# Product documentation

KNX push-button universal / KNX push-button extension

Art. no. ..109.1ST/..109.1TE

ALBRECHT JUNG GMBH & CO. KG
Volmestraße 1
58579 Schalksmühle
GERMANY

Telefon: +49 2355 806-0
Telefax: +49 2355 806-204
kundencenter@jung.de
www.jung.de

Issue: 09.03.2020
TD 62975320

# Table of Contents


1 Information on the product ..... 3
1.1 Product catalogue ..... 3
1.2 Function ..... 3
1.3 Device components ..... 5
1.4 Delivery state ..... 6
1.5 Technical data ..... 7
1.6 Accessories ..... 7
2 Safety instructions ..... 8
3 Fitting and electrical connection ..... 9
4 Commissioning ..... 14
5 Operation ..... 20
6 Application programs ..... 21
7 Scope of functions ..... 22
8 Channel-oriented device functions ..... 25
8.1 Rockers and button functions ..... 26
8.1.1 Switching ..... 27
8.1.2 Dimming ..... 28
8.1.3 Venetian blind ..... 34
8.1.4 Value transmitter ..... 41
8.1.5 Scene extension ..... 56
8.1.6 2-channel operation ..... 58
8.1.7 Controller extension ..... 70
8.2 Status LED ..... 80
8.2.1 Basic functions ..... 82
8.2.2 Colour settings ..... 87
8.2.3 Brightness settings ..... 90
9 Channel-independent device functions ..... 107
9.1 Temperature measurement ..... 108
9.2 Alarm signal ..... 114
9.3 Disabling function ..... 118
9.4 Energy saving mode ..... 133

# 1 Information on the product

## 1.1 Product catalogue

| Product name | Article number | Use | Design |
|---|---|---|---|
| KNX push-button universal 1-gang | ..10911ST | Sensor | FM (Flush-mounted) |
| KNX push-button universal 2-gang | ..10921ST | Sensor | FM (Flush-mounted) |

## 1.2 Function

### General

The device is KNX Data Secure capable. KNX Data Secure offers protection against manipulation in building automation and can be configured in the ETS project. Detailed specialist knowledge is required. A device certificate, which is attached to the device, is required for safe commissioning. During mounting, it is recommended to remove the certificate from the device and to store it securely.

The device can be updated. Firmware updates can be carried out conveniently using the Jung ETS service App (additional software).

### Push-button sensor functions

When its buttons are actuated, the device sends telegrams to the KNX, depending on the ETS parameter settings. These can be telegrams for switching, for dimming or for controlling blinds. Value transmitters and scene extension functions can also be programmed. The value transmitter functions include, for example, temperature, brightness and colour temperature value transmitters.

The device can be used as a controller extension, i.e. as an operation and display element of a room temperature controller.

The operation concept can be configured in the ETS either as a rocker function or alternatively as a button function. With the rocker function, two buttons lying one above the other are combined into one rocker. In the button function, each button is evaluated as single-area operation. If two buttons are used as a rocker function, then, depending on the configuration, it is also possible to trigger special functions using full-surface rocker operation.

All buttons or single buttons of the device can be disabled using the disabling function. At the beginning of disabling, during active disabling or at the end of disabling, the assigned buttons perform parameterised behaviour.

### Push-button extension

Optionally, the number of buttons of the device can be expanded to include up to 4 additional buttons, by connecting an Push-button extension to the basic device. Configuration and commissioning of the Push-button extension is clearly structured and easy to perform using the application program of the basic device.

### LED function

The device possesses a three colour status LED per rocker. These status LEDs can either be switched on or off permanently, or can function as a status indicator for a button or rocker. As an alternative, the LEDs can also be activated via separate communication objects. The LEDs can either indicate the switching status of an object statically or by flashing signal operating states of room temperature controllers or indicate results of logical value comparison operations.

The colour of the status LED can either be parameterised for all status LEDs together or separately for each status LED. The colour of the status LED can also be set optionally for the 3-colour individual control via objects.

The brightness of all status LEDs is adjustable in six stages using a common parameter. A separate communication object allows the brightness to be reduced, e.g. during night hours.

A special function of the status LED is the alarm signal. The device can display an alarm by synchronously flashing all red status LEDs.

i If no application or a wrong application is loaded into the device, the status LED flashes blue. In this case the device is discharged and has no function.

### Measurement of the room temperature

As a supplement to the controller extension the device has an integrated temperature sensor that makes it possible to measure and forward the local room temperature. A temperature value received via an object can optionally supplement the room temperature measurement performed by the internal temperature sensor in order to improve the measurement result.

### Energy saving mode

The device has an energy-saving mode to save electrical energy during operation. If the function is used, the device switches to the energy saving mode after a preset time without operation or controlled by an external telegram to a separate object.

## 1.3 Device components

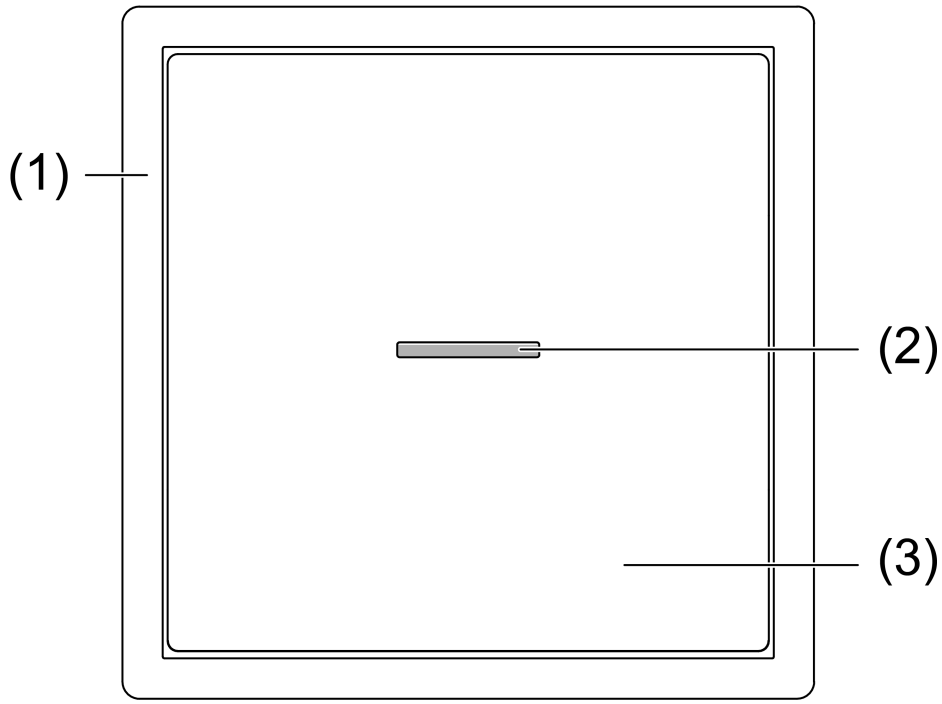


Image 1: Device components 1-channel

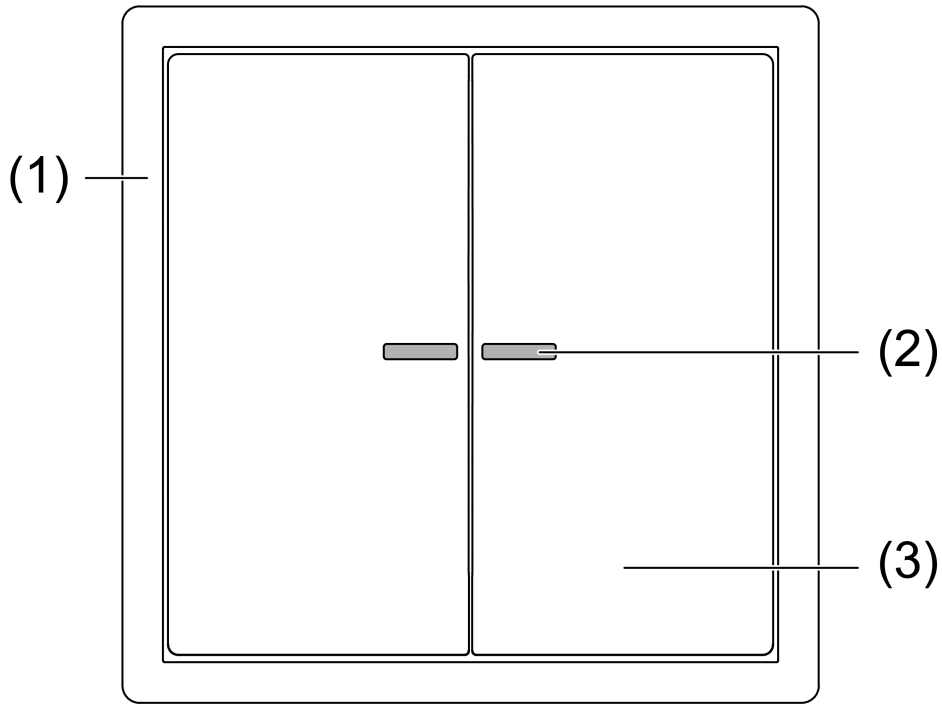


Image 2: Device components 2-channel

(1) Design frame (accessories)
(2) Status LED
(3) Operating rockers (accessory)

## 1.4 Delivery state

In the delivery state, the device is discharged. When the bus voltage is connected, the status LEDs signal this status by flashing blue.

The status LED belonging to the button is switched on when the button is pressed and switched off again when the button is released. Each time a button is pressed the colour of the status LED changes (red -> green -> blue). If no further operation takes place within 10 seconds, the corresponding status LED switches back to the signalling for the discharged state (flashing).

In the delivery state, the device does not send any telegrams to the bus.

## 1.5 Technical data

### KNX

| | |
|---|---|
| KNX medium | TP256 |
| Safety | X-mode |
| Commissioning mode | S-mode |
| Rated voltage | DC 21 ... 32 V SELV |
| Current consumption KNX | |
| without push-button extension | 4.9 ... 8.2 mA |
| without external switching contacts | 4.9 ... 8.2 mA |
| with push-button extension | 6.6 ... 9.4 mA |
| with external switching contacts | 7.7 ... 11.0 mA |
| Connection mode KNX | Standard device connection terminal |
| Connecting cable KNX | EIB-Y (St)Y 2x2x0.8 |
| Protection class | III |

### Push-button extension connection (see accessories)

| | |
|---|---|
| Number | 1 |
| Cable type | J-Y(St)Y 2×2×0.8 |
| Cable length | max. 30 m |

### External switching contact connection

| | |
|---|---|
| Number | max. 20 |
| Cable type | J-Y(St)Y 2×2×0.8 |
| Cable length | max. 30 m |

### Ambient conditions

| | |
|---|---|
| Ambient temperature | -5 ... +45 °C |
| Storage/transport temperature | -25 ... +70 °C |

## 1.6 Accessories

| | |
|---|---|
| KNX push-button extension 1-gang | Art. no. ..10911TE |
| KNX push-button extension 2-gang | Art. no. ..10921TE |
| Rocker for KNX push-button 1-gang | Art. no. ..101.. |
| Rocker for KNX push-button 2-gang | Art. no. ..102.. |

# 2 Safety instructions

Electrical devices may only be mounted and connected by electrically skilled persons.

Serious injuries, fire or property damage possible. Please read and follow manual fully.

These instructions are an integral part of the product, and must remain with the end customer.

# 3 Fitting and electrical connection

**DANGER!**

**Mortal danger of electric shock.**

- Cover up live parts in the installation environment.

**Mounting and connecting the device (see figure 3)**

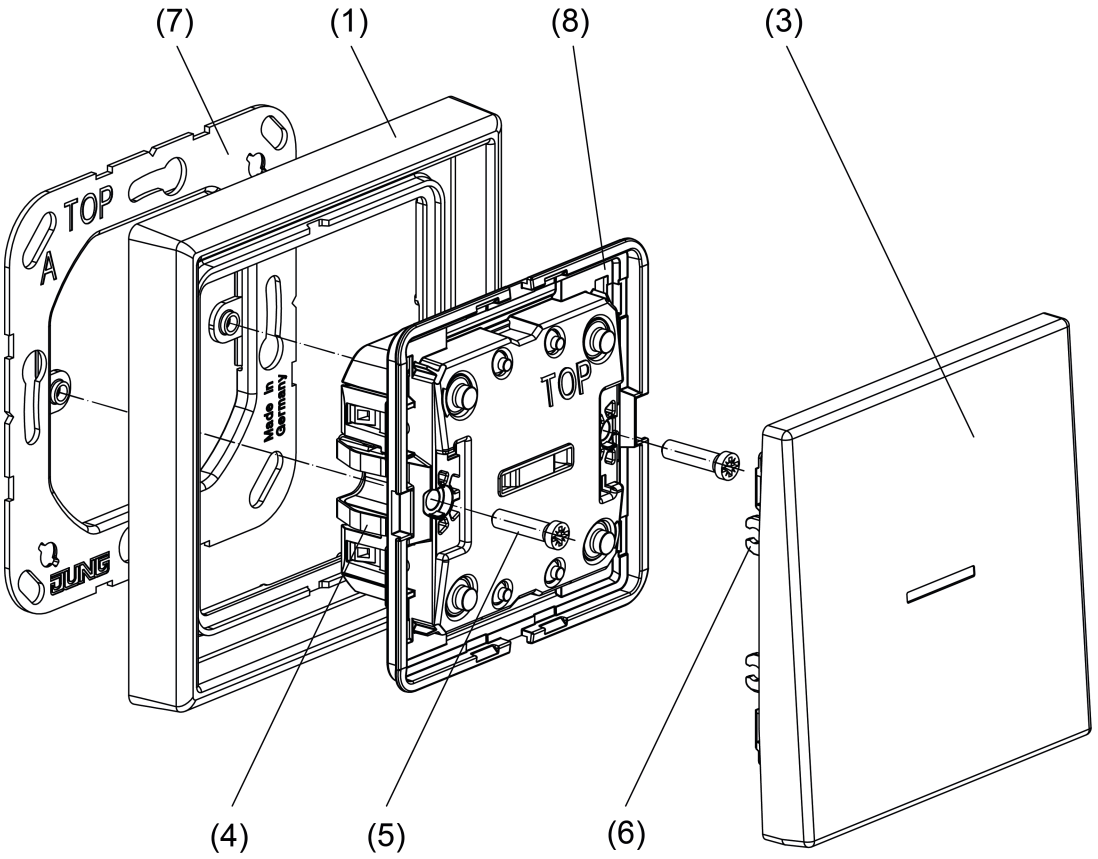


Image 3: Fitting the device

(1) Design frame
(3) Operating rocker (top = narrow side)
(4) Retaining spring
(5) Screw
(6) Retaining clamp
(7) Supporting frame
(8) Device module

**DANGER!**

Danger of electric shock

When mounting with 230 V socket outlets under a common cover, there is a danger of electrical shocks in the event of a fault.

- Use only the enclosed plastic screws for fastening to the supporting frame!

In secure operation (preconditions):

– Secure commissioning is activated in the ETS.

– Device certificate entered/scanned or added to the ETS project. A high resolution camera should be used to scan the QR code.

– Document all passwords and keep them safe.

- Mount supporting frame in the right orientation on an appliance box.

i Note marking TOP and marking A (= in front).

i Use the supplied box screws.

- Connect device to the KNX using KNX connecting terminal (red= +, black = -).
- Optional: Connect (see accessories) to connection terminal (see figure 6).
- Optional: Connect external switching contacts (e.g. push-button or reed contacts) to the connection terminal (see figure 7).
- In secure operation: It is advisable to remove the secure certificate from the device.
- Attach device onto the supporting frame.
- Optional: Screw the device to the supporting frame. Tighten the plastic screws only lightly.
- Snap on rockers.

i The rockers suitable for the device must be ordered separately (see accessories).

i The device should be used in an air-tight appliance box. Drafts cause incorrect temperature values to be measured.

**Dismantling the operating rockers (see figure 4)**

The operating rockers can be dismantled using a dismantling wedge or a screwdriver (see figure 5).

i The recesses are intended for a blade width of up to 4 mm.

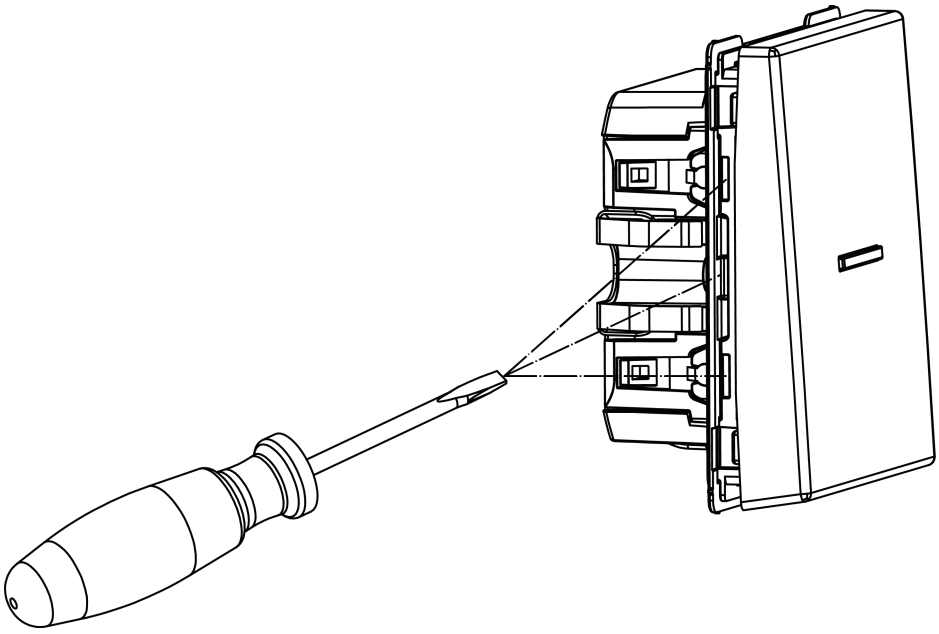

Image 4: Dismantling rocker from the side

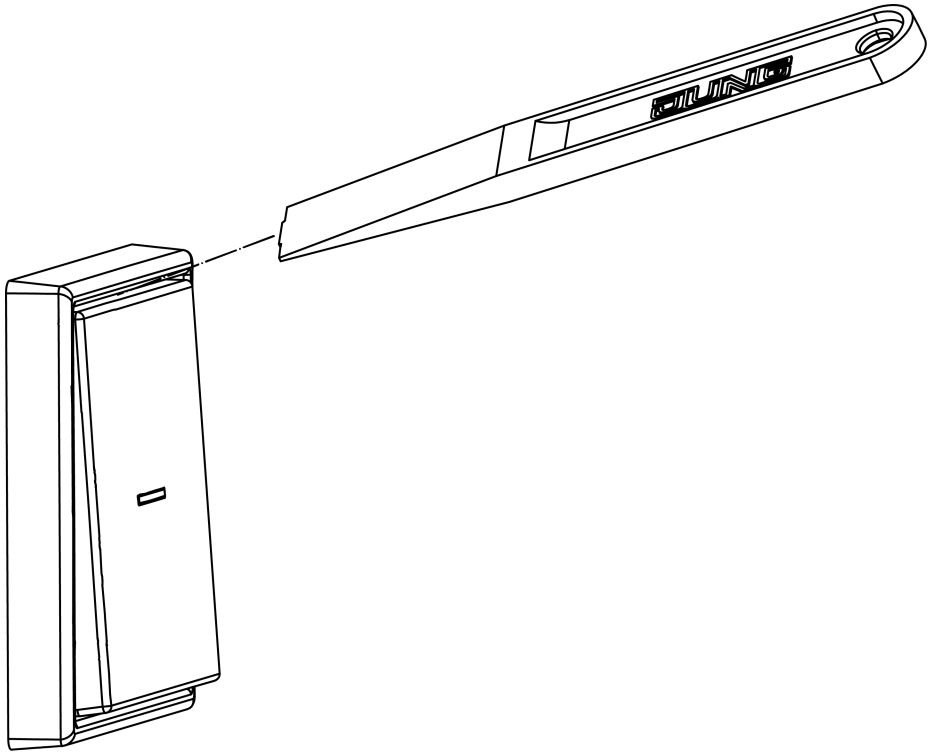

Image 5: Dismantling rocker from above

**connecting (see figure 6)**

i Only possible with Push-button universal.

- Connect to Push-button universal with a separate line (e.g. J-Y(St)Y 2×2×0.8).
  Note colour coding: black = SW, red = RT, white = WS, yellow = GE
  The connection terminal blocks are included with the .

i Do not exceed the cable length (max. 30 m).

One Push-button universal can be connected per .

Do not connect any external voltage.

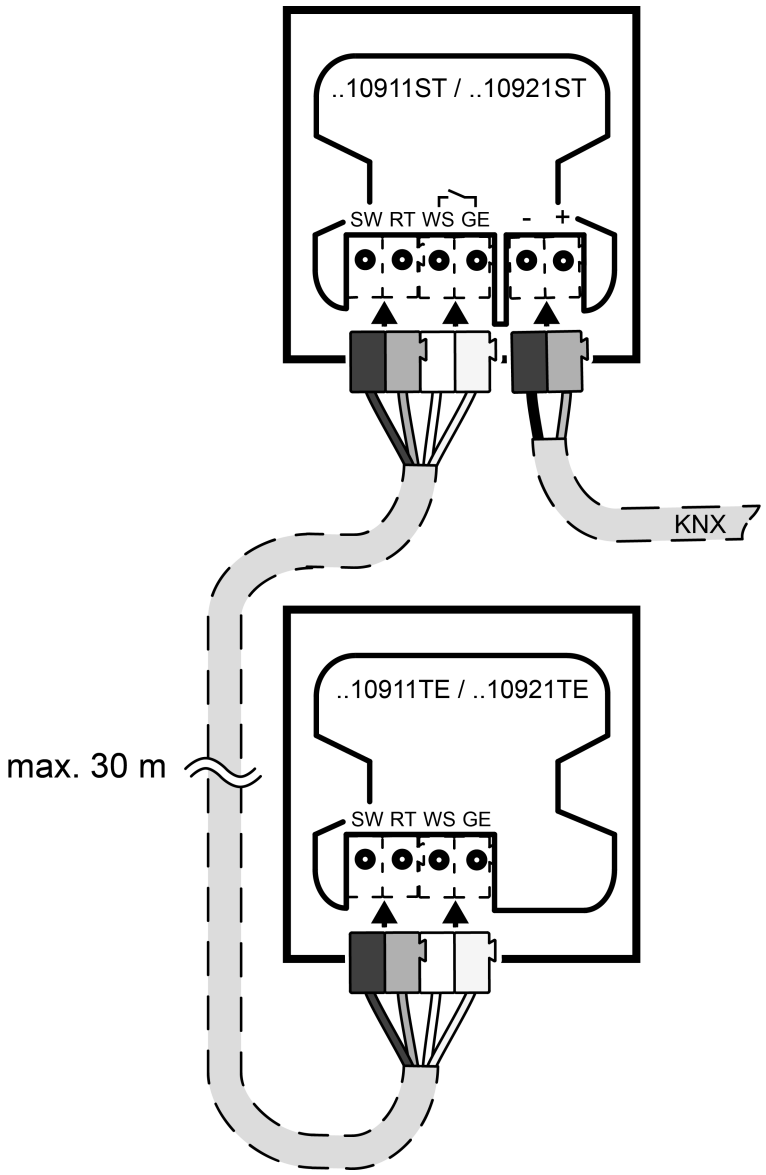


Image 6: Connecting push-button extension

**Connecting external switching contacts (see figure 7)**

Only possible with Push-button universal.

