adjust the current setpoint shifting of the controller.

Starting from the current setpoint shifting (object 52), each key-press on an extension will adjust the setpoint in the corresponding direction by one counter value step. Each time the setpoint is adjusted a new shifting is transmitted to the room temperature regulator via the output object of the controller extension (object 53). The controller itself checks the received value for the minimum and maximum limits and adjusts the new setpoint shifting and accepts the new counter value into the output object (object 52).

Due to the standard data point type used as the output and input object of the controller extension and the weighting of the individual stage by the controller itself, each extension is able to determine whether a shifting took place, in which direction it took place and by how many steps the setpoint was shifted. This requires that the communication objects are connected on all controller extensions and the controller. The information for the step value as feedback from the controller enables the extension to continue the adjustment anytime at the right point. The extensions may also respond to a reset of the setpoint shifting by the controller and correctly display the current adjustment.

5.3 Display functions of the controller extension

The controller extension can display the status of the room temperature regulator. So, the set or room temperature and the setpoint shifting on the extension can be displayed as well as the current operating mode and the message whether the system is heating or cooling.
Certain communication objects must be connected to the main controller for the controller extension to display this information. The following shows the individual display information and the corresponding objects…

Display of the operating mode:

The extension can display the operating mode currently set on the controller. Just like on the controller itself the mode is indicated by the " ⌂ " (comfort), " ⌂ " (standby), " ☾ " (night) and " ❄ " (frost/heat protection) symbols.
This display information is provided by the *"controller status"* communication object 36 that absolutely needs to be connected to the *"controller status"* object of the main controller (1-byte!). It is not possible to use the display information to distinguish whether the operating mode has been set via a forced object or via the 'normal' operating switch-over in case of a KOONEX switch-over.
In addition, a comfort mode extension of the controller is not indicated by the " ⌂☾ " or " ⌂❄ " symbols. In this case only the comfort mode symbol is displayed.

The operating mode may also be switched-over via the push button functions of a controller extension (cf. "5.2 Push button functions of the controller extension"). A switch-over via the programming menu is not possible on the extension.

Display of a setpoint shifting:

The extension can indicate whether a basic setpoint shifting has been adjusted on the controller. If a basic setpoint shifting is active, the extensions will show the " ✋ " hand symbol on the display. This requires that the *"current setpoint shifting controller extension"* communication object 52 is connected to the *"current setpoint shifting"* object 52 of the main controller.

A basic setpoint shifting may also be set via the push button functions of a controller extension (cf. "5.2 Setpoint shifting' controller extension function").

Display of set-temperature:

The controller extension can display the set temperature of the room temperature regulator. If this display is used, the *"set temperature controller extension"* communication object 50 must be linked to the *"set temperature"* object 50 of the main controller. In addition, the display of the extension must be configured for the display of the setpoint. This requires the *"display of"* parameter in the *"Display" parameter* branch to be parameterized for a setting that allows the display of the set-temperature. At the controller extension, the setpoint temperature will always be displayed as an "absolute" temperature.

Display of room temperature and room temperature measurement:

The display of the controller extension may also indicate the currently determined room temperature (actual-temperature). This requires the display of the extension to be configured for the display of room temperature. The "display of" parameter in the "Display" parameter branch must be parameterized to a setting that allows the display of the room temperature.
The extension measures the room temperature independently in the same way as the controller. Thus the controller extension can determine the actual-temperature via its own internal sensor or additionally via another external sensor. It is also possible to evaluate and transmit a temperature alarm. The parameters required for temperature measurement can be found in the "room temperature measuring" parameter branch (cf. "4.5 Room temperature measurement" and following).
Normally, the room temperature determined on the extension will also be transmitted to the main controller (as an external sensor) in order to be integrated there into the room temperature regulator. If such integration takes place, the "actual-temperature" communication object 23 of the controller extension must be connected to the *"external temperature sensor"* object 24. The external temperature sensor has to be enabled on the main controller.

Display of the "heating" and "cooling" messages:

If actuating variables are greater than "0" the main controller will show on the display that heating and cooling energy is requested. This is indicated by the " + " symbol for heating or by the " - " symbol for cooling.
A controller extension may also show this information on the display. In heating mode or mixed-mode the *"indication heating"* communication object 37 must be connected to the *"indication heating"* object 37 of the main controller. In cooling mode or mixed-mode the *"indication cooling"* communication object 38 must be connected to the *"message cooling"* object 38 of the main controller.
A simultaneous heating or cooling can be displayed on the controller extension. Therefore, the objects of the extension may not be active at the same time "1"!

Fan coil actuator stages display

Same as with the function as a room temperature regulator, a controller extension can display the fan stages of a fan coil actuator. Compared with the controller function, there is no difference in the parameterization or activation of the fan symbol. The fan coil display is described in detail in chapter "4.10 Fan coil actuator stages display".

5.4 Overview of the communication objects

The following shows and explains all communication objects of the controller extension in the *"controller extension function"* parameter branch…

Object			Summarized description
	23	Actual-temperature	For the transmission of the room temperature measured on the main controller.
	24	External temperature sensor	For the connection of an additional external temperature sensor on the extension.
	28	Operating mode switch-over controller extension:	For the switch-over and transmission of the operating mode to the main controller.
	32	Forced-object operating mode controller extension	For the switch-over and transmission of the forced-operating mode to the main controller.
	33	Presence object controller extension	For the switch-over and transmission of the presence status to the main controller. Also for activating the status-LED of a presence button.
	36	Controller status indication of controller extension	For the display of different symbols. Also for addressing the status-LED of a function key to switch-over the operating mode.
	37	Message heating controller extension	For the display of the heating symbol.
	38	Message cooling controller extension	For the display of the cooling symbol.
	50	Set temperature controller extension	For showing the set-temperature of the main controller on the display of the extension.
	52	Current setpoint shifting controller extension	For receiving the counter value to adjust the setpoint of the main controller.
	53	Setting setpoint shifting controller extension	For setting a new counter value to adjust the setpoint for the main controller.
	79	Temperature alarm 1	For transmitting a temperature alarm on the bus.
	80	Temperature alarm 2	For transmitting a temperature alarm on the bus.

5.5 Behaviour of the controller extension after a reset

The different display and control functions of the controller extension are controlled via different communication objects as described in the previous chapters. A main controller must transmit the current status to the extensions, i.e. updating the communication objects so that after a programming operation or after the return of bus voltage all status information is available for the initialisation of the extension. This takes place automatically during the initialisation of the main controller.
For this reason all extensions should be put into operation first. Only then the main controller should be connected and programmed.
For larger KNX/EIB installations where the extensions are sometimes distributed over several lines, the remaining lines should also be initialized after a reset of one line.

Alternatively, the extensions may actively request the current object values from the bus during an initialization. The *"State"* flag in the parameter branch of the object can be set to *"on"* for the communication objects of the controller extension. In this case the extension will separately request the object values via a value-read telegram (ValueRead) during the initialization. The main controller responds to this read request via a value response thus updating the object of the extension in the process. It is important to set the "*Read*" flag on the corresponding objects of the main controller!
If several extensions are used, the status request should only be activated on one extension. The remaining controller extensions will be updated by a response of the main controller as long as they are connected by the same group address.

6. Timers

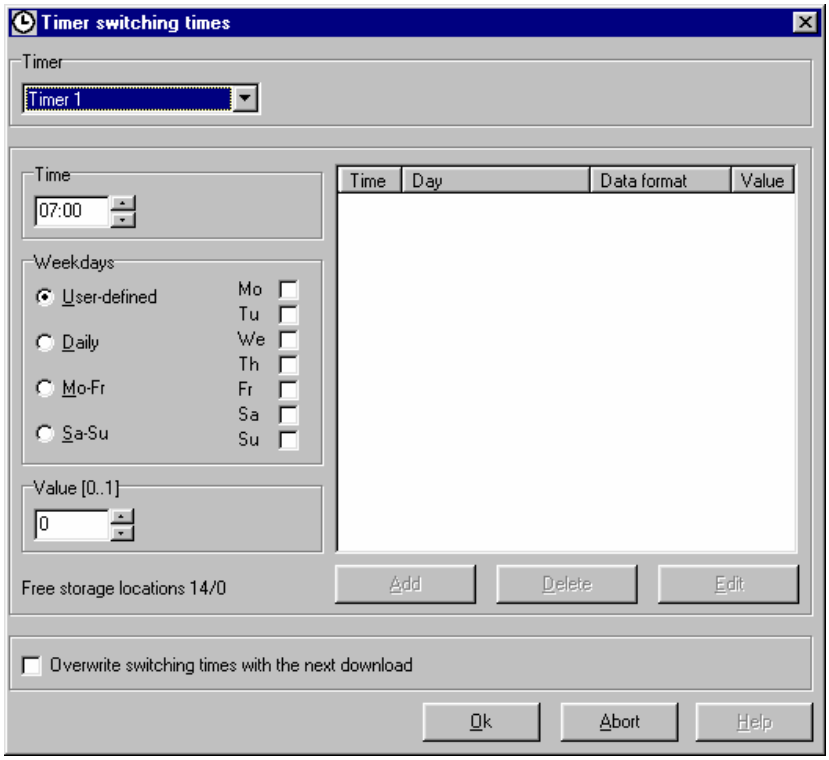
6.1 Function

The B.IQ push button with room thermostat (RTR) and display has up to two separate timers. These functions allow time-controlled transmission of switching commands (ON/OFF), value telegrams (0…255) or light scene recall telegrams (1…8) to the bus through separate objects, depending on the data type.

You can use such commands, among other things, for triggering move or position objects of blinds or roller shutters. However, any other bus control options are also possible by such commands.

You can use the *"timer 1"* or *"timer 2"* parameters in the *"B.IQ RTR push button"* parameter branch to separately enable the timers. When the functions are in the deactivated state (default) the parameters and the objects of the timers will be hidden.

As soon as one of the two timers has been enabled, the *"timer"* menu item will be activated in the *"timer editor"* main menu. When you select this menu item the "timer switching times" window will appear. In dependence on the parameterization of the two timers, you can parameterize from this window each of the up to 14 switching times down to the minute, together with the associated switching command (value or scene), depending on the parameterized data type:



In the left part of the window, you can define the time of the switching program down to the minute. Furthermore, you can define the weekdays on which the switching time event is to take place. You can choose from the "user-defined" (Mo, Tu, …, Su), "daily" (Mo - Su), "Mo – Fr" or "Sa – Su" options.

Each switching program will occupy one memory location in the B.IQ RTR push button. For the "daily", "Mo – Fr" or "Sa – Su" 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 storage 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).

After you have set the switching time the weekday and the control command, you can click on the "add" button to create the switching program. You can create the maximum of 14 switching programs per timer from the program list in the right part 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 button.

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

By local operation in the programming mode, you can edit the switching times of both timers. A precondition to do so is that "complete operation" through the display buttons on the push button has been enabled (depending on the parameter). Thus, you can subsequently change the times or commands programmed through 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 "timer 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, value or scene commands. The locally set switching times will thus remain unchanged.

Notes on the timers:

- The switching times can be configured down to the minute. The switching times are checked by the push button time control also at minute intervals when the timer is active.
  The internal clock of the push button should be set by an external time control signal at least every hour to keep the time error as low as possible.
  In exceptional cases, there may be major differences between the time followed up in the push button and the time received from the bus so that switching times events will not be executed (skipped switching minute). For this reason, synchronization from the bus should not take place at preset switching times.
  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 button will read "--:--" if the time is indicated on 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 several switching times have been parameterized to the same time on the same day only the command of that switching time, whose switching time number is of higher order, will be transmitted to the bus.
- The timers will only execute the given switching programs after they have received a valid time.
- Receiving a date is not required for the function of the timers.

6.2 Activating and deactivating the timers

If enabled, you can separately activate or deactivate the timers by local operation in the programming mode (refer to "1.4 Programming mode/local operation") and/or by push button actuation (push button function). When this function has been activated the " ◷ " symbol will light up on the display, and the switching programs will be executed in chronological order in accordance with the parameterized switching times. Please note that this " ◷ " symbol is not exclusive and will also light up when a room temperature timer has been activated.
After switching programs have been fed into the device the timers will be activated immediately after the initializing phase, and the programs will be executed. A valid time and a valid weekday 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.

