

Technical Manual

MDT Binary Inputs



BE-08000.01

BE-08024.01

BE-08230.01

BE-16000.01

BE-16024.01

BE-16230.01

1 Content

1 Content.....	2
2 Overview.....	4
2.1 Overview devices.....	4
2.2 Exemplary circuit diagrams	4
2.3 Structure & Handling	8
2.4 Functions.....	8
2.4.1 Overview of the functions	9
2.6 Starting up	10
3 Communication objects	11
3.1 Communication objects per channel.....	11
3.2 Communication objects logic	12
3.3 Default settings of the communication objects	13
4 Reference ETS-Parameter	14
4.1 General Settings	14
4.2 Configuration.....	16
4.3 Identical parameter	17
4.3.1 Blocking object	17
4.4 Parameter Channels grouped.....	17
4.4.1 Dimming	18
4.4.2 Shutter	20
4.4.3 Switch	21
4.5 Parameters channels unique.....	22
4.5.1 Switch	22
4.5.2 Scene	32
4.5.3 Counter.....	34
4.5.4 Switch short/long	36
4.5.5 One button Dimming.....	39
4.5.6 One-button Shutter	40
4.6. Logic.....	41
4.6.1 Logic object type switch	43
4.6.1 Logic object type scene	45
5 Index	46
5.1 Register of illustrations.....	46
5.2 List of tables.....	47

6 Attachment.....	48
6.1 Statutory requirements.....	48
6.2 Routine disposal.....	48
6.3 Assemblage.....	48
6.4 4-Bit Dim command.....	49
6.5 Datasheet.....	50

2 Overview

2.1 Overview devices

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

- **BE-08000.01** Binary Input 8-fold, 4TE, for floating contacts
- **BE-16000.01** Binary Input 16-fold, 8TE, for floating contacts
- **BE-08024.01** Binary Input 8-fold, 4TE, 12-24V DC/ 24V AC
- **BE-16024.01** Binary Input 16-fold, 8TE, 12-24V DC/ 24V AC
- **BE-08230.01** Binary Input 8-fold, 4TE, 230V AC
- **BE-16230.01** Binary Input 16-fold, 8TE, 230V AC/

2.2 Exemplary circuit diagrams

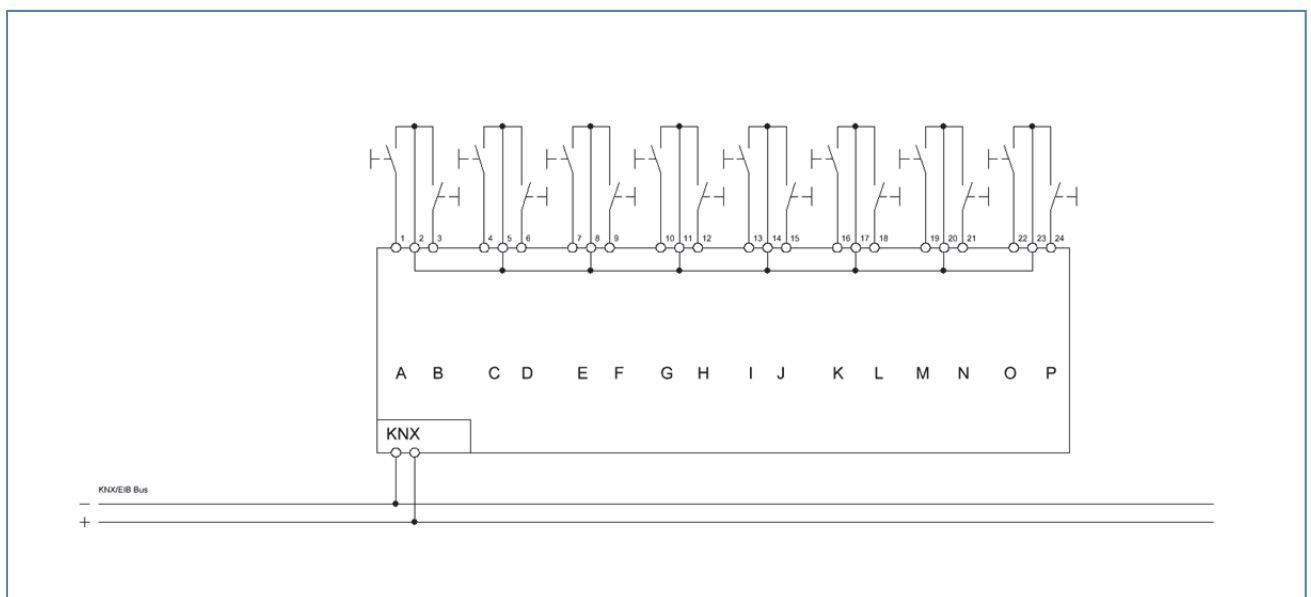


Figure 1: Exemplary circuit diagram BE16000.01 standard configuration for floating contacts

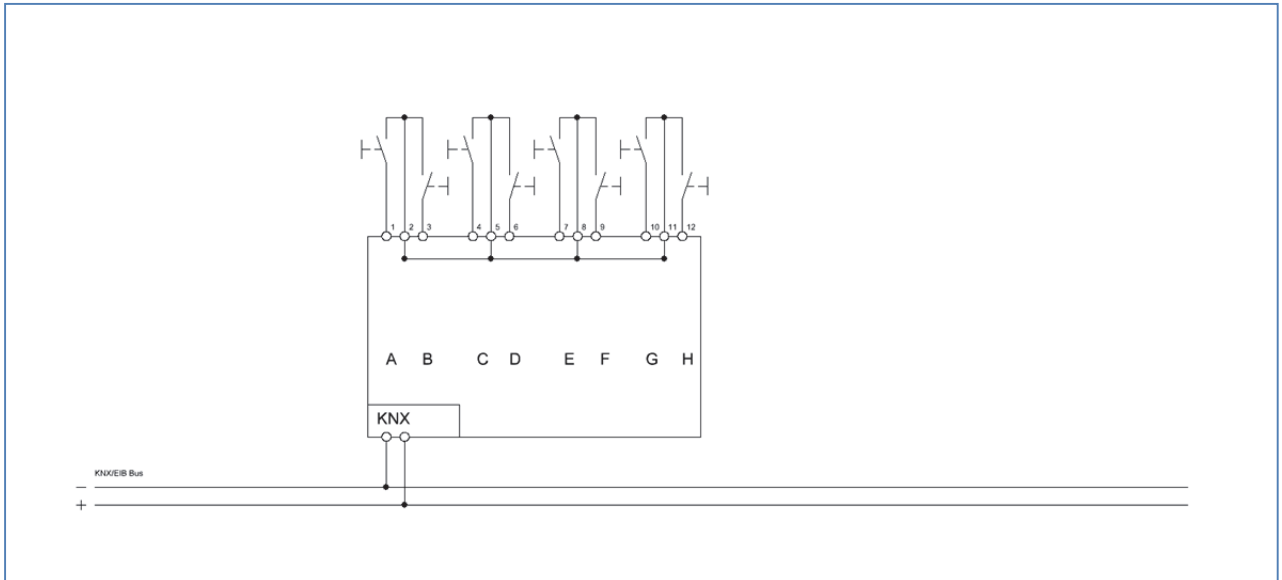


Figure 2: Exemplary circuit diagram BE08000.01 standard configuration for floating contacts

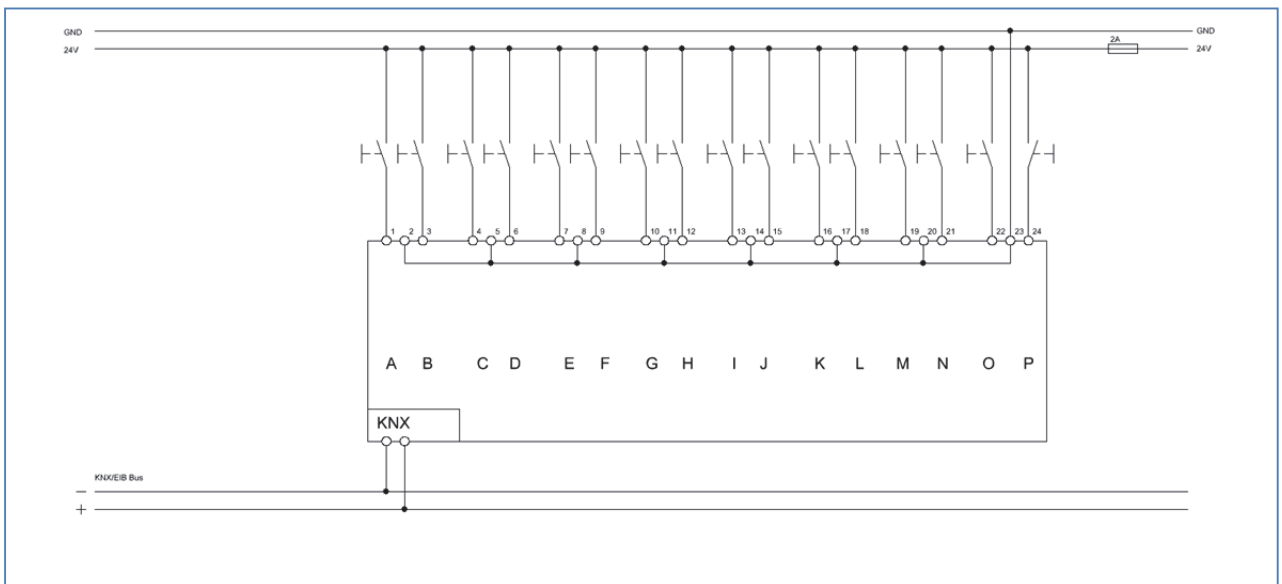


Figure 3: Exemplary circuit diagram BE16024.01 standard configuration for 24V AC/DC input voltage