- Connect external switching contacts (e.g. push-button or reed contacts) with a separate line (e.g. J-Y(St)Y 2×2×0.8) to Push-button universal (connection terminal ⌐– / WS GE). A suitable device connection terminal is available as an accessory (2050GEWS).

Do not exceed the cable length (max. 30 m).

A maximum of 20 external switching contacts can be connected in parallel or in series.

Do not connect any external voltage.

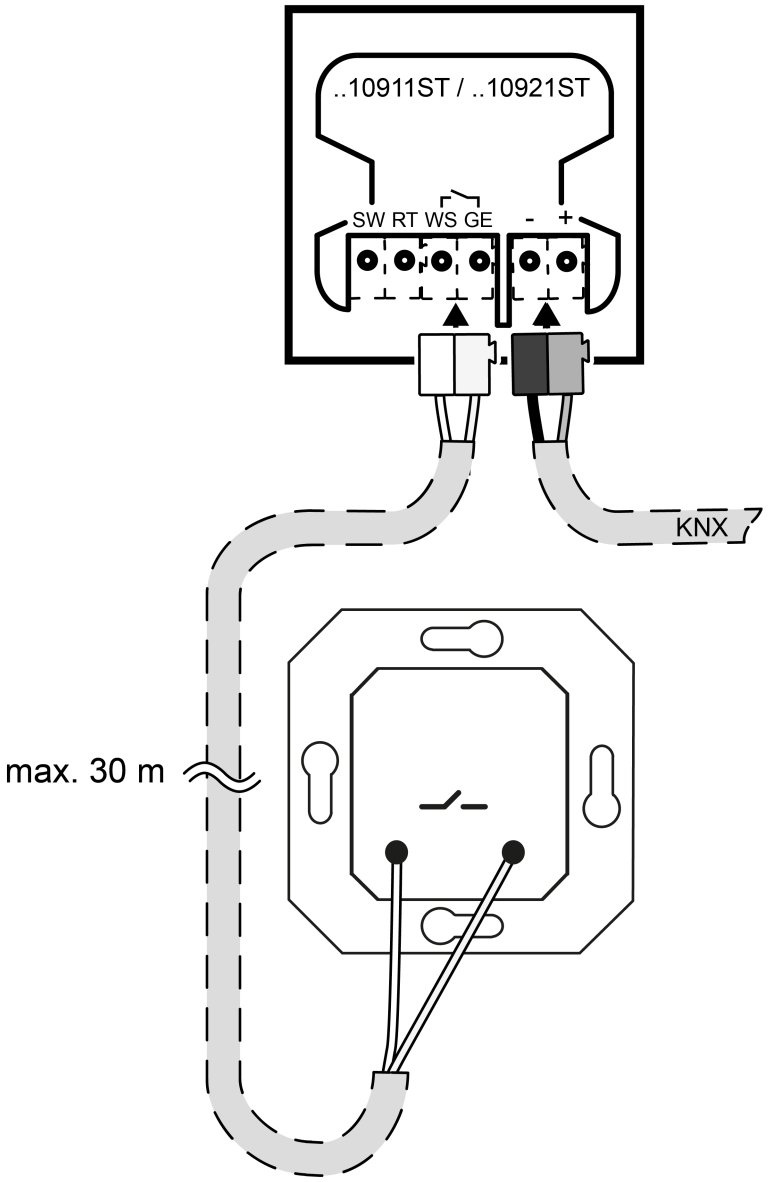


Image 7: Connecting installation push-button

# 4 Commissioning

After connection and mounting, the device can be commissioned. The commissioning is basically confined to programming with the ETS and attaching the decorative operating areas.

i The does not receive any physical address of its own. It is activated by the loaded application program of the basic module.

### Preconditions in secure operation

- A dedicated application program is required.
- Secure commissioning is activated in the ETS.
- Device certificate entered/scanned or added to the ETS project. A high resolution camera should be used to scan the QR code.
- Document all passwords and keep them safe.

### Programming the physical address

The device does not have a separate programming button or LED. Programming mode is activated by a defined and time-delayed press of the push-buttons at the upper left (9) and lower right (10). The active programming mode is signaled on the base module by the rapid flashing (approx. 8 Hz) of the blue status LED. To program the physical address, the decorative control surfaces can be snapped onto the device.

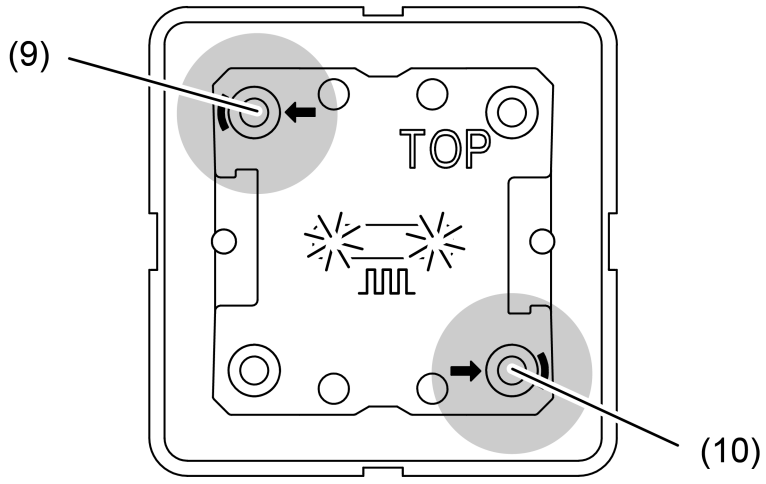


Image 8: Buttons for activating Programming mode

i If the device does not contain an application program, or contains an incorrect one, the blue status LEDs flash slowly (approx. 0.75 Hz).

i Activation of the programming mode without attached rockers.

Precondition: The device is connected and ready for operation.

- Activate programming mode: Press and hold push-button at the upper left (9). Then press push-button at the lower right (10) (see figure 8).

  The status LEDs flash rapidly (approx. 8 Hz). Programming mode is activated.

Use suitable objects to push the buttons (e.g. thin screwdriver, tip of a ball-point pen, etc.)

To exclude any inadvertent activation of Programming mode during 'normal' use of the control surface in later operation, the time between the first and the second actuation must be at least 200 ms. Pressing simultaneously (time between first and second actuation < 200 ms) will not result in an activation of Programming mode!

- Load the physical address into the device.

  The status LEDs return to their previous state – off, on, or flashing. Physical address is programmed.

- Write the physical address on the device label.
- Programming mode ends:
  - Automatically after adoption of the physical address
  - By pressing any button on the basic module

If Programming mode is to be activated or deactivated in a device which is already programmed with a valid application, there is the possibility that telegrams will be transmitted to the bus at the time the button is pressed. The telegram transmitted depends on the button function programmed.

The does not receive any physical address of its own. The programming mode cannot be activated or deactivated on the extension.
When programming mode is active, the can be operated normally. Then the telegrams corresponding to the project design are also transmitted to the bus.

**Programming the application program**

Afterwards, the application program has to be programmed into the device using the ETS. Project design and commissioning with ETS from version5.7.3and above.

The ETS detects automatically whether a valid application has already been programmed into the device before. To reduce the programming time, the ETS downloads the whole application only if the device was programmed beforehand with another application or with no application at all. In all other cases, the ETS makes a time-optimised partial download in which only the modified data is loaded into the device.

## 4.1 Safe-state mode

The safe state mode stops the execution of the loaded application program. A master reset can be performed if the device is not functioning correctly, for example by:

– faulty project design or
– faulty commissioning

i Only the system software of the device is still functional. ETS diagnosis functions and programming of the device are possible.

### Activating the safe-state mode

- Switch off the bus voltage.
- Press and hold down the top left and bottom right button.
- Switch on the bus voltage.

  The safe-state mode is activated. The status LEDs flash slowly (approx. 1 Hz).

i Only release the the top left and bottom right buttons when the LEDs flash.

### Deactivating safe-state mode

- Switch off bus voltage or carry out ETS programming.

## 4.2 Master reset

The master reset deletes the loaded application program from the device. A master reset can be performed if the device is not functioning correctly, for example by:

– faulty project design or
– faulty commissioning

The master reset restores the basic device setting (physical address 15.15.255, firmware remains in place). The device must then be recommissioned with the ETS.

During secure operation: A master reset deactivates device security. The device can then be recommissioned with the device certificate.

**Performing a master reset**

Precondition: The safe-state mode is activated.

- Press and hold down the top left and bottom right button for > 5 seconds.

  The device performs a master reset. The status LEDs flash rapidly (approx. 4 Hz).

  The device restarts. The status LEDs flash slowly (approx. 0.75 Hz).

Only release the the top left and bottom right buttons when the LEDs flash.

## 4.3 Restoring the device to factory settings

Devices can be reset to factory settings with the ETS Service App. This function uses the firmware contained in the device that was active at the time of delivery (delivery state). Restoring the factory settings causes the devices to lose their physical address and configuration.

## 4.4 Flashing sequence of the status LEDs

| Status LED status | Flash frequency |
|---|---|
| Application discharged / stopped | approx. 0.75 Hz |
| Safe-state mode | approx. 1 Hz |
| Status display flashes | approx. 2 Hz |
| Alarm signal | approx. 2 Hz |
| Master reset | approx. 4 Hz |
| Programming mode | approx. 8 Hz |
| Full-surface operation | approx. 8 Hz |

# 5 Operation

## Operating areas

The device consists of up to 2 operating areas, depending on the device variant. The operating concept of an operating area can be configured in the ETS as a rocker function or as a button function.

With the rocker function, an operating area is divided into two buttons (top/bottom) with the same basic function.

In the button function either an operating area is divided into 2 functionally separate buttons (double-area operation), or an operating area is evaluated as single-area operation (only one button).

If the rocker function is parameterised, special functions can be triggered using full-surface operation. These special functions are available for the dimming, Venetian blind and 2-channel operation rocker functions.

Optionally, the number of operating areas can be expanded to include up to 2 additional operating areas, by connecting an to the basic device. Configuration and commissioning of the is clearly structured and easy to perform using the application program of the basic device.

Centrally on each operating area, there is a status LED, which can be connected to the control function, according to the function of the rocker or buttons. A status LED can also signal completely independent display information, flash or be permanently on or off. Besides functions that can be set using the ETS, the status LED also indicates that the device is in the programming mode for commissioning or diagnosis purposes.

# 6 Application programs

| | |
|---|---|
| ETS search paths: | Push-button/ push-button, 1-gang/ Push-button universal 1-gang |
| | Push-button/ push-button, 2-gang/ Push-button universal 2-gang |
| Configuration: | S-mode standard |

### Application program available for Push-button universal 1-gang

| | |
|---|---|
| Name | Push-button universal 1-gang 115111 |
| Version: | 1.1 for ETS version 5.7.3 onwards |
| from mask version | 07B0 |
| Summarized description | Multifunctional push-button sensor application. |
| | 1 rockers or 2 push-buttons on the basic device Can be expanded to 3 rockers or 2 buttons |
| | Temperature measurement, alarm signalling, disabling function and energy saving mode can be activated. |

### Application program available for Push-button universal 2-gang

| | |
|---|---|
| Name | Push-button universal 2-gang 115211 |
| Version: | 1.1 for ETS version 5.7.3 onwards |
| from mask version | 07B0 |
| Summarized description | Multifunctional push-button sensor application. |
| | 2 rockers or 4 push-buttons on the basic device Can be expanded to 4 rockers or 8 buttons |
| | Temperature measurement, alarm signalling, disabling function and energy saving mode can be activated. |

# 7 Scope of functions

## General

- KNX Data Secure capable
- Firmware updates are possible
- The number of control surfaces can be expanded using Push-button extension.

## Push-button sensor functions

- Operating concept can be configured (rocker function or as a button)
- Button evaluation for button function concept can be configured (single-area or double-area operation)
- Button function can be configured (switching, dimming, Venetian blind, value transmitter, scene extension, 2-channel operation and controller extension, no function)
- Rocker function can be configured (switching, dimming, Venetian blind, value transmitter, scene extension, 2-channel operation and controller extension, no function)

  Switching: The command on pressing and/or releasing is adjustable (no reaction, switch on, switch off, toggle).

  Dimming: The command on pressing, the time between switching and dimming, the dimming in different levels, the telegram repetition on long press and the transmission of a stop telegram at the end of the press is adjustable.

  Venetian blind: The command on pressing and the operating concept is adjustable. The operating concept can be adjusted in the times for short and long actuation and slat adjustment.

  Value transmitter: The operating mode (1-byte, 2-byte, 3-byte or 6-byte value transmitter) and the value is adjustable. The value adjustment can optionally be activated by long button-press

  Scene extension: The operating mode (with or without storage function) and the scene number is adjustable.

  2-channel operation: Up to two telegrams can be transmitted to the KNX by one button-press. The operating concept can be adjusted and the time for short and long actuation adapted. The function of the channels is adjustable separately.

  Controller extension: The function (operating mode selection, forced operating mode switch over, presence button and setpoint shift) is adjustable.
- Full-surface operation can be configured(only for the rocker functions Dimming, Venetian blind (in "Long – Short or Short" operating concept) and 2-channel operation ("channel 1 or channel 2 in operating concept)

  With full-surface operation, switching telegrams and scene recall requests can be triggered on the KNX in addition to and independently of the configured rocker function.
- Disabling function can be activated

The rockers or buttons can be disabled via a 1-bit object. The following settings are possible: polarity of the disabling object, behaviour at the beginning and at the end of disabling. During an active disable, all or some of the rockers / buttons can have no function, can perform the function of a selected button or execute one of two presettable disabling functions.

### Status LED

– Function can be configured

  The function is selected either per status LED or by 3-colour individual control via objects.

  When selecting the function for each status LED, the following functions can be configured: Always ON, Button-press display, Telegram acknowledgment, Status display, Inverted status display, Activation via separate LED object, Operating mode display, Controller status display, Presence status, Setpoint value shift display, Comparator without/with sign, Logic link, Bit-coded evaluation

– Colour can be configured

  The colour is selected either together for all status LEDs or separately for each status LED of the device.

  The status LED can light up in red, green or blue according to choice. An automatic colour change is also possible depending on the function. When switched-off, the status LEDs can light up in a designed colour, creating an orientation light.

– Brightness can be configured

  The brightness of the status LED can be set to five levels.

  Through night reduction, the brightness of the status LED can be reduced at night using a communication object.

– Alarm signal can be activated

  All status LEDs of the device can flash simultaneously in the event of an alarm. The following settings are possible: Value of alarm signalling object for the states alarm / no alarm, alarm acknowledge by actuation of a button, transmission of the acknowledge signal to other devices.

– Energy saving mode can be activated

  If the energy-saving mode is activated, all LEDs are switched off. The following settings are possible: activate energy saving mode, deactivate energy saving mode, polarity of the object.

### Controller extension functions

– The controller extension can be configured as a function of a rocker or button.

  Full control of a room temperature controller (operating modes, presence functions and setpoint shift).

– The displays of the controller extension can be configured as a function of the status LED

Full-featured indication of the controller status via the status LED of the extension (heating / cooling reporting, setpoint shift, room temperature, setpoint temperature and current operating mode).

– Temperature measurement can be activated

  Measurement of the room temperature with an internal sensor or optionally by determining the measured value of the internally measured temperature with an external temperature.

# 8 Channel-oriented device functions

The following subchapters provide a description of the device functions. Each subchapter consists of the following sections:

- Functional description
- Table of parameters
- Object list

### Functional description

The functional description explains the function and provides helpful tips on project design and usage of the function. Cross references support you in your search for further information.

### Table of parameters

The table of parameters lists all parameters associated with the function. Each parameter is documented in a table as follows.

| Name of the parameter | Parameter values |
|---|---|
| Parameter description | |

### Object list

The object list specifies and describes all communication objects associated with the function. Each communication object is documented in a table.

| | |
|---|---|
| Object no. | This column contains the object number of the communication object. |
| Function | This column contains the function of the communication object. |
| Name | This column contains the name of the communication object. |
| Type | This column contains the length of the communication object. |
| DPT | This column assigns a datapoint type to a communication object. Datapoint types are standardized in order to ensure interoperability of KNX devices. |
| Flag | This column assigns the communication flags in accordance with the KNX specification. |
| K flag | activates / deactivates the communication of the communication object |
| L flag | enables externally triggered reading of the value from the communication object |
| S flag | enables externally triggered writing of the value to the communication object |
| Ü flag | enables transfer of a value |
| A flag | enables updating of an object value in case of feedback |
| I flag | enforces updating of the communication object value when the devices is switched on (reading at init) |

## 8.1 Rockers and button functions

The various functions that can be configured in the ETS for each rocker or button on the device are described below. The functions can be parameterised for the basic device as well as for the push-button extension.

### Table of parameters

The following parameters are available on the "General" parameter page.

| | |
|---|---|
| Push-button extension | Active<br>**Inactive** |
| If a push-button extension is connected to the basic device, it must be activated at this point.<br>A push-button extension can only be activated if a push-button extension is also connected to the basic device. | |

| | |
|---|---|
| Type of push-button extension | 1-gang<br>2-gang<br>Installation button |
| The type of connected push-button extension is specified. The associated communication objects and parameter groups in the ETS are shown corresponding to this setting. | |

### 8.1.1 Switching

For each rocker or button with the function set to "switching", the ETS indicates two 1-bit communication objects. The parameters permit fixing the value the "switching" object is to assume on pressing and/or releasing (ON, OFF, TOGGLE – toggling of the object value). No distinction is made between a brief or long press.

#### 8.1.1.1 Table of parameters

The following parameters are available for the individual buttons, depending on the set operating concept. The default settings change in accordance with the set operating concept.

| | |
|---|---|
| Command on pressing | no reaction<br>ON<br>OFF<br>**TOGGLE** |
| This parameter defines the reaction when the button is pressed. | |

| | |
|---|---|
| Command on releasing | **no reaction**<br>ON<br>OFF<br>TOGGLE |
| This parameter defines the reaction when the button is released. | |

#### 8.1.1.2 Object list

The following communication objects are available for the individual rockers or buttons, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 37, 40, ..., 58 | Switching | Button/rocker *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmission of switching telegrams (ON, OFF). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 38, 41, ..., 59 | Switching feedback | Button/rocker *n* - input | 1-bit | 1,001 | C, -,W, -, U |
| 1-bit object for receiving feedback telegrams (ON, OFF). | | | | | |

## 8.1.2 Dimming

For each rocker or button with the function set to "dimming", the ETS indicates two 1-bit and a 4-bit object. Generally, the device transmits a switching telegram after a brief press and a dimming telegram after a long press. In the standard parameterisation the device transmits a telegram for stopping the dimming action after a long press. The time needed by the device to detect an actuation as a long actuation can be set in the parameters.

The "Dimming" function distinguishes between double-area operation (UP, DOWN) and single-area operation (TOGGLE). The parameter "Command on pressing" defines the single-surface or double-surface dimming function.

| dual-area operation | single-area operation |
|---|---|
| Brighter (ON) | Brighter / darker (TOGGLE) |
| Darker (OFF) | Brighter (TOGGLE) |
| | Darker (TOGGLE) |

With double-area operation, the device transmits a telegram for switching on or off after a brief press, and a telegram for increasing the brightness ("Brighter") or decreasing the brightness ("Darker") after a long press.

With single-area operation, the device transmits ON and OFF telegrams in an alternating pattern ("TOGGLE") for each brief press, and the "brighter" and "darker" telegrams in an alternating pattern for long actuations.

### Feedback

If an actuator is controlled from multiple control elements, the actuator must report its switching status to the 1-bit object "Switching feedback" of the button or rocker. Due to the feedback, the device detects that the actuator has changed its switching status by input from another element and adjusts the dimming direction accordingly.

The dimming direction is always only evaluated and switched locally, unless the actuator changes its switching status due to input from multiple elements (e.g. lighting ON / change of brightness value only). The 4-bit dimming objects are not synchronized via the bus.

### Advanced configuration options

The device has advanced parameters for the dimming function. If necessary, these advanced parameters can be activated and thus be made visible.
The advanced parameters can be used to determine whether the device is to cover the full adjusting range of the actuator with one dimming telegram continuously ("Increase brightness by 100 %", "Reduce brightness by 100 %") or whether the dimming range is to be divided into several small levels (50 %, 25 %, 12.5 %, 6 %, 3 %, 1.5 %).

In the continuous dimming mode (100%), the device transmits a telegram only at the beginning of the long press to start the dimming process and generally a stop telegram after the end of the press. For dimming in small levels it may be useful if the

device repeats the dimming telegram in case of a sustained press for a presettable time (parameter "Telegram repetition"). The stop telegram after the end of the press is then not needed.

i When the parameters are hidden ("Advanced parameters = deactivated"), the dimming range is set to 100 %, the stop telegram is activated and the telegram repetition is deactivated.

**Full-surface operation with the dimming function**

When a rocker is used for dimming, the device needs some time at the beginning of each operation in order to distinguish between a short and a long operation. When full-surface operation is enabled, the device can make use of this time span to evaluate the otherwise invalid simultaneous actuation of both buttons of the rocker.

Full-surface operation of a rocker switch is detected by the device when both buttons are pressed at the same time. When the device has detected a valid full-surface actuation, the status LED flashes quickly at a rate of about 8 Hz for the duration of the actuation. Full-surface operation must have been detected before the first telegram has been transmitted by the dimming function (switching or dimming). If this is not so (e.g. one of the two buttons is pressed too late), the full-surface operation will not be correctly executed.

A full-surface operation is independent. It has a communication object of its own an can optionally be used for switching (ON, OFF, TOGGLE – toggling of the object value) or for scene recall without or with storage function. In the last case, the full-surface actuation on causes a scene to be recalled in less than a second. If the device is to send the telegram for storing a scene, full-surface actuation must be maintained for more than five seconds. If full-surface actuation ends between the first and the fifth second, the device will not send any telegrams.