In addition, you can temporarily suppress the transmission of a bus command by the timers via separate disabling objects. To facilitate such disabling function set the *"blocking object"* parameter in the *"timer X"* (X = 1 or 2) parameter branch to *"yes"*. In this case, disabling object 60, *"disabling timer 1"*, or disabling object 62, *"disabling timer 2"* will be enabled. You can parameterize their polarities.
When the timer has been disabled through the bus, the " ◷ " symbol will disappear from the display. The symbol will not go out if other timers have been activated (timer 1 or 2, or room temperature timer).
During the period of an active disabling function, no commands will be sent to the bus. If you re-enable the room temperature timer exactly at the moment of a parameterized switching time the switching program concerned will not be executed.
Activation or deactivation of the timer during the disabling phase will be stored and effected after the end of disabling.

7. Scene function

7.1 Scene definition

Similar to a light scene push button, the B.IQ push button with room thermostat (RTR) and display has a scene function. Under this function, you can save in the push button up to eight different scenarios. Each scene can trigger up to eight bus outputs (scene objects). You can configure switching, value or Shutter position commands.

You can generally use the *"scene function"* parameter in the *"B.IQ RTR push button"* 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 position (1 byte)	0...100 % Position "0" = top

*: 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 to the bus via the output objects. For each scene output, you can parameterize whether a command should be sent at all when a scene is being recalled. The *"transmit output signal = 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 button so that they will not get lost after a bus voltage failure.

7.2 Scene recall/scene saving

You can recall a scene by:

- the scene extension object (object 72):
  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 buttons, display panels or complex scene control systems.
- a local push button function on the push button sensor:
  In addition, you can recall a stored scene by locally actuating a button on the push button. If you have parameterized the push button function to *"light scene extension/recall"*, and if you want the button to act as *"internal scene request"*, you can recall the scenes stored in the B.IQ push button with room thermostat (RTR) and display by a short push button actuation (< 1 s). For this purpose, you must specify the corresponding scene number (1 to 8) in the ETS plug-in (refer to "3. Push button Functions", page 44).
  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 B.IQ RTR push button. You can save a scene by:

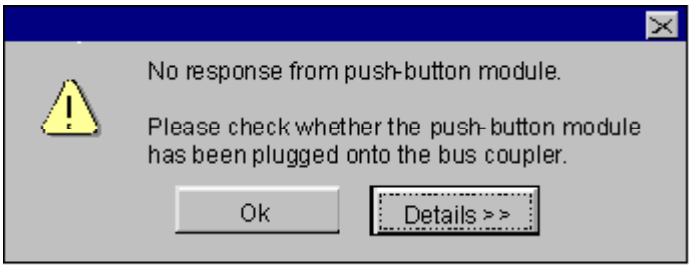
- the scene extension object (object 72):
  Via the extension object, a storage telegram is received. According to the scene number, the scene control of the B.IQ push button with room thermostat (RTR) and display will request the current values of the scene objects from the actuators via the bus and save them permanently.
- a local button function on the push button:
  When parameterizing an *"internal scene request"* with the memory function enabled, you can actuate the button 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 B.IQ push button with room thermostat (RTR) and display 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 B.IQ push button with room thermostat (RTR) and display 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.
In general, switching commands, value commands or Shutter positions can be saved anew.

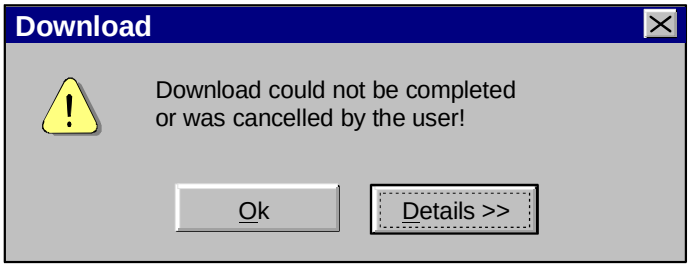
To enable the actuators on the bus to respond to the read request from the B.IQ push button with room thermostat (RTR) and display set the read flag ("R" flag) on the actuator objects concerned.
In order to avoid communication problems when recalling or saving scenes do not change the communication flags ("C" flags) of the scene objects on the B.IQ RTR push button.

8. Messages in a programming process

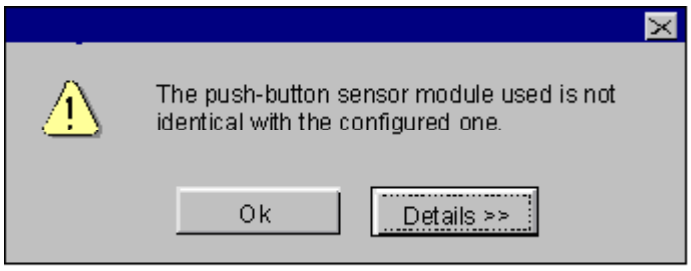
After you have programmed the B.IQ push button with room thermostat (RTR) and display with the aid of the ETS plug-in, you can program it in the ETS start-up environment.
During a programming process, the following messages can appear:



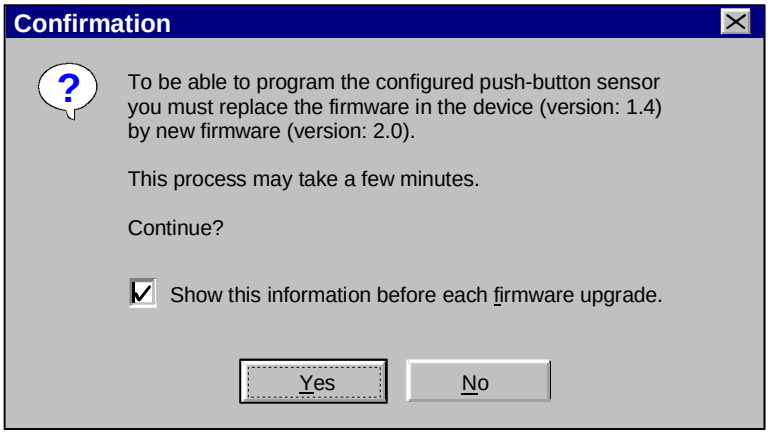
Reason: Attempt to load the application data into the device.
Cause: No B.IQ push button with room thermostat (RTR) and display has been plugged onto the bus coupling unit.
Remedy: Plug the B.IQ push button with room thermostat (RTR) and display onto the bus coupling unit. When doing so, keep in mind 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 button. Even for application data partial programming, you must have plugged on the push button.



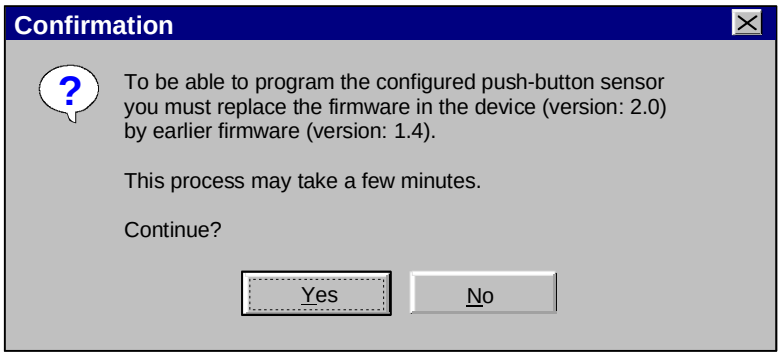
Reason: Attempt to load the application data into the device.
Cause: The programming process was cancelled through the "cancel" button, or there was a communication error.
Remedy: Start a new programming process.
Note: During a programming process, especially when the firmware is being programmed, larger data quantities are sent to the device via the bus. In this connection, the intelligent programming algorithm of the B.IQ push button with room thermostat (RTR) and display can recognize communication errors itself and re-transfer the erroneous data. 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 B.IQ RTR push button, 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.



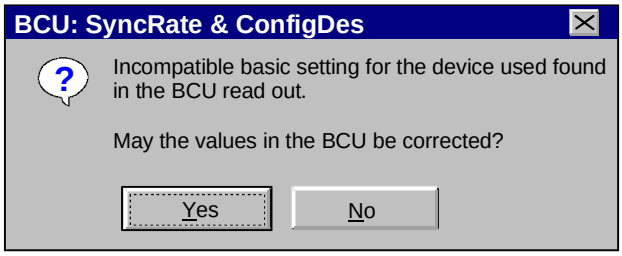
Reason: Attempt to load the application data into the device.
Cause: A B.IQ push button with room thermostat (RTR) and display variant other than the configured one has been plugged onto the bus coupling unit (e. g. 5gang-type configured and 2gang-type plugged on).
Remedy: Plug on the variant which conforms with the configuration.



Reason: Attempt to load the application data into the device.
Cause: A B.IQ push button with room thermostat (RTR) and display containing earlier firmware (e. g. V 1.4) is being programmed with some later B.IQ push button with room thermostat (RTR) and display software version (e. g. V 2.0).
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 downwards compatible.
Note: If you uncheck the "show this information before each firmware upgrade" box this message will not appear again, even though you are programming further earlier B.IQ push button with room thermostat (RTR) and display versions.
You can re-check this box later in the ETS plug-in of the B.IQ push button with room thermostat (RTR) and display from the "settings - options" menu on the "hardware" tab.



Reason: Attempt to load the application data into the device.
Cause: You want to program a B.IQ push button with room thermostat (RTR) and display containing new firmware (e. g. V 2.0). In this case, the version in the device is later than that specified by the B.IQ push button with room thermostat (RTR) and display software used (e. g. V 1.4).
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 upwards compatible. In such case, you should reinstall current B.IQ push button with room thermostat (RTR) and display software.
Depending on the changes resulting therefrom, it can possibly become necessary to configure a new device in the ETS.



Reason: Attempt to load the application data into the device.
Cause: The B.IQ push button with room thermostat (RTR) and display has been plugged onto a bus coupling unit which does not go with the configuration of the push button. 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 B.IQ push button with room thermostat (RTR) and display 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 button.

Parameters		
Description:	**Values:**	**Comment:**
B.IQ RTR push button		
Push button functions	Disabled **Enabled**	The parameter determines whether the push button function is switched-on or switched-off.
Room temperature regulator function	Disabled **Enabled**	The parameter determines whether the room temperature regulator function is switched-on or switched-off.
Controller extension function	**Disabled** Enabled	The parameter determines whether the controller extension function is switched-on or switched-off. Only if room temperature regulator function is switched-off!
Scene function	Disabled **Enabled**	The parameter determines whether the scene function is switched-on or switched-off.
Timer 1	**Disabled** Enabled	This parameter specifies whether timer 1 is activated or deactivated.
Timer 2	**Disabled** Enabled	This parameter specifies whether timer 2 is activated or deactivated.
Alarm function after pulling off the application module	**Disabled** Enabled	When the B.IQ push button with room thermostat (RTR) and display 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.
Alarm object data format	**Switching telegram, 1 bit** Value telegram, 1 byte Value telegram, 2-byte	To specify the data format of the alarm message.
Telegram after removal	OFF telegram **ON telegram**	To specify the value of the switching telegram sent when an alarm message is being raised. Only for data format = "switching telegr., 1 bit".
Value after removal	0 to 255, **255** (for "reset value = no") 1 to 255, **255** (for "reset value = yes")	To specify the value of the value telegram sent when an alarm message is being raised. Only for data format = "value telegram, 1 byte".
Value after removal	0 to 65535, **65535** (for "reset value = no") 1 to 65535, **65535** (for "reset value = yes")	To specify the value of the value telegram sent when an alarm message is being raised. Only for data format = "value telegram, 2-byte".
Reset value	**No** Yes	To specify whether the alarm value should be automatically reset to its inverse value ("0", no alarm) after the application module has been re-plugged.
Light duration of status LED at operation indication	1 s 2 s **3 s**	To define the period for which the status-LED will be lit as operation indication.