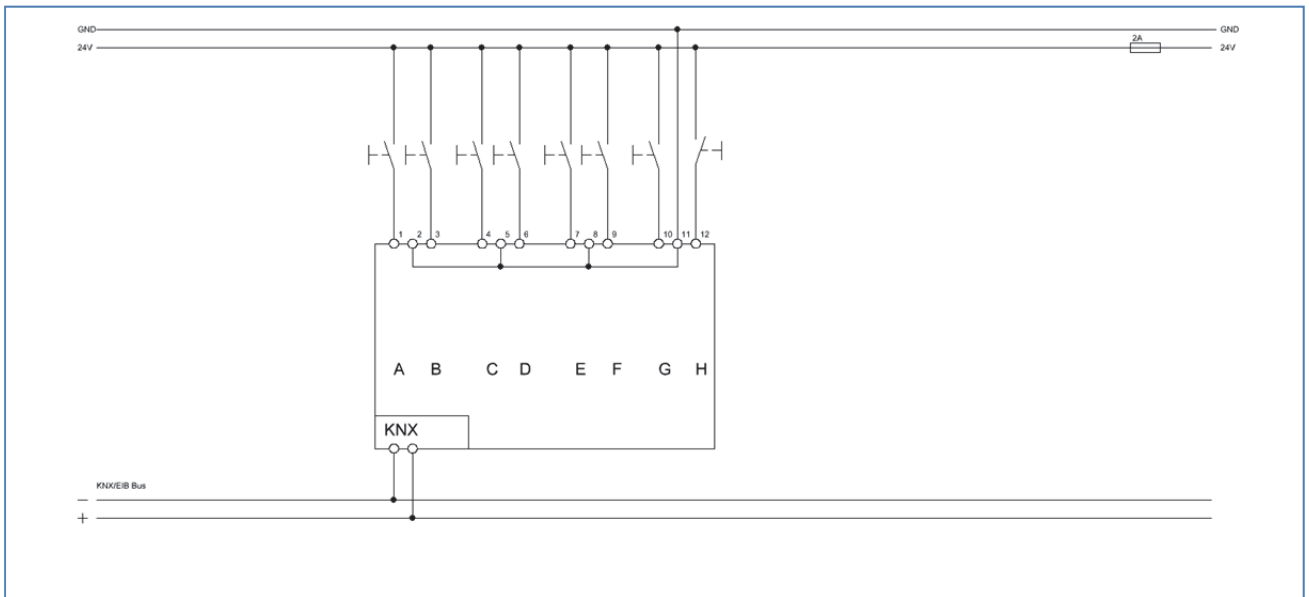


Figure 4: Exemplary circuit diagram BE08024.01 standard configuration for 24V AC/DC input voltage

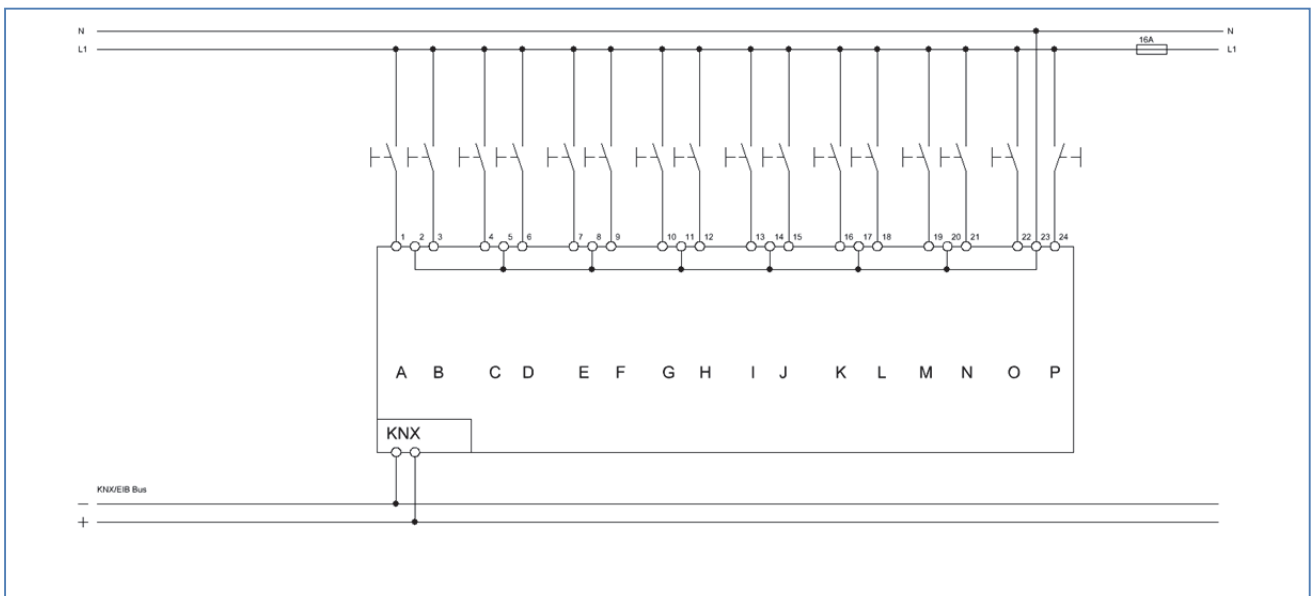


Figure 5: Exemplary circuit diagram BE16230.01 standard configuration for 230V AC input voltage

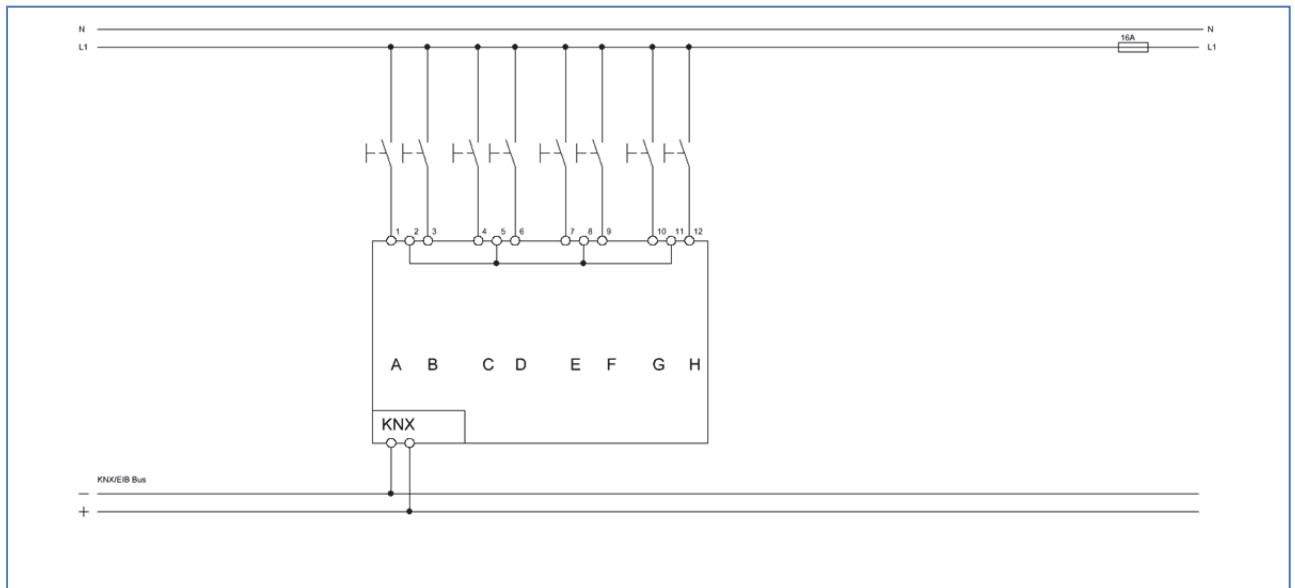


Figure 6: Exemplary circuit diagram BE08230.01 standard configuration for 230V AC input voltage

2.3 Structure & Handling

The Binary Inputs (here: BE-16000.01) contains of one status LED per channel. An activated channel is indicated by a lit LED. The programming function gets activated by the programming button. An activated programming button can be recognized by a lit programming LED.

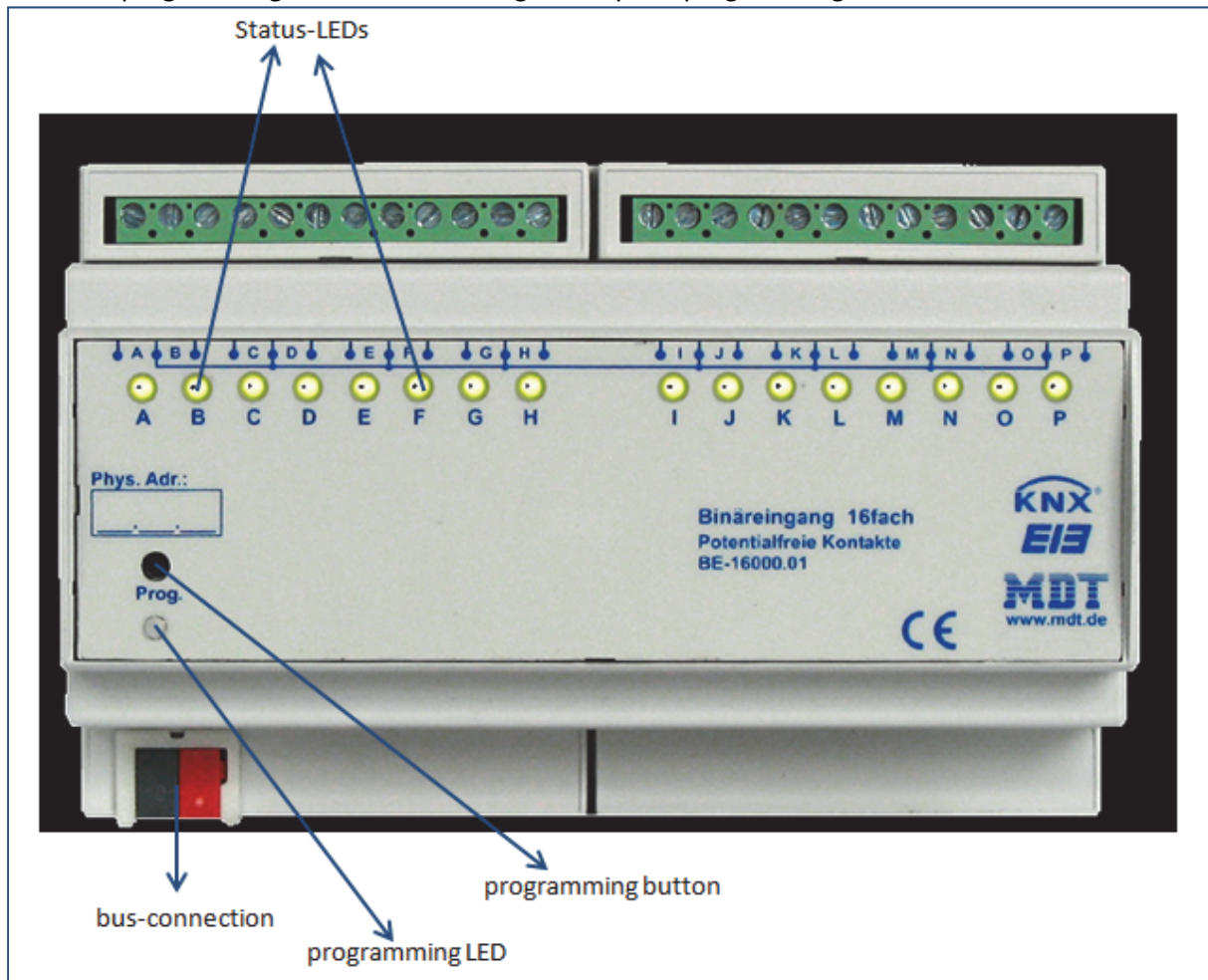


Figure 7: Overview hardware module binary input (BE-16000.01 for floating contacts)

2.4 Functions

The functionality is identical for every channel. The device contains of eight or sixteen channels 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.