### 8.1.2.1 Table of parameters

The following parameters are available for the individual buttons, depending on the set operating concept. The default settings change in accordance with the set operating concept.

| | |
|---|---|
| Command on pressing | No function*<br>Brighter (ON)<br>Darker (OFF)<br>**Brighter / darker (TOGGLE)**<br>Brighter (TOGGLE)<br>Darker (TOGGLE) |
| This parameter defines the reaction when a button is pressed.<br>If the device is to toggle on a brief press, the corresponding switching objects of other sensors with the same function must be interlinked.<br>* Only for operation concept = button function | |

| | |
|---|---|
| Time between switching and dimming | **0** ... 59 s \| 100 ... **400** ... 990 ms |
| This parameter defines how long the button must be pressed for a dimming telegram to be transmitted. | |

| | |
|---|---|
| Advanced parameters | Active<br>**Inactive** |
| When the advanced parameters are activated, the ETS shows the following parameters. | |

| | |
|---|---|
| Increase brightness by | 1.5 %<br>3 %<br>6 %<br>12.5 %<br>25 %<br>50 %<br>**100 %** |
| This parameter sets the relative dimming level when the brightness is increased. On each button-press, the brightness is changed at maximum by the configured step width.<br>It is recommendable, particularly with a smaller dimming level, when the device repeats the dimming telegrams automatically (see "telegram repetition"). | |

| Reduce brightness by | 1.5 %<br>3 %<br>6 %<br>12.5 %<br>25 %<br>50 %<br>**100 %** |
|---|---|
| This parameter sets the relative dimming level when the brightness is reduced. On each button-press, the brightness is changed at maximum by the configured step width.<br>It is recommendable, particularly with a smaller dimming level, when the device repeats the dimming telegrams automatically (see "telegram repetition"). | |

| Stop telegram | **Active**<br>Inactive |
|---|---|
| On "Active" the device transmits a telegram for stopping the dimming process when the button is released.<br><br>i When the device transmits telegrams for dimming in smaller levels, the stop telegram is generally not needed. | |

| Telegram repetition | Active<br>**Inactive** |
|---|---|
| This parameter can be used to activate telegram repetition for dimming. With telegram repetition activated, the device cyclically sends relative dimming telegrams (in the parameterised step width) to the bus if the button is pressed long. | |

| Time between two telegrams | **200 ms**<br>300 ms<br>400 ms<br>500 ms<br>750 ms<br>1 s<br>2 s |
|---|---|
| This parameter defines the interval at which the dimming telegrams are automatically repeated in the telegram repetition mode.<br>This parameter is only visible if "Telegram repetition = active"! | |

| Full-surface operation | Active<br>**Inactive** |
|---|---|
| When the full-surface operation is activated, the ETS shows the following parameters. | |

| Function for full-surface operation | **Switching**<br>Scene recall without storage function<br>Scene recall with storage function |
|---|---|
| In case of full-surface operation, this parameter defines the function that is to be used. The ETS shows the corresponding communication object and the other parameters.<br>This parameter is only visible if "Full-surface actuation = Active"! | |

| Command for full-surface operation | ON<br>OFF<br>**TOGGLE** |
|---|---|
| This parameter defines the value of the transmitted telegram when a full-surface operation has been sensed. "TOGGLE" changes over the current object value.<br>This parameter is only visible if "function for full-surface operation = Switching"! | |

| Scene number (1 … 64) | **1**, 2, ..., 64 |
|---|---|
| This parameter defines the scene number, which is to be transmitted to the bus after a scene recall or during storage of a scene.<br>This parameter is only visible if "function for full-surface operation = scene recall ..."! | |

### 8.1.2.2 Object list

The following communication objects are available for the individual buttons or rockers, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 73, 77, ..., 101 | Switching | Button/rocker *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmission of switching telegrams (ON, OFF). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 74, 78, ..., 102 | Dimming | Button/rocker *n* - output | 4-bit | 3,007 | C, -, -, T, - |
| 4-bit object for the transmission of relative dimming telegrams. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 75, 79, ..., 103 | Switching feedback | Button/rocker *n* - input | 1-bit | 1,001 | C, -,W, -, U |
| 1-bit object for receiving feedback telegrams (ON, OFF). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 67, 70 | Switching (full-surface operation) | Button/rocker *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for the transmission of switching telegrams (ON, OFF) when there is full-surface operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 68, 71 | Switching feedback (full-surface operation) | Button/rocker *n* - input | 1-bit | 1,001 | C, -,W, -, U |
| 1-bit object for receiving feedback telegrams (ON, OFF) for full-surface operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 244, 247 | Scene extension (full-surface operation) | Button/rocker *n* - output | 1-bit | 1,001 | C, -, -, T, - |
| 1-byte object for recalling or for storing one of 64 scenes max. from a scene push-button sensor in case of full-surface operation. | | | | | |

### 8.1.3 Venetian blind

For each rocker or button with the function set to "Venetian blind" the ETS indicates the two 1-bit objects "STEP operation" and "MOVE operation".

The "Venetian blind" function distinguishes between double-area operation (UP, DOWN) and single-area operation (TOGGLE). The "Command on pressing" parameter defines the single-area or double-area blind function.

| dual-area operation | single-area operation |
|---|---|
| UP | TOGGLE |
| DOWN | |

With an operating area as a rocker, the double-surface Venetian blind function is preset. This means that the device e.g. with a press of the top button, transmits a telegram for an upward movement and, after a press of the bottom button, transmits a telegram for a downward movement.

In the separate buttons function, the device is preprogrammed for single-surface Venetian blind function. In this case, the device alternates between the directions of the long time telegram (TOGGLE) on each long actuation of the sensor. Several short time telegrams in succession have the same direction.

#### Feedback

If the actuator can be controlled from several sensors, a faultless single-area operation requires that the long time objects of the control elements are interlinked. The device would otherwise not be able to detect that the actuator has been addressed from another sensor, in which case it would have to be actuated twice during the next use in order to produce the desired reaction.

#### Operation concept for the Venetian blind function

For the control of Venetian blind, roller shutter, awning or similar drives, the device supports four operation concepts in which the telegrams are transmitted in different time sequences. The device can therefore be used to operate a wide variety of drive configurations.

Operation concept "short – long – short"
In the operation concept "short – long – short", the device shows the following behaviour:

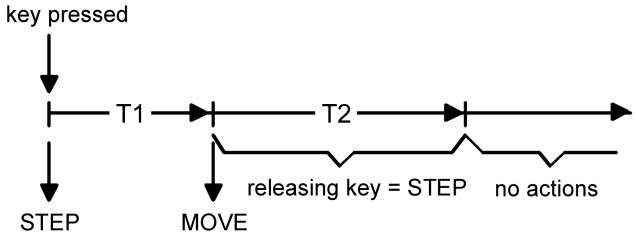


Image 9: Operation concept "short – long – short"

– Immediately on pressing the button, the device transmits a short time telegram. Pressing the button stops a running drive and starts time T1 ("time between short time and long time command"). No other telegram will be transmitted, if the key is released within T1. This short time serves the purpose of stopping a continuous movement.
  The "time between short and long time command" in the device should be selected shorter than the short-time operation of the actuator to prevent a jerky movement of the blind.
– If the button is kept depressed longer than T1, the push-button transmits a long time telegram after the end of T1 for starting up the drive and time T2 ("slat adjusting time") is started.
– If the button is released within the slat adjusting time, the device sends another short time telegram. This function is used for adjusting the slats of a blind. The function permits stopping the slats in any position during their rotation.
  The "slat adjusting time" should be chosen as required by the drive for a complete rotation of the slats. If the "slat adjusting time" is selected longer than the complete travelling time of the drive, a pushbutton function is possible as well. This means that the drive is active only when the button is kept depressed.
– If the button is kept depressed longer than T2, the device transmits no further telegram. The drive remains on until the end position is reached.

Operation concept "long – short":
In the operation concept "long – short", the device shows the following behaviour:

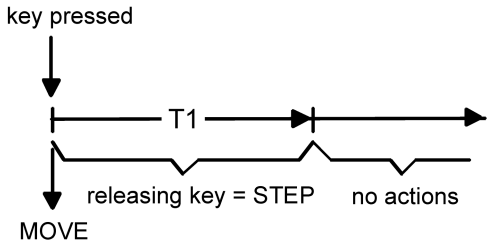


Image 10: Operation concept "long – short"

– Immediately on pressing the button, the device transmits a long time telegram. The drive begins to move and time T1 ("slat adjusting time") is started.
– If the button is released within the slat adjusting time, the device sends a short time telegram. This function is used for adjusting the slats of a blind. The function permits stopping the slats in any position during their rotation.
  The "slat adjusting time" should be chosen as required by the drive for a complete rotation of the slats. If the "slat adjusting time" is selected longer than the complete travelling time of the drive, a pushbutton function is possible as well. This means that the drive is active only when the button is kept depressed.
– If the button is kept depressed longer than T1, the device transmits no further telegram. The drive remains on until the end position is reached.

Operation concept "short – long"
In the operation concept "short – long", the device shows the following behaviour:

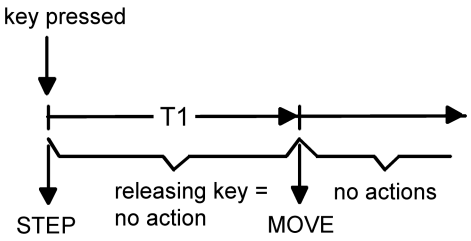


Image 11: Operation concept "short – long"

– Immediately on pressing the button, the device transmits a short time telegram. Pressing the button stops a running drive and starts time T1 ("time between short time and long time command"). No other telegram will be transmitted, if the key is released within T1. This short time serves the purpose of stopping a continuous movement.
  The "time between short and long time command" in the device should be selected shorter than the short-time operation of the actuator to prevent a jerky movement of the blind.
– If the button is kept depressed longer than T1, the push-button transmits a long time telegram after the end of T1 for starting the drive.
– No further telegram is transmitted when the button is released. The drive remains on until the end position is reached.

Operation concept "long – short or short":
In the operation concept "long – short or short", the device shows the following behaviour:

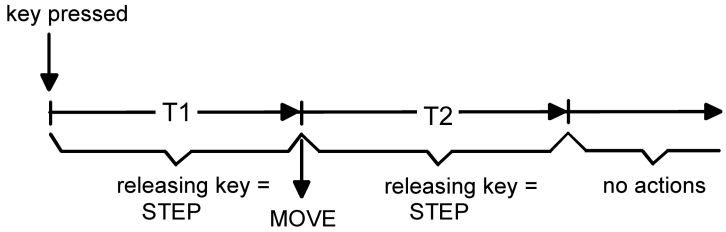


Image 12: Operation concept "long – short or short"

– Immediately on pressing the button, the device starts time T1 ("time between short time and long time command") and waits. If the button is released again before T1 has elapsed, the device transmits a short time telegram. This telegram can be used to stop a running drive. A stationary drive rotates the slats by one level.
– If the button is kept depressed after T1 has elapsed, the device transmits a long time telegram and starts time T2 ("slat adjusting time").
– If the button is released within T2, the device sends another short time telegram. This function is used for adjusting the slats of a blind. The function permits stopping the slats in any position during their rotation.
  The "slat adjusting time" should be chosen as required by the drive for a complete rotation of the slats. If the "slat adjusting time" is selected longer than the complete travelling time of the drive, a pushbutton function is possible as well. This means that the drive is active only when the button is kept depressed.

– If the button is kept depressed longer than T2, the device transmits no further telegram. The drive remains on until the end position is reached.

i In this operation concept, the device will not transmit a telegram immediately after depressing one side of the rocker. This principle permits detecting a full-surface operation when the sensor is configured as a rocker.

**Full-surface operation with Venetian blind function**

If a rocker is configured for Venetian blind operation and the operation concept "long – short or short" is used, the device needs some time at the beginning of each operation in order to distinguish between a short and a long operation. When full-surface operation is enabled, the device can make use of this time span to evaluate the otherwise invalid simultaneous actuation of both buttons of a rocker.

Full-surface operation of a rocker switch is detected by the device when both buttons are pressed at the same time. When the device has detected a valid full-surface actuation, the status LED flashes quickly at a rate of about 8 Hz for the duration of the actuation. Full-surface operation must have been detected before the first telegram has been transmitted by the Venetian blind function (short time or long time). If this is not so (e.g. one of the two buttons is pressed too late), the full-surface operation will not be correctly executed.

A full-surface operation is independent. It has a communication object of its own an can optionally be used for switching (ON, OFF, TOGGLE – toggling of the object value) or for scene recall without or with storage function. In the last case, the full-surface actuation on causes a scene to be recalled in less than a second. If the device is to send the telegram for storing a scene, full-surface actuation must be maintained for more than five seconds. If full-surface actuation ends between the first and the fifth second, the device will not send any telegrams.

### 8.1.3.1 Table of parameters

The following parameters are available for the individual buttons, depending on the set operating concept. The default settings change in accordance with the set operating concept.

| | |
|---|---|
| Command on pressing | UP<br>DOWN<br>**TOGGLE** |
| This parameter defines the running direction of the drive on pressing the button. If the setting is "TOGGLE", the direction is changed after each long time command. If several devices are to control the same drive, the long time objects of the devices must be interlinked to ensure that the running direction can be changed correctly. | |

| | |
|---|---|
| Operation concept | **short – long – short**<br>long – short<br>short – long<br>long – short or short |
| For Venetian blind control, four different operation concepts can be selected. For these concepts, the ETS shows further parameters. | |

| | |
|---|---|
| Time between short-time and long-time command | **0** ... 4 min \| **0** ... 59 s \| 100 ... **400** ... 990 ms |
| This parameter sets the time after which the long-time operation will be evaluated on pressing the button.<br>This parameter is not visible with "Operation concept = long – short"! | |

| | |
|---|---|
| Slat adjusting time | **0** ... 4 min \| **0** ... 59 s \| 100 ... **500** ... 990 ms |
| The time during which a transmitted MOVE telegram can be terminated by releasing the button (STEP) is set here. This function serves to adjust the slats of a blind.<br>This parameter is not visible with "Operation concept = short - long"! | |

| | |
|---|---|
| Full-surface operation | Active<br>**Inactive** |
| When the full-surface operation is activated, the ETS shows the following parameters.<br>Full-surface operation can only be programmed if "Operation concept = long – short or short"! | |

| Function for full-surface operation | **Switching**<br>Scene recall without storage function<br>Scene recall with storage function |
|---|---|
| In case of full-surface operation, this parameter defines the function that is to be used. The ETS shows the corresponding communication object and the other parameters.<br>This parameter is only visible if "Full-surface actuation = Active"! | |

| Command for full-surface operation | ON<br>OFF<br>**TOGGLE** |
|---|---|
| This parameter defines the value of the transmitted telegram when a full-surface operation has been sensed. "TOGGLE" changes over the current object value.<br>This parameter is only visible if "function for full-surface operation = Switching"! | |

| Scene number (1 … 64) | **1**, 2, ..., 64 |
|---|---|
| This parameter defines the scene number, which is to be transmitted to the bus after a scene recall or during storage of a scene.<br>This parameter is only visible if "function for full-surface operation = scene recall ..."! | |

### 8.1.3.2 Object list

The following communication objects are available for the individual buttons or rockers, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 113, 116, ..., 134 | Short time operation | Button/rocker *n* - output | 1-bit | 1,007 | C, -, -, T, - |
| 1-bit object for the transmission of telegrams with which a Venetian blind or shutter drive motor can be stopped or with which the blind slats can be adjusted by short time operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 114, 117, ..., 135 | Long-time operation | Button/rocker *n* - output | 1-bit | 1,008 | C, -,W, T, - |
| 1-bit object for the transmission of telegrams with which a Venetian blind or shutter drive motor can be can be moved upwards or downwards. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 67, 70 | Switching (full-surface operation) | Button/rocker *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for the transmission of switching telegrams (ON, OFF) when there is full-surface operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 68, 71 | Switching feedback (full-surface operation) | Button/rocker *n* - input | 1-bit | 1,001 | C, -,W, -, U |
| 1-bit object for receiving feedback telegrams (ON, OFF) for full-surface operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 244, 247 | Scene extension (full-surface operation) | Button/rocker *n* - output | 1-bit | 1,001 | C, -, -, T, - |
| 1-byte object for recalling or for storing one of 64 scenes max. from a scene push-button sensor in case of full-surface operation. | | | | | |

### 8.1.4 Value transmitter

For each rocker or each button with the function set to "value transmitter", the ETS indicates up to 5 objects. On pressing a button, the configured value is transmitted to the bus. In case of a rocker function, different values can be configured for both buttons.

#### Value ranges

The "Function" parameter determines the value range used by the value transmitter.

| Function | Function | Lower numerical limit | Upper numerical limit |
|---|---|---|---|
| 1-byte value transmitter | 0...255 | 0 | 255 |
| 1-byte value transmitter | 0...100% | 0% | 100% |
| 1-byte value transmitter | -128...127 | -128 | 127 |
| 1-byte value transmitter | 0...255% | 0% | 255% |
| 1-byte value transmitter | 0...360° | 0° | 360° |
| 2-byte value transmitter | 0...65535 | 0 | 65535 |
| 2-byte value transmitter | -32768...32767 | -32768 | 32767 |
| 2-byte value transmitter | Temperature value | 0 °C | 40 °C |
| 2-byte value transmitter | Brightness value | 0 lux | 1500 lux |
| 2-byte value transmitter | Colour temperature value | 1000 K | 10000 K |
| 3-byte value transmitter | RGB/HSV with colour wheel sequence | #000000 | #FFFFFF |
| 3-byte value transmitter | RGB/HSV with brightness adjustment | #000000 | #FFFFFF |
| 6-byte value transmitter | RGBW/HSVW | #000000 + 0 | #FFFFFF + 255 |

For each of these ranges, the value that can be transmitted to the bus for each button actuation is configurable.

### Value adjustment by long button-press

If the value adjustment feature is activated in the ETS, the button for adjusting the value must be kept depressed longer than the configured time period after pressing the button until the start of the adjustment in order to vary the current value of the value transmitter. The value adjustment function continues to be active until the button is released again.

- With the 1 byte and 2 byte value transmitter functions the value is adjusted across the entire number range.
- With the 3 byte value transmitter function in the function RGB/HSV with colour wheel adjustment, the colour hue (H) is adjusted in the range from 0 to 360°.
- With the 3 byte value transmitter function in the RGB / HSV function with brightness adjustment, the brightness value (V) is adjusted in the range from 0 to 100%.

Upon activating the "value adjustment by long button-press" parameters, further parameters are displayed in the ETS for configuring the value adjustment.

| Example 1: Value adjustment without overflow |
|---|
| - Function = value transmitter 1 byte (0...255) |
| - Value (0...255) = 227 |
| - Step width = 5 |
| - Starting value on value adjustment = same as configured value |
| - Direction of value adjustment = toggling (alternating) |
| - Time between two telegrams = 0.5 s |

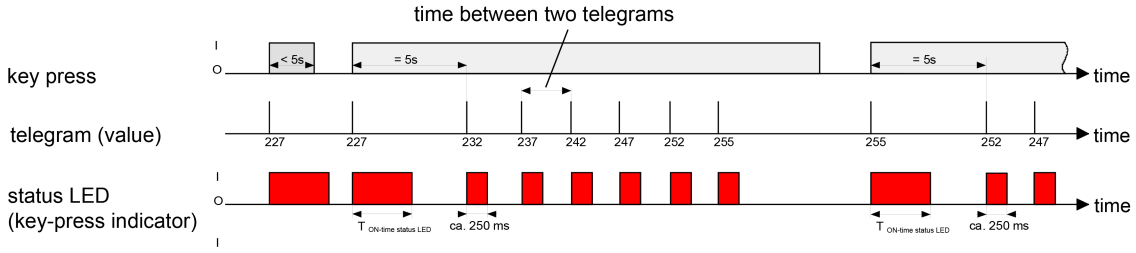


Image 13: Example of value adjustment without value range overflow

| Example 2: Value adjustment with overflow |
|---|
| - Function = value transmitter 1 byte (0...255) |
| - Value (0...255) = 227 |
| - Step width = 5 |
| - Starting value on value adjustment = same as configured value |
| - Direction of value adjustment = toggling (alternating) |
| - Time between two telegrams = 0.5 s |

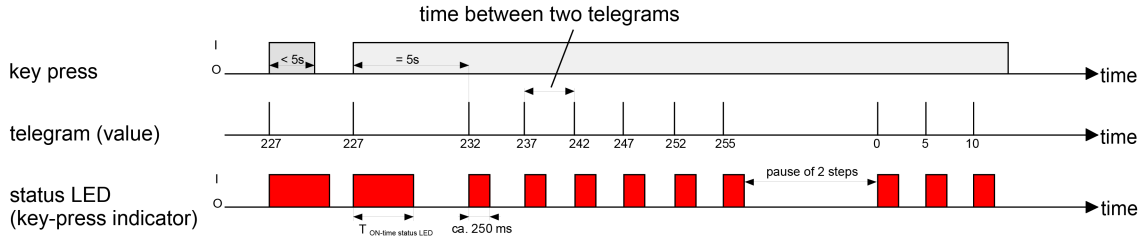


Image 14: Example of value adjustment with value range overflow

The value adjustment by long button-press is not available with the function "6 byte RGBW/HSVW".

During a value adjustment, the newly adjusted values are only in the volatile RAM memory of the extension module. The stored values are thereby replaced by the preset values programmed in the ETS when a reset of the device occurs (bus voltage failure or ETS programming).

During a value adjustment, a status LED parameterised for the "actuation display" function flashes for each newly sent value if this key is assigned to the status LED for value adjustment.

With the 1-byte value transmitter in the "Value transmitter 0…100 %" function, the step width of the adjustment will also be indicated in "%". If the starting value of the communication object is used, it may happen in this case during value adjustment that the value last received via the object must be rounded and adapted before a new value can be calculated on the basis of the step width and transmitted. Due to the computation procedure used, the new calculation of the value may be slightly inaccurate.

### 8.1.4.1 Table of parameters

The following parameters are available for the individual buttons, depending on the set operating concept. The default settings change in accordance with the set operating concept.

| Function | **1 byte (0...255)**<br>1 byte (0...100%)<br>1 byte (-128...127)<br>1 byte (0...255%)<br>1 byte (0...360°)<br>2 byte (0...65535)<br>2 byte (-32768...32767)<br>2 byte temperature value<br>2 byte brightness value<br>2 bytes colour temperature value<br>3 bytes RGB/HSV with colour wheel sequence<br>3 bytes RGB/HSV with brightness adjustment<br>6 bytes RGBW/HSVW |
|---|---|
| The "value transmitter" function distinguishes between 1 byte, 2 byte 3 byte and 6 byte values.<br>The following parameters and their settings depend on the setting for this parameter. | |

| Value (0…255) | **0**...255 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 1 byte (0...255)". | |

| Value (0…100 %) | **0**...100 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 1 byte (0...100%)". | |

| Value (-128...127) | -128...**0**...127 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 1 byte (-128...127)". | |

| Value (0…255%) | **0**...255 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 1 byte (0...255%)". | |

| Value (0…360°) | **0**...360 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 1 byte (0...360°)". | |

| Value (0…65535) | **0**...65535 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 2 byte (0...65535)". | |

| Value (-32768…32767) | -32768...**0**...32767 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 2 byte (-32768...32767)". | |

| Temperature value (0...40 °C) | 0...**20**...40 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 2-byte temperature value". | |

| Brightness value (0, 50, ..., 1500 lux) | 0, 50, ..., **300**, ..., 1500 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 2-byte brightness value". | |

| Colour temperature (1000, 1100, ..., 10000 K) | 1000, 1100, ..., **2700**, ..., 10000 |
|---|---|
| This parameter defines the object value when the button is pressed.<br>Visible only if "Function = 2 byte colour temperature value". | |

| Value (RGB/HSV) | **#000000** ... #FFFFFF |
|---|---|
| This parameter determines the object values of the value transmitter 3 byte (or value transmitter 6 byte), brightness value (V), saturation (S) and colour hue (H) objects when the button is pressed.<br>It is visible for "Function = 3 byte RGB/HSV with colour wheel sequence", "Function = 3 byte RGB/HSV with brightness adjustment" and "6 byte RGBW/HSVW".<br>The value (RGB/HSV) is configured by means of a colour picker.<br>With the "6 byte RGBW / HSVW" function, the W value is configured using a separate slider. | |

| Value (W) | **0** ... 255 |
|---|---|
| This parameter defines the object value of the white level (W) object when the button is pressed.<br>Visible only if "Function = 6 byte RGBW/HSVW". | |

| Value adjustment by long button-press | Active<br>**Inactive** |
|---|---|
| If the value adjustment is activated by a long button-press, the ETS shows further parameters.<br>If a status LED is configured for the "actuation display" function and is assigned to the button for value adjustment, then this flashes during a value adjustment. The status LED symbolises that a new telegram has been transmitted.<br>i The value adjustment by long button-press is not available with the function "6 byte RGBW/HSVW". | |

| Starting value in case of value adjustment | Same as configured value<br>Same as value after last adjustment<br>**Same as value from communication object** |
|---|---|
| Value adjustment can begin with different starting values.<br>With "Same as parameterised value": After each long press, the device always starts with the value configured in the ETS.<br>With "Same as value after last adjustment": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.<br>With "Same as value from communication object": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.<br>This parameter is only visible if "Value adjustment by long button-press = Active"!<br>i The start value of the value adjustment is different for both buttons of a rocker if the setting is "same as value after last adjustment". If the value adjustment works for both buttons of a rocker and the last rocker adjustment is to be taken into account, the setting "same as value from communication object" must be configured.<br>i This selection is only available for the following functions: 1-byte (...) and 2-byte (...) | |

| | |
|---|---|
| Starting value in case of value adjustment | Same as configured value<br>Same as value after last adjustment<br>**same as value from feedback object (1-byte colour hue/H-value)**<br>as value from feedback object (3-byte RGB) |

Value adjustment can begin with different starting values.

