KNX

KNX Push-button Pro T

Universal 1841/1.0

Application description

This document describes the software application 1841/1.0. The software application is designed to program the KNX Push-button Pro T.

MTN6185-03xx | MTN6185-04xx | MTN6185-60xx

03/21-1841/1.0

Schneider Electric

Legal information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.

Trademarks

Other brands and registered trademarks are the property of the relevant owner.

Warnings

Read through the following instructions carefully and familiarise yourself with the device prior to installation, operation and maintenance. The warnings listed below can be found throughout the documentation and indicate potential risks and dangers, or specific information that clarifies or simplifies a procedure.

The addition of a symbol to "Danger" or "Warning" safety instructions indicates an electrical danger that could result in serious injuries if the instructions are not followed.

This symbol represents a safety warning. It indicates the potential risk of personal injury. Follow all safety instructions with this symbol to avoid serious injuries or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation that will inevitably result in serious or fatal injury if the instructions are not observed**.**

⚠ WARNING

WARNING indicates a possible danger that could result in death or serious injuries if it is not avoided.

⚠ CAUTION

CAUTION indicates a possible danger that could result in minor injuries if it is not avoided.

NOTE

NOTE provides information about procedures that do not present any risk of physical injury.

Symbols

ETS settings

Additional information

The information provided must be complied with, otherwise program or data errors may occur.

ETS operation

Requirements for safe operation

Knowledge of the basic rules for operating programs using Windows® is a prerequisite for operation.

The ETS is the **software for the KNX system**, and is not manufacturer-specific. Knowledge of ETS operation is required. This also includes selection of the correct sensor or actuator, transferring it to the line and commissioning it.

ETS tabs, parameters and values

Overview - setting functions

The following overview allows you to **understand** the steps needed to access the functions and settings. This overview also provides you with the correct sequence for accessing the functions.

Button	Select button function	Scene
	⌇	
	Select scene function	Extended
	Number of objects	Two
⌇		
Scene extended	...	...

Example

Meaning: First go to the *Button* tab and set the *Select button function* parameter to value *Scene*. Further parameters will then appear in the tab. These can be used to change settings. A new tab will also open.

Special features of the ETS software

Restoring defaults

You can set the factory-specified defaults using the *Default parameters* service button in the ETS5.

You can use the *Default* and *Default parameters* service buttons to switch all parameters back to the settings on delivery (following consultation). The ETS will then permanently delete all manual settings.

Express settings

You can use the *Express settings* to call up pre-set functions. Later, you simply connect group addresses to the functions. The group addresses then remain connected to the functions even if you change the assignment to the buttons (Express settings --> 11).

Extended settings

With the *Extended settings*, you can configure individual functions with a wide range of options, if necessary (Extended settings --> 25).

Dependent functions and parameters

Many functions are affected by how other functions are set. This means that dependent functions can only be seen and selected in the ETS when the upstream function is enabled.

- If you de-select functions or change parameters, previously connected group addresses may be removed in the process.
- The values of some parameters only become active once the functions influenced by these parameters are activated.
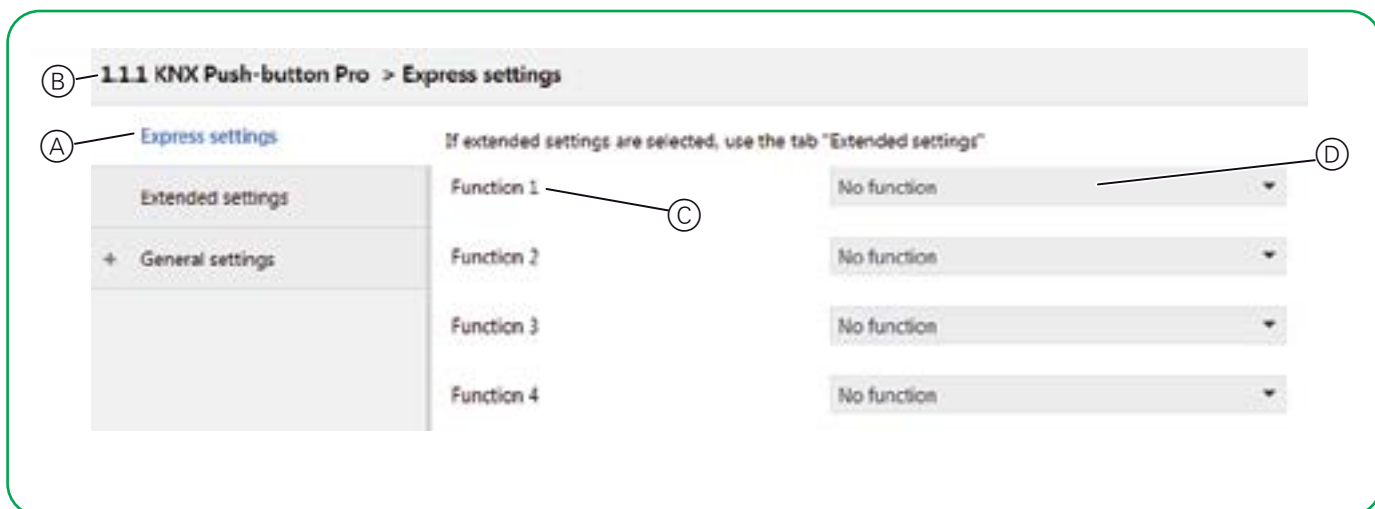
Appropriate ETS version

The application is suitable for ETS5. You cannot use earlier versions, such as ETS4 or ETS3.

User interface

In the ETS, the device parameters are opened using the *Edit parameters* service button. The user interface is divided into 2 sections: The tabs are on the left and the parameters on the right, together with their values.

- Ⓐ Tab
- Ⓑ Name of device
- Ⓒ Parameter
- Ⓓ Input fields for parameter values

Table of contents

1   For your safety

> ⚠️ **WARNING**
>
> **Risk of serious damage to property and personal injury due to incorrect electrical installation.**
>
> Safe electrical installation can only be ensured if the person in question can prove basic knowledge in the following areas:
> - Connection to installation networks
> - Connecting several electrical devices
> - Laying electric cables
> - Connecting and establishing KNX networks
> - Commissioning KNX installations
>
> These skills and experience are normally only possessed by certified specialists who are trained in the field of electrical installation technology. If these minimum requirements are not met or are disregarded in any way, you will be personally liable for any damage to property or personal injury.

1.1   Qualified personnel

This document is aimed at personnel who are responsible for setting up, installing, commissioning and operating the device and the system in which it is installed.

Detailed expertise gained by means of training in the KNX system is a prerequisite.

2 General information on the Universal 1841/1.0 application

You can use this application to program the KNX Push-button Pro T. You can adjust the number of buttons between 1 and 4. There is a status indication with adjustable colors and brightnesses for each of the up to 4 buttons.

You can use the *Express settings* to call up pre-set functions. Later, you simply connect group addresses to the functions. The group addresses then remain connected to the functions even if you change the assignment to the buttons ([Express settings --> 11](#)).

With the *General settings*, you can change the LED operating modes, the locking function for the buttons and the proximity function. These settings apply both to the configuration with *Express settings* and the configuration with *Extended settings* ([General settings --> 18](#)).

With the *Extended settings*, you can configure individual functions, such as the slider function or the extended edge function, if necessary ([Extended settings --> 25](#)).

You can find an overview of all the group objects for this ETS application at the end of this document: [Overview of group objects --> 66](#).

2.1 Components and programming environment

The device is commissioned using KNX-certified software. The application and the technical descriptions are updated regularly and can be found on the Internet.

This application can be run in conjunction with the ETS5 software.

2.2 Overview of application functions

Express settings

- Enable touchless function
- Pre-set button functions
    - No function
    - Toggle on/off
    - Switching on
    - Switching off
    - Dimming brighter/darker
    - Dimming brighter
    - Dimming darker
    - Moving blind up
    - Moving blind down
    - Scene
    - Enable extended settings
- Pre-set assignment of functions and buttons
    - Retain group addresses in the event of changes
- Select number of buttons (1–4)

General settings

- Proximity function
    - How to trigger the function
- Proximity sensor sensitivity
- LED modes
    - Adjustable colors and brightnesses
    - Normal and night mode
    - LEDs used in night mode
- Locking function for push-buttons
    - Lock with master button
- Temperature sensor
    - Units and offset
    - Sending behaviour

Extended settings

- Switch and toggle
    - 2 objects
    - 1 bit, 1 byte
- Dimming
    - Dimming steps
- Blind
    - Control with positional values
    - Single-button operation
- Edge function
    - 2 objects
    - Functions for short and long button actions
    - Cyclical sending and staircase lighting function
    - Priority control (2 bit)
    - Dimming (4 bit)
    - Dimming with values (1 byte)
    - Values (1 byte)
- Edges with 2 byte values
    - Floating point value
    - Whole number with sign
    - Whole number without sign
- 8 bit slider
    - With limit values
    - Send cyclically
    - Increase or reduce stepwise
- Scenes
    - Calling up and saving
    - Call up 2 scenes
    - Cyclical sending and staircase lighting function

Group addresses

- Maximum number of different group addresses: 250
- Maximum assignments: 250

3   Express settings

You can use the *Express settings* to call up pre-set functions. Later, you simply connect group addresses to the functions. The group addresses then remain connected to the functions even if you change the assignment to the buttons.

Express settings	Function x	No function
		Toggle On/Off
		Switching On
		Switching Off
		Dimming brighter/darker
		Dimming brighter
		Dimming darker
		Moving blind up
		Moving blind down
		Scene
		Enable extended settings

3.1   Touchless function

Proximity sensor

The touchless function uses a proximity sensor. As soon as a hand is within 5 cm in front of the surface, the sensor detects proximity and triggers the switching function.

Proximity sensor sensitivity

As the **color of the push-button** affects the range of proximity, you can adjust the proximity sensor sensitivity to low, medium, or high. Recommended values are listed in the ETS (tab *Proximity sensor sensitivity*).

If you prefer touchless operation, click *Enable touchless function*.

Express settings	Enable touchless function	Function1
General settings		
Proximity function		
Proximity sensor sensitivity		
LED modes		
Locking function for buttons		
Temperature sensor		

The number of buttons is then automatically set to 1.

Select from the drop-down menu and assign a function to the button:

Express settings	Enable touchless function	Function 1	Toggle On/Off
			Switching On
			Switching Off
			Scene

If you enable touchless function the touch buttons are inactive and extended settings unavailable.

Touchless function settings

General settings	
Touchless function	Touchless function timeout = 100 ms * factor (5-50)
	Disable touchless function by object (Yes/No)

Set up timeout range to avoid sending telegram too often (default value 500 ms). You can also disable touchless function by an object. Set up reaction parameters with *Touchless function active with* option (default value 1).

Group objects

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Switch object	Function 1	1 bit	Sends, receives	1.001 switch
1	Scene object	Function 1	1 byte	Sends	18.001 scene control
3	Status feedback object	Function 1	1 bit	Receives	1.001 switch
20	Disable touchless function	Touchless function	1 bit	Receives	1.003 enable

3.2   Configuration of buttons

If you disable touchless function the touch buttons are active and extended settings available.

You can activate a maximum of 4 buttons and assign 1 function to each. The push-button has 6 areas that are activated differently, depending on the number of buttons selected.

1 button   **2 buttons**   **3 buttons**   **4 buttons**
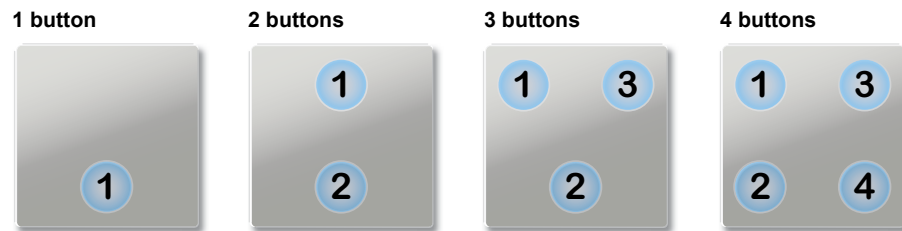
By default, 4 buttons are activated and 1 function is assigned to each. The default setting is a simple assignment of the 4 functions to 4 buttons.

Button	Function
Button 1	Function 1
Button 2	Function 2
Button 3	Function 3
Button 4	Function 4

You configure the push-button in 4 steps.

① Select 1 to 4 functions
② Define the number of buttons, from 1 to 4
③ Optionally, change the connection between button and function
④ Assign group addresses

You can also perform steps ② and ③ after connecting the group addresses.

The functions are fully set. With the Scene function, you set the scene address. A suitable actuation of the status indication is already configured for all functions. The following chapters describe the functions.

With the *Extended settings*, you can activate button functions and customise these to your needs (Extended settings --> 25).

Example of changing the assignment

Functions	4 buttons Default setting	4 buttons New assignment	2 buttons New assignment
Switch light 1	Button 1	Button 1	-
Switch light 2	Button 2	Button 3	-
Blind up	Button 3	Button 2	Button 1
Blind down	Button 4	Button 4	Button 2

3.3 Switching

With *Express settings*, you can switch the lighting or other consumers with 1 or 2 buttons. In single-button operation, you switch on and off alternately with 1 button. To do this, choose the function *Toggle On/Off*.

In two-button operation, you switch only either on or off with 1 button. To do this, choose the function *Switching On* or *Switching Off* for each button.

Express settings	Function x	Toggle On/Off
		Switching On
		Switching Off

Status indication

The following section describes the behaviour of the button illumination in the default setting.

In the express setting *Switching*, the status indication has 2 states.

State	Status indication
Illumination on	Illuminates bright white
Illumination off	Illuminates dark white

The status indication is controlled via the *Status feedback object* for the button function. The *Status feedback object* receives the values "1" and "0" sent from a connected channel.

Proximity function   In the default setting, the proximity function is activated (Proximity function --> 21).

Night mode   You can use the *Night mode input* object to toggle between normal and night mode. If no proximity is detected, one of the status indications for the push-button is switched on during night mode for orientation (Normal and night mode --> 19).

Chapter LED modes --> 18 contains the general settings for the LEDs. For example, you can also display the states of the status indication with 2 colors, green and white.

18 Group objects

The switching command function is carried out via the *Switch object*. The status indication is controlled via the *Status feedback object*.

Group objects for "Switching" express setting

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Switch object	Function 1	1 bit	Sends, receives	1.001 switch
3	Status feedback object	Function 1	1 bit	Receives	1.001 switch

Single-button operation

In the case of a two-way or central circuit, the switching channel is also switched via another sensor. With single-button operation, it is possible to adjust to the current state of the switching channel. To do this, the state of the switching channel ("1" or "0") has to be loaded to the push-button.

• To do this, also connect the group address of the *Status feedback object* for the connected channel to the *Switch object* for the button function.

