State 07/2013

Technical Manual

MDT Push Button/
Push Button Plus

BE-TA5502.01
BE-TA5504.01
BE-TA5506.01
BE-TA5508.01
BE-TA55P2.01
BE-TA55P4.01
BE-TA55P6.01
BE-TA55P8.01

Further Documents:

Datasheet:

https://www.mdt.de/EN_Downloads_Datasheets.html

Assembly and Operation Instructions:

https://www.mdt.de/EN_Downloads_Instructions.html

Solution Proposals for MDT products:

https://www.mdt.de/EN_Downloads_Solutions.html

1 Content

2 Overview

2.1 Overview devices

The manual refers to the following devices, which are in our assortment of push buttons. Actually we can offer you the following push buttons (Order Code respectively printed in bold type):

- **BE-TA5502.01** push button 2-fold
- **BE-TA5504.01** push button 4-fold
- **BE-TA5506.01** push button 6-fold
- **BE-TA5508.01** push button 8-fold
- **BE-TA55P2.01** push button 2-fold, Plus
  - 2 LED status displays, 4 Logic blocks
- **BE-TA55P4.01** push button 4-fold, Plus
  - 4 LED status displays, 4 Logic blocks
- **BE-TA55P6.01** push button 6-fold, Plus
  - 6 LED status displays, 4 Logic blocks
- **BE-TA55P8.01** push button 8-fold, Plus
  - 8 LED status displays, 4 Logic blocks

2.2 Exemplary circuit diagrams

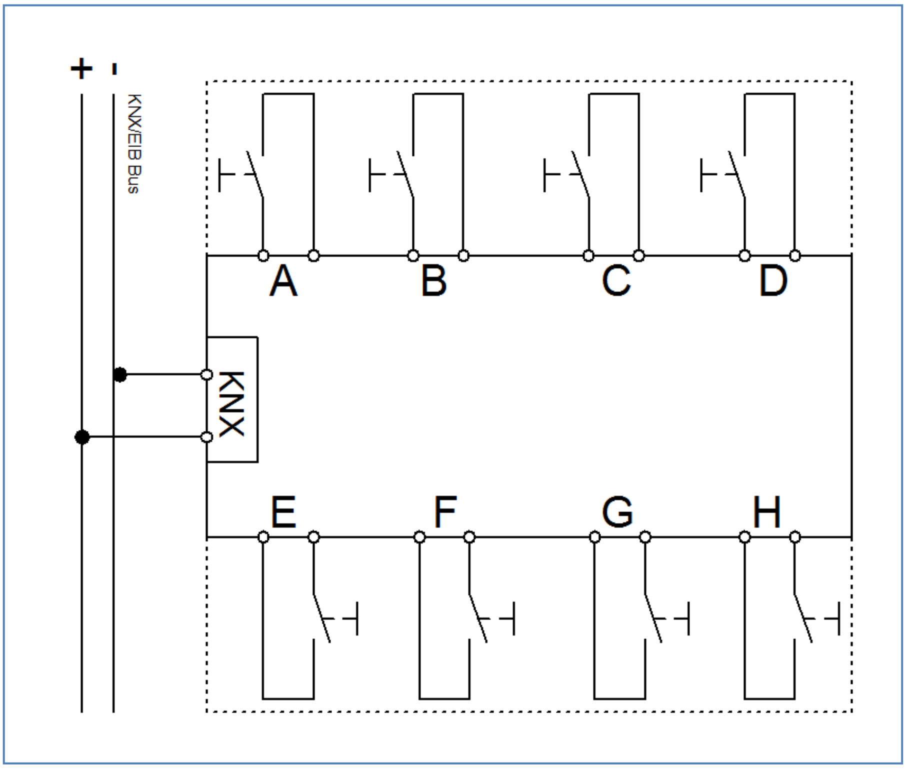


Figure 1: Exemplary circuit diagram BE-TA5508.01-Design with 8 push buttons

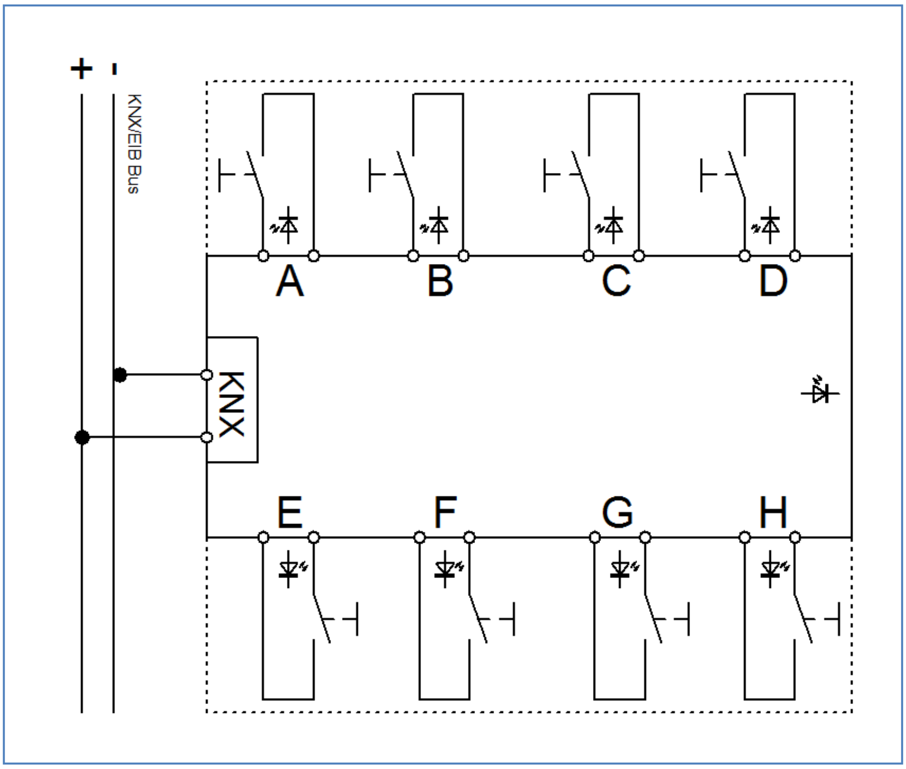


Figure 2: Illustration 2: Exemplary circuit diagram BE-TA550P8.01-Plus-design with 8 push buttons

2.2 Usage & areas of use

The push button contains of almost all of the functions of the binary input. It is designed for flush mounting. By a pushing a button the push button can call parameterized functions like dimming or call whole scenes.
The plus variant contains additional of up to 8 bicolored LEDs for the respective buttons and a bicolored orientation LED and a 4 logic blocks. The LEDS can be parameterized individually.

2.4 Structure & Handling

The push button contains, depending on the design, of 2 to 8 buttons, which can be parameterized individually. Additional LEDs exists at the plus variant. The bus can be connected at the back of the push buttons. Furthermore all push buttons contains of the standard elements programming button and programming LED at the side of the push buttons.
The Illustration shows an 8-fold push button, at the left a normal one and at the right the plus variant:

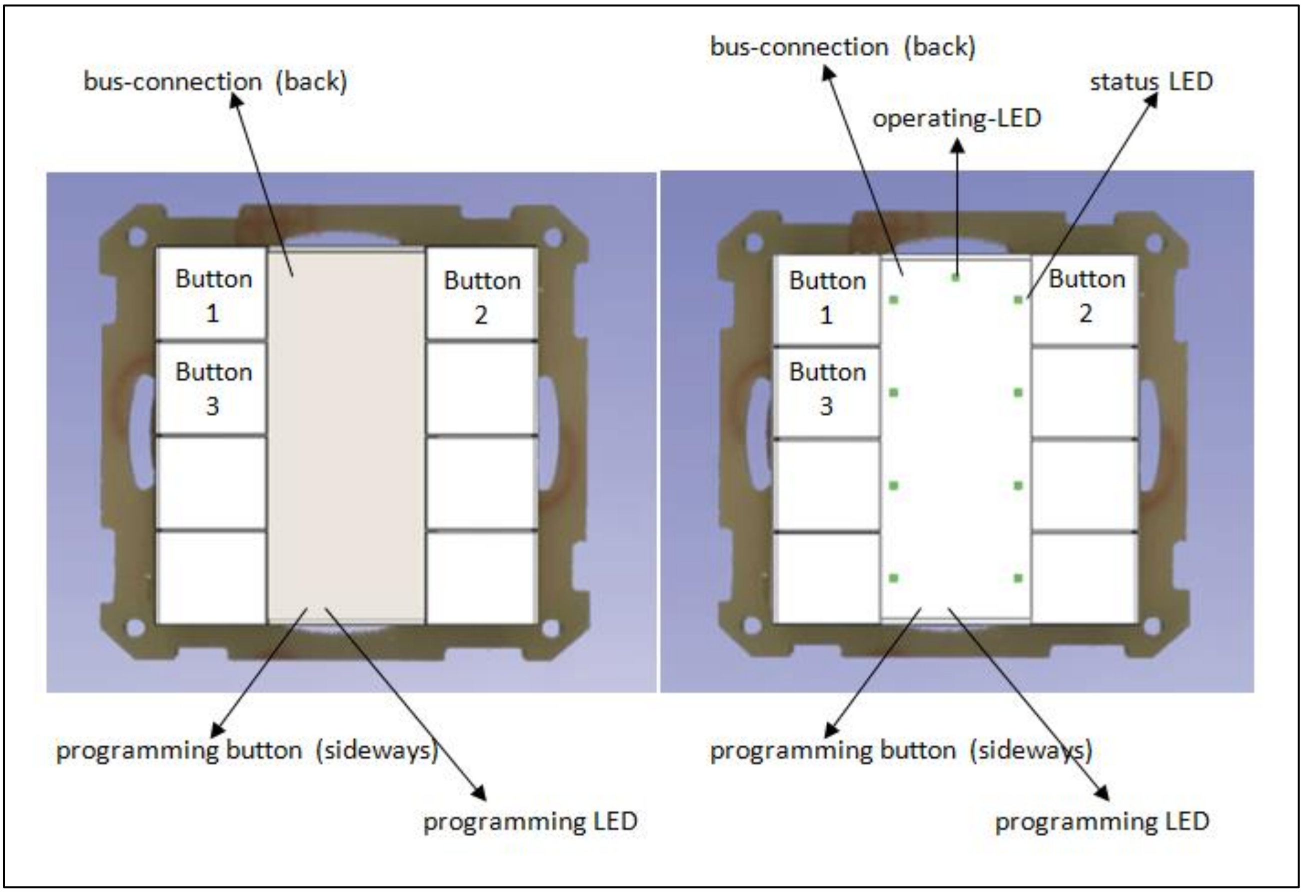


Figure 3: Overview hardware module Push Button(left: BE-TA5508.01; right: BE-TA55P8.01)

The push buttons contains of an invisible cover plate, which is included at the delivery of the push buttons. This cover plate is for the protection of the labeling. When the labeling was inserted, the cover plate can be installed. The cover plate has two lugs at every side, which must engage in the intended executions. For the installation of the cover plate, one side is inserted into the execution. Now you must curve the cover plate a bit, so the second lug fits into the execution. The cover plate should be installed with the plain side up.
There are 2 opportunities for the labeling of the push buttons. When the push button is not to be labeled, a grey insert plate is included, which can be inserted behind the invisible cover plate. If you want to label the push button, you will find a free pattern in the download section at our homepage www.mdtautomation.de. This copy pattern can be adapted to your parameterization and inserted behind the clear cover plate without the grey insert plate.
For dismantling the cover plate, one button is pushed. Now you can lift the cover plate best with a pointy object. So the cover plate jumps out of its execution and can be removed.

2.5 Functions

The functionality is identical for every channel. The device contains of 2, 4, 6 or 8 buttons based on the hardware design.
The designation of the channels is always in a consecutive alphabetic order.
There are three possible functionalities for each channel:

- **Disabled**

No function is set to the channel, so this channel does not contain of any communication objects.

- **Channels grouped**

If you select a pair of channel as “channels grouped”, you will be able to parameterize the pair of channels as dimming-function, shutter-function or switching- function.

- **Channels unique**

If you select a pair of channels as “channels unique”, you will be able to parameterize each channel for itself as switch, counter, scene, switch short/long, one button dimming or one button shutter.

At the plus variant, there are additional 4 logic functions (and/or) containing of up to two additional input objects. Furthermore the plus variant contains of one bicolored LED per channel, which are individual parameterize able, and one bicolored operating LED.

2.5.1 Overview of the functions

General settings	Debounce time	10-120ms, selectable in steps
	Time for keystroke long	0,1-30s, selectable in steps
Channels grouped	Dimming function	brighter/darker function can be assigned to the channels freely
	Shutter function	up/down function can be assigned to the channels freely
	Switching function	off/on telegrams can be assigned to the channels freely
Channels unique	Switching function	• switching function • toggle function • status function • time functions ○ switch on/off delay • edge evaluation • forced settings • sending of byte-values
	Scene function	• memory function • selection of different scenes
	Switch short/long	• On-/Off-/toggle function • short/long independent parameterize able
	One button dimming	• steps of dimming • telegram repetition
	One button shutter	• shutter function with only one button
Logic functions *(only at the plus variant)*	AND-function	• switching function • scene function • inverting
	OR-function	• switching function • scene function • inverting
Configuration of LEDs *(only at the plus variant)*	Status-LEDs	• connections to internal objects possible • connections to external objects possible • reaction to pushing a button possible • LED display behavior parameterize able • luminescent behavior parameterize able • LED priority parameterize able(from hardware version 1.1)
	Operating LED	• on-/off switchable • controlling with ext. objects possible
	Blocking function	• all LEDs lockable by the blocking object

Table 1: Functional overview push buttons

2.6. Settings at the ETS-Software

Selection at the product database:

Manufacturer: MDT Technologies
Product family: Push buttons
Product type: Push buttons/Push buttons plus
Medium Type: Twisted Pair (TP)
Product name: addicted to the used type, e.g.: BE-TA55P8.01 Push button 8-fold, plus variant
Order number: addicted to the used type, e.g.: BE-TA55P8.01

The available parameters depend to the chosen product type. The additional functions for the plus variant are not shown at the normal push buttons.