With "Same as parameterised value": After each long press, the device always starts with the value configured in the ETS.

With "Same as value after last adjustment": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.

With "same as value on feedback object (1-byte colour hue/H-value)": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.

With "same as value from feedback object (3-byte RGB)": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.

This parameter is only visible if "Value adjustment by long button-press = Active"!

i The start value of the value adjustment is different for both buttons of a rocker if the setting is "same as value after last adjustment". If the value adjustment works for both push-buttons of a rocker and the last rocker adjustment is to be taken into account, the setting "same as value from feedback object ..." must be configured .

i This selection is only available for the following functions: 3 bytes RGB/HSV with colour wheel sequence.

| Starting value in case of value adjustment | Same as configured value<br>Same as value after last adjustment<br>**same as value on feedback object (1-byte brightness/V-value)**<br>as value from feedback object (3-byte RGB) |
|---|---|

Value adjustment can begin with different starting values.

With "Same as parameterised value": After each long press, the device always starts with the value configured in the ETS.

With "Same as value after last adjustment": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.

With "same as value on feedback object (1-byte brightness/V-value)": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.

With "same as value from feedback object (3-byte RGB)": After a long press, the device starts with the value transmitted by itself or by another device with this group address as the last value.

This parameter is only visible if "Value adjustment by long button-press = Active"!

i The start value of the value adjustment is different for both buttons of a rocker if the setting is "same as value after last adjustment". If the value adjustment works for both push-buttons of a rocker and the last rocker adjustment is to be taken into account, the setting "same as value from feedback object ..." must be configured .

i This selection is only available for the following functions: 3 byte RGB/HSV with brightness adjustment.

| Direction of value adjustment | upwards<br>downwards<br>**toggling (alternating)** |
|---|---|

With a long press, the device can either vary the values always in the same direction or it stores the direction of the last adjustment and reverses it on the next button-press.

This parameter is only visible if "Value adjustment by long button-press = Active"!

i This selection is only available for the following functions: 1-byte (...) and 2-byte (...)

| Direction of the colour sequence | Colour sequence in clockwise direction (red -> green -> blue -> red -> ...)<br>Colour sequence in anti-clockwise direction (red -> blue -> green -> red -> ...)<br>**Toggling colour sequence (alternating whenever a button is pressed for a longer period of time)** |
|---|---|
| With a long press, the device can either vary the values always in the same direction or it stores the direction of the last adjustment and reverses it on the next button-press.<br>This parameter is only visible if "Value adjustment by long button-press = Active"!<br>i This selection is only available for the following functions: 3 bytes RGB/HSV with colour wheel sequence. | |

| Direction of the brightness adjustment | brighter<br>**darker**<br>toggling (alternating) |
|---|---|
| With a long press, the device can either vary the values always in the same direction or it stores the direction of the last adjustment and reverses it on the next button-press.<br>This parameter is only visible if "Value adjustment by long button-press = Active"!<br>i This selection is only available for the following functions: 3 byte RGB/HSV with brightness adjustment. | |

| Step width | 1...**15** |
|---|---|
| In a value adjustment, the device determines the new telegram value from the previous value and the preset step width. If the value falls below the lower limit of the adjustment range or if it exceeds the upper limit, the sensor adapts the step width of the last step automatically.<br>This parameter is only visible if "Value adjustment by long button-press = Active"!<br>This value range is only available for the following functions: 1 byte (0 ... 255), 1 byte (0 ... 100%), 1 byte (-128 ... 127), 1 byte (0 ... 255%) and 1 byte (0 ... 360 °). | |

| Step width | **1**, 2, 5, 10, 20, 50, 75, 100, 200, 500, 750, 1000 |
|---|---|
| In a value adjustment, the device determines the new telegram value from the previous value and the preset step width. If the value falls below the lower limit of the adjustment range or if it exceeds the upper limit, the sensor adapts the step width of the last step automatically.<br>This parameter is only visible if "Value adjustment by long button-press = Active"!<br>This value range is only available for the following functions: 2-byte (0 ... 65535) and 2- byte (-32768 ... 32767). | |

| Step width | **1** |
|---|---|
| This parameter is only visible if "Value adjustment by long button-press = Active"!<br>This value range is only available for the following functions: 2-byte temperature value. | |

| Step width | 1, 10, 20, ..., **500**, ..., 1000 |
|---|---|
| This parameter is only visible if "Value adjustment by long button-press = Active"!<br>This value range is only available for the following functions: 2-byte colour temperature value. | |

| Step width | 1**50** |
|---|---|
| This parameter is only visible if "Value adjustment by long button-press = Active"!<br>This value range is only available for the following functions: 2-byte brightness value. | |

| Step width | **1**, 2, 4, 5, 10, 20, 25, 30, 50, 60 |
|---|---|
| In a value adjustment, the device determines the new telegram value from the previous value and the preset step width. If the value falls below the lower limit of the adjustment range or if it exceeds the upper limit, the sensor adapts the step width of the last step automatically.<br>This parameter is only visible if "Value adjustment by long button-press = Active"!<br>This value range is only available for the following functions: 3-byte RGB/HSV with colour wheel sequence and 3-byte RGB/HSV with brightness adjustment. | |

| Time period after pressing button until the start of value adjustment | 0.5 s<br>1 s<br>2 s<br>3 s<br>**5 s** |
|---|---|
| This parameter determines the time from when the device starts the value adjustment after a key is pressed.<br>This parameter is only visible if "Value adjustment by long button-press = Active"! | |

| Time between two telegrams | **0.5 s**<br>1 s<br>2 s<br>3 s |
|---|---|
| This parameter defines the interval at which the device transmits new telegrams during a value adjustment.<br>This parameter is only visible if "Value adjustment by long button-press = Active"! | |

| Value adjustment with overflow | Active<br>**Inactive** |
|---|---|
| If value adjustment is to be effected without overflow (setting "inactive") and if the device reaches the lower limit of the adjustment range or the upper limit during value adjustment, the adjustment will be stopped automatically by the sensor.<br>If the value adjustment with overflow is programmed (setting "active") and if the device reaches the lower or the upper limit, it will transmit the value of this range limit and then add a pause the duration of which corresponds to two levels. Thereafter, the device transmits a telegram with the value of the other range limit and continues the value adjustment in the same direction. | |

### 8.1.4.2 Object list

The following communication objects are available for the individual buttons or rockers, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter 0...255 | Button/rocker *n* - output | 1 bytes | 5,010 | C, -,W, T, - |
| 1-byte object for the transmission of values from 0 to 255.<br>i These objects are only visible if function: 1-byte (0...255). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter 0…100 % | Button/rocker *n* - output | 1 bytes | 5,001 | C, -,W, T, - |
| 1-byte object for transmitting values from 0 to 100%.<br>i These objects are only visible if function: 1-byte (0...100%). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter -128...127 | Button/rocker *n* - output | 1 bytes | 6,010 | C, -,W, T, - |
| 1-byte object for the transmission of values from -128 to 127.<br>i These objects are only visible if function: 1-byte (-128...127). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter 0…255 % | Button/rocker *n* - output | 1 bytes | 5,004 | C, -,W, T, - |
| 1-byte object for transmitting values from 0 to 255%.<br>i These objects are only visible if function: 1-byte (0...255%). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter 0…360° | Button/rocker *n* - output | 1 bytes | 5,003 | C, -,W, T, - |
| 1-byte object for transmitting values from 0 to 360°.<br>i These objects are only visible if function: 1-byte 0...360°). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter 0...65535 | Button/rocker *n* - output | 2 bytes | 7,001 | C, -,W, T, - |
| 2-byte object for the transmission of values from 0 to 65535.<br>i These objects are only visible if function: 2-byte (0...65535). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter -32768...32767 | Button/rocker *n* - output | 2 bytes | 8,001 | C, -,W, T, - |
| 2-byte object for the transmission of values from -32768 to 32767.<br>i These objects are only visible if function: 2-byte (-32768...32767). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Temperature value transmitter | Button/rocker *n* - output | 2 bytes | 9,001 | C, -,W, T, - |
| 2-byte object for transmitting temperature values from 0 to 40 °C.<br>i These objects are only visible if function: 2-byte temperature value. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Brightness value transmitter | Button/rocker *n* - output | 2 bytes | 9,004 | C, -,W, T, - |
| 2-byte object for transmitting brightness values from 0 to 1500 Lux.<br>i These objects are only visible if function: 2-byte brightness value. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Colour temperature value transmitter | Button/rocker *n* - output | 2 bytes | 7,600 | C, -,W, T, - |
| 2-byte object for transmitting colour temperatures from 1000 to 10000 Kelvin.<br>i These objects are only visible if function: 2-byte colour temperature value. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter 3 bytes (colour wheel sequence) | Button/rocker *n* - output | 3 bytes | 232,600 | C, -, -, T, - |
| 3-byte object for transmitting 3-byte colour information.<br>i These objects are only visible if function: 3-byte RGB/HSV with colour wheel sequence. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | Value transmitter 3-byte (brightness adjustment) | Button/rocker *n* - output | 3 bytes | 232,600 | C, -, -, T, - |
| 3-byte object for transmitting 3-byte colour information.<br>i These objects are only visible if function: 3-byte RGB/HSV with brightness adjustment. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 143, 150, ..., 192 | 6-byte value transmitter | Button/rocker *n* - output | 6 bytes | 251,600 | C, -, -, T, - |
| 6-byte object for transmitting 6-byte colour information.<br>i These objects are only visible if function: 6-byte RGBW/HSVW. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 144, 151, ..., 193 | Colour hue (H) | Button/rocker *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting the colour hue.<br>i These objects are only visible if functions:<br>- 3-byte RGB/HSV with colour wheel sequence<br>- 3-byte RGB/HSV with brightness adjustment<br>- 6-byte RGBW/HSVW | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 145, 152, ..., 194 | Saturation (S) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the saturation.<br>i These objects are only visible if functions:<br>- 3-byte RGB/HSV with colour wheel sequence<br>- 3-byte RGB/HSV with brightness adjustment<br>- 6-byte RGBW/HSVW | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 146, 153, ..., 195 | Brightness (V) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the brightness value.<br>i These objects are only visible if functions:<br>- 3-byte RGB/HSV with colour wheel sequence<br>- 3-byte RGB/HSV with brightness adjustment<br>- 6-byte RGBW/HSVW | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 147, 154, ..., 196 | White level (W) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the white level.<br>i These objects are only visible if function: 6-byte RGBW/HSVW. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 149, 156, ..., 198 | Feedback brightness value (V) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -,W, -, U |

1-byte object for receiving the brightness value.

i These objects are only visible with the following configuration:

- "Function" parameter = 3-byte RGB/HSV with brightness adjustment
- "Starting value of value adjustment" parameter = same as value from feedback object (1-byte brightness/V-value)

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 149, 156, ..., 198 | Feedback colour hue (H) | Button/rocker *n* - output | 1 bytes | 5,003 | C, -,W, -, U |

1-byte object for receiving the colour hue.

i These objects are only visible with the following configuration:

- "Function" parameter = 3-byte RGB/HSV with colour wheel sequence
- "Starting value of value adjustment" parameter = same as value from feedback object (1-byte/colour hue/H-value)

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 149, 156, ..., 198 | Value transmitter feedback RGB | Button/rocker *n* - output | 3 bytes | 232,600 | C, -,W, -, U |

3-byte object for receiving 3-byte colour information.

i These objects are only visible with the following configuration:

- The "Function" parameter" = 3-byte RGB/HSV with colour wheel sequence or 3 byte RGB/HSV with brightness adjustment and
- "Starting value of value adjustment" parameter = same as value from feedback object (3-byte RGB)

### 8.1.5 Scene extension

For each rocker or button with the function set to "scene extension unit", the ETS indicates the "Function" parameter which distinguishes between the following settings:

- "Scene extension without storage function",
- "Scene extension with storage function",

In the scene extension function, the device transmits a preset scene number (1…64) via the "scene extension" communication object to the bus after a button-press. This feature permits recalling scenes stored in other devices and also storing them, if the storage function is used.

Function for the setting "… Scene extension without storage function":

- A button actuation results in a simple recall of the scene.
- A long button-press has no further or additional effect.

Function for the setting "… Scene extension with storage function":

- A button actuation of less than one second results in a simple recall of the scene.
- A button-press of more than five seconds, generates a storage instruction. In the scene extension function, a storage telegram is in this case transmitted to the bus. The internal scene is stored. The internal scene control module will then request the current scene values for the actuator groups used from the bus.

i A button actuation lasting between one and five seconds will be discarded as invalid.

### 8.1.5.1 Table of parameters

The following parameters are available for the individual buttons, depending on the set operating concept. The default settings change in accordance with the set operating concept.

| Function | **Scene extension without storage function**<br>Scene extension with storage function |
|---|---|
| This parameter defines the functionality of the scene extension.<br>If the device is used as a scene extension, the scenes can either be stored in one or several other KNX devices (e.g. light scene push button sensor). During a scene recall or in a storage function, the device transmits a telegram with the respective scene number via the extension object of the button. | |

| Scene number (1 … 64) | **1**...64 |
|---|---|
| In accordance with the KNX standard, objects with data type 18.001 "Scene Control" can retrieve or store up to 64 scenes by their numbers. The parameter defines the scene number to be transmitted when the button is pressed. | |

### 8.1.5.2 Object list

The following communication objects are available for the individual buttons or rockers, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 214, 217, ..., 235 | Scene extension | Button/rocker *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for recalling or for storing one of 64 scenes max. from a scene push button sensor. | | | | | |

### 8.1.6 2-channel operation

The "2-channel operation" function allows two function channels to be operated with a single press of a button. In some situations it is desirable to control two different functions with a single press of a button and to transmit different telegrams.

For both channels, the parameters "Function channel 1" and "Function channel 2" can be used to determine the communication object types to be used.

The following functions are available:

- 1-bit switching
- Value transmitter 1 byte (0...255)
- Value transmitter 1 byte (0...100%)
- Value transmitter 1 byte (-128...127)
- Value transmitter 1 byte (0...255%)
- Value transmitter 1 byte (0 ... 360 °)
- Value transmitter 2 byte (0...65535)
- Value transmitter 2 byte (-32768...32767)
- Value transmitter 2-byte temperature value
- Value transmitter 2-byte brightness value
- Recalling scene (external)
- Value transmitter RGB
- Value transmitter RGBW

The object value that the device is to transmit on a button actuation can be selected depending on the selected function.

The "1 bit switching" type permits selecting whether an ON or an OFF telegram is to be transmitted or whether the object value is to be switched over (TOGGLE) and transmitted on the press of a button.

With parameterization as a value transmitter ("1 byte ..." or "2 byte ...") the object value can be selected within the value range.

"Recalling scene (...)" can be used to set the scene number to be transmitted to the bus when a button is pressed.

The status LEDs can be configured independently.

i Unlike in the other rocker and button functions, the application program assigns the "Telegram acknowledge" function instead of the "Button-press display" function to the status LED. In this mode, the status LED lights up for approx. 250 ms with each telegram transmitted.

**Operation concept channel 1 or channel 2**

In this operation concept, exactly one telegram will be transmitted on each press of a button.

- A brief press causes the device to transmit the telegram channel 1.

– A long press causes the device to transmit the telegram for channel 2.

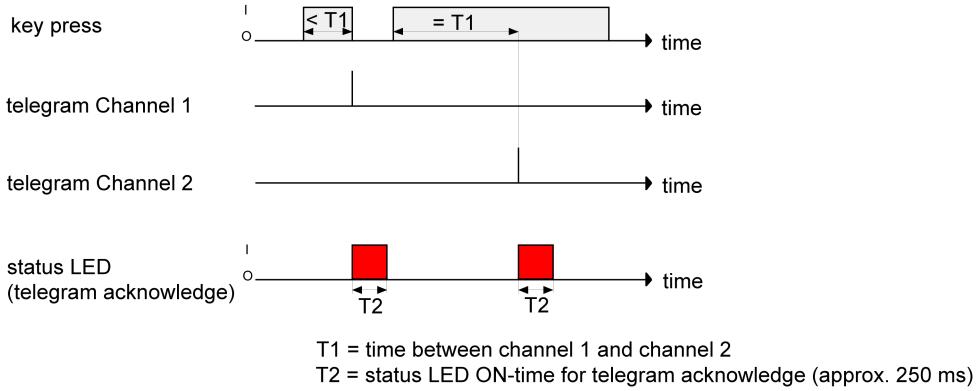


Image 15: Example of operation concept "Channel 1 or Channel 2"

The time required for distinguishing between a short and a long operation is defined by the parameter "Time between channel 1 and channel 2". If the button is pressed for less than the configured time, only the telegram to channel 1 is transmitted. If the length of the button-press exceeds the time between channel 1 and channel 2, only the telegram to channel 2 will be transmitted. This concept provides the transmission of only one channel. To indicate that a telegram has been transmitted, the status LED lights up for approx. 250 ms in the "Telegram acknowledge" setting.
In this operation concept, the push-button sensor will not transmit a telegram immediately after the rocker has been depressed.

**Operation concept channel 1 and channel 2**

With this operation concept, one or alternatively two telegrams can be transmitted on each button-press.

– A brief press causes the device to transmit the telegram channel 1.
– A long press causes the device to transmit first the telegram for channel 1 and then the telegram for channel 2.

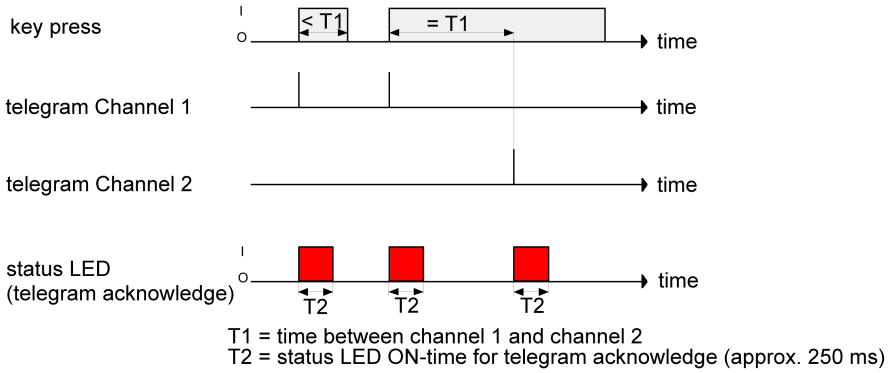


Image 16: Example for operation concept "Channel 1 and channel 2"

The time required for distinguishing between a short and a long operation is defined by the parameter "Time between channel 1 and channel 2". In this operation concept, a button-press sends this telegram is immediately to channel 1. If the button is held depressed for the configured time, the telegram for the second channel is transmitted as well. If the button is released before the time has elapsed, no further telegram will

be transmitted. This operation concept, too, offers the configurable possibility of having the transmission of a telegram signalled by the status LED (setting "Telegram acknowledge").

### Full-surface operation with 2-channel operation

When a rocker is programmed for 2-channel operation and if the operation concept "channel 1 or channel 2" is used, the device needs some time at the beginning of each operation in order to distinguish between a short and a long operation. When full-surface operation is enabled, the device can make use of this time span to evaluate the otherwise invalid simultaneous actuation of both buttons of a rocker.

Full-surface operation of a rocker switch is detected by the device when both buttons are pressed at the same time. When the device has detected a valid full-surface actuation, the status LED flashes quickly at a rate of about 8 Hz for the duration of the actuation. The full-surface operation must have been detected before the first telegram has been transmitted by the 2-channel function. If this is not so (e.g. one of the two buttons is pressed too late), the full-surface operation will not be correctly executed.