There are additional 4 logic functions (and/or) containing of up to two additional input objects.

2.4.1 Overview of the functions

General settings	Debounce time	10-120ms, selectable in steps
	Time for keystroke long	0,1-30s, selectable in steps
	Limitation of telegrams	max. number of telegrams per 10s, can be parameterized freely
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	<ul style="list-style-type: none"> • switching function • toggle function • status function • time functions <ul style="list-style-type: none"> ○ switch on/off delay • edge evaluation • forced settings • sending of byte-values
	Scene function	<ul style="list-style-type: none"> • memory function • selection of different scenes
	Counter function	<ul style="list-style-type: none"> • counting function • edge evaluation • counting steps can be chosen freely
	Switch short/long	<ul style="list-style-type: none"> • On-/Off-/toggle function • short/long independent parameterize able
	One button dimming	<ul style="list-style-type: none"> • steps of dimming • telegram repetition
	One button shutter	<ul style="list-style-type: none"> • shutter function with only one button
Logic functions	AND-function	<ul style="list-style-type: none"> • switching function • scene function • inverting
	OR-function	<ul style="list-style-type: none"> • switching function • scene function • inverting

Table 1: Functional overview binary input

2.5 Settings at the ETS-Software

Selection at the product database:

Manufacturer: MDT Technologies

Product family: Binary Input

Product type: Universal Input

Medium Type: Twisted Pair (TP)

Product name: addicted to the used type, e.g.: BE-16024.01 Input 16-fold, 8TE, 24VAC/DC

Order number: addicted to the used type, e.g.: BE-16024.01

2.6 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.

The communication objects are used for the assignment of the group addresses.

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 counter 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...	C	R	W	T	U
0	Input A / B	Dimming on/off			1 bit	C	R	-	T	-
1	Input A / B	Dimming			4 bit	C	R	-	T	-
10	Input C	Switch			1 bit	C	R	-	T	-
11	Input C	Value for toggle			1 bit	C	-	W	T	U
17	Input D	Scene			1 Byte	C	R	-	T	-
20	Input E	Reset counter			1 bit	C	-	W	-	U
23	Input E	Counter			4 Byte	C	R	-	T	-
25	Input F	Switch			1 bit	C	R	-	T	-
30	Input G / H	Shutter Down/Up			1 bit	C	R	-	T	-
31	Input G / H	Stop/Blinds Open/Close			1 bit	C	R	-	T	-

Figure 8: 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	Reset counter	reset the counter value	DPT 1.001	In, Write
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	Scene	scene control	DPT 18.001	Out, Read
2	Value for change of direction	reversal of direction for shutters	DPT 1.001	Out, Read
2	push-button long	sends action for long keystroke	DPT 1.001	Out, Read
3	Counter	counting	DPT 12.001	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

There are communication objects for the logic function at every binary input 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 binary input and the numbers from 40 at an 8-fold binary input. The first logic block gets assigned the first three numbers, so at an 8-fold input from 40 to 42 and at a 16-fold input from 80 to 82. 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 9: communication objects logic

If a logic block is disabled, no communication objects will be shown. Therefore no addressing is possible. Every binary input 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 Default settings of the communication objects

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

Default settings									
Nr.	Channel/Input	Function	Length	Priority	C	R	W	T	U
0	Input A	Switch	1 Bit	Low	X	X		X	
0	Input A	Shutter	1 Bit	Low	X	X		X	
0	Input A	Send value	1 Byte	Low	X	X		X	
0	Input A	Send forced setting	2 Bit	Low	X	X		X	
0	Input A	Reset counter	1 Bit	Low	X		X		X
0	Input A	Dimming on/off	1 Bit	Low	X	X		X	
0	Input A	push-button short	1 Bit	Low	X	X		X	
0	Input A	push-button short	1 Byte	Low	X	X		X	
0	Input A/B	Switch on/off	1 Bit	Low	X	X		X	
0	Input A/B	Dimming on/off	1 Bit	Low	X	X		X	
0	Input A/B	Shutter down/up	1 Bit	Low	X	X		X	
1	Input A	Value for toggle	1 Bit	Low	X	X		X	
1	Input A	Stop/Blinds open/close	1 Bit	Low	X	X		X	
1	Input A	Dimming	4 Bit	Low	X	X		X	
1	Input A/B	Dimming	4 Bit	Low	X	X		X	
1	Input A/B	Stop/Blinds open/close	1 Bit	Low	X	X		X	
2	Input A	Scene	1 Byte	Low	X	X		X	
2	Input A	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	Input A	Push-button long	1 Bit	Low	X	X		X	
2	Input A	Push-button long	1 Byte	Low	X	X		X	
3	Input A	Counter	4 Byte	Low	X	X		X	
4	Input A	Blocking object	1 Bit	Low	X		X		X
+5	next input								
80/40	Logic input 1 A	Logic input 1 A	1 Bit	Low	X		X		X
81/41	Logic input 1 B	Logic input 1 B	1 Bit	Low	X		X		X
82/42	Logic output 1	Logic output 1	1 Bit	Low	X	X		X	
82/42	Logic output 1 scene	Logic output 1 scene	1 Byte	Low	X	X		X	
+3	next logic block								

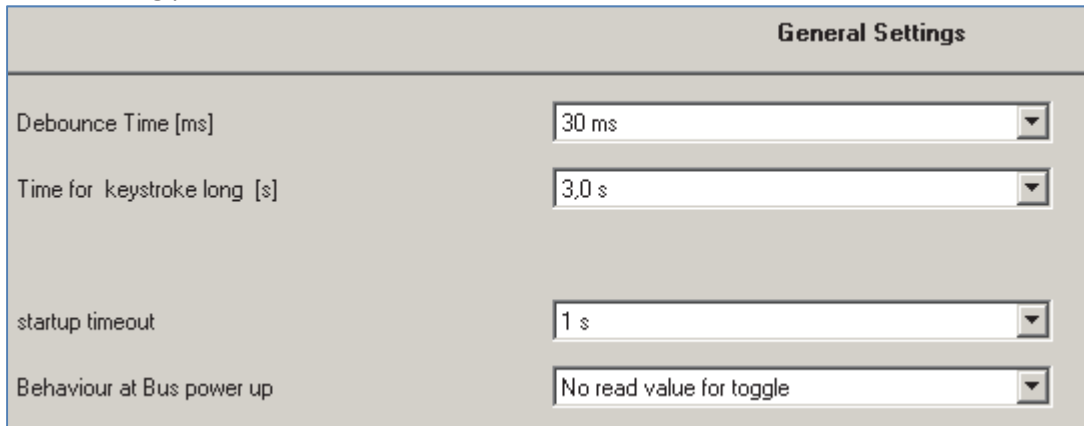
Table 4: Communication objects – default settings

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.

4 Reference ETS-Parameter

4.1 General Settings

The following parameters exist once and affect all channels:



General Settings	
Debounce Time [ms]	30 ms
Time for keystroke long [s]	3,0 s
startup timeout	1 s
Behaviour at Bus power up	No read value for toggle

Figure 10: General settings

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

ETS-text	Dynamic range [default value]	comment
Debounce time	10..120 ms [10]	The inputs are debounced with this time
Time for keystroke long	0,1-30 sec [0.8 sec]	releases the time when the ETS recognizes a long keystroke
Behavior at bus power up	<ul style="list-style-type: none"> ▪ No read value for toggle ▪ Read value for toggle 	activates the reading of the value for toggle at bus power up

Table 5: General settings

1. The setting option for the debounce time is for debouncing of the input signals from mechanical switches
2. The parameter “Time for keystroke long” allocates a static value to the binary input from when a long keystroke is recognized. This parameter is important for functions, which have different functions for a long and a short keystroke.
3. The parameter “Behavior at bus power up” defines the behavior of the binary input at a bus power return. The setting “Read value for toggle” effects that all communication objects “value for toggle” are read. So the Input knows the current status of the objects. If you choose the setting “no read value for toggle”, the input will not know the current status of the actor. So the input assumes an unconfirmed value for the objects “value for toggle” and sends always a “0”-signal at the next operation. Only now the input 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 input will send immediately the right value for toggling.