2.7. Starting up

After wiring the allocation of the physical address and the parameterization of every channel follow:

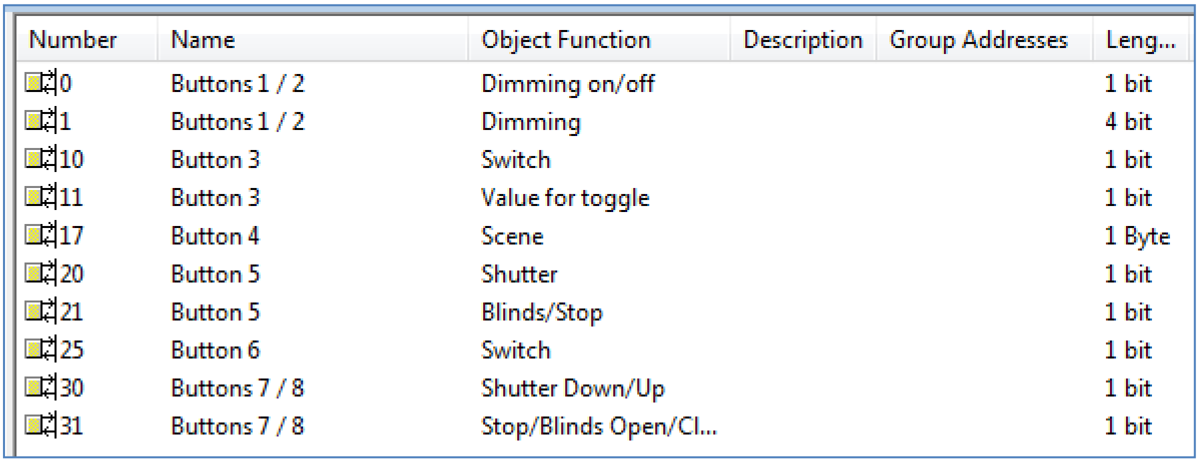
(1) Connect the interface with the bus, e.g. MDT USB interface
(2) set bus power up
(3) Press the programming button at the device(red programming LED lights)
(4) Loading of the physical address out of the ETS-Software by using the interface(red LED goes out, as well this process was completed successful)
(5) Loading of the application, with requested parameterization
(6) Switch the power supply on
(7) If the device is enabled you can test the requested functions(also possible by using the ETS-Software)

3 Communication objects

3.1 Communication objects per channel

The communication objects appear for every channel in dependence of the respective parameterization. 10 numbers (0-9, 10-19,…) for communication objects are automatically assigned for every pair of channels. The numeration is consecutive, so the channel pair A/B can only have the numbers from 0 to 9, the same if they parameterized as grouped channels or unique channels. With every following channel pair the numbers of the objects increase by 10, even if a channel pair is disabled. If you choose a channel pair as unique, the channel, which is first in the alphabet, will become the first 5 numbers and the other one will become the following five numbers (e.g. channel A-->0-4 and channel B-->5-9). The numeration of the channels is always the same, even if some channels are disabled.
At the plus variant, there are additional objects for the LEDs and the logic blocks. The communication objects for the logic follow on the objects for the channels. There are up to 12 objects for the logic function, so 12 numbers are reserved for the logic function, e.g. the numbers from 40 to 51 at an 8-fold push button. The communication objects for the LEDs start consequently with the first numbers after the logic blocks. According to the parameterization, one communication object per LED can be shown. So there are 9 objects at the plus variant of an 8-fold push button and one blocking object.
The follow

The following illustration shows the communication objects for the channels. Channel A/B is selected as “channels grouped” and parameterized as a dimming function. The channels C/D are selected as “channels unique” thus every channel can become an individual function. Channel C is parameterized as switching-function and channel D is parameterized as scene-function. The channels E/F are also selected as unique channels. Channel E is parameterized as shutter and channel F as switch with the sub function “send status”. The channels G and H are selected as grouped with a shutter function:



Number	Name	Object Function	Description	Group Addresses	Leng...
0	Buttons 1 / 2	Dimming on/off			1 bit
1	Buttons 1 / 2	Dimming			4 bit
10	Button 3	Switch			1 bit
11	Button 3	Value for toggle			1 bit
17	Button 4	Scene			1 Byte
20	Button 5	Shutter			1 bit
21	Button 5	Blinds/Stop			1 bit
25	Button 6	Switch			1 bit
30	Buttons 7 / 8	Shutter Down/Up			1 bit
31	Buttons 7 / 8	Stop/Blinds Open/Cl...			1 bit

Figure 4: Communication objects per channel

If a channel pair is selected as disabled, no objects will be shown. So there are no opportunities for programming this channel.
These are the available objects for each channel:

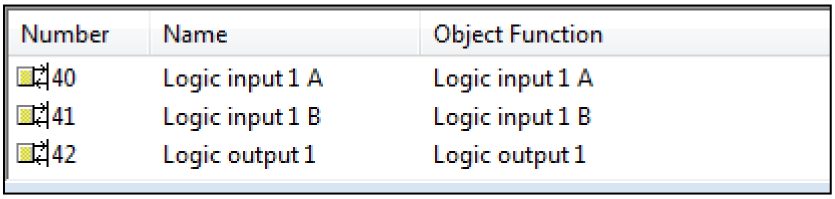
Nr.	Function	Usage	Data type	
0	Switch	edge control	DPT 1.001	Out, Read
0	Send forced setting	force control/switch	DPT 2.001	Out, Read
0	Shutters down/up	driving of shutters	DPT 1.008	Out, Read
0	Dimming on/off	toggling of the dimming lights	DPT 1.001	Out, Read
0	Switch on/off	two button switching	DPT 1.001	Out, Read
0	Send value	sends the parameterized value	DPT 5.001	Out, Read
0	push-button short	sends action for short keystroke	DPT 1.001	Out, Read
1	Value for toggle	edge control with toggle function	DPT 1.001	In, Write
1	Stop/Blinds open/close	driving of the blinds/ stopping movement of the shutters	DPT 1.009	Out, Read
1	Dimming	dimming	DPT 3.007	Out, Read
2	Value for change of direction	reversal of direction for shutters	DPT 1.001	Out, Read
2	Scene	scene control	DPT 18.001	Out, Read
2	push-button long	sends action for long keystroke	DPT 1.001	Out, Read
4	Blocking object	blocks the related channel	DPT 1.001	In, Write
+5	**next channel**			

Table 2: communication objects per channel

3.2 Communication objects logic

→only at the plus variant

There are communication objects for the logic function at every push button additional to the communication objects per channel. These objects can be parameterized and shown independent from the parameterization of the channels. The logic objects have the numbers from 80 at a 16-fold push button and the numbers from 40 at an 8-fold push button. The first logic block gets assigned the first three numbers, so at an 8-fold push button from 40 to 42. Every following block increases the numbers by 3.
The addressing can be made by using the communication objects for the logic analogous to the addressing by the channels.
The following communication objects for the logic can be shown:



Number	Name	Object Function
40	Logic input 1 A	Logic input 1 A
41	Logic input 1 B	Logic input 1 B
42	Logic output 1	Logic output 1

Figure 5: communication objects logic

If a logic block is disabled, no communication objects will be shown. Therefore no addressing is possible. Every push button contains of 4 logic blocks for which the following objects can be shown:

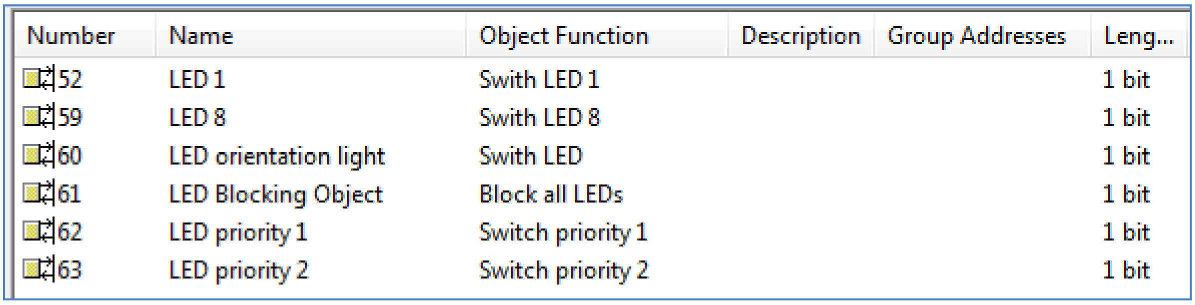
Nr.	Function	Usage	Data type	
40/80	Logic input 1 A	Logic input	DPT 1.001	In, Write
41/81	Logic input 1 B	Logic input	DPT 1.001	In, Write
42/82	Logic output 1	Logic output	DPT 1.001	Out, Read
42/82	Logic output 1 scene	Logic output scene	DPT 18.001	Out, Read
+3	**next logic block**			

Table 3: communication objects logic

3.3 Communication objects LED

→only at the plus variant

The available LEDs can be controlled by different methods. According to the method, one communication object can be shown for every LED, which afterwards can be connected to the group addresses in any way. So there are up to 9 communication objects available at the 8-fold push button. There is additional one blocking object for the LEDs and two objects for the priority of the LEDs.
The following illustration shows the communication objects, which can be shown:



Number	Name	Object Function	Description	Group Addresses	Leng...
52	LED 1	Swith LED 1			1 bit
59	LED 8	Swith LED 8			1 bit
60	LED orientation light	Swith LED			1 bit
61	LED Blocking Object	Block all LEDs			1 bit
62	LED priority 1	Switch priority 1			1 bit
63	LED priority 2	Switch priority 2			1 bit

Figure 6: Communication objects LEDs

The following communication objects are available:

Nr.	Function	Usage	Data type	
22/32/ 42/52	LED 1	switch LED	DPT 1.001	In, Write, Out Read
+1	next LED			
30/40/ 50/60	LED orientation light	switch LED	DPT 1.001	In, Write, Out, Read
31/41/ 51/61	LED blocking object	block all LEDs	DPT 1.001	In, Write, Out, Read
32/42/ 52/62	LED priority 1*	switch priority 1	DPT 1.001	In, Write, Out, Read
33/43/ 53/63	LED priority 2*	switch priority 1	DPT 1.001	In, Write, Out, Read

Table 4: Communication objects LEDs

***=from hardware version 1.1**

3.4 Default settings of the communication objects

The following chart shows the default settings for the communication objects:

Default settings									
Nr.	**Button**	**Function**	**Length**	**Priority**	**C**	**R**	**W**	**T**	**U**
0	Button 1	Switch	1 Bit	Low	X	X		X	
0	Button 1	Shutter	1 Bit	Low	X	X		X	
0	Button 1	Send value	1 Byte	Low	X	X		X	
0	Button 1	Dimming on/off	1 Bit	Low	X	X		X	
0	Button 1	push-button short	1 Bit	Low	X	X		X	
0	Button 1	push-button short	1 Byte	Low	X	X		X	
0	Button 1	Send forced setting	2 Bit	Low	X	X		X	
0	Buttons 1/2	Switch on/off	1 Bit	Low	X	X		X	
0	Buttons 1/2	Dimming on/off	1 Bit	Low	X	X		X	
0	Buttons 1/2	Shutter down/up	1 Bit	Low	X	X		X	
1	Button 1	Value for toggle	1 Bit	Low	X	X		X	
1	Button 1	Stop/Blinds open/close	1 Bit	Low	X	X		X	
1	Button 1	Dimming	4 Bit	Low	X	X		X	
1	Buttons 1/2	Dimming	4 Bit	Low	X	X		X	
1	Buttons 1/2	Stop/Blinds open/close	1 Bit	Low	X	X		X	
2	Button 1	Scene	1 Byte	Low	X	X		X	
2	Button 1	Value for toggle	1 Bit	Low	X		X	X	X
2	Button 1	Value for change of direction	1 Bit	Low	X		X	X	X
2	Button 1	Push-button long	1 Bit	Low	X	X		X	
2	Button 1	Push-button long	1 Byte	Low	X	X		X	
4	Button 1	Blocking object	1 Bit	Low	X		X		X
10/20/30/40	Logic input 1 A*	Logic input 1 A	1 Bit	Low	X		X		X
11/21/31/41	Logic input 1 B*	Logic input 1 B	1 Bit	Low	X		X		X
12/22/32/42	Logic output 1*	Logic output 1	1 Bit	Low	X	X		X	
12/22/32/42	Logic output 1 scene*	Logic output 1 scene	1 Byte	Low	X	X		X	
12/22/32/42	Logic output 1*	Logic output 1 value	1 Byte	Low	X	X		X	
22/32/42/52	LED 1*	LED switch	1 Bit	Low	X	X	X	X	X
30/40/50/60	LED orientation light*	LED switch	1 Bit	Low	X	X	X	X	X
31/41/51/61	LED blocking object*	block all LEDs	1 Bit	Low	X	X	X	X	X
32/42/52/62	LED priority 1**	switch priority 1	1 Bit	Low	X		X	X	X
33/43/53/63	LED priority 2**	switch priority 1	1 Bit	Low	X		X	X	X

Table 5: Communication objects – default settings

You can see the default values for the communication objects from the upper chart. According to requirements the priority of the particular communication objects as well as the flags can be adjusted by the user. The flags allocates the function of the objects in the programming thereby stands C for communication, R for Read, W for write, T for transmit and U for update.

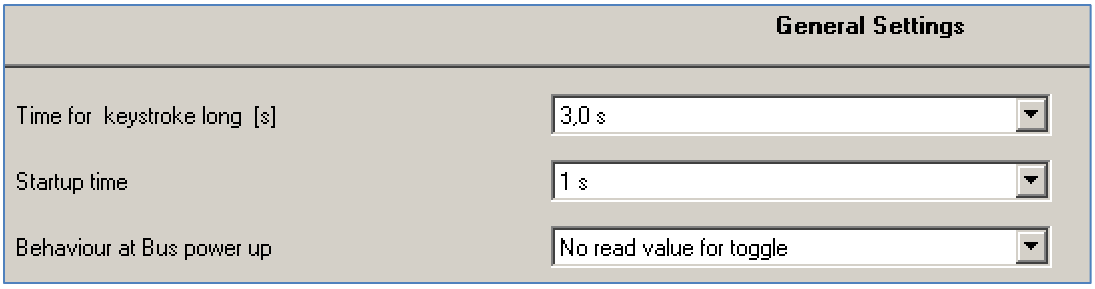
***= only at the plus variant**

****=from hardware version 1.1**

4 Reference ETS-Parameter

4.1 General Settings

The following parameters exist once and affect all channels:



General Settings

Time for keystroke long [s] 3,0 s

Startup time 1 s

Behaviour at Bus power up No read value for toggle

Figure 7: General settings

The following chart shows the dynamic range for the general settings:

ETS-text	Dynamic range **[default value]**	comment
Time for keystroke long	0,1-30 sec **[0.8 sec]**	releases the time when the ETS recognizes a long keystroke
Startup time	1 – 60s **[1s]**	time between programming and functional start of the device
Behavior at bus power up	▪ **No read value for toggle** ▪ Read value for toggle	activates the reading of the value for toggle at bus power up

Table 6: General settings

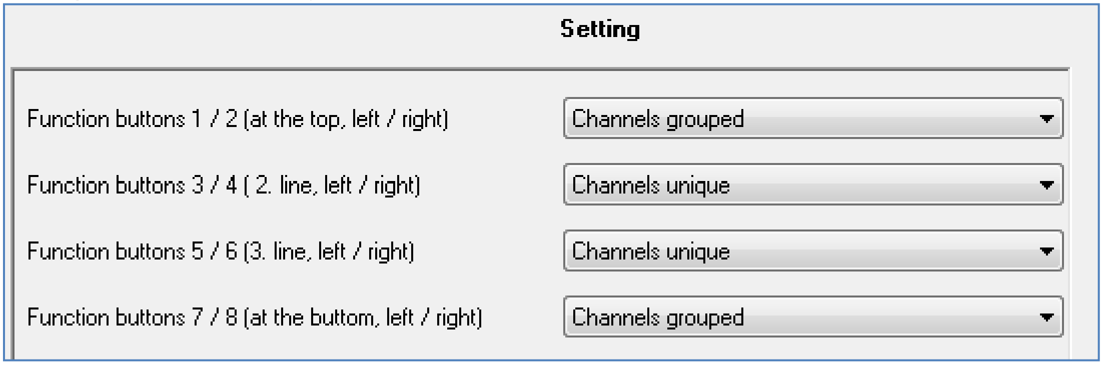
1. All push buttons have a fixed debouncing time, which cannot be changed by the user. The debounce time is adjusted in-plant to the push buttons.

   1. The parameter "Time for keystroke long" allocates a static value to the push button from when a long keystroke is recognized. This parameter is important for functions, which have different functions for a long and a short keystroke.

   2. The parameter "Behavior at bus power up" defines the behavior of the push button at a bus power return. The setting "Read value for toggle" effects that all communication objects "value for toggle" are read. So the push button knows the current status of the objects. If you choose the setting "no read value for toggle", the push button will not know the current status of the actor. So the push button assumes an unconfirmed value for the objects "value for toggle" and sends always a "0"-signal at the next operation. Only now the push button knows the status of the actor and can send the right values. But if you choose the read of these values at a bus power up, the push button will send immediately the right value for toggling.

4.2 Configuration

Setting of the functionality of the channels:



Setting

Function buttons 1 / 2 (at the top, left / right) — Channels grouped

Function buttons 3 / 4 (2. line, left / right) — Channels unique

Function buttons 5 / 6 (3. line, left / right) — Channels unique

Function buttons 7 / 8 (at the buttom, left / right) — Channels grouped

Figure 8: Usage of the buttons

ETS-text	Dynamic range **[default value]**	comment
Function Button A/B –[O/P]	▪ **disabled** ▪ Channels grouped ▪ Channels unique	Operating mode of the channels

Table 7: Parameter channel-configuration

There are 3 different operating modes for every button(have a look at chart 7). The followig options to parameterize the channels are dependent to the choosen operating mode. If you disable the channel, there will be no options to parameterize this channel.

4.3 Identical parameter

4.3.1 Blocking object

As well for grouped channels as for unique channels the blocking object can be activated. At the unique channels one blocking object for every channel can be activated. For grouped channels, you can activate one blocking object for both channels. The communication object for a channel appears as soon as it is activated for a channel. So there are up to 8 blocking objects parameterize able at a 8-fold push button. The corresponding channel of the blocking object is blocked by sending a logical 1. A blocked channel is not controllable as long as it is blocked. By sending a logical 0, the channel can be unblocked again.