Push button assistance function	**Disabled** Enabled	To specify whether the push button assistance function is enabled. When the function has been enabled and a button is actuated (except the display buttons) the display will show a help text which can describe the assigned button function. The button help text can be specified in the plug-in separately for each push button or rocker.
Push button assistance function		To define the push button assistance functioning.
	Key-controlled	To execute the push button assistance function press the corresponding function key twice. The first key actuation will only read the push button assistance text on the display.
	Time-controlled	To execute the push button assistance function press the corresponding function key for a defined period. If you press the key for a shorter time the display will only read the push button assistance text, without the key function being executed. Only for "push button assistance function" = "ON".
Time between display and function triggering (0…5) * 1 sec.	0…5; **0**	To define the period a function key must remain to be pressed so that its function is executed. If a key is pressed for a shorter time the display will only read the push button assistance text. Setting *"0"* will cause the execution of the key function immediately upon its actuation, in addition to the display of the assistance text. Only for "push button assistance function" = "ON" and "push button assistance function" = "time-controlled"

Operation via display buttons		The B.IQ push button with room thermostat (RTR) and display has several local operator levels:
	No operation	'Normal operation' and local operation of the regulator by actuation of the display buttons for shifting the basic set value possible.
	Limited operation	Switching over into the programming mode is possible. → 'Normal operation' including set value shifting and switching over the operating mode and resetting of the different heating and/or cooling set values is possible.
	Complete operation	Full access to the device for local operation. In addition to limited operation, it grants the user access to the up to three timers (activation/deactivation of the timers and changing of the individual switching programs) and to the "settings" menu (activation/deactivation of the push button assistance function, the LCD background light and key disabling as well as of the display of the version data). It only grants the user access to the two timers (activation/deactivation of the timers and changing of the individual switching programs) and to the "settings" menu (activation/deactivation of the push button assistance function, the LCD background light and key disabling as well as of the display of the version data).
Reset text message		Upon receipt of a 14-byte alarm text message, the text received will be read on the display of the push button module. The text must be reset to enable the display to return to normal. This parameter defines how a 14-byte alarm text message can be reset.
	Via key actuation	The alarm text message can be acknowledged and reset by pressing any key.
	Via object	An alarm text message can solely be reset via separate communication object 74, *"reset text/alarm message"*.
	Via key actuation and object	An alarm text message can be reset either by pressing any key on the push button module, or via separate communication object 74, *"reset text/alarm message"*.
Polarity	Inverted (reset = 0)	To define the polarity of communication object 74, *"reset text/alarm message"*.
	Not inverted (reset = 1)	Only for "reset text message" = "via object" or "via key actuation and object".

Parameters		
Description:	**Values:**	**Comment:**
Display		
Illumination		The triggering of the display background light and of the device ON LED (blue) can be defined.
	ON	To permanently switch on the light.
	Automatic switch-off	The light will switch on upon a button actuation and will be automatically disabled after the preset time has elapsed.
	Switching via object	The light is triggered via object 22.
Kind of switching	**ON**	When the light is switched via the object it can be specified whether the light should remain permanently enabled ("ON"), or whether it should automatically go out after the parameterized time has elapsed.
	Automatic switch-off	Only for "illumination = switching via object".
Automatic illumination switch-off (1…1200) * 1 s	0 s to 1200 s, **10 s**	To specify the ON period of the display background light and of the device ON LED (blue). The time can be re-triggered.
		Only for " illumination = automatic switch-off" or "kind of switching = automatic switch-off".
Setpoint shifting display		This parameter defines the type of reading of the basic setpoint shift and of the setpoint temperature in the push button module display.
	Absolute display	The absolute reading shows the currently adjusted setpoint temperature of the active operating mode. For a new setpoint shift, the new temperature value will be calculated by the push button module and fully read on the display.
	Relative display	The relative display will only show the current setpoint shifting in ℃ (offset) without also reading the setpoint temperature derived thereof.
Display of	Time / room temperature Time / outside temperature Time / light scene Outside temperature **Room temperature** Setpoint temperature Setpoint temperature / text Date / time Outside / room temperature Date / time / outside temperature Date / time / room temperature Value display	Various kinds of information can be read by the display. You can select them from here.

Data type	**Switching** Value (1-byte) Value (2-byte)	To define the data type for the value display. Depending on the setting, the communication object and further parameters can be created. Only for "display of = value display".
Descriptive text	[20-character text] (Default text, depending on data type.)	For value display, a descriptive text with a maximum length of 20 characters can be entered. This text will be shown in the upper display line. The actual value will be in the line below. Only for "display of = value display".
Text for '0' telegram	[10-character text]; **OFF**	For the "switching" data type, it can be defined here which text is to appear on the display when a "0" telegram was received through the *"value display"* object. Only for "display of = value display" and "data type" = "switching".
Text for '1' telegram	[10-character text]; **ON**	For the "switching" data type, it can be defined here which text is to appear on the display when a "1" telegram was received through the *"value display"* object. Only for "display of = value display" and "data type = switching".
Sign	**No** Yes	For the "value" data type, sign evaluation and thus the KNX data point can be set. If "yes" is selected the received value will be evaluated and displayed with its sign via the *"value display"* object. "No" will cause value display without sign. Only for "display of = value display" and "data type = value (1-byte)". The value will also be visible under the condition of "data type = value (2-byte)" and "number format = integer".
Display format	0…255 **0…100** 0…360	To match the display format to the KNX data point type for 1-byte value display without sign. Only for "display of = value display" and "data type = value (1-byte)" and "sign = no".
Unit	**No** Yes	In addition, a unit for the display value can be firmly specified for any display format. This parameter will enable the display (setting *"yes"*). Only for "display of = value display" and "data type" = "switching".

Unit text	[10-character text] (Default text, depending on data type.)	When the display is enabled, you can parameterize in the ETS plug-in the *"unit text"* with a maximum of 10 characters. This unit text will be shown on the display of the push button module, directly behind the display value. Only for "display of = value display" and "data type = value" and "unit = yes".
Number format	Integer **Floating-point**	To match number format to the KNX data point type for 2-byte value display. Only for "display of = value display" and "data type = value (2-byte)".
Display format	No digit after decimal point **1 digit after decimal point** 2 digits after decimal point	To specify the number of digits displayed after the decimal point for 2-byte floating-point value display. Only for "display of = value display" and "data type = value (2-byte)" and "number format = floating point".
Text display	**No** Yes	To activate the 3 x 1 bit text displays. When this function is enabled the associated parameters and communication objects will become visible.
Text 1	[20-character text]; **Text 1**	To predefine display text 1. Only for "text display = yes".
Text 2	[20-character text]; **Text 2**	To predefine display text 2. Only for "text display = yes".
Text 3	[20-character text]; **Text 3**	To predefine display text 3. Only for "text display = yes".
Reset text display		Upon receipt of a text message, the text received will be read on the display of the push button module. The text must be reset to enable the display to return to normal. This parameter defines how a called text message can be reset.
	Via separate object	In addition to the three objects for calling the predefined text messages, another 1-bit communication object for resetting all predefined text messages is available. Once a telegram that conforms with the parameterized polarity is received via object 78, *"text 1-3 display/reset display"*, the push button module will hide all predefined text displays.
	Via display objects	The predefined texts will be called separately via correspondingly associated objects *"text 1 display"*, *"text 2 display"* or *"text 3 display"*. Resetting proceeds through the same objects with inverted polarity. Hence, the display of

	Time-controlled	predefined texts 1-3 will only be entirely hidden when all three calling objects have the corresponding polarity value for resetting. The predefined display texts will be reset altogether once the parameterized display period has elapsed. This period will be restarted with each call of a text. Only for "text display = yes".
Display period	1 min 2 min **3 min** 4 min 5 min 7 min 10 min	To set the text display period for time-controlled display. Only for "text display = yes" and "reset text display = time-controlled".
Display object polarity	Inverted (display = 0) **Not inverted (display = 1)**	To parameterize the polarity of the display objects if the text display is not to be reset through the display objects. Only for "text display = yes" and "reset text display = via separate object or time-controlled".
Display object polarity	Inverted (display = 0; reset = 1) **Not inverted (display = 1; reset = 0)**	To parameterize the polarity of the display objects if the text display is to be reset through the display objects. Only for "text display = yes" and "reset text display = via display objects".
Separate object polarity	Inverted (reset = 0) **Not inverted (reset = 1)**	To parameterize the polarity of the separate communication objects for resetting the text display. Only for "text display = yes" and "reset text display = via separate object".
Text display priority	**Lower than text message** Higher than text message	In addition to calling the predefined texts via 3 x 1 bit, the B.IQ push button with room thermostat (RTR) and display can display an alarm text message. The display priority can be set by this parameter if the alarm text display is additionally used. When being received, an alarm text message can overwrite a predefined text message. When being called, a predefined text message can overwrite an alarm text message. Once a higher-priority text display is reset, i. e. hidden, the push button module will show the lowest-priority message if the same is still active. Only for "text display = yes".

Parameters		
Description:	**Values:**	**Comment:**
Push button function - disable		
Disabling function	**Push button not disabled** Function of all rockers like rocker 1…n Single rocker disabled Push button disabled	This parameter defines the behaviour of the push button when the disabling function is active. The disabling function is deactivated. When the disabling function is active all rockers of the B.IQ push button with room thermostat (RTR) and display will behave in the same way as the parameterized one. When the disabling function is active specific individual rockers of the B.IQ push button with room thermostat (RTR) and display can be disabled. When the disabling function is active the entire push button will be disabled.
Polarity of blocking object	Inverted (disabling = 0) **Not inverted (disabling = 1)**	To specify the disabling object polarity.
Function like rocker	**Rocker 1** (3-, 4- and 5gang) Rocker 2 (3-, 4- and 5gang) Rocker 3 (3-, 4- and 5gang) Rocker 4 (4- and 5gang) Rocker 5 (5gang)	When the disabling function is active all rockers of the B.IQ push button with room thermostat (RTR) and display will behave in the same way as the parameterized one. Only for disabling function = "function of all rockers like rocker 1…n".
Rocker x disabled? X = 1 to 3 (3gang) X = 1 to 4 (4gang) X = 1 to 5 (5gang)	**No** Yes	To specify whether rocker X will be disabled when the disabling function is active, i. e. button actuation (left and right) of this rocker will show no function. Only for disabling function = "single rocker disabled".

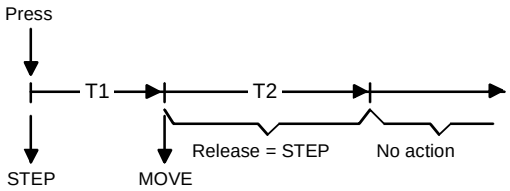
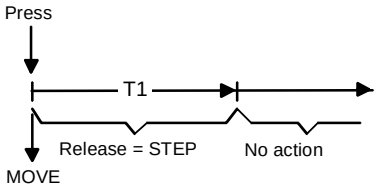
Parameters		
Description:	**Values:**	**Comment:**
Push button function - general		
Rocker X: concept of operation X = 1 to 3 (3gang) X = 1 to 4 (4gang) X = 1 to 5 (5gang)	 **2 push buttons (2 objects)** rocker (1 object) Without function	For the B.IQ RTR push button, two button functions or one rocker function can be assigned to each of the individual rockers. Two independent push button functions are assigned to rocker X. A rocker function is assigned to rocker X. Rocker X has no function, i. e. button actuation (left or right) will have no effect, and the status-LEDs of this rocker cannot be triggered.
Push Push button 1 actuation push button function (3-, 4- and 5gang)		
Function of key 1	No function **Switching / pushing** Dimming Shutter Value transmitter 1-byte Value transmitter 2-byte Two telegrams Operating mode switch-over * Setpoint shifting * Light-scene extension / recall Room temperature timer control ** Timer operation *** Controller extension ****	Determines the function of key 1. *: The operating mode switch-over and the setpoint shifting are only visible with activated room temperature regulator function! **: The room temperature timer control can only be parameterized with enabled room temperature timer! ***: Timer operation can only be parameterized when at least one control function has been enabled. ***: The controller extension is visible only if controller extension functions activated!
Function of key 1 = "no function"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of status-LED	 always off always ON Via status object	With function of key 1 = "no function", only the status-LED of the key can be addressed via the relevant object. Pressing a key has no effect. The status-LED is always off. The status-LED is always on. The status-LED indicates the object status of the separate LED object.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "switching / pushing"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always off	The status-LED is always off.
	always ON	The status-LED is always on.
	Status indication (switching object)	The status-LED indicates the object status.
	Inverted status indication (switching object)	The status-LED indicates the inverted object status.
	Operation indication	The status-LED lights up for the parameterized of time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Command on pressing the push button	No function ON **TOGGLE** OFF	Determines the command transmitted on pressing the push button.
Command on releasing the push button	**No function** ON OFF TOGGLE	Determines the command transmitted on releasing the push button.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "dimming"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Status indication (switching object)	The status-LED indicates the object status.
	Inverted status indication (switching object)	The status-LED indicates the inverted object status.
	Operation indication	The status-LED lights up for the parameterized period of time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Command on pressing the push button, push button function		Determines the response to a key-press.
	Darker (OFF)	A short operationwill trigger an OFF telegram, a long operationwill trigger a dimming telegram (darker).
	Brighter (ON)	A short-time operationwill trigger an ON telegram; a long operationwill trigger a dimming telegram (brighter).
	Brighter/ darker (TOGGLE)	The internally stored switching status will be switched-over with a short-time key-press. If the saved status is ON (OFF), an OFF- (ON) telegram will be triggered. A long operationwill cause the transmission of a "darker" telegram followed by a "brighter" telegram and vice versa.
Dimming brighter by	**100 %** 50 % 25 % 12,5 % 6 % 3 % 1,5 %	Determines the maximum dimming increments for a dimming telegram. A dimming telegram can increase brightness by a maximum of X %. This parameter depends on the set key function.
Dimming darker by	**100 %** 50 % 25 % 12,5 % 6 % 3 % 1,5 %	Determines the maximum dimming increments for a dimming telegram. A dimming telegram can reduce brightness by a maximum of X %. This parameter depends on the set key function.