### 8.1.6.1 Table of parameters

The following parameters are available for the individual buttons, depending on the set operating concept. The default settings change in accordance with the set operating concept.

| Operation concept | **Channel 1 or channel 2**<br>Channel 1 and channel 2 |
|---|---|
| This is where the 2-channel operation concept is defined.<br>If the setting "channel 1 or channel 2" is selected, the device decides dependent on the button-press duration, which of the channels will be used.<br>If the setting "channel 1 and channel 2" is selected, the device transmits only the telegram of channel 1 on a short button-press and both telegrams on a sustained button-press. | |

| Function channel 1 (Function channel 2) | No function<br>1-bit switching<br>1 byte (0...255)<br>1 byte (0...100%)<br>1 byte (-128...127)<br>1 byte (0...255%)<br>1 byte (0...360°)<br>2 byte (0...65535)<br>2 byte (-32768...32767)<br>2 byte temperature value<br>2 byte brightness value<br>Recalling scene (external)<br>Value transmitter RGB<br>Value transmitter RGBW |
|---|---|
| This parameter defines the channel function and specifies which other parameters and which communication object are to be displayed for channel 1 or channel 2 respectively. | |

| Time between channel 1 and channel 2 | 0...**3**...25 s \| 0...990 ms |
|---|---|
| Depending on the selected operation concept, this parameter defines the interval at which the device transmits the telegram for channel 1 and the telegram for channel 2. A time from 100 ms to 25.5 s can be set. | |

| Command for channel 1 (2) | ON<br>OFF<br>**TOGGLE** |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 1 bit switching". | |

| Value (0...255) for channel 1 (2) | **0**...255 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 1-byte (0...255)". | |

| Value (0...100%) for channel 1 (2) | **0**...100 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 1-byte (0...100%)". | |

| Value (-128...127) for channel 1 (2) | -128...**0**...127 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 1-byte (-128...127)". | |

| Value (0...255%) for channel 1 (2) | **0**...255 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 1-byte (0...255%)". | |

| Value (0...360°) for channel 1 (2) | **0**...360 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 1-byte (0...360°)". | |

| Value (0...65535) for channel 1 (2) | **0**...65535 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 2-byte (0...65535)". | |

| Value (-32768...32767) for channel 1 (2) | -32768...**0**...32767 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 2-byte (-32768...32767)". | |

| Temperature value (0…40 °C) for channel 1 (2) | 0...**20**...40 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 2-byte temperature value". | |

| Brightness value (0, 50, …, 1500 lux) for channel 1 (2) | 0...**300**...1500 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>Visible only if "Function channel 1 (2) = 2-byte brightness value". | |

| Scene number (1…64) for channel 1 (2) | **1**...64 |
|---|---|
| This parameter defines the object value transmitted to the bus when the button is pressed.<br>It is only visible if "Function channel 1 (2) = Recalling scene (external)". | |

| Value (RGB/HSV) for channel 1 (2) | **#000000** ... #FFFFFF |
|---|---|
| This parameter determines the object values of the colour hue (H), saturation (S), brightness value (V), which is transmitted to the bus when the button is pressed.<br>It is visible if "Function channel 1 (2) = value transmitter RGB and value transmitter RGBW". | |

| Value (W) for channel 1 (2) | **0** ... 255 |
|---|---|
| This parameter defines the object value of the white level (W) object when the button is pressed.<br>It is only visible if "Function channel 1 (2) = Value transmitter RGBW". | |

| Full-surface operation | Active<br>**Inactive** |
|---|---|
| When the full-surface operation is activated, the ETS shows the following parameters.<br>Full-surface operation can only be programmed if "Operation concept = Channel 1 or channel 2"! | |

| Function for full-surface operation | **Switching**<br>Scene recall without storage function<br>Scene recall with storage function |
|---|---|
| In case of full-surface operation, this parameter defines the function that is to be used. The ETS shows the corresponding communication object and the other parameters.<br>This parameter is only visible if "Full-surface actuation = Active"! | |

| Command for full-surface operation | ON<br>OFF<br>**TOGGLE** |
|---|---|
| This parameter defines the value of the transmitted telegram when a full-surface operation has been sensed. "TOGGLE" changes over the current object value.<br>This parameter is only visible if "function for full-surface operation = Switching"! | |

| Scene number (1 … 64) | **1**, 2, ..., 64 |
|---|---|
| This parameter defines the scene number, which is to be transmitted to the bus after a scene recall or during storage of a scene.<br>This parameter is only visible if "function for full-surface operation = scene recall ..."! | |

### 8.1.6.2 Object list

The following communication objects are available for the individual buttons or rockers, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 switching | Button/rocker *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmitting switching telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 switching | Button/rocker *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmitting switching telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 value 0...255 | Button/rocker *n* - output | 1 bytes | 5,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 value 0...255 | Button/rocker *n* - output | 1 bytes | 5,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 value 0...100 % | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 value 0...100 % | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 value -128...127 | Button/rocker *n* - output | 1 bytes | 6,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 value -128...127 | Button/rocker *n* - output | 1 bytes | 6,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 value 0...255 % | Button/rocker *n* - output | 1 bytes | 5,004 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 value 0...255 % | Button/rocker *n* - output | 1 bytes | 5,004 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 value 0...360° | Button/rocker *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 value 0...360° | Button/rocker *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 value 0...65535 | Button/rocker *n* - output | 2 bytes | 7,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 value 0...65535 | Button/rocker *n* - output | 2 bytes | 7,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 value -32768...32767 | Button/rocker *n* - output | 2 bytes | 8,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 value -32768...32767 | Button/rocker *n* - output | 2 bytes | 8,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 temperature value | Button/rocker *n* - output | 2 bytes | 9,001 | C, -, -, T, - |
| 2-byte object for transmitting temperature values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 temperature value | Button/rocker *n* - output | 2 bytes | 9,001 | C, -, -, T, - |
| 2-byte object for transmitting temperature values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 brightness value | Button/rocker *n* - output | 2 bytes | 9,004 | C, -, -, T, - |
| 2-byte object for transmitting brightness values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 brightness value | Button/rocker *n* - output | 2 bytes | 9,004 | C, -, -, T, - |
| 2-byte object for transmitting brightness values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 scene (external) 1…64 | Button/rocker *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 scene (external) 1…64 | Button/rocker *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 (RGB) | Button/rocker *n* - output | 3 bytes | 232,600 | C, -, -, T, - |
| 3-byte object for transmitting RGB values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 (RGB) | Button/rocker *n* - output | 3 bytes | 232,600 | C, -, -, T, - |
| 3-byte object for transmitting RGB values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 249, 262 ..., 340 | Channel 1 (RGBW) | Button/rocker *n* - output | 6 bytes | 251,600 | C, -, -, T, - |
| 6-byte object for transmitting RGB values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 250, 263, ..., 341 | Channel 2 (RGBW) | Button/rocker *n* - output | 6 bytes | 251,600 | C, -, -, T, - |
| 6-byte object for transmitting RGB values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 251, 264 ..., 342 | Channel 1 colour hue (H) | Button/rocker *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting the colour hue on channel 1, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 255, 268 ..., 346 | Channel 2 colour hue (H) | Button/rocker *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting the colour hue on channel 2, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 252, 265 ..., 343 | Channel 1 saturation (S) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the saturation on channel 1, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 256, 269 ..., 347 | Channel 2 saturation (S) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the saturation on channel 2, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 253, 266 ..., 344 | Channel 1 brightness value (V) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the brightness value on channel 1, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 257, 270 ..., 348 | Channel 2 brightness value (V) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the brightness value on channel 2, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 254, 267 ..., 345 | Channel 1 white level (W) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the white level on channel 1, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 258, 271 ..., 349 | Channel 2 white level (W) | Button/rocker *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the white level on channel 2, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 67, 70 | Switching (full-surface operation) | Button/rocker *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for the transmission of switching telegrams (ON, OFF) when there is full-surface operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 68, 71 | Switching feedback (full-surface operation) | Button/rocker *n* - input | 1-bit | 1,001 | C, -,W, -, U |
| 1-bit object for receiving feedback telegrams (ON, OFF) for full-surface operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 244, 247 | Scene extension (full-surface operation) | Button/rocker *n* - output | 1-bit | 1,001 | C, -, -, T, - |
| 1-byte object for recalling or for storing one of 64 scenes max. from a scene push-button sensor in case of full-surface operation. | | | | | |

### 8.1.7 Controller extension

The "controller extension" button or rocker function can be used to control a KNX room temperature controller.

The controller extension itself is not involved in the regulating process. With it, the user can operate the single-room regulation from different places in the room. It can also be used to adjust central heating control units which are located, for instance, in a sub-distribution unit.

Typical KNX room temperature controllers generally offer different ways of influencing the room temperature control:

- Operating mode switch:
  Switching over between different modes of operation (e.g. "Comfort", "Night" ...) with different setpoint temperatures assigned to each mode by the controller.
- Presence button:
  Signalling the presence of a person in a room. The signalling may also be combined with a configured switchover in the mode of operation.
- Setpoint shift:
  Adjustment of the setpoint temperature via a temperature offset (DPT 9.002) or via levels (DPT 6.010).

The controller extension is operated using the push button functions of the device. In this way, it is possible to completely control a room temperature controller by changing the operating mode, by predefining the presence function or by readjusting the setpoint shift.

In addition, the device can – independent of the controller extension function – indicate the state of one or more room temperature controllers with the status LEDs of the rockers or buttons. This feature permits the indication of operating modes or the bit-oriented evaluation of different status objects of controllers. In case of the controller extension functions "Setpoint shift" or "Presence function", the status LEDs can also signal the state of the corresponding functions directly.

### 8.1.7.1 Operating mode switchover

Switchover of the controller operating mode can be effected in accordance with the standard function block for room temperature controllers defined in the KNX handbook using two 1-byte communication objects. The operating mode can be switched over with the normal and with the forced objects. The "Operating mode switchover" object offers a selection between the following modes:

- Comfort mode
- Standby mode
- Night operation
- Frost/heat protection mode

The "Forced object operating mode" communication object has a higher priority. It permits forced switching between the following modes of operation:

- Auto (normal operating mode switchover)
- Comfort mode
- Standby mode
- Night operation
- Frost/heat protection mode

The operating mode transmitted to the bus on a button press of the controller extension is defined by the parameter "Operating mode on pressing". Depending on the parameterized operating concept, either a button press activates one of the above modes (with the "rocker function" and "button function" operating concepts), or each button press toggles between two or three modes (only with the "rocker function" operating concept).

Notes on multiple selection:
In order to ensure that a change-over from one operating mode to another works properly even from different locations, the operating mode objects of the controller and those of all controller extensions must be interlinked and have their "Write" flag set. In the objects concerned, this flag is set by default
By checking the linked operating mode switchover feedback object, the controller extension knows which of the possible operating modes is active. Based on this information, the device switches over into the next operating mode in sequence when a button is pressed. In the event that none of the possible operating modes is active, the next operating mode in the sequence is activated. As far as switching over between the forced operating modes and "Auto" is concerned, the device switches into the "Auto" operating mode when none of the configured operating modes is active.

If a status LED is to indicate the current operating mode, the status LED function must be programmed for "Operating mode indication" and its status object be linked with the corresponding group address for operating mode change-over with normal or high priority.

### 8.1.7.2 Presence function

All operating areas with a function set to "Presence function" have the two communication objects "Presence function" and "Presence function feedback". The parameter "Presence function on pressing" defines the object value transmitted to the bus on pressing a button.

In order to ensure that the object value transmitted in the "Presence TOGGLE" setting is always the correct one, the presence object of the room temperature controller and the feedback objects of the controller extensions must be interlinked and have their "Write" flag set. In the extension objects concerned, this flag is set by default.

The status LED of a presence function button can directly indicate the presence status (setting "Presence status indicator").

### 8.1.7.3 Setpoint shift

The setpoint shift is another available function of the controller extension. It makes use of either two 2-byte communication objects with datapoint type 9.002 or two 1-byte communication objects with datapoint type 6.010 (integer with sign).

This extension function allows shifting of the basic setpoint for the temperature on a room temperature controller by pressing a button. Operation of the extension is generally the same as the operation of the main controller. A button configured as a setpoint shifting button reduces or increases the setpoint shift value on each press by one step respectively. The direction of the value adjustment is defined by the parameters "Temperature difference on pressing" or "Setpoint shift on pressing".

The status LED of a setpoint shifting button can directly indicate the setpoint shifting status (setting "Setpoint value shift indicator").

#### Type of setpoint shift

The device provides two options for setpoint shifting. Depending on the setting of the parameter "Type of setpoint shift", the shift takes place via the 2-byte communication object "Setpoint shift specification" (acc. to KNX DPT 9.002) or via the 1-byte-communication object "Setpoint shift specification" (acc. to KNX DPT 6.010).

The setting "Via offset (DPT 9.002)" defines the temperature difference in Kelvin by which the setpoint temperature will be shifted up or down when the button is pressed. For a setpoint value shift, the controller extension makes use of the two communication objects "Setpoint shift specification" and "Current setpoint shift". The "Current setpoint shift" communication object informs the extension about the current state of the room temperature controller. Based on this value and the respective parameter, the controller extension determines the new level size which it transmits via the "Setpoint shift specification" communication object to the room temperature controller.

With the "Via levels (DPT 6.010)" setting, only the direction of the setpoint shift on the extension is defined. For a setpoint value shift, the controller extension makes use of the two communication objects "Setpoint shift specification" and "Current setpoint shift". The "Current setpoint shift" communication object informs the extension about the current state of the room temperature controller. Based on this value and the respective parameter, the controller extension determines the new level size which it transmits via the "Setpoint shift specification" communication object to the room temperature controller.

#### Communication with main controller

In order to enable the device to effect a setpoint shift in a room temperature controller, the controller must have input and output objects for setpoint shifting. In this case, the output object of the controller must be linked with the input object of the extension unit and the input object of the controller must be linked with the output object of the extension via an independent group address.

All objects are of the same datapoint type and have the same value range. A setpoint shift is interpreted by count values: a shift in positive direction is expressed by positive values whereas a shift in negative direction is represented by negative object values. An object value of "0" means that no setpoint shift has been activated.

Via the "Current setpoint shift" object of the controller extensions, which is linked with the room temperature controller, the extensions are enabled to determine the current setpoint shift position. Starting from the value of the communication object, each button-press on an extension will adjust the setpoint in the configured direction. Each time the setpoint is adjusted, the new shift is transmitted to the room temperature controller via the "Controller extension setpoint value specification" object of the controller extension. The controller itself checks the received value for the minimum and maximum temperature limits (see controller documentation) and adjusts the new setpoint shift if the values are valid. When the new count value is accepted as valid, the controller transfers this value to its output object for setpoint shifting and retransmits the value to the extension as feedback.

Due to the standard data point type used as the output and input object of the controller extension, each extension unit is able to determine whether a shift took place, in which direction it took place and by which value (DPT 9.002) or by how many levels (DPT 6.010) the setpoint was shifted.

i In "Via levels (DPT 6.010)" function, the weighting of the respective level is done by the controller itself.

i This requires that the communication objects are connected on all controller extensions and the controller. The feedback information from the controller enables the extension to continue the adjustment anytime at the right point.

### 8.1.7.4 Table of parameters

The following parameters are available for the individual buttons, depending on the set operating concept. The default settings change in accordance with the set operating concept.

| Function | **Operating mode switchover**<br>Forced oper. mode switchover<br>Presence function<br>Setpoint shift |
|---|---|
| A controller extension can optionally switch over the operating mode with normal or high priority (forced), change the presence state or change the current room temperature setpoint value. With regard to the setting of this parameter, the ETS shows further parameters. | |

| Operating mode on pressing | **Comfort mode**<br>Standby mode<br>Night operation<br>Frost/heat protection mode<br>Comfort mode -> Standby mode ->*<br>Comfort mode -> Night mode ->*<br>Standby mode -> Night mode ->*<br>Comfort mode -> Standby mode -> Night mode ->*<br>Comfort mode -> Standby mode -> Night mode -> Frost/heat protection mode ->* |
|---|---|
| If the controller extension is to change over the operating mode of the room temperature controller with normal priority, the extension can – when operated – either switch on a defined operating mode or change over between different operating modes.<br>This parameter is only visible if "Function = operating mode switchover".<br>* Only for operation concept = button function<br>The options marked with a * are only available if the operating concept is set to the button function.<br>i Toggling between operating modes (identified by "->") is only possible if the object value has been received via the "Operating mode switchover feedback" object. | |

| | |
|---|---|
| Forced operating mode on pressing | Auto<br>(Normal operating mode change-over)<br>**Comfort mode**<br>Standby mode<br>Night operation<br>Frost/heat protection mode<br>Comfort mode -> Standby mode ->*<br>Comfort mode -> Night mode ->*<br>Standby mode -> Night mode ->*<br>Comfort mode -> Standby mode -> Night mode ->*<br>Comfort mode -> Standby mode -> Night mode -> Frost/heat protection mode ->*<br>Auto -> Comfort mode ->*<br>Auto -> Standby mode ->*<br>Auto -> Night mode ->*<br>Auto -> Frost/heat protection mode ->*<br>Auto -> Comfort mode -> Standby mode -> Night mode ->*<br>Auto -> Comfort mode -> Standby mode -> Night mode -> Frost/heat protection mode ->* |
| If the controller extension is to change over the operating mode of the room temperature controller with high priority, the extension can – when operated – either enable change-over with normal priority (Auto), switch on a defined operating mode with high priority or change over between different operating modes.<br>This parameter is only visible if "Function = forced operating mode switchover".<br>* Only for operation concept = button function<br>The options marked with a * are only available if the operating concept is set to the button function.<br>i Toggling between forced object operating modes (identified by "->") is only possible if the object value has been received via the "Forced object operating mode feedback" object. | |

| | |
|---|---|
| Presence function on pressing | Presence OFF<br>Presence ON<br>**Presence TOGGLE** |
| On pressing a button, the controller extension can switch the presence state of the room temperature controller either on or off in a defined way or change over between both states ("Presence TOGGLE").<br>This parameter is only visible if "Function = presence function". | |

| Type of setpoint shift | **Via offset (DPT 9.002)**<br>Via levels (DPT 6.010) |
|---|---|
| Depending on the setting of the parameter "Type of setpoint shift", the shift takes place via the 2-byte communication object "Setpoint shift specification" (acc. to KNX DPT 9.002) or via the 1-byte-communication object "Setpoint shift specification" (acc. to KNX DPT 6.010).<br>This parameter is only visible if "Function = Setpoint shift". | |

| Temperature difference on pressing | -2 K<br>-1.5 K<br>-1 K<br>-0.5 K<br>0.5 K<br>1 K<br>1.5 K<br>2 K |
|---|---|
| The temperature difference is defined in Kelvin here by which the setpoint temperature will be shifted up or down when the button is pressed.<br>For a setpoint value shift, the controller extension makes use of the two communication objects "Setpoint shift specification" and "Current setpoint shift".<br>The "Current setpoint shift" communication object informs the extension about the current state of the room temperature controller. Based on this value and the respective parameter, the controller extension determines the new level size which it transmits via the "Setpoint shift specification" communication object to the room temperature controller.<br>This parameter is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via offset". | |

| Setpoint shift on pressing | Reduce setpoint value (level size)<br>**Increase setpoint (level size)** |
|---|---|
| This parameter defines the direction of the setpoint shift on the extension.<br>For a setpoint value shift, the controller extension makes use of the two communication objects "Setpoint shift specification" and "Current setpoint shift".<br>The "Current setpoint shift" communication object informs the extension about the current state of the room temperature controller. Based on this value and the respective parameter, the controller extension determines the new level size which it transmits via the "Setpoint shift specification" communication object to the room temperature controller.<br>This parameter is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via levels". | |

### 8.1.7.5 Object list

The following communication objects are available for the individual buttons or rockers, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 380, 387, ..., 429 | Operating mode switchover | Button/rocker n - controller extension - output | 1 bytes | 20,102 | C, -,W, T, - |
| 1-byte object for changing over a room temperature controller between the Comfort, Standby, Night and Frost/heat protection operating modes.<br>This object is only visible if "Function = operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 381, 388 ..., 430 | Operating mode switchover feedback | Button/rocker n - controller extension - input | 1 bytes | 20,102 | C, -,W, -, U |
| 1-byte object for receiving the operating mode of a room temperature controller.<br>This object is only visible if "Function = operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 380, 387, ..., 429 | Operating mode forced-control | Button/rocker n - controller extension - output | 1 bytes | 20,102 | C, -,W, T, - |
| 1-byte object for changing over a room temperature controller under forced control between the Automatic, Comfort, Standby, Night and Frost / heat protection operating modes<br>This object is only visible if "Function = forced operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 381, 388 ..., 430 | Forced object operating mode feedback | Button/rocker n - controller extension - input | 1 bytes | 20,102 | C, -,W, -, U |
| 1-byte object for receiving the operating mode of a room temperature controller.<br>This object is only visible if "Function = forced operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 380, 387, ..., 429 | Presence function | Button/rocker n - controller extension - output | 1-bit | 1,018 | C, -,W, T, - |
| 1-bit object for changing over the presence status of a room temperature controller.<br>This object is only visible if "Function = presence function". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 381, 388 ..., 430 | Presence function feedback | Button/rocker n - controller extension - input | 1-bit | 1,018 | C, -,W, -, U |
| 1-bit object for receiving the presence status of a room temperature controller.<br>This object is only visible if "Function = presence function". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 380, 387, ..., 429 | Preset setpoint shifting | Button/rocker n - controller extension - output | 2 bytes | 9,002 | C, -,W, T, - |
| 2-byte object for presetting a basic setpoint shift in Kelvin. The value "0" means that no shift is active . The values can be specified between -670760 K and 670760 K.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via offset (DPT 9.002)". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 381, 388 ..., 430 | Current setpoint shifting | Button/rocker n - controller extension - input | 2 bytes | 9,002 | C, -,W, -, U |
| 2-byte object for receiving the feedback from the current basic setpoint shift in Kelvin.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via offset (DPT 9.002)". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 380, 387, ..., 429 | Preset setpoint shifting | Button/rocker n - controller extension - output | 1 bytes | 6,010 | C, -,W, T, - |
| 1-byte object for presetting a basic setpoint shift. The value "0" means that no shift is active . The value is depicted in a two's complement in the positive or negative direction.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via levels (DPT 6.010)". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 381, 388 ..., 430 | Current setpoint shifting | Button/rocker n - controller extension - input | 1 bytes | 6,010 | C, -,W, -, U |
| 1-byte object for receiving the feedback from the current basic setpoint shift.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via levels (DPT 6.010)". | | | | | |

## 8.2 Status LED

Each operating area of the basic device has a status LED.

The 1-gang device variant has one status LED (see figure 1) and the 2-gang device variant has 2 status LEDs (see figure 2).

The configurable functions of the status LED adapt to the configured functions of the rockers or buttons.

A status LED can either be assigned to a rocker or two buttons.

The devices KNX push-button universal / KNX push-button extension differ in the configuration of the status LED as follows.

| Device variant | Status LED |
|---|---|
| KNX push-button universal 1-gang | A three-colour status LED (RGB) |
| KNX push-button universal 2-gang | Two three-colour status LEDs (RGB) |
| KNX push-button extension 1-gang | No status LED |
| KNX push-button extension 2-gang | No status LED |

### Independent functions of the status LED

A variety of functions of the status LED can be configured independently of the configured rocker or button function. These functions either define a fixed lighting status of the status LED or have a separate communication object.

The following functions can always be configured for each Status LED:

- always OFF
- always ON
- Control via separate LED object
- Operating mode display
- Controller status indication
- Comparator without sign (1 byte)
- Comparator with sign (1 byte)
- Logic link
- Bit-coded evaluation

### Dependent functions of the status LED

A variety of functions of the status LED can be configured depending on the configured rocker or button function.

The following functions are configurable for each Status LED depending on the configured rocker or button function.

- Button-press display
- Telegram acknowledgment
- Status indication

- inverted status display
- Presence status
- Setpoint value shift display

Besides the functions that can be set separately for each status LED , all status LEDs are also used together for LED alarm signalling.

## 8.2.1 Basic functions

**"always OFF" or "always ON"**

The corresponding status LED is always switched off or always switched on depending on the parameter setting.

**"button-press display"**

This function can be configured for each status LED if the rocker or button is configured to "switching", "dimming", "Venetian blind", "value transmitter", "scene extension" or "controller extension":

- With the rocker function, each actuation of one of the two buttons is displayed.
- With the key function, the parameter "Assignment of the status LED" decides whether the actuation of both keys or a single key is displayed.

A status LED used as button-press display is switched on by the device each time the corresponding rocker or button is pressed. The parameter "Light period of status LED for button-press indicator" on the parameter page "General -> Status LED" determines how long the status LED for all status LEDs remains on together. Even if the device only sends a telegram when you release it, the status LED lights up regardless of whether you press the rocker or button.

i In the "value transmitter" device function with activated value adjustment by a long button-press, a status LED configured for the "Actuation display" function flashes each time a new value is transmitted.

**"telegram acknowledgment"**

This function can be configured for each status LED if the rocker or button is configured to "2-channel operation".

- With the rocker function, each telegram of one of the two buttons is acknowledged.
- With the button function, the parameter "Assignment of the status LED" decides whether the telegrams of both buttons or of a single button are acknowledged.

If a status LED is used for telegram acknowledgement, the status LED lights up when both channels are transmitted for about 250 ms each.

**"Status display" and "inverted status display"**

These functions can be configured for each status LED if the rocker or button is parameterised to "switching" or "dimming":

- With the rocker function, the switching status of the rocker is displayed.
- With the button function, the parameter "Assignment of the status LED" decides which of the two buttons shows the switching status.

In the rocker or button functions "switching" and "dimming", the status LEDs can also be linked internally to the "Switching feedback" object and thus signal the current switching state of the actuator group.
It is possible to indicate or evaluate the inverted object value.

i After a bus reset or after ETS programming, the value of the LED object is always "OFF".

**"control via separate LED object"**

Each status LED indicates the state of a separate LED communication object. Here the LED can be switched on or off statically via the 1-bit object value received, or also activated as flashing. If multiple status LEDs are configured to "flashing" and switched on, they will flash synchronously.
It is possible to indicate or evaluate the inverted object value.

i After a bus reset or after ETS programming, the value of the LED object is always "OFF".