Number	**Name**	**Length**	**Usage**
4	Blocking object	1 Bit	blocks the related channel by sending a logical 1

Table 8: Communication object blocking object

4.4 Parameter Channels grouped

The chart shows the setting options for grouped channels:

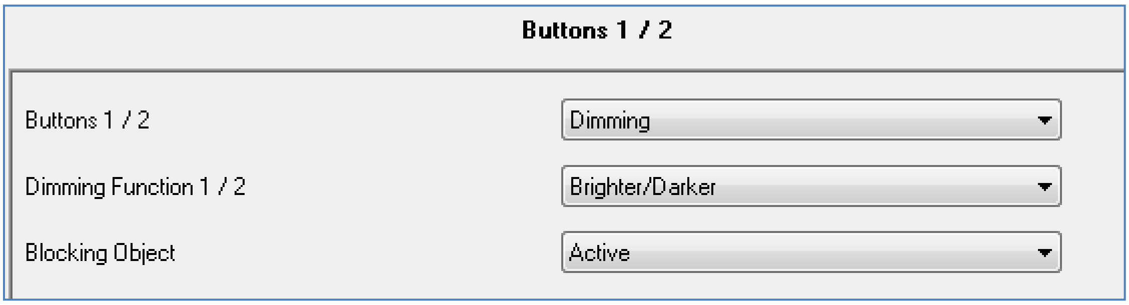
ETS-text	Dynamic range **[default value]**	comment
Button A/B	▪ **Dimming** ▪ Shutter ▪ Switch	Operating mode of the channel
Dimming function A/B	▪ **Brighter/Darker** ▪ Darker/Brighter	Defines which channel should dim up and which should dim down
Shutter function A/B	▪ **Up/Down** ▪ Down/Up	Defines which channel should drive the shutter a down and which up
Switch function A/B	▪ **On/Off** ▪ Off/On	Defines which channel should switch off and which on
Blocking Object	▪ **Inactive** ▪ Active	The blocking object can be displayed for every pair of channels

Table 9: Parameter Channels grouped

By choosing channels as grouped, two channels become one common function. The grouped function is called dual surface, like dual surface dimming, and dual surface shutter. In contrast to the single surface functions, one action can be performed independent form the other one. One input performs always one function. The assignment for the buttons can be made individually, so it is possible to configure which button should for example drive the shutters up and which down.

4.4.1 Dimming

The dual surface dimming function (channels grouped) is for controlling dimming actuators by start-stop dimming commands.
The following parameters are visible, when a pair of channels is chosen as dimming-function:



Buttons 1 / 2

Buttons 1 / 2	Dimming
Dimming Function 1 / 2	Brighter/Darker
Blocking Object	Active

Figure 9: Parameter dual surface dimming

Number	Name	Length	Usage
0	Dimming on/off	1 Bit	Switching function of the dimming process; action for a short keystroke
1	Dimming	4 Bit	Dimming function; action for a long keystroke

Table 10: Communication objects dual surface dimming

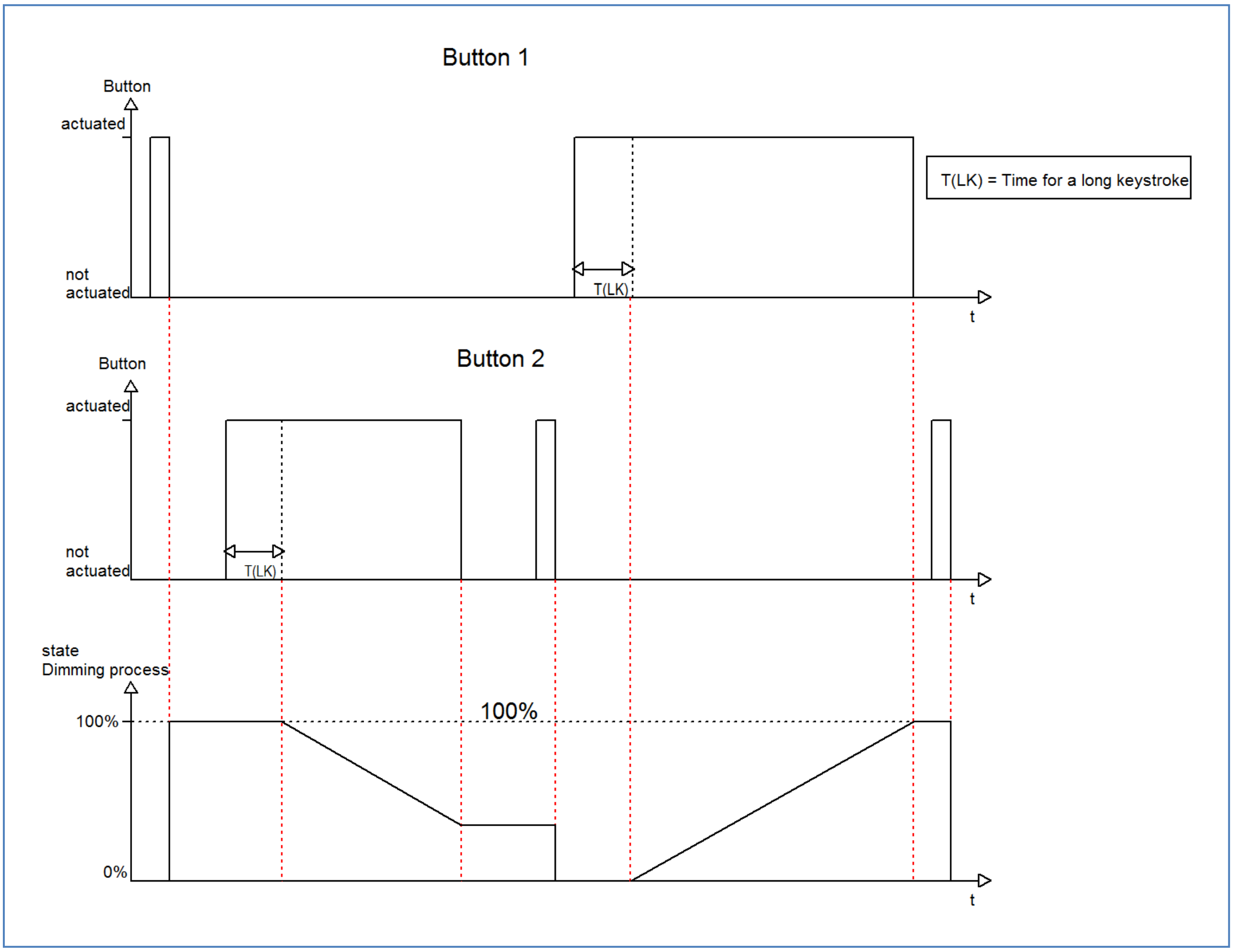
When a pair of channels is parameterized as dimming function, two objects are shown. One object reacts to a short keystroke, the switching object “Dimming on/off”, and the other object reacts to a long keystroke, the dimming object “dimming”.
It is possible to parameterize this function as brighter/darker or as darker/brighter. The first function belongs always to the first button. If you switch this parameter, the function will be switched automatically.
By choosing the dimming function (channel A/B) as brighter/darker, the function reacts in this way:
A short keystroke at button A switches the lights on. The lights are switched off by a short keystroke at button B. A long keystroke dims the lights step by step until releasing the long keystroke. The lights are dimmed brighter at button A and darker at button B. The push button starts always with the last brightness level, before switching off.
The step size is set fixed to 100% at the dual surface dimming. It is a start-stop dimming. that means the lights are dimmed as long as you hold the button. After releasing the button a stop value is sent, which stops the dimming process. So you can dim the lights with only one keystroke from 0% to 100% or from 100% to 0%, by pushing the button long enough.

The chart shows the correlations between the dimming- and the switching-object:

	Function Brighter/Darker			**Function Darker/Brighter**	
Button	Button A	Button B		Button A	Button B
Dimming function	Brighter	Darker		Darker	Brighter
Switching function	On	Off		Off	On

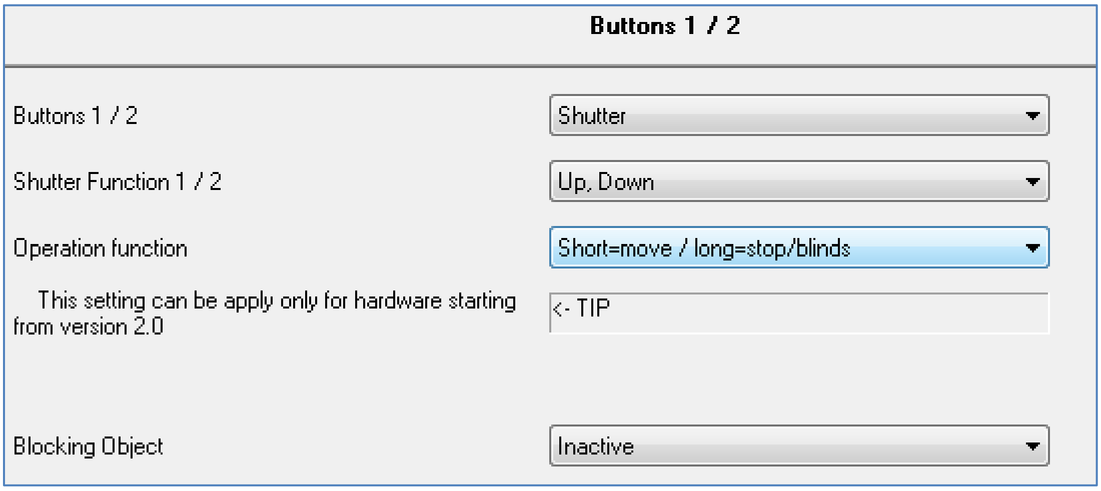
Table 11: Dimming function

The following diagram shows the dual surface dimming function:



4.4.2 Shutter

The two-button shutter-function triggers shutter actuators, which can drive shutter and blinds.
The following parameters are shown, when a pair of channel is adjusted as shutter function:


Buttons 1 / 2

Buttons 1 / 2: Shutter

Shutter Function 1 / 2: Up, Down

Operation function: Short=move / long=stop/blinds

This setting can be apply only for hardware starting from version 2.0: <- TIP

Blocking Object: Inactive

Figure 10: Two-button shutter function

Number	Name	Length	Usage
0	Shutter Down/Up	1 Bit	Driving function for the shutters, action for a long keystroke
1	Stop/Blinds Open/Close	1 Bit	Stop/Adjustment of the blinds, action for a short keystroke

Table 12: Communication objects Two-button shutter function

If you choose a pair of channels as shutter function, two communication objects will appear for this pair of channel. On the one hand the stop/blind adjustment object called “Stop/Blinds Open/Close”, which responds to a short keystroke and on the other hand the driving object called “Shutter Down/Up”, which responds to a long keystroke.
The driving object is for moving the shutters up and down. The stop-/blind adjustment object is for the adjustment of the blinds and additional it stops a running movement of the shutter.
Every shutter actuator controls with a 0-signal the up-movement and with a 1-signal the down movement. So the push button sends these signals to the corresponding driving commands.
From hardware version 2.0 (have a look at the print of the side of the device: RX.X), it is additional possible to switch the functions for a long and a short keystroke. So it can be chosen whether he shutter/blinds shall be driven via a long or a short keystroke. The Stop-/Blind adjustment object is adjusted by the other operating concept.

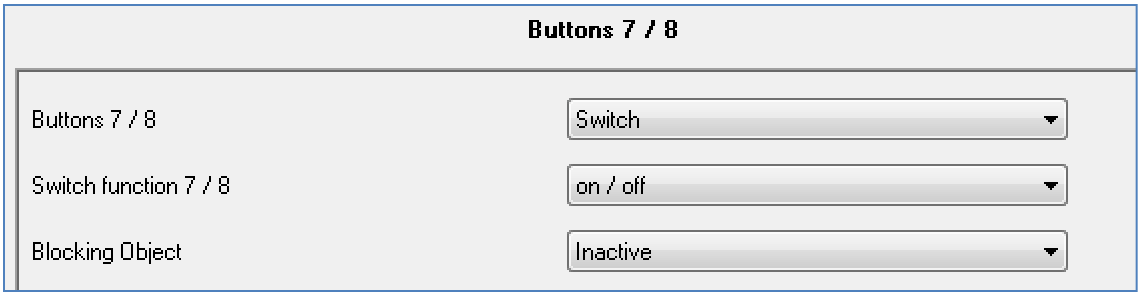
The Chart shows the correlations between the Stop-/Blind adjustment object and the driving object for the individual channels:

	Function Down/Up			Function Up/Down	
Button	Button A	Button B		Button A	Button B
Stop-/Blind adjustment object	Down	Up		Up	Down
Driving object	Stop/close blinds	Stop/open blinds		Stop/open blinds	Stop/close blinds

Table 13: shutter function

4.4.3 Switch

The values for on and off can be assigned freely at the switching function for the grouped channels. If you adjust a pair of channel as switch, the following parameters will be shown:



Buttons 7 / 8

Buttons 7 / 8	Switch
Switch function 7 / 8	on / off
Blocking Object	Inactive

Figure 11: Two-button switching function

Simple functions, like an alternating circuit, can be programmed easily by using the grouped switch function. The 1 bit communication object sends in dependence of the parameterization a 0- or a 1-signal for the first button and the inverted signal for the second channel. So you can chose which channel should switch off and which should switch on.

The following chart shows the corresponding communication object:

Number	Name	Length	Usage
0	Switch On/Off	1 Bit	Switching object for the dual surface switching function

Table 14: Communication object Two-button switching function

4.5 Parameters channels unique

There are 6 different operating modes for the unique channels, which can be adjusted for each channel:

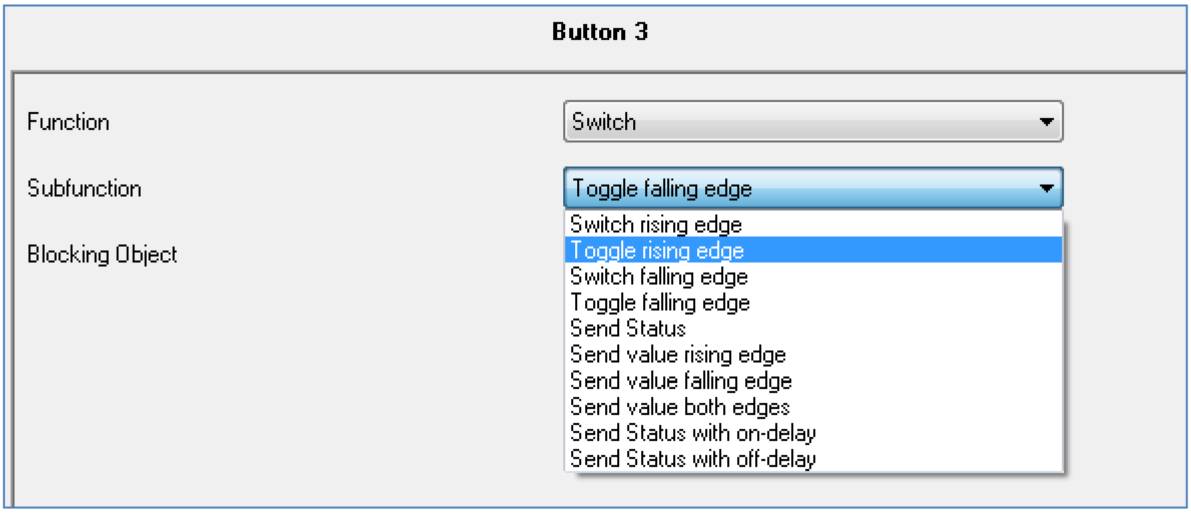
- Inactive
- Switch
- Scene
- Switch short/long
- One button dimming
- One button shutter

After the assignment of the operating mode the further parameterization can be done. If the channel is selected as inactive, no further parameterization will be possible.

4.5.1 Switch

The switching function is for switching the corresponding output on, off and toggling it. There is a multitude of sub-functions at the switching function, which enables the user to evaluate edges and integrate times to the switching process.