3.4   Dimming

With *Express settings*, you can switch and dim the dimmable lighting with 1 or 2 buttons.

A short button action switches on or off. A long button action dims brighter or darker. Releasing after pushing and holding the button ends the dimming process. To aid operation, the status indication flashes during a long button action.

If single-button dimming is activated, you switch alternately on and off and dim alternately brighter and darker. If two-button dimming is activated, you choose the function *Dimming brighter* or *Dimming darker* for each button.

Express settings	Function x	Dimming brighter/darker
		Dimming brighter
		Dimming darker

Status indication

The following section describes the behaviour of the button illumination in the default setting.

In the express setting *Dimming*, the status indication has 3 states.

State	Status indication
Operation	Illuminates bright white
Long operation	Flashes white
Release	Illuminates dark white

The status indication is controlled via the *Status feedback object value* for the button function. If the lighting is switched on and the brightness level > 0, the LED illuminates.

The status indication also evaluates the operating time and flashes if you press and hold the button. This way, you can see that the lighting will be dimmed brighter or darker.

Proximity function   In the default setting, the proximity function is activated (Proximity function --> 21).

Night mode   You can use the *Night mode input* object to toggle between normal and night mode. If no proximity is detected, one of the status indications for the push-button is switched on during night mode for orientation (Normal and night mode --> 19).

Chapter LED modes --> 18 contains the general settings for the LEDs. For example, you can also display the states of the status indication with 2 colors, green and white.

Group objects

Switching is carried out via the *Switch object*. Dimming is carried out via the *Dimming object*. The status indication is controlled via the button operation and the *Status feedback object*.

Group objects for "Dimming" express setting

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Switch object	Function 1	1 bit	Sends, receives	1.001 switch
2	Dimming object	Function 1	4 bit	Sends, receives	3.007 dimmer step
3	Status feedback object	Function 1	1 bit	Receives	1.001 switch

Single-button operation

In the case of a 2-way or central circuit, the dimmer is also switched and dimmed via another sensor. With single-button operation, it is possible to adjust to the current state of the dimming channel. To do this, the state of the dimming channel has to be loaded to the push-button.

- To load the switching command function, connect the group address of the 1 bit *Status feedback object* for the connected channel to the *Switch object* for the button function.
- To load the dimming direction, also connect the group address of the *Dimming object* for the other sensor to the *Dimming object* for the button function. This is only necessary if the group addresses are different.

3.5   Blind

You can use the *Express settings* to control the blinds with 2 buttons. To do this, choose the function *Moving blind up* or *Moving blind down* for each button.

You can now move the blind either up or down by pressing and holding the corresponding button. A short button action stops the travel. A short button action also adjusts the slats in steps. To aid operation, the status indication lights up when pressed briefly and flashes when pressed and held.

Express settings	Function x	Moving blind up
		Moving blind down

Status indication

The following section describes the behaviour of the button illumination in the default setting. In the express setting Blind, the status indication has 3 states.

State	Status indication
Operation	Illuminates bright white
Long operation	Flashes white
Release	Illuminates dark white

Activation on operation

The status indication lights up when pressed briefly and flashes when pressed and held. The status indication lights up dark white when the button is not pressed. This way, you can see when it is flashing that the blinds are moving.

Proximity function

In the default setting, the proximity function is activated (Proximity function --> 21).

Night mode

You can use the *Night mode input* object to toggle between normal and night mode. If no proximity is detected, one of the status indications for the push-button is switched on during night mode for orientation (Normal and night mode --> 19).

Chapter LED modes --> 18 contains the general settings for the LEDs. For example, you can also display the states of the status indication with 2 colors, green and white.

Group objects

The blinds are moved via the *Movement object*. The blinds are stopped and adjusted via the *Stop/step object*. The status indication is controlled solely via button operation.

Group objects for "Blind" express setting

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Stop/step object	Function 1	1 bit	Sends, receives	1.007 step
2	Movement object	Function 1	1 bit	Sends, receives	1.008 up/ down

3.6 Scene

With *Express settings*, you can call up and save a scene for each button. To do so, select the *Scene* function.

A short button action calls up a scene. A long button action saves the current values for the scene. To aid operation, the status indication lights up when pressed briefly and flashes when pressed and held.

Express settings	Function x	Scene

Call up room functions   You can use a scene to change multiple room functions at the touch of a button. Loading a scene allows you, for example, to dim the room lighting to a specific value, move the blinds into the desired position and switch on the power supply to the socket-outlets in a room.

Save room functions   You can change the values for the individual room functions in a scene. To do this, use further button functions such as switching, dimming or move blind. You can use these button functions to change the values for the room functions consecutively. You then save the new values to the scene button by pressing and holding the button.

Setting the scene function   After you have selected the scene function, the parameter *Scene address (0-63)* appears. You can use this value to call up a scene in actuators and in scene modules. The values for saving (128 – 191) are assigned automatically.

Button functions	Function x	Scene
	Scene address (0-63)	

Status indication

The following section describes the behaviour of the button illumination in the default setting. In the express setting *Scene*, the status indication has 3 states.

State	Status indication
Short operation	Illuminates bright white
Long operation	Flashes bright white
No operation	Illuminates dark white

Activation on operation   The status indication lights up when pressed briefly and flashes when pressed and held. The status indication lights up dark white when the button is not pressed. This way, you can see when it is flashing that a scene is being saved.

Proximity function   In the default setting, the proximity function is activated (Proximity function --> 21).

Night mode   You can use the *Night mode input* object to toggle between normal and night mode. If no proximity is detected, one of the status indications for the push-button is switched on during night mode for orientation (Normal and night mode --> 19).

Chapter LED modes --> 18 contains the general settings for the LEDs. For example, you can also display the states of the status indication with 2 colors, green and white.

Group objects

The scenes are called up and saved via the *Scene object*. The status indication is controlled by button action.

Group objects for "Scene" express setting

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Scene object	Function 1	1 byte	Sends	18.001 scene control

3.7 Transition to extended settings

If you want to customise the configuration of a button, select the button function *Enable extended settings*.

The relevant function then appears under the tab *Extended settings*. You can select a button function in the parameters for the function.

Express settings	Function x	Enable extended settings
Extended settings		
Function x	Function x	Toggle
		Switching
		Dimming
		Blind
		Edges 1 bit, 2 bit (priority), 4 bit, 1 byte values
		Edges with 2 byte values
		8 bit slider
		Scene

Further information is available in chapter Extended settings --> 25.

4 General settings

General settings apply to all buttons. These settings apply both to the configuration with *Express settings* and with *Extended settings.*

General settings

Proximity function
Proximity sensor sensitivity
LED modes
Locking function for buttons
Temperature sensor

Proximity function

If you come within 5 cm of the push-button, the proximity function is triggered. When proximity is detected, the status indications switch on and then switch off again after the off delay elapses. Optionally, the proximity function can also be triggered by telegram.

Proximity sensor sensitivity

This setting is valid for the *Touchless function* and *Proximity function*. You can choose the level of sensitivity from recommended settings listed in ETS5.

LED operating modes

There is a status indication for each button. In the default settings, the *Status On* state is shown in bright white and the *Status Off* state is shown in dark white. You can adjust the color and brightness of each of these states. You can also make further settings for normal and night modes.

Locking function for buttons

You can lock each button individually. Alternatively, you can specify one button as the master button. All other buttons then have the same function as the master button.

Temperature sensor

You can enable a temperature sensor that measures the temperature in degrees of Celsius or Fahrenheit. Select type of object and set up internal actual temperature correction. There is also an option for setting an interval of sending (depending on time and temperature deviation).

4.1 LED modes

Status indication

There is a status indication for each button. In the default settings, the *Status On* state is shown in bright white and the *Status Off* state is shown in dark white. If the status indication is active, it lights up either dark white or bright white.

The following illustration shows a push-button with 4 button functions and an active status indication. All 4 buttons are lit up and show either the on status or the off status.
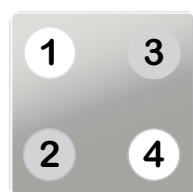
The LEDs can also be switched off. If the proximity function is activated and no proximity is detected, all LEDs are switched off. In night mode, the default setting is for only 1 LED to be switched on.

Colors

The colour settings apply to all status indications for the push-button. You can assign the following colors.

Status On	Status Off
White	Green
Green	White
White	White

LED modes	Colour of status indication

Normal and night mode

You can adjust the brightness of the status indication to suit the individual room use for normal and night modes. In order to ensure that the light of the status displays does not disturb, e.g. in the bedroom, the status displays are illuminated with reduced brightness in night mode.

This is toggled via the *Night mode input* object (e.g. with a KNX time switch or a KNX twilight sensor).

Brightness

First adjust the colour for the states of the status indication. The parameters with the brightness values change depending on the colour selection.

You can set brightness values for *Status On* and *Status Off*. If you have selected white and green, adjust the brightness values for the following parameters.

- Brightness of white LED in normal mode
- Brightness of green LED in normal mode
- Brightness of white LED in night mode
- Brightness of green LED in night mode

If you have selected white for both states, adjust the brightness values for the following parameters.

- Brightness of white LED with Status On in normal mode
- Brightness of white LED with Status Off in normal mode
- Brightness of white LED with Status On in night mode
- Brightness of white LED with Status Off in night mode

Status indications in night mode

Night mode   **Value:** *Only 1 status indication visible*

In night mode, only 1 LED lights up with the *Status On* colour. If you approach the button, all status indications are activated and show the currently applicable states. After the off delay elapses, only 1 status indication lights up. This is the default setting. If the proximity function is switched off, all status indications light up in night mode.

Value: *All status indications visible*

In night mode, the status indications behave in the same way as during normal operation. Only the brightness changes, according to the night mode settings. In the default setting, the brightness is reduced.

LED modes	Colour of status indication

> Brightness of white LED in normal mode
>
> ...
>
> Which LEDs are used in night mode?

Orientation light

If no proximity is detected, one of the status indicators is switched on and thus acts as an orientation light. As soon as the push-button detects proximity, all status indicators light up and respond as in normal operation

Group objects

Switch on night mode

Toggling between normal and night modes is carried out via the *Night mode input* object. Night mode is switched on with the value "1" and switched off with the value "0".

Group object for night mode

No.	Name	Object function	Length	Properties	DPT ETS4/5
16	Night mode input	Night mode LEDs	1 bit	Receives	1.001 switch

4.2   Locking function for buttons

You can use the locking function to prevent unauthorised or accidental triggering of room functions. Buttons are locked with a KNX time switch or another KNX push-button.

You can lock each button individually. Alternatively, you can specify one button as the master button. All other buttons then have the same function as the master button.

You can adjust the following parameters.

Locking function for buttons	Use locking function	Yes
	Execute lock at object value 1 or 0	
	Type of lock	
	Behaviour of status indication	
	Lock button 1 - 4 (Yes/No)	
	Which button is master?	

Type of lock

Value: *Set separately for each button*
With this *Type of lock* you can lock each of the 4 buttons individually. When a button is locked, it does not execute a function.

Value: *All buttons function like master button*
With this *Type of lock* you specify one button as a master button. If you press any button, the action that was set for the master button is executed.

Application example of using the master button:
During the normal mode, each button has its own function. In night mode, the buttons are locked and behave the same as the master button (e.g. switching the lights). You can only switch the lights with any of them and the other functions are not active.

Status indication

You can adjust the status indication behaviour when the locking function is activated.

Locking function for buttons	Behaviour of status indication on lock	Like in normal mode
		Flashes alternately for 5 s at button press
		Flashes alternately

Group objects

You activate the button lock via the *Locking object*. You can set whether the value "1" or the value "0" locks.

Group object for locking function

No.	Name	Object function	Length	Properties	DPT ETS4/5
13	Locking object	Button lock	1 bit	Sends, receives	1.003 enable

When a locking function is activated via the *Locking object*, all current button functions are reset or interrupted.

4.3   Proximity function

If you come within 5 cm of the push-button, the proximity function is triggered. When proximity is detected, the status indications switch on and then switch off again after the off delay elapses. There are additional options for night mode (). In the default setting, the proximity function is activated.

You can set the following functions.

Proximity function	How is the proximity function triggered?	
	Off delay status indication = base * factor	
	Base	
	Factor (3 - 255)	
	Type of output object	
	Value of output telegram (1 - 255)	
	Send output telegram cyclically?	Yes
	Cycle time = base x factor	
	Base	
	Factor (3 - 255)	

Triggering the proximity function

In the default setting, the proximity function is triggered by the built-in proximity sensor. The type of triggering can be adjusted.

Value: *Never*

The proximity function is deactivated. The status LED and orientation light are not affected. (Status indication remains on.)

Value: *Sensor*

The proximity function is triggered via the internal proximity sensor.

The **Proximity** and **No proximity** states control the status indication.

Value: *Proximity object*

The proximity function is triggered via the *Proximity input* object.

The proximity object has the same function as the internal proximity sensor.
- An On telegram activates the **Proximity** state.
- An Off telegram activates the **No proximity** state.

Application example with 1 presence detector

The illuminated status indications make it easier to find the push-button and its functions more quickly. You can use a presence detector to control the status indication via the proximity object. If a person is present, the status indication switches on. After the person leaves the room, the status indication switches off.

Value: *Sensor or proximity object*

The proximity function is triggered via the sensor or the *Proximity input* object.

The sensor and the proximity object are linked to each other. The result of the link corresponds to an OR link.

Proximity detected	Proximity object = 1	State
Yes	No	Proximity
No	Yes	Proximity
Yes	Yes	Proximity
No	No	No proximity

i   You can use the proximity object to lock and enable the proximity function. An On telegram locks the proximity function. The status indication is then not affected when proximity is detected. An Off telegram enables the proximity function via the sensor again.

Status indication

In the **Proximity** state, the status indication is switched on. The brightness depends on the settings for normal and night modes.

Operating mode	Proximity
Normal operation	All status indications active*
Night operation	All status indications active*

After the switch to the **No proximity** state, the off delay starts. After this time has elapsed, the status indications are switched off. In night mode, default setting with 1 status indication is activated. Alternatively, all status indications are also switched off during night mode (<u>Normal and night mode --> 19</u>).

Operating mode	No proximity
Normal operation	All status indications switched off
Night mode	1 status indication remains switched on (default)
	All status indications switched off

* The settings for status indications are described in chapter <u>Status indication --> 61</u>.

Off delay

You use the *Off delay* to set how long the status indications remain switched on after activation. The off delay can be set between 0,3 s and 255 min. The default setting is 10 s.

Proximity output

The **Proximity** and **No proximity** states control the *Proximity output* object.

The proximity output can be set as a 1 bit or 1 byte object.
- The 1 bit object sends the values "1" and "0".
- The 1 byte object sends an adjustable value (1–255) and the value "0".