**"Operating mode display"**

In this configuration the status LED has its own 1-byte communication object.
If a status LED is to indicate the operating mode, the communication object of the status LED must be linked with the matching object of a room temperature controller (e. g. Controller status). The desired operating mode that the LED is to indicate can then be selected with the parameter "Status LED ON with". The LED is then lit up when the corresponding operating mode has been activated at the controller.

i After a bus reset or after ETS programming, the value of the LED object is always "0" (automatic).

**"Controller status display"**

The status LED can indicate the controller status in the data formats "KNX-compliant" or "Controller general". The KNX-compliant objects or general controller objects are offered depending on the configuration. The objects should be connected to the communication objects of the main controller with the same functions via group addresses.
The status objects combine different informations. The "Status LED on with" parameter is used to select what information should be evaluated and displayed via the status LED.

The following information is available for selection with "KNX-compliant":

- Controller error status ("0" = no error / "1" = error)
- Operating mode ("0" = Cooling / "1" = Heating)
- Controller disabled ("0" = Controller enabled / "1" = Controller disabled)
- Frost alarm ("0" = Frost protection temperature exceeded / "1" = Frost protection temperature undershot)
- Heat alarm ("0" = Heat protection temperature exceeded / "1" = Heat protection temperature undershot)

- Controller inactive (Is active in the "Heating and cooling" operating mode when the measured room temperature lies within the deadband. This status information is as a rule always "0" for the individual operating modes "heating" or "cooling"! Is inactive if controller is disabled.)
- Comfort mode extension ("0" = extension inactive/ "1" = extension active)
- Window open ("0" = Window closed / "1" = Window open)
- Additional level active ("0" = Additional level inactive / "1" = Additional level active)

The following table shows the evaluation of the three KNX-compliant objects.

| Status LED ON with | Object RHCC - DPT22.101 | Object RTSM - DPT21.107 | Object RTC - DPT22.103 |
|---|---|---|---|
| Controller error status | ✓ (bit 0) | ☒ | ✓ (bit 0) |
| Operating mode | ✓ (bit 8) | ☒ | ✓ (bit 1) |
| Controller disabled | ✓ (bit 12) | ☒ | ✓ (bit 2) |
| Frost alarm | ✓ (bit 13) | ☒ | ✓ (bit 3) |
| Heat alarm | ✓ (bit 14) | ☒ | ✓ (bit 4) |
| Controller inactive | ☒ | ☒ | ✓ (bit 5) |
| Additional level active | ☒ | ☒ | ✓ (bit 6) |
| Open window | ☒ | ✓ (bit 0) | ☒ |
| Comfort mode extension | ☒ | ✓ (bit 3) | ☒ |

The following information is available for selection with "Controller general":

- Comfort mode ("0" = Comfort mode inactive / "1" = Comfort mode active)
- Standby mode ("0" = Standby mode inactive / "1" = Standby mode active)
- Night mode ("0" = Night mode inactive / "1" = Night mode active)
- Frost/heat protection mode ("0" = Frost/heat protection mode inactive / "1" = Frost/heat protection mode active)
- Controller disabled ("0" = Controller enabled / "1" = Controller disabled)
- Heating / cooling ("0" = Cooling / "1" = Heating)
- Controller inactive ("0" = Controller active / "1" = Controller inactive (dead band))
- Frost alarm ("0" = no frost alarm / "1" = frost alarm)
- Normal/forced operation ("0" = Forced operation/ "1" = Normal operation)
- Comfort mode extension ("0" = no comfort extension/"1" = comfort extension)
- Window open ("0" = Window closed / "1" = Window open)
- Additional level active ("0" = Additional level inactive / "1" = Additional level active)

– Dew point alarm ("0" = Controller not disabled / "1" = Controller disabled (dew point operation))

The following table shows the evaluation of the two general objects.

| Status LED ON with | Object "controller status" | Object "controller status addition" |
|---|---|---|
| Comfort mode | ✓ (bit 0) | ☒ |
| Standby mode | ✓ (bit 1) | ☒ |
| Night operation | ✓ (bit 2) | ☒ |
| Frost/heat protection mode | ✓ (bit 3) | ☒ |
| Controller disabled | ✓ (bit 4) | ☒ |
| Heating / cooling | ✓ (bit 5) | ☒ |
| Controller inactive | ✓ (bit 6) | ☒ |
| Frost alarm | ✓ (bit 7) | ☒ |
| Normal/Forced operating mode | ☒ | ✓ (bit 0) |
| Comfort mode extension | ☒ | ✓ (bit 1) |
| Open window | ☒ | ✓ (bit 4) |
| Additional level active | ☒ | ✓ (bit 5) |
| dew point alarm | ☒ | ✓ (bit 7) |

After a bus reset or after ETS programming, the value of the LED object is always "0".

**"Presence status display" and "Inverted presence status display"**

These functions can be configured for each status LED if the rocker or button is parameterised to "controller extension" with the "presence" function:

– With the rocker function, the presence status of the rocker is displayed.

– With the button function, the parameter "Assignment of the status LED" decides which of the two buttons shows the presence status:

When the presence status is displayed the LED evaluates the value of the object "Feedback presence function" and switches either on or off, depending on the parameter configuration in the ETS.

**"Setpoint value shift display"**

This function can be configured for each status LED if the rocker or button is parameterised to "controller extension" with the "setpoint shift" function.

With the rocker function, the setpoint shift of the rocker is displayed.

With the button function, the parameter "Assignment of the status LED" decides which of the two buttons shows the setpoint shift.

When a setpoint shift is indicated the LED evaluates the value of the "Current setpoint shift" object and switches either on or off, depending on the parameter configuration in the ETS.

#### "Comparator without sign (1 byte)" and "Comparator with sign (1 byte)"

The status LED can indicate whether a parameterized reference value is greater than, equal to or less than the 1-byte object value of the status object. This comparator can be used for unsigned (0 … 255) or for signed integers (-128 … 127). The data format of the comparison is defined by the function of the status LED.

The status LED lights up only if the comparison is "true".

i After a reset or after ETS programming, the value of the LED object is always "0".

#### "Logic link"

The status LED shows the initial state of the internal logic gate. The logic link is separated from the button or rocker function. The logic gate has up to 8 input objects. The inputs can optionally be OR, AND or XOR linked. The status-LED is switched on if the initial state corresponds to "1".

i After a bus reset or after ETS programming, the value of the LED objects is always "0".

#### "Bit-coded evaluation"

The bit-coded evaluation logically links individual bits. The bits to be linked are selected in the ETS. The type of evaluation (1-byte, 2-byte or 4-byte) can be parameterised for this purpose. The status LED is switched on according to the parameterised linking behaviour (AND, OR).

i After a bus reset or after ETS programming, the value of the LED object is always "0".

## 8.2.2 Colour settings

### User-defined colour setting

The colour of the status LEDs can be adjusted. If no 3-colour individual control is configured in the parameterisation, red, green and blue can be selected from for the status LED colours in the ETS. In the colour configuration a distinction is made between whether all of the status LEDs have the same colour (common colour setting), or whether alternatively various colours can be configured for each LED (separate colour setting).

The difference is as follows:

– All status LEDs have the same colour.

  If common colour setting is desired, then the "Colour" parameter on parameter page "General -> Status-LED" must be configured to the settings "red", "green" or "blue". The status LEDs light up unchangeably in the configured colour later during operation, if they are switched on.

– The status LEDs have various colours.

  If the separate colour setting is desired, then the parameter "Colour" on parameter page "General -> Status-LED" must be configured to the setting "Colour selection per status LED". In this case additional parameters become visible on the parameter pages of the individual status LEDs. The parameters "Colour of the status LED" can then be used individually to define the desired colour for each status LED. The LED lights up in the configured colour if it is subsequently switched on regularly in operation, in accordance with the basic configuration "Function of the status LED".

### Superposed function

– In addition, with separate colour setting it is possible to configure a superposed function separately for each status LED. The superposed function can be used change the colour of a status LED via a communication object during operation of the device. It is also possible here to change the display function.

  When a superposed function is enabled, additional parameters and a communication object become visible in the ETS. It is thus possible to configure which colour the status LED should have when a superposed function is active, and which display function is then executed. The parameter "Selection of the superposed function" defines the display function, and thus the data format of the object. The following selections are available:
  "Control via separate LED object" (1-bit) or alternatively
  "Comparator without / with sign" (1-byte). The fundamental function of these superposed display functions is the same as the functions of the basic display of a status LED.
  In accordance with the selection of the display function and the parameter configuration resulting from it, the superposed function can be switched on or off via the 1-bit or 1-byte object. The status LED will only light up in the superposed colour when a function is switched on. When a superposed function is switched off the status LED will be activated according to its basic configuration (regular colour and display function).

### Automatic colour change

For the user-defined colour settings, an automatic colour change can be configured for the following LED functions:

- Operating mode display
- Controller status display (controller general)
- Setpoint value shift display
- Comparator without sign
- Comparator with sign

In this case, the colour of the corresponding status LED does not depend on the user specification via ETS parameter or communication object (superimposed function). Instead, the device then automatically decides which in colour the status LED should light up, based on the function value. The parameter "Status LED = ON for" indicates the colour which is set in dependence on the function value. This parameter cannot be changed.

The superposed function is initially always inactive after a device reset. The superposed function is only executed when a telegram is received via the corresponding object.

Regardless of the basic configuration of the status LED and the superposed function, the LEDs always flash red when an LED alarm message is active. An alarm message has a higher priority and thus overrides the basic display and the superposed function. After deactivation of the LED alarm signalling, all LEDs will immediately return to the state corresponding to their configuration and communication objects.

During colour configuration it must be ensured that different colours are configured for the basic display and the superposed function. If this is not done (the colours are the same), then when the display is static it is not possible to determine which display function is being indicated.

When the superposed function is activated via a 1-bit object it is possible to have the status LED flash in the superposed colour. During flashing the status LED switches cyclically between the "switched-on" and "switched-off" states. No colour change is performed between the regular colour and the superposed colour.

The configuration of user-defined colours of the status LED and the superimposed functions is only possible in the described manner when the 3-colour individual control is not parameterised .

### 3-colour individual control with separate communication objects

Each status LED can visualise statuses completely separately from the buttons, using three separate communication objects. In this case, a 1-bit object can be configured in the ETS for each colour, whereby each status LED can be statically switched on or off via the received object value, or also activated as flashing. At the same time, the colour of the LED is predefined by the actuated object.

The colour in which the LED lights up depends on which of the three objects of the LED concerned last received a value. The telegram polarity can be configured on the parameter page of the status LED.

The order in which the telegrams are received is saved internally when different colours of the LED are switched on or off again in succession. Since the colour that was last switched on is always displayed, colours that were previously switched on are saved. The saved colour is switched on after switching off the active colour.

| Example | |
|---|---|
| Initial state | LED is switched off |
| LED green on | LED turns green |
| LED blue on | LED turns blue |
| LED red on | LED turns red |
| LED blue off | LED turns red |
| LED red off | LED turns green |

If two or all of the three objects of a LED are associated with a group address, no receiving sequence is recognisable. in this case, it is stipulated that the sequence green -> blue -> red then applies.

For a 3-colour individual control, the telegram polarity of the LED control can be configured on the parameter page of a status LED as follows:
- 1 = LED static ON / 0 = LED static OFF
- 1 = LED static OFF / 0 = LED static ON
- 1 = LED flashes / 0 = LED static OFF
- 1 = LED static OFF / 0 = LED flashes

A flashing LED always changes between the active colour and the OFF state. Flashing between several colours, e.g. red and green, is not possible.

The configuration of the 3-colour individual control as described is only possible when no user-defined colour configuration has been parameterised .

## 8.2.3 Brightness settings

The brightness of all status LEDs is defined in the ETS. The "Brightness" parameter on the "General -> Status-LED" parameter page can be used to set the regular brightness of all status LEDs in 6 levels (level 0 = OFF, level 1 = dark, ..., level 5 = bright).

### Night reduction

Optionally the brightness can be changed during operation of the device, controlled by the night reduction.

Changing may be advisable, for example, to reduce the brightness during nighttime hours. If change-over of the brightness via the object is required, the "LED night reduction" must be activated on the "General -> Status-LED" parameter page. In this case the "Night reduction" communication object becomes visible in the ETS. As soon as a "1" telegram is received via this object, the device switches over to the "Brightness of all LEDs in night reduction" configured in the ETS. If a "0" telegram is received via the object, the device switches back to regular brightness.

The change-over of the LED brightness is always performed softly by means of a brief dimming process. Dimming up to a higher level value results in quicker dimming than with dimming to a lower level value. This results in a slow soft dimming that is pleasing for the human eye. The dimming speeds are fixed and therefore not changeable.

After a button has been actuated while night reduction is active, the illuminated LEDs of the device is controlled to light up with regular brightness for 30 seconds. Increasing the brightness, especially with significantly reduced brightness values or LEDs even switched off, permits status changes to be identified more easily (or identified at all) in night mode.

In the ETS it is possible to perform configuration in accordance with the possible selection of required stage values for the regular and reduced brightness. No check is made whether a reduced brightness level is configured for the reduced brightness level. This also makes it possible to use the object to switch over the object to larger brightness levels in comparison to the regular brightness. It is recommended, however, to set the brightness value for the night reduction lower than the regular brightness.

After a device reset, the regular brightness for switched-on LEDs is always effective. Switch-over by night reduction will only take place when a telegram is written to the respective object after a reset.

When the status LED is activated via the regular display function or via the superposed function, it is possible to let the status LED flash. During flashing the LEDs switch synchronously between the "switched-on" and "switched-off" states in the active brightness. This is not interpreted as a change of state of the display function, by means of which the brightness is therefore also not switched over automatically.

When LED alarm signalling is active, the status LEDs of the device always flash with the regular brightness. The device automatically deactivates LED night reduction as long as LED alarm signalling is active and updates it when LED alarm signalling is switched off and the night reduction object still is at "1"-active.

### 8.2.4 Table of parameters

The following parameters are configured on the "General" -> Status-LED" parameter page.

| | |
|---|---|
| Function and colour | **user-defined (function selection per status-LED)**<br>3-colour-individual control via objects |
| At this point, the display concept of the status LED is selected. In the setting "User-defined", the usual LED functions are available, e.g. status display or button-press display. These also include the user-defined colour selection and the superimposed display of several functions. The settings are selected separately for each status LED on the corresponding parameter page.<br>Alternatively, the different colours of each status LED can be addressed via their own communication objects. The setting "3-colour individual control via objects" is valid for all the status LEDs. Three objects are created for the three colours (switching colour red, switching colour green and switching colour blue) for each status LED (TSM & TSEM). The object which received its value last determines the colour that the status LED lights up in. This setting causes the additional parameter "Control of the status LED via object value" to be shown on the parameter pages of the status LED. | |

| | |
|---|---|
| Colour | **red**<br>green<br>blue<br>Colour selection per rocker/button |
| With a user-defined colour configuration, a distinction is made between whether all of the status LEDs have the same colour (settings "red", "green" or "blue"), or whether alternatively various colours can also be configured for the LEDs (setting "colour selection rocker/button"). With colour selection per rocker switch/button it is possible to set the colour on the parameter pages of the individual status LEDs.<br>This parameter is only visible with user-defined function and colour selection. | |

| | |
|---|---|
| Light duration of status LED for button-press display | 1 s<br>2 s<br>**3 s**<br>4 s<br>5 s |
| This parameter defines the switch-on time the status LED is lit up to indicate actuation. The setting concerns all status LEDs whose function is set to "Button-press display". | |

| Brightness | Level 0 (OFF)<br>Level 1 (dark)<br>Level 2<br>Level 3<br>**Level 4**<br>Level 5 (bright) |
|---|---|
| The brightness level for all status LEDs is defined at this point. | |

| Night reduction | Active<br>**Inactive** |
|---|---|
| Whether the parameter and communication object for reducing the brightness for all status LEDs is defined here.<br><br>i If a button operation takes place when reduced brightness is activated, then all the active displays are displayed with the regular brightness for a period of about 30 seconds. | |

| Brightness for night reduction | Level 0 (OFF)<br>Level 1 (dark)<br>**Level 2**<br>Level 3<br>Level 4<br>Level 5 (bright) |
|---|---|
| The brightness of all status LEDs, the operation LED and the labelling field illumination is reduced to the specified level as soon as the communication object "LED night reduction" receives this in accordance with the configured polarity.<br><br>i There is no check of whether the reduced level has a lower value than the regular brightness level. | |

| Polarity of the "LED night reduction" object | **1 = active / 0 = not active**<br>0 = active / 1 = not active |
|---|---|
| This parameter defines the polarity of the "LED night reduction" object. | |

The following parameters are configured on the parameter pages "Status-LED *n* - function".

| | |
|---|---|
| Function of status LED | always OFF<br>always ON<br>**Button-press display**<br>Telegram acknowledgment<br>Status indication<br>inverted status display<br>Control via separate LED object<br>Operating mode display<br>Controller status indication<br>Setpoint value shift display<br>Presence status<br>Inverted presence status<br>Comparator without sign (1 byte)<br>Comparator with sign (1 byte)<br>Logic link<br>Bit-coded evaluation |
| The ETS automatically compiles the selection of functions of the status LED depending on the set rocker or button function. Only functions that make sense in combination with the parameterised rocker or button function are offered for selection. | |

The following selection of LED basic functions can be configured for each rocker or button function.

| | |
|---|---|
| Function of status LED | always OFF<br>always ON<br>Control via separate LED object<br>Operating mode display<br>Controller status indication<br>Comparator without sign (1 byte)<br>Comparator with sign (1 byte)<br>Logic link<br>Bit-coded evaluation |

always OFF: Irrespective of the pushbutton or rocker function, the status LED is switched off permanently.

always ON: Irrespective of the pushbutton or rocker function, the status LED is switched on permanently.

Control via separate LED object: The status LED indicates the state of its own, separate 1-bit LED object. This setting causes the additional parameter "Control of the status LED via object value" to be shown.

Operating mode display: The status LED indicates the state of a KNX room temperature controller via a separate 1-byte communication object. This setting causes the additional parameter "Status LED ON with" to be shown.

Controller status display: The status LED indicates the state of the internal room temperature controller or the controller extension. This setting causes the additional "controller status" and "Status LED on with" parameters to be displayed.

Comparator without sign (1 byte): The status LED is activated depending on a comparison. In this configuration there is a separate 1-byte communication object available via which the unsigned reference value (0...255) is received. This setting causes the additional parameter "Status LED ON with" to be shown.

Comparator with sign (1 byte): The status LED is activated depending on a comparison. In this configuration there is a separate 1-byte communication object available via which the positive or negative reference value (-128...127) is received. This setting causes the additional parameter "Status LED ON with" to be shown.

Logic link: The status LED indicates whether the parameterised logic behaviour is fulfilled. The number of logic inputs that affect the logic output (status LED) according to the parameterised logic behaviour can be configured.

Bit-coded evaluation: The status LED indicates whether the parameterised linking behaviour is fulfilled. Up to 32 bits can be evaluated. All activated bits affect the lighting behaviour of the status LED according to the parameterised link behaviour.

The following selection of status LED functions can be configured **in addition** to the basic functions for the rocker or button functions "switching" and "dimming".

| Function of status LED | Button-press display<br>Status indication<br>inverted status display |
|---|---|
| button press display: The status LED indicates a button actuation. The ON time is set on the parameter page "General" in common for all status LEDs that are configured as actuation displays.<br>status display: The status LED indicates the state of the communication object "Switching". If the object value is "ON", the status LED is illuminated. If the object value is "OFF" the status LED is switched off.<br>inverted status display: The status LED indicates the state of the communication object "Switching". If the object value is "OFF", the status LED is illuminated. If the object value is "ON" the status LED is switched off. | |

The following selection of status LED functions can be configured **in addition** to the basic functions for the rocker or button function "2-channel operation".

| Function of status LED | Button-press display<br>Telegram acknowledgment |
|---|---|
| button press display: The status LED indicates a button actuation. The ON time is set on the parameter page "General" in common for all status LEDs that are configured as actuation displays.<br>telegram acknowledge: The status LED indicates the transmission of a telegram in 2-channel operation. | |

The following selection of status LED functions can be configured **in addition** to the basic functions for the rocker or button function "controller extension -> presence button"

| Function of status LED | Button-press display<br>Presence status<br>Inverted presence status |
|---|---|
| button press display: The status LED indicates a button actuation. The ON time is set on the parameter page "General" in common for all status LEDs that are configured as actuation displays.<br>Presence status: The status LED indicates the state of the presence button of the controller operation or in case of controller extension operation. The LED lights up if the presence function is activated. The LED is off if the presence function is inactive.<br>Presence status: The status LED indicates the state of the presence button of the controller operation or in case of controller extension operation. The LED lights up if the presence function is inactive. The LED is off if the presence function is activated. | |

The following selection of status LED functions can be configured **in addition** to the basic functions for the rocker or button function "controller extension -> setpoint shift"

| Function of status LED | Button-press display<br>Setpoint value shift display |
|---|---|
| button press display: The status LED indicates a button actuation. The ON time is set on the parameter page "General" in common for all status LEDs that are configured as actuation displays.<br>Setpoint value shift display: The status LED indicates the state of a setpoint shift of the controller operation or in case of controller extension operation. This setting causes the additional parameter "Status LED" to be shown. | |

The following parameter is visible on the parameter pages "Status-LED *n* - function" if the function of the status LED is configured to "control via separate LED object".

| Control of the status LED via object value | **1 = LED static ON / 0 = LED static OFF**<br>1 = LED static OFF / 0 = LED static ON<br>1 = LED flashes / 0 = LED static OFF<br>1 = LED static OFF / 0 = LED flashes |
|---|---|
| This parameter defines the telegram polarity of the 1-bit object "status LED".<br>The LED can be switched on or off statically. In addition, the received switching telegram can be evaluated in such a way that the LED flashes. | |

The following parameter is visible on the parameter pages "Status-LED *n* - function" if the function of the status LED is configured to "Operating mode display".

| Status LED ON with | Automatic mode<br>**Comfort mode**<br>Standby mode<br>Night operation<br>Frost/heat protection mode |
|---|---|
| The values of a communication object with data type 20.102 "HVAC Mode" are defined as follows:<br>0 = Automatic<br>1 = Comfort<br>2 = Standby<br>3 = Night<br>4 = Frost/heat protection<br>The value "Automatic" is used only by the "forced operating mode switchover" objects.<br>The status LED is illuminated when the object receives the value configured here. | |

The following parameters are visible on the parameter pages "Status-LED *n* - function" if the function of the status LED is configured to "Controller status display".

| Controller status | **KNX compliant**<br>Controller general |
|---|---|
| Room temperature controllers can transmit their current status to the KNX. The data formats "KNX compliant" and "Controller general" are usually available for this. This parameter adjusts the "Controller status display" function of the status LED to the status message's status format of the room temperature controller.<br>The selection options of the "Status LED ON with" parameter as well as the available communication objects adapt depending on this setting.<br>In the "KNX-compliant" setting, the device provides the 3 communication objects "Controller status", "Controller status RTSM" and "Controller status RTC".<br>If the setting is "controller general", the device provides the 2 communication objects "Controller status" and "Controller status addition". | |

| Status LED ON with | **Controller error status**<br>Operating mode (Heating = 1 / Cooling = 0)<br>Controller disabled (dew point operation)<br>Frost alarm<br>Heat alarm |
|---|---|
| This parameter is only visible if "Controller status = KNX compliant".<br>The status LED shows the information of the controller status according to the parameterization. | |

| Status LED ON with | **Comfort mode**<br>Standby mode<br>Night operation<br>Frost/heat protection mode<br>Controller disabled<br>Heating / Cooling (Heating = 1 / Cooling = 0)<br>Controller inactive (deadband operation)<br>Frost alarm<br>Normal/Forced operating mode<br>Comfort mode extension<br>Open window<br>Additional level active<br>dew point alarm |
|---|---|
| This parameter is only visible if controller status = controller general.<br>The status LED shows the information of the controller status according to the parameterization. | |