The following parameters are shown, when the channel is selected as switch:



Button 3

Function: Switch

Subfunction: Toggle falling edge

Blocking Object

Figure 12: Parameter switch

Various sub-functions are available at a switching output. Most of these sub-functions contain also of further parameterization-options. The different sub-functions as well as their parameterization-options are described in the following segments:

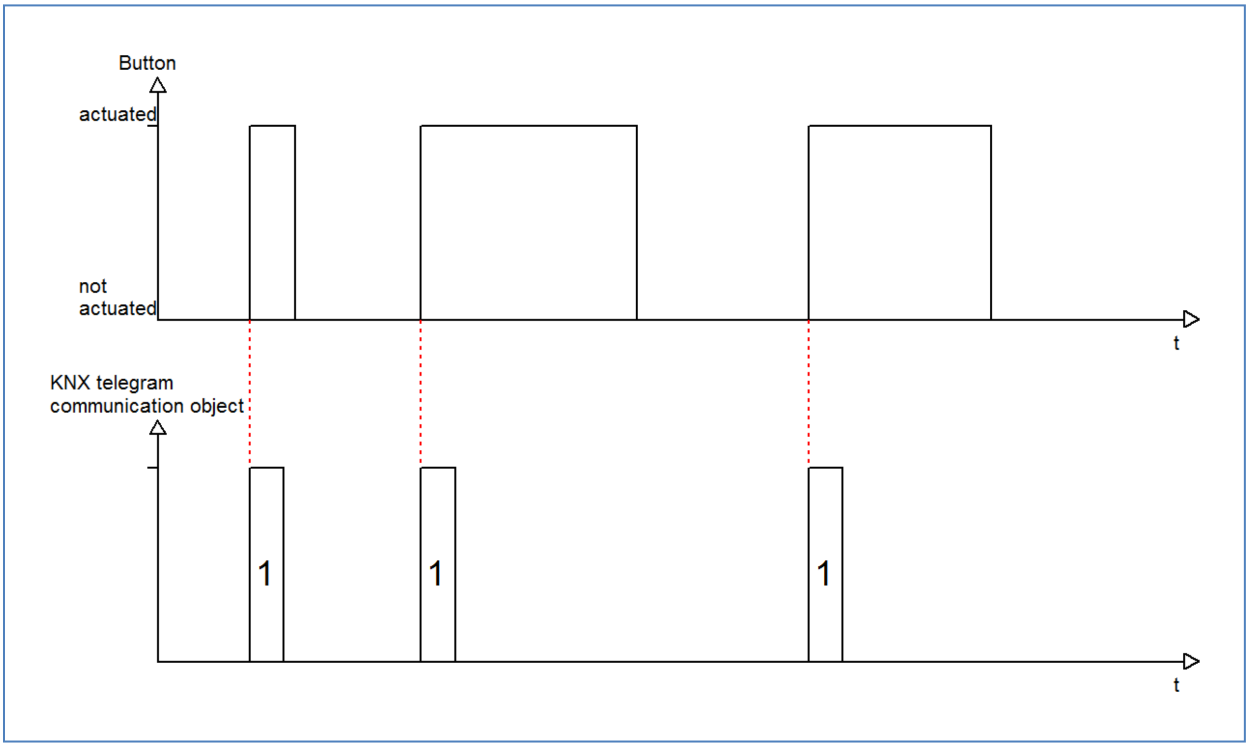
4.5.1.1 Switch falling/rising edge

The following setting options are available, when the sub-function switch falling/rising edge was adjusted:

ETS-text	Dynamic range **[default value]**	comment
Value for rising/falling edge	▪ **On** ▪ Off	switches on/off at a falling/rising edge

Figure 13: Parameter switch rising/falling edge

The sub-function "switch rising edge" or "switch falling edge" sends only a signal at the adjusted edge. You can parameterize whether a 0-signal or a 1-signal should be sent. There is no inverted signal at subsiding the edge. This function always sends only one adjusted signal.
The following diagram shows this sub-function for rising edges. As soon as the state changes from 0 to 1, the push button sends an On-pulse (=1-signal):



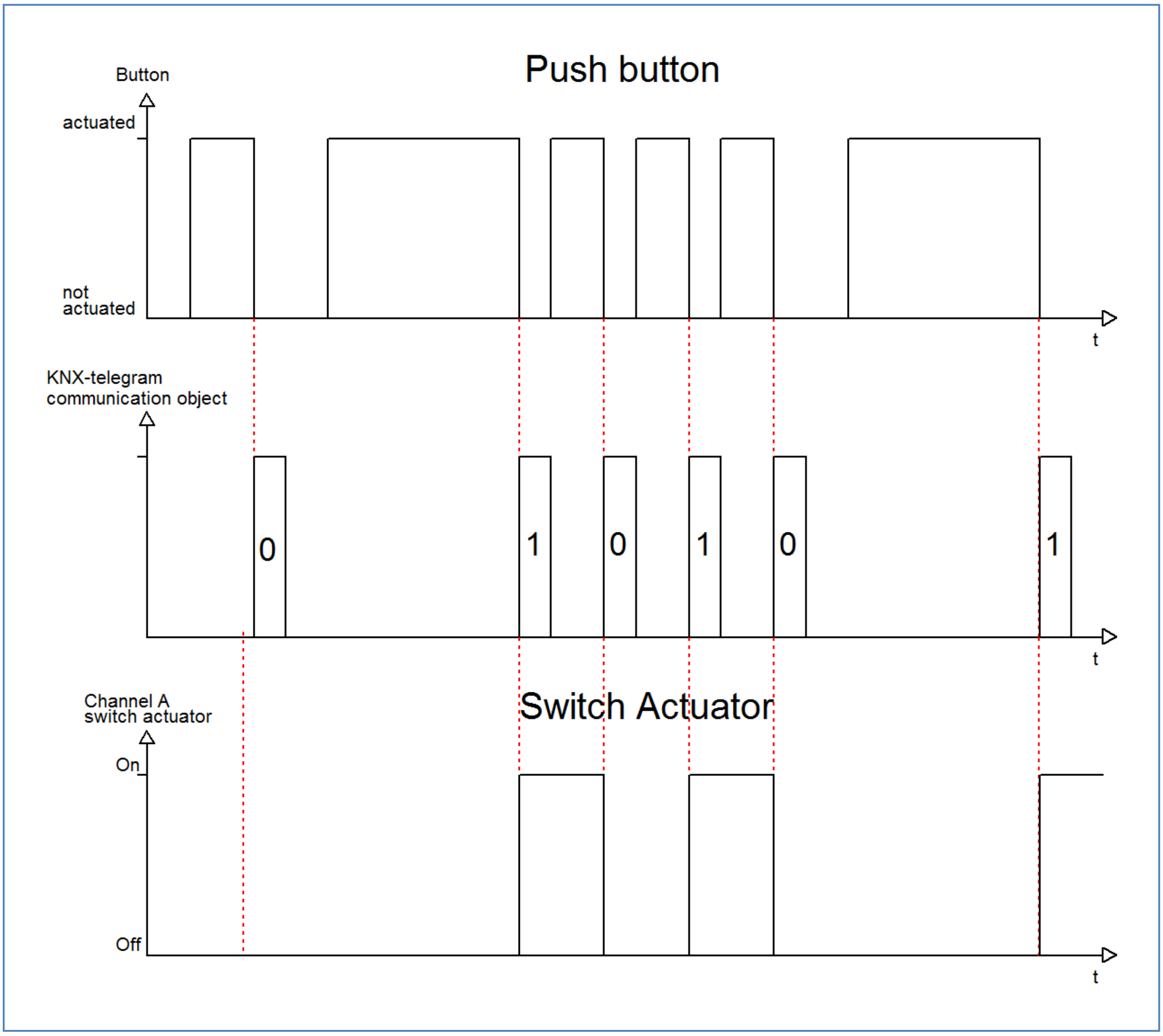
The following chart shows the corresponding communication object:

Number	Name	Length	Usage
0	Switch	1 Bit	Switching function, no differences between a long and a short keystroke

Table 15: Communication object switch rising/falling edge

4.5.1.2 Toggle rising/falling edge

The sub-function “toggle rising edge“ or “toggle falling edge” toggles at the adjusted edge. That means, the current value of the communication object is inverted at every switching process. By using this function an edge based alternating circuit can be realized.
The following diagram describes this sub-function. As soon as the state changes from 1 to 0, the push button sends the inverted signal. The signal is send always as a short pulse:



The following chart shows the corresponding communication objects:

Number	Name	Length	Usage
0	Switch	1 Bit	Switching function; no differences between long and short keystroke
1	Value for toggle	1 Bit	status object, indicates the switching state of the channel

Table 16: Communication objects toggle rising/falling edge

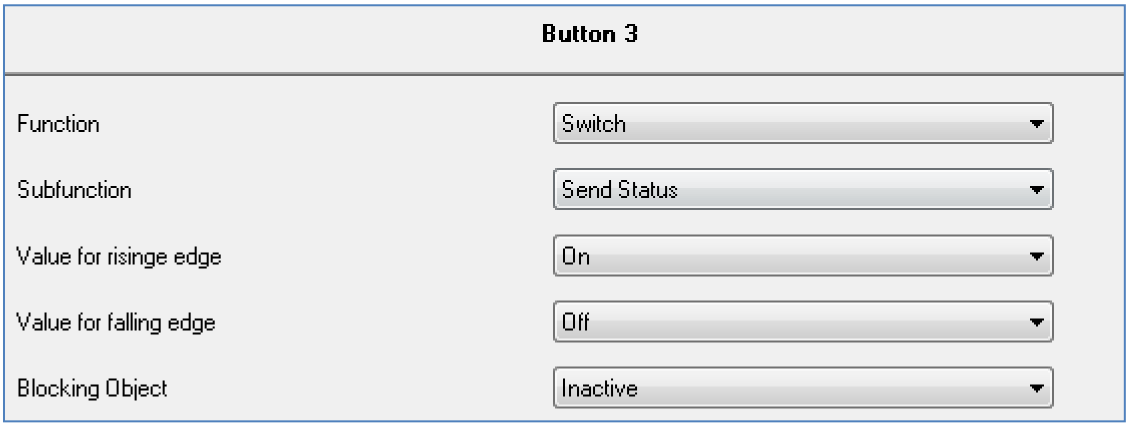
To be sure that the push button toggles at every switching process, you have to connect the status object of the push button “Value for toggle” with the status object of the actuator. When the push button should work without an actuator, the object has to be connected to the switching object “switch”. The connection is important, because the push button cannot invert the signal, when it does not know its current state.

By undocking this communication object, you have more choices to program the push button. So you can use the object “Value for toggle” for visualizations or additional functions and you will be more free in design your project.

So you have for example the option to visualize the switching process by connecting the status-object to a switching object of a LED or something else.

4.5.1.3 Send Status

By using the sub-function „Send status" the push button sends always the parameterized signal for the corresponding edge. The following window is shown for the sub-function "Send status":



Button 3

Function	Switch
Subfunction	Send Status
Value for risinge edge	On
Value for falling edge	Off
Blocking Object	Inactive

Figure 14: Sub-function send status

These settings are available:

ETS-text	Dynamic range **[default value]**	comment
Value for rising edge	▪ **On** ▪ Off	switches on/off at a rising edge
Value for falling edge	▪ On ▪ **Off**	switches on/off at a falling edge

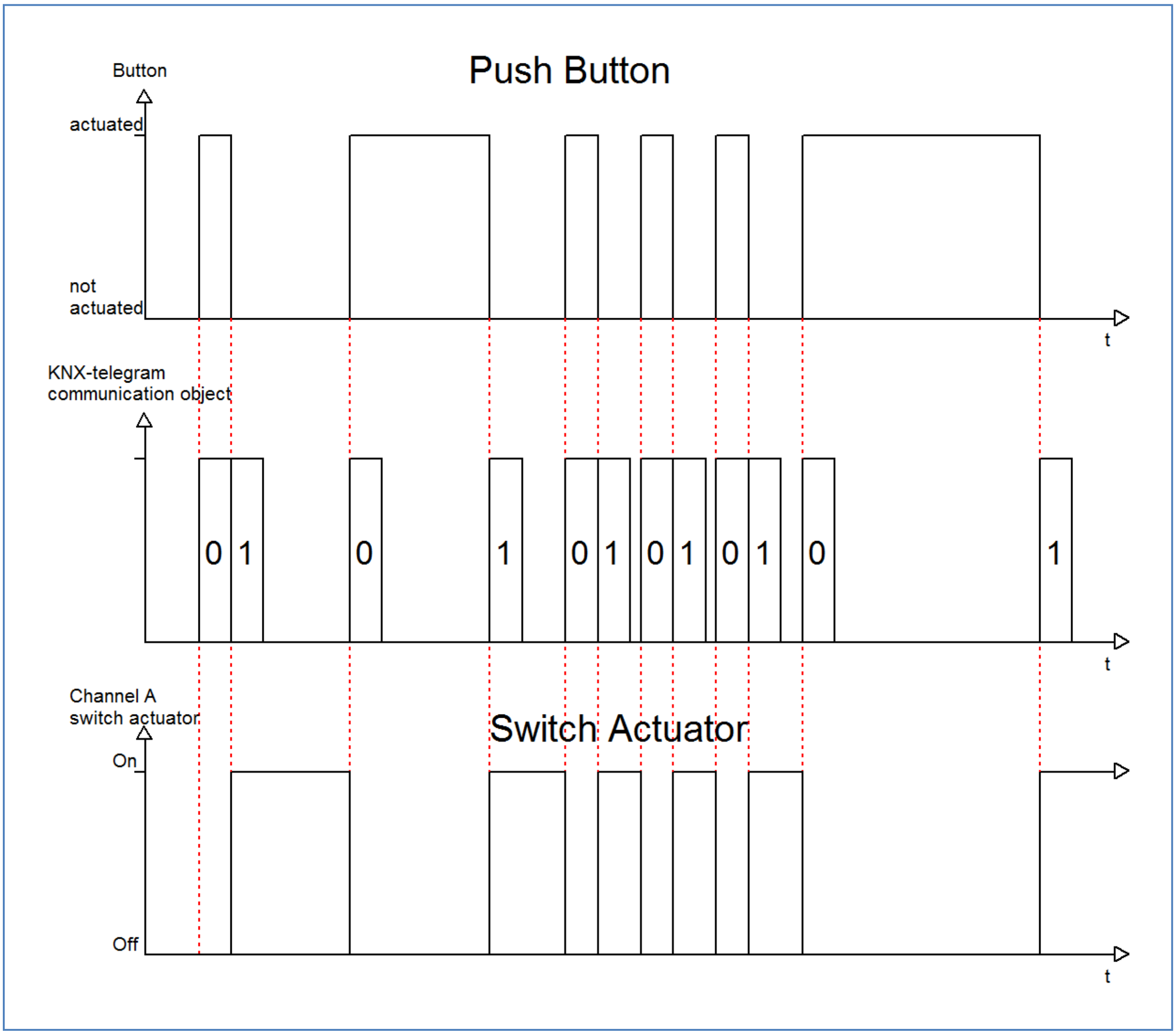
Table 17: Parameter Send status

The corresponding communication object is shown at the following chart:

Number	**Name**	**Length**	**Usage**
0	Switch	1 Bit	Switching function; no differences between long and short keystroke

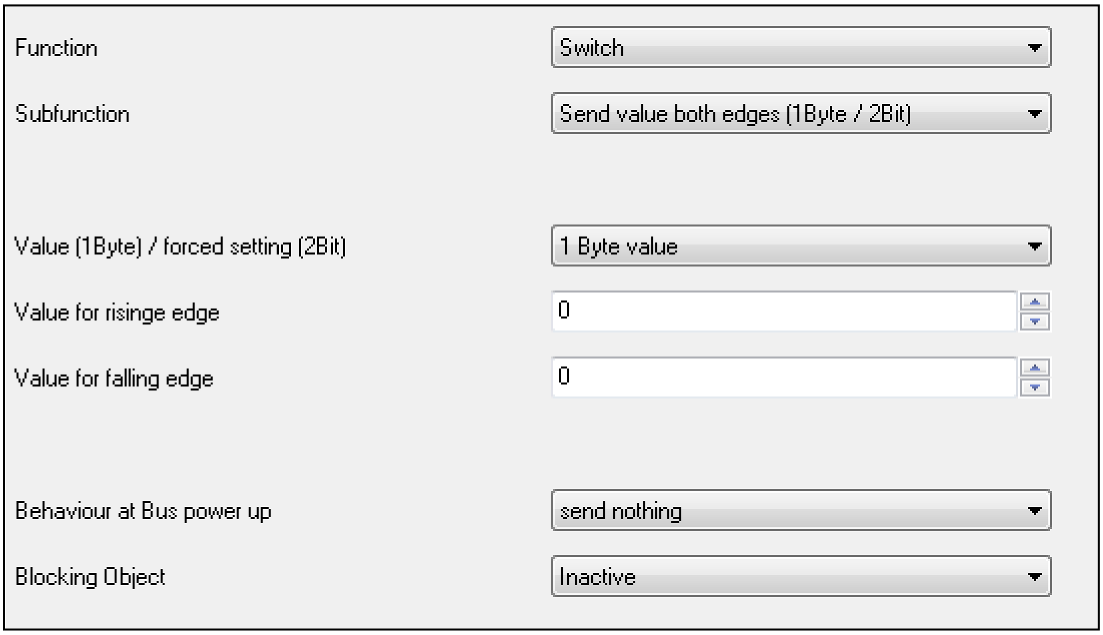
Table 18: Communication object send status