The values can be sent cyclically. The cycle time is 0.3 s to 255 min. The default setting is 10 s. When the state changes from **Proximity** to **No proximity** and vice versa, the telegrams are sent immediately.

Group objects

You can select the following group objects.

Group objects for "Proximity function"

No.	Name	Object function	Length	Properties	DPT ETS4/5
14	Proximity input	Proximity function	1 bit	Receives	1.001 switch
15	Proximity output	Proximity function	1 bit	Sends	1.001 switch
15	Proximity output	Proximity function	1 byte	Sends	5.005 ratio (0...255)

4.4   Temperature sensor

The push-button has a built-in temperature sensor. If you enable temperature measurement, you can then set up the following parameters:

Temperature sensor	Enable temperature sensor?	Yes
	Temperature unit	
	Object type selection	
	Correct internal actual temperature = 0.1 K * factor (-120 - 120)	
	Send actual temperature at deviation (0-2 K)	
	Send actual temperature every (Never, 1-60 min)	

Temperature unit and object type

Select either *Celsius* or *Fahrenheit*.

If your unit is set to Fahrenheit you automatically get 2 byte object option and the menu *Object type selection* remains hidden. It only appears if you select Celsius unit.

Temperature sensor	Enable temperature sensor?	Yes
	Temperature unit	Celsius
	Object type selection	2 byte
		4 byte
		2 byte and 4 byte

Temperature correction

Adjust room temperature to achieve the required **thermal comfort**.

Measured values depend on where the push-button is mounted in the room. The temperature measurement is different when exposed to a draught or close to heat sources than at other places in the room.

With *Correct internal actual temperature* you can set an offset to the actual measured temperature form -12 K to +12 K. The following applies: Actual temperature = measured temperature + correction value.

Interval for sending values

You can set the **deviation value** at which you want your sensor to send telegram (0 - 2K). You can also set the telegram sending **interval** (0 - 60 min).

Group objects

You can select the following group objects.

Group objects for "Temperature sensor"

No.	Name	Object function	Length	Properties	DPT ETS4/5
18	Actual temperature Fahrenheit	Temperature sensor	2 byte	Sends	9.027 temperature (°F)
18	Actual temperature Celsius 2B	Temperature sensor	2 byte	Sends	9.001 temperature (°C)
19	Actual temperature Celsius 4B	Temperature sensor	4 byte	Sends	14.068 temperature (°C)

5 Extended settings

If you want to customise the configuration of a button, select the button function *Enable extended settings*. You can then choose one of 8 functions and configure it individually.

Functions

The *Toggle*, *Switching*, *Dimming* and *Blind* functions have additional settings for the corresponding room function. Dimmable lights are switched and dimmed using 2 brightness values, for example. Blinds are controlled with *positioning values*, for example.

The *Edges 1 bit, 2 bit (priority), 4 bit, 1 byte values* function can be used every-where. It includes the functions *Switching*, *Priority control*, *Dimming* and *Send values*. You can select 2 functions for each. You can also set the actions for short and long operation of the button. Cyclical and delayed sending is also possible. Examples of individual solutions can be found in section .

The *Edges with 2 byte values* function sends values of the floating point or integer type. For example, brightness values are sent to control the lighting.

The *8 bit slider* function is used to send a series of values. The values are in-creased or decreased in steps. For example, multiple scenes are called up in succession with 1 button.

The *Scene* function offers additional settings for simultaneous control of multiple room functions. For example, with 1 button a scene is called up immediately and a second scene is called up after a delay.

Selection

If you want to customise the configuration of a button, select the button function *Enable extended settings*.

Express settings	Function x	Enable extended settings
Extended settings		
Function x	Function x	Toggle
		Switching
		Dimming
		Blind
		Edges 1 bit, 2 bit (priority), 4 bit, 1 byte values
		Edges with 2 bit values
		8 bit slider
		Scene

Extended settings are not available with enabled *Touchless function*.

Overview of button functions

- Toggle
  – 2 objects
  – 1 bit, 1 byte
- Switching
  – 2 objects
  – 1 bit, 1 byte
- Dimming

- – Step dimming
- Blind
  - – Control with positioning values
  - – Single-button operation
- Edges 1 bit, 2 bit (priority), 4 bit, 1 byte values
  - – 2 objects
  - – Functions for short and long button actions
  - – Cyclical sending and staircase lighting function
  - – Priority control (2 bit)
  - – Dimming (4 bit)
  - – Dimming with values (1 byte)
  - – Values (1 byte)
- Edges with 2 byte values
  - – Floating point value
  - – Integer with sign
  - – Integer without sign
- 8 bit slider
  - – With limit values
  - – Send cyclically
  - – Increase or reduce stepwise
- Scene
  - – Calling up and saving
  - – Calling up 2 scenes
  - – Cyclical and sending and staircase lighting function

The settings for status indications are described in chapter Status indication --> 61.

Toggling

With the *Toggle* function, you switch on and off alternately with 1 button. This involves single-button operation.

You can change and extend the *Toggle* with the following functions.
- Send simultaneously with 2 objects
- Switch on and off and send values
- Trigger status indication

Extended settings		
Function x	Function x	Toggle
	↻	
	Number of objects	
	Type of object A/B	
	How is the status indication triggered?	
	LED behaviour	

1 bit, 1 byte button functions

Number of objects   You can select 2 objects. You can specify the object type of each object.

Send On and Off   With the 1 bit object type, you can switch on and off alternately with each button action. The current object value is inverted and then sent to the bus. The values "1" and "0" are sent alternately.

Send values   With the 1 byte object type, you can send 2 values alternately with each button action. You can select the values to be absolute (0 – 255) or a percentage.

Status indication

In the default setting, the status indication is triggered by the *Switch/Value object*.

Object value	Status indication
On (> 0)	Illuminates bright white
Off (= 0)	Illuminates dark white

You can find comprehensive settings in chapter Status indication --> 61.

Colors **and brightness values**

You can find the settings for colors and brightness in chapter LED modes --> 18.

Application example with on/off and brightness values

In a corridor, a lighting strip is to remain switched on at a low brightness after the lights are switched off, to aid orientation. 2 objects are set for this function.

	Main lighting Switch object A	Orientation lighting Value object B
Switching On	On	Value 1 = 80 %
Switching Off	Off	Value 2 = 20 %

Group objects

You can select the following group objects.

Group objects for "Toggle" function

No.	Name	Object function	Length	Properties	DPT ETS4/5
1/2	Switch object A/B	Function x	1 bit	Sends, receives	1.001 switch
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.001 percentage (0..100 %)
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.005 ratio (0-255)
3	Status feedback object	Function x	1 bit	Receives	1.001 switch
3	Status feedback object value	Function x	1 byte	Receives	5.004 percent (0–255%)
17	Collected status feedback object	Feedback	4 byte	Receives	27.001 bit-combined info on/off

Single-button operation

In the case of a two-way or central circuit, the switching channel is also switched via another sensor. With single-button operation, it is possible to adjust to the current state of the switching channel. To do this, the state of the actuator has to be loaded to the push-button. For *Switch objects* (1 bit), this involves the values "1" and "0".

- To do this, also connect the group address of the status feedback object for the connected channel to the *Switch object* (1 bit) for the button function.

Loading is also possible for values. Adjust the same values for all push-buttons, e.g. 70 % for value 1 and 0 % for value 2. This way, you can switch on a dimmer at a selected brightness from multiple locations, and switch it off again. If the last sent or received value is the same as value 1, value 2 is sent the next time that the button is pressed, otherwise value 1 is sent.

- For a two-way circuit, check that the 1 byte objects for the 2 push-buttons are linked to the same group address.
- In the case of a central push-button, also connect the central group address of the 1 byte object to the 1 byte objects of the other push-buttons.

5.1 Switching

With the *Switching* function, you switch only either on or off with 1 button. This is a two-button operation.

You can change and extend the *Switching* with the following functions.

- Send simultaneously with 2 objects
- Switch on and send a value
- Switch off and send a value
- Send 2 values
- Trigger status indication

Extended settings		
Function x	Function x	Switching
	⌇	
	Number of objects	
	Type of object A/B	
	How is the status indication triggered?	
	LED behaviour	

1 bit, 1 byte button functions

Number of objects   You can select 2 objects. You can specify the object type of each object.

Send On and Off   The 1 bit object type is used for normal switching.

Send values   You can use the 1 byte object type to send a value. You can select the value to be absolute (0 – 255) or a percentage.

Status indication

In the default setting, the status indication is triggered by the *Switch/Value object*.

Object value	Status indication
On (> 0)	Illuminates bright white
Off (= 0)	Illuminates dark white

You can find comprehensive settings in chapter .

Colors and brightness values

You can find the settings for colors and brightness in chapter .

Group objects

You can select the following group objects.

Group objects for "Switching" function

No.	Name	Object function	Length	Properties	DPT ETS4/5
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.001 percentage (0..100 %)
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.005 ratio (0-255)
3	Status feedback object	Function x	1 bit	Receives	1.001 switch
3	Status feedback object value	Function x	1 byte	Receives	5.004 percent (0–255%)
17	Collected status feedback object	Feedback	4 byte	Receives	27.001 bit-combined info on/off

5.2   Dimming

With *Dimming*, you can switch and dim the dimmable lighting with 1 or 2 buttons.

Two-button dimming is the default setting. A short button action switches on or off. A long button action dims brighter or darker. Releasing after pushing and holding the button ends the dimming process.

You can change and extend the dimming function with the following functions:
*   Dim alternately brighter or darker with each button (single-button operation)
*   Only dim brighter or only darker with each button (two-button operation)
*   Operating time for long button action
*   Dim in steps with multiple dimming commands
*   Send dimming command cyclically
*   Trigger status indication and LED behaviour

Single-button dimming

Extended settings		
Function x	Function x	Dimming

> ↫
>
> Long operation = 100 ms *
> factor (4-250)
>
> Dimming direction
>
> Step dimming (brighter)
>
> Step dimming (darker)
>
> Send dimming command
> cyclically
>
> How is the status indication
> triggered?
>
> LED behaviour

In the default setting, a single command is sufficient to run through the dimming range. Hold the button down until the required level of brightness has been reached. When you release the button, the *Dimming object* sends a stop telegram and ends the dimming process.

Step dimming   If desired, you can divide the dimming process into dimming steps (1/2 – 1/64 brighter or darker). If you select 1/4 brighter, you can dim a maximum of 25 % brighter with each button action. The dimming process also ends in this case when you release the button.

Send cyclically   You can send dimming commands cyclically. The dimming process ends when you release the button.

ⓘ   In the case of a two-way or central circuit, the dimmer is also switched and dimmed via another sensor. With single-button operation, it is possible to adjust to the current state of the dimming channel. To do this, the state of the dimming channel has to be loaded to the push-button.

- To load the switching command function, connect the group address of the 1 bit *Status feedback object* for the connected channel to the *Switch object* for the button function.
- For a two-way circuit, check that the *Dimming objects* for the 2 push-buttons are linked to the same group address.
- In the case of a central circuit, also connect the central group address of the *Dimming object* for the central push-button to the *Dimming objects* of the other push-buttons.

Two-button dimming

The settings for single-button and two-button operation are largely similar. For dimming with dimming steps, you can use the parameter *Send stop telegram after release*. In the default setting, as with single-button operation, a stop telegram ends the dimming process when the button is released. If, however, no stop telegram is sent, the dimming process continues even after the button is released. A long button action is then enough to dim a step brighter or darker. If you select the dimming step *1/4 brighter*, you can dim from minimum to maximum brightness with 4 long button actions.

Send stop telegram

Status indication

In the default setting, the status indication is triggered by the *Switch object*.

Object value	Status indication
On	Illuminates bright white
Off	Illuminates dark white

You can find comprehensive settings in chapter .

Colors and brightness values

You can find the settings for colors and brightness in chapter LED modes --> 18.

Group objects

You can select the following group objects.

Group objects for "Dimming" function

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Switch object	Function x	1 bit	Sends, receives	1.001 switch
2	Dimming object	Function x	4 bit	Sends, receives	3.007 dimmer control
3	Status feedback object	Function x	1 bit	Receives	1.001 switch
3	Status feedback object value	Function x	1 byte	Receives	5.004 percentage (0–100%)
17	Collected status feedback object	Feedback	4 byte	Receives	27.001 bit-combined info on/off

5.3   Blind

With the *Blind* function, you can raise and lower a blind and adjust the slats.

In the default setting, you raise a blind and adjust the slats. To lower the blind, you need a second button function. You can choose from the following operation concepts.

- Raise and lower the blind alternately and adjust the slats with each button (single-button blind operation).
- Only raise or only lower the blind and adjust the slats with each button (two-button blind operation).
- Move the blind to a previously specified position and adjust the slats.
- Move the blind back and forth between 2 previously specified positions and adjust the blinds.

Two-button operation of blind

You can now move the blind either up or down by pressing and holding the corresponding button. A short button action stops the travel. A short button action also adjusts the slats in steps. You can set the operating time for the long button action.

Extended settings		
Function x	Function x	Blind
	⟲	
	Movement direction of blind	Moving blind up
	Long operation = 100 ms * factor (4-250)	

The blind is moved up or down via the *Movement object*, while the stopping and adjusting of the slats is controlled via the *Stop/step object*. You need 2 button functions, each of which you connect to the same group addresses.

Single-button operation of blind

A long button action alternately raises or lowers the blind. The current movement direction of the blind is always dependent on the previous action. You can set the operating time for the long button action.

Pause on change in direction

You can adjust the slats in the same direction in multiple steps. To do so, briefly press the button repeatedly until the desired position is reached. The slats adjust in the same direction provided that the next button action follows within an adjustable pause time. Once this *pause* elapses, the slat direction of rotation changes.

Extended settings		
Function x	Function x	Blind
	⌇	
	Movement direction of blind	Moving blind up/down
	Long operation = 100 ms * factor (4-250)	
	Pause for change slat direction = 100 ms * factor (5-50)	

The blind is moved alternately up and down via the *Movement object*, while the stopping and adjusting of the slats is controlled via the *Stop/step object*.

In the case of a two-way or central circuit, the blind is also controlled via another sensor. With single-button operation, it is possible to adjust the change of direction. To do this, the state of the other *Movement* and *Stop/step objects* has to be loaded.

- For a two-way circuit, check that the *Movement objects* for the 2 push-buttons are linked to the same group address. The same applies to the *Stop/step objects*.
- In the case of a central circuit, also connect the central group addresses of the *Movement object* for the central push-button to the movement objects of the other push-buttons. The same applies to the *Stop/step objects*.

Moving blind with positioning values

If the blind actuator supports activation of positions, you can use this function to set 1 or 2 positions. You can select the positioning values to be absolute (0 – 255) or a percentage.