The following parameters are visible on the parameter pages "Status-LED *n* - function" if the function of the status LED is configured to "Comparator without sign".

| Status LED ON with | **Reference value greater than received value**<br>Reference value less than received value<br>Reference value equal to received value |
|---|---|
| The status LED indicates whether the configured reference value is greater or less than or equal to the value of the "Status LED" object". | |

| Reference value | **0**...255 |
|---|---|
| This parameter defines the reference value to which the value of the "Status LED" object is compared. | |

The following parameters are visible on the parameter pages "Status-LED *n* - function" if the function of the status LED is configured to "Comparator with sign".

| Status LED ON with | **Reference value greater than received value**<br>Reference value less than received value<br>Reference value equal to received value |
|---|---|
| The status LED indicates whether the configured reference value is greater or less than or equal to the value of the "Status LED" object". | |

| Reference value | -128...**0**...127 |
|---|---|
| This parameter defines the reference value to which the value of the "Status LED" object is compared. | |

The following parameters are visible on the parameter pages "Status-LED *n* - function" if the function of the status LED is configured to "Logic link".

| Logic behaviour | **OR**<br>AND<br>XOR |
|---|---|
| The status LED indicates whether the parameterised logic behaviour is fulfilled.<br>If "OR", the status LED lights up as soon as an input is true ("1").<br>If "AND", the status LED lights up when all inputs are true ("1").<br>If "XOR", the status LED lights up when an odd number of inputs are true ("1"). | |

| Number of logic inputs | **2**...8 |
|---|---|
| Here, the number of logic inputs that affect the logic output (status LED) according to the parameterised logic behaviour is configured. | |

The following parameters are visible on the parameter pages "Status-LED *n* - function" if the function of the status LED is configured to "Bit-coded evaluation".

| Linking behaviour | **OR**<br>AND |
|---|---|
| The status LED indicates whether the parameterised linking behaviour is fulfilled.<br>If "OR", the status LED lights up as soon as an input is true ("1").<br>If "AND", the status LED lights up when all inputs are true ("1"). | |

| Type of evaluation | **1 bytes**<br>2 bytes<br>4 bytes |
|---|---|
| The type of evaluation and thus the number of bytes is configured here.<br>According to this setting, the ETS displays a table for selecting the individual bits. Up to 32 bits can be activated for the evaluation.<br>All activated bits affect the lighting behaviour of the status LED according to the parameterised link behaviour. | |

The following parameters on the parameter pages "Status-LED *n* - function" become visible for the following configuration:

- Function and colour = "user-defined (function selection per status-LED)"
- Colour = "Colour selection per rocker/button"

| Automatic colour change of the status LED | Active<br>**Inactive** |
|---|---|
| When user-defined colour settings are used, an automatic colour change can be configured here for the LED functions "Operating mode display", "Controller status display", "Setpoint shift display" and "Comparator" . If the function has been enabled (setting "Active"), the colour of the corresponding status LED does not depend on the user specification by ETS parameter or communication object (superimposed function). Instead, the device then automatically decides which in colour the status LED should light up, based on the function value.<br>If the setting is "Inactive", the colour of the status LED, and optionally a superimposed function, can be configured. | |

| Status LED ON with | Setting depends on the LED function / read-only |
|---|---|
| The parameter indicates the colour which is set for an automatic colour change, depending on the function value. This parameter is only visible for an automatic colour change and cannot be changed. | |

The following selection of colours for the status LED can be parameterised for the "always ON" function of the status LED.

| | |
|---|---|
| Colour of the status LED | **State On = red**<br>State On = green<br>State On = blue |
| If separate colour settings for the status LEDs are required, then this parameter can be used individually to define the desired colour for each status LED. The LED lights up in the configured colour if it is subsequently switched on regularly in operation of the device according to the basic configuration "Function of the status LED".<br>This parameter is only visible when no automatic colour change is configured. | |

The following selection of status LED colours can be configured for the functions "Button-press display", "Telegram acknowledgment", "Status display", "Inverted status display", "Activation via separate LED object", "Operating mode display", "Controller status display", "Presence status", "Setpoint value shift display", "Comparator ..." , "Logic link" and "Bit-coded evaluation".

| | |
|---|---|
| Colour of the status LED | **State Off = off, state On = red**<br>State Off = off, state On = green<br>State Off = off, state On = blue<br>State Off = red, state On = green<br>State Off = red, state On = blue<br>State Off = green, state On = red<br>State Off = green, state On = blue<br>State Off = blue, state On = red<br>State Off = blue, state On = green |
| If separate colour settings for the status LEDs are required, then this parameter can be used individually to define the desired colour for each status LED. The LED lights up in the configured colour if it is subsequently switched on regularly in operation of the device according to the basic configuration "Function of the status LED".<br>This parameter is only visible when no automatic colour change is configured. | |

The following parameters are configured on the parameter pages "Status-LED *n* - superimposed function".

| | |
|---|---|
| Superposed function | Active<br>**Inactive** |
| With separate colour setting it is additionally possible to configure a superposed function separately for each status LED. The superposed function can be used change the colour of a status LED via a communication object during operation of the device. It is also possible here to change the display function. The superposed function of a status LED is enabled when this parameter is configured to "active".<br>This parameter is only visible when no automatic colour change is configured. | |

| Selection of the superposed LED function | Control via separate LED object<br>Comparator without sign (1-byte)<br>Comparator with sign (1-byte) |
|---|---|
| Control via separate LED object: With the superposed function the status LED indicates the state of a separate 1 bit LED object. This setting causes the additional parameter "Control of the status LED via object value" to be shown.<br>Comparator without sign (1 byte): In the superposed function the status LED is activated depending on a comparison. In this configuration there is a separate 1-byte communication object available via which the unsigned reference value (0...255) is received. This setting causes the additional parameter "Superposed function ON with" to be shown.<br>Comparator with sign (1 byte): In the superposed function the status LED is activated depending on a comparison. In this configuration there is a separate 1-byte communication object available via which the positive or negative reference value (-128...127) is received. This setting causes the additional parameter "Superposed function ON with" to be shown. | |

| Colour of the status LED for superposed function | red<br>**green**<br>blue |
|---|---|
| If the superposed function is enabled, this parameter can be used to define the desired superposed colour. The LED lights up in the configured colour if the superposed function is later activated in operation of the push button sensor.<br>This parameter is only visible if the parameter "Superposed function" is set to "enabled". | |

| Control of the status LED via object value | **1 = superposed Funct. ON / 0 = superposed Funct. OFF**<br>1 = superposed Funct. OFF / 0 = superposed Funct. ON<br>1 = superposed Funct. flashes / 0 = superposed Funct. OFF<br>1 = superposed Funct. OFF / 0 = superposed Funct. flashes |
|---|---|
| If the "Selection of the superposed LED function" is set to "Control via separate LED object", this parameter can be used to specify the telegram polarity of the "Superposed polarity" 1-bit object of the status LED concerned.<br>The superposed function can be switched on or off statically. In addition, the received switching telegram can be evaluated in such a way that the LED flashes when a superposed function is active. | |

| Superposed function ON when | **Reference value greater than received value**<br>Reference value less than received value<br>Reference value equal to received value |
|---|---|
| In the superposed function the status LED indicates whether the configured reference value is greater or less than or equal to the value of the "Superposed value function" object".<br>This parameter is only visible when "Selection of the superposed LED function" = "Comparator without sign" / "Comparator with sign". | |

| Reference value (0 … 255) | **0**...255 |
|---|---|
| This parameter defines the reference value to which the value of the "Superposed value function" object is compared.<br>This parameter is only visible when "Selection of the superposed LED function" = "Comparator without sign". | |

| Reference value (-128 … 127) | -128...**0**...127 |
|---|---|
| This parameter defines the reference value to which the value of the "Superposed value function" object is compared.<br>This parameter is only visible when "Selection of the superposed LED function" = "Comparator with sign". | |

### 8.2.5 Object list

The following communication objects are available for the individual buttons or rockers, depending on the set operating concept. The name of the object corresponds to the selection of the operating concept and can be adjusted by the parameter "Name of ..." .

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 449, 473 | Switching | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1-bit object for activation of the status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 449, 473 | Operating mode display | Status LED *n* - Input | 1 bytes | 20,102 | C, -, W, -, - |
| 1-byte object for activation of the status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 449, 473 | Controller status - KNX-compliant | Status LED *n* - Input | 2 bytes | 20,101 | C, -, W, -, - |
| 2-byte object for activation of the status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 459, 483 | Controller status RTSM - KNX-compliant | Status LED *n* - Input | 1 bytes | 21,107 | C, -, W, -, - |
| 1-byte object for activation of the status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 460, 484 | Controller status RTC - KNX-compliant | Status LED *n* - Input | 2 bytes | 22,103 | C, -, W, -, - |
| 2-byte object for activation of the status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 449, 473 | Controller status - controller general | Status LED *n* - Input | 1 bytes | | C, -, W, -, - |
| 1-byte object for activation of the status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 459, 483 | Controller status addition - controller general | Status LED *n* - Input | 1 bytes | | C, -, W, -, - |
| 1-byte object for activation of the status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 449, 473 | Value (0...255) | Status LED *n* - Input | 1 bytes | 5,010 | C, -, W, -, - |
| 1-byte object for activation of the status LED according to the parameter setting. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 449, 473 | Value (-128...127) | Status LED *n* - Input | 1 bytes | 6,010 | C, -, W, -, - |
| 1-byte object for activation of the status LED according to the parameter setting. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 451, 475 | Logic link input 1 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 1 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 452, 476 | Logic link input 2 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 2 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 453, 477 | Logic link input 3 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 3 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 454, 478 | Logic link input 4 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 4 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 455, 479 | Logic link input 5 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 5 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 456, 480 | Logic link input 6 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 6 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 457, 481 | Logic link input 7 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 7 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 458, 482 | Logic link input 8 | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1 bit object for activation of the status LED according to the parameter setting. This object describes the input 8 of the logic link. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 450, 474 | Superposed function | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1-bit object for forced-control activation of the status LEDs. This can be used to change the colour and display information of individual status LEDs according to priority. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 450, 474 | Superposed function | Status LED *n* - Input | 1 bytes | 5,010 | C, -, W, -, - |
| 1-byte object for forced-control activation of the status LEDs. This can be used to change the colour and display information of individual status LEDs according to priority. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 450, 474 | Superposed function | Status LED *n* - Input | 1 bytes | 6,010 | C, -, W, -, - |
| 1-byte object for forced-control activation of the status LEDs. This can be used to change the colour and display information of individual status LEDs according to priority. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 451, 475 | Switching colour red | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1-bit object for activation of the red status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 452, 476 | Switching colour green | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1-bit object for activation of the green status LED. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 453, 477 | Switching colour blue | Status LED *n* - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1-bit object for activation of the blue status LED. | | | | | |

# 9 Channel-independent device functions

The following subchapters provide a description of the device functions. Each subchapter consists of the following sections:

- Functional description
- Table of parameters
- Object list

### Functional description

The functional description explains the function and provides helpful tips on project design and usage of the function. Cross references support you in your search for further information.

### Table of parameters

The table of parameters lists all parameters associated with the function. Each parameter is documented in a table as follows.

| Name of the parameter | Parameter values |
|---|---|
| Parameter description | |

### Object list

The object list specifies and describes all communication objects associated with the function. Each communication object is documented in a table.

| | |
|---|---|
| Object no. | This column contains the object number of the communication object. |
| Function | This column contains the function of the communication object. |
| Name | This column contains the name of the communication object. |
| Type | This column contains the length of the communication object. |
| DPT | This column assigns a datapoint type to a communication object. Datapoint types are standardized in order to ensure interoperability of KNX devices. |
| Flag | This column assigns the communication flags in accordance with the KNX specification. |
| K flag | activates / deactivates the communication of the communication object |
| L flag | enables externally triggered reading of the value from the communication object |
| S flag | enables externally triggered writing of the value to the communication object |
| Ü flag | enables transfer of a value |
| A flag | enables updating of an object value in case of feedback |
| I flag | enforces updating of the communication object value when the devices is switched on (reading at init) |

## 9.1 Temperature measurement

### Basic principles

The device possesses an integrated temperature sensor, using which the room temperature can be detected. Alternatively (e.g. if the device has been installed in an unfavourable location or in case of operation in difficult conditions, e.g. in a moist atmosphere) or additionally (e.g. in large rooms or halls), a second external sensor linked via bus telegrams can be used to determine the actual value.

The temperature measurement is activated on the "General" parameter page and configured on the "Room temperature measurement" parameter page.

When choosing the installation location for the device or the external sensors, the following points should be considered:

- The device or temperature sensor should not be used in multiple combinations, especially together with flush-mounted dimmers.
- Do not install the temperature sensor in the area of large electrical consumers (avoid heat influences).
- The push button sensor should not be installed in the vicinity of radiators 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, windows or ventilation units and at least 1.5 m above the floor.

### Temperature measurement and measured value determination

The device possesses an integrated temperature sensor. This temperature sensor can be used to measure the ambient temperature and forward it to a room temperature controller via the 2-byte object "Actual temperature".

The room temperature measurement can optionally be supplemented with an external sensor. The external sensor is linked to the device (for example a KNX room temperature controller) via the bus by means of the additional 2-byte communication object "External temperature".

The "Room temperature measurement by" parameter in the "Room temperature measurement" parameter node specifies the sensors to detect the room temperature. The following settings are possible:

- "internal temperature sensor"

  The temperature sensor integrated in the device is activated. Thus, the actual temperature value is determined only locally on the device.

- "internal sensor and received temperature value"

  With this setting the internal as well as the external temperature sensor is active. The external sensor must either be a KNX room temperature controller coupled via the "External temperature" 2-byte object or another bus device

with temperature detection.
When evaluating the internal and the external sensors, the real actual temperature is made up from the two measured temperature values. The weighting of the temperature values is defined by the "Creation 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 device 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
Receiving temperature value: 22.3 °C
Determination of measured value: 30 % to 70 %

-> TResult internal = T internal · 0.3 = 6.45 °C,
-> TResult external = Texternal = 22.3 °C · 0.7 = 15.61 °C
-> TResult actual = TResult internal + TResult external = 22.06 °C

### Transmission of the actual temperature

The determined actual temperature can be actively transmitted to the bus via the "Actual temperature"or "actual temperature not adjusted" 2-byte objects.

The room temperature can be transmitted to the bus either after a change in a parameterised temperature value or cyclically after a parameterised cycle time.

The value "0" deactivates the transmission when room temperature changes and the cyclical transmission of the room temperature. If both parameters are set to zero, the room temperature is not transmitted to the bus.

### Calibrating the measured values

In some cases during room temperature measurement, it may be necessary to adjust the single temperature values. Adjustment becomes necessary, for example, if the temperature measured by the sensors stays permanently below or above the actual temperature in the vicinity of the sensor. To determine the temperature deviation, the actual room temperature should be detected with a reference measurement using a calibrated temperature measuring device.

The "Calibration ..." parameter enables the temperature calibration to be calibrated in 0.1 K-levels. Thus, the calibration is only set statically once.

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.

The "Actual temperature" object is always used to transmit the adjusted temperature value to the bus.
When determining the measured value using combined sensors, the two adjusted values are used to calculate the actual value.

### 9.1.1 Table of parameters

"General" parameter page

| | |
|---|---|
| Temperature measurement | Active<br>**Inactive** |
| This parameter activates the temperature measurement. Additional parameters and objects become visible. | |

"Room temperature measurement" parameter page

| | |
|---|---|
| Room temperature measurement using | **internal temperature sensor**<br>Internal temperature sensor and external temperature |
| The "Room temperature measurement using" parameter specifies the sensors to detect the room temperature.<br>"internal temperature sensor": The temperature sensor integrated in the device is activated. Thus, the actual temperature value is determined only locally on the device. In this configuration, the feedback control will start directly after a device reset.<br>"Internal temperature sensor and external temperature": This setting is used to combine the selected temperature sources. The external temperature is received via the "External temperature" 2-byte object. | |

| | |
|---|---|
| Determination of measured value from internal / external ratio | 10% to 90%<br>20% to 80%<br>30% to 70%<br>40% to 60%<br>**50% to 50%**<br>60% to 40%<br>70% to 30%<br>80% to 20%<br>90% to 10% |
| The weighting of the measured temperature value for the internal and external sensors is specified here. That results in an overall value, which will be used for the further interpretation of the room temperature.<br>This parameter is only visible with "Room temperature measurement using = internal sensor and external temperature"! | |

| | |
|---|---|
| Internal sensor calibration | -12.8...**0**...12.7 |
| Determines the value in Kelvin by which the internal sensor’s measured value is adjusted.<br>This parameter is only visible when the temperature detection system requires an internal sensor. | |

| External temperature calibration | -12.8...**0**...12.7 |
|---|---|
| Determines the value in Kelvin by which the external sensor’s room temperature value is calibrated.<br>This parameter is only visible when the temperature detection system requires an external sensor. | |

| Request time for external temperature | **0**...255 |
|---|---|
| The polling time for the external sensor’s temperature value is specified here. In the "0" setting, the external sensor is not automatically polled by the controller. In this case, the sensor must transmit its temperature value itself.<br>This parameter is only visible when the temperature detection system requires an external sensor. | |

| Transmission after room temperature change by | 0...**0.2**...25.5 |
|---|---|
| Determines the size of the value change of the room temperature in Kelvin after which the current value is automatically transmitted to the bus via the "Actual temperature" object. If set to "0", the actual value will not be transmitted to the bus depending on a room temperature change. | |

| Cyclical transmission of room temperature | 0...**15**...255 |
|---|---|
| This parameter specifies whether and at what time in minutes the determined room temperature is to be periodically output via the "Actual temperature" object. If the setting is "0", the current room temperature will not be transmitted cyclically to the bus. | |

| Provide uncalibrated room temperature | Active<br>**Inactive** |
|---|---|
| If necessary, the unadjusted room temperature can additionally be transmitted to the bus as an information value via the object "Actual temperature, unadjusted" and, for example, be displayed in visualisations. This parameter enables the corresponding object. | |

### 9.1.2 Object list

The name of the following objects can be specified by the parameter "Name of the room temperature measurement".

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 501 | Actual-temperature | Toom temperature - output | 2 bytes | 9,001 | C, R, -,T, - |
| 2-byte object for displaying the actual temperature (room temperature) determined internally. Possible value range: -99.9 °C to +99.9 °C / Measuring range of internal temperature sensor: -40 °C to +125 °C.<br>The temperature value is always output in the format "°C". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 502 | External temperature | Room temperature - input | 2 bytes | 9,001 | C, -, W, -, - |
| 2-byte object for coupling an external KNX room temperature sensor or a controller extension. Thus cascading of multiple temperature sensors for room temperature measurement. Possible range of values: -99.9 °C to +99.9 °C.<br>The temperature value must always be specified in the format "°C". | | | | | |

The name of the following object can be specified by the parameter "Name of the further temperature measurement".

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 503 | Actual temp. not adjusted | Toom temperature - output | 2 bytes | 9,001 | C, R, -,T, - |
| 2-byte object for the display of the determined actual temperature. The actual temperature is either determined by the internal sensor or by a combination of the internal sensor with an external temperature. The output value does not take the configured value for calibration into account. Internal to external measured value formation is taken into account. Possible value range: -99.9 °C to +99.9 °C / Measuring range of internal temperature sensor: 0 °C to +40 °C.<br>The temperature value is always output in the format "°C". | | | | | |

## 9.2 Alarm signal

The device permits signalling of an externally reported alarm via its status LED. The alarm can be triggered, for instance, in case of burglary or fire through a KNX central alarm unit. The device signals an alarm by all status LEDs of the device flashing synchronously. This alarm indication can be separately enabled with the parameter "alarm signalling" on the "General" parameter page.

When alarm signalling is enabled, the ETS displays the communication object "Alarm signalling" and further alarm function parameters on a separate parameter page.

The "Alarm signalling" object is used as an input for activating or deactivating the alarm signal display. The polarity of the object can be selected. When the object value corresponds to the "Alarm" condition, all status LEDs are always flashing red with a frequency of approx. 2 Hz. If there is an alarm, the behaviour of the status LED as configured in the ETS for normal operation have no significance. The LEDs adopt their originally configured behaviour only after the alarm signalling function has been deactivated. Changes of the state of the LEDs during an alarm - if they are controlled by separate LED objects or if they signal push-button functions - are internally stored and recovered at the end of the alarm.

i Regardless of the colour configuration of the status LED in normal operation and with a superposed function, the LEDs always flash red when a display alarm signal is active.

i When a display alarm signal is active, the status LEDs of the device always flash with the regular brightness ("Brightness of all status LEDs" parameter). The device automatically deactivates night reduction as long as the alarm signalling display is active and updates it when alarm signalling is switched off and the night reduction object still is at "1"-active.

Apart from the possibility of deactivating an alarm signal via the alarm object, it can also be deactivated locally by a button-press on the device itself. The "Reset alarm signalling by a button actuation" parameter defines the button response during an alarm:

– If this parameter is set to "Active", active alarm signal displaying can be deactivated by a button-press on the device. This button function does not cause the configured function of the pressed button to be executed. Only after then next button-press will the configuration of the button be evaluated and a telegram be transmitted to the bus, if applicable.
– If "Active" has been selected, alarm signalling can only be deactivated via the alarm signalling object. A button function will always directly execute the configured button function.

i With the disabling function configured, alarm signalling cannot be reset by a disabled button.

If alarm signalling can be deactivated by a button actuation, the "Alarm acknowledgement object" parameter defines whether an additional alarm acknowledge telegram is to be transmitted to the bus via the separate object "Alarm signalling acknowledge" after triggering by this button-press.

Such an acknowledge telegram can, for instance, be sent via a 'listening' group address to the "Alarm signalling" objects of other push button sensors in order to reset the alarm status there as well. Attention must be paid during resetting of an alarm to the selectable polarity of the acknowledgement object.

If the setting is "Alarm when OFF and alarm reset when ON", the bus must actively write "0" to the alarm object in order to activate the alarm after a reset or after programming with the ETS.

An active alarm message is not stored so that the alarm indication is generally deactivated after a device reset or after programming with the ETS.