The parameter "Value for rising edge" defines whether the channel should send an 1-signal (value: On) or a 0-signal (value: Off). If you want for example switch a channel of a switch actuator, you will have to choose different values for the rising and the falling edge. Otherwise the push button sends the same signal twice, for example an On-signal.
The cyclic sending causes that the state of the push button is sent periodically in certain parameterize able intervals. Then the push button sends the parameterized value for the corresponding edge.
A common application for this parameter is for example the observation of windows, which are equipped with window-contacts. So a display can for example show whether all windows are closed or not. Furthermore an alarm device can operate with this function.
The following diagram describes this sub-function. In this example, the push button sends a 1-signal for a falling edge and a 0-signal for a rising edge. Additional the diagram shows the connection with a switch actuator, which was parameterized with a normal switching function:



4.5.1.4 Send Value rising/falling/both edges

There are two further sub-functions at the sub-function Send Value. On the one hand you can send 1 Byte Values and on the other hand you can activate a forced setting (2 Bit). These functions can be parameterized according to your wishes.
The following illustration shows this parameter:



Function: Switch
Subfunction: Send value both edges (1Byte / 2Bit)
Value (1Byte) / forced setting (2Bit): 1 Byte value
Value for risinge edge: 0
Value for falling edge: 0
Behaviour at Bus power up: send nothing
Blocking Object: Inactive

Figure 15: Sub-function send value

After activating the sub function „Send value“, you have to choose which values should be sent. The setting options are shown at the chart:

ETS-text	Dynamic range **[default value]**	comment
Value (1 Byte)/ forced setting(2 Bit)	▪ **1 Byte Value** ▪ 2 Bit Value(forced setting)	Choice between 1 Byte- and 2 Bit-Value

Table 19: Parameter send value

If you have activated the setting “1 Byte”, the following settings are possible:

ETS-text	Dynamic range [default value]	comment
Value for rising/falling edge	0-255 [0]	Assignment, which value should be send for the falling/rising edge

Table 20: Parameter send value, 1 Byte object

The 1 Byte communication object can send any value in its dynamic range at both edges. The dynamic range is thereby from 0-255. Depending on parameterization the push button sends the adjusted values for the rising or the falling edge or for both edges.
The following chart shows the according communication object:

Number	**Name**	**Length**	**Usage**
0	Send value	1 Byte	sends the parameterized value

Table 21: Communication object Parameter Send value-1 Byte object

The setting option 2 Bit value (forced setting) has the following options to parameterize this function:

ETS-text	Dynamic range **[default value]**	comment
Send forced setting at rising/falling edge	▪ **Forced setting not active** ▪ Forced setting off ▪ Forced setting on	Assignment, which forced setting should be send at which edge

Table 22: Dynamic range send value-forced setting

The forced setting object allows for example to control the automatic brightness control of presence detectors.

The forced setting object can send 3 different states:

- **Forced setting not active (control=0; value=0)**
  The forced setting object has no influence on the receiver. For example at a presence detector, the automatic function (motion detector operation) would be switched on.
- **Forced setting off (control=1; value=0)**
  The forced setting object switches the receiver unconditionally off. For example a presence detector, would be switched permanent off. Detected motions have no influence on the output.
- **Forced setting on (control=1, value=1)**
  The forced setting object switches the receiver unconditionally on. For example a presence detector, would be switched permanent on. Detected motions have no influence on the output.

The according communication object is shown at the chart:

Number	Name	Length	Usage
0	Send forced setting	2 Bit	sends the adjusted forced setting

Table 23: Communication object Send value-forced setting

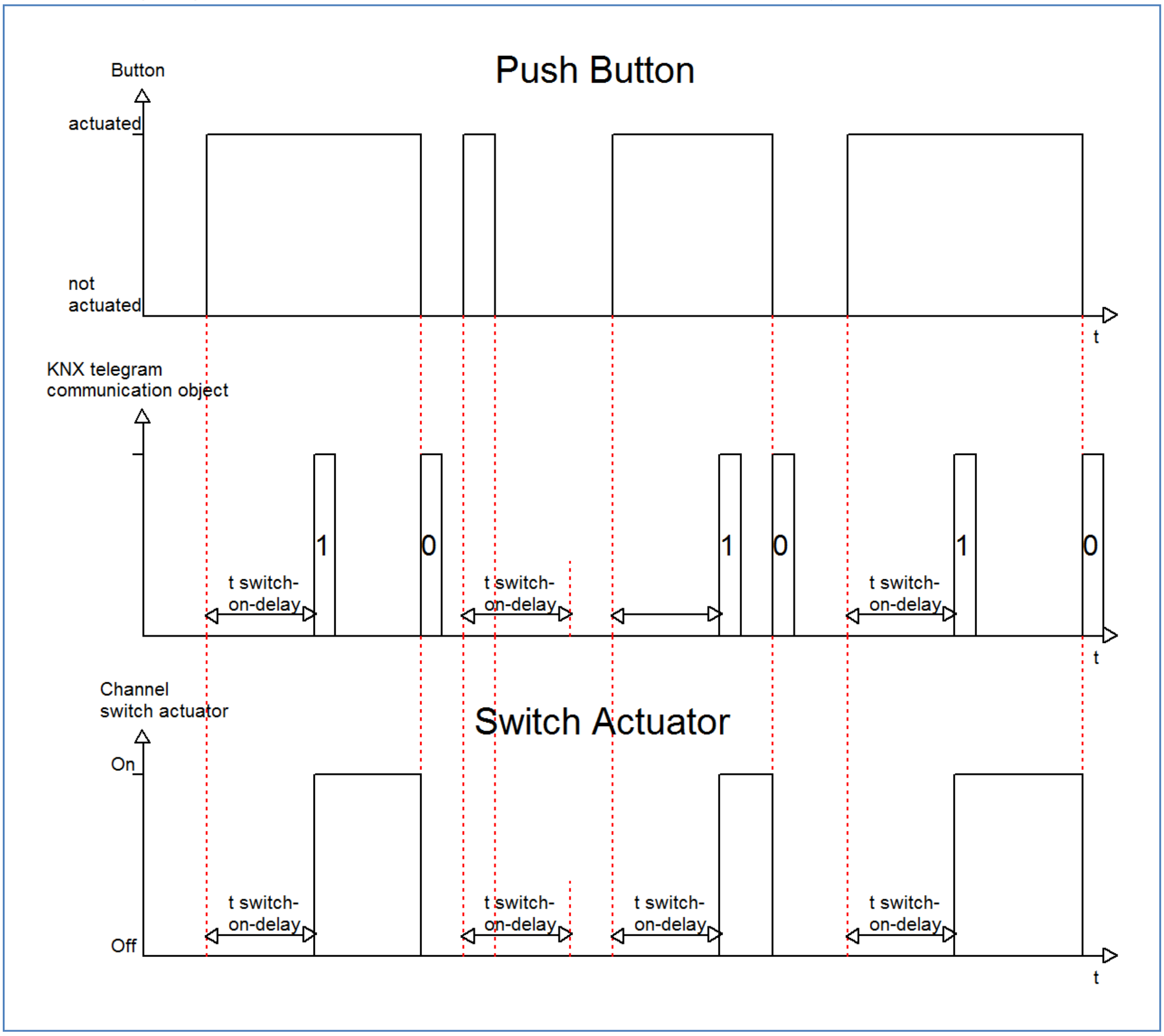
4.5.1.5 Send value with on/off delay

The following setting options are available at the function "Send value with on/off delay":

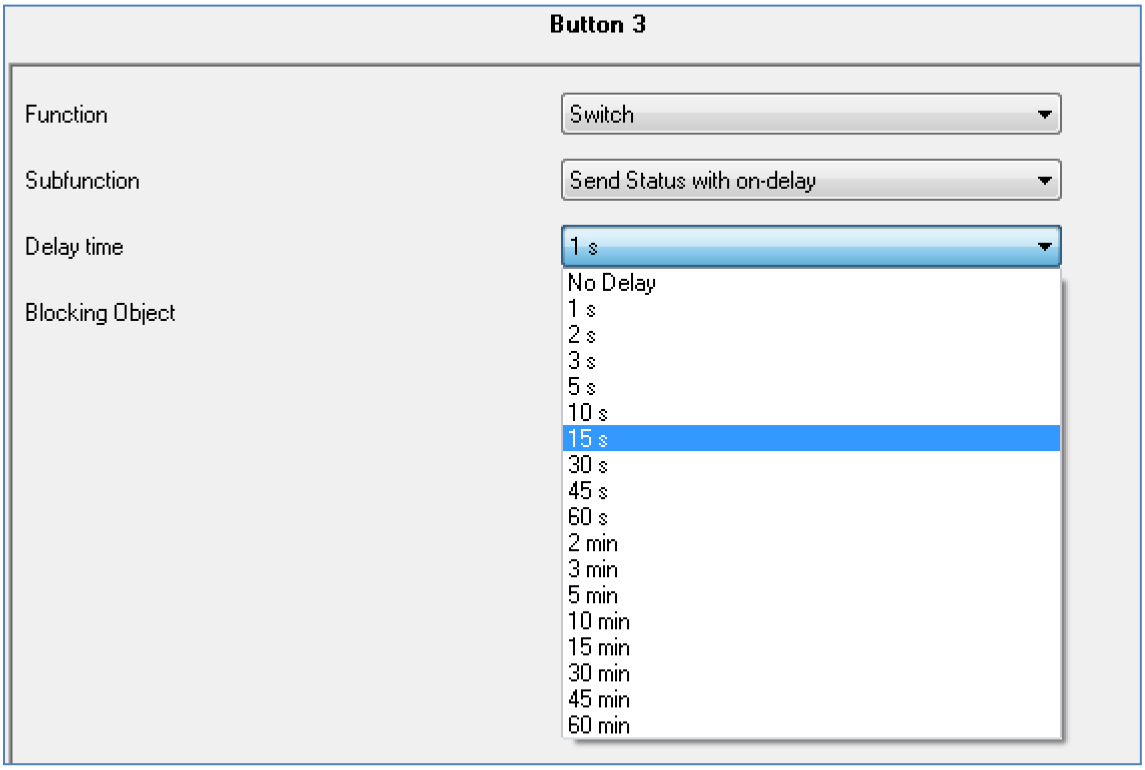
ETS-text	Dynamic range **[default value]**	comment
Delay time	0-60min [1s]	Adjustment of the delay time for the sending process

Table 24: Parameter Send value with delay

The sub-function "Send value with on/off delay" allows that the push button sends its value after a parameterized time. At the on-delay, the time starts when the associated button was switched on and at the off-delay, the time starts when the associated button was switched off. The push button sends always its current value at this function. If the value changes before the time ran out, the on-delay will expire. For example, when an input with a parameterized on-delay is switched off, before it was switched on, the input remains off.
The following diagram describes the sub-function „Send value with on-delay":



You can see the adjusted settings, which were made in the ETS for this setting:



Button 3

Function: Switch

Subfunction: Send Status with on-delay

Delay time: 1 s

Blocking Object

Figure 16: Send value with on-delay

The following chart shows the communication object:

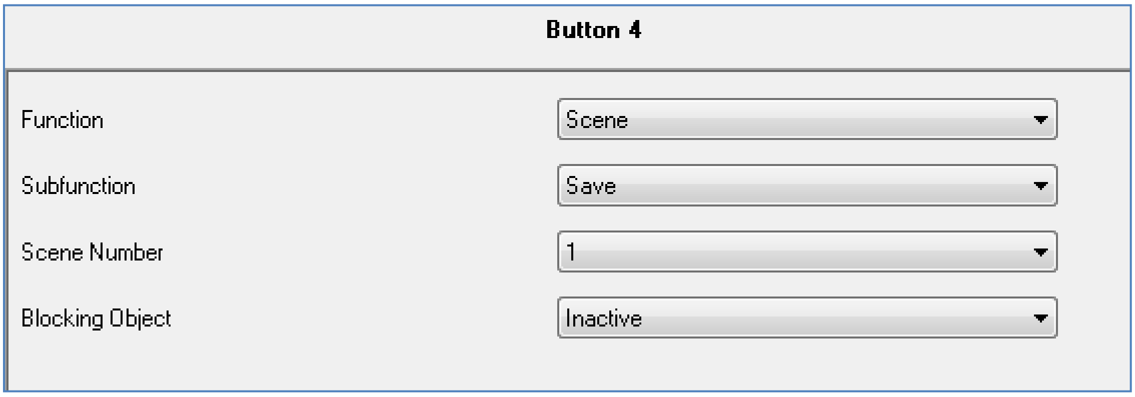
Number	Name	Length	Usage
0	Switch	1 Bit	Switching function; no differences between long and short keystroke

Table 25: Communication object send value with delay

4.5.2 Scene

The scene function calls scenes, which are saved in actuators. Scene numbers in the push button and the actuators must be identical. It is possible to save scenes by a long keystroke if the saving function was activated.

The following illustration shows the setting options for this parameter:



Button 4

Function	Scene
Subfunction	Save
Scene Number	1
Blocking Object	Inactive

Figure 17: Parameter Scene

The following chart shows the dynamic range of this parameter:

Sub-function	Dynamic range **[default value]**	comment
Saving function	▪ No save ▪ **Save**	Saving function is selected by a long keystroke
Scene number	1-64 [1]	Scene number must be identical with the one in the actuators
Blocking object	▪ **Inactive** ▪ Active	have a look at 4.3.1 blocking object

Table 26: sub-function scene

The chart shows the communication objects for this parameter:

Number	**Name**	**Length**	**Usage**
2	Scene	1 Byte	calls the depending scene

Table 27: Communication object Parameter scene

The scene function calls scenes, which were stored in actuators. Scenes contain of parameterized states of several actuators, which can be called with only one keystroke by using the scene function. Additional to the call of scenes, scenes can be saved at the call of a push button by a long keystroke. When the saving function was activated, a long keystroke at the push button saves the current state of the actuators to the depending scene.

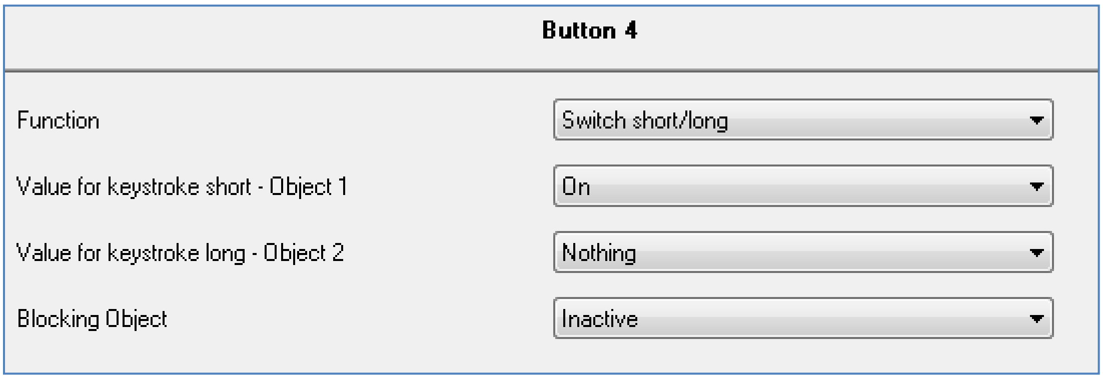
For calling a scene or saving a new value for the scene, you have to send the accordingly code to the relevant communication object for the scene:

Scene	Retrieve		Save	
	Hex.	Dez.	Hex.	Dez.
1	0x00	0	0x80	128
2	0x01	1	0x81	129
3	0x02	2	0x82	130
4	0x03	3	0x83	131
5	0x04	4	0x84	132
6	0x05	5	0x85	133
7	0x06	6	0x86	134
8	0x07	7	0x87	135
9	0x08	8	0x88	136
10	0x09	9	0x89	137
11	0x0A	10	0x8A	138
12	0x0B	11	0x8B	139
13	0x0C	12	0x8C	140
14	0x0D	13	0x8D	141
15	0x0E	14	0x8E	142
16	0x0F	15	0x8F	143
17	0x10	16	0x90	144
18	0x11	17	0x91	145
19	0x12	18	0x92	146
20	0x13	19	0x93	147
21	0x14	20	0x94	148
22	0x15	21	0x95	149
23	0x16	22	0x96	150
24	0x17	23	0x97	151
25	0x18	24	0x98	152
26	0x19	25	0x99	153
27	0x1A	26	0x9A	154
28	0x1B	27	0x9B	155
29	0x1C	28	0x9C	156
30	0x1D	29	0x9D	157
31	0x1E	30	0x9E	158
32	0x1F	31	0x9F	159

Table 28: Calling and saving scenes

4.5.3 Switch short/long

The parameter switch short/long can assign the push button different switching processes for a long and a short keystroke.
The following illustration shows the sub-functions for this parameter:



Button 4

Function: Switch short/long

Value for keystroke short - Object 1: On

Value for keystroke long - Object 2: Nothing

Blocking Object: Inactive

Figure 18: Parameter switch short/long

The sub-functions for this parameter are shown in the chart below:

Sub-function	Dynamic range **[default value]**	comment
Value for keystroke short - Object 1	▪ On ▪ **Off** ▪ Toggle ▪ Send value ▪ Nothing	Action for a short keystroke
Value for keystroke long - Object 2	▪ On ▪ Off ▪ Toggle ▪ Send value ▪ **Nothing**	Action for a long keystroke
Blocking object	▪ **Inactive** ▪ Active	have a look at 4.3.1 blocking object

Table 29: Sub-functions parameter switch short/long

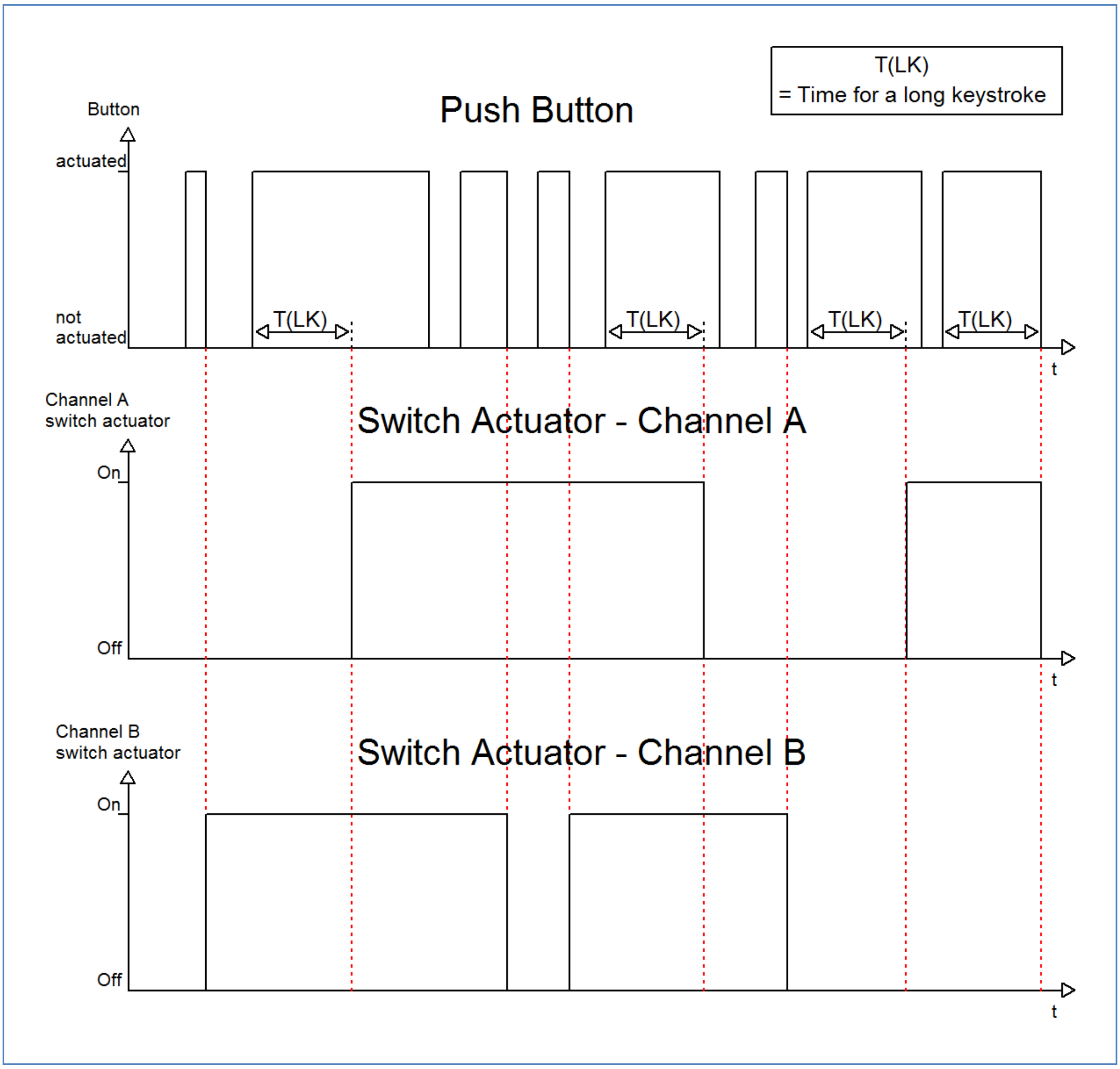
The chart shows the associated communication objects:

Number	**Name**	**Length**	**Usage**
0	push-button short	1 Bit/1 Byte	Switching function short keystroke
2	push-button long	1 Bit/ 1 Byte	Switching function long keystroke

Table 30: Communication object parameter switch short/long

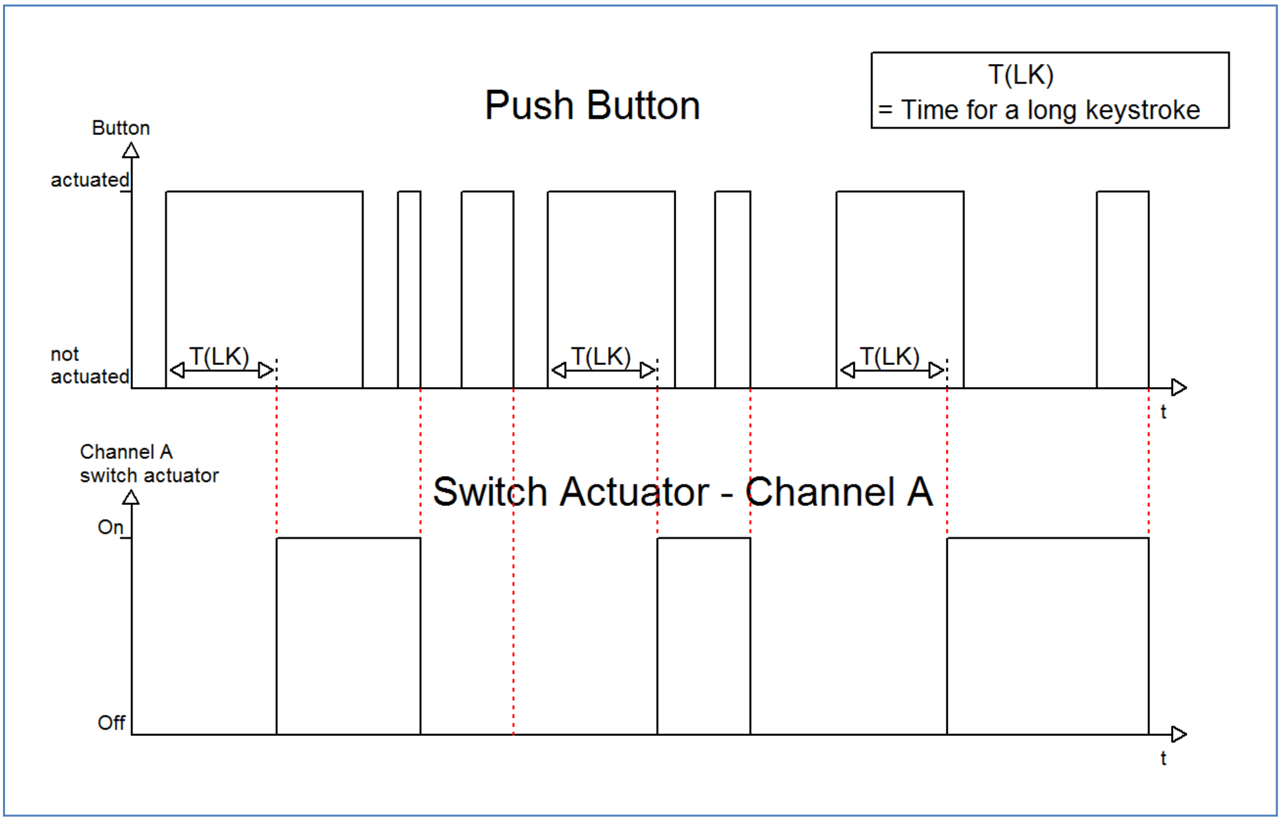
The parameter “switch short/long” can control for example two channels of an actuator by using only one button. Furthermore you can switch a channel with a long keystroke on and with a short keystroke off. For both objects, a function can be set individually. Therefore the sub-functions on, off, toggle and nothing are available. Two communication objects are displayed, which can be connected in any way. By activating the sub-function “toggle” an additional communication object appears, called “value for toggling”. This object is a status object for the push button and must be connected to the status-object of the actuator (have a look at: 4.5.1 Toggle)

The following diagram shows the behavior of this parameter. Both objects (push-button and push-button long) were set to toggle. The object for the long keystroke is connected to channel A of the switch actuator and the object for the short keystroke is connected to channel B:


T(LK)
= Time for a long keystroke
Push Button
Button
actuated
not actuated
T(LK)
T(LK)
T(LK)
T(LK)
t
Switch Actuator - Channel A
Channel A switch actuator
On
Off
t
Switch Actuator - Channel B
Channel B switch actuator
On
Off
t

In this example the push button toggles Channel B with a short keystroke. The Channel A does not react to a short keystroke. This one reacts only at a long keystroke with toggling.

The following diagram shows a further application example for this parameter. In this example, the object for a long keystroke switches the channel A of a switch actuator on. A short keystroke switches the channel off. The three communication objects were connected in only one group address:



If the sub function “Send value” is selected, the following additional settings appear:

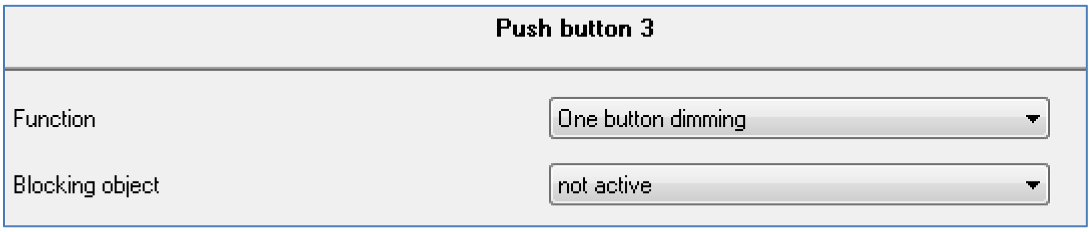
Sub-function	Dynamic range [default value]	comment
Value for keystroke short/long	Send value	chosen sub-function: Send value
Send value	▪ **1 Byte-Value [0...255]** ▪ Scene number	Selection of the value, which shall be sent
1 Byte-Value [0…255]	0-255 [0]	Selection of the byte value, which shall be sent if byte value is chosen
Scene number	1-64 [1]	Selection of the scene number, which shall be sent if scene number is chosen

Table 31: Sub function Send value at switch short/long

Any value can be sent for the sub function „Send value“ at a short/long keystroke. As well scenes can be called as any byte value can be sent. So it is for example possible to call different scenes for a long and a short keystroke or sending absolute height/brightness commands.

4.5.4 One button Dimming

At the dimming function for the single channels, the dimming process is proceeded by only one channel.



Push button 3

Function	One button dimming
Blocking object	not active

Figure 19: Parameter one-button dimming

At the following chart, the sub functions for this parameter are shown:

Sub-function	Dynamic range **[default value]**	comment
Blocking object	▪ **Inactive** ▪ Active	have a look at 4.3.1 blocking object

Table 32: Sub function one-button dimming

The chart shows the available communcication objects:

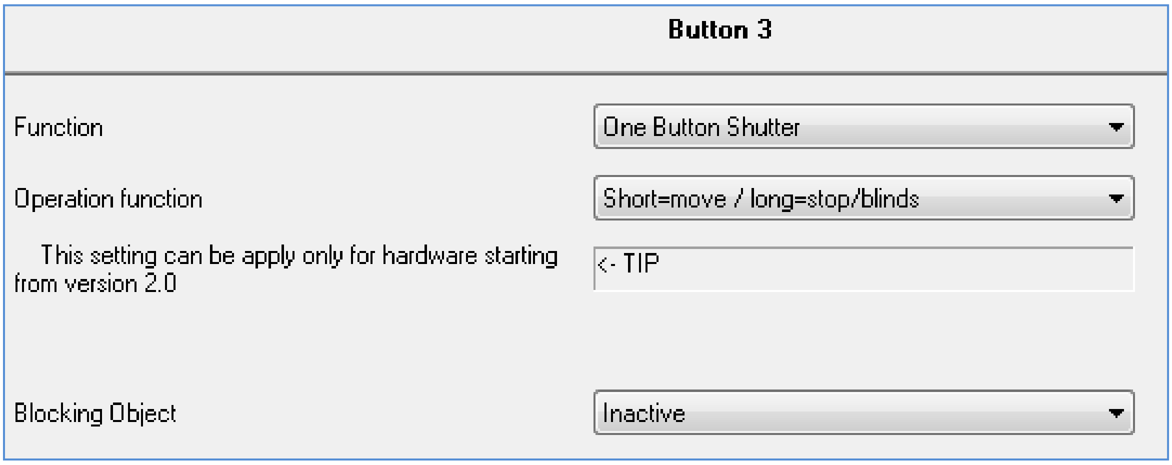
Number	**Name**	**Length**	**Usage**
0	Dimming on/off	1 Bit	Switching function for the dimming process; action for the short keystroke
1	Dimming	4 Bit	dimming function; action for a long keystroke
2	Value for toggle	1 Bit	status object, must be connected with the status function of the actuator for getting feedback of the current switching process

Table 33: Communication objects one-button dimming

At the one-button dimming, the dimming process is executed by one single channel. So it is possible to dim the lights via only one button.
By a long keystroke the communication “Dimming” is called, which is responsible for the dimming process and by a short keystroke the object “Dimming on/off” is called which is responsible for the switching.
The dimming direction is toggled by every keystroke, so if you have dimmed darker, the next time will be dimmed brighter and vice versa.
The one-button dimmeing is a start stop dimming, that means when the dimming function is active a darker or brighter command is sent until the button is released again. After releasing the button a stop command is sent, which stops the dimming process. The dimming step is set fixed to 100%. So with only one button activation the lights can be dimmed from 0% to 100% or from 100% to 0%.

4.5.5 One-button Shutter

The shutter function for the unique channels, often called one-surface shutter, performs the shutter-function by using only one channel.



Button 3

Function: One Button Shutter

Operation function: Short=move / long=stop/blinds

This setting can be apply only for hardware starting from version 2.0 <- TIP

Blocking Object: Inactive

Figure 20: Parameter one-button shutter

The sub-functions for this parameter are shown in the chart below:

Sub-function	Dynamic range **[default value]**	comment
Blocking object	▪ **Inactive** ▪ Active	have a look at 4.3.1 blocking object

Table 34: Sub-functions one-button shutter

The chart shows the communication objects for this parameter:

Number	Name	Length	Usage
0	Shutter	1 Bit	Driving function of the shutter, action for a long keystroke
1	Blinds/Stop	1 Bit	Stop/ Adjustment of blinds; action for a short keystroke
2	Value for change of direction	1 Bit	Shows the last driving command

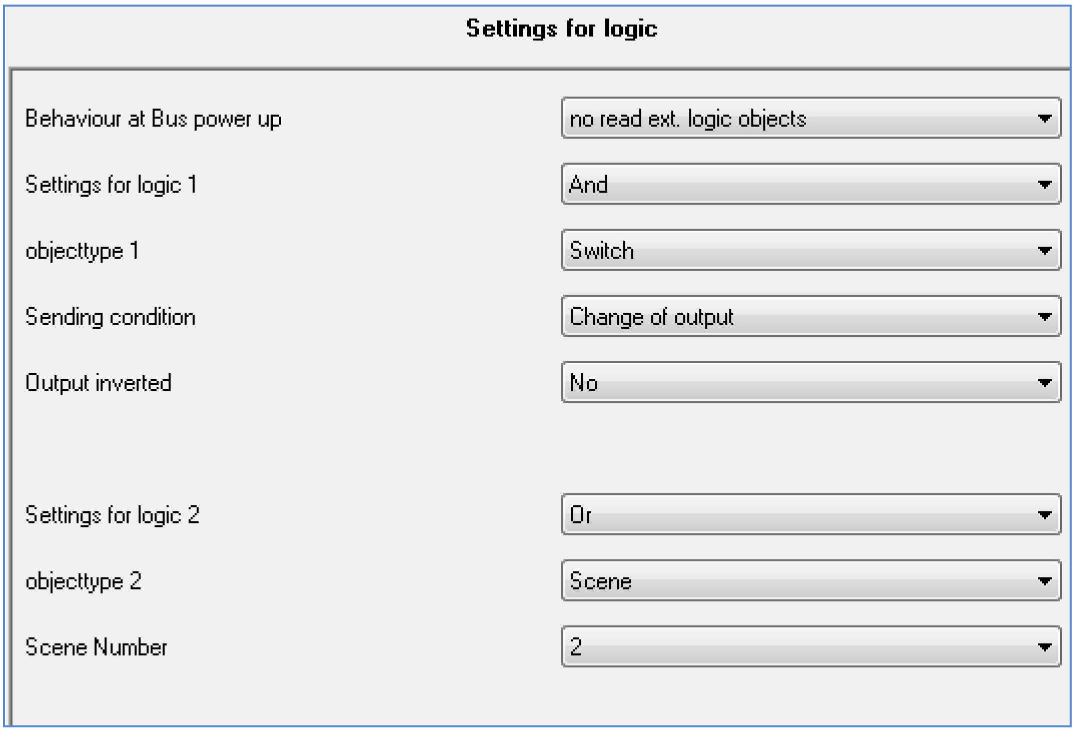
Table 35: Communication objects one-button dimming

The one-surface dimming is performed by using only one channel. The communication object “Shutter” is addressed by a long keystroke and performs the up- and down-movement of the shutter. The direction of movement depends to the last direction of movement. If the shutter were driven up at the last time, they will be driven down at the next time. So the direction of movement changes after every movement.
The communication object “Blinds/Stop” is addressed by a short keystroke. Addressing this object stops a running movement of the shutter. Furthermore it will adjust the blinds if a shutter function is selected for this channel. The direction of the adjustment changes also here after every movement in the same way like the up/down moving of the shutter.
From hardware version 2.0 (have a look at the print oft eh side of the device: R:X.X), it is possible to switch the functions for the short and the long keystroke. So it can be chosen whether a short or a long keystroke shall drive the shutter/blinds. The Stop-/ Adjustment object gets the other operating concept.
The object “Value for change of direction” serves as state object. It must be connected to the direction object of the actuator. So the button sends always the complementary value as before.