Blind position and slat position

If positioning is activated, when you press the button, the set values for the blind position and slat position are sent.

Number of positionings

If you have set 1 position, the values for the blind and slats are sent when you press the button briefly. If you have set 2 positions, you specify a total of 4 values. You send the values for position 1 after a brief button action and the values for position 2 after a long button action.

Extended settings		
Function x	Function x	Blind
	⌇	
	Direction of movement of the blind	Moving blind with positioning values
	Number of positionings	
	Scaling (short operation)	
	Position 1 of blind	
	Position 1 of slats	

You send the values for the positions via the 1 byte objects *Position of blind* and *Position of slat*. The *Movement object* and *Stop/step object* are not available if you move the blinds with positioning values.

Status indication

In the default setting, the status indication is triggered by pressing the button.

State	Status indication
Operation	Switched on (bright white)
No operation	Switched on (dark white)

If the blind actuator returns feedback for the current position of the blind, it is possible to display whether the blind is completely open. The status feedback can be evaluated via the *status feedback value* for the button function.

You can display whether blind movement is locked. Blind actuators can raise the blinds in the event of strong winds and lock them against being lowered again. This safety function is generally triggered via an On telegram.

① Select the value *Status feedback object equal to 0 = Flashes*.

② Connect the group address of the safety function to the *Status feedback object* of the push-button.

You can find comprehensive settings in chapter Status indication --> 61.

Colors and brightness values

You can find the settings for colors and brightness in chapter LED modes --> 18.

Group objects

You can select the following group objects:

Group objects for "Blind" function

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Stop/step object	Function 1	1 bit	Sends, receives	1.007 step
2	Movement object	Function 1	1 bit	Sends, receives	1.008 up/down
1	Blind position	Function 1	1 byte	Sends	5.005 ratio (0-255)
2	Slat position	Function 1	1 byte	Sends	5.005 ratio (0-255)

5.4   Edges 1 bit, 2 bit (priority), 4 bit, 1 byte

The *edge function* for 1 bit, 2 bit, 4 bit and 1 byte has a wide range of adjustment options for a variety of individual applications. In the default setting, you switch on 1 consumer when you operate the button and switch it off when you release the button.

You can choose between the normal and extended edge functions and set the following functions.

* Send with 2 objects
* Type per object: 1 bit, 2 bit (priority control commands), 4 bit (dimming commands), 1 byte (0–100 %) or (0–255)
* Values per object
* Action on operation and on release
* Additionally, actions for long and short operation (extended edge function)
* Additionally, send cyclically and with delay (extended edge function)
* Trigger status indication

A list of applications with the edge function is provided in chapter Application examples for edge function --> 42.

	Extended settings	
	Function x	
	Function x	Edges 1 bit, 2 bit (priority), 4 bit, 1 byte values
	Select edge function	Normal (operate, release)
		Extended (+ long and short operation)
	Long operation = 100 ms * factor (4-250)	[Extended edge function]
	Number of objects	
	How is the status indication triggered?	
	LED behaviour	

Normal edge function

With the normal edge function, you can specify which actions should be carried out when a button is pressed, and which should be carried out when a button is released.

You can set 2 objects independently of each other.

- 1 bit
- 2 bit (priority control commands)
- 4 bit (dimming commands)
- 1 byte (0-100 %)
- 1 byte (0-255)

You can select 2 values for each object.

Object type	Selection 1 [value]	Selection 2 [value]
1 bit	1	0
> 1 bit	Value 1	Value 2

Actions for "Normal edge function"

You can set the following actions.

Value: *Sends [value]*
Sends the value in question once and stops cyclical sending.

Value: *Sends its value*
The current object value is sent. Therefore you can, for example, send a value with the sending group address that was previously received via another group address. In so doing, you save a value in the push-button and this value is sent when needed.

Value: *Toggles*
With a 1 bit object, the inverted object value is always sent. If the object last sent or received an On telegram, an Off telegram is sent next. Accordingly, after an Off telegram, an On telegram is sent.

For the other object types, either value 1 or value 2 is sent. Values that have not been set can also be received via the bus. If the object last sent or received value 1, value 2 is sent next; otherwise, value 1 is sent.

In the case of a two-way or central circuit, set the same values for all sending objects, e.g. 70 % for value 1 and 0 % for value 2.

Value: *None*
No action is carried out

Parameters per object

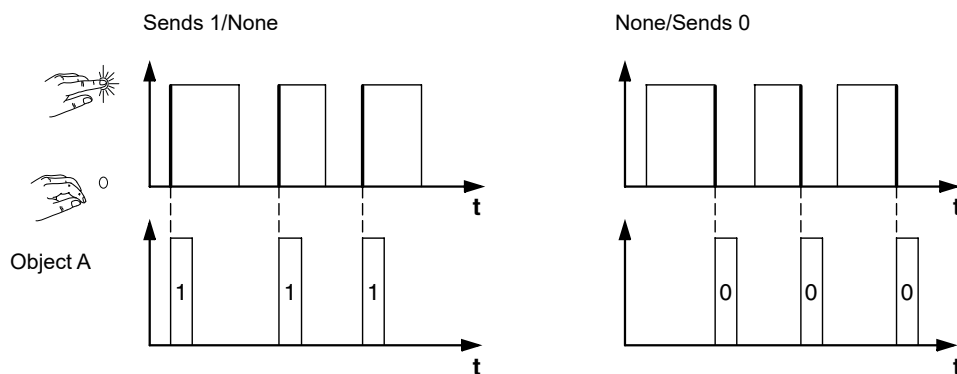There is an additional tab under the Function tab for each group object.

Extended settings	
Function x	
Function x - edges object A	Type of object A
	Action on operation
	Action at release

Principle of the normal edge function

The following diagrams show how the edge function behaves when a button is pressed and released.

🖐✳	Operate button
🖐°	Release button
Object A	Sends telegrams
Sends 1/ None	Action on operation/action on release

Example 1   Object A = 1 bit

Sends 1/None   None/Sends 0

Object A

Example 2   Object A = 1 bit

Sends 1/Sends 0   Sends 0/Sends 1

Object A

Example 3   Object A = 1 bit

Toggles/None

None/Toggles
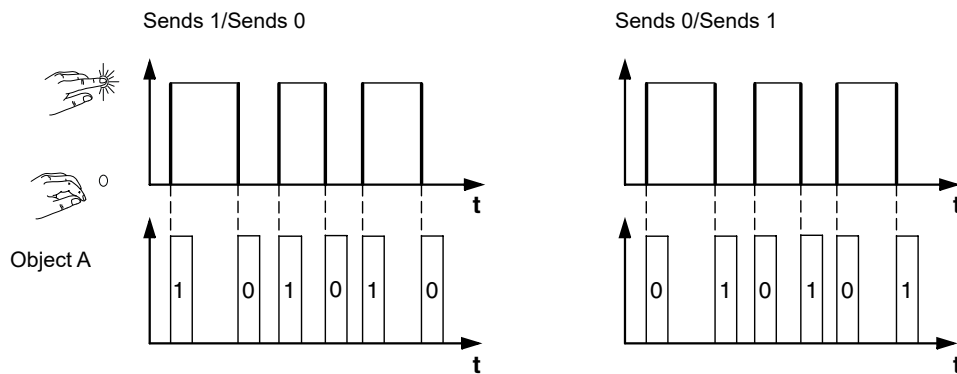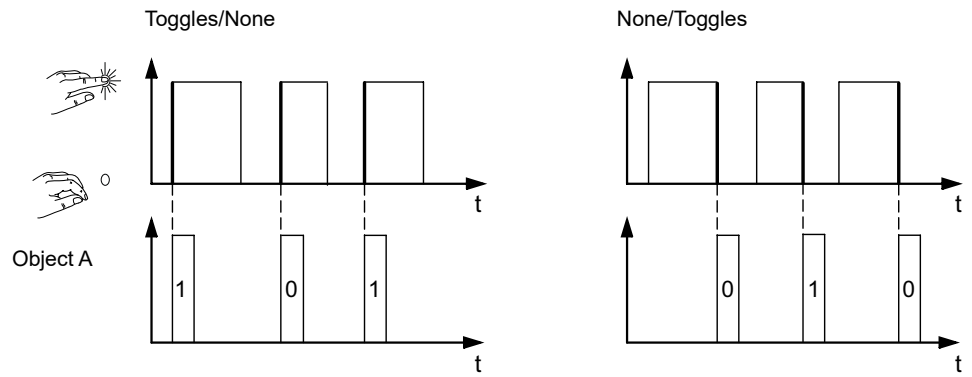
Object A

Example 4   Object A = 1 byte continuously 0–255
Value 1 = 255
Value 2 = 50

Sends value 1/Sends value 2

Toggles/None

Object A

Example 5
Deadman control
Toggle priority control
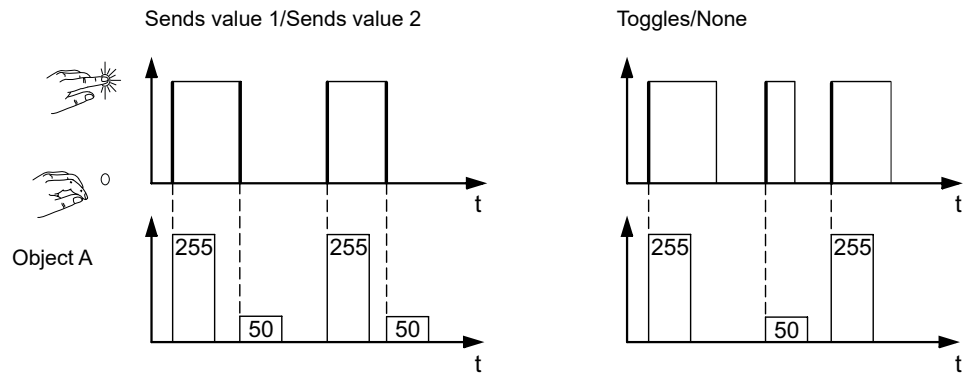
Object A = 2 bit (priority control)
Value 1 = 11 (switch on with priority)
Value 2 = 10 (switch off with priority)

Sends value 1/Sends value 2

Toggles/None

Object A
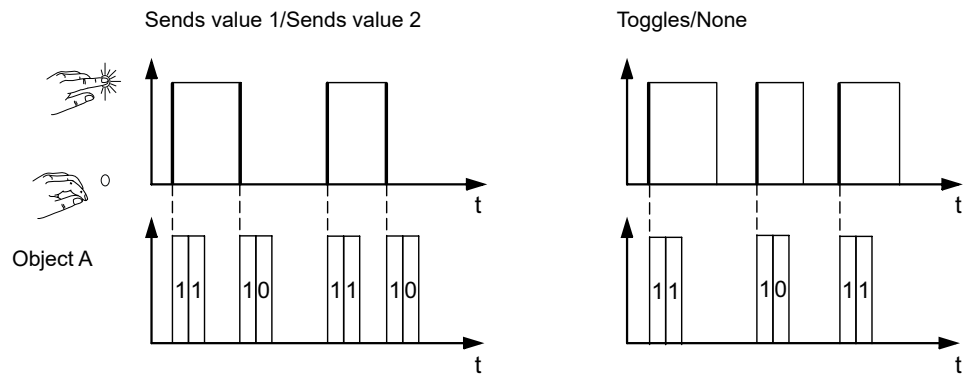
Deadman control

Toggle with priority

Extended edge function

With the extended edge function, an even wider range of functions is available. In addition to the normal edge function, the extended function differentiates between short and long operation. In total, you set 4 actions for operation and release.

① **Direct action on operation**
   – The action is executed each time the button is operated.
② **Action on release before the long operating time has elapsed**
   – The action is only executed after a short operation.
③ **Action on achieving the long operating time**
   – The action is executed directly when the button is pressed and held. You press the button until the action (e.g. switch light) is carried out.
④ **Action on release after achieving the long operating time**
   – The action is also performed on release after pressing and holding.
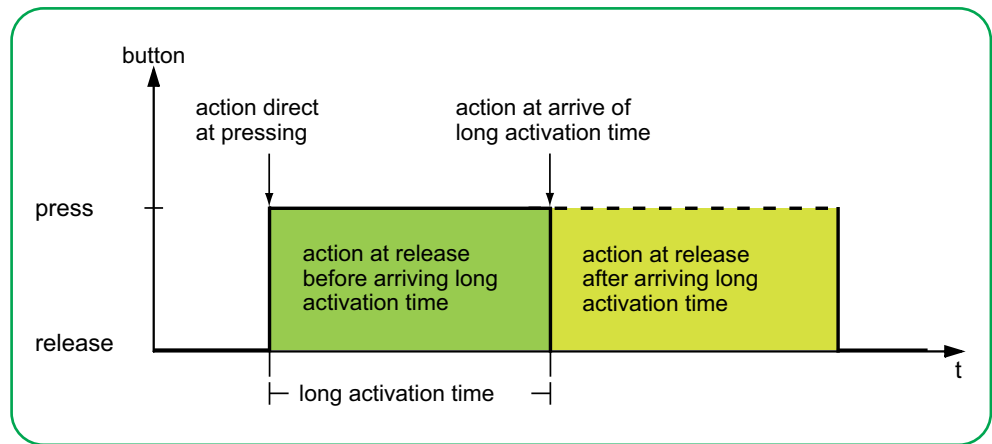
You set the relevant action for each operation phase. In addition to the normal edge function, you can set a cycle time for each object. You can send once or cyclically. For example, you can use the action *Toggles cyclically, sends immediately, then cyclically* to toggle cyclically between lighting scenes. You can use the action *Sends value 1 and then value 2 after a cycle time* to execute a staircase lighting function. Examples with the extended edge function are provided in chapter Application examples for edge function --> 42.

ℹ
- When setting the parameters, remember that you have to set all 4 operating phases for the push-button to function as required.
- In order to read the object values, you have to set the **R**ead flags manually.

You can set 2 objects independently of each other.
- 1 bit
- 2 bit (priority control commands)
- 4 bit (dimming commands)
- 1 byte (0-100 %)
- 1 byte (0-255)

You can select 2 values for each object.

Object type	Selection 1 [value]	Selection 2 [value]
1 bit	1	0
> 1 bit	Value 1	Value 2

Actions for "Extended edge function"

You can set the following actions.

Value: Sends [value]

Sends the value in question.

Value: *Sends [value] immediately and then cyclically*

If no cycle time is running, the value is sent immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, the value is sent and a new cycle time is started. The value then continues to be sent cyclically. You can use this function to dim in steps, for example (e.g. 1/8 brighter).

Value: *Sends [value] only cyclically*

If no cycle time is running, the value is sent immediately and a new cycle time is started. If a cycle time is already running, it is not interrupted; the value is sent after the current cycle time has elapsed, and a new cycle time is started. The value then continues to be sent cyclically. You can use this function to monitor the push-button function, for example.