4.2 Configuration

Setting of the functionality of the channels:

Setting

Function Input A / B	Channels grouped ▼
Function Input C / D	Channels unique ▼
Function Input E / F	disabled ▼
Function Input G / H	<div style="border: 1px solid #ccc; padding: 2px;"> disabled Channels grouped Channels unique </div>
Function Input I / J	disabled ▼
Function Input K / L	disabled ▼
Function Input M / N	disabled ▼
Function Input O / P	disabled ▼

Figure 11: Usage of the channels

ETS-text	Dynamic range [default value]	comment
Function Input A/B –[O/P]	<ul style="list-style-type: none"> ▪ disabled ▪ Channels grouped ▪ Channels unique 	Operating mode of the channels

Table 6: Parameter channel-configuration

There are 3 different operating modes for every channel(have a look at chart 6). The following options to parameterize the channels are dependent to the chosen 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 16 blocking objects parameterize able at a 16-fold binary input. 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 corresponding channel by sending a logical 1

Table 7: 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
Input A/B	<ul style="list-style-type: none"> ▪ Dimming ▪ Shutter ▪ Switch 	Operating mode of the channel
Dimming function A/B	<ul style="list-style-type: none"> ▪ Brighter/Darker ▪ Darker/Brighter 	Defines which channel should dim up and which should dim down
Shutter function A/B	<ul style="list-style-type: none"> ▪ Up/Down ▪ Down/Up 	Defines which channel should drive the shutter a down and which up
Switch function A/B	<ul style="list-style-type: none"> ▪ On/Off ▪ Off/On 	Defines which channel should switch off and which on
Blocking Object	<ul style="list-style-type: none"> ▪ Inactive ▪ Active 	The blocking object can be displayed for every pair of channels

Table 8: 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 inputs can be made individually, so it is possible to configure which input should for example drive the shutters up and which down.

4.4.1 Dimming

The two-button 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:

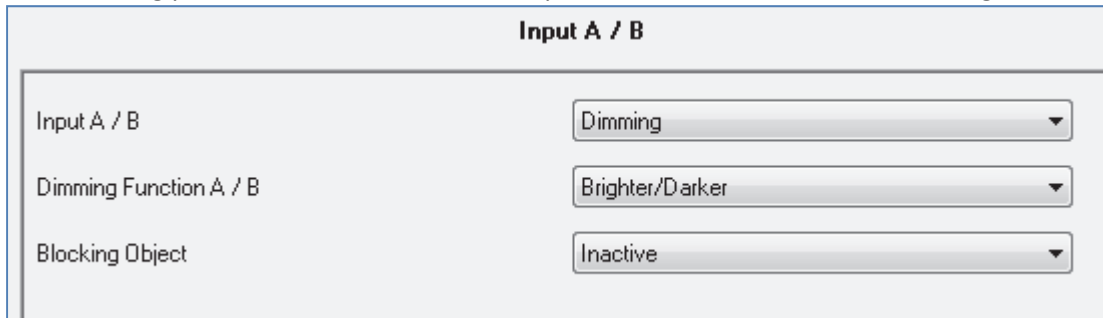


Figure 12: Parameter two-button dimming

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 9: Communication objects two-button 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 input. 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 input A switches the lights on. The lights are switched off by a short keystroke at input B. A long keystroke dims the lights step by step until releasing the long keystroke. The lights are dimmed brighter at input A and darker at input B. The binary input starts always with the last brightness level, before switching off.

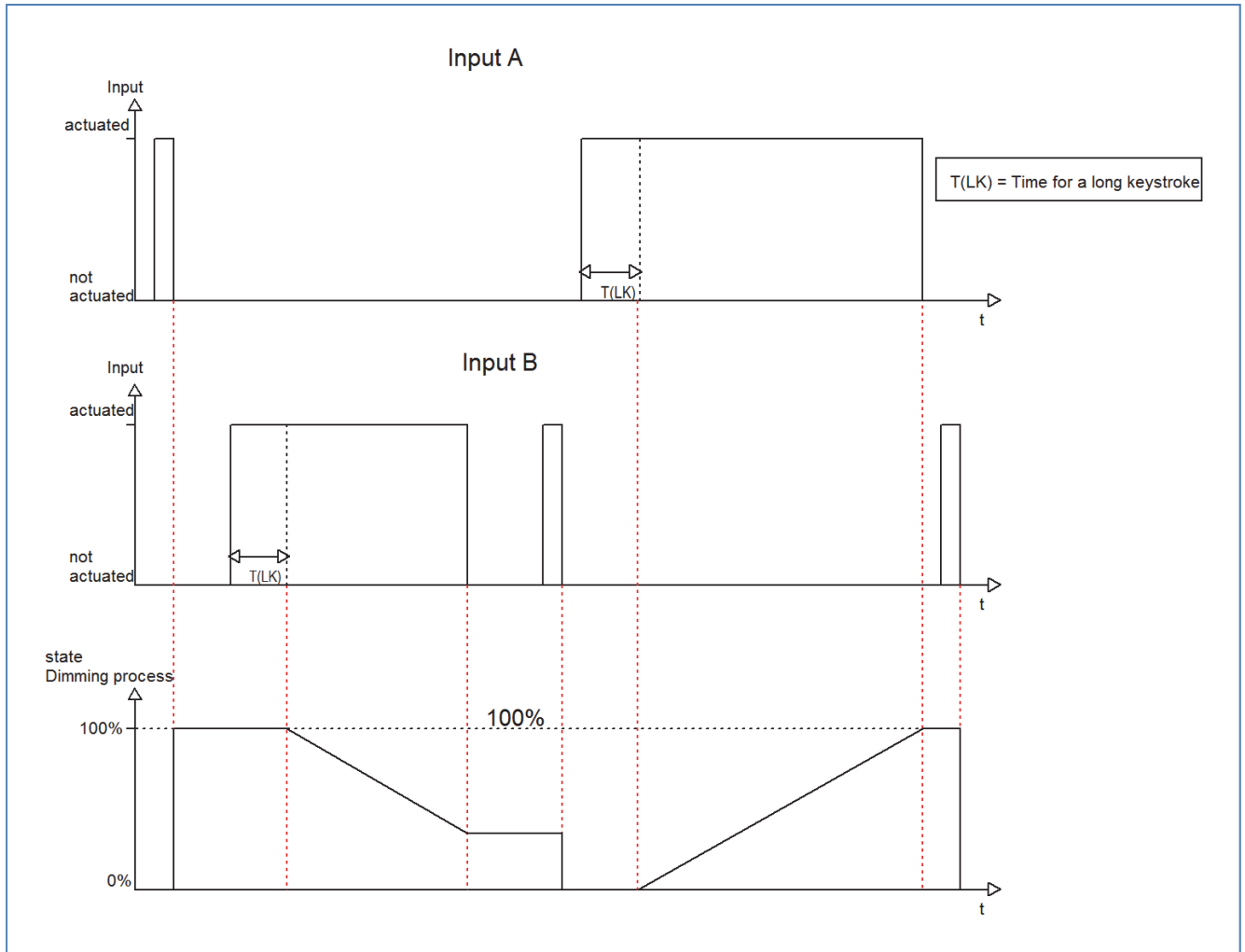
The step size is set fixed to 100% at the dual surface dimming. It is a start-stop dimming, that means the lights are dimmed as long as you hold the button. After releasing the button a stop value is sent, which stops the dimming process. So you can dim the lights with only one keystroke from 0% to 100% or from 100% to 0%, by pushing the button long enough.

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

Input	Function Brighter/Darker		Function Darker/Brighter	
	Input A	Input B	Input A	Input B
Dimming function	Brighter	Darker	Darker	Brighter
Switching function	On	Off	Off	On

Table 10: 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:

Input A / B

Input A / B Shutter ▼

Shutter Function A / B Up, Down ▼

Operating function Short=move / long=stop/blinds ▼

This setting can be apply only for hardware starting from version 2.6 <- TIP

Blocking Object Inactive ▼

Figure 13: 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 11: 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 binary input sends these signals to the corresponding driving commands.

From hardware version 2.6 (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:

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

Table 12: 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:

Input A / B

Input A / B	<input style="width: 95%;" type="text" value="Switch"/>
Switch function A / B	<input style="width: 95%;" type="text" value="on / off"/>
Blocking Object	<input style="width: 95%;" type="text" value="Inactive"/>

Figure 14: 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 input and the inverted signal for the second channel. So you can choose which channel is supposed to switch off and which to 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 13: Communication object two-button switching function

4.5 Parameters channels unique

There are 7 different operating modes for the unique channels, which can be adjusted for each channel:

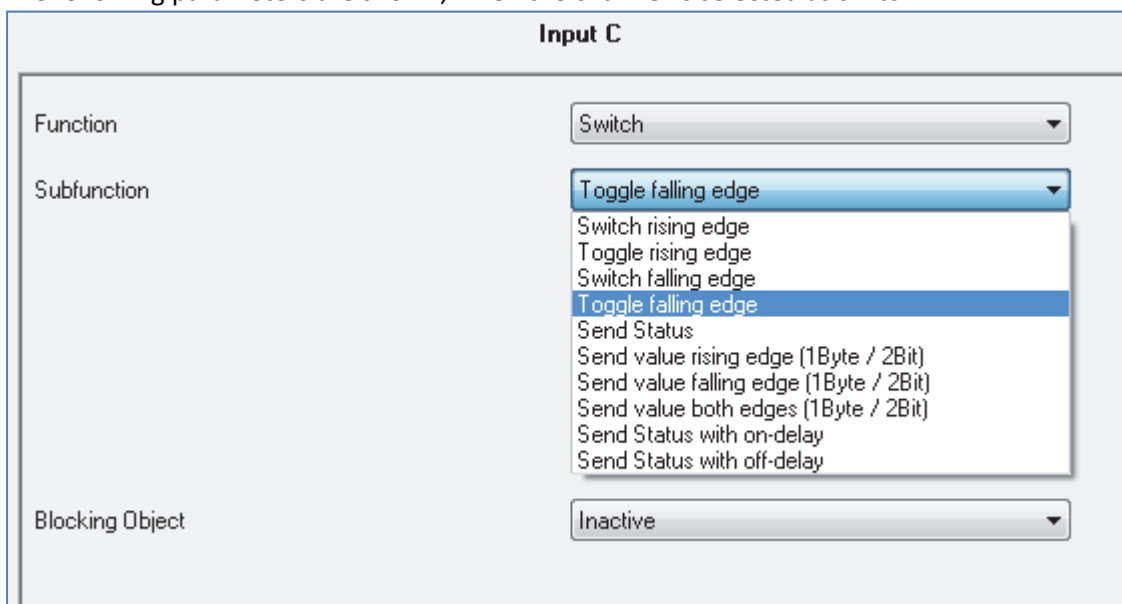
- Inactive
- Switch
- Scene
- Counter
- Switch short/long
- One button dimming
- One button shutter

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:



The screenshot shows a configuration window titled "Input C". It contains three main sections:

- Function:** A dropdown menu set to "Switch".
- Subfunction:** A dropdown menu with "Toggle falling edge" selected. The list of options includes:
 - Toggle falling edge (selected)
 - Switch rising edge
 - Toggle rising edge
 - Switch falling edge
 - Toggle falling edge
 - Send Status
 - Send value rising edge (1Byte / 2Bit)
 - Send value falling edge (1Byte / 2Bit)
 - Send value both edges (1Byte / 2Bit)
 - Send Status with on-delay
 - Send Status with off-delay
- Blocking Object:** A dropdown menu set to "Inactive".