Stop telegram	**YES** NO	One or no telegram is transmitted on releasing the push button.
Time between switching and dimming (0,1… 51) * 1 sec	sec … 51 sec , 0.4 sec (increments: 0.1 sec)	Time from which on the function assigned to the operation(dimming) is executed.
Telegram repetition	**NO** YES	Periodical repetition of dimming telegram during a key-press.
Time between two telegrams	**200 ms** 750 ms 300 ms 1 sec 400 ms 2 sec 500 ms	Time between two telegrams with set telegram repetition. A new dimming telegram is transmitted each time this time has elapsed. Only if telegram repetition = “yes”.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "shutter"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Operation indication	The status-LED lights up for the parameterized period of time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.

Operation concept	**Step – Move - Step** Move - Step	Determines the telegram sequence following a key-press. Step – Move – Step:  Pressing a key will transmit a STEP and time T1 is started ("time between step and move operation"). No other telegram will be transmitted, if the key is released within T1. This step serves to stop a continuous run in progress. If the key is pressed longer than T1, a MOVE is automatically transmitted after T1 has elapsed while time T2 ("Lamella adjustment time") is started. A STEP will be transmitted, if the key is released again within T2. This function is used to adjust the Lamellas. T2 should correspond to the time required for a 180°rotation. Move – Step:  Pressing a key will transmit a MOVE and time T1 is started ("Lamella adjustment time"). A STEP will be transmitted, if the key is released again within T1. This function is used to adjust the Lamellas. T1 should correspond to the time required for a 180°rotation of the Lamellas.
Time between step and move operation (0.1 ... 127.5) * 1 sec	0.1 sec ... 127.5 sec , 0.3 sec (increments: 0.1 sec)	Time from which on the function assigned to the operationis executed. Only if operation concept = "Short – Long – Short".

Function of shutter push button	**UP**	A short-time operationwill trigger a STEP telegram (UP); a long operationwill trigger a MOVE telegram (up).
	DOWN	A short-time operationwill trigger a STEP telegram (DOWN); a long operationwill trigger a MOVE telegram (down).
	TOGGLE	With this setting the direction, which is internally stored and tracked via the bus, will be switched-over with each long actuation (MOVE). If a short-time actuation transmits a STEP telegram, then this STEP is always switched in the opposite direction of the last MOVE. Several STEP telegrams transmitted successively are always switched in the same direction.
Lamella adjustment time (0 ... 127.4) * 1 sec	0 sec … 127.5 sec , 0.6 sec (increments: 0.1 sec)	Time during which a transmitted MOVE telegram can be terminated by releasing the key (STEP). This function serves to adjust the Lamellas of a shutter.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1= "Value transmitter 1-byte"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Operation indication	The status-LED lights up for the parameterized period of time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Value (0...255)	0 ... 255, **255**	Determines the value to be transmitted.
Adjustment of values by long pressing of a push button	**Disabled** Enabled	Pressing the key for at least 5 sec will cause the current value periodically to be reduced by the parameterized increments and transmitted. The previously transmitted value is saved after releasing the key. This parameter determines whether a value adjustment is possible.
Time between two telegrams	0.5 sec; **1 sec**; 2 sec; 3 sec	Time between two periodical telegrams (long key-press).
Step width (1...10)	1 ... 10, **10**	Increments by which the adjusted value is reduced with a long key-press.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1= "Value transmitter 2-byte"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Operation indication	The status-LED lights up for the parameterized of time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Function as	**Temperature value transmitter** Brightness value transmitter Value transmitter	Determines the function to be executed.
Temperature value (0...40) * 1 °C	0 ... 40 °C in 1 °C increments, **25 °C**	Adjusting the temperature value to be transmitted. Only for the function as = "temperature value transmitter"
Brightness value (0...1500) * 1 lux	0 ... 1500 lux in 50 lux increments, **500-lux**	Adjusting the brightness value to be transmitted. Only for the function as = "brightness value transmitter"
Value (0...65535)	0 ... 65535, **0**	Adjusting the 2-byte value to be transmitted. Only for the function as "value transmitter"
Adjustment of values by long pressing of a push button	**Disabled** Enabled	Pressing the key for at least 5 sec will cause the current value periodically to be reduced by the parameterized increments and transmitted. The previously transmitted value is saved after releasing the key. This parameter determines whether a value adjustment is possible.
Time between two telegrams	0.5 sec; 1 sec; 2 sec; 3 sec	Time between two periodical telegrams (long key-press).

Step width	Temperature value transmitter: **1 ℃** Brightness value transmitter: **50 lux** Value transmitter: 1 75 2 100 5 200 **10** 500 20 750 50 1000	Increments by which the adjusted value is reduced with a long key-press.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "two telegrams"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Status indication (switching object)	The status-LED indicates the object status.
	Inverted status indication (switching object)	The status-LED indicates the inverted object status.
	Operation indication	The status-LED lights up for the parameterized time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Type of 1st object	**Switching** Value	This parameter defines the data format of the first communication object.
Type of 2nd object	**Switching** Value	This parameter defines the data format of the second communication object.

Operating concept		Distinction is made between two different operating concepts. This parameter defines the key evaluation and specifies the telegram transmission.
	Always two telegrams	The first telegram is always sent at the same time as the key is being actuated. The presetting effects the sending of the second telegram after a certain delay time started upon the key actuation has elapsed. Thus, you need not hold the key pressed to send the second telegram.
	Either telegram 1 or 2	For this operating concept, always only one of the two parameterized telegrams will be set to the bus upon a key actuation. The time how long the key is being actuated determines which of the two telegrams will be sent.
Delay between the 1st and 2nd telegram	**YES** NO	The parameter determines whether a time between object 1 and object 2 will be activated (set to "yes") or whether the telegrams of both objects will be transmitted directly one after another without any delay on pressing the push button. In this case the chronological sequence cannot be determined. Releasing the key has no other or additional function. Only for "Operating concept = always two telegrams".
Time between the 1st and 2nd telegram (1…1800sec)	1...1800 s; **10 s**	This is where the time is defined that has to elapse before the telegram of the second object is transmitted on the bus. The transmission of the second object does not require to keep the key pressed. The delay time can be re-triggered by actuating the key several times. Alternatively, you can deactivate the delay time so that the two telegrams will be sent to the bus one by one upon one key actuation. In this case, you cannot predetermine the time sequence of the telegrams. Only with activated delay time!
Actuation period for the 2nd telegram (0.5…60) * 1 s	0.5...60 s; **1 s**	This parameter defines the time period for distinguishing between short-time and long-time key actuation. If you actuate the key for a time shorter that the parameterized period only the first telegram will be sent when you release the key. If you exceed the parameterized actuation period only the second telegram will be transmitted. You can parameter periods between 0.5 and 60 seconds Only for "Operating concept = either telegram 1 or 2".

Command when pressing key for the 1st object	No function ON **TOGGLE** OFF	This is where the command (ON, OFF) is parameterized which will be transmitted via the object 1 on the bus on pressing the push button. Only if "type of 1st object" = "switching"!
Command when pressing key for the 2nd object	No function ON **TOGGLE** OFF	This is where the command (ON, OFF) is parameterized which will be transmitted via the object 2 on the bus on pressing the push button. Only if "type of 2nd object" = "switching"!
Value when pressing key for the 1st object	0...255; **255**	This is where the value (0…255) is parameterized which will be transmitted via the object 1 on the bus on pressing the push button. Only if "type of 1st object" = "value"!
Value when pressing key for the 2nd object	0...255; **255**	This is where the value (0…255) is parameterized which will be transmitted on the bus via the object 2 on pressing the push button. Only if "type of 2nd object" = "value"!

Push button function – General – push button 1 – status of push button 1 (3gang, 4gang and 5gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "operating mode switch-over"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Indication of operating mode is active	The status-LED lights up as soon as an operating mode is activated via the assigned key.
	Indication of operating mode is inactive	The status-LED lights up as soon as an operating mode is activated via the assigned key.
	Operation indication	The status-LED lights up for the parameterized time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.

Operating mode on pressing a push button	**Comfort operation** Standby operation Night operation Frost / heat protection operation Presence button	Determines the room temperature regulator function, which is to be activated when pressing a key. *: The presence button can only be parameterized if "presence detection via presence button" is enabled.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "setpoint shifting"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Operation indication	The status-LED lights up for the parameterized time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Command on pressing the push button	**Decrease step value** Increase step value	It determines whether the step value for the basic temperature setpoint shifting for the internal controller takes place in negative direction (decreasing step value) or whether it takes place in positve direction (increasing step value). The increments for the temperature adjustment are determined via the controller (not via the controller extension)!

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "light-scene extension / recall""		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.

Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Operation indication	The status-LED lights up for the parameterized time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Function as		Defines the functionality of the scene recall.
	Light-scene extension	A operationwill recall an ‘external light-scene’ via the light-scene extension object.
	Internal scene request *	A operationwill recall an ‘internal’ scene.
		*: This setting can only be parameterized, if the light-scene function of the push button RTR is enabled.
Light-scene (1...64)	1 ... 64, **1**	Determines the light-scene number to be transmitted via the object. Only if "function as = light-scene extension"
Scene extension 1...8	1 ... 8, **1**	Determines the number of the internal scene to be recalled. Only if "function as = internal scene request"!
Memory function	**No** Yes	This parameter enables the memory function. If function is enabled, a long operation(> 5 sec) will trigger the transmission of a memory telegram or it will store the internal scene according to the parameterized number.

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = " room temperature timer control"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Indication of room temperature timer is active	The status-LED lights up as soon as the room temperature timer is activated via key actuation.
	Indication of room temperature timer is inactive	The status-LED lights up as soon as the room temperature timer is activated via key actuation.
	Operation indication	The status-LED lights up for the parameterized time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Reaction on pressing a push button	Activate room temperature timer (ON)	A operationwill activate the timer. The parameterized switching programs will be executed.
	Deactivate room temperature timer (OFF)	A operationwill deactivate the timer. The parameterized switching programs will not be executed or suppressed.
	Deactivate / activate room temperature timer (TOGGLE)	A operationwill activate or deactivate the timer. Switching between executing and suppressing the switching programs

Push button function – General – push button 1 – status of push button 1 (3 gang, 4 gang and 5 gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if “function of the status-LED” = “ via status object”!
Function of key 1 = “timer operation”		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Indication of timer is active	The status-LED lights up as soon as the timer is activated via key actuation.
	Indication of timer is inactive	The status-LED lights up as soon as the timer is activated via key actuation.
	Operation indication	The status-LED lights up for the parameterized time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Function	Both timers enabled: **Timer 1** Timer 2 Only timer 1 enabled: **Timer 1** Only timer 2 enabled: **Timer 2**	Depending on which timer(s) has (have) been enabled, this parameter specifies which of the timer(s) should be triggered by the push button function.
Reaction on pressing a push button	Activate timer (ON)	Actuating the push button will activate the corresponding timer. The parameterized switching programs will be executed.
	Deactivate timer (OFF)	Actuating the push button will deactivate the corresponding timer. The parameterized switching programs will not be executed or will be suppressed, respectively.
	Deactivate / activate timer (TOGGLE)	Actuating the push button will activate or deactivate the over-over between executing and suppressing the switching programs.