Value: *Sets object value to [value] (readable only)*

The value is written to the object and is not sent. Any active cycle time is terminated. If you want the value to be read by a visualisation, for example, you have to set the **R**ead flag for the object.

Value: *Toggles*

Sends the set values alternately. The toggling is also controlled via the bus.

With a 1 bit object, the inverted object value is always sent. If the object last sent or received an On telegram, an Off telegram is sent next. Accordingly, after an Off telegram, an On telegram is sent.

For the other object types, either value 1 or value 2 is sent. Values that have not been set can also be received via the bus. If the object last sent or received value 1, value 2 is sent next for the next action; otherwise, value 1 is sent.

In the case of a two-way or central circuit, set the same values for all sending objects, e.g. 70 % for value 1 and 0 % for value 2.

Value: *Toggles, sends immediately, then cyclically*

If no cycle time is running, the value is toggled once, sent immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, the value toggled once is sent and a new cycle time is started. Then, the value continues to be sent cyclically, but without further toggling (see *Toggles*).

Value: *Toggles, only sends cyclically*

If no cycle time is running, the value is toggled once, sent immediately and a new cycle time is started. If a cycle time is already running, this is not interrupted, the value toggled once is sent after the current cycle time has elapsed, and a new cycle time is started. Then, the value continues to be sent cyclically, but without further toggling (see *Toggles*).

Value: *Toggles and is not sent*

The toggled value is written to the object and is not sent. Any active cycle time is terminated (see *Toggles*). If you want the value to be read by a visualisation, for example, you have to set the **R**ead flag for the object.

Value: *Toggles cyclically, sends immediately, then cyclically*

If no cycle time is running, the value is toggled, sent immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, the toggled value is sent and a new cycle time is started. Then, the value continues to be toggled and sent cyclically (see *Toggles*).

You can use this function to switch cyclically between lighting scenes, for example.

Value: *Toggles cyclically, only sends cyclically*

If no cycle time is running, the toggled value is sent immediately and a new cycle time is started. If a cycle time is already running, it is not interrupted; the toggled value is sent after the current cycle time has elapsed, and a new cycle time is started. Then, the value continues to be toggled and sent cyclically (see *Toggles*).

Value: *Toggles cyclically and is not sent*

The toggled value is written to the object and is not sent. Subsequently, the value is always toggled cyclically and the new value is written to the object (see *Toggles*). If you want the value to be read by a visualisation, for example, you have to set the **R**ead flag for the object.

Value: *Sends its value* (not for 2 bit priority control)

The current object value is sent. Any active cycle time is terminated. Therefore you can, for example, send a value with the sending group address that was previously received via another group address. In so doing, you save a value in the push-button and this value is sent when needed.

Value: *Sends its value immediately and then cyclically* (not for 2 bit priority control)

If no cycle time is running, the current object value is sent immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, the current object value is sent and a new cycle time is started. Subsequently, the current object value continues to be sent cyclically.

Value: *Cyclically increase current object value by value 1* (for 1 byte only)

If no cycle time is running, *value 1* is added to the current object value, the object value is sent, and a new cycle time is started. If a cycle time is already running, it is not interrupted; the current object value with *value 1* added is sent and a new cycle time is started. You can use this function to increase and send multiple values in succession by holding down the button, for example. When the value "255" is exceeded, the value is reset to 0.

Example
- Current object value = 255
- Value 1 = 5
- Next value sent = 4

If, for example, you increase the current object value of "255" by the value "5", the value "4" is sent.

If you want to always send the same values, select in range 0 - 255 or 0% - 100%. However, the bus can overwrite the values and move them. The values are always increased from the current object value.

If you want to increase the values only in 1 direction and only up to a maximum value, select the function *8 bit slider* with limit values.

Value: *Cyclically reduce current object value by value 2* (for 1 byte only)

If no cycle time is running, *value 2* is subtracted from the current object value, the object value is sent, and a new cycle time is started. If a cycle time is already running, it is not interrupted; the current object value with *value 2* subtracted is sent and a new cycle time is started. You can use this function to subtract and send multiple values in succession by holding down the button, for example. If the value falls below "0", the value is reset to 255.
- Current object value = 0
- Value 1 = 5
- Next value sent = 250

If you want to always send the same values, select in range 0 - 255 or 0% - 100%. However, the bus can overwrite the values and move them. The values are always subtracted from the current object value.

If you want to reduce the values only in 1 direction and only down to a minimum value, select the function *8 bit slider* with limit values.

Value: *Sends [value A] and then [value B] after a cycle time*
Value 1 is sent immediately, and value 2 is sent after a cycle time, regardless of whether a cycle time is already running or not. For a 1 bit object, "1" is sent immediately and, after a cycle time elapses, "0" is sent. With this staircase lighting function, you can call up the comfort mode for a room temperature control unit, for example, and switch back to standby mode after the cycle time elapses. You set the duration via the cycle time.

Value: *None (stops cyclical sending)*
No action is carried out, and any active cycle time is stopped. Select this function if you also want to stop cyclical sending.

Value: *No change*
The current action is retained and any active cyclical sending is retained. You select this action for the release if, for example, you have activated the action *Sends value 1 and then value 2 after a cycle time*.

Value: *None (stop after current cycle time)*
No action is currently carried out, but any active cycle time is not stopped. It runs through until the end and then the corresponding value is sent once.

Parameters per object   There is an additional tab under the Function tab for each group object.

Extended settings	
Function x	
Function x - edges object A	Type of object A
	Direct action on operation
	Action on release before the long operating time has elapsed
	Action on achieving the long operating time
	Action on release after achieving the long operating time
	Value 1
	Value 2
	Cycle time = basis * factor

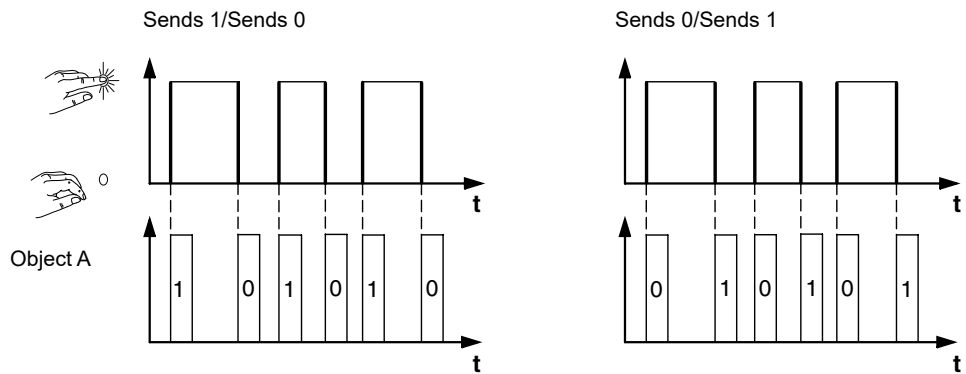# Application examples for edge function

The following activation sequence chart shows the phases into which the edge function is divided.

"Extended edge function" sequence chart
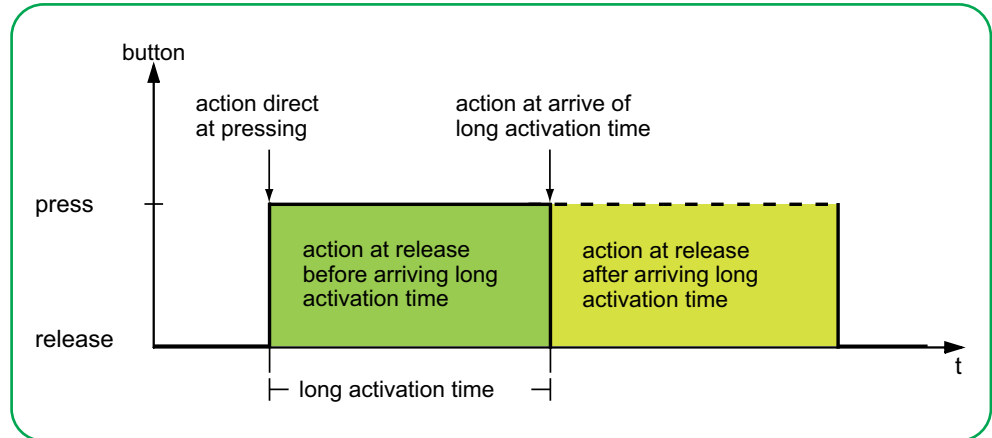
"Normal edge function" sequence chart

Object A = 1 bit

|	
🖐✴	Operate button
🖐○	Release button
Object A	Sends telegrams
Sends 1/ Sends 0	Action on operation/action on release

Staircase lighting function with cleaning light function

With a brief button action, the switch actuator switches on the light. A long but-
ton action extends the staircase lighting function (= cleaning light function) until a
second, long button action switches off the actuator. The switch actuator requires a
staircase lighting function and a locking function for this function.

- Number of objects = 2 (object A/B)
- Object A/B = 1 bit
- Object A:
  – *Action on release before the long operating time has elapsed* = *Sends 1*
  – Other actions = *None (stops cyclical sending)*
- Object B:
  – *Action on achieving the long operating time* = *Toggles.*
  – Other actions = *None (stops cyclical sending)*

To do this, connect object A to the *Switch object* and object B to the *Locking object*
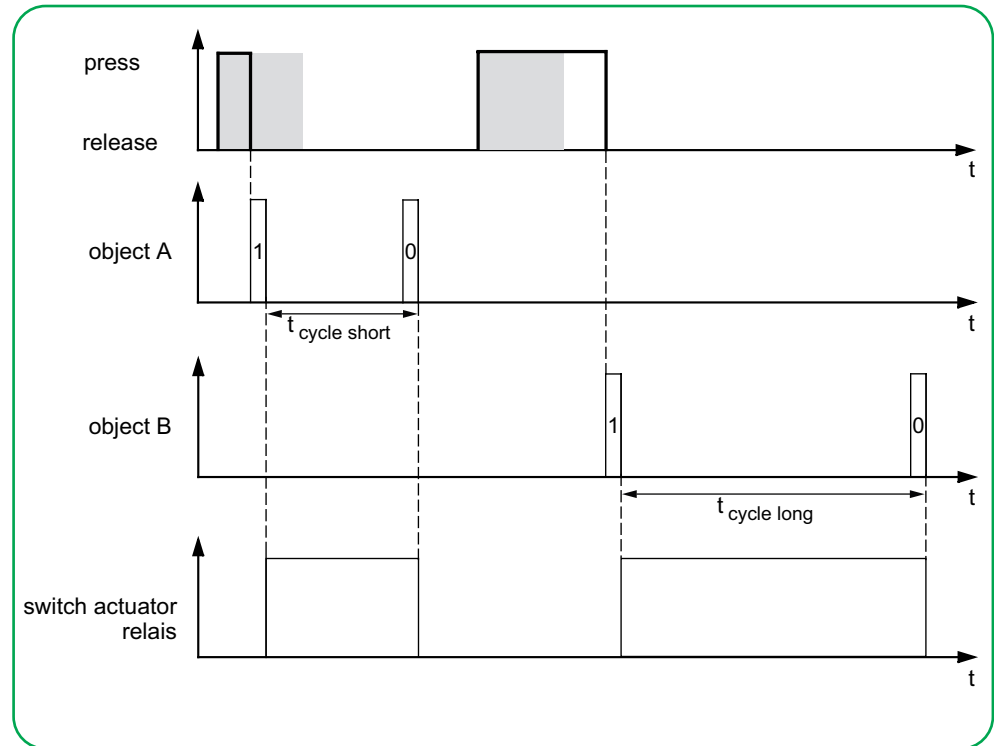of the switch actuator.
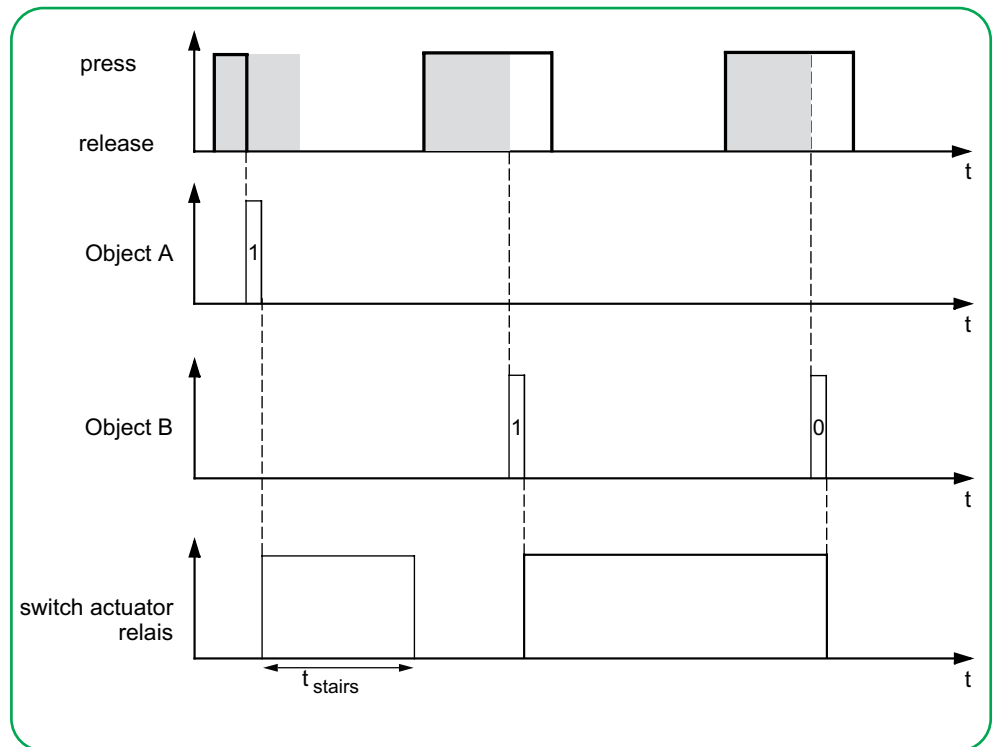
Short and long staircase lighting time

You can use this function to implement a short and a long staircase lighting time with the push-button. The switch actuator does not need a staircase lighting function for this requirement.

With a brief button action, the switch actuator switches on the light, and after a set cycle time (e.g. 3 minutes), it switches it back off again. With a long button action, the same function is carried out, but with a longer cycle time (e.g. 6 minutes).

- Number of objects = 2 (object A/B)
- Object A/B = 1 bit
- Object A:
  - *Action on release before the long operating time has elapsed* = *Sends 1 and then 0 after a cycle time* (3 min)
  - Other actions = *No change*
- Object B:
  - *Action on release after achieving the long operating time* = *Sends 1 and then 0 after a cycle time* (6 min)
  - Other actions = *No change*

To do this, connect object A and object B to the *Switch object* of the switch actuator.

