System D

KNX Push Button Dynamic Labeling

Universal 1851/1.1a
Push-button 3-gang 1854/1.1a
Push-button 2-gang 1853/1.1a
Push-button 1-gang 1852/1.1a

Application description

MTN6191-6010 / MTN6192-6010 / MTN6193-6010 / MTN6194-6010

12/22

www.se.com

Legal information

Trademarks

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

The ETS is the **software for the KNX system.** It 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.

Appropriate ETS version

The application is suitable for ETS5 or higher version (hereinafter reffered to as “ETS”).

ETS tabs, parameters and values

Overview - setting functions

The following overview helps you to understand how to access the functions.

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

Example

Meaning:

1. Go to the *Button* tab and set the *Select button function* parameter to value *Scene*.
2. Further parameters then appear in the tab. You can use them to change settings.
3. A new tab also opens.

Special features of the ETS software

Restoring defaults

Default Parameters button

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.

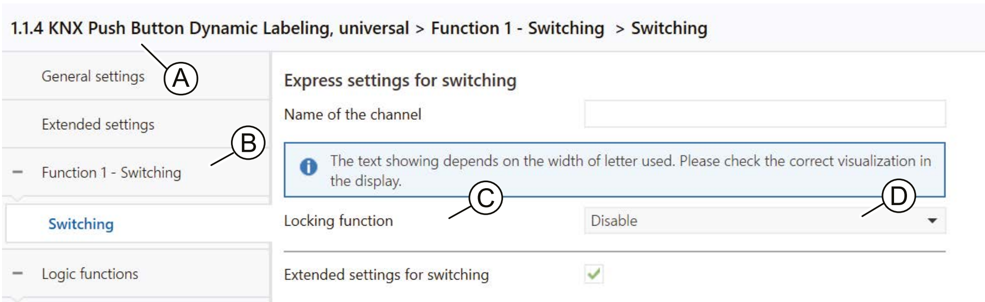
Dependent functions and parameters

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.

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.



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

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 ETS software.

Group objects in the ETS

No.	Name	Object function	Length	Properties	DPT ETS
1	Function 1	Scene	1 byte	Sends	18.001 scene control
41	Function 1	Status feedback object	1 bit	Sends, Receives, Updates	1.001 switch

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

Group addresses

As the group address only consists of a **sequence of numbers**, it is very important to briefly describe it in the ETS, to assign a name (usually the designation of the device and the basic function of the device.

No.	Name	Object function	Description	Group Addresses
1	Input A	Switch telegram	Central ON	11/2/2

Table of contents

1 For your safety

DANGER

Hazard of electric shock, or arc flash.

Safe electrical installation must be carried out only by skilled professionals. Skilled professionals must prove profound knowledge in the following areas:

- Connecting to installation networks
- Connecting several electrical devices
- Laying electric cables
- Connecting and establishing KNX networks
- Safety standards, local wiring rules and regulations.

Failure to follow these instructions will result in death or serious 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 Overview of functions

The number of buttons and functions varies by module. The following overview refers to the universal module.

Channel	Level 2	Level 3
General settings	Button function	Function 1 – 8
	Device type	Device type
		Device setting for the left and right side
		Number of buttons
	Connect button with function	Button function – Left/Right side
		Button 1 – 8
	Middle field display	Indication type of line 1 – 4
Extended settings	Startup-delay	Additional startup-delay of application
	Device health	Cyclic sending live signal
	Night mode	
	Cleaning mode	Time setting for cleaning mode
	Interface language	
	Display settings	Backlight level in normal/night mode
		Standby mode
	Orientation indicator	Indicator visibility
		Color and brightness of LED
	Proximity function	How the proximity function is triggered
		Type of output object
		Send output value cyclically
	Internal temperature sensor	Internal sensor compensation
		Datapoint type for temperature object
		Send temperature when the result changes by
		Cyclically send temperature
Function 1 – 8	Express/Settings for function 1 – 8	No function
	Icon configuration	Toggle
		Switch
		Dimming
		Blind
		Edges 1-bit, 2-bit (priority), 4-bit, 1-byte
		Edges with 2-byte values
		8-bit slider
		Scene
		RGB lighting
		Color temperature control
		Temperature decrease
		Temperature increase
Logic	Logic functions	1st – 8th logic function

Group addresses, group objects

Nr. of group addresses	250
Nr. of maximum assignments	250
Group objects	150

Overview of group objects → 74.

3 General settings

In the *General settings* section, you can select the functions and the number of buttons on the left and right. You can assign specific functions to the buttons and select icons for the middle display.

Long and short operation

Long and short operations are preset:

- For a short operation, press < 0,5 s.
- For a long operation, press ≥ 0,5 s.
- To save a scene, press ≥ 10 s.

3.1 Button function

Depending on the type of device, you have a certain number of buttons (2 – 8). You can assign a function to each button.

General settings	Button function	
	Function 1 – 8	No function
		Toggle
		Switch
		Dimming
		Blind
		Edges 1-bit, 2-bit (Priority), 4-bit, 1-byte values
		Edges with 2-byte values
		8-bit slider
		Scene
		RGB lighting
		Colour temperature control
		Temperature decrease
		Temperature increase

3.2 Device type

Depending on the product type, the product diagram displays here.

For the 1/2/3-gang type, you can not change the number of buttons on the right and left. The universal type allows you to select the number of left and right buttons from 1 to 4.

General settings	Device type	
	Device type	Universal/1-gang/2-gang/3-gang
	Device setting for	Left side/Right side
	Number of buttons	1 – 4

3.3 Connect buttons with function

You can change the default function number assignment for each button.

Default function assignment

Button function	Left side	Right side
Button 1&2	Function 1	Function 2
Button 3&4	Function 3	Function 4
Button 5&6	Function 5	Function 6
Button 7&8	Function 7	Function 8

3.4 Middle field display

This setting allows you to choose what will display in the middle of the screen.

General settings	Middle field display	
	Indication type of line 1 – 4	Text only Icon only Icon + text Temperature only Temperature + text Not used
	Description	
	Icon preview	
	Icon	
	Color for icon status	White/ Green/ Blue/ Red/ Orange/ Yellow

Text and/or icons

If you choose to display text or icons, in the next step, you select the desired icon from the menu and enter text with a maximum length of 10 characters.

The text length depends on the width of letter used. Check the correct visualization on the display.

The number of lines you can set depends on the device type:

- 1-gang → 1 line
- 2-gang → 2 lines
- 3-gang → 3 lines
- Universal → 4 lines

Dimming icon

If you select the dimming icon you can also tick the dimming level.

If you tick the dimming level, you then need to set the Stay time parameter. It is the time interval to receive the dimming value from the bus via the *Brightness level status feedback* object. If the value is received, the dimming level displays in the middle field for the set time. After the time expires, the dimming icon displays.

Temperature and text

If you choose to display the temperature, you can select the type of sensor, set the measurement interval, and the time to return from the setpoint to the current temperature.

General settings	Middle field display		
	Temperature only Temperature + Text	Actual temperature from	Internal/External
	(Temperature + text)	Description of setpoint	Max. 10 characters
	(Temperature + text)	Description of internal/external sensor	Max. 10 characters
	(External sensor)	Interval time of external sensor	1 – 255, unit = 1 min
		Return time from setpoint to actual temperature	0,5 – 10 s
		Control mode	Heating Cooling Heating and Cooling

If you select the *Temperature only* or *Temperature + Text* in combination with set-point adjustment function (see more in Temperature decrease/increase → 65),

the setpoint temperature displays when you push the button. After a release, the actual temperature displays after a preset interval.

The actual temperature comes from internal sensor or external sensor via bus.

If you select the external sensor, you have to set the interval of sending the temperature value request in the next step. If the device do not get the value in the interval time, the middle field display shows ‘’-- °C’’.

The display color of the setpoint temperature can additionally be changed by the heating or cooling mode.

Control mode

The control mode is 1-bit information. If you select *Heating*, the display color changes to red. With *Cooling*, the color is blue. If you select *Heating and Cooling*, the color linked with the *Heating and cooling mode* object automatically change over according to the value from the bus.

Group objects

See chapter Overview of group objects → 74.

4 Extended settings

In the Extended settings tab you can set other device parameters such as:

- Startup delay
- Live signal sending
- Night mode
- Cleaning mode
- Language
- Display backlight level
- Standby delay
- Orientation indicator
- Proximity function
- Internal temperature sensor

4.1 Startup delay

Startup delay is a time by which the device startup delays after power-on. The device initialization time is not included.

Set the value from 0 (disabled) – 30 seconds.

During the delay period the device sends no telegrams to the bus and the channels do not change their state.

After the delay expires, the telegrams are sent and the state of channels is set according to the parameters defined.

Any telegrams received from group objects during the delay period get stored. Replies to these telegrams are sent after expiry of the delay time.

You can use the startup delay to reduce load on the bus and supply circuit after power-on. It also informs you that the bus is ready for communication and the devices are powered.

4.2 Device health

Cyclic sending live signal

You can set up **cyclic sending** of messages from your device. When there is no signal received, the device either does not work or is missing.

Set the value from 0 (disabled) – 255 seconds.

4.3 Night mode

Day and night mode

You can set the day and night mode switching via the 1-bit *Night mode input* object.

The night mode function is on by default.

Group objects

See Overview of group objects → 74.

4.4 Cleaning mode

You can set the time to exit cleaning mode after you turn it on.

Press and hold one button on each side of the panel for 10 seconds at the same time to trigger the countdown.

The countdown displays on the screen. All buttons are temporarily disabled during the countdown.

Time setting for cleaning mode

The device turns on automatically after the time you set (5 s – 60 s).

4.5 Programming mode

Activation of the programming mode

You can activate the programming mode in two ways:

- Press the KNX programming button on the back side of the device.
- Press one button at the front on each side at the same time as if you want to start cleaning mode and add an extra 5 seconds (10 s + 5 s = 15 s).

4.6 Interface language

Set your interface language.