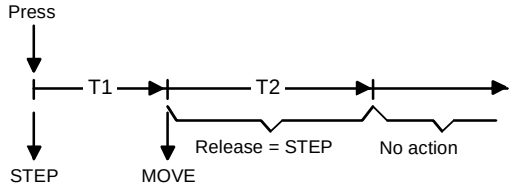
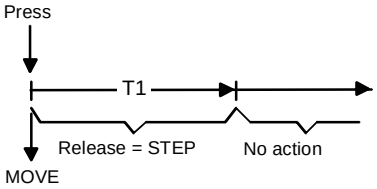
Push button function – General – push button 1 – status of push button 1 (3gang, 4gang and 5gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Function of key 1 = "controller extension"		
Text for push button assistance	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short button actuation (< 0.4 s) when the button assistance function has been enabled.
Function of the status-LED		Determines the function of the status-LED.
	always OFF	The status-LED is always off.
	always ON	The status-LED is always on.
	Indication of key function is active	The status-LED lights up as soon as function is activated via the assigned key.
	Indication of key function is inactive	The status-LED lights up as soon as a function is deactivated via the assigned key.
	Operation indication	The status-LED lights up for the parameterized time when key is actuated.
	Via status object	The status-LED indicates the object status of the separate LED object.
Function	**Normal operating mode switch-over**	Defines the functionality of the controller extension.
	Forced operating mode switch-over	
	Presence button	
	Setpoint shifting	
Operating mode on pressing a push button	**Comfort operation** Standby operation Night operation Frost / heat protection operation	Determines the operating mode, which is to be transmitted on the bus and activated on the controller when pressing a key. Only if "function" = "normal operating mode switch-over"!
Operating mode on pressing a push button	Auto **Comfort operation** Standby operation Night operation Frost / heat protection operation	Determines the forced operating mode, which is to be transmitted on the bus and activated on the controller when pressing a key. Only if "function" = "forced operating mode switch-over"!
Command on pressing the push button	**Decrease step value** Increase step value	It determines whether the step value for the basic temperature setpoint shifting of the internal controller takes place in negative direction (decreasing step value) or whether it takes place in positive direction (increasing step value). The increments for the temperature adjustment are determined via the controller (not via the controller extension)! Only if "function" = "setpoint shifting"!

Push button function – General – push button 1 – status of push button 1 (3gang, 4gang and 5gang)		
Polarity of status object	Inverted (on = 0) **Not inverted (on = 1)**	Defines the polarity of the status-LED object. Only if "function of the status-LED" = " via status object"!
Push button function – General – push button 2 see push button 1! (3 gang, 4 gang and 5 gang)		
Push button function – General – push button 3 see push button 1! (3 gang, 4 gang and 5 gang)		
Push button function – General – push button 4 see push button 1! (3 gang, 4 gang and 5 gang)		
Push button function – General – push button 5 see push button 1! (3 gang, 4 gang and 5 gang)		
Push button function – General – push button 6 see push button 1! (3 gang, 4 gang and 5 gang)		
Push button function – General – push button 7 see push button 1! (4 gang, 5 gang)		
Push button function – General – push button 8 see push button 1! (4 gang, 5 gang)		
Push button function – General – push button 9 see push button 1! (5 gang)		
Push button function – General – push button 10 see push button 1! (5 gang)		

Parameters		
Description:	**Values:**	**Comment:**
Push button function – General – rocker 1 (3 gang, 4 gang and 5 gang)		
Function of rocker 1	No function **Switching** Dimming Shutter Two telegrams Operating mode switch-over * Controller extension **	Determines the function of rocker 1. *: The operating mode switch-over is visible only if the room temperature regulator function is switched-on! **: The controller extension is visible only if the controller extension function switched-on!
Function of rocker 1 = "no function"		
Text for push button assistance [left]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the left button (< 0.4 s) when the push button assistance function has been enabled.
Text for push button assistance [right]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the right button (< 0.4 s) when the push button assistance function has been enabled.
If function of rocker 1 = "no function", only the status-LED of the rocker can be activated via the corresponding status object. Actuating a key or a rocker has no effect. Only status-LED parameter (see "status rocker 1")!		
Function of rocker 1 = "switching"		
Text for push button assistance [left]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the left button (< 0.4 s) when the push button assistance function has been enabled.
Text for push button assistance [right]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the right button (< 0.4 s) when the push button assistance function has been enabled.
Command on pressing a rocker	Left = ---, right = --- Left = OFF, right = ON **Left = ON, right = OFF** Left = TOGGLE, right = TOGGLE Left = OFF, right = OFF Left = ON, right = ON	Determines the commands to be transmitted when pressing both keys.
Function of rocker 1 = "dimming"		

Text for push button assistance [left]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the left button (< 0.4 s) when the push button assistance function has been enabled.
Text for push button assistance [right]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the right button (< 0.4 s) when the push button assistance function has been enabled.
Command on pressing a rocker		Determines the response to a operationon the rocker.
	Left = Brighter (ON), Right = Darker (OFF)	A short operation(left key) will trigger an ON telegram; a long operation(left key) will trigger a dimming telegram (brighter). A short operation(right key) will trigger an OFF telegram; a long operation(right key) will trigger a dimming telegram (darker).
	Left = darker (OFF), Right = brighter (ON)	A short operation(left key) will trigger an OFF telegram; a long operation(left key) will trigger a dimming telegram (darker). A short operation(right key) will trigger an ON telegram; a long operation(right key) will trigger a dimming telegram (brighter).
	Left = TOGGLE, right = TOGGLE	The internally stored switching status is toggled via a short-time key-press. If the saved state is ON (OFF), an OFF- (ON) telegram will be triggered. A long operationwill transmit a "darker" telegram followed by a "brighter" telegram and vice versa.
	Left = Brighter (ON), Right = brighter (ON)	A short operation(left key) will trigger an ON telegram; a long operation(left key) will trigger a dimming telegram (brighter). A short operation(right key) will also trigger an ON telegram; a long operation(right key) will also trigger a dimming telegram (brighter).
	Left = darker (OFF), Right = Darker (OFF)	A short operation(left key) will trigger an OFF telegram; a long operation(left key) will trigger a dimming telegram (darker). A short operation(right key) will also trigger an OFF telegram; a long operation(right key) will also trigger a dimming telegram (darker).
Dimming brighter by	**100 %** 6 % 50 % 3 % 25 % 1,5 % 12,5 %	Determines the maximum dimming increments for a dimming telegram. A dimming telegram can increase brightness by a maximum of X %.
Dimming darker by	**100 %** 6 % 50 % 3 % 25 % 1,5 % 12,5 %	Determines the maximum dimming increments for a dimming telegram. A dimming telegram can reduce brightness by a maximum of X %.
Stop telegram?	**Yes** No	One or no telegram is transmitted when releasing one of the keys (left or right).

Time between switching and dimming (0,1 ... 51) * 1 sec	0.1 sec ... 51 sec , 0.4 sec (increments: 0.1 sec)	Time after which the function assigned to a long operation(dimming) is executed.
Telegram repetition	**No** Yes	Enables the periodical repetition of dimming telegrams during a key-press.
Time between two dimming telegrams	**200 ms** 750 ms 300 ms 1 sec 400 ms 2 sec 500 ms	Time between two telegrams when telegram repetition is active. A new dimming telegram is transmitted after this time has elapsed. Only if telegram repetition = "yes"!
Function of rocker 1 = "shutter"		
Text for push button assistance [left]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the left button (< 0.4 s) when the push button assistance function has been enabled.
Text for push button assistance [right]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the right button (< 0.4 s) when the push button assistance function has been enabled.

Operation concept	**Step – Move - Step** Move - Step	Determines the telegram sequence following a key-press. Step – Move – Step:  Pressing the key will transmit a STEP command and time T1 ("time between step and move operation") is started. No other telegram will be transmitted, if the key is released within T1. This step serves to stop a continuous run in progress. If the key is pressed longer than T1, a MOVE command is automatically transmitted after T1 has elapsed and time T2 ("Lamella adjustment time") is started. A STEP command will be transmitted, if the key is released again within T2. This function is used for Lamella adjustment. T2 should correspond to the time required for a 180°rotation of the Lamellas. Move – Step:  Pressing a key will transmit a MOVE command and time T1 ("Lamella adjustment time") is started. A STEP command will be transmitted, if the key is released again within T1. This function is used for Lamella adjustment. T1 should correspond to the time required for a 180°rotation of the Lamellas.
Time between step and move operation (0.1 ... 127.5) * 1 sec	0.1 sec ... 127.5 sec , 0.3 sec (increments: 0.1 sec)	Time after which the function assigned to the operationis executed. Only if Operation concept = "Short – Long – Short".

Command on pressing a rocker	**Left shutter UP / right shutter DOWN**	A short operation(left key) will trigger a STEP telegram (UP); a long operation(left key) will trigger a MOVE telegram (up). A time operation(right key) will trigger a STEP telegram (UP); a long operation(right key) will trigger a MOVE telegram (down).
	Left shutter DOWN / right shutter UP	A short operation(left key) will trigger a STEP telegram (UP); a long operation(left key) will trigger a MOVE telegram (down). A time operation(right key) will trigger a STEP telegram (DOWN); a long operation(right key) will trigger a MOVE telegram (Up).
	Left shutter TOGGLE / right shutter TOGGLE	With this setting the direction, which is internally stored and tracked via the bus, will be switched-over with each long actuation (MOVE). If a short-time actuation transmits a STEP telegram, then this STEP is always switched in the opposite direction of the last MOVE. Several STEP telegrams transmitted successively are always switched in the same direction.
	Left shutter UP / right shutter UP	A short operation(left key) will trigger a STEP telegram (UP); a long operation(left key) will trigger a MOVE telegram (up). A short operation(right key) will also trigger a STEP telegram (UP); a long operation(right key) will also trigger a MOVE telegram (Up).
	Left shutter DOWN / right shutter DOWN	A short operation(left key) will trigger a STEP telegram (DOWN); a long operation(left key) will trigger a MOVE telegram (down). A short operation(right key) will also trigger a STEP telegram (UP); a long operation(right key) will also trigger a MOVE telegram (down).
Lamella adjustment time (0 ... 127.4) * 1 sec	0 sec ... 127.5 sec , 0.6 sec (increments: 0.1 sec)	Time during which a transmitted MOVE telegram can be terminated by releasing the key (STEP). This function serves to adjust the Lamellas.

Function of key 1 = "two telegrams"		
Text for push button assistance [left]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the left button (< 0.4 s) when the push button assistance function has been enabled.
Text for push button assistance [right]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the right button (< 0.4 s) when the push button assistance function has been enabled.
Type of 1st object	**Switching** Value	This parameter defines the data format of the first communication object.
Type of 2nd object	**Switching** Value	This parameter defines the data format of the second communication object.
Operating concept		Distinction is made between two different operating concepts. This parameter defines the key evaluation and specifies the telegram transmission.
	Always two telegrams	The first telegram is always sent at the same time as the key is being actuated. The presetting effects the sending of the second telegram after a certain delay time started upon the key actuation has elapsed. Thus, you need not hold the key pressed to send the second telegram.
	Either telegram 1 or 2	For this operating concept, always only one of the two parameterized telegrams will be set to the bus upon a key actuation. The time how long the key is being actuated determines which of the two telegrams will be sent.
Delay between the 1st and 2nd telegram	**Yes** No	The parameter determines whether a time will be activated between object 1 and object 2 (set to "yes") or whether the telegrams of both objects will be transmitted directly one after another without any delay on pressing the push button. In this case the chronological sequence cannot be determined. Releasing the key has no other or additional function. Only for "Operating concept = always two telegrams".

Time between the 1st and 2nd telegram (1…1800sec)	1...1800sec; **10 sec**	This is where the time is defined that has to elapse before the telegram of the second object will be transmitted on the bus. The transmission of the second object does not require to keep the key pressed. The delay time can be re-triggered by actuating the key several times. Alternatively, you can deactivate the delay time so that the two telegrams will be sent to the bus one by one upon one key actuation. In this case, you cannot predetermine the time sequence of the telegrams. Only with activated delay time!
Actuation period for the 2nd telegram (0.5…60) * 1 s	0.5...60 s; **1 s**	This parameter defines the time period for distinguishing between short-time and long-time key actuation. If you actuate the key for a time shorter that the parameterized period only the first telegram will be sent when you release the key. If you exceed the parameterized actuation period only the second telegram will be transmitted. You can param. periods between 0.5 and 60 seconds Only for "Op. concept = either telegram 1 or 2".
Command when pressing the left key for the 1st object	No function **ON** TOGGLE OFF	This is where the command (ON, OFF) is parameterized which will be transmitted on the bus via the object 1 when pressing the left key. Only if "type of 1st object" = "switching"!
Command when pressing the right key for the 1st object	No function ON TOGGLE **OFF**	This is where the command (ON, OFF) is parameterized which will be transmitted on the bus via the object 1 when pressing the right key. Only if "type of 1st object" = "switching"!
Command when pressing the left key for the 2nd object	No function **ON** TOGGLE OFF	This is where the command (ON, OFF) is parameterized which will be transmitted on the bus via the object 2 when pressing the left key. Only if "type of 2nd object" = "switching"!
Command when pressing the right key for the 2nd object	No function ON TOGGLE **OFF**	This is where the command (ON, OFF) is parameterized which will be transmitted on the bus via the object 2 when pressing the right key. Only if "type of 2nd object" = "switching"!
Value when pressing the left key for the 1st object	0...255; **0**	This is where the value (0…255) is parameterized which will be transmitted on the bus via the object 1 on pressing the push button. Only if "type of 1st object" = "value"!