Figure 15: 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

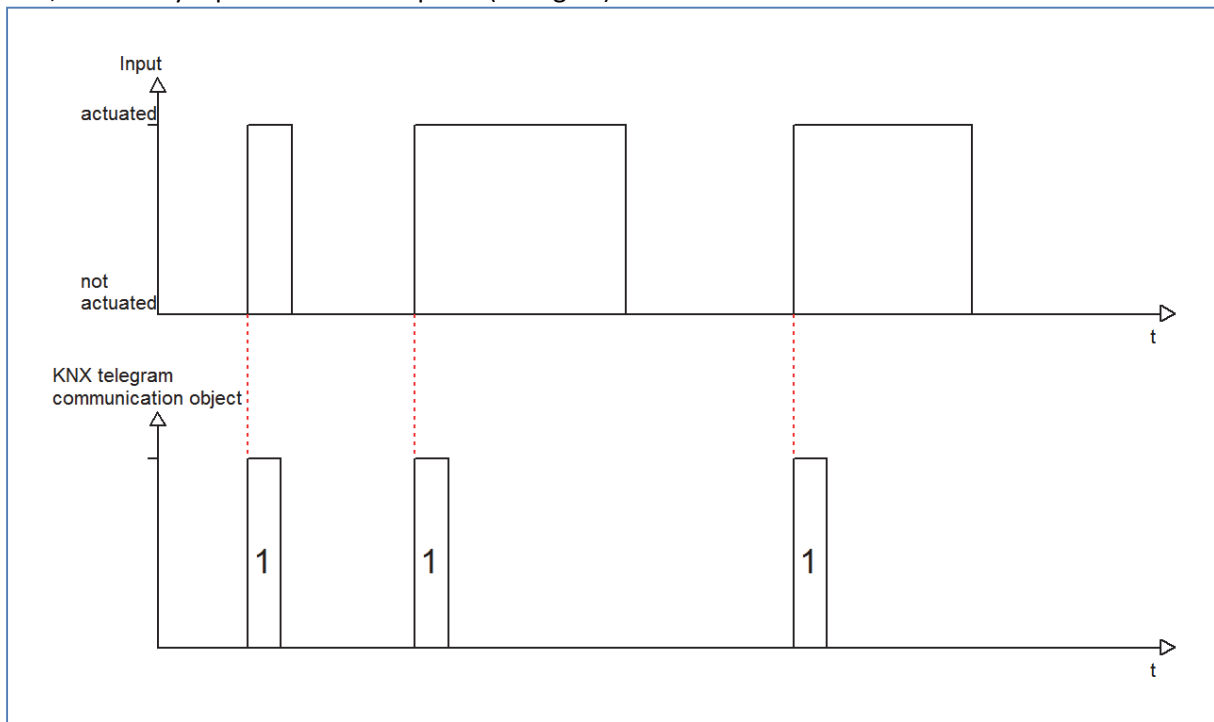
The following setting options are available, when the sub-function switch falling/rising edge was adjusted:

ETS-text	Dynamic range [default value]	comment
Value for rising/falling edge	<ul style="list-style-type: none"> • On • Off 	switches on/off at a falling/rising edge

Table 14: 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 binary input sends an On-pulse (=1-signal):



The following chart shows the corresponding communication object:

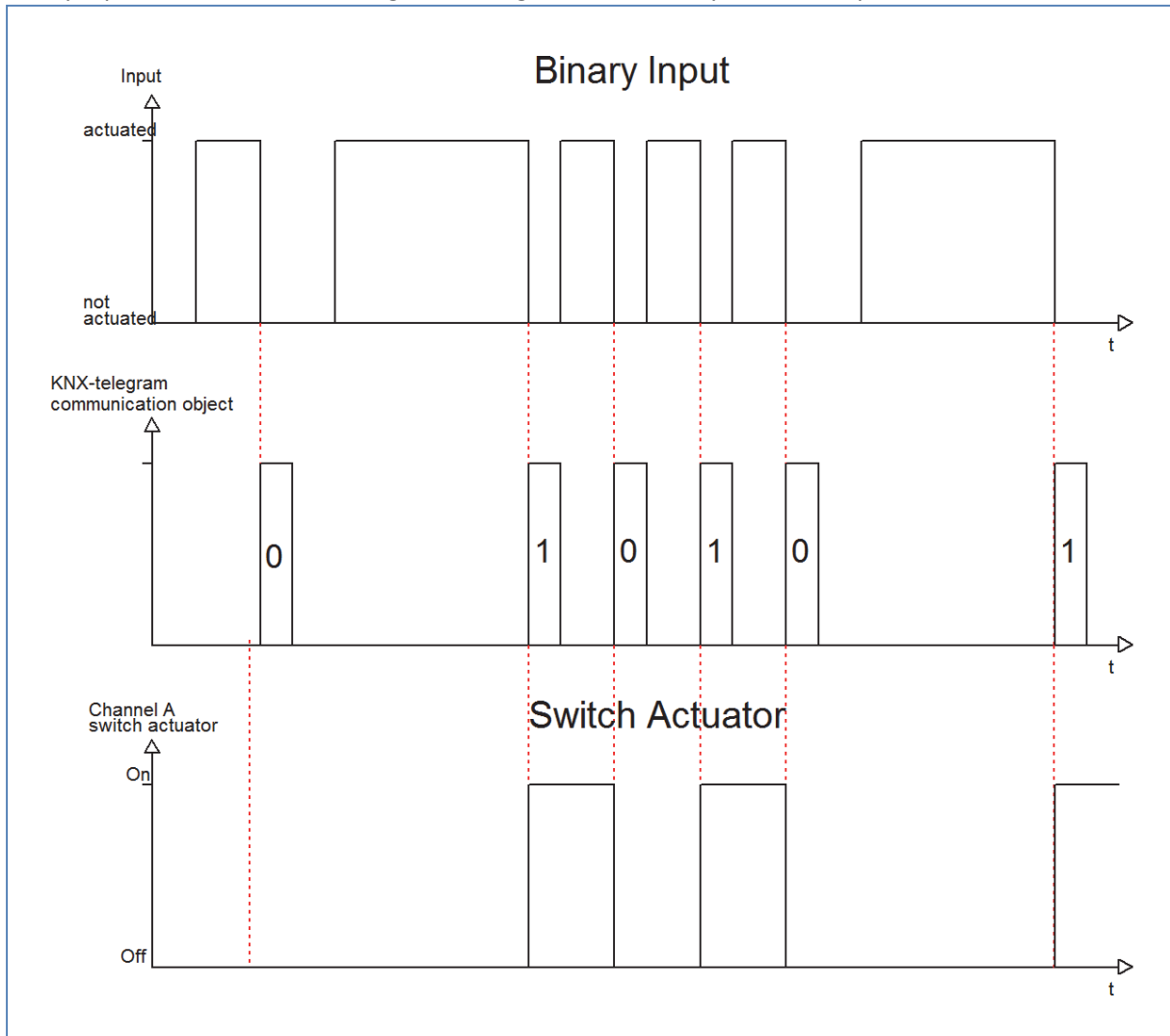
Number	Name	Length	Usage
0	Switch	1 Bit	Switching function, no differences between a long and a short keystroke

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

By undocking this communication object, you have more choices to program the input. 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 input sends always the parameterized signal for the corresponding edge. The following window is shown for the sub-function “Send status”:

Input C

Function	Switch
Subfunction	Send Status
Value for rising edge	On
Value for falling edge	Off
Blocking Object	Inactive
Behaviour at Bus power up	send nothing
Send cyclic activ	Off

Figure 16: Sub-function send status

These settings are available:

ETS-text	Dynamic range [default value]	comment
Value for rising edge	<ul style="list-style-type: none"> • On • Off 	switches on/off at a rising edge
Value for falling edge	<ul style="list-style-type: none"> • On • Off 	switches on/off at a falling edge
Send cyclic	<ul style="list-style-type: none"> • Off • On 	switches the cyclic sending on/off
Time interval for send cyclic	1-3000s [1]	Time between two sending processes
Behavior at bus power up	<ul style="list-style-type: none"> • send nothing • send status 	defines the behavior at a bus power up