Switching the light on/off permanently, or switching off after a cycle time has elapsed

With a brief button action, the switch actuator switches the light permanently on or off. A long button action switches the light on, and after a set cycle time (e.g. 6 minutes), it is switched back off again. Due to the adjustable cycle time in the push-button, the switch actuator for this function does not require a staircase lighting function.

- Number of objects = 2 (object A/B)
- Object A/B = 1 bit
- Object A:
  - *Action on release before the long operating time has elapsed* = *Toggles*

      – Other actions = *None (stops cyclical sending)*
- Object B:
  - *Action on achieving the long operating time = Sends 1 and then 0 after a cycle time* (6 min)
  - Other actions = *No change*

To do this, connect object A and object B to the *Switch object* of the switch actuator.

Electronic protection against theft

This example will show you how to program electronic theft protection for the push-button. It is activated by a brief button action and then sends cyclically. As soon as the push-button is separated, this can be shown on a display.

- Number of objects = 1 (object A)
- Object A = 1 bit
- Object A:
  - *Action on release before the long operating time has elapsed = Sends 1 immediately and then cyclically* (10 min)
  - Other actions: *No change*

Connect object A to an object that listens cyclically for telegrams (e.g. a *Safety object*). The monitoring time set on the *Safety object* must be longer than the cycle time of the push-button. If the *Safety object* receives no telegrams from the push-button during this time, an adjustable reaction is activated (e.g. channel is switched on).
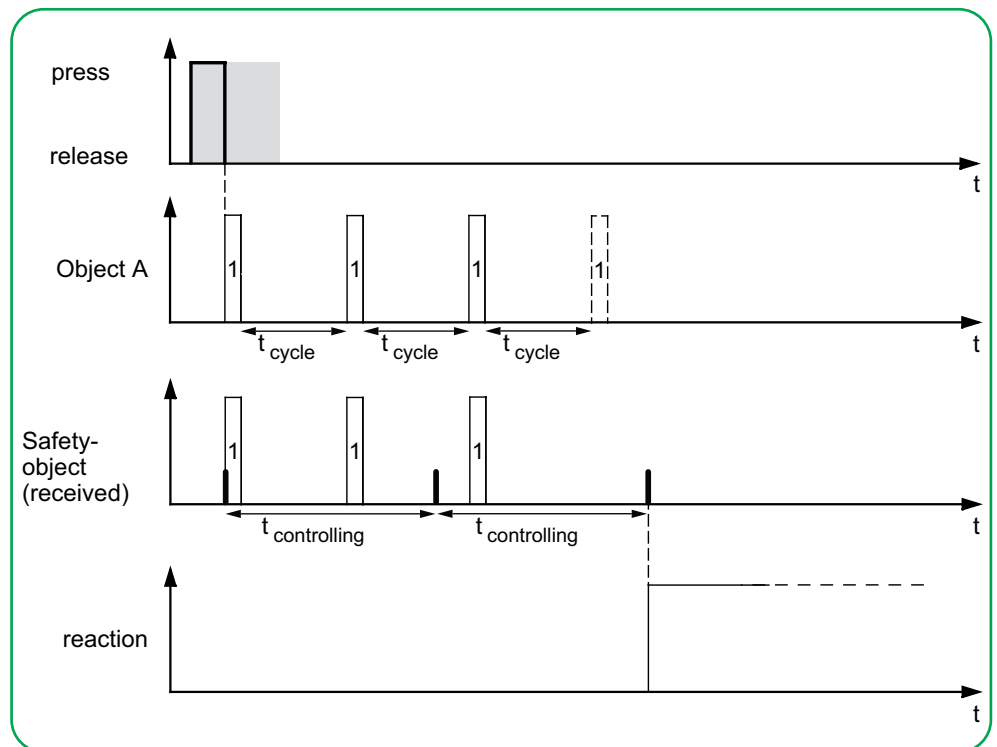
Effect lighting

This example shows you how to program effect lighting, for example for a display window. A long button action switches between 2 different lighting scenes. A short button action stops the toggling and sends a scene address that switches everything off. The scene module of the actuator that was activated is used to retrieve the scene.

- Number of objects = 2 (object A/B)
- Object A/B = 1 byte continuously 0–255
- Object A (*Value 1* = 3):
  – *Action on release before the long operating time has elapsed* = Sends value 1
  – Other actions = *None (stops cyclical sending)*
- Object B (*Value 1* = 1, *Value 2* = 2):
  – *Action on release after achieving the long operating time* = *Toggles cyclically, sends immediately, then cyclically* (1 min)
  – Other actions = *None (stops cyclical sending)*

To do this, connect object A and object B to a scene module.
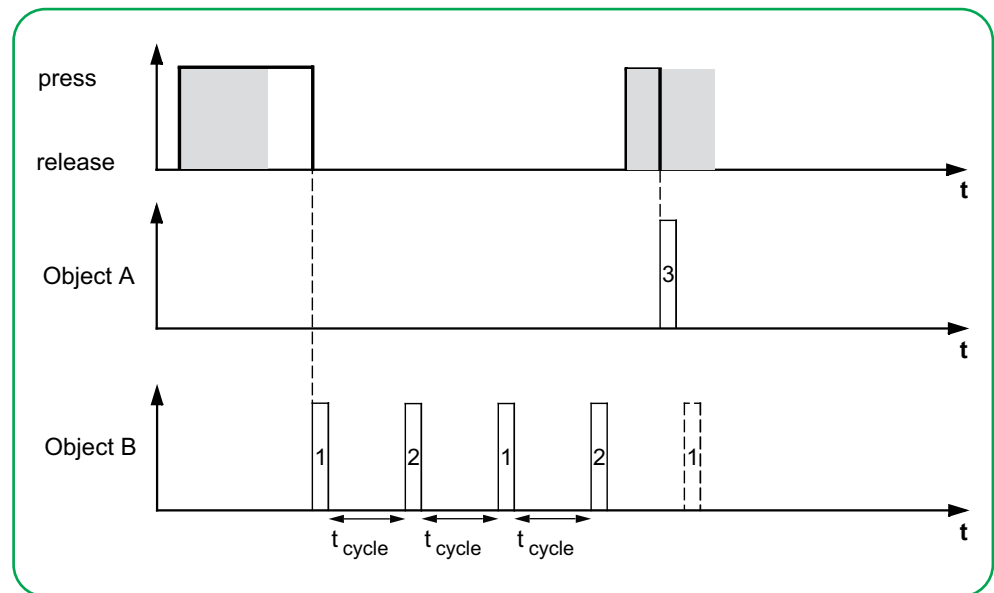
Status indication

In the default setting, the status indication is triggered by the *Switch/Value object*.

Object value	Status indication
On (> 0)	Illuminates bright white
Off (= 0)	Illuminates dark white

You can find comprehensive settings in chapter .

Colors and brightness values
You can find the settings for colors and brightness in chapter .

Group objects

You can select the following group objects.

Group objects for "Edge" function

No.	Name	Object function	Length	Properties	DPT ETS4/5
1/2	Switch object A/B	Function x	1 bit	Sends, receives	1.001 switch
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.001 percent (0-100 %)
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.005 ratio
1/2	Dimming object A/B	Function x	4 bit	Sends, receives	3.007 dimming control
1/2	Priority control object A/B	Function x	2 bit	Sends, receives	2.001 prio. switching
3	Status feedback object	Function x	1 bit	Receives	1.001 switch
3	Status feedback object value	Function x	1 byte	Receives	5.001 percent (0–100 %)

ⓘ **Single-button operation**

In the case of a two-way or central circuit, the push-button function is also controlled via another sensor. With single-button operation, it is possible to adjust to the current state. To do this, the last value sent has to be loaded to the push-button. For *Switch objects* (1 bit), this involves the values "1" and "0".

In the case of toggling with switch actuators, you can use the status feedback function of the switch actuator.

• Also connect the group address of the *Status feedback object* for the connected channel to the *Switch object* (1 bit) for the button function.

Loading values is possible for all object types. Set the same values for all push-buttons, e.g. 70 % for value 1 and 0 % for value 2. If the last sent or received value is the same as value 1, value 2 is sent the after the next action, otherwise value 1 is sent.

• For a two-way circuit, check that the objects for the 2 push-buttons are linked to the same group address.

• In the case of a central push-button, also connect the central group address to the objects of the other push-buttons.

5.5 Edges with 2 byte values

With edge function, you can send a 2 byte object in floating point format or in integer format (with or without sign). In the default setting, you send the value "10" in floating point format on operation.

You can choose between the normal and extended edge functions and set the following functions.

- Send values in floating-point format
- Send values in integer format with or without sign
- Values per object
- Action on operation and on release
- Actions for long and short operation (extended edge function)
- Send cyclically and with delay (extended edge function)
- Trigger status indication

Extended settings		
Function x	Function x	Edges with 2 byte values
	Select edge function	Normal (operate, release)
		Extended (+ long and short operation)
[Normal]	Action on operation	
	Action on release	
[Extended]	Long operation = 100 ms * factor (4-250)	
	Direct action on operation	
	Action on release before the long operating time has elapsed	
	Action on achieving the long operating time	
	Action on release after achieving the long operating time	
[Extended]	Cycle time = basis * factor	
[Edge function]	How is the status indication triggered?	
	LED behaviour	

There is an additional tab under the Function tab for setting the 2 byte values.

Function x - edges values	Type of object	Floating point
		Integer with sign (-32768…32767)
		Integer without sign (0-65535)
[Floating point]	Base 1/2 (possible values in brackets)	0.01...327.68
	Factor 1/2 (0-2047)	0-2047
[Integer with sign]	Value 1/2 (-32768...32767)	-32768...32767
[Integer without sign]	Value 1/2 (0-65535)	0-65535

Normal edge function

With the normal edge function, you can specify which actions should be carried out when a button is pressed, and which should be carried out when a button is released.

Actions for "Normal edge function"

You can set the following actions.

Value: *Sends value 1*
Sends value 1 and stops cyclical sending.

Value: *Sends value 2*
Sends value 2 and stops cyclical sending.

Value: *Sends its value*
The current object value is sent. Therefore you can, for example, send a value with the sending group address that was previously received via another group address. In so doing, you save a setpoint in the push-button and this value is sent when needed.

Value: *None*
No action is carried out

Extended edge function

With the extended edge function, an even wider range of functions is available. In addition to the normal edge function, the extended function differentiates between short and long operation. In total, you set 4 actions for operation and release.

① **Direct action on operation**
  – The action is executed each time the button is operated.
② **Action on release before the long operating time has elapsed**
  – The action is only executed after a short operation.
③ **Action on achieving the long operating time**
  – The action is executed directly when the button is pressed and held. You press the button until the action (e.g. send setpoint) is carried out.
④ **Action on release after achieving the long operating time**
  – The action is also performed on release after pressing and holding.
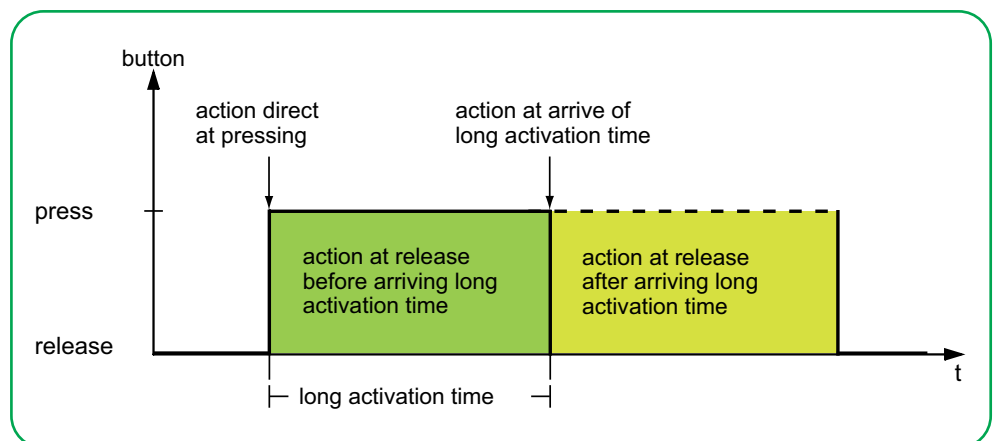
You set the relevant action for each operation phase. In addition to the normal edge function, you can set a cycle time for each object. You can send once or cyclically. You can use the action *Sends value 1 and then value 2 after a cycle time* to send a second value after a delay.

i

- When setting the parameters, remember that you have to set all 4 operating phases for the push-button to function as required.
- In order to read the object values, you have to set the **R**ead flags manually.

Actions for "Extended edge function"

You can send up to 2 values.

Selection 1 [value]	Selection 2 [value]
Value 1	Value 2

You can set the following actions.

Value: *Sends [value]*
Sends the value in question once and stops cyclical sending.

Value: *Sends [value] immediately and then cyclically*
If no cycle time is running, the value is sent immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, the value is sent and a new cycle time is started. The value then continues to be sent cyclically.

Value: *Sends [value] only cyclically*
If no cycle time is running, the value is sent immediately and a new cycle time is started. If a cycle time is already running, it is not interrupted; the value is sent after the current cycle time has elapsed, and a new cycle time is started. The value then continues to be sent cyclically.

Value: *Sets object value to [value] (readable only)*
The value is written to the object and is not sent. Any active cycle time is terminated. If you want the value to be read by a visualisation, for example, you have to set the **R**ead flag for the object.

Value: *Sends its value*
The current object value is sent. Any active cycle time is terminated. Therefore you can, for example, send a value with the sending group address that was previously received via another group address. In so doing, you save a setpoint in the push-button and this value is sent when needed.

Value: *Sends value 1 and then value 2 after a cycle time*
Value 1 is sent immediately, and value 2 is sent after a cycle time, regardless of whether a cycle time is already running or not. You can use this function to send a second setpoint after a delay, for example. You set the duration via the cycle time.

Value: *None (stops cyclical sending)*
No action is carried out, and any active cycle time is stopped. Select this function if you also want to stop cyclical sending.

Value: *No change*
The current action is retained and any active cyclical sending is retained. You select this action for the release if, for example, you have activated the action *Sends value 1 and then value 2 after a cycle time*.

Status indication

In the default setting, the status indication is triggered by pressing the button.

State	Status indication
Operation	Illuminates bright white
No operation	Illuminates dark white

You can find comprehensive settings in chapter Status indication --> 61.

Colors and brightness values

You can find the settings for colors and brightness in chapter LED modes --> 18.

Group objects

You can select the following group objects.