Value when pressing the right key for the 1st object	0...255; **0**	This is where the value (0…255) is parameterized which will be transmitted on the bus via the object 1 when pressing the right key. Only if "type of 1st object" = "value"!
Value when pressing the left key for the 2nd object	0...255; **255**	This is where the value (0…255) is parameterized which will be transmitted on the bus via the object 2 on pressing the push button. Only if "type of 2nd object" = "value"!
Value when pressing the right key for the 2nd object	0...255; **255**	This is where the value (0…255) is parameterized which will be transmitted on the bus via the object 2 when pressing the right key. Only if "type of 2nd object" = "value"!
Function of rocker 1 = "operating mode switch-over"		
Text for push button assistance [left]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the left button (< 0.4 s) when the push button assistance function has been enabled.
Text for push button assistance [right]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the right button (< 0.4 s) when the push button assistance function has been enabled.
Operating mode when pressing a key	**Switching over between the operating modes comfort, standby, night and frost / heat protection operation** (no other adjustment possible!)	Determines the function of rocker 1.
Function of rocker 1 = "controller extension"e"		
Text for push button assistance [left]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the left button (< 0.4 s) when the push button assistance function has been enabled.
Text for push button assistance [right]	[20-character text] The parameterized push button function is preset.	Here, the push button assistance text can be defined which will be displayed upon a short actuation of the right button (< 0.4 s) when the push button assistance function has been enabled.

Function	**Normal operating mode switch-over** Forced operating mode switch-over Presence button Setpoint shifting	Defines the functionality of the controller extension.
Operating mode on pressing a push button	**Switching over between the operating modes comfort, standby, night and frost / heat protection operation** (no other adjustment possible!)	A operationwill switch-over between comfort, standby, night and frost/heat protection. Only if "function" = "normal operating mode switch-over"!
Operating mode on pressing a push button	**Switching over between the operating modes auto, comfort, standby, night and frost / heat protection operation** (no other adjustment possible!)	A operationwill switch-over between auto, comfort, standby, night and frost/heat protection. Only if "function" = "forced operating mode switch-over"!
Command on pressing a push button	**decrease left step value / increase right step value** increase left step value / decrease right step value	It determines whether the step value for the basic temperature setpoint shifting takes place in negative direction upon operation(decreasing step value) or whether it takes place in negative direction upon operation(increasing step value). Only if "function" = "setpoint shifting"!
Rocker 2 see rocker 1! (3 gang, 4 gang and 5 gang)		
Rocker 3 see rocker 1! (3 gang, 4 gang and 5 gang)		
Rocker 4 see rocker 1! (4 gang, 5 gang)		
Rocker 5 see rocker 1! (5 gang)		

Parameters		
Description:	**Values:**	**Comment:**
Push button function – general – rocker 1 – state of rocker 1 (3 gang, 4 gang and 5 gang)		
Function of rocker 1 = "no function", "switching", "dimming", "shutter", "two telegrams, "operating mode switch-over" and "controller extension".		
Function of left status-LED	always OFF always ON **Via status object**	Determines how the status-LED of the left rocker is activated. It can be either permanently switched on or off or alternatively it can be activated via its own status communication object.
Polarity of status object left	Inverted (on = 0) **Not inverted (on = 1)**	Determines the polarity of the status object of the left status-LED. Only if "function of left status-LED" = " via status object"!
Function of right status-LED	always OFF always ON **Via status object**	This is where it is determined how the status-LED of the right rocker is addressed. It can be either switched on or off at all times or alternatively it can be addressed via its own status communication object.
Polarity of status object right	Inverted (on = 0) **Not inverted (on = 1)**	Determines the polarity of the status object of the right status-LED. Only if "function of left status-LED" = " via status object"!
Push button function – general – rocker 2 – state of rocker 2 see state of rocker 1 (3 gang, 4 gang and 5 gang)		
Push button function – general – rocker 3 – state of rocker 3 see state of rocker 1 (3 gang, 4 gang and 5 gang)		
Push button function – general – rocker 4 – state of rocker 4 see state of rocker 1 (4 gang, 5 gang)		
Push button function – general – rocker 5 – state of rocker 5 see state of rocker 1 (5 gang)		

Parameters		
Description:	**Values:**	**Comment:**
Room temperature regulator function		
Operating mode switch-over	**Via value (1-byte)**	The switch-over of the operating modes via the bus takes place according to the KONNEX specification via a 1-byte value object. In addition, a higher-ranking forced-object is available for this setting.
	Via switching (4 x 1-bit)	The 'classic' switch-over of the operating modes via the bus is via separate 1-bit objects.
Use fan stages display	**No**	To enable the fan stages display for a fan coil actuator (setting *"yes"*). When this function is enabled the associated communication objects will become visible.
	Yes	
Fan stage display object type		To define the feedback message data format of the fan coil actuator for fan stage display.
	3 x 1 bit	Feedback proceeds per stage via separate 1-bit objects.
	1-byte	Common feedback for all stages via a 1-byte object.
Control circuits (FA)	**1 control circuit**	The room temperature regulator addresses only one control circuit.
	2 control circuits	The room temperature regulator can address up to two control circuits.
Heating / cooling mode (FA)	**Heating** Cooling Heating and cooling * Basic and additional heating * Basic and additional cooling * Basic /additional heating/cooling *	Setting the heating/cooling mode *: The "heating and cooling" mixed-mode and the two stage controlled operation are not available when using two control circuits!
Additional stage inhibit object (FA)		The additional stages can be separately disabled via the bus. The parameter enables the disable object.
	No	The additional stages cannot be separately disabled.
	Yes	The additional stages cannot be separately disabled via the disable object. Only if in two-stage heating or cooling mode!

Send variable heating and cooling to one common object (FA)	**No** Yes	If the parameter is set to "yes", the actuating variable will be transmitted on a shared object during heating or cooling. This function is used, if the same heating system is used to cool the room in the summer and used to heat the room in the winter. Only with "heating and cooling" mixed-mode, if applicable, with additional stages!
Type of heating control (if applicable, for basic and additional stage) (FA)	**Continuous PI control** Switching PI control (PWM) Switching 2-point control (ON/OFF)	Selecting a control algorithm (PI or 2-point) with data format (1-byte or 1-bit) for the heating system.
Type of heating (if applicable, for basic and additional stage) (FA)	**Hot water heating (5 K / 150 min)** Underfloor heating (5 K / 240 min) Electric heating (4 K / 100 min) Fan convector (4 K / 90 min) Split unit (4 K / 90 min) Via control parameter	Adapting the PI algorithm to different heating systems using experience values for the proportional range and reset time control parameters. Separate input of control parameter. Only if "type of heating control" = "PI"!
Proportional range heating (10 ... 127) * 0.1 K (FA)	10...127, **50**	Separate setting of the "proportional range" control parameter. Only if "type of heating control" = "via control parameter"!
Reset time heating (0 ... 255) * 1 min; 0 = inactive (FA)	0...255, **150**	Separate setting of the "reset time" control parameter. Only if "type of heating control" = "via control parameter"!
Heating 2-point controller hysteresis upper limit (5 ... 127) * 0.1 K (FA)	5...127, **5**	Definition of switch-on and switch-off temperature for heating. Only if "type of heating control" = "2-point"!
Heating 2-point controller hysteresis lower limit (-128 ... –5) * 0.1 K (FA)	-128...-5, **-5**	Definition of switch-on and switch-off temperature for heating. Only if "type of heating control" = "2-point"!
Type of cooling control (if applicable, for basic and additional stage) (FA)	**Continuous PI control** Switching PI control PWM) Switching 2-point control (ON/OFF)	Selecting a control algorithm (PI or 2-point) with data format (1-byte or 1-bit) for the cooling system. Only if "Send variable heating and cooling to one common object" = "no"! If "Send variable heating and cooling to one common object" = "yes", the parameter settings for "type of heating control" will be accepted.

Type of cooling (if applicable, for basic and additional stage) (FA)	**Cooling ceiling (5 K / 240 min)** Fan convector (4 K / 90 min) Split unit (4 K / 90 min) Via control parameter	Adapting the PI algorithm to different cooling systems using experience values for the proportional range and reset time control parameters. Separate input of control parameter. Only if "type of cooling control" = "PI"!
Proportional range cooling (10 ... 127) * 0.1 K (FA)	10...127, **50**	Separate setting of the "proportional range" control parameter. Only if "type of cooling control" = "via control parameter"!
Reset time cooling (0 ... 255) * 1 min; 0 = inactive (FA)	0...255, **240**	Separate setting of the "reset time" control parameter. Only if "type of cooling control" = "via control parameter"!
Cooling 2-point controller hysteresis upper limit (5 ... 127) * 0.1 K (FA)	5...127, **5**	Definition of switch-on and switch-off temperatures for cooling. Only if "type of cooling control" = "2-point"!
Cooling 2-point controller hysteresis lower limit (-128 ... –5) * 0.1 K (FA)	-128...-5, **-5**	Definition of switch-on and switch-off temperatures for cooling Only if "type of cooling control" = "2-point"!
Operation of controller inhibitable		It is possible to disable the local control of the room temperature regulator (all rockers). If control is actively disabled the "  " symbol will appear on the display.
	No	Disable function is deactivated.
	Always disabled	Always disables the controller control.
	Via bus	Enables the "controller operation disable" object 39.
Switch-off controller (dew point)	**NO** Via bus	This parameter enables the "disable controller " object 40. There is no control until enabled (actuating variables = 0). If controller is actively disabled (dew point mode), the "  " symbol will appear on the display.
Frost/heat protection		It is possible to determine how the room temperature regulator switches into the frost/heat protection.
	Automatic frost protection	The frost protection automatic is activated. Depending on the room temperature this allows an automatic switch-over into the frost protection mode.
	Via window status	The switch-over into the frost/heat protection takes place 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.	Determines the decrease temperature by which the room temperature has to decrease within one minute in order for the controller to switch into the frost protection mode. The "OFF" setting will deactivate the frost protection automatic. Only if "frost/heat protection = Automatic frost protection"!
Frost protection period in automatic mode (1...255) * 1 min.	1 ... 255, **20**	Defines the time after which the controller (in frost protection automatic) will automatically deactivate the frost protection again. Only with enabled automatic frost protection!
Window status delay (0...255) * 1 min.; 0 = inactive	0 ... 255, **0**	Defines the delay time after which the frost/heat protection will be activated via the window status. Only if "frost/heat protection = via window status"!
Switch-over between heating and cooling		In parameterized mixed mode it is possible to switch-over between heating and cooling.
	Automatically	Depending on the operating mode and the room temperature, the switch-over takes place automatically.
	Via object	Switch-over takes place solely via the *"heating/cooling switch-over"* object 35. Only with "heating and cooling" or "basic / additional heating/cooling" mixed modes!
Heating / cooling switch-over after a reset	**Heating** Cooling Heating/cooling switch-over before reset	Determines the heating/cooling mode that after a return of bus voltage. Only if "switch-over between heating and cooling = via object"!
Automatic heating/cooling switch-over transmission	**On changing the heating/cooling** On changing the output value	Determines when a telegram will be automatically transmitted on the bus via the *"heating/cooling switch-over"* object 35. Only if "switch-over between heating and cooling = automatically"!
Cyclical transmission heating/cooling switch-over (0...255) * 1 min; 0 = inactive	0 ... 255, **0**	Determines whether the current object status of *"heating/cooling switch-over"* object 35 is to be periodically output on the bus (with automatic switch-over). The cycle time may be adjusted. The "0" setting will deactivate the periodic transmission of the object value. Only if "switch-over between heating and cooling = automatically"!