Extended settings	Interface language	English German Spanish French Italian

4.7 Display settings

In the *Display settings* section, you can set the display backlight level in normal, night, and standby mode.

Extended settings	Display settings	
	Backlight level in normal mode	5 – 100 %
	Backlight level in night mode	5 – 100 %
	Standby mode	Tick/Untick
	Temperature display units	Celsius/Fahrenheit

It is possible to set the units for displaying the temperature. This setting affects all the temperature displays, including the middle field and the screen saver tempera-ture.

If you allow standby mode, you can also customize the appearance of the standby display.

Extended settings	Display settings	
	Standby mode	✓
	Standby backlight level	5 – 100 %
	Standby display	Unchange Date and time Temperature Date, time and temper-ature
	Delay time for normal to standby	1 – 255, unit = 1 s
	Delay time from standby back to normal after wake-up	Disable/0,5 s – 3 s
	Room temperature display	Internal/External/Both
	Description of internal/external sensor	Max. 10 characters
	Interval time of external sensor	1 – 255, unit = 1 min.

The device switches back to normal mode via the proximity sensor or buttons.

Standby display mode

If you choose *Date and time* interface and enable the proximity function you can use this parameter to set when your interface automatically returns to normal mode when waking up.

If you choose *Temperature* or *Date, time, and temperature*, you can select which temperature you want to display on the standby screen (internal/external/both).

Delay time for normal to standby

If the user does not operate the device within the preset time, the device switches from normal to standby mode. The time interval can be influenced by the proximity *Off delay* parameter.

Delay time from standby back to normal after wake-up

The device awakes first to standby and then enters normal mode either with preset delay or if you press any button.

Unchange standby display mode

If you disable the proximity function and select *Unchange* for standby display, you have to press any button to return from standby to normal mode.

See more in Proximity function → 17.

Room temperature display

You can choose to display the temperature from internal, external, or both sensors and name them.

Group objects

See Overview of group objects → 74.

4.8 Orientation indicator

Orientation light helps you with orientation in the dark. You can set the color of the LED and its brightness.

Extended settings	Orientation indicator	Disable/Visible in night mode/Always visible
	Color of LED	Warm/Cool white
	Brightness of LED	0 – 100 %

Group objects

See Overview of group objects → 74.

4.9 Proximity function

If you come within 12 cm of the device, the *Proximity function* triggers. The display switches on and switches off again after off delay elapses.

The *Proximity function* is activated by default. You can adjust the proximity triggering (default: built-in proximity sensor):

Extended settings	How the Proximity function is triggered	Never Proximity object Sensor Sensor or Proximity object

Value: *Never*

The function is deactivated.

The display is not affected.

Value: *Proximity object*

The proximity function is triggered via the 1-bit *Proximity input* object.

The proximity object has the same function as the internal proximity sensor.

- 1 telegram activates the **proximity** state.
- 0 telegram activates the **no proximity** state.

Value: *Sensor*

The proximity function is triggered via the internal proximity sensor. The internal sensor sends a 1-bit or 1-byte signal to the bus.

The **proximity** and **no proximity** states control the status indication.

Value: *Sensor or Proximity object*

The proximity function is triggered via the internal sensor or the external input object.

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

If the proximity sensor detects **no proximity**, it sends a 0 telegram to the bus.

Example

1 presence detector

Off delay status indication

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 and the preset off delay time has elapsed, the display switches off.

Off delay status indication

This parameter sets the time for the status indication and screen backlight to be turned off.

Extended settings	Off delay status indication = basis x factor	
	Basis	0,1 s/1 s/1 min
	Factor	5 – 255

When there is proximity sensing, the status is indicated according to the current object value or operation. After leaving, the status indication and screen backlight turn off as soon as the delay time elapses.

When the screen is off, the object value can be updated through the bus, but no indication is given.

During standby, if you select the *Unchange* option (see Display settings → 16), the icon status will be updated.

If there is a standby mode:

The proximity sensor cannot directly enter the normal mode from the off-screen state (you need to configure it according to the standby mode).

If there is no standby mode:

The proximity sensing can make the screen go from off to normal mode.

Type of output object

If the *Sensor* is involved as a trigger of the *Proximity function*, **proximity** and **no proximity** states control the *Proximity output* object.

The proximity output can be set as:
- 1-bit object - sends values 1 and 0 (no proximity)
- 1-byte object - sends an adjustable value and 0 (no proximity)

Proximity	Type of output object	No object 1 bit 1 byte

Send output value cyclically

You can set sending the output values cyclically.
After the device is powered up for one cycle, the output object starts to send the current output value cyclically.
Cyclical sending and sending changes are independent and do not interfere with each other.

Group objects

See Overview of group objects → 74.

4.10 Internal temperature sensor

The device has a built-in internal temperature sensor. You can set parameters for measuring and sending telegrams.

Extended settings	*Internal temperature sensor*	
	Internal sensor compensation	0,1K * factor (- 30 – +30)
	Temperature unit	Celsius/Fahrenheit
	Object type selection	2 byte/4 byte/2 byte and 4 byte
	Send temperature when the result changes by	Disable/0,1 – 2 K
	Cyclically send temperature	0 – 255, unit = 1 min, 0 = inactive

Internal sensor compensation

You can set a **compensation value** for the sensor. This is useful, for example, if the device is mounted at an unfavourable position in the room. The temperature recording is different when exposed to a draught or close to sources of heat, for example, compared to other places in the room.

The following applies:

Actual temperature = measured temperature + compensation value

Temperature unit

You can set the unit of temperature data. When you select Fahrenheit, the internal data convert before sending.

You can set two parameters for sending the measured temperature to the bus:

Send temperature when the result changes by

- **Temperature difference**: The sensor compares the current temperature with the last value transmitted. If the current measured temperature is higher or lower than the selected deviation, the sensor sends the value to the bus.

Cyclically send temperature

- **Time interval**: The sensor transmits temperature values cyclically after the pre-set time interval. (e.g. to visualisation software).

You can use one or a combination of both parameters.

Group objects

See Overview of group objects → 74.

5 Functions

5.1 Toggle

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 the status indication

Function X – Toggle	*Express settings for toggle*	
Toggle	Name of the channel	*12 bytes allowed*
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1
	Behavior indication when locking	No Display lock icon
	How the status indica-tion is trigerred	Switch/value object A/B = On/Off Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Always on = Off Always on = On
	Extended settings for toggle	✓

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

Status indication

In the default setting, the status indication is triggered by the *Switch/value object = On/Off*.

How the status indication is triggered

Trigger	Note
Switch/value object A/B = On/Off	If the object is 1 byte, the value > 0 = the state is On, and the value 0 = Off.
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.

How the status indication is triggered

Trigger	Note
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Always on = Off/On:	The icon is always Off/On.

Extended settings for toggle

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.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

5.2 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

Function X – Switch	*Express settings for switching*	
Switch	Name of the channel	*12 bytes allowed*
	Locking function	Disable Locking = 1/ Unlocking = 0 Locking = 0/ Unlocking = 1
	Behavior indication when locking	No Display lock icon
	How the status indication is trigerred	Switch/value object A/B = On/Off Status feedback object 1 bit Status feedback object 1 byte Operation = On/Release = Off Operation = Off/Release = On Always on = Off Always on = On
	Extended settings for switching	✓

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

Extended settings for switching

1 bit, 1 byte button functions
Number of objects

You can select 2 objects. You can specify the object type of each object (1 bit or 1 byte.

The 1 bit object type is used for normal switching.

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 *Status feedback object 1 bit* object.

How the status indication is triggered

Trigger	Note
Switch/value object A/B = On/Off	If the object is 1 byte, the value > 0 = the state is On, and the value 0 = Off.
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.

How the status indication is triggered

Trigger	Note
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

5.3 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

Function X – Dimming	*Express settings for dimming*	
Dimming	Name of the channel	*12 bytes allowed*

	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1
	Behavior indication when locking	No Display lock icon
	How the status indica-tion is trigerred	Switch/value object A = On/Off Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Long operation = On/Release = Off Always on = Off Always on = On
	Extended settings for dimming	✓

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

Status indication

In the default setting, the status indication is triggered by the *Switch/value object A = On/Off*.

How the status indication is triggered

Trigger	Note
Switch/value object A = On/Off	If the object is 1 byte, the value > 0 = the state is On, and the value 0 = Off.
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object value 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Long operation = On / Release = Off	The long operation state = On Release button state (or short operation) = Off
Always on = Off/On:	The icon is always Off/On.

Extended settings for dimming

Function X – Dimming	Extended settings for dimming	✓
Dimming	Time for long operation	4 – 250, unit = 100 ms
	Dimming direction	Dimming brighter Dimming darker Dimming brighter/darker
	Step dimming	1/2 to 1/64

	Send stop telegram after release	
	Send dimming command cyclically	✓
	Basis	0,1 s 1 s 1 min
	Factor	3 – 255

Single-button dimming

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 to 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.

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*.

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.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the backlight color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	

	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

5.4 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.

Function X – Blind	*Express settings for blind*	
Blind	Name of the channel	*12 bytes allowed*
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1
	Behavior indication when locking	No Display lock icon
	Movement direction of blind	Moving up/down
	How the status indication is trigerred	Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Long operation = On/Release = Off Always on = Off Always on = On
	Extended settings for blind	✓

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock). The locked button can be indicated by a lock icon or not at all. If you select *Disable*, the locking function becomes inactive.

Extended settings for Blind

Function X – Blind	Extended settings for blind	✓
Blind	Movement direction of blind	Moving up Moving down Moving up/down Moving with positioning values
	Time for long operation	4 – 250, unit = 100 ms

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.

Function X – Blind	Extended settings for blind	
Blind	Movement direction of blind	Moving up or Moving down
	Time for long operation	4 – 250, unit = 100 ms

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 for change slat 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.

Function X – Blind	Extended settings for blind	✓
Blind	Movement direction of blind	Moving up/down
	Pause for change slat direction	5 – 50, unit = 100 ms
	Time for long operation	4 – 250, unit = 100 ms

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.