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

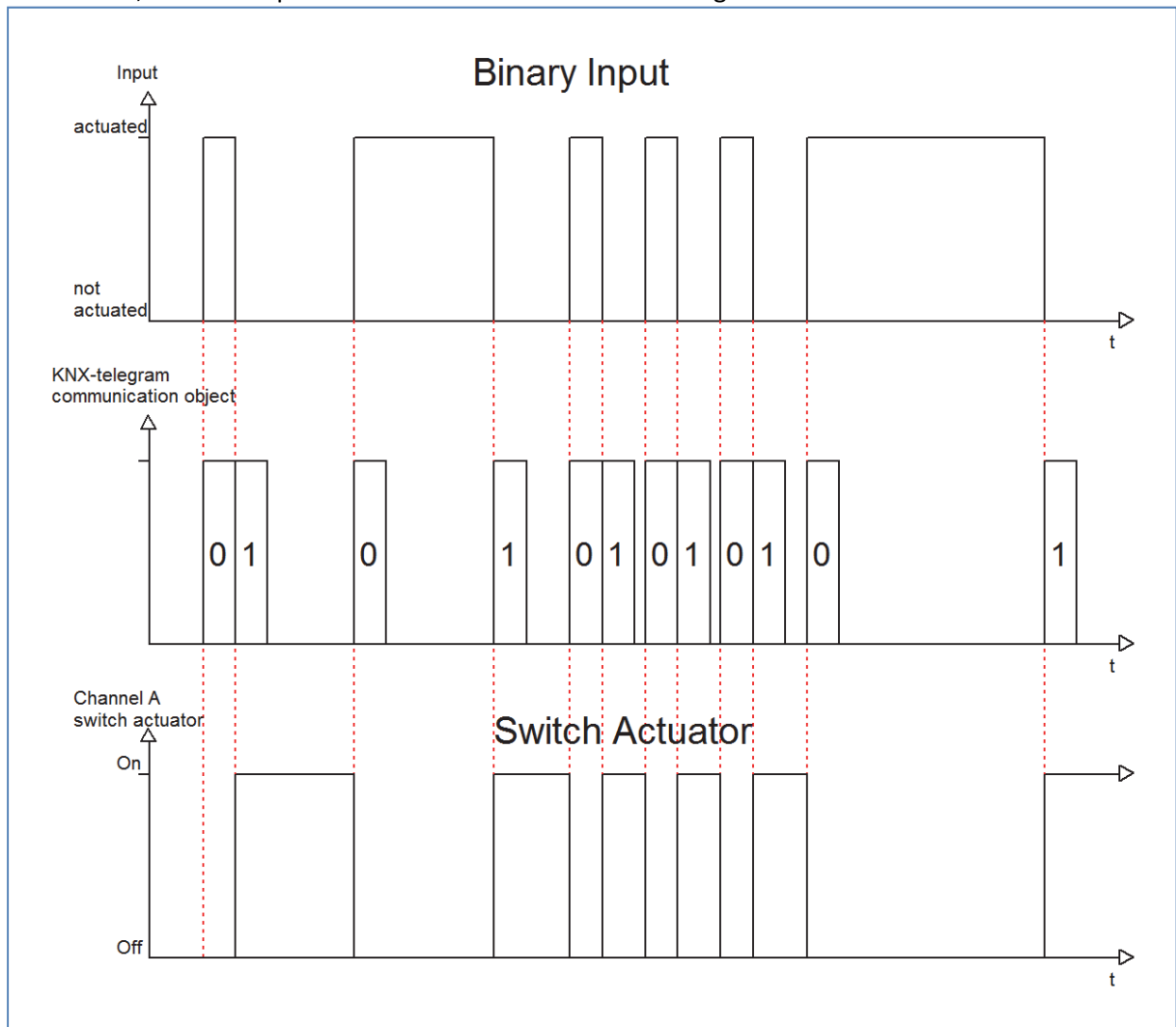
Table 18: Communication object send status

The parameter “Value for rising edge” defines whether the channel is supposed to 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 input sends the same signal twice, for example an On-signal.

The cyclic sending causes that the state of the input is sent periodically in certain parameterize able intervals. Then the input 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 input sends an 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:

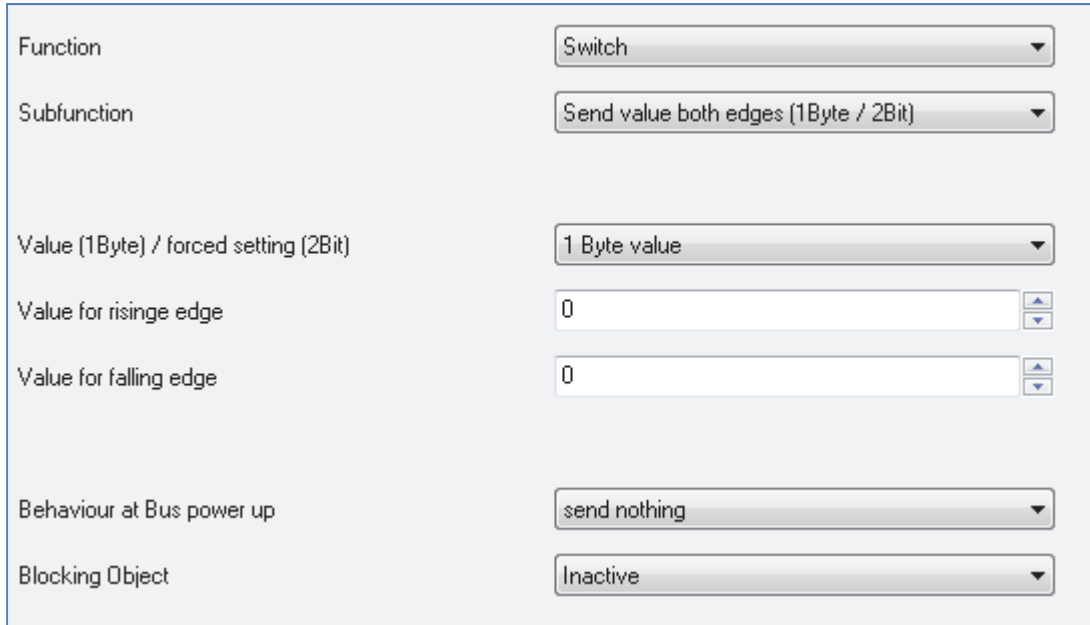


Figure 17: Sub-function send value

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)	<ul style="list-style-type: none"> • 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 input 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	<ul style="list-style-type: none"> ▪ 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

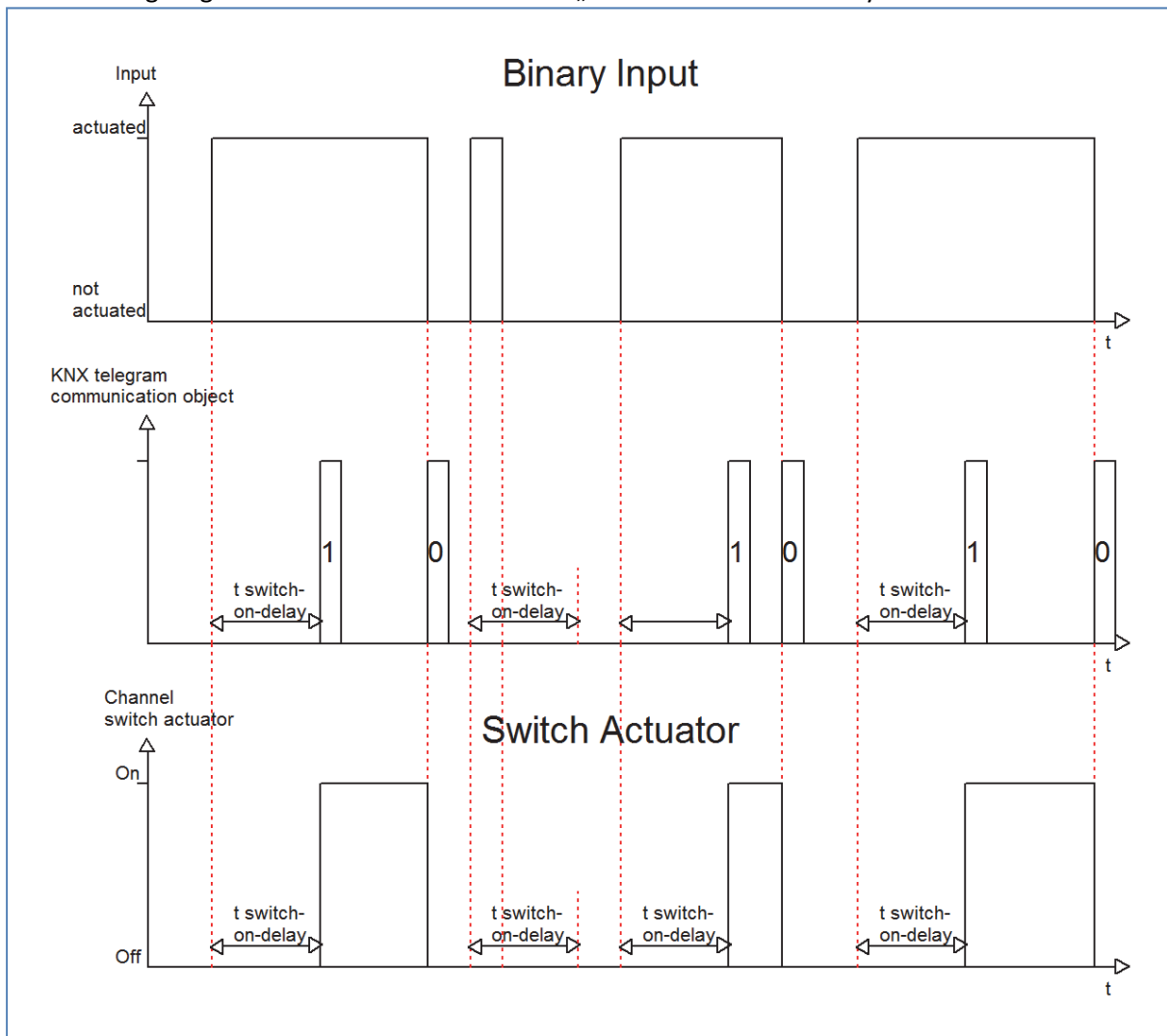
The following setting options are available at the function “Send value with on/off delay”:

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 binary input sends its value after a parameterized time. At the on-delay, the time starts when the associated input was switched on and at the off-delay, the time starts when the associated input was switched off. The binary input 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:

Input D

Function	<input type="text" value="Switch"/>
Subfunction	<input type="text" value="Send Status with on-delay"/>
Delay time	<input type="text" value="5 s"/>
Blocking Object	<input type="text" value="Inactive"/>

Figure 18: Send value with on-delay

The following chart shows the communication object:

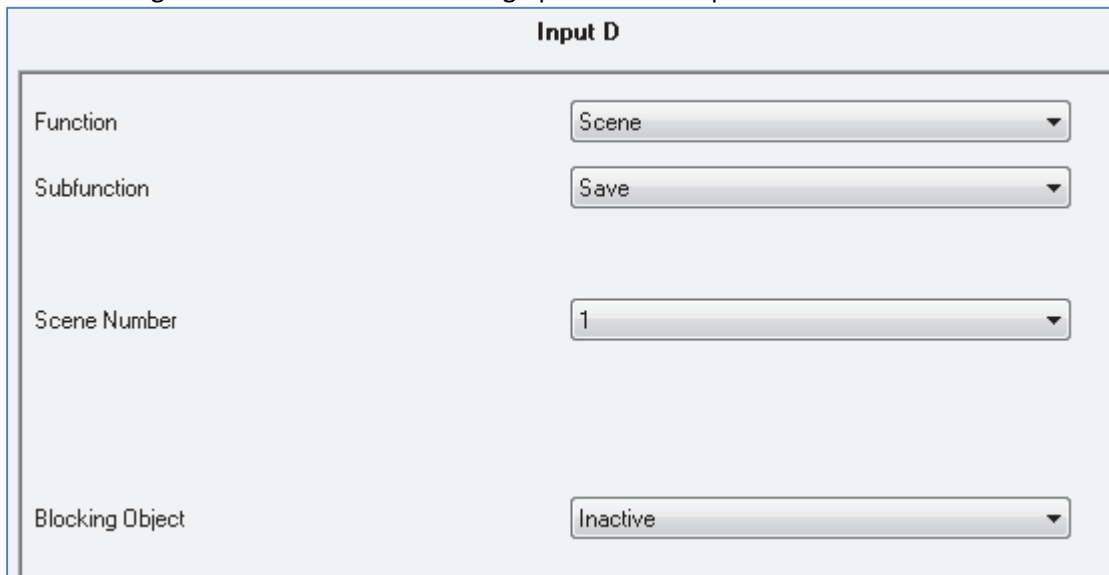
Number	Name	Length	Usage
0	Switch	1 Bit	Switching function; no differences between long and short keystroke

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 binary input 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:



The screenshot shows a configuration window titled "Input D". It contains four rows of settings, each with a label on the left and a dropdown menu on the right:

- Function: Scene
- Subfunction: Save
- Scene Number: 1
- Blocking Object: Inactive

Figure 19: Parameter Scene

The following chart shows the dynamic range of this parameter:

Sub-function	Dynamic range [default value]	comment
Saving function	<ul style="list-style-type: none"> ▪ 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	<ul style="list-style-type: none"> ▪ 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 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 binary input by a long keystroke. When the saving function was activated, a long keystroke at the binary input 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:

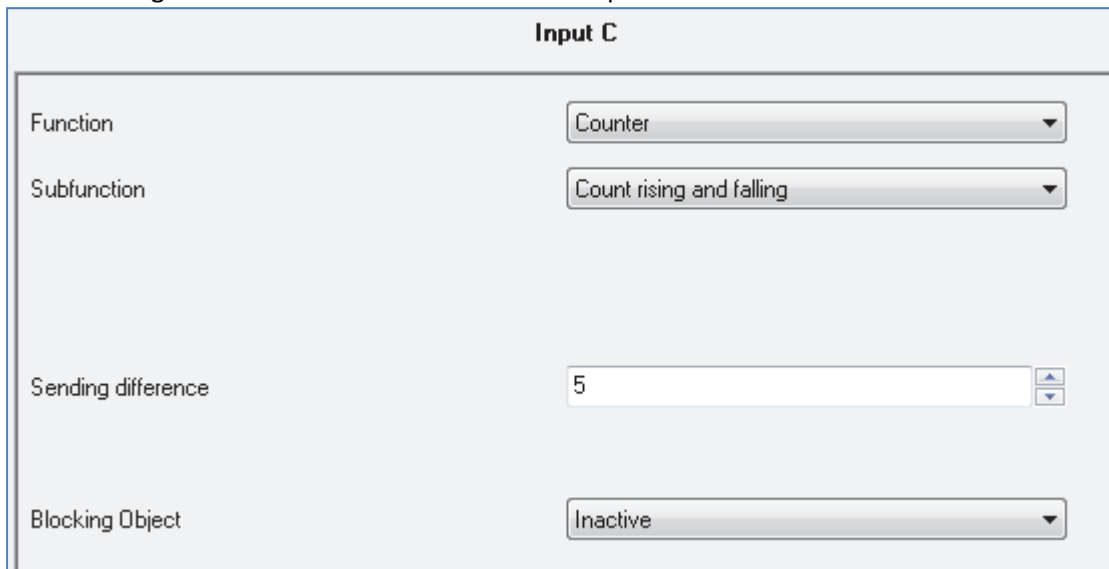
Scene	Retrieve		Save	
	Hex.	Dez.	Hex.	Dez.
1	0x00	0	0x80	128
2	0x01	1	0x81	129
3	0x02	2	0x82	130
4	0x03	3	0x83	131
5	0x04	4	0x84	132
6	0x05	5	0x85	133
7	0x06	6	0x86	134
8	0x07	7	0x87	135
9	0x08	8	0x88	136
10	0x09	9	0x89	137
11	0x0A	10	0x8A	138
12	0x0B	11	0x8B	139
13	0x0C	12	0x8C	140
14	0x0D	13	0x8D	141
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 Counter

By using the counter function the number of switching processes can be counted. On doing this the binary input can also make differences between the edges.

The following chart shows the sub-function of this parameter:



The screenshot shows a configuration window titled "Input C". It contains four rows of settings:

- Function:** A dropdown menu set to "Counter".
- Subfunction:** A dropdown menu set to "Count rising and falling".
- Sending difference:** A text input field containing the number "5", with up and down arrow buttons to its right.
- Blocking Object:** A dropdown menu set to "Inactive".

Figure 20: Sub-function counter

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

Sub-function	Dynamic range [default value]	comment
Sub-function	<ul style="list-style-type: none"> ▪ Count rising ▪ Count falling ▪ Count rising and falling 	at "count rising" only the rising edges will be counted at "count falling" only the falling edges will be counted
Sending difference	0-65535 [5]	the binary input sends the current value of the counter, when the sending difference is reached
Blocking object	<ul style="list-style-type: none"> ▪ Inactive ▪ Active 	have a look at 4.3.1 blocking object

Table 29: Sub-function counter

The chart shows the depending communication objects:

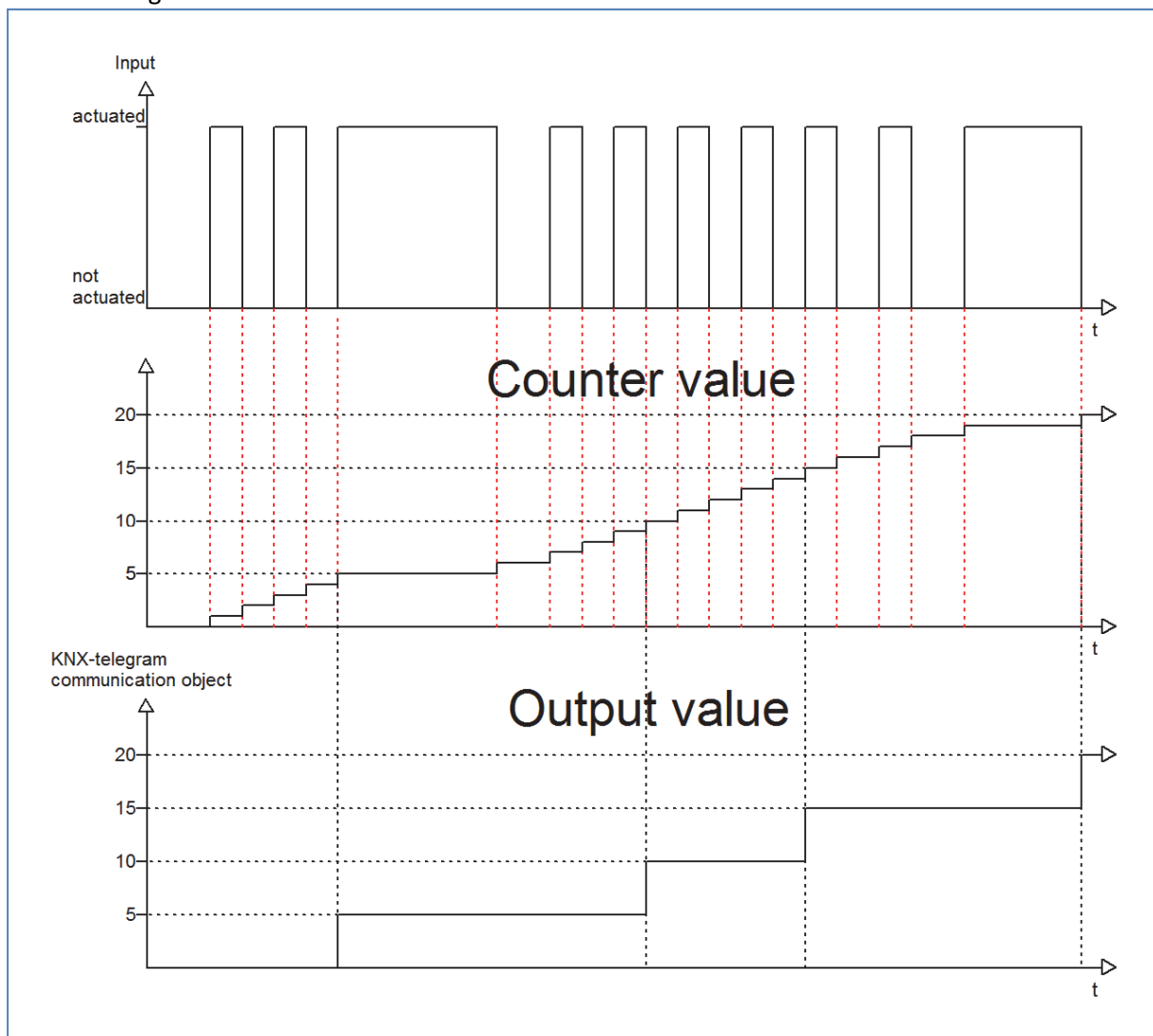
Number	Name	Length	Usage
0	Reset Counter	1 Bit	resets the counter
3	Counter	4 byte	outputs the current counter value

Table 30: Communication object Parameter Counter

There is a setting which edge should be counted at the counter-function. If you select “Count rising and falling”, the counter will count for toggling two switching processes. If only the rising or the falling edge should be counted, the counter will count only one switching process for toggling. The sending difference adjusts when the counter should send a value. When you have for example adjusted 10 as sending difference, the counter will count every switching process but send its value only at reaching 10.

The communication object “reset counter” sets the counter back to 0. This communication object reacts as well to a “0” as to a “1”.

The following diagram shows the function of the counter with the setting “count rising and falling” and a sending difference of 5.