Valve protection	**No** Yes	The valve is periodically opened (every 24 hours). Works against calcification and thus prevents the valve to become stuck.

Room temperature function – Set Point Values		
Setpoint shifting step value	**0.1 K** 0.5 K	You can use this parameter to define the temperature step width of a setpoint shift. Each setpoint shifting key actuation or some shifting via the controller extension will change the setpoint temperature value, depending on the incrementation. In addition, this parameter specifies the shifting step width for temperature changes made from the programming menu and via the *"basic setpoint"* object. The push button module rounds the temperature values received via the object and matches the values to the parameterized step value.
Own setpoints for 2nd control circuit	**No** Yes	When both control circuits are used, the second circuit can have its own setpoints. The "yes" setting will enable the setpoint presetting of the second control circuit. Only if "control circuits = 2 control circuits"!
Basic temperature after reset (7.0 ... 40.0) * 1 °C	7.0 °C … + 40 °C, 21 °C	Determines the basic setpoint after the initialization.
Basic temperature in the 2nd control circuit after reset (7.0 ... 40.0) * 1 °C	7.0 °C … + 40 °C, **21 °C**	Determines the basic setpoint of the second control circuit after the initialization. Only if "own setpoints for 2nd control circuit = yes"!
Accept modification of shift of basic setpoint value permanently	**No** Yes	The temperature of the current operating mode of both control circuits can be adapted via the basic setpoint shifting (via display keys). The temperature can be shifted upwards and downwards within the predetermined value range. The "no" setting will delete the temperature shift when switching-over into another operating mode. The "yes" setting will keep the temperature shift when switching-over into another operating mode.
Modification of the basic temperature setpoint value	**deactivated** Permit via display buttons Permit via object Permit via display buttons and bus	Determines whether a change of the basic temperature of the first control circuit is possible via the bus or locally on the device.

Modification of the basic temperature setpoint in the 2nd control circuit	**deactivated** Permit via object	Determines whether a change of the basic temperature of the second control circuit is possible via the bus or locally on the device. Only if "Own setpoints for 2nd control circuit = yes"!
Accept modification of the basic temperature setpoint value permanently	**No** Yes	This parameter determines whether the basic temperature value, which has been adjusted via the bus or locally on the device, is to be permanently (set to "yes") or solely temporarily (set to "no") stored in memory. When set to "yes" the changed basic value will remain even after switching-over into another the operating mode and after a reset. Only if "Modification of the basic temperature setpoint value" = "permit via display buttons", "permit via bus" or "permit via display buttons and bus"!
Modification of the setpoints "cooling"	**deactivated** Permit via display buttons	Allows to change the cooling setpoints of the first control circuit on the device when in mixed mode. The "temperatures cooling" menu – with deactivated change – is not accessible in programming mode! Only if "heating/cooling mode" = "heating and cooling", if applicable, with additional stages!
1st control circuit standby temperature change	**deactivated** Permit via display buttons	Allows to change the cooling setpoints of the first control circuit on the device.
1st control circuit night temperature change	**deactivated** Permit via display buttons	Allows to change the night temperature of the first control circuit on the device.
Frost protection setpoint temperature (7...40) * 1 °C	7 °C … + 40 °C, **7 °C**	Determines the set-temperature with activated frost protection. Only if "heating/cooling mode" = "heating" or "heating and cooling", if applicable, with additional stages!
Heat protection setpoint temperature (7...45) * 1 °C	7 °C … + 45 °C, **35 °C**	Determines the set-temperature with activated heat protection. Only if "heating/cooling mode" = "cooling" or "heating and cooling", if applicable, with additional stage!

Dead band position		The comfort set-temperatures for heating and cooling are derived from the basic setpoint in consideration of the adjusted Dead band. The Dead band (temperature zone for which there is neither heating nor cooling) is the difference between the comfort set-temperatures.
	Symmetrical	Symmetrical: The Dead band preset in the ETS plug-in is divided in two parts at the basic setpoint. The comfort set-temperatures are derived directly from the basic setpoint resulting from the half Dead band.
	Asymmetrical	Asymmetrical: With this setting the comfort set-temperature for heating equals the basic setpoint! The preset Dead band takes only effect from the basic setpoint in the direction of comfort temperature for cooling. Thus the comfort set-temperature for cooling is derived directly from the comfort setpoint for heating. Only with the "heating and cooling" or "basic / additional heating/cooling" mixed modes!
Dead band between heating and cooling (0...127) * 0.1 K	0 ... 127, **20**	The comfort set-temperatures for heating and cooling are derived from the basic setpoint in consideration of the adjusted Dead band. The Dead band (temperature zone for which there is neither heating nor cooling) is the difference between the comfort set-temperatures. Only with the "heating and cooling" or "basic / additional heating/cooling" mixed modes!
Dead band shift	**deactivated** Permit via display keys	Determines whether the Dead band and thus the comfort temperature for cooling may be adjusted on the device in programming mode with "temperature values cooling". Only with "heating and cooling" or "basic /additional heating/cooling" and if "Modification of the setpoints "cooling" = permit via display buttons"!
Difference between basic and additional stages (0...127) * 0.,1 K	0 ... 127, **20**	In a two stage control mode it is necessary to determine the temperature difference to the basic stage with which the additional stage is to be incorporated into the control Only in two stage control operation!
Transmission at setpoint temperature modification by (0...255) * 0.1 K; 0 = no autom. transmission	0 ... 255, **1**	Determines the size of the value change required to automatically transmit the current value via the "set-temperature" object. When using both control circuits with separate setpoints both setpoints can be transmitted.
Cyclical transmission of setpoint temperature (0...255) * 1 min; 0 = inactive	0 ... 255, **0**	Determines whether the set-temperature is to be periodically output via the *"set-temperature"* object. When using both control circuits with separate setpoints both setpoints can be transmitted.

Upward adjustment of basic setpoint temperature (0...10) * 1 K	0 ... 10, **3**	Determines the maximum adjustment range for the upward adjustment of the basic set-temperature. (cf. "Changing the setpoint of the basic temperature" parameter!)
Downward adjustment of basic setpoint temperature (-10...0) * 1 K	-10 ... 0, **-3**	Determines the maximum adjustment range for the downward adjustment of the basic set-temperature. (cf. "Changing the setpoint of the basic temperature" parameter!)
Lower the setpoint temperature during standby operation (heating) (-128...0. * 0,1 K	-128 ... 0, **-20**	The value by which the standby set-temperature for heating is lowered compared to the basic setpoint. Only if "heating/cooling mode" = "heating" or "heating and cooling", if applicable, with additional stages!
Lower the setpoint temperature during night operation (heating) (-128...0. * 0,1 K	-128 ... 0, **-40**	The value by which the night set-temperature for heating is lowered compared to the basic setpoint. Only if "heating/cooling mode" = "heating" or "heating and cooling", if applicable, with additional stages!
Lower the setpoint temperature during standby operation (heating – 2nd contr. circuit) (-128...0. * 0,1 K	-128 ... 0, **-20**	The value by which the standby set-temperature for heating is lowered compared to the basic setpoint. Only if "heating/cooling mode" = "heating" and only with two control circuits with separate setpoints!
Lower the setpoint temperature during night operation (heating – 2nd contr. circuit) (-128...0. * 0,1 K	-128 ... 0, **-40**	The value by which the night set-temperature for heating is lowered compared to the basic setpoint. Only if "heating/cooling mode" = "heating" and only with two control circuits with separate setpoints!
Raise the setpoint temperature during standby operation (cooling) (0...127) * 0,1 K	0 ... 127, **20**	The value by which the standby set-temperature for cooling is raised compared to the basic setpoint. Only if "heating/cooling mode" = "cooling" or "heating and cooling", if applicable, with additional stages!
Raise the setpoint temperature during night operation (cooling) (0...127) * 0,1 K	0 ... 127, **40**	The value by which the night set-temperature for cooling is raised compared to the basic setpoint. Only if "heating/cooling mode" = "cooling" or "heating and cooling", if applicable, with additional stages!

Raise the setpoint temperature during standby operation (cooling – 2nd contr. circuit) (0...127) * 0,1 K	0 ... 127, **20**	The value by which the standby set-temperature for cooling is raised compared to the basic setpoint. Only if "heating/cooling mode" = "cooling" and only with two control circuits with separate setpoints!
Raise the setpoint temperature during night operation (cooling – 2nd contr. circuit) (0...127) * 0,1 K	0 ... 127, **40**	The value by which the night set-temperature for cooling is raised compared to the basic setpoint. Only if "heating/cooling mode" = "cooling" and only with two control circuits with separate setpoints!

Room temperature regulator function – functionality		
Operation mode after reset	Restore operation mode before reset **Comfort operation** Standby operation Night operation Frost / heat protection operation	Parameter for setting the operating mode after the push button's initialisation phase (e.g. after the return of bus voltage). If set to "restore operation mode before reset", a frequent change of the operating mode (several times a day) might affect the product life of the device as the non-volatile storage is designed only for storing permanent values.
Presence detection		If a person is present in the room, it is best to have the controller switch into the comfort mode or comfort mode extension. This parameter determines which detector is used.
	None	There will be no presence detection.
	Via object	The presence detection takes place via a separate object or via a presence button on the push button (key function).
Type of presence detection	Presence button	The presence detection takes place via a presence button on the push button (key function) or via the presence object (e.g. external push button). In case of a detected presence the comfort mode extension or the comfort mode will be activated.
	Presence detector	The presence detection takes place via an external presence detector. The detector will be coupled via the presence object. If a presence is detected, the comfort mode will be called for as long as the presence detector detects movement. Only if "presence detection = switching via Object!"

Lengh of comfort prolongation (0...255) * 1 min; 0 = off	0 ... 255, **30**	During a presence detection the controller may temporarily switch into the comfort extension – depending on the active operating mode. The parameter determines the time after which the comfort mode extension is automatically terminated. Only if "type of presence detection = "presence button"!

Room temperature regulator function – room temperature measuring (FA)		
Temperature detection		Determines which sensor will be used for the room temperature measurement of the first control circuit.
	Internal sensor	Internal sensor: Sensor installed inside the push button RTR.
	External sensor	External sensor: an external sensor coupled via the bus, e.g. for complicated measuring conditions (swimming pools or similar).
	Internal and external sensor	Internal and external sensor: Both sensors are used, for example, in large rooms. Only with one control circuit!
Creating of measuring value internal against external	10 % to 90 % 20 % to 80 % 30 % to 70 % 40 % to 60 % **50 % to 50 %** 60 % to 40 % 70 % to 30 % 80 % to 20 % 90 % to 10 %	Determines the weighting of the measured temperature value for the internal and external sensors. That results in an overall value, which will be used for the further interpretation of the room temperature. Only with one control circuit and if "temperature detection = internal and external sensor"!
Adjustment internal sensor (-128...127) * 0.1 K	-128 ... 127, **0**	Determines the value by which the internal sensor's room temperature value is calibrated. Only if "temperature detection = internal sensor" or "internal and external sensor" or in case of two control circuits!
Adjustment external sensor (-128...127) * 0.1 K	-128 ... 127, **0**	Determines the value by which the external sensor's room temperature value is calibrated. Only if "temperature detection = external sensor" or "internal and external sensor" or in case of two control circuits!
Scanning time for external sensor (0...255) * 1 min; 0 = inactive	0 ... 255, **0**	Determines the measurement period for the external sensor's temperature value. "0" = sensor automatically transmits its temperature value. Only if "temperature detection = external sensor" or "internal and external sensor" or in case of two control circuits!

Transmission at room temperature modification by (0..255) * 0.1 K; 0 = no automatic transmission	0 ... 255, **3**	Determines the size of the value change for the room temperature of the first control circuit to automatically transmit the current value on the bus via the "actual temperature" object 23.
Cyclical transmission of room temperature (0...255) * 1 min; 0 = inactive	0 ... 255, **15**	Determines whether or when the determined room temperature of the first control circuit is to be periodically output via the "actual temperature" object 23.
Send temperature alarm via object	**No** Yes	This parameter enables the "temperature alarm" function if set to "yes".
Lower temperature limit value (0...40 °C)	0...40 °C; **7 °C**	Lower temperature value for temperature alarm. Only if "Send temperature alarm via object" = "yes"!
Upper temperature limit value (0...40 °C)	0...40 °C; **35 °C**	Upper temperature value for temperature alarm. Only if "Send temperature alarm via object" = "yes"!