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.

Function X – Blind	*Extended settings for dimming*	
Blind	Movement direction of blind	Moving with positioning values
	Number of positionings	1 (short operation) 2 (short/long operation)
	Type of position values	Moving blind with positioning values
	Number of positionings	0 – 100% 0 – 255
	Positon 1/2 of blind	
	Position 1/2 of slats	

You send the values for the positions via the 1-byte objects *Blind position* and *Slat position*.
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.

How the status indication is triggered

Trigger	Note
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Long operation = On / Release = Off	The long operation state = On Release button state (or short operation) = Off
Always on = Off/On:	The icon is always Off/On.

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 object value* for the button function.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the backlight color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	

	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

5.5 Edges 1-bit, 2-bit (Priority), 4-bit, 1-byte values

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 → 38.

Function X – Edges Edges 1-bit, 2-bit, 4-bit, 1-byte	*Express settings for edges 1-bit, 2-bit, 4-bit, 1-byte*	
	Name of the channel	*12 bytes allowed*
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1
	Behavior indication when locking	No Display lock icon
	Type of object A	1 bit 2 bit (priority control) 4 bit 1 byte (0 – 100 %) 1 byte (0 – 255)

	Action on operation Action on release	Send value 1 Send value 2 Toggles Sends its value None
	How the status indication is trigerred	Switch/value object A/B = On/Off Status feedback object 1 bit Status feedback object 1 byte Operation = On/Release = Off Operation = Off/Release = On Always on = Off Always on = On
	Extended settings for blind	

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

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.

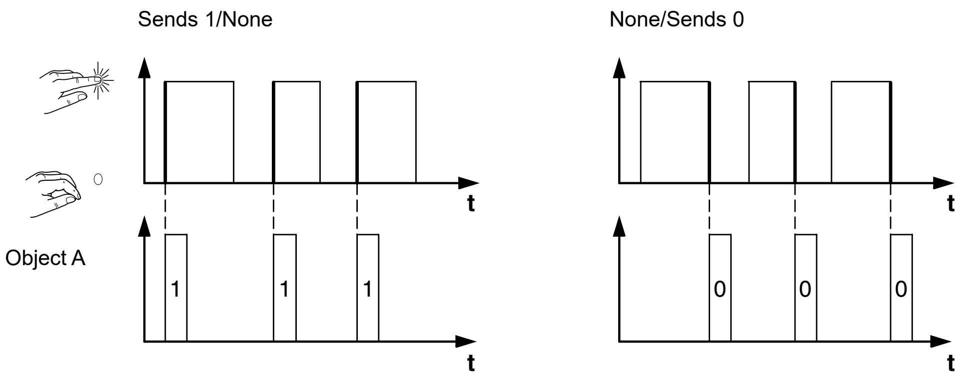
Value: *None*
No action is carried out

Principle of the normal edge function

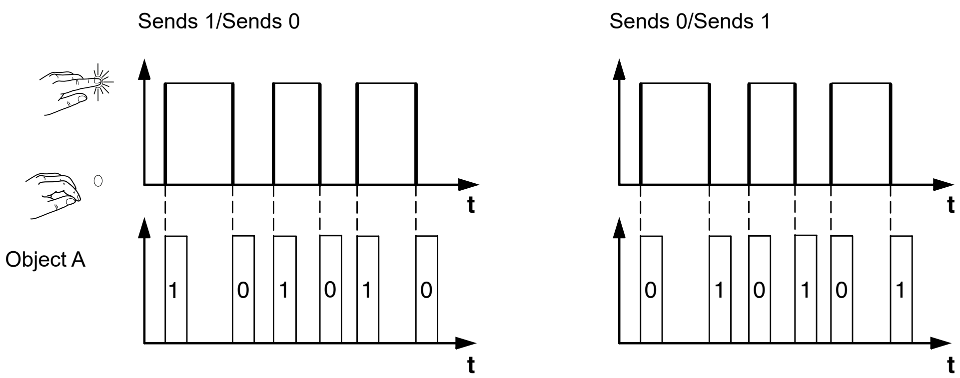
The following diagrams show how the edge function behaves when a button is pressed and released.

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

Example 1 Object A = 1 bit



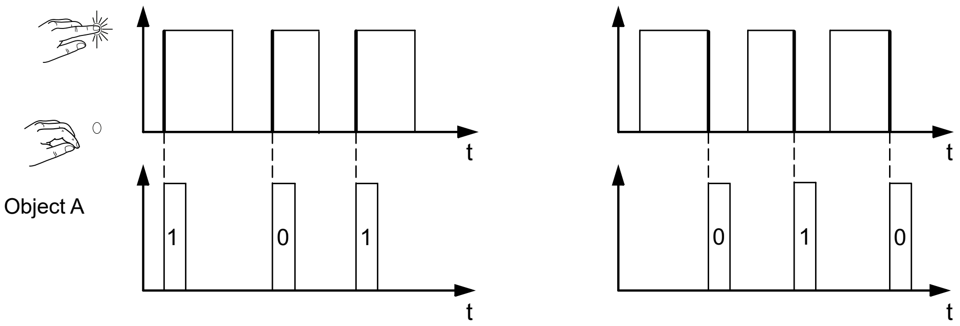
Example 2 Object A = 1 bit



Example 3

Object A = 1 bit

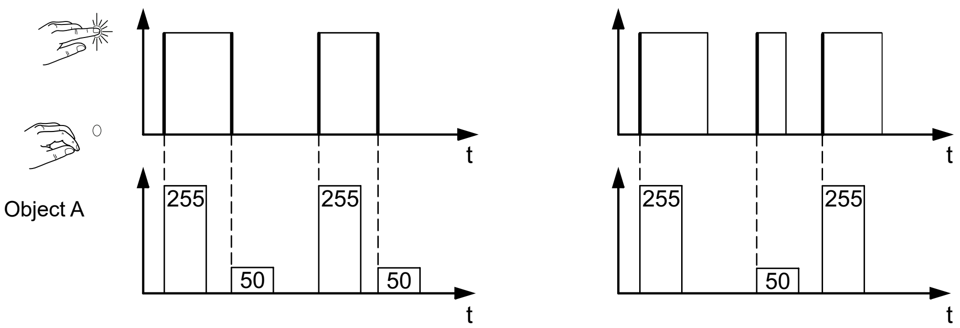
Toggles/None | None/Toggles



Example 4

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

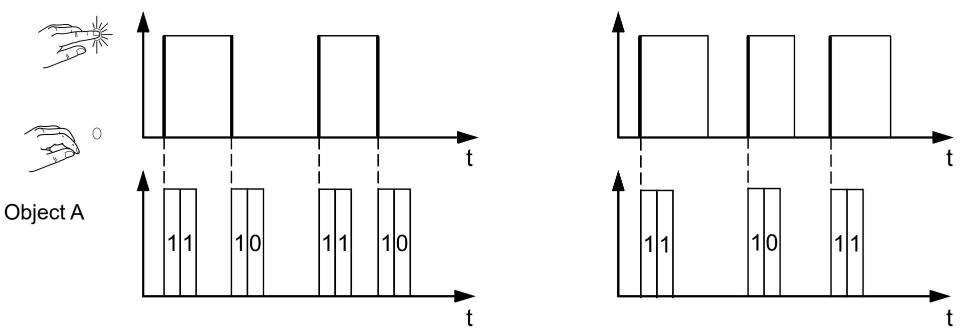
Sends value 1/Sends value 2 | Toggles/None



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



Deadman control | Toggle with priority

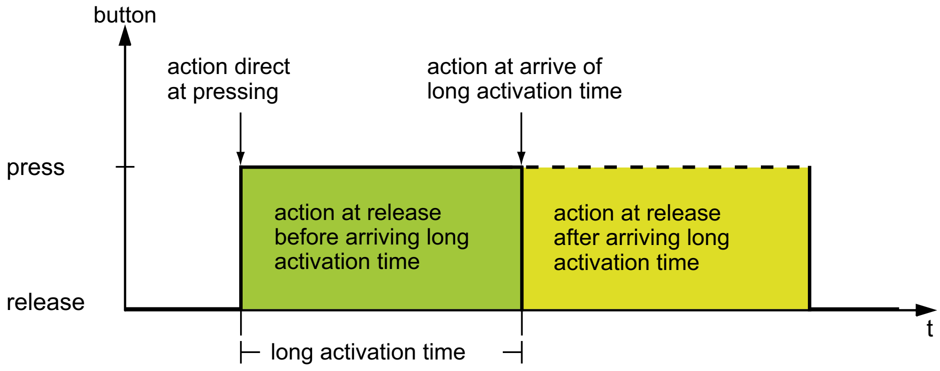
Extended edge function

With the extended edge function, a 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.

Function X – Edges		
Edges 1 bit, 2 bit, 4 bit, 1 byte	Extended settings for edges	✓
Edges object A/B	Type of object A/B	1 bit 2 bit (priority control) 4 bit 1 byte (0 – 100%) 1 byte (0 – 255)
	Direct action on operation	See the Actions for extended edge function → 35
	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 1/2	Switch on with priority (11) Switch off with priority (10) Remove priority control (00) or Dim-darker-stop To min. brightness 1/2 darker 1/4 darker 1/8 darker 1/16 darker 1/32 darker 1/64 darker Dim-brighter-stop To max. brightness 1/2 brighter 1/4 brighter 1/8 brighter 1/16 brighter 1/32 brighter 1/64 brighter or 100% 90% 80% 75% 70% 60% 50% 40% 30% 25% 20% 10% 0% or 0..255
	Cycle time	Basis * factor

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 → 38.

- 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 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. 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: *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.