<div style="float:left">Group objects for edges with 2 byte values</div>

No.	Name	Object function	Length	Properties	DPT in ETS4
1	Value object A	Function x	byte	Sends, receives	7.001 pulses 2 byte without sign
1	Value object A	Function x	2 byte	Sends, receives	8.001 pulse difference 2 byte with sign
1	Value object A	Function x	2 byte	Sends, receives	9.004 lux 2 byte floating point value
3	Status feedback object	Function x	1 bit	Receives	1.001 switch
3	Status feedback object value	Function x	1 byte	Receives	5.001 percentage (0–100 %)

5.6 8 bit slider

With the *slider function*, you can send values stepwise or cyclically with 1 button. The values are increased or reduced in steps.

You can operate the button in 2 ways:

- Send a value with each button action (press 5 times to send 5 values).
- Send values cyclically when pressing and holding the button.

If you want to send multiple values, press and hold the button until the last value is sent.

By default, the object value is raised by the value "10" if you release the button before the long operation time elapses.

<div style="float:left">Parameter</div>

You can select the following functions.

- Slider with or without limit values
- Step width of slider
- Either increase or reduce values per button action
- Either increase or reduce values until release
- Either increase or reduce values cyclically from the starting value
- First increase then reduce values per button action
- Toggle direction and send values cyclically

Extended settings		
Function x	Function x	8 bit slider
	Long operation = 100 ms * factor (4-250)	
	How the status indication is triggered?	
	LED behaviour	

In an additional tab under the *Function* tab, you set the *limit values*, the *step width* and the *actions* for the slider.

Function x - slider ⸝	Slider function	
	Direct action on operation	
	Action on release before the long operating time has elapsed	
	Action on achieving the long operating time	
	Action on release after achieving the long operating time	
	Limit value 1 (0-255)	[only for slider function with limit values]
	Limit value 1 (0-255)	[only for slider function with limit values]
	Value of step width	
	Cycle time = base x factor	
	Base	
	Factor (3-255)	

Setting the slider

Extended edge function

For the slider, you always use the extended edge function. The actions differentiate between short and long operation. In total, you set 4 actions for operation and release.

① **Direct action on operation**
   – The action is executed each time the button is operated.
② **Action on release before the long operating time has elapsed**
   – The action is only executed after a short operation.
③ **Action on achieving the long operating time**
   – The action is executed directly when you press and hold the button. You press the button until the action is carried out.
④ **Action on release after achieving the long operating time**
   – The action is also performed on release after pressing and holding.
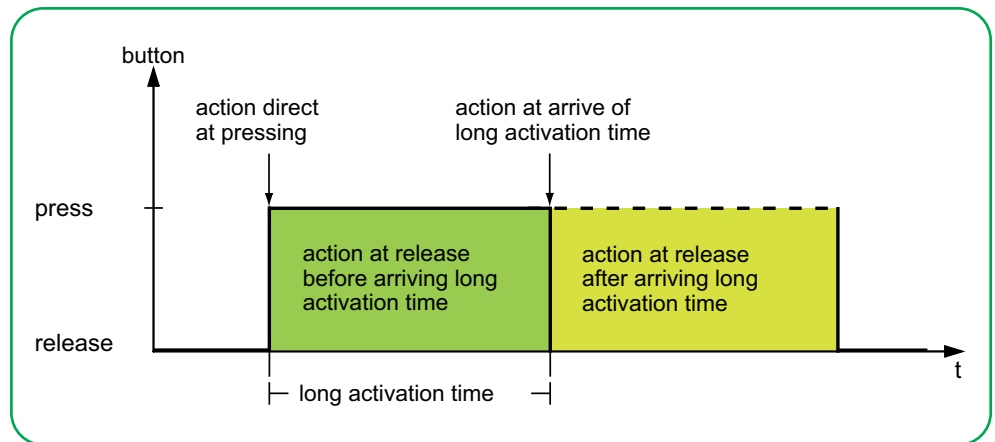
You set the relevant action for each operation phase. You can choose between one-time and cyclically repeated actions.

- When setting the parameters, remember that you have to set all 4 operating phases for the push-button to function as required.

You can use the slider **with and without limit values**.

Without limit values

If you select no limit values for an action, the value returns to value "0" once the maximum value of "255" is exceeded. If the value falls below the minimum value of "0", the value returns to value "255".

With limit values

If you select limit values for an action, the behaviour at the upper and lower limit depends on the action in question. With the action *Increase current object value cyclically*, values are increased up to *limit value 2* and the value does not return to *limit value 1* for this action. With the action *Increase stepwise within limits*, once the upper limit value is reached, the value continues to increase by 1 step width from the lower limit value. Some actions can only be selected together with limit values.

- Limit value 1 is always the lower limit and limit value 2 is the upper limit. Make sure that limit value 1 is always lower than limit value 2.
- If you want to increase or reduce the values consistently by the same interval, select coordinated values. The difference between the upper limit value and the lower limit value must be a multiple of the step width.
   – Example: Limit value 1 = 5, limit value 2 = 50, step width = 5.

Actions   You can set the following actions.

Value: *Send value 1, then increase cyclic. by step width*

You can only select this action with limit values. *Limit value 1* is sent immediately and a new cycle time starts. Next, the value is increased and sent cyclically until *limit value 2* is reached. *Limit value 2* is also sent if the last step is smaller than the step width.

Example
- Limit value 1 = 10
- Limit value 2 = **55**
- Step width = 10
- Sent values: 10, 20, 30, 40, 50, **55**, 10, 20, ...

Cyclical sending can be interrupted. In the next cyclical sending, the value starts again at *limit value 1*. The values are not overwritten from the bus even if the **W**rite flag is set.

Value: *Send value 2, then reduce cyclically by step width*

You can only select this action with limit values. *Limit value 2* is sent immediately and a new cycle time starts. Next, the value is reduced and sent cyclically until *limit value 1* is reached. *Limit value 1* is also sent if the last step is smaller than the step width.

Example
- Limit value 1 = **15**
- Limit value 2 = 50
- Step width = 10
- Sent values: 50, 40, 30, 20, **15**, 50, 40, ...

Cyclical sending can be interrupted. In the next cyclical sending, the value starts again at *limit value 2*. The values are not overwritten from the bus even if the **W**rite flag is set.

Value: *Increase current object value cyclically*

The current object value is increased cyclically by the set step width.

With limit values, the value is increased and sent cyclically until *limit value 2* is reached. *Limit value 2* is also sent if the last step is smaller than the step width.

Example
- Limit value 1 = 10
- Limit value 2 = **55**
- Step width = 10
- Sent values: 10, 20, 30, 40, 50, **55,** 10, 20, ...

If you use limit values for this action, you have to set the **W**rite flag. You also need a second action to reduce the values again. The action *Reduce current object value cyclically* is a suitable complement. To ensure that the values do not shift, select the same step width and the same matching limit values for both actions.

Alternatively, you can use the action *Reverse slide direction and send cyclically* to alternately increase and reduce values.

Without limit values, the value is increased and sent cyclically until the largest possible value is reached. Subsequently, the value "0" is sent and increased cyclically again.

Example
- Step width = 10
- Sent values: ... 230, 240, 250, **0,** 10, 20 ...

The value "255" is only sent if it is reached with the selected step width.

Value: *Increase current object value once*

The current object value is increased once by the set step width. Any active cycle time is terminated.

With limit values, the action can be repeated until *limit value 2* is reached. *Limit value 2* is also sent if the last step is smaller than the step width.

- Limit value 1 = 10
- Limit value 2 = **55**
- Step width = 10
- Sent values: 10, 20, 30, 40, 50, **55**

If you use limit values for this action, you have to set the **W**rite flag. You also need a second action to reduce the values again. The action *Reduce current object value once* is a suitable complement. To ensure that the values do not shift, select the same step width and the same matching limit values for both actions.

Alternatively, you can use the action *Stepwise to the limit values and back again* to increase and reduce values stepwise.

Without limit values, the action can be repeated until the largest possible value is reached. In subsequent actions, the value "0" is sent and increased stepwise again.

Example

- Step width = 10
- Sent values: ... 230, 240, 250, **0,** 10, 20 ...

The value "255" is only sent if it is reached with the selected step width.

Value: *Reduce current object value cyclically*

The current object value is reduced cyclically by the set step width.

With limit values, the value is sent until *limit value 1* is reached. *Limit value 1* is also sent if the last step is smaller than the step width.

If you use limit values for this action, you have to set the **W**rite flag. You also need a second action to increase the values again. The action *Increase current object value cyclically* is a suitable complement. To ensure that the values do not shift, select the same step width and the same matching limit values for both actions.

Alternatively, you can use the action *Reverse slide direction and send cyclically* to alternately increase and reduce values.

Without limit values, the value is reduced until the smallest possible value is reached. Subsequently, the value "255" is sent and reduced cyclically again.

Example

- Step width = 10
- Sent values: ... 25, 15, 5, **255,** 245, 235 ...

The value "0" is only sent if it is reached with the selected step width.

Value: *Reduce current object value once*

The current object value is reduced once by the set step width. Any active cycle time is terminated.

With limit values, the action can be repeated until the minimum value (value 1) is reached. *Limit value 1* is also sent if the last step is smaller than the step width.

If you use limit values for this action, you have to set the **W**rite flag. You also need a second action to increase the values again. The action *Increase current object value once* is a suitable complement. To ensure that the values do not shift, select the same step width and the same matching limit values for both actions.

Alternatively, you can use the action *Stepwise to the limit values and back again* to increase and reduce values stepwise.

Without limit values, the action can be repeated until the smallest possible value is reached. In subsequent actions, the value "255" is sent and reduced stepwise again.

Example

- Step width = 10
- Sent values: ... 25, 15, 5, **255,** 245, 235 ...

The value "0" is only sent if it is reached with the selected step width.

Value: *Reverse slide direction and send cyclically*

The slider direction is reversed and increased or reduced cyclically in the opposite direction by the set step width.

With limit values, the value is sent cyclically until a limit value is reached. The limit value is always sent at the end. Cyclical sending can also be stopped before the limits are reached. Before the next cyclical sending, the direction is reversed and the value is increased or reduced by 1 step width.

Example

- Limit value 1 = 0
- Limit value 2 = 250
- Step width = 50
- Cyclical sending: 50, 100, 150, 200, 250
- Next cyclical sending: 200, 150, 100, 50, 0

Without limit values, the value is increased cyclically in 1 direction until the largest possible value is reached, the value "0" is sent and increased further cyclically in steps. In the other direction, the value is reduced cyclically until the smallest possible value is reached, the value "250" is sent and reduced further cyclically in steps.

Cyclical sending can also be stopped before the end values are reached. Before the next cyclical sending, the direction is reversed and the value is increased or reduced by 1 step width.

Example

Step width 50

Cyclical sending: 50, 100, 150, 200

Next cyclical sending: 150, 100, 50

When increasing, the value "255" is only sent if it is reached with the selected step width. When reducing, the value "0" is only sent if it is reached exactly with the selected step width.

Value: *Stepwise to the limit values and back again*

You can only select this action with limit values. The current object value is changed by 1 step width each time. When a limit value is reached, the slide direction is reversed for the next action.

Example

- Limit value 1 = 0
- Limit value 2 = **55**
- Step width = 10
- Sent values: 0, 10, 20, 30, 40, 50, **55**, 45, 35, 25, ...

Value: *Increase stepwise within limits*

You can only select this action with limit values. The current object value is increased by 1 step width each time. Once the largest possible value is reached, the minimum value (value 1) is sent the next time the button is operated. If the maximum value cannot be reached with the specified step width, it is not sent.

Example

- Limit value 1 = 10
- Limit value 2 = **55**
- Step width = 10
- Sent values: 10, 20, 30, 40, **50**, 10, 20 ...

Value: *Decrease stepwise within limits*

You can only select this action with limit values. The current object value is reduced by 1 step width each time. Once the smallest possible value is reached, the maximum value (value 2) is sent the next time the button is operated. If the minimum value cannot be reached with the specified step width, it is not sent.

Example

- Limit value 1 = **15**
- Limit value 1 = 50
- Step width = 10
- Sent values: 50, 40, 30, **20**, 50, 40 ...

Value: *None (stops cyclical sending)*

No action is carried out, and any active cycle time is stopped.

Value: *No change*

No action is carried out, and any active cycle time is continued.

Status indication

In the default setting, the status indication is triggered by pressing the button.

State	Status indication
Operation	Illuminates bright white
No operation	Illuminates dark white

You can find comprehensive settings in chapter Status indication --> 61.

Colors and brightness values

You can find the settings for colors and brightness in chapter LED modes --> 18.

Group objects

You can select the following group objects.

Group objects for "8 bit slider"

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Value object A	Function 1	1 byte	Sends	5.004 percentage (0-255 %)

5.7   Scene

You can use the scene function to call up and save scenes in actuators and in scene modules. Alternatively to the normal scene function, you can select an extended function. In the default setting, you call up scene address "0" with a short button action and save the values for the scene with a long button action.

Parameter   You can select the following functions.

| Extended settings		
Function x	Function x	Scene
	Long operation = 100 ms * factor (4-250)	
	Select edge function	Normal (operate, release)
		Extended (+ long and short operation)
	Scene address	[Normal edge function]
	Number of objects	[Extended edge function]
	How is the status indication triggered?	
	LED behaviour	

Normal scene function

A short button action calls up a scene. A long button action saves the current values for the scene.

Call up room functions   You can use a scene to change multiple room functions at the touch of a button. Loading a scene allows you, for example, to dim the room lighting to a specific value, move the blinds into the desired position and switch on the power supply to the socket-outlets in a room.

Save room functions   You can change the values for the individual room functions in a scene. To do this, use further button functions such as switching, dimming or move blind. You can use these button functions to change the values for the room functions consecutively. You then save the new values to the scene button by pressing and holding the button.

Settings   In the normal scene function, the parameter *Scene address (0-63)* appears. You can use this value to call up a scene in actuators and in scene modules. The values for saving (128 – 191) are assigned automatically. You can adjust the duration of a long button action. The default setting is a duration of 3 s.

Extended scene function

With the extended scene function, an even wider range of functions is available. In total, you set 4 actions for operation and release.