Room temperature regulator function – variable and status output (FA)		
Automatic transmission at modification by (0...100) * 1 %; 0 = inactive	0 ... 100, **3**	Determines the size of the actuating variable change that will automatically transmit the continuous actuating variables via the actuating variable objects. Only if at least one type of control is parameterized to "continuous PI control"!
Cycle time of the switching variable (1...255) * 1 min	1 ... 255, **15**	Determines the cycle time for the pulse width modulated actuating variable (PWM). Only if at least one type of control is parameterized to "switching PI control (PWM)"!
Cycle time for automatic transmission (0...255) * 1 min; 0 = inactive	0 ... 255, **10**	Time interval for the periodic transmission of the actuating variable via the actuating variable objects. Only if at least one type of control is parameterized to "continuous PI control" or "switching 2-point control"!
Output of the heating variable	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output heating Only if "heating/cooling mode = heating" or "heating and cooling"!

Output of the heating variable in the 2nd control circuit	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output heating 2nd control circuit Only for one control circuit if "heating/cooling mode = heating"!
Output of the heating basic stage variable	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output basic stage heating Only if "heating/cooling mode= basic and additional heating" or "basic/additional heating/cooling"!
Output of the heating additional stage variable	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output additional stage heating Only if "heating/cooling mode= basic and additional heating" or "basic/additional heating/cooling"!
Output of the cooling variable	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output cooling Only if "heating/cooling mode = cooling" or "heating and cooling"!
Output of the cooling variable in the 2nd control circuit	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output cooling 2nd control circuit Only for two control circuits if "heating/cooling mode = cooling"!
Output of the cooling basic stage variable	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output basic stage cooling Only if "heating/cooling mode= basic and additional cooling" or "basic/additional heating/cooling"!

Output of the cooling additional stage variable	inverted **Normal**	continuous: Actuating variable = 100 % - normal actuating variable switching: Actuating variable = 1 – normal actuating variable normal actuating variable output additional stage cooling Only if "heating/cooling mode= basic and additional cooling" or "basic/additional heating/cooling"!
Heating indication	**No** Yes	Enables the "heating" message function and thus the "message heating" object 37. The message applies solely to the first control circuit.
Cooling indication	**No** Yes	Enables the "cooling" message function and thus the "message cooling" object 38. The message applies solely to the first control circuit.
Status indication of controller		The controller can output its current operating status.
	No status	No status will be output.
	Controller general	The controller status is generally output by the 1-byte object (*"controller status"*).
	Transmit individual state	The controller status preset by the "individual state" parameter will be output via the 1-bit object (Object 36 *"status indication of controller status"*).
Single state	**Comfort operation activated** Standby operation activated Night operation activated Frost/heat protection active Controller disabled Heating/cooling Controller inactivated Frost alarm	Determines the controller status to be transmitted. Only if "status indication of controller" = "transmit individual state"!

Room temperature regulator function – room temperature timer		
Room temperature timer	On **Off**	Enables the room temperature timer.
Lock room temperature timer via object	 Yes **No**	The disable function can suppress the timer's switching programmes via the bus. Enables the disable function and the *"disabling room-temperature-timer"* object 55. The disable function of the heating timer is not enabled.
Polarity of blocking object	inverted (disable = 0) **Not inverted (disable = 1)**	Determines the polarity of the timer's disable object. Only if "Lock room temperature timer via object = "yes"!

Scene function		
Data type Output 1	**Switching** Value Shutter/blind position	Determines the data type for the scene output.
Value type	**0...100 %** 0...255	Defines the value type for 1-byte scene objects. Depending on this setting it is possible to enter percentage values or dimensionless values for the scene commands. Only if "data type" = "value"!
Data type Output 2 to 8	See data type output 1!	

Scene function – [1] Scene 1		
Name	[Text], **Scene 1**	A description for the internal scene can be entered at this point. This description is used exclusively for a better orientation in the ETS plug-in and will not be downloaded into the device.
Transmit output signal	Yes **No**	It is possible to determine – when recalling a scene – whether a scene command is to be transmitted via the selected scene output.
Value	**On** Off	Defines the switching value, which is transmitted on the bus during a scene recall. Only if “transmit output signal = yes” and “data type = switching”!
Value (0…100) * 1 %	0 to 100 %, **0 %**	Defines the value, which is transmitted on the bus during a scene recall. Only if “transmit output signal = yes” and “data type = value” and “value type = 0…100 %”!
Value (0…255)	0 … 255, **0**	Defines the value, which is transmitted on the bus during a scene recall. Only if “transmit output signal = yes” and “data type = value” and “value type = 0…255”!
Shutter/blind position (0…100) * 1 % (0 => up)	0 to 100 %, 0 %	Defines the shutter position value, which is transmitted on the bus during a scene recall. Only if “transmit output signal = yes” and “data type = shutter/blind position”!
Scene function – [X] Scene X, X = 2 to 8 see scene 1!		

Controller extension function		
Use fan stages display	**No**	To enable the fan stages display for a fan coil actuator (setting *"yes"*). When this function is enabled the associated communication objects will become visible.
	Yes	
Fan stage display object type		To define the feedback message data format of the fan coil actuator for fan stage display.
	3 x 1 bit	Feedback proceeds per stage via separate 1-bit objects.
	1-byte	Common feedback for all stages via a 1-byte object.

Controller extension function – room temperature measuring (FA)		
Temperature detection		Determines which sensor will be used for the room temperature measurement on the controller extension.
	Internal sensor	Internal sensor: Sensor installed inside the push button RTR.
	Internal and external sensor	Internal and external sensor: Both sensors are used, for example, in large rooms.
Creating of measuring value internal against external	10 % to 90 % 20 % to 80 % 30 % to 70 % 40 % to 60 % **50 % to 50 %** 60 % to 40 % 70 % to 30 % 80 % to 20 % 90 % to 10 %	Determines the weighting of the measured temperature value for the internal and external sensors. That results in an overall value, which will be used for the further interpretation of the room temperature. Only if "temperature detection = internal and external sensor"!
Adjustment internal sensor (-128...127) * 0.1 K	-128 ... 127, **0**	Determines the value by which the internal sensor's room temperature value is calibrated. Only if "temperature detection = internal sensor" or "internal and external sensor" or in case of two control circuits!
Adjustment external sensor (-128...127) * 0.1 K	-128 ... 127, **0**	Determines the value by which the external sensor's room temperature value is calibrated. Only if "temperature detection" = "internal and external sensor"!
Scanning time for external sensor (0...255) * 1 min; 0 = inactive	0 ... 255, **0**	Determines the measurement period for the external sensor's temperature value. "0" = sensor automatically transmits its temperature value. Only if "temperature detection" = "internal and external sensor"!

Transmission at room temperature modification by (0..255) * 0.1 K; 0 = no automatic transmission	0 ... 255, **3**	Determines the size of the value change of the room temperature after which the current values are automatically transmitted on the bus via the "actual temperature" object 23.
Cyclical transmission of room temperature (0...255) * 1 min; 0 = inactive	0 ... 255, **15**	Determines whether or when the determined room temperature of the first control circuit is to be periodically output via the "actual temperature" object 23.
Send temperature alarm via object	**No** Yes	This parameter enables the "temperature alarm" function if set to "yes".
Lower temperature limit value (0...40 °C)	0...40 °C; **7 °C**	Lower temperature value for temperature alarm. Only if "Send temperature alarm via object" = "yes"!
Upper temperature limit value (0...40 °C)	0...40 °C; **35 °C**	Upper temperature value for temperature alarm. Only if “Send temperature alarm via object” = “yes”!

Timer 1		
Name	[20-character text], **timer 1**	Here, you can assign a name timer 1. This name will also be downloaded into the device and indicated in the programming menu.
Data format	**Switching** Value Scene request	To specify whether switching or value or a scene recall commands shall be sent to the bus when the timer is active. You can define the commands themselves in the timer editor.
Blocking object (e. g. sun sensor)		The execution of the switching times of timer 1 can be suppressed via the bus by the disabling function.
	Yes	To enable the disabling function and object 55, *"disabling timer 1"*.
	No	The timer 1 disabling function is not enabled.
Polarity of blocking object	Inverted (disabling = 0) **Not inverted (disabling = 1)**	To specify the timer 1 disabling object polarity. Only for "blocking object = "yes".
For timer 2, refer to timer 1!		

Software information

- **Parameter access**

  In order to be able to set all the push button's parameters, the access must be set to "high access" in the ETS plug-in! In order to set the type of access the "high access" menu item in the "configuration" menu has to be selected or deselected.

- **Dimming function (push button functions)**

  For the status-LED to function properly for the status indication the connected dimming actuator has to transmit back its status to the switching object (key function) or to the status object (rocker function) (set "transmit" flag on actuator).
  In order to function properly when using the key function (brighter / darker (TOGGLE)) the connected dimming actuator must also transmit back its status to the switching object.
  With key or rocker function only the switching object is tracked internally and externally. The dimming object (direction) will only internally be tracked so that – when using extensions (2 or more push buttons reduce/increase brightness of one lamp) – the direction (+/-) is not always switched-over on pressing the push button again.
  The two-surface control (key function) requires the objects of the associated keys to be assigned the same group address.

- **Value transmitter function (push button functions)**

  When adjusting a value via a long operationthe newly set values will be stored in RAM only, i.e. these values will be replaced by the preset values – originally programmed via the ETS – following a voltage failure or bus reset.

- **Status indication (push button functions)**

  The status-LED (status indication) indicates the current status of the switching object (key function). If a key is actuated (e.g. ON) and if the push button does not get a positive acknowledgement (IACK) from an addressed actuator, the object status is updated and the corresponding status-LED lights up.

- **System requirements for the ETS plug-in**

  Operating system: Windows 9x, ME, Nt 4.0, 2000, XP

  ETS: ETS 2 V 1.2 a or higher, recommended from ETS 3.0d Patch A

  PC: recommended Pentium I-Processor (or similar), 166 MHz, 32 MB or higher

- **ETS functions**

  The "device info" or "device memory viewer" ETS functions are not possible for the B.IQ RTR push button.

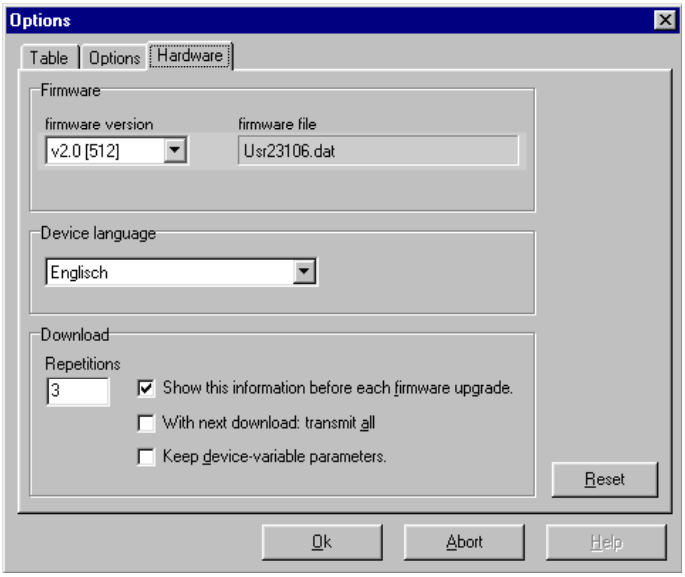
  Even executing the "shrink database" ETS function will lead to project data corruption for the B.IQ push button with room thermostat (RTR) and display up to including version V 1.3 and should be avoided under any circumstances. Install ServiceRelease "a" for the ETS2 V 1.3 as a remedy.

- **Firmware**

The B.IQ push button with room thermostat (RTR) and display 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, you can even bring earlier B.IQ RTR push buttons to the latest standard without having to replace the device. Only the B.IQ push button with room thermostat (RTR) and display software in the ETS must be up-to-date in this connection. 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 already factory-pre-programmed in the push button.

A firmware download takes a few minutes. During a firmware download, the display reads the message **"Firmware download - loading….."**.

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:



- Device language: The language (text read-outs on the display) of the B.IQ push button with room thermostat (RTR) and display device can be parametrized
- 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 automatically detect 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 B.IQ push button with room thermostat (RTR) and display versions containing 'inappropriate firmware.
  Later, you can reactivate this check box in the ETS plug-in from the opt. dialog on the "hardware" tab.
- If you have checked the "keep device-variable parameters" box the temperature set values (decrease/increase/standby, night, dead band, basic set value) of the first control circuit 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.