### 9.2.1 Table of parameters

"General" parameter page

| | |
|---|---|
| Alarm signal | Active<br>**Inactive** |
| This parameter can be used to enable alarm signal displaying.<br>When alarm signalling is enabled, the ETS displays further parameters and up to two further communication objects. | |

The following parameters are visible on the "alarm signalling" parameter page if LED alarm signalling has been activated.

| | |
|---|---|
| Polarity of the alarm signalling object | **Alarm when ON and alarm reset when OFF**<br>Alarm when OFF and alarm reset when ON |
| The alarm signalling object is used as an input for activating or deactivating alarm signal displaying. | |

| | |
|---|---|
| Reset alarm signalling by a button actuation? | **Active**<br>Inactive |
| If this parameter is set to "Active", active alarm signal displaying can be deactivated by a button-press on the device.<br>This button function does not cause the configured function of the pressed button to be executed. Only after then next button-press will the configuration of the button be evaluated and a telegram be transmitted to the bus, if applicable.<br>If "Active" has been selected, alarm signalling can only be deactivated via the alarm signalling object. A button-press will always execute the configured button function. | |

| | |
|---|---|
| Alarm acknowledgement object | Active<br>**Inactive** |
| If alarm signalling can be deactivated by a button actuation, this parameter defines whether an additional alarm acknowledge telegram is to be transmitted to the bus via the separate object "Alarm signalling acknowledge" after triggering by this button-press. | |

| | |
|---|---|
| Acknowledge alarm signalling by | ON telegram<br>**OFF telegram** |
| This parameter sets the polarity of the "Alarm signalling acknowledge" object.<br>This parameter presetting depends on the selected polarity of the alarm message object. | |

### 9.2.2 Object list

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 1 | Alarm signal | Alarm signalling - Input | 1-bit | 1,001 | C, -, W, -, - |
| 1-bit object for the reception of an alarm signalling (polarity configurable). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 4 | Alarm message acknowledge | Alarm signalling - Output | 1-bit | 1,001 | C, -, -, T, - |
| 1-bit object for transmitting the acknowledgement of an alarm signalling (polarity configurable). | | | | | |

## 9.3 Disabling function

### Configuration

With the 1-bit communication object "Disabling", the operating areas of the device can be partly or completely disabled. During a disable, the rockers or buttons can also temporarily execute other functions.

An active disable applies only to the functions of the rockers or buttons. The functions of the status LED and the temperature measurement are not affected by the disabling function.

With alarm signalling configured, the alarm signalling cannot be reset by a disabled button.

The disabling function and the associated parameters and communication objects are enabled if the "Disabling function" parameter on the "General" parameter page is set to "Active".

You can parameterize the polarity of the disabling object. In case of polarity inversion (disabled = 0 / enabled = 1), the disabling function is not activated immediately after a bus reset or after ETS programming (object value = "0"). There must first be an object update "0" until the disabling function will be activated. Telegram updates from "0" to "0" or from "1" to "1" on the "Disabling" object remain without effect.

After a device reset, the disabling function is deactivated and must be activated via the bus.

### Configuring the reaction during a disable

In an active disable, either all buttons of the device or only individual buttons may be affected by the disable. You can furthermore define in the ETS whether disabled buttons will not show any response when pressed or, alternatively, will behave like another button of the device. This can be used to limit the control function of the device completely or partially.

Precondition: The disabling function must be activated.

- Set the "Button assignment" parameter to "All buttons assigned".

  The disabling function affects all buttons. As soon as any button of the device is pressed while a disabling function is active, the device executes the "behaviour when a disabling function is active".

- Set the "Button assignment" parameter to "Individual buttons assigned".

  The disabling function affects only the buttons that are assigned on the "Disable function" parameter page. As soon as one of the assigned buttons is pressed while a disabling function is active, the "Behaviour when a disabling function is active" for this button is executed. All other, non-disabled buttons respond normally when pressed.

- Set the parameter "Behaviour when a disabling function is active" to "No response when pressed".

The disabled buttons do not respond when pressed. The status LEDs of the disabled buttons remain off if the display function is configured to "Button-press display" or "Telegram acknowledgement".

- Set the parameter "Behaviour when a disabling function is active" to "No response when pressed like...". Also configure the parameters "All assigned top buttons behave like" and "All assigned bottom buttons behave like" to the required button number or disabling function as a reference button.

  All buttons assigned to the disabling function behave as defined in the parameters for the two specified reference buttons of the device. Different or identical buttons can be configured separately for all the top and bottom operating buttons. The two 'virtual' disabling functions of the device can also be configured as a reference button.
  The telegrams are transmitted to the bus via the communication objects of the specified reference buttons. The status LEDs of the reference buttons are controlled according to their function. The status LEDs of the disabled buttons remain off if the display function is configured to "Button-press display" or "Telegram acknowledgement".

i If a button evaluation is taking place at the time of activation / deactivation of a disabling function, this function is aborted immediately and with it also the pertaining button function. It is first necessary to release all buttons before a new button function can be executed if so permitted by the state of disabling.

### 9.3.1 Table of parameters

"General" parameter page

| | |
|---|---|
| Disabling function | Active<br>**Inactive** |
| With this parameter, the disabling function of the device can be centrally activated.<br>If "Active", the ETS shows further communication object and parameters. | |

"Disable function" parameter page

| | |
|---|---|
| Polarity of disabling object | **1 = disable / 0 = enable**<br>0 = disable / 1 = enable |
| This parameter defines the value of the disabling object at which the disabling function is active. | |

| | |
|---|---|
| Button assignment | **All buttons assigned**<br>individual buttons assigned |
| If disabling is active, either all buttons of the basic device or extension, or only individual buttons may be affected by the disable. This can be used to limit the control function of the device completely or only partially.<br>"All buttons assigned": The disabling function affects all buttons. As soon as any button of the device is pressed while a disabling function is active, the "Behaviour when a disabling function is active" is executed.<br>"Individual buttons assigned": The disabling function affects only the assigned buttons. As soon as one of the assigned buttons is pressed while a disabling function is active, the "Behaviour when a disabling function is active" for this button is executed. All other, non-disabled buttons respond normally when pressed. | |

| | |
|---|---|
| Button *n* | Active<br>**Inactive** |
| With the setting "Individual buttons assigned", these parameters will be shown whenever a button can be assigned to the disabling function. The parameters define the assignment of the buttons to the disabling function. | |

| | |
|---|---|
| Behaviour at the beginning of the disabling function | **no reaction**<br>Reaction like button *n* when pressed<br>Reaction like button *n* when released<br>Reaction like disabling function *n* when pressed<br>Reaction like disabling function *n* when released |
| Besides disabling of rocker and button functions, the device can also trigger a specific function immediately at the time of activation of the disabling state.<br>This function can:<br>– correspond to the function assigned to any of the buttons in the non-disabled state ("Reaction like button *n* …") and<br>– be defined on the following parameter pages ("Reaction like disabling function *n* …"). | |

| | |
|---|---|
| Behaviour when the disabling function is active | **no reaction to button-press**<br>Reaction to a button-press like... |
| You can define here whether disabled buttons will not show any response when pressed or, alternatively, will behave like another button of the device or like a virtual disabling function.<br>"no reaction to button-press": The disabled buttons do not respond when pressed.<br>"Reaction to a button-press like...": The disabled buttons can either execute the function of a button that has already been configured or the function of a separate disabling function. The parameters "All assigned top buttons behave like" and "All assigned bottom buttons behave like" define the function of the buttons assigned to the disabling function. | |

| | |
|---|---|
| All assigned upper buttons behave as | **Button 1**<br>Button 2 ...<br>(Selection depends on device variant!)<br>Disabling function 1<br>Disabling function 2 |
| If a specific button function is to be assigned during disabling to all or to individual buttons, this parameter can be used to select the desired button the function of which will then be executed. During disabling, all assigned upper buttons behave like the one parameterized here.<br>The desired functions can either correspond to the function of an existing button or they can be configured as special disabling functions.<br>This parameter is only visible with "Behaviour when a disabling function is active" = "Reaction to a button-press like..."! | |

| | |
|---|---|
| All assigned lower buttons behave as | **Button 1**<br>Button 2 ...<br>(Selection depends on device variant!)<br>Disabling function 1<br>Disabling function 2 |
| If a specific button function is to be assigned during disabling to all or to individual buttons, this parameter can be used to select the desired button the function of which will then be executed. During disabling, all assigned lower buttons behave like the one parameterized here.<br>The desired functions can either correspond to the function of an existing button or they can be configured as special disabling functions.<br>This parameter is only visible with "Behaviour when a disabling function is active" = "Reaction to a button-press like..."! | |

| | |
|---|---|
| Behaviour at the end of the disabling function | **no reaction**<br>Reaction like button *n* when pressed<br>Reaction like button *n* when released<br>Reaction like disabling function *n* when pressed<br>Reaction like disabling function *n* when released |
| Besides disabling of rocker and button functions, the push-button sensor can also trigger a special function immediately at the end of disabling.<br>This function can:<br>– correspond to the function assigned to any of the buttons in the non-disabled state ("Reaction like button *n* …") and<br>– be defined on the following parameter pages ("Reaction like disabling function *n* …") | |

Parameter page "Disabling function -> Disabling function 1 / Disabling function 2"

The functions "switching", "dimming", "Venetian blind", "value transmitter", "scene extension", "2-channel operation" and "controller extension" are available for the two disabling functions. These functions behave like the button functions of the device (same parameters).

## 9.3.2 Object list

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 9 | Disabling | Disabling function - input | 1-bit | 1,002 | C, -, W, -, - |
| 1-bit object for transmission of switching telegrams (ON, OFF). | | | | | |

**Disabling function: Switching**

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 61, 64 | Switching | Disabling function *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmission of switching telegrams (ON, OFF). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 62, 65 | Switching feedback | Disabling function *n* - input | 1-bit | 1,001 | C, -,W, -, U |
| 1-bit object for receiving feedback telegrams (ON, OFF). | | | | | |

**Disabling function: Dimming**

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 105, 109 | Switching | Disabling function *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmission of switching telegrams (ON, OFF). | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 106, 110 | Dimming | Disabling function *n* - output | 4-bit | 1,007 | C, -, -, T, - |
| 4-bit object for the transmission of relative dimming telegrams. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 107, 111 | Switching feedback | Disabling function *n* - input | 1-bit | 1,001 | C, -,W, -, U |
| 1-bit object for receiving feedback telegrams (ON, OFF). | | | | | |

**Disabling function: Venetian blind**

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 137, 140 | Short time operation | Disabling function *n* - output | 1-bit | 1,008 | C, -, -, T, - |
| 1-bit object for the transmission of telegrams with which a Venetian blind or shutter drive motor can be stopped or with which the blind slats can be adjusted by short time operation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 138, 141 | Long-time operation | Disabling function *n* - output | 1-bit | 1,008 | C, -,W, T, - |
| 1-bit object for the transmission of telegrams with which a Venetian blind or shutter drive motor can be can be moved upwards or downwards. | | | | | |

### Disabling function: Value transmitter

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter 0...255 | Disabling function *n* - output | 1 bytes | 5,010 | C, -,W, T, - |
| 1-byte object for the transmission of values from 0 to 255. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter 0…100 % | Disabling function *n* - output | 1 bytes | 5,001 | C, -,W, T, - |
| 1-byte object for transmitting values from 0 to 100%. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter -128...127 | Disabling function *n* - output | 1 bytes | 6,010 | C, -,W, T, - |
| 1-byte object for the transmission of values from -128 to 127. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter 0…255 % | Disabling function *n* - output | 1 bytes | 5,004 | C, -,W, T, - |
| 1-byte object for transmitting values from 0 to 255%. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter 0…360° | Disabling function *n* - output | 1 bytes | 5,003 | C, -,W, T, - |
| 1-byte object for transmitting values from 0 to 360°. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter 0...65535 | Disabling function *n* - output | 2 bytes | 7,001 | C, -,W, T, - |
| 2-byte object for the transmission of values from 0 to 65535. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter -32768...32767 | Disabling function *n* - output | 2 bytes | 8,001 | C, -,W, T, - |
| 2-byte object for the transmission of values from -32768 to 32767. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Temperature value transmitter | Disabling function *n* - output | 2 bytes | 9,001 | C, -,W, T, - |
| 2-byte object for transmitting temperature values from 0 to 40 °C. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Brightness value transmitter | Disabling function *n* - output | 2 bytes | 9,004 | C, -,W, T, - |
| 2-byte object for transmitting brightness values from 0 to 1500 Lux. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Colour temperature value transmitter | Disabling function *n* - output | 2 bytes | 7,600 | C, -,W, T, - |
| 2-byte object for transmitting colour temperature values from 1000 to 10000 Kelvin. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter 3 bytes (colour wheel sequence) | Disabling function *n* - output | 3 bytes | 232,600 | C, -, -, T, - |
| 3-byte object for transmitting 3-byte colour information. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 199, 206 | Value transmitter 3-byte (brightness adjustment) | Disabling function *n* - output | 3 bytes | 250,600 | C, -, -, T, - |
| 3-byte object for transmitting 3-byte colour information. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 200, 207 | Colour hue (H) | Disabling function *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting the colour hue. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 201, 208 | Saturation (S) | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the saturation. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 202, 209 | Brightness (V) | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the brightness value. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 205, 212 | Feedback colour hue (H) | Disabling function *n* - output | 1 bytes | 5,003 | C, -,W, -, U |
| 1-byte object for receiving the colour hue. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 205, 212 | Feedback brightness value (V) | Disabling function *n* - output | 1 bytes | 5,001 | C, -,W, -, U |
| 1-byte object for receiving the brightness value. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 205, 212 | Value transmitter feedback RGB | Disabling function *n* - output | 3 bytes | 232,600 | C, -,W, -, U |
| 3-byte object for receiving 3-byte colour information. | | | | | |

### Disabling function: Scene extension

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 238, 241 | Scene extension | Disabling function *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for recalling or for storing one of 64 scenes max. from a scene push button sensor. | | | | | |

### Disabling function: 2-channel operation

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 switching | Disabling function *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmitting switching telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 switching | Disabling function *n* - output | 1-bit | 1,001 | C, -,W, T, - |
| 1-bit object for transmitting switching telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 value 0...255 | Disabling function *n* - output | 1 bytes | 5,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 value 0...255 | Disabling function *n* - output | 1 bytes | 5,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 value 0...100 % | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 value 0...100 % | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 value -128...127 | Disabling function *n* - output | 1 bytes | 6,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 value -128...127 | Disabling function *n* - output | 1 bytes | 6,010 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 value 0...255 % | Disabling function *n* - output | 1 bytes | 5,004 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 value 0...255 % | Disabling function *n* - output | 1 bytes | 5,004 | C, -, -, T, - |
| 1-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 value 0...65535 | Disabling function *n* - output | 2 bytes | 7,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 value 0...65535 | Disabling function *n* - output | 2 bytes | 7,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 value -32768...32767 | Disabling function *n* - output | 2 bytes | 8,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 value -32768...32767 | Disabling function *n* - output | 2 bytes | 8,001 | C, -, -, T, - |
| 2-byte object for transmitting value telegrams on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 temperature value | Disabling function *n* - output | 2 bytes | 9,001 | C, -, -, T, - |
| 2-byte object for transmitting temperature values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 temperature value | Disabling function *n* - output | 2 bytes | 9,001 | C, -, -, T, - |
| 2-byte object for transmitting temperature values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 brightness value | Disabling function *n* - output | 2 bytes | 9,004 | C, -, -, T, - |
| 2-byte object for transmitting brightness values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 brightness value | Disabling function *n* - output | 2 bytes | 9,004 | C, -, -, T, - |
| 2-byte object for transmitting brightness values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 scene (external) 1…64 | Disabling function *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 scene (external) 1…64 | Disabling function *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 2 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 scene (internal) 1…8 | Disabling function *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 scene (internal) 1…8 | Disabling function *n* - output | 1 bytes | 18,001 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 353, 366 | Channel 1 (RGB) | Disabling function *n* - output | 3 bytes | 232,600 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 354, 367 | Channel 2 (RGB) | Disabling function *n* - output | 3 bytes | 232,600 | C, -, -, T, - |
| 1-byte object for transmitting scene values on channel 1 if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 355, 368 | Channel 1 colour hue (H) | Disabling function *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting the colour hue, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 359, 372 | Channel 2 colour hue (H) | Disabling function *n* - output | 1 bytes | 5,003 | C, -, -, T, - |
| 1-byte object for transmitting the colour hue, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 356, 369 | Channel 1 saturation (S) | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the saturation, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 360, 373 | Channel 2 saturation (S) | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the saturation, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 357, 370 | Channel 1 brightness value (V) | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the brightness, if 2-channel operation is activated. | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 361, 374 | Channel 2 brightness value (V) | Disabling function *n* - output | 1 bytes | 5,001 | C, -, -, T, - |
| 1-byte object for transmitting the brightness, if 2-channel operation is activated. | | | | | |

**Disabling function: controller extension**

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 436, 443 | Operating mode switchover | Disabling function *n* - controller extension - output | 1 bytes | 20,102 | C, -,W, T, - |
| 1-byte object for changing over a room temperature controller between the Comfort, Standby, Night and Frost/heat protection operating modes.<br>This object is only visible if "Function = operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 437, 444 | Operating mode switchover feedback | Disabling function *n* - controller extension - input | 1 bytes | 20,102 | C, -,W, -, U |
| 1-byte object for receiving the operating mode of a room temperature controller.<br>This object is only visible if "Function = operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 436, 443 | Operating mode forced-control | Disabling function *n* - controller extension - output | 1 bytes | 20,102 | C, -,W, T, - |
| 1-byte object for changing over a room temperature controller under forced control between the Automatic, Comfort, Standby, Night and Frost / heat protection operating modes<br>This object is only visible if "Function = forced operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 437, 444 | Forced object operating mode feedback | Disabling function *n* - controller extension - input | 1 bytes | 20,102 | C, -,W, -, U |
| 1-byte object for receiving the operating mode of a room temperature controller.<br>This object is only visible if "Function = operating mode change-over". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 436, 443 | Presence function | Disabling function *n* - controller extension - output | 1-bit | 1,018 | C, -,W, T, - |
| 1-bit object for changing over the presence status of a room temperature controller.<br>This object is only visible if "Function = presence function". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 437, 444 | Presence function feedback | Disabling function *n* - controller extension - input | 1-bit | 1,018 | C, -,W, -, U |
| 1-bit object for receiving the presence status of a room temperature controller.<br>This object is only visible if "Function = presence function". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 436, 443 | Preset setpoint shifting | Disabling function *n* - controller extension - output | 2 bytes | 9,002 | C, -,W, T, - |
| 2-byte object for presetting a basic setpoint shift in Kelvin. The value "0" means that no shift is active . The values can be specified between -670760 K and 670760 K.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via offset (DPT 9.002)". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 437, 444 | Current setpoint shifting | Disabling function *n* - controller extension - input | 2 bytes | 9,002 | C, -,W, -, U |
| 2-byte object for receiving the feedback from the current basic setpoint shift in Kelvin.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via offset (DPT 9.002)". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 436, 443 | Preset setpoint shifting | Disabling function *n* - controller extension - output | 1 bytes | 6,010 | C, -,W, T, - |
| 1-byte object for presetting a basic setpoint shift. The value "0" means that no shift is active . The value is depicted in a two's complement in the positive or negative direction.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via levels (DPT 6.010)". | | | | | |

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 437, 444 | Current setpoint shifting | Disabling function *n* - controller extension - input | 1 bytes | 6,010 | C, -,W, -, U |
| 1-byte object for receiving the feedback from the current basic setpoint shift.<br>This object is only visible if "Function = Setpoint shift" and "Type of setpoint shift = Via levels (DPT 6.010)". | | | | | |

## 9.4 Energy saving mode

The device has an energy-saving mode to save electrical energy during operation. If the function is used, the device switches to the energy saving mode after a preset time without operation or controlled by an external telegram to a separate object. In the energy saving mode, essential display functions of the device are switched off. The status LED is then without any function. The energy saving mode can be deactivated by actuating a button or by a special telegram. Afterwards, the device is fully functional again.

The energy saving mode can only be parameterised in the ETS if no alarm message is parameterised!

### Activating energy saving mode

The device has two different activation options for setting the device to the energy saving mode. These can either be combined together or used separately.

Firstly, the device can be set to the energy saving mode by a group telegram via a communication object designated for this purpose. To do this, the telegram polarity that triggers the activation of the energy saving mode must be defined in the ETS.

Secondly, it is possible to switch to the energy saving mode automatically if no button has been actuated on the device within a defined time period. The time for this case is defined in the ETS. Each operation restarts the time for activating the energy saving mode.

On activation of energy-saving mode, all status LEDs are switched off under forced control.

Any activation attempts of the energy saving mode are ignored while the programming mode of the device is active.

### Deactivating energy saving mode

The device has two different options for deactivation of energy-saving mode, which can be optionally combined.

Firstly, it is always possible to deactivate energy-saving mode automatically, as soon as the device is operated.

On the other hand, deactivation can also take place by a group telegram via the communication object designated for this purpose. For this purpose, the telegram polarity that triggers the deactivation of the energy saving mode must be defined in the ETS.

If an operation deactivates the energy saving mode, the device always executes the configured operating function immediately as well (e.g. switching, dimming, etc ...).

If the transmission flag is set, then other devices can be informed of the deactivation of energy-saving mode by actuating a button on the local device, causing them also to leave energy-saving mode (precondition: all the devices are linked to the same group address and the deactivation via an object must be

possible in the parameterisation of the other devices). When energy-saving mode is deactivated when the transmission flag is set, the device sends an "Energy-saving mode deactivated" telegram to the bus, according to the inverted activated telegram polarity.

The device will activate the energy saving mode even if the control surfaces are disabled. The energy saving mode (first operation) can also be deactivated by a disabled button. The configured operating functions (switching, dimming...) will not be executed thereby, however.

### 9.4.1 Table of parameters

"General" parameter page

| | |
|---|---|
| Energy saving mode | Active<br>**Inactive** |
| The energy saving mode can be enabled here.<br>If the alarm signalling is enabled, the energy saving mode cannot be enabled. | |

The following parameters are visible on the "Energy saving mode" parameter page if the energy saving mode has been activated.

| | |
|---|---|
| Activating energy saving mode | **by object**<br>automatically by time<br>automatically by time or by object |
| This parameter defines how the energy saving mode is activated in the device.<br>Firstly, the device can be set to the energy saving mode by a group telegram via a communication object designated for this purpose.<br>Secondly, it is possible to switch to the energy saving mode automatically if no button has been actuated within a defined time period. | |

| | |
|---|---|
| Deactivating energy saving mode | **automatically on operation**<br>automatically on operation or via object |
| This parameter defines how the energy saving mode is deactivated in the device.<br>Firstly, it is possible to deactivate energy-saving mode automatically, as soon as the device is operated. If an operation of the device deactivates the energy saving mode, the device always executes the configured operating function immediately as well (e.g. switching, dimming, etc.).<br>Secondly, the energy saving mode can be deactivated by a group telegram via a communication object designated for this purpose. However, this possibility can only be combined with the automatic deactivation on operation. | |

| | |
|---|---|
| Polarity of the "Energy saving mode" object | **0 = activate / 1 = deactivate**<br>1 = activate / 0 = deactivate |
| This parameter defines the telegram polarity for the object for activating or deactivating the energy saving mode. | |

| | |
|---|---|
| Time to activate energy saving mode | **1**...60 min |
| This parameter defines the time that must elapse after an operation so that the device activates the energy saving mode. Each operation restarts the time. | |

### 9.4.2 Object list

| Object no. | Function | Name | Type | DPT | Flag |
|---|---|---|---|---|---|
| 35 | Activate / deactivate | Energy saving mode - input | 1-bit | 1,003 | C, -,W, -, U |
| 1-bit object for activating or deactivating the energy saving mode. If the transmission flag is set, then other devices can be informed of the deactivation of energy-saving mode through operation on the local device, causing them also to leave energy-saving mode (precondition: all the devices are linked to the same group address and the deactivation via an object must be possible in the parameterisation of the other devices). When energy-saving mode is deactivated when the transmission flag is set, the device sends an "Energy-saving mode deactivated" telegram to the bus, according to the inverted activated telegram polarity. | | | | | |

ALBRECHT JUNG GMBH & CO. KG
Volmestraße 1
58579 Schalksmühle
GERMANY

Telefon: +49 2355 806-0
Telefax: +49 2355 806-204
kundencenter@jung.de
www.jung.de