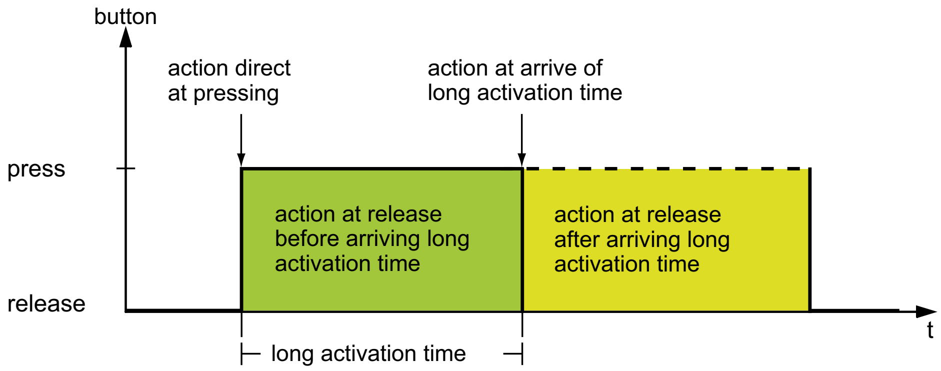
Value: *No change*
The current action is retained and any active cyclical sending is retained. You select this action for the release if, for example, you have activated the action *Sends value 1 and then value 2 after a cycle time*.

Parameters per object

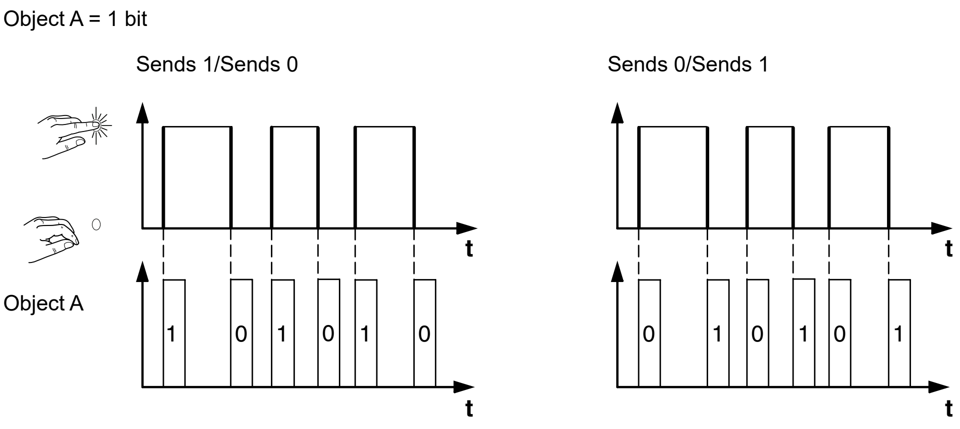
Application examples for edge function

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



	Operate button
	Release button
Object A	Sends telegrams
Sends 1/ Sends 0	Action on operation/action on release

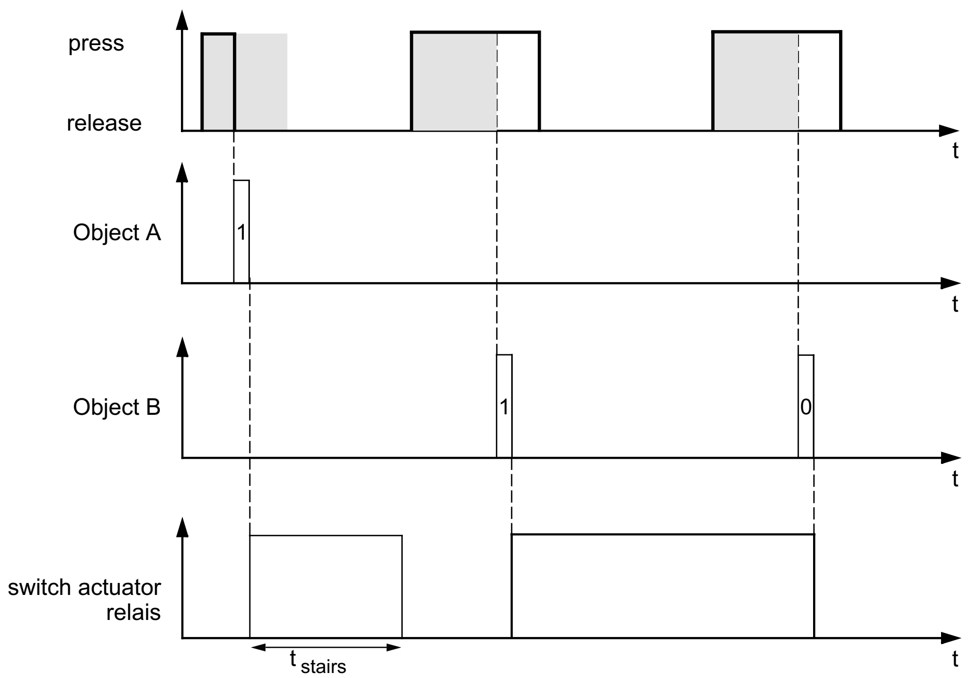
Application example for staircase cleaning

Staircase lighting function with cleaning light function

With a brief button action, the switch actuator switches on the light. A long button 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.



Application example for setting the staircase lighting time

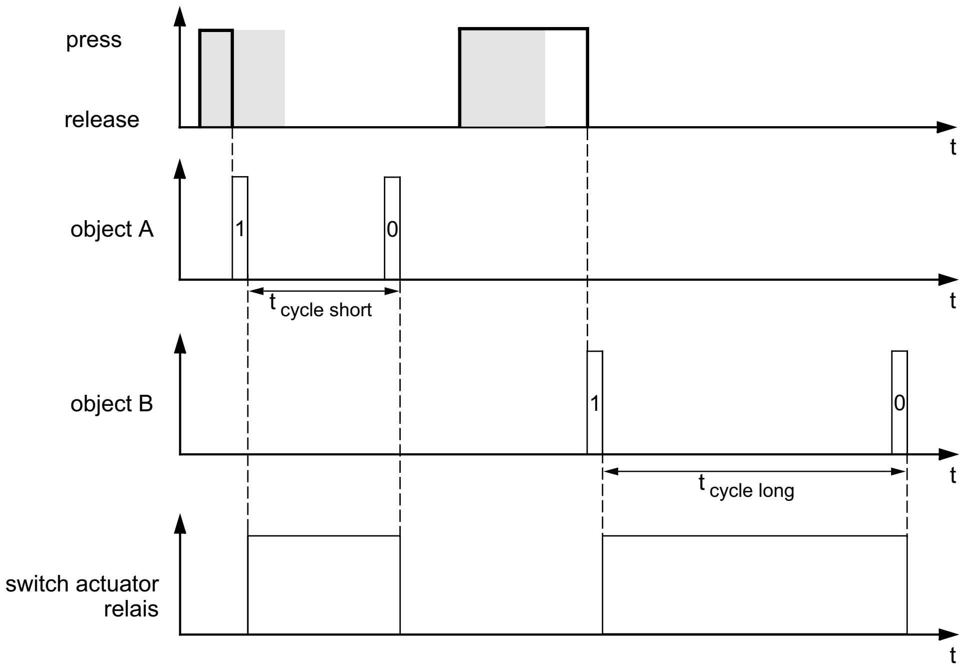
Short and long staircase lighting time

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.



Application example for switching defined light

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.

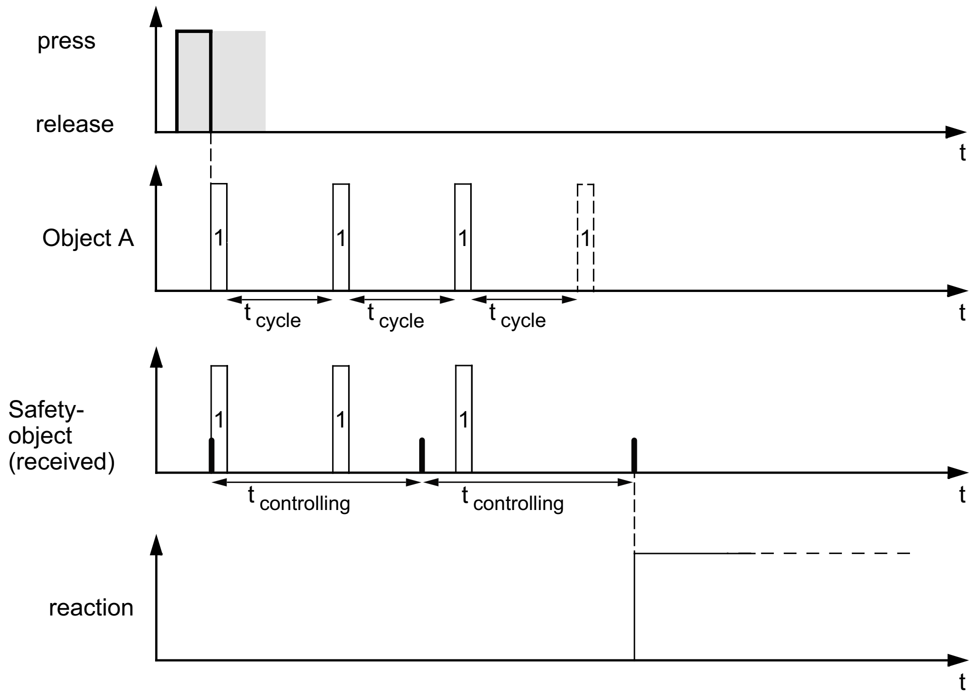
Application example for setting up theft protection

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).