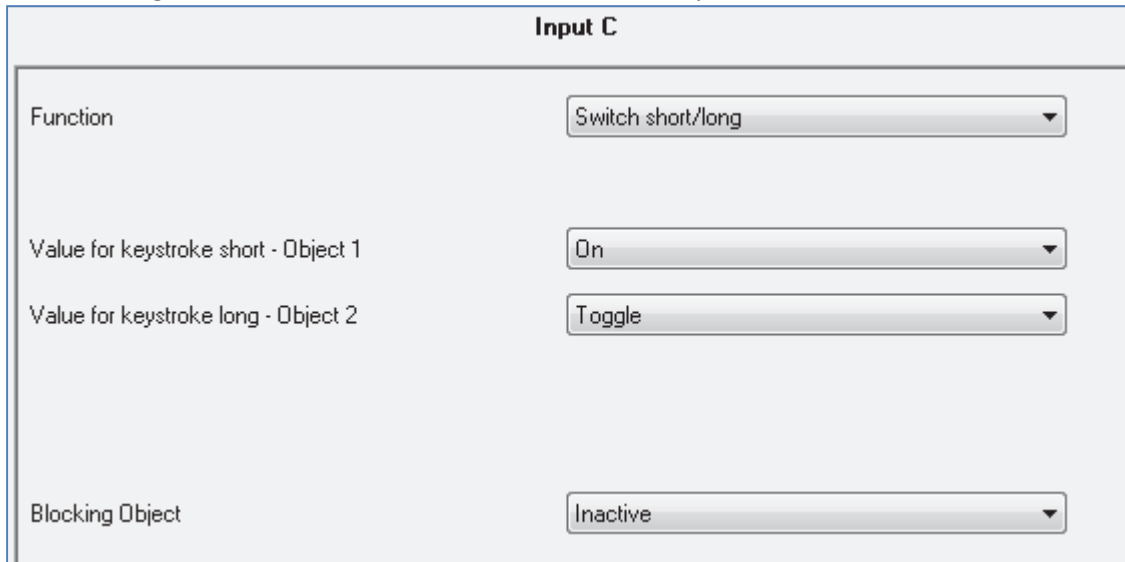
At this example, the counter sends only a value, when the counter value is a multiple of the sending difference. So the counter value is a continual rising value and the output value a periodic rising value.

As soon as the group address, in which the communication object “reset counter” is connected to, is addressed with a logical “0” or a logical “1”, the counter value as well as the output value will be set back to 0.

4.5.4 Switch short/long

The parameter switch short/long can assign the input different switching processes for a long and a short keystroke.

The following illustration shows the sub-functions for this parameter:



The screenshot shows a configuration window titled "Input C". It contains four dropdown menus:

- Function:** Set to "Switch short/long"
- Value for keystroke short - Object 1:** Set to "On"
- Value for keystroke long - Object 2:** Set to "Toggle"
- Blocking Object:** Set to "Inactive"

Figure 21: 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	<ul style="list-style-type: none"> ▪ On ▪ Off ▪ Toggle ▪ Send value ▪ Nothing 	Action for a short keystroke
Value for keystroke long - Object 2	<ul style="list-style-type: none"> ▪ On ▪ Off ▪ Toggle ▪ Send value ▪ Nothing 	Action for a long keystroke
Blocking object	<ul style="list-style-type: none"> ▪ Inactive ▪ Active 	have a look at 4.3.1 blocking object

Table 31: Sub-functions parameter switch short/long

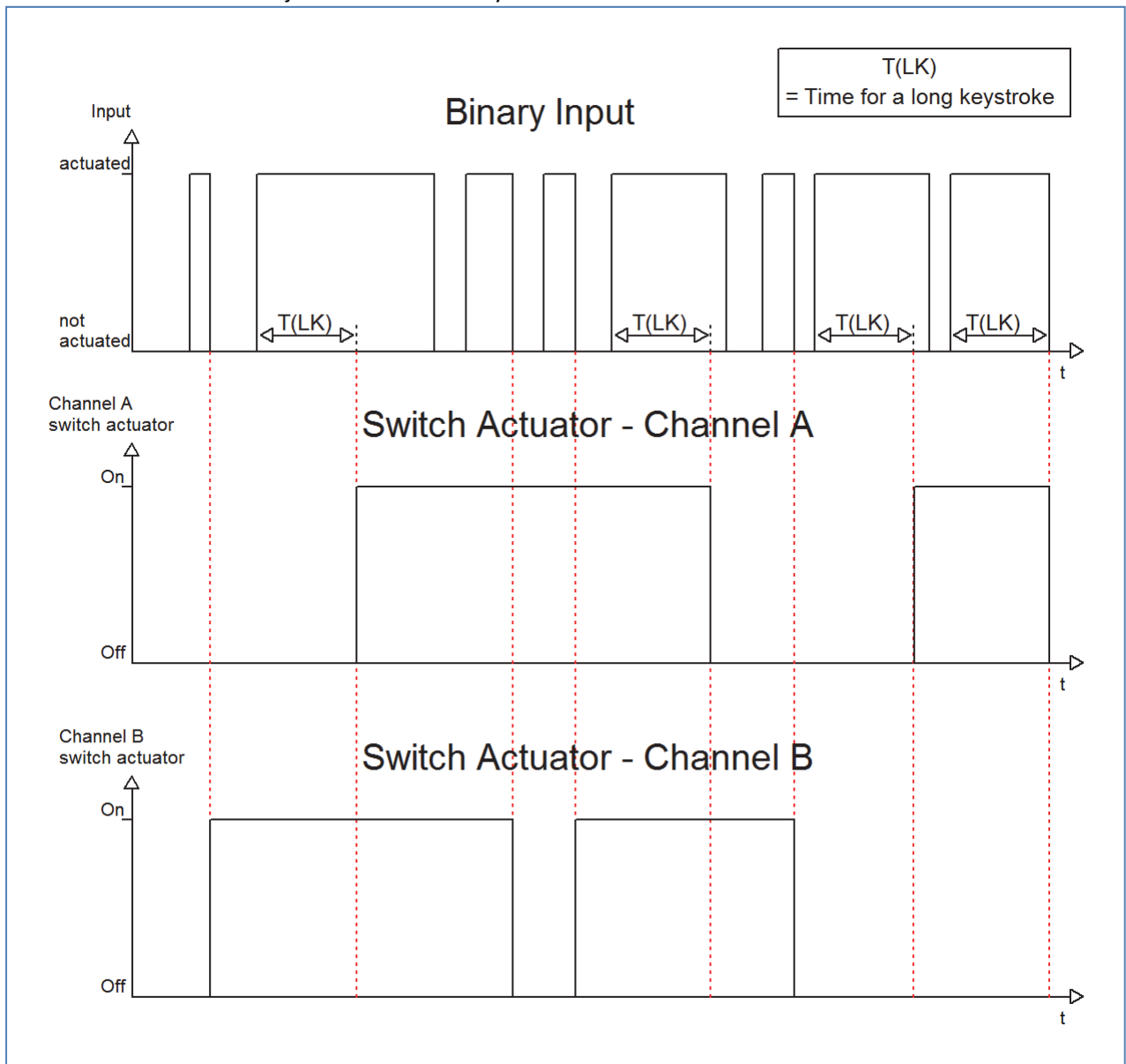
The chart shows the associated communication objects:

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

Table 32: Communication object parameter switch short/long

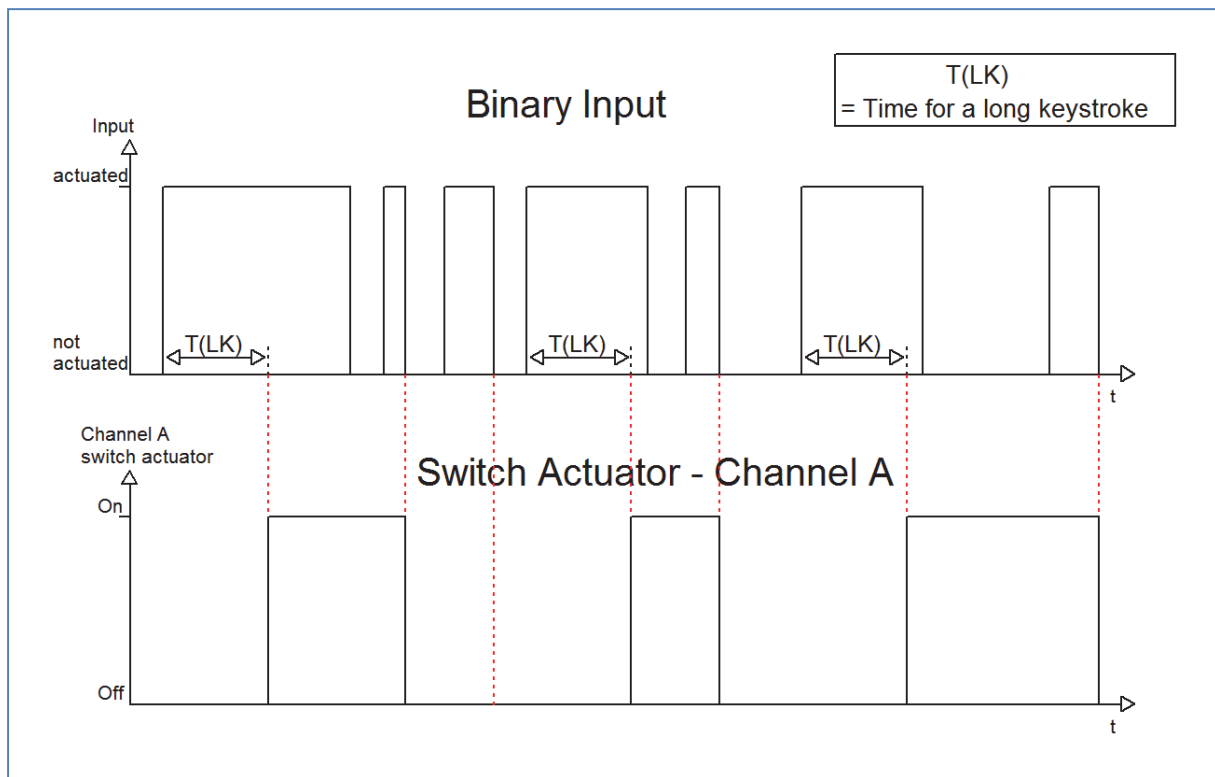
The parameter “switch short/long” can control for example two channels of an actuator by using only one input. Furthermore you can switch