4.6. Logic (only at the plus variant)

The functions, described in this segment (4.6), are only available at the plus variant.

The push buttons contain of 4 individually switchable and parameterize able logic blocks. At the following page, the logic blocks can be activated and the general settings can be made:



Settings for logic

Behaviour at Bus power up: no read ext. logic objects

Settings for logic 1: And

objecttype 1: Switch

Sending condition: Change of output

Output inverted: No

Settings for logic 2: Or

objecttype 2: Scene

Scene Number: 2

Figure 21: Activation logic blocks

The following parameter can be adjusted once and is valid for all of the 4 logic blocks:

Sub-function	Dynamic range **[default value]**	comment
Behavior at bus power up	▪ **no read ext. logic objects** ▪ read ext. logic objects	sub-function indicates whether the external logic objects should be read or not at a bus power up

Table 36: Common Parameter logic blocks

If the read of the external logic at bus power up is activated, the status of all external logic objects will be read at a bus power up. So the logic operation is evaluated new. If this function is not active, the push button will hold the status before bus power outage.

The Chart shows the setting options for the logic blocks. The logic blocks can be assigned a logic function and an object type, the usage of this logic block:

Setting per logic **[default value]**	Dynamic range **[default value]**	comment
- **disabled** - And - Or	- **Switch** - Scene	Every logic block can be adjusted as And- or as Or-function. Additional the object type (usage) can be adjusted for every block.

Table 37: Dynamic range logic

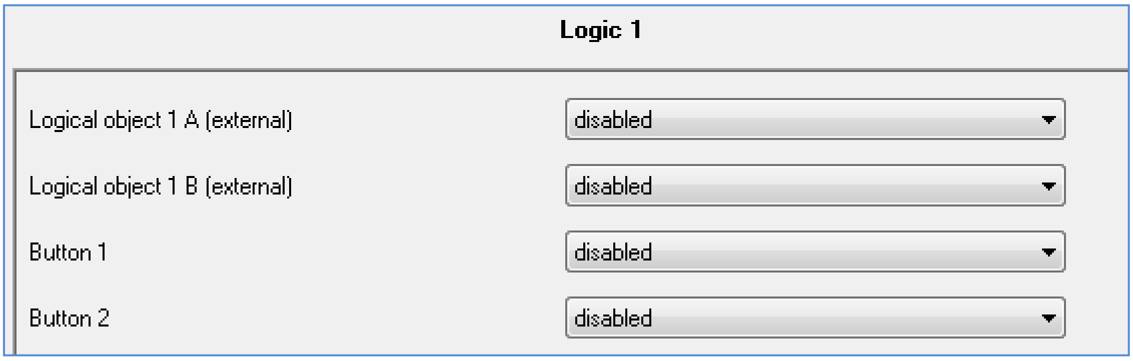
The following chart shows the communication objects for the logic functions:

Number	Name	Length	Usage
80	Logic input 1A	1 Bit	Communication object for an external logic; is only displayed when an external logic was activated
81	Logic input 1B	1 Bit	the same like logic input 1A
82	Logic Output 1	1 Bit	Output logic for switch is activated (=1-signal) when the logic block is true
82	Logic Output 1 Scene	1 Byte	Output logic for scenes is activated (=1-signal) when the logic block is true

Table 38: Communication objects logic

The communication objects for the other 3 possible logic blocks are the same like the first one. Three numbers are reserved for every logic block, so the next logic block starts at number 83.

As soon as a logic block is activated, a new sub-menu appears at the left selection list. In this menu can be set, which buttons should be connected to the logic block. Two external logic blocks can be activated additional. The external logic objects can be connected to communication objects of other devices by using the displayed communication objects “logic input 1 A&B”.



Logic 1

Logical object 1 A (external): disabled

Logical object 1 B (external): disabled

Button 1: disabled

Button 2: disabled

Figure 22: Setting logic

The read of the inputs (number depends to the device type) can be activated for every channel and two external objects. They can be read normal or inverted.

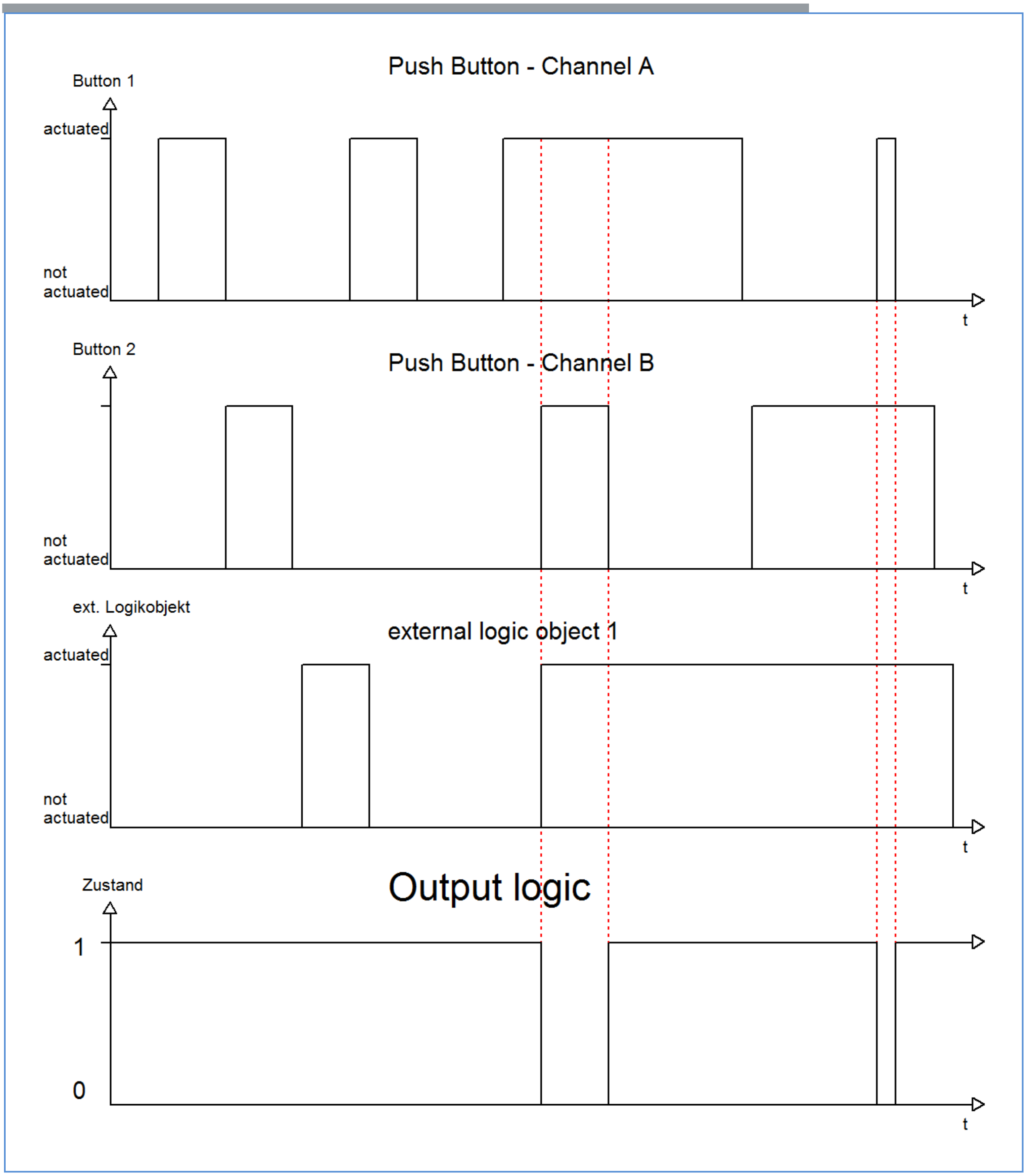
4.6.1 Logic object type switch

The chart shows the possible sub-functions for the logic sub-function switch:

Sub-function	Dynamic range **[default value]**	comment
Sending condition	▪ **not automatic** ▪ change of input ▪ change of output	Adjustment indicates, when the state of the logic block should be sent
Output inverted	▪ **No** ▪ Yes	Adjustment indicates, whether the output should be inverted or not

Table 39: Logic sub-function switch

The sending condition adjusts, when the push button should send a signal on the bus. By adjusting the sending condition “change of input”, the push button sends a signal at every change of any input whether that causes a change of the logic operation or not. The setting “change of output” causes that the push button sends only a signal when the logic changes its current status.
The sub-function Output inverted indicates whether the output signal should be issued inverted (that means reversed 1->0 and 0->1) or normal.
The following diagram shows the logic operation switch as an and-function. The logic reads in this example the channels A and B as well as an external logic object. The Output is inverted:



The logic function is only satisfied, when button A and B as well as the external logic object have a 1-signal. The inversion of the output causes that the output is switched on, when the logic is not satisfied and switched off, when the logic is satisfied.

4.6.1 Logic object type scene

This logic function calls scenes, when the logic function is satisfied.

The chart shows the possible sub-functions for the logic operation scene:

Sub-function	Dynamic range **[default value]**	comment
Scene number	1-64 [2]	Scene number must be the same like the one you want to call with the logic-function

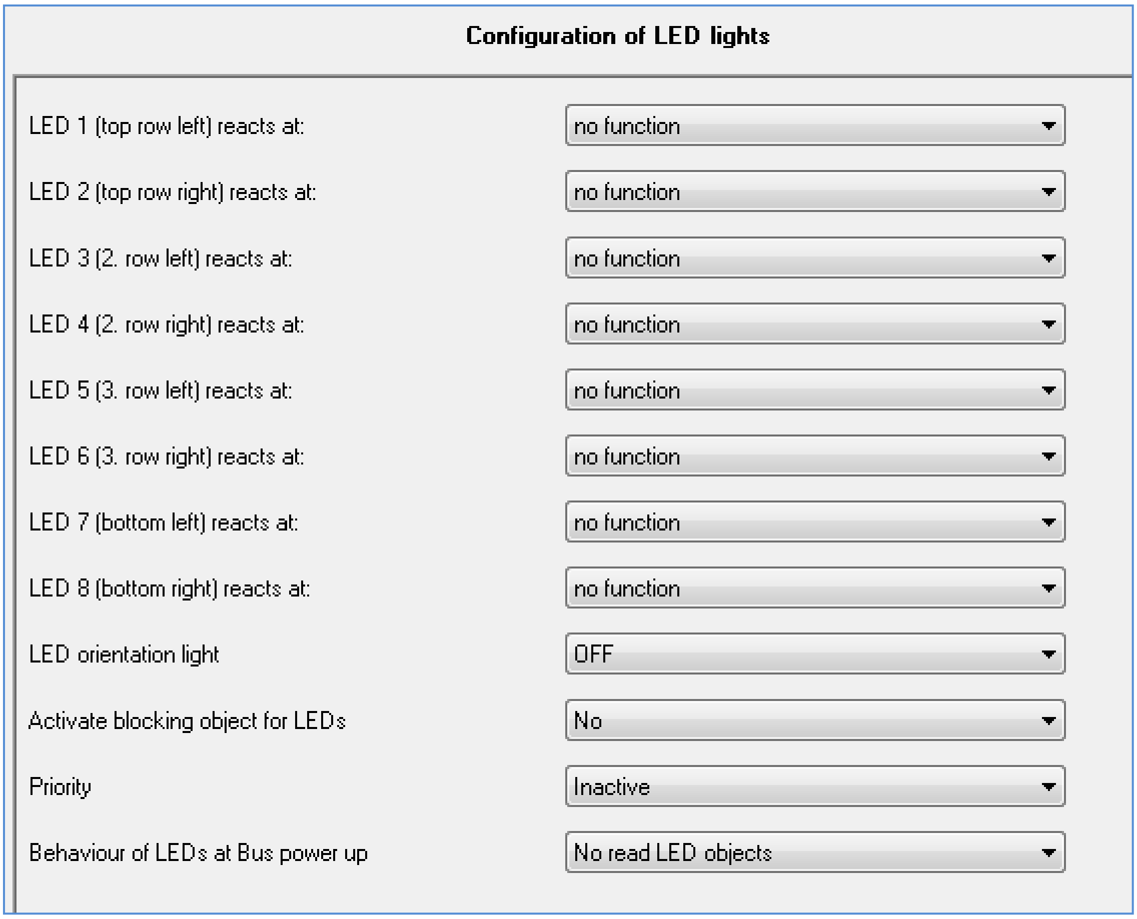
Table 40: Logic sub-function scene

The logic function for the scenes works like a normal logic function. As soon as the logic function is satisfied, the communication object will send the adjusted scene-number. The communication object has the length of 1 Byte, so that it can be connected to other communication objects of scenes.
All sub-functions, like in a normal logic function can be parameterized. So you can set the logic function as an AND- or an OR-function and connect all inputs of the push button and additional 2 external logic objects to the logic function.

4.7 LED lights (only at the plus variant)

The functions, described in this segment (4.7), are only available at the plus variant.