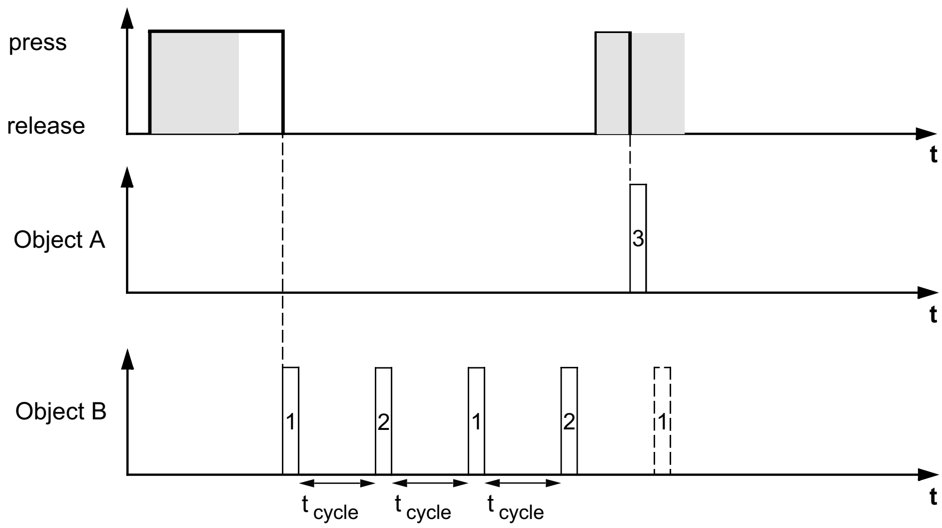
Application example for effect lighting

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 A = On/Off*.

How the status indication is triggered

Trigger	Note
Switch/value object A/B = On/Off	If the object is 1 byte, the value > 0 = the state is On, and the value 0 = Off.
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object value 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.

How the status indication is triggered

Trigger	Note
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

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.

5.6 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
- Additionally, actions for long and short operation (extended edge function)
- Additionally, send cyclically and with delay (extended edge function)
- Trigger status indication

Function X – Edges	*Express settings for edges 2-byte*	
Edges 2-byte	Name of the channel	*12 bytes allowed*
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1
	Behavior indication when locking	No Display lock icon
	Action on operation Action on release	Send value 1 Send value 2 Sends its value None
	How the status indication is trigerred	Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Always on = Off Always on = On
	Extended settings for edges 2-byte	

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

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

Edges 2-byte Edges values	Type of object	Floating point Integer with sign (- 32768…32767) Integer without sign (0 – 65535)
[Floating point]	Basis 1/2 (possible values in brackets)	0,01...32768
	Factor 1/2 (0-2047)	0 – 20,47
[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.

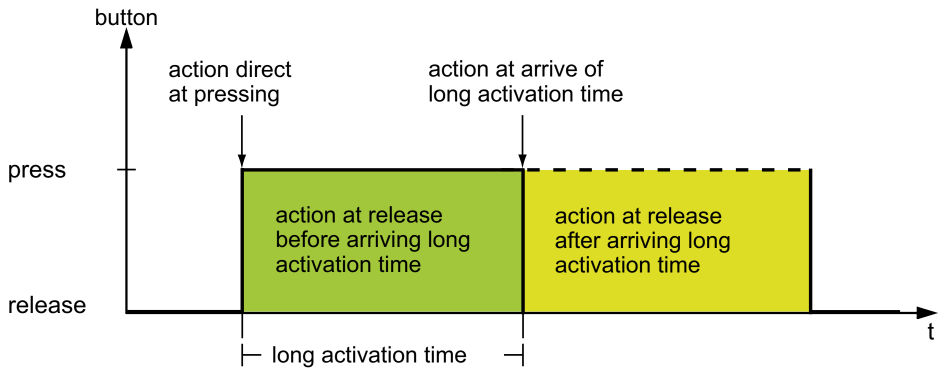
Value: *None*
No action is carried out

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.

Function X – Edges	Extended settings for edges	✓
Edges 2byte	Time for long operation	4 – 250, unit = 100 ms
Edges object A/B	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	Sends value 1 Sends value 1 immediately and then cyclically Sends value 1 only cyclically Sets object value to value 1 (readable only) Sends value 2 Sends value 2 immediately and then cyclically Sends value 2 only cyclically Sets object value to value 2 (readable only) Sends its value Sends value 1 and then value 2 after a cycle time None (stops cyclical sending) No change
	Cycle time	Basis * factor

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.

- 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 and releasing the button.

How the status indication is triggered

Trigger	Note
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object value 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the backlight color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	

	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

5.7 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 the button 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.

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

Parameter

You can select the following functions.

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

Function X – 8-bit slider	*Settings for 8-bit slider*		
	Name of the channel	*12 bytes allowed*	
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1	
	Behavior indication when locking	No Display lock icon	
	Time for long operation	4 – 250, unit = 100 ms	
	Slider function	With limit values	Without limit values
	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	Send value 1, then increase cyclic. by step width Send value 2, then reduce cyclically by step width Increase current object value cyclically Increase current object value once Reduce current object value cyclically Reduce current object value once Reverse slide direction and send cyclically Stepwise to the limit values and back again Increase stepwise within limits Decrease stepwise within limits None (stops cyclical sending) No change	Increase current object value cyclically Increase current object value once Reduce current object value cyclically Reduce current object value once Reverse slide direction and send cyclically None (stops cyclical sending) No change
	Limit value 1	0 – 255	
	Limit value 2	0 – 255	
	Value of step width	1 – 255	
	Cycle time	Basis × factor	
	How the status indication is triggered?		

Locking function

In the *Express settings*, you can name the channel and set the button lock.

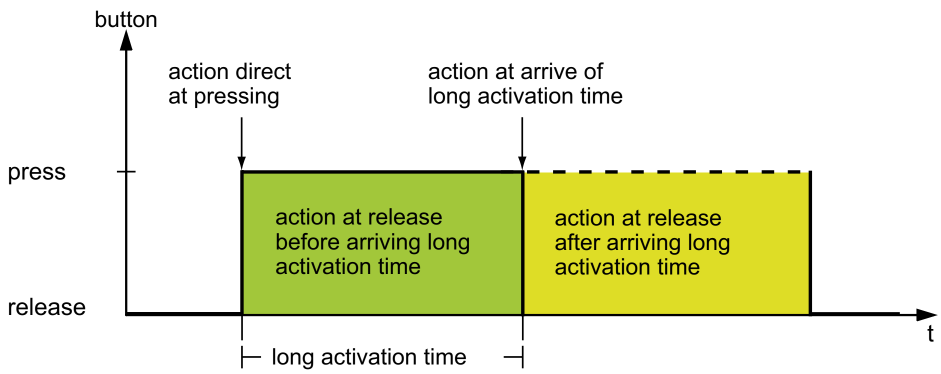
The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

Setting the slider

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 for Slider

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 the *Switch/value object A = On/Off*.

How the status indication is triggered

Trigger	Note
Switch/value object A = On/Off	If the object is 1 byte, the value > 0 = the state is On, and the value 0 = Off.
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object value 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Long operation = On / Release = Off	The long operation state = On Release button state (or short operation) = Off
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

5.8 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.

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.

Function X – Scene	*Express settings for scene*	
Scene	Name of the channel	*12 bytes allowed*
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1
	Behavior indication when locking	No Display lock icon
	Scene address	0 – 63
	How the status indication is rigerred	Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Long operation = On/Release = Off Always on = Off Always on = On
	Extended settings for scene	

Settings

In the express settings for scene, 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 are assigned automatically. You can adjust the duration of a long button action. The default setting is a duration of 600 ms.

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

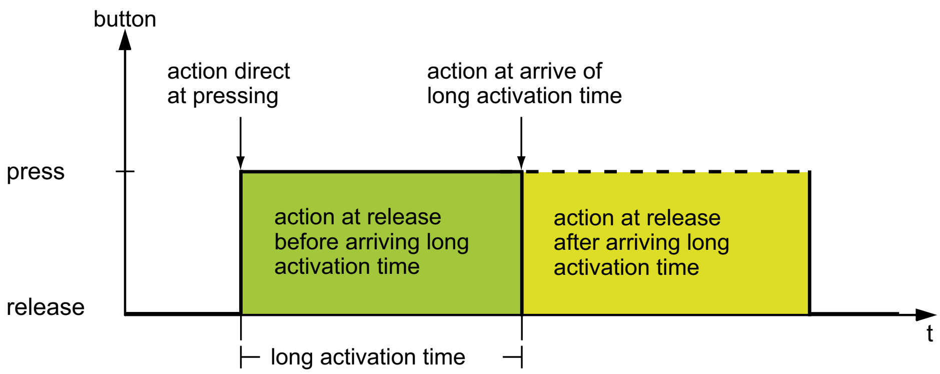
Extended scene function

With the extended scene function, an even wider range of functions is available. You can adjust the duration of a long button action. The default setting is a duration of 600 ms.

Function X – Scene	Extended settings for edges	✓
	Time for long operation	4 – 250, unit = 100 ms
	Number of objects	1 2
Scene object A/B	Direct action on operation	
	Action on release before the long operating time has elapsed	Sends value 1 Sends value 2 Toggles Toggles, sends immediately, then cyclically Sends value 1 and then value 2 after a cycle time None (stops cyclical sending) No change
	Action on achieving the long operating time	
	Action on release after achieving the long operating time	
	Value 1/2 Scene address (0 – 63)	0 – 63
	Value 1/2 to retrieve/save the scene	Retrieve Save
	Cycle time	Basis × factor

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:

Status indication

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

How the status indication is triggered

Trigger	Note
Switch/value object A = On/Off	If the object is 1 byte, the value > 0 = the state is On, and the value 0 = Off.
Switch/value object B = On/Off	If the object is 1 byte, the value > 0 = the state is On, and the value 0 = Off.
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object value 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.

How the status indication is triggered

Trigger	Note
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Long operation = On / Release = Off	The long operation state = On Release button state (or short operation) = Off
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

See Overview of group objects → 74.

5.9 RGB lighting

The *RGB lighting* function allows you to set a specific RGB or RGBW LED color. In ETS, you set the RGB/W value and connect it to a specific button.
The user calls up the set lighting color by pressing the button.

You can change the following settings:
- Type of object value
- Distinction between long and short operation
- RGB/W value
- Tunable white color temperature control

Function 1 – RGB lighting	*Settings for RGB lighting*	
RGB lighting	Name of the channel	*12 bytes allowed*

Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1	
Behavior indication when locking	No Display lock icon	
RGB strip type	RGB	RGBW
Object type	1 × 3 byte 3 × 1 byte	1 × 6 byte 4 × 1 byte
Number of operations	1 (short operation)	2 (short/long oper-ation)
Time for long operation	4 – 250, unit = 100 ms	
Send value (short operation)		
RGB value 1		
White value 1		
Send value (long operation)		
RGB value 1		
White value 1		
How the status indication is trigerred	Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Long operation = On/Release = Off Always on = Off Always on = On	

You can dim each color with separate bytes or you can dim all colors together through one group object.

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

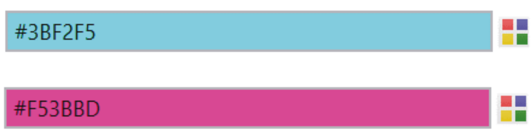
RGB color codes

RGB

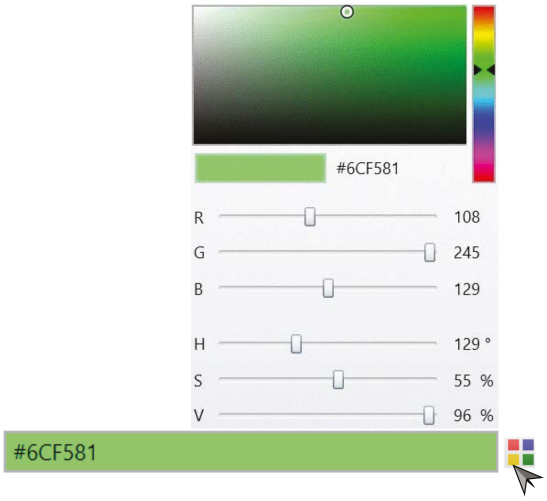
On each color scale (Red, Green, Blue) it is possible to select 256 levels or shades of the appropriate color (from 0 to 255). By mixing them together you can generate over 16 milion different color combinations.

RGB value

If you know the specific **color code**, you can simply write it directly in the *RGB value* box. The box is immediately colored according to the entered value.



The second option is to click on the color window icon at the bottom right and use the cursor to select a color on the **color scale**.



RGB colors

The following table shows the values for the basic RGB colors.

Red	Green	Blue	Colour
255	0	255	Pink
255	0	0	Red
255	127	0	Orange
255	255	0	Yellow
127	255	0	Green-yellow
0	255	0	Green
0	255	255	Blue-green
0	0	255	Blue
255	255	255	White

Note that the colours may be displayed differently, depending on the colour control device and the lamps being used.

Short and long operation

You can set the distinction between short and long operation and choose the response values for long and short press.

Status indication

How the status indication is triggered

Trigger	Note
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.

How the status indication is triggered

Trigger	Note
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Long operation = On / Release = Off	The long operation state = On Release button state (or short operation) = Off
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

You can set each output individually to 1 byte value for each color or you can choose the option where all colors are contained in one group object sent to the bus.

See Overview of group objects → 74.

5.10 Colour temperature control

With the *Colour temperature control* function, you can send the preset percentage of brightness and the value of color temperature expressed in Kelvins (K).

Colour temperature values

You can select the minimum and the maximum colour temperature values. The usable value range depends on the control device and the lamps. The control device has to support the 2-byte data point type 7.600.

Function X – Colour temperature	*Settings for colour temperature control*

Colour temperature	Name of the channel	*12 bytes allowed*	
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1	
	Behavior indication when locking	No Display lock icon	
	Number of operations	1 (short operation)	2 (short/long operation)
	Time for long operation	4 – 250, unit = 100 ms	
	Send value (short operation)		
	Brightness value 1		
	Colour temperature value 1		
	Send value (long operation)		
	Brightness value 1		
	Colour temperature value 1		
	How the status indication is trigerred	Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Always on = Off Always on = On	

With lighting, the colour temperatures differ according to the type of source:
- Candle light: 1500K
- Incandescent bulb: 2600K – 2700K
- Halogen bulb: 2700K – 3000K
- Fluorescent light: 4500K – 6500K
- Daylight: 5500K – 7000K

The higher the number of Kelvin, the cooler white the light is. The lower the number of Kelvin is the warmer and more yellower the light is.

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

Short and long operation

You can set the distinction between short and long operation and choose the response values for long and short press.

Status indication

How the status indication is triggered

Trigger	Note
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object value 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Long operation = On / Release = Off	The long operation state = On Release button state (or short operation) = Off
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

A value for brightness and color temperature are transmitted via the value objects.

See Overview of group objects → 74.

5.11 Temperature decrease/increase

The *Temperature decrease* function allows you to change the temperature setpoint value. When you assign this function to the button and press it, the setpoint temperature displays in the middle field if you select *Temperature only* or *Temperature + text* in General settings (Middle field display → 13).

The *Temperature increase* function is the same as the Temperature decrease function. The only difference is that it is used for increasing the setpoint temperature value.

You can change the following settings:

Function X – Temperature decrease/increase	*Settings for temperature decrease/increase*	
Temperature decrease/increase	Name of the channel	*12 bytes allowed*
	Locking function	Disable Locking = 1/Unlocking = 0 Locking = 0/Unlocking = 1
	Object type	Temperature setpoint (DPT 9.001) Temperature shift (DPT 9.002)
	Temperature adjust step	0,1 – 0,5 K
Temperature decrease	Minimum setpoint	10 – 32, unit = 1°C
Temperature decrease	Minimum setpoint shift	- 10 – 0, unit = 1 K
Temperature increase	Maximum setpoint	10 – 32, unit = 1 °C
Temperature increase	Maximum setpoint shift	0 – 10, unit = 1 K
	How the status indication is trigerred	Status feedback object 1 bit Status feedback object value 1 byte Operation = On/Release = Off Operation = Off/Release = On Always on = Off Always on = On

Locking function

In the *Express settings*, you can name the channel and set the button lock.

The button is locked via the 1-bit *Button locking object* with values 1 (lock) and 0 (unlock). You can reverse value sending: 0 (lock) and 1 (unlock).

The locked button can be indicated by a lock icon or not at all.
If you select *Disable*, the locking function becomes inactive.

Object type

You can send the values via the following objects to the bus by pressing a button:

Object type	Adjustment options	Devices
Temperature setpoint (DPT 9.001)	Temperature setpoint	MTN6730-0002 or most of the temperature control panels.
	Absolute setpoint value	
	Relative setpoint value	
Temperature shift (DPT 9.002)	Reference temperature difference	MTN6215-5910
	2 bytes	MTN6730-0003

To display the temperature setpoint value in the middle field, you need to set the Setpoint input of the middle field display in the same group with temperature setting object of the valve.

Status indication

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

How the status indication is triggered

Trigger	Note
Status feedback object 1 bit	1 = On 0 = Off according to the value of the 1-bit external feedback object.
Status feedback object value 1 byte	If the feedback value of the external object is 1 byte, the value > 0 = On, and the value 0 = Off.
Operation = On / Release = Off	The pressed button state = On Release button state = Off Operation = Off/Release = On: The pressed button state = Off, and the release button state = On.
Always on = Off/On:	The icon is always Off/On.

Icon configuration

In the *Icon configuration* section, you can set the type of indication on your screen. You can also choose the icons for status On and Off of each function and the back-light color of icons.

Function X - Name of the function	*Icon configuration*	
Name of the function	Indication type	Icon only Icon + Name of the channel No icon
	Icon preview	
	Icon for status on	
	Color for status on	White Green Blue Red Orange Yellow
	Icon preview	
	Icon for status off	
	Color for status off	White Green Blue Red Orange Yellow

Group objects

There are two objects for this function:

- Setpoint output object - pressing the button decreases/increases the setpoint temperature value of the setpoint output object.
- Setpoint input object - receives temperature setpoint value from the bus after sending out the write request.

See Overview of group objects → 74.

6 Logic function

In complex KNX installations, the logic function serves to establish special logic operations between sensors and actuators. There is a wide range of possible settings for executing numerous logic functions for controlled KNX devices (e.g. dimming or switch actuators, various sensors etc).

The logic function is particularly suitable for summarising messages (e.g. the lighting status in rooms), linking conditions (e.g. rain or wind sensor activates a safety function) or programming an additional toggle between manual and automatic (e.g. disabling brightness-dependent lighting control for a video presentation).
Due to the large number of possible settings, the logic module is particularly well suited to the areas of security, comfort or energy saving.

The outputs can also be shown on the visualization device.
By default, all 8 possible logic functions/blocks are deactivated. You have to enable the required amount of the functions.

Logic functions 1st Logic	1st Logic function	✓

You can choose from one of the following logic operations for each logic block.

1st Logic	Function of channel	AND OR XOR Threshold comparator Format convert

Always set all parameters on the first block before parameterising the next block.

Never connect the output and the input of the same logic block to one another, as this can cause the device to malfunction.

6.1 AND, OR, XOR

The operations have either the value 1 or 0. The input and output can also be inverted.

AND

The logic AND operation output is only **true** when **all of its inputs are true**, otherwise the output is false.

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

OR

The logic OR operation output is only **true** if **one or more of its inputs** are true, otherwise the output is false.

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

XOR

The logic exclusive-OR or XOR operation gives a true output when the number of true inputs is odd.

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

The difference between OR and XOR

The difference between the OR and XOR logic operations is that the output from the XOR gate is logical 1 if and only if there is an unequal number of 1 and 0 inputs.

In the simple case of an XOR operation with two inputs, this means that the inputs must be different to one another to obtain the output 1. 1 must be present at precisely one of the two inputs.

A	B	OR	XOR
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

In contrast to a simple OR logic operation, the condition is deemed not to be met if a 1 is present at both inputs.

With an XOR operation, the result in this case is a 0. Each additional input at the gate alters the behaviour accordingly

A	B	C	OR	XOR
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Setting

The first block of functions is described together, because all three operations have the same parameters and values.

Input behaviour

You can use up to 8 inputs (a - h). By default, all the inputs are disconnected.

The input telegrams can be inverted for each input. In addition, a fixed value (0 or 1) can be assigned.

1st Logic	Function of channel	AND
	Input a-h	Disconnected Normal Inverted
	Default value	0 1

Output behaviour

Criteria for the sending behaviour at the output can be defined.

1st Logic	Result is inverted	No/Yes
	Read input object value after bus voltage recovery	No/Yes
	Output send when	Receiving a new telegram (on the input)
		Every change of output object
	Send delay time = basis x factor	6375 s => basis (0-25s) * factor (0-255)
	Basis	
	Factor	

If you click *Yes* for *Read input object value after bus voltage recovery*, the logic module sends a read telegram to all inputs asking about their values.

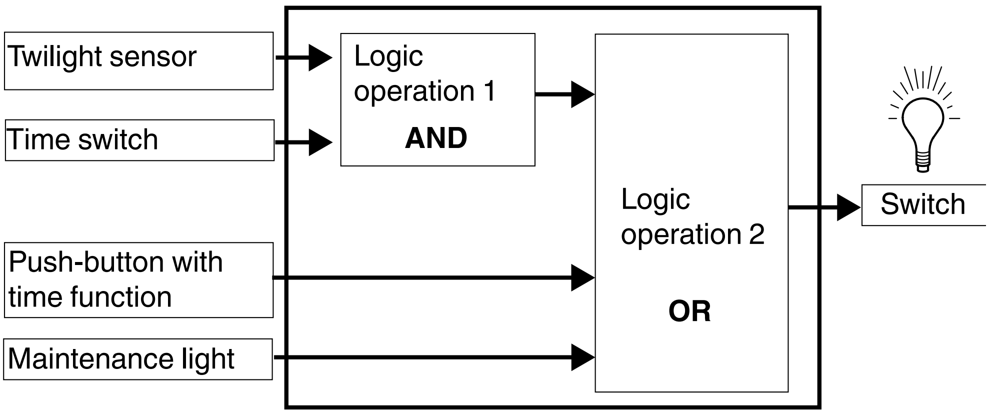
If one or more inputs do not respond, the bus keeps on trying to collect missing responses.

Output send when option allows you to set wheter the output should be sent after receiving a new telegram at the input or at every change of the output object.

This setting is wise if a rapid response is expected (e.g. weather alarm at the blind actuator). This function also helps to prevent bus overload.

Example

- A light-sensitive switch switches the lighting on automatically.
- The light is switched off between 23:00 and 06:00.
- In the morning, the light switches on from 06:00 when it is dark.
- In addition, the light can be switched on for 5 minutes at any time via a push-button.
- A continuous light function is possible for maintenance purposes.



6.2 Threshold comparator

Threshold comparator compares the input value with the threshold.

1st Logic	Function of channel	Threshold comparator
	Threshold value data type	4 bit, 1/2/4 byte
	Threshold value	0 – 255
	If Object value < Threshold value	
	If Object value = Threshold value	
	If Object value != Threshold value	Do not send telegram/Send value 1/0
	If Object value > Threshold value	
	If Object value ≤ Threshold value	
	If Object value ≥ Threshold value	

You can set a threshold, select its comparison type and choose which value to send after comparison:

- 0
- 1
- Do not send telegram

Output send when option allows you to set wheter the output should be sent after receiving a new telegram at the input or at every change of the output object.

This setting is wise if a rapid response is expected. It also helps to prevent bus overload.

6.3 Format convert

The format converter allows you to decompose or combine different data types. It is typically used when a sender and receiver do not support the same data format or when you need to solve special requirements.

1st Logic	Function of channel	Format convert
	Function	2 × 1 Bit → 1 × 2 Bit
		8 × 1 Bit → 1 × 1 Byte
		1 × 1 Byte → 1 × 2 Byte
		2 × 1 Byte → 1 × 2 Byte
		2 × 2 Byte → 1 × 4 Byte
		1 × 1 Byte → 8 × 1 Bit
		1 × 2 Byte → 2 × 1 Byte
		1 × 4 Byte → 2 × 2 Byte
		1 × 3 Byte → 3 × 1 Byte
		3 × 1 Byte → 1 × 3 Byte

Basic application

1 × 1 byte → 8 × 1 bit: This function can be used to decompose bit-oriented information sent as 1 byte to individual bits, for example:

- Controller status of room temperature controllers
- Failure status of DALI groups and ECGs

1 × 3 byte → 3 × 1 byte

Converts RGB 3 byte combined value to three separate 1 byte values for red, green and blue.

3 × 1 byte → 1 × 3 byte

Combines three 1 byte values (red, green, blue) to one RGB 3 byte combined value.

Group objects

See Overview of group objects → 74.

7 Bus voltage

7.1 Behaviour when bus voltage is connected/recovered

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

7.2 Behaviour when bus voltage fails

Any status LEDs that were lit are switched off. Group objects are not saved except the locking object if the locking function is enabled. The locking object will be recalled after power cycling.

7.3 Configuration mode

After a download, the device goes into configuration mode for a few seconds. During this time, the LED flashes.

8 Object display structure

Main channel	Object rang	Note
	125-131	Extended settings Channel associated objects
Function 1 - Toggle/...	73-78	The object and channel names {{0}} vary with the parameter. The description and channels display according to function.
Function 2 - Toggle/...	79-84	
Function 3 - Toggle/...	85-90	
Function 4 - Toggle/...	91-96	
Function 5 - Toggle/...	97-102	
Function 6 - Toggle/...	103-108	
Function 7 - Toggle/...	109-114	
Function 8 - Toggle/...	115-120	
Middle field display	121-124	
1st Logic	1-9	Objects 1 to 8 display independently from the channel.
2nd Logic	10-18	
3rd Logic	19-27	
4th Logic	28-36	
5th Logic	37-45	
6th Logic	46-54	
7th Logic	55-63	
8th Logic	64-72	
Auxiliary functions	133-137	

9 Overview of group objects

This list provides the numbers for uniquely identifying a group object. The data point types (DPT) in this application are preset.

Logic

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
1	1th Logic	Input a	1bit	C,W,T,U	visible if not disconnected	AND, OR, XOR	1.002 boolean
2	1th Logic	Input b	1bit	C,W,T,U			
3	1th Logic	Input c	1bit	C,W,T,U			
4	1th Logic	Input d	1bit	C,W,T,U			
5	1th Logic	Input e	1bit	C,W,T,U			
6	1th Logic	Input f	1bit	C,W,T,U			
7	1th Logic	Input g	1bit	C,W,T,U			
8	1th Logic	Input h	1bit	C,W,T,U			
9	1th Logic	Logic result	1bit	C,T			
1	1th Logic	Threshold value input	4bit 1byte 2byte 4byte	C,W,U	Length according to parameter select	Threshold comparator	3.007 dimming 5.010 counter pulses 7.001 pulses 12.001 counter pulses
9	1th Logic	Logic result	1bit	C,T			1.002 boolean
1	1th Logic	Input 1bit-bit0	1bit	C,W,U	2 x 1bit → 1 x 2bit	Format convert	1.002 boolean
2	1th Logic	Input 1bit-bit1	1bit	C,W,U			1.002 boolean
9	1th Logic	Output 2bit	2bit	C,T			2.001 switch control
1	1th Logic	Input 1bit-bit0	1bit	C,W,U	8 x 1bit → 1 x 1byte		1.002 boolean
2	1th Logic	Input 1bit-bit1	1bit	C,W,U			
3	1th Logic	Input 1bit-bit2	1bit	C,W,U			
4	1th Logic	Input 1bit-bit3	1bit	C,W,U			
5	1th Logic	Input 1bit-bit4	1bit	C,W,U			
6	1th Logic	Input 1bit-bit5	1bit	C,W,U			
7	1th Logic	Input 1bit-bit6	1bit	C,W,U			
8	1th Logic	Input 1bit-bit7	1bit	C,W,U			
9	1th Logic	Output 1byte	1byte	C,T			5.010 counter pulses
1	1th Logic	Input 1byte	1byte	C,W,U	1 x 1byte → 1 x 2byte		5.010 counter pulses
9	1th Logic	Output 2byte	2byte	C,T			7.001 pulses
1	1th Logic	Input 1byte-low	1byte	C,W,U	2 x1byte → 1 x 2byte		5.010 counter pulses
2	1th Logic	Input 1byte-high	1byte	C,W,U			5.010 counter pulses
9	1th Logic	Output 2byte	2byte	C,T			7.001 pulses
1	1th Logic	Input 2byte-low	2byte	C,W,U	2 x 2byte → 1 x 4byte		7.001 pulses
2	1th Logic	Input 2byte-high	2byte	C,W,U			
9	1th Logic	Output 4byte	4byte	C,T			12.001 counter pulses
1	1th Logic	Input 1byte	1byte	C,W,U	1 x 1byte → 8 x 1bit		5.010 counter pulses
2	1th Logic	Output 1bit-bit0	1bit	C,T			1.002 boolean
3	1th Logic	Output 1bit-bit1	1bit	C,T			
4	1th Logic	Output 1bit-bit2	1bit	C,T			
5	1th Logic	Output 1bit-bit3	1bit	C,T			
6	1th Logic	Output 1bit-bit4	1bit	C,T			
7	1th Logic	Output 1bit-bit5	1bit	C,T			
8	1th Logic	Output 1bit-bit6	1bit	C,T			
9	1th Logic	Output 1bit-bit7	1bit	C,T			

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
1	1th Logic	Input 2byte	2byte	C,W,U	1 x 2byte → 2 x 1byte	Format convert	7.001 pulses
8	1th Logic	Output 1byte-low	1byte	C,T			5.010 counter pulses
9	1th Logic	Output 1byte-high	1byte	C,T			
1	1th Logic	Input 4byte	4byte	C,W,U	1 x 4byte → 2 x 2byte		12.001 counter pulses
8	1th Logic	Output 2byte-low	2byte	C,T			7.001 pulses
9	1th Logic	Output 2byte-high	2byte	C,T			
1	1th Logic	Input 3byte	3byte	C,W,U	1 x 3byte → 3 x 1byte		232.600 RGB value 3 x (0..255)
7	1th Logic	Output 1byte-low	1byte	C,T			5.010 counter pulses
8	1th Logic	Output 1byte-middle	1byte	C,T			
9	1th Logic	Output 1byte-high	1byte	C,T			
1	1th Logic	Input 1byte-low	1byte	C,W,U	3 x 1byte → 1 x 3byte		5.010 counter pulses
2	1th Logic	Input 1byte-middle	1byte	C,W,U			
3	1th Logic	Input 1byte-high	1byte	C,W,U			
9	1th Logic	Output 3byte	3byte	C,T			232.600 RGB value 3 x (0..255)
10 – 18	2nd Logic						
19 – 27	3rd Logic						
28 – 36	4th Logic						
37 – 45	5th Logic						
46 – 54	6th Logic						
55 – 63	7th Logic						
64 – 72	8th Logic						

Functions

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
73	Function 1 The name of the object refers to the parameter *Name of the channel* of the function. The name of the function changes with the parameter description. If the parameter description is empty, Function 1 displays by default.	Switch object A	1bit	C,T	Switch values are sent alternately during operation.	Toggle: Extended settings disable	1.001 switch
75		Switch status object A	1bit	C,W			
73		Switch object A, Value object A	1bit/ 1byte	C,T	visible according to object type	Toggle: Extended settings	1.001 switch, 5.001 percentage (0..100%), 5.010 counter pulses
74		Switch object B, Value object B	1bit/ 1byte	C,T	visible according to object type&2 objects		
75		Switch status object A, Value status object A	1bit/ 1byte	C,W	visible according to object type		
76		Switch status object B, Value status object B	1bit/ 1byte	C,W	visible according to object type&2 objects		

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
73		Switch object A	1bit	C,T	Switching: Extended settings disable		1.001 switch
73		Switch object A, Value object A	1bit, 1byte	C,T	Visible according to object type	Switching: Extended settings	1.001 switch, 5.001 percent-age(0..100%), 5.010 counter pulses
74		Switch object B, Value object B	1bit, 1byte	C,T	Visible according to object type and objects		
73		Switch object	1bit	C,T		Dimming	1.001 switch
74		Dimming object	4bit	C,T			3.007 dimming
75		Switch status object	1bit	C,W	Only visible if you select Dimming brighter/darker		1.001 switch
73		Movement object	1bit	C,T	Moving blind up, Moving blind down, Moving blind up/ down	Blind	1.008 up/down
74		Stop/step	1bit	C,T			1.007 step
75		Last direction	1bit	C,W	Only visible if you select Moving blind up/down	Blind Extended settings	1.008 up/down
73		Blind position	1byte	C,T	Moving blind with positioning values		5.001 percentage (0..100%), 5.010 counter pulses
74	Function 1 The name of the object refers to the parameter *Name of the channel* of the function. The name of the func-tion changes with the parameter description. If the parameter descrip-tion is empty, Function 1 displays by default.	Slat position	1byte	C,T			
73		Switch object A, Priority control object A, Dim-ming object A, Value object A	1bit, 2bit, 4bit, 1byte	C,T	Visible according to object type	Edges 1bit, 2bit (Priority), 4bit, 1byte values	1.001 switch, 2.001 switch control, 3.007 dimming, 5.010 counter puls-es, 5.001 percent-age (0..100%)
74		Switch object B, Priority control object B, Dim-ming object B, Value object B		C,T	visible according to object type and objects		
75		Switch status object A, Priority control status object A, Dimming status object A, Value status object A	1bit, 2bit, 4bit, 1byte	C,W	visible according to object type		1.001 switch, 2.001 switch control, 3.007 dimming, 5.010 counter puls-es, 5.001 percent-age (0..100%)
76		Switch status object B, Priority control status object B, Dimming status object B, Value status object B		C,W	visible according to object type&2 objects		
73		Value object A	2byte	C,T	visible according to object type	Edges with 2byte values	9.004 lux(Lux), 7.001 pulses, 8.001 pulses difference
75		Value status object A		C,W			
73		Value object A	1byte	C,T		8 bit slider	5.004 percentage (0..255%)
75		Set value object A		C,W			
73		Scene object A	1byte	C,T		Scene	18.001 scene control

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
74		Scene object B	1byte	C,T	Scene Extended settings		18.001 scene control
73		RGB dimming value	3byte	C,T	RGB: 3byte option	RGB lighting	232.600 RGB value 3 x (0..255)
73		RGBW dimming value	6byte	C,T	RGBW: 6byte option		251.600 DPT_Colour_RGBW
73		Red dimming value	1byte	C,T	1byte option		5.001 percentage(0..100%)
74		Green dimming value	1byte	C,T			
75		Blue dimming value	1byte	C,T			
76		White dimming value	1byte	C,T			
73		Brightness value	1byte	C,T		Colour temperature control	5.001 percentage(0..100%)
74		Colour temperature value	2byte	C,T			7.600 absolute colour temperature
73	Function 1 The name of the object refers to the parameter *Name of the channel* of the function. The name of the function changes with the parameter description. If the parameter description is empty, Function 1 displays by default.	Setpoint output	2byte	C,T	When operating the button, the temperature setpoint is lowered.	Temperature decrease	9.001 temperature (°C)
74		Setpoint input	2byte	C,W,T,U	The temperature setpoint is received from the bus and a read request is sent at power-up.		
73		Setpoint output	2byte	C,T	When operating the button, the reference temperature difference is lowered.		9.002 temperature difference (K)
74		Setpoint input	2byte	C,W,T,U	The reference temperature difference value is received from the bus and a read request is sent at power-up		
73		Setpoint output	2byte	C,T	When operating the button, the temperature setpoint is raised.	Temperature increase	9.001 temperature (°C)
74		Setpoint input	2byte	C,W,T,U	The temperature setpoint is received from the bus and a read request is sent at power-up.		
73		Setpoint output	2byte	C,T	When operating the button, the reference temperature difference is adjusted upwards.		9.002 temperature difference (K)
74		Setpoint input	2byte	C,W,T,U	The reference temperature difference value is received from the bus and a read request is sent at power-up.		

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
77		Button Locking object	1bit	C,W	Visible when the locking function is enabled. When locked, the buttons are not operational and the specific value of the object is defined by the parameter	Locking	1.003 enable
78		Status feedback object, Status feedback object value	1bit, 1byte	C,W,T,U	1bit object type visible if the option is *Status feedback object = On/Off* 1byte object type visible if the option is *Status feedback object value = On/Off*		1.001 switch 5.004 percentage (0..255%)
79 – 84	Function 2	The same as function 1					
85 – 90	Function 3						
91 – 96	Function 4						
97 – 102	Function 5						
109 – 108	Function 6						
109 – 114	Function 7						
115 – 120	Function 8						

Other objects

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
121	Middle field display line 1	Brightness level status feedback object	1byte	C,W	Visible if dimming level is enabled		5.001 percentage (0..100%)
122	Middle field display line 2		1byte	C,W			
123	Middle field display line 3		1byte	C,W			
124	Middle field display line 4		1byte	C,W			
125	Night mode	Night mode input	1bit	C,W,T,U	Visible when parameter *Night mode* is enabled		1.024 day/night
126	Proximity function	Proximity input	1bit	C,W	Visible when parameter option *How the Proximity function is triggered* including Proximity object		1.001 switch
127		Proximity output	1bit, 1byte	C,T	Visible according to the parameter option		1.001 switch, 5.010 counter pulses
128	Temperature measurement	Actual temperature 2Byte	2byte	C,R,T	Displays according to parameter option.		9.001 temperature (°C), 9.027 temperature (°F)
129	Temperature measurement	Actual temperature 4Byte	4byte	C,R,T			

Nr.	Name	Object function	Length	Properties	Note	Function description	DPT
130	General	Live signal	1bit	C,T			1.001 switch
131		Date	3byte	C,W			11.001 date
132		Time	3byte	C,W			10.001 time of day
138		External temperature sensor	2byte	C,W,T,U	Receives external temperature sensor measurements, sends periodic read requests, and sends power-ups.		9.001 temperature
139	Middle field display line 1	External actual temperature	2byte	C,W,T,U	Receives external temperature sensor measurements, sends periodic read requests, and sends power-ups.		9.001 temperature
140		Heating and Cooling mode	1bit	C,W,T,U	The heating/cooling mode is received from the bus and a read request is sent at power-up.		1.100 cooling/ heating
141		Setpoint input	2byte	C,W	Receives the bus temperature setpoint. When there is data reception, the screen immediately switches to display the set temperature. After the delay it switches to the ambient temperature.		9.001 temperature
142	Middle field display line 2	External actual temperature	2byte	C,W,T,U		The same as Middle field display line 1	9.001 temperature
143		Heating and Cooling mode	1bit	C,W,T,U			1.100 cooling/ heating
144		Setpoint input	2byte	C,W			9.001 temperature
145	Middle field display line 3	External actual temperature	2byte	C,W,T,U			9.001 temperature
146		Heating and Cooling mode	1bit	C,W,T,U			1.100 cooling/ heating
147		Setpoint input	2byte	C,W			9.001 temperature
148	Middle field display line 4	External actual temperature	2byte	C,W,T,U			9.001 temperature
149		Heating and Cooling mode	1bit	C,W,T,U			1.100 cooling/ heating
150		Setpoint input	2byte	C,W			9.001 temperature
	Auxiliary function	The following objects are hidden					

10 Index

T

U

Schneider Electric SA
35 rue Joseph Monier
92500 Rueil Malmaison - France
Phone: +33 (0) 1 41 29 70 00
Fax: +33 (0) 1 41 29 71 00

If you have technical questions, please contact the Customer Care Centre in your country.
schneider-electric.com/contact

MTN619x-6010_SW_EN 12/22