① **Direct action on operation**
   – The action is executed each time the button is operated.
② **Action on release before the long operating time has elapsed**
   – The action is only executed after a short operation.
③ **Action on achieving the long operating time**
   – The action is executed directly when the button is pressed and held. You press the button until the action is carried out.
④ **Action on release after achieving the long operating time**
   – The action is also performed on release after pressing and holding.
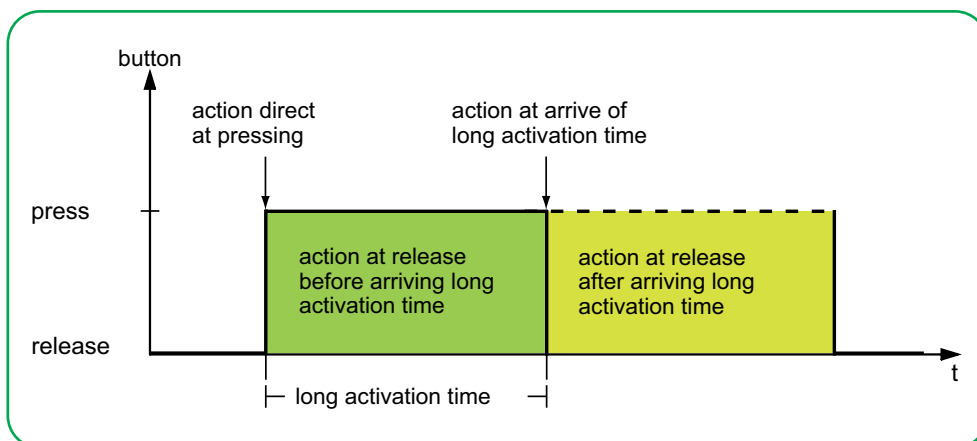
You set the relevant action for each operation phase. You can set 2 values for the scene address for each *Scene object*. For both values, you can set whether the scene is called up or saved. The appropriate value for saving a scene is automatically derived from the scene address.

In addition to the normal scene function, you can set a cycle time. You can send once or cyclically. You can use the action *Sends value 1 and then value 2 after a cycle time* to send a second scene address after a delay.

You can set the actions for 1 or 2 *Scene objects*.

- When setting the parameters, remember that you have to set all 4 operating phases for the push-button to function as required.

Actions for extended "Scene" function

You can set the following actions.

Value: *Send value 1*
Sends the value 1 in question once and stops cyclical sending.

Value: *Send value 2*
Sends the value 2 in question once and stops cyclical sending.

Value: *Toggles*
Sends the set values alternately. The toggling is not controlled via the bus. Received telegrams are not evaluated.

Value: *Toggles cyclically, sends immediately, then cyclically*
If no cycle time is running, the value is toggled once, sent immediately and a new cycle time is started. If a cycle time is already running, it is interrupted, the value toggled once is sent and a new cycle time is started. Then, the value continues to be sent cyclically, but without further toggling. The toggling is not controlled via the bus. Received telegrams are not evaluated.

Value: *Sends value 1 and then value 2 after a cycle time*
Sends scene address 1 immediately, and scene address 2 after a cycle time, regardless of whether a cycle time is already running or not. With this action, you can call up a scene for an adjustable duration and then switch back to another scene.

Value: *None (stops cyclical sending)*
No action is carried out, and any active cycle time is stopped.

Value: *No change*
No action is carried out, and any active cycle time is continued.

For each *Scene object*, there is a further tab under the Function tab:

Function x - scene object x	Direct action on operation
	Action on release before the long operating time has elapsed
	Action on achieving the long operating time
	Action on release the long operating time
	Value 1 Scene address (0-63)
	Value 1 to retrieve/save the scene
	Value 2 Scene address (0-63)
	Value 2 to retrieve/save the scene
	Cycle time = base * factor
	Base
	Factor (3-255)

Status indication

In the default setting, the status indication is triggered by pressing the button.

State	Status indication
Operation	Switched on (bright white)
No operation	Switched on (dark white)

You can find comprehensive settings in chapter .

Colors and brightness values
You can find the settings for colors and brightness in chapter .

Group objects

You can select the following group objects.

Group objects for "Scene" function

No.	Name	Object function	Length	Properties	DPT ETS4/5
1/2	Scene object A/B	Function x	1 byte	Sends	18.001 scene control

5.8 Status indication

You can set the status indication individually for each button function. Status indication is triggered by group objects, buttons, objects and buttons together. You can as well leave the status indication on permanently (Always on, Always off, Flashes).

LED behaviour

If either condition is met, the LED for *Status On* lights up, if not, the LED for *Status Off* lights up.

The setting for the *LED behaviour* specifies which LED lights up. In the default setting, the LED lights up or flashes bright white when *Status On* is activated. The LED lights up dark white when *Status Off* is activated.

Alternatively, the activation depends on a button operation or a group object, 1 of the LEDs can be switched on permanently or both LEDs can be switched off permanently.

i LED activation can also be switched off. For example, the proximity function is activated in the default setting (Proximity function --> 21). If no proximity is detected, all LEDs remain switched off in normal operation. LED activation is only reactivated when a person approaches. For more information about activating all LEDs, see chapter LED modes --> 18.

You set the activation of the status indication on the tab for the corresponding button function.

Extended settings	
Function x	Function x
	...
	How is the status indication triggered?
	LED behaviour

Activation by objects

If the status indication is activated by objects, the system evaluates whether the object value is equal to or greater than "0".

Following objects activate status indication:

Object (trigger)	Object value and operation
[Switch/value object]	1 bit/1 byte
	Edges 1 bit, 2 bit (priority), 4 bit, 1 byte values
	Switch/value object A
	Switch/value object B
[Status feedback object]	1 bit/1 byte
[Collected status feedback object]	4 byte

i Switch/value object is not available for "Edges with 2 byte value" and "Scene"

Status feedback object 1 bit

Extended settings		
Function x	How the status indication is triggered?	Status feedback object 1 bit
	LED behaviour	Status feedback object = On/Off
		Status feedback object = On/Off inverted
		Status feedback object equal to 1 = Flashes
		Status feedback object equal to 0 = Flashes
		Long operation = Flashes / Feedback 1 bit = On/Off
		Long operation = Flashes / Feedback 1 bit equal to 1 = Flashes

Status feedback object 1 byte

Extended settings		
Function x	How the status indication is triggered?	Status feedback object 1 byte
	LED behaviour	Status feedback object value = On/Off
		Feedback 1 byte not equal to 0 = Flashes
		Feedback 1 byte equal to 0 = Flashes
		Long operation = Flashes / Feedback 1 byte = On/Off
		Feedback 1 byte equal to 0 = On
		Feedback 1 byte equal to 255 = On

Switch/value object

Extended settings		
Function x	How the status indication is triggered?	Switch/value object
	LED behaviour	Switch/value object A = On/Off
		Switch/value object B = On/Off
		Switch/value object A not equal to 0 = Flashes
		Switch/value object B not equal to 0 = Flashes
		Switch/value object A equal to 0 = Flashes
		Switch/value object B equal to 0 = Flashes

For *Collected status feedback object*, you set the channel of the actuator that sends the status feedback (channel 1–16). You can connect a shared *Collected status feedback object* for all buttons and set the corresponding actuator channel for each button function

Collected status feedback object

Extended settings		
Function x	How the status indication is triggered?	Collected status feedback object
	Which channel sends?	None
		1..16

Parameter value	Object value	Status indication
[Switch/value object] = On/Off	> 0	Switched on (status On)
	= 0	Switched on (status Off)
[Status feedback object] = On/Off	> 0	Switched on (status On)
	= 0	Switched on (status Off)

Parameter value	Object value	Status indication
[Switch/value object] not equal to 0 = Flashes	> 0	Flashes (status On)
	= 0	Switched on (status Off)
[Switch/value object] equal to 0 = Flashes	> 0	Switched on (status Off)
	= 0	Flashes (status On)
[Status feedback object] not equal to 0 = Flashes	> 0	Flashes (status On)
	= 0	Switched on (status Off)
[Status feedback object] equal to 0 = Flashes	> 0	Switched on (status Off)
	= 0	Flashes (status On)
Collected status feedback object = On/Off	> 0	Switched on (status On)
	= 0	Switched on (status Off)

Activation by button

If the status indication is activated by button operation, both operation and long operation can be evaluated.

Button operation

Extended settings		
Function x	How the status indication is triggered?	Button operation
	LED behaviour	Operation = On / Release = Off
		Operation = Flashes / Release = Off
		Long operation = On / Release = Off
		Operation = Off / Long operation = Flashes / Release = Off
		Operation = On / Long operation = Flashes / Release = Off

Parameter value	Operation	Status indication
Operation = On / Release = Off	Operation	Switched on (status On)
	Release	Switched on (status Off)
Operation = Flashes / Release = Off	Operation	Flashes (status On)
	Release	Switched on (status Off)
Long operation = On / Release = Off	Short operation	Switched on (status Off)
	Long operation	Switched on (status On)
	Release	Switched on (status Off)
Operation = Off / Long operation = Flashes / Release = Off	Short operation	Switched on (status Off)
	Long operation	Flashes (status On)
	Release	Switched on (status Off)
Operation = On / Long operation = Flashes / Release = Off	Short operation	Switched on (status On)
	Long operation	Flashes (status On)
	Release	Switched on (status Off)

Activation by object and button

The long operation and a status feedback can also be evaluated together. This function is set in the express function for dimming. You select this function if you want to display a device state and the long operation.

Parameter value	Object value and operation	Status indication
Long operation = Flashes / Feedback 1 bit = On/Off	> 0	Switched on (status On)
	= 0	Switched on (status Off)
	Long operation	Flashes (status On)
Long operation = Flashes / Feedback 1 byte = On/Off	> 0	Switched on (status On)
	= 0	Switched on (status Off)
	Long operation	Flashes (status On)

Permanent activation

The status indication can also be switched on, switched off or flash permanently. The objects and button operation are not evaluated. However, the settings for the proximity function and for night mode also apply here.

If the proximity function is active and no proximity is detected, all status indications are switched off in normal operation. If the default behaviour for night mode is selected, only 1 status indication lights up during night mode ().

Always on or off

Extended settings		
Function x	How the status indication is triggered?	Always on or off
	LED behaviour	Always on = Off
		Always on = On
		Flashes
		Always off

Parameter value	Object value and operation	Status indication
Always on = On	Not evaluated	Switched on (status On)
Always on = Off	Not evaluated	Switched on (status Off)
Flashes	Not evaluated	Flashes (status On)
Always off	Not evaluated	Switched off

6 Behaviour when bus voltage is connected/recovered

6.1 Behaviour when bus voltage is connected / recovered

Depending on the application settings, the status LEDs are switched on or off or they flash.

6.2 Behaviour when bus voltage fails

Any status LEDs that were lit are switched off.

The current values of the group objects are not saved.

7   Overview of group objects

No.	Name	Object function	Length	Properties	DPT ETS4/5
1	Switch object	Function 1	1 bit	Sends, receives	1.001 switch
2	Dimming object	Function 1	4 bit	Sends, receives	3.007 dimming control
1	Stop/step object	Function 1	1 bit	Sends, receives	1.007 step
2	Movement object	Function 1	1 bit	Sends, receives	1.008 up/down
2	Dimming object	Function 1	4 bit	Sends, receives	3.007 dimming control
1	Scene object	Function 1	1 byte	Sends	18.001 scene control
1/2	Switch object A/B	Function x	1 bit	Sends, receives	1.001 switch
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.001 percentage (0..100 %)
1/2	Value object A/B	Function x	1 byte	Sends, receives	5.005 ratio (0-255)
1	Blind position	Function 1	1 byte	Sends	5.005 ratio (0-255)
2	Slat position	Function 1	1 byte	Sends	5.005 ratio (0-255)
1/2	Priority control object A/B	Function 1	2 bit	Sends, receives	2.001 prio. switching
1/2	Dimming object A/B	Function 1	4 bit	Sends, receives	3.007 dimmer step
1	Value object A	Function x	2 byte	Sends, receives	7.001 pulse (2 byte without sign)
1	Value object A	Function x	2 byte	Sends, receives	8.001 pulses difference (2 byte with sign)
1	Value object A	Function x	2 byte	Sends, receives	9.004 lux (Lux)
1	Value object A	Function 1	1 byte	Sends	5.004 percentage (0..255 %) (8 bit slider)
1/2	Scene object A/B	Function 1	1 byte	Sends	18.001 scene control
3	Status feedback object	Function 1	1 bit	Receives	1.001 switch
3	Status feedback object value	Function 1	1 byte	Sends	5.004 percentage (0..255 %)
13	Locking object	Button lock	1 bit	Sends, Receives	1.003 enable
14	Proximity input	Proximity function	1 bit	Receives	1.001 switch
15	Proximity output	Proximity function	1 bit	Sends	1.001 switch
15	Proximity output	Proximity function	1 byte	Sends	5.005 ratio (0..255)
16	Night mode input	Night mode LEDs	1 bit	Receives	1.001 switch
17	Collected status feedback object	Feedback	4 byte	Receives	27.001 bit-combined info on/off
18	Actual temperature Fahrenheit	Temperature sensor	2 byte	Sends	9.027 temperature (°F)
18	Actual temperature Celsius 2B	Temperature sensor	2 byte	Sends	9.001 temperature (°C)
19	Actual temperature Celsius 4B	Temperature sensor	4 byte	Sends	14.068 temperature (°C)
20	Disable touchless function	Touchless function	1 bite	Receives	1.003 enable

This list provides the numbers for uniquely identifying a group object. The numbers 0 to 3 are assigned to button function 1.

Numbers	Push-button function
1 to 3	1
4 to 6	2
7 to 9	3
10 to 12	4

DPT

The data point types (DPT) in this application are preset.

8 Index

O

Operation
    Assignment of buttons → 13
    LED operating modes → 18
    Locking function → 20
    Night mode → 19
    Normal operation → 19
    Number of buttons → *See* Number of buttons
    Proximity function → 21
    Single-button operation → 13, 14, 15, 26, 28, 29, 30, 31, 32, 47
    Touchless function → 11, 18, 25
Orientation light → *See* LED operating modes

P

Position values → 32
    Blind position (object) → 33
    Slat position (object) → 33
Priority control object → 34, 38
Proximity function → 18, 21
    Proximity input (object) → 21
    Proximity output (object) → 22
    Proximity sensor → 21
    Proximity sensor sensitivity → 11, 18

R

Roller shutter → *See* Blind
Room functions → *See* Extended settings; *See* Express settings

S

Save room functions → 16, 58
Scene → 16, 58
    Scene address → 16, 58, 60
    Scene object → 17, 60
Send values cyclically → *See* Priority control object; *See* Slider; *See* Extended edge function
Send values stepwise → *See* Slider
Single-button operation → *See* Operation
Slat position (object) → 33
Slider → 51
Staircase lighting → 41, 43, 44
Status feedback object → 12, 13, 28, 29, 31, 33, 47, 51, 61
Status feedback object value → 14, 15, 47, 51
Step dimming → 30
Stop/step object → 16, 33
Switch object → 12, 13, 14, 28, 30, 43, 44, 45, 47

T

Temperature sensor → 18, 23
    Deviation value → 23
    Temperature correction → 23
Touchless function → *See* Operation

V

Values (1 byte) → 26, 28, 34, 38
Values (2 byte)
    Value object → 48

MTN6185-03_04_60_SW_EN_1841_1.0