The LED display can visualize different switching processes and keystrokes. Every LED can light green or red. You can also parameterize when the LE D should light green and when red.
The Illustration shows the configuration of the LED display:



Configuration of LED lights

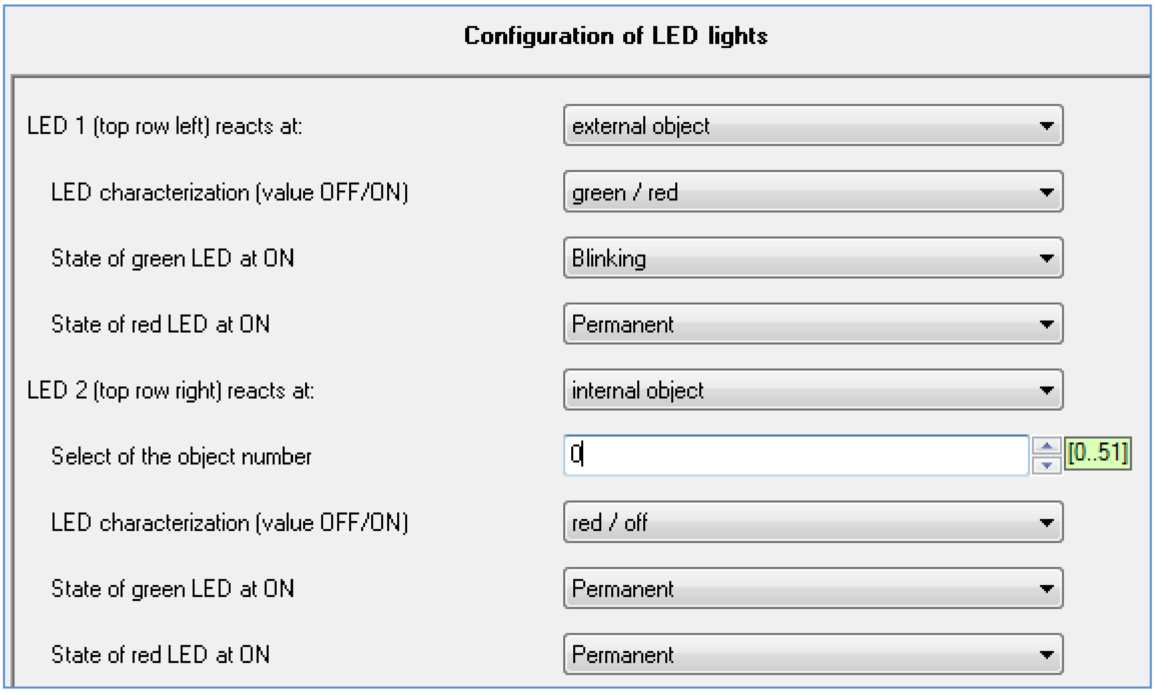
LED 1 (top row left) reacts at:	no function
LED 2 (top row right) reacts at:	no function
LED 3 (2. row left) reacts at:	no function
LED 4 (2. row right) reacts at:	no function
LED 5 (3. row left) reacts at:	no function
LED 6 (3. row right) reacts at:	no function
LED 7 (bottom left) reacts at:	no function
LED 8 (bottom right) reacts at:	no function
LED orientation light	OFF
Activate blocking object for LEDs	No
Priority	Inactive
Behaviour of LEDs at Bus power up	No read LED objects

Figure 23: Configuration LED display

The push buttons (at the plus variant) contains of one LED per button and one orientation LED. Additional a blocking object for the LEDs can be shown, which blocks all LEDs. The parameterization of the LEDs is described in the following segments.

4.7.1 LEDs per button

The following illustration shows the setting options for the LEDs:



Configuration of LED lights

LED 1 (top row left) reacts at:	external object
LED characterization (value OFF/ON)	green / red
State of green LED at ON	Blinking
State of red LED at ON	Permanent
LED 2 (top row right) reacts at:	internal object
Select of the object number	0 [0..51]
LED characterization (value OFF/ON)	red / off
State of green LED at ON	Permanent
State of red LED at ON	Permanent

Figure 24: Configuration LEDs per button

The following chart shows the dynamic range for the setting of the LEDs:

Sub-function	Dynamic range **[default value]**	comment
LED X reacts at:	▪ **no function** ▪ external object ▪ internal object ▪ button activation	Adjustment of the switching/toggling condition of the depending LED
Select of the object number	0-51 [0]	Adjustment of the internal connection. Only displayed, when LED should react to an internal object.
LED characterization (Value OFF/ON)	▪ **off/green** ▪ off/red ▪ green/red ▪ red/green ▪ green/off ▪ red/off	indicates the behavior of the LED when switched on and switched off
State of green LED at ON	▪ **permanent** ▪ blinking	Adjustment of the luminescent behavior of the green LED, when switched on.
State of red LED at ON	▪ **permanent** ▪ blinking	Adjustment of the luminescent behavior of the red LED, when switched on.

Table 41: Dynamic range LEDs per button

The parameter “LED (1-8) reacts to” can be adjusted when the LED should switch on or toggled. This 4 setting options are available and cause the following operations:

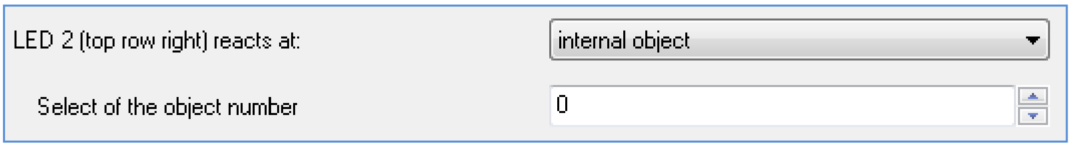
- **no function**
  The LED is switched off and cannot be controlled. So there are no following parameterization options for this LED.
- **external object**
  If the LED should react to an external object, a communication object will be shown for this LED. The communication object can be connected to any group address afterwards. So the LED can also show a switching process of an actor, which is independent from the push button.
  The chart shows the according communication object:

Number	Name	Length	Usage
	LED 1-2/4/6/8	1 Bit	switch LED

Table 42: Communication object external object LED

The number of the communication object depends to the hardware design (2-fold/4-fold/6-fold/8-fold) and the used LED.

- **internal object**
  The LED can react to every internal communication object. Internal communication objects are called the objects of the push button. If this function is activated, the following window will appear:



LED 2 (top row right) reacts at: internal object

Select of the object number 0

Figure 25: Configuration internal connection LED

Because there is already a fixed connection between the LED and a communication object, now further communication object is necessary. The LED can be connected to every object independent from the size of the object.

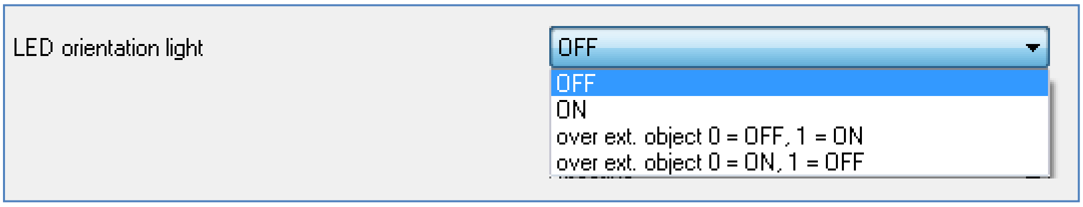
- **button activation**
  By choosing this action, the LED reacts to every activation of the associated button. The action how the LED should react to an activated/inactivated button can be parameterized individual by the function LED characterization. The value for “on” will be send when the button is activated and the value for “off” when it is not activated.

Further can be set for the LED when they should switch the green light on/off and when the right light on/off. This behavior can be adjusted by the parameter “LED characterization”. The dynamic range is shown in the chart 33 (former side). Thereby the first value stands for the switched state and the second for the deactivated state.
Additional the luminescent behavior can be set of every LED individual by the parameter “State of green/red LED at ON”. Every LED can shine permanent or flashing.

4.7.2 Orientation LED/light

The push button contains additional to the LEDs per button of one LED orientation light. This orientation light can serve as an orientation light or being controlled by an external object. The orientation light shines standardly green.

The following illustration shows the setting option for this paramter:



LED orientation light: OFF

OFF
ON
over ext. object 0 = OFF, 1 = ON
over ext. object 0 = ON, 1 = OFF

Figure 26: Configuration LED orientation light

The following chart shows the dynamic range of this parameter:

Sub-function	Dynamic range **[default value]**	comment
LED orientation light	▪ **OFF** ▪ ON ▪ over ext. object 0=OFF, 1=ON ▪ over ext. object 0=ON, 1=OFF	Adjustment of the controlling and luminescent behavior of the orientation light

Table 43: Dynamic range LED orientation light

Four choices are available for the controlling of the orientation light. Firstly the LED can be switched permanent off. So the LED is deactivated and has no further functions. If the LED should be used as orientation light, the setting ON switches the LED permanent on.
Furthermore the orientation LED can be controlled by an external object. For this setting can be additional adjusted at which signal the LED should switch on. By activation the controlling by an external object, an additional communication object will be shown. This communication object can be controlled of any device.
The following chart shows the communication object for the controlling by an external object:

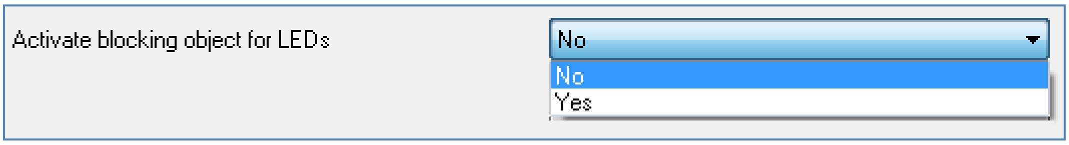
Number	**Name**	**Length**	**Usage**
30/40/50/60	LED orientation light	1 Bit	switch LED

Table 44: Communication object LED orientation light

4.7.3 Blocking object for LEDs

Analogous to the blocking objects for the channels, a blocking object for the LEDs can be shown. This blocking object blocks all LEDs, when it is triggered.

The following illustration shows the setting options for this parameter:



Activate blocking object for LEDs: No

Figure 27: Blocking object LEDs

The following chart shows the dynamic range of this parameter:

Sub-function	Dynamic range **[default value]**	comment
Activate blocking object for LEDs	▪ **No** ▪ Yes	Activation of the blocking object for the LEDs

Table 45: Dynamic range blocking object LEDs

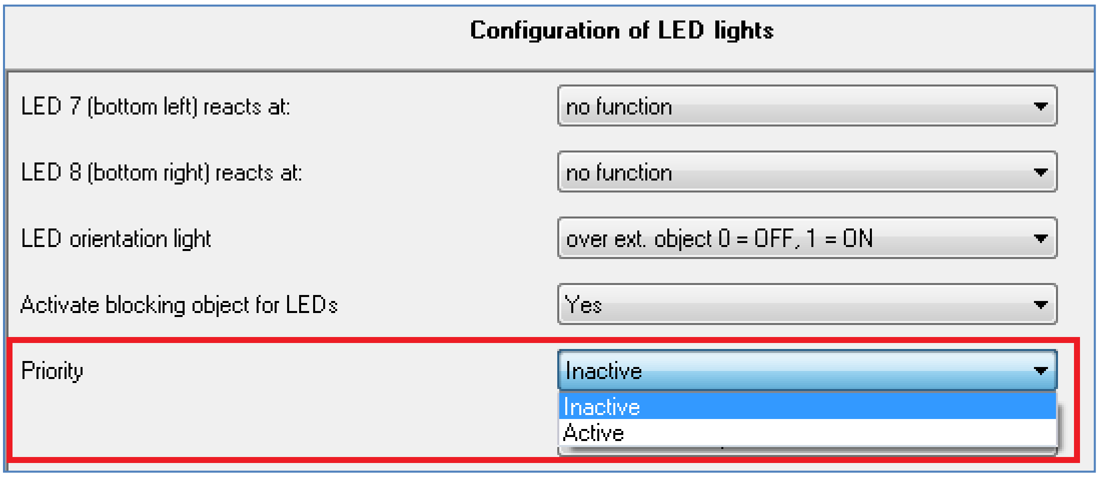
In difference to the blocking objects per channel, there is only one blocking object for the LEDs, which blocks all LEDs. When the LED blocking object is triggered, that means the blocking object becomes a logical "1", all LEDs are blocked and cannot be controlled while the blocking function is active. The LEDs, which were switched on before the blocking process, are switched off. By sending a logical "0", the blocking process is deactivated. Now it is possible to control the LEDs as usual.
The chart shows the associated communication object:

Number	**Name**	**Length**	**Usage**
31/41/51/61	LED blocking object	1 Bit	blocks all LEDs

Table 46: Communication object blocking LEDs

4.7.4 LED priority

The LED priority can allocate every LED, except the orientation light, a certain behavior at the activation of one of the both priority objects.
To configure the LED priority, you have to activate this function at the LED configuration:

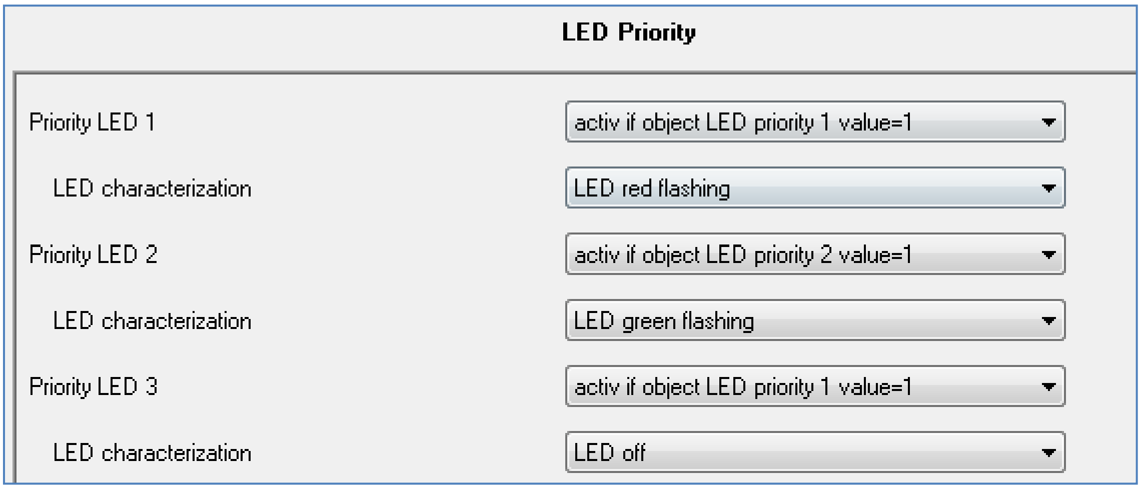


Configuration of LED lights

LED 7 (bottom left) reacts at:	no function
LED 8 (bottom right) reacts at:	no function
LED orientation light	over ext. object 0 = OFF, 1 = ON
Activate blocking object for LEDs	Yes
Priority	Inactive

Figure 28: Activation LED priority

When the LED priority is activated, a sub menu "LED priority" appears at the left drop-down menu. The further parameterization can be done at this submenu:



LED Priority

Priority LED 1	activ if object LED priority 1 value=1
LED characterization	LED red flashing
Priority LED 2	activ if object LED priority 2 value=1
LED characterization	LED green flashing
Priority LED 3	activ if object LED priority 1 value=1
LED characterization	LED off

Figure 29: Sub menu LED priority

There is a parameterization option for every LED (except the orientation light) at this sub menu. Every LED can react either to the first priority object or to the second. You can also set whether the Led should react to a 0-signal or a 1-signal of the priority object.

The dynamic range of the LED priority is shown in this chart:

Sub-function	Dynamic range **[default value]**	comment
Priority LED 1-2/4/6/8	▪ **not active** ▪ active if object LED priority 1 value=1 ▪ active if object LED priority 1 value=0 ▪ active if object LED priority 2 value=1 ▪ active if object LED priority 2 value=0	Activation of the LED priority for the single LEDs

Table 47: LED priority

If the LED priority was activated for one LED, that means another setting than “not active” was chosen, a new parameter appears at which the LED characterization can be set.

The dynamic range for the LED characterization is shown at the following chart:

Sub-function	Dynamic range **[default value]**	comment
LED characterization	▪ **LED Off** ▪ LED red ▪ LED red flashing ▪ LED green ▪ LED green flashing	Adjustment of the LED characterization for an activated LED priority

Table 48: LED-characterization at priority

The following chart shows the relevant communication objects for this parameter:

Number	**Name**	**Length**	**Usage**
32/42/52/62	LED priority 1	1 Bit	switch priority 1
33/43/53/63	LED priority 2	1 Bit	switch priority 2

Table 49: Communication objects LED priority

5 Index

5.1 Register of illustrations

5.2 List of tables

6 Attachment

6.1 Statutory requirements

The above-described devices must not be used with devices, which serve directly or indirectly the purpose of human, health- or lifesaving. Further the devices must not be used if their usage can occur danger for humans, animals or material assets.
Do not let the packaging lying around careless, plastic foil/ -bags etc. can be a dangerous toy for kids.

6.2 Routine disposal

Do not throw the waste equipment in the household rubbish. The device contains electrical devices, which must be disposed as electronic scrap. The casing contains of recyclable synthetic material.

6.3 Assemblage

⚠ **Risk for life of electrical power!**
All activities on the device should only be done by an electrical specialist. The county specific regulations and the applicable EIB-directives have to be observed.

6.4 4-Bit Dim command

The following chart describes the 4-Bit dimming command:

Decimal	Hexadecimal	Binaer	Dim command
0	0	0000	Stop
1	1	0001	100% Darker
2	2	0010	50% Darker
3	3	0011	25% Darker
4	4	0100	12,5% Darker
5	5	0101	6,25% Darker
6	6	0110	3,13% Darker
7	7	0111	1,56% Darker
8	8	1000	Stop
9	9	1001	100% Brighter
10	A	1010	50% Brighter
11	B	1011	25% Brighter
12	C	1100	12,5% Brighter
13	D	1101	6,25% Brighter
14	E	1110	3,13% Brighter
15	F	1111	1,56% Brighter

6.5 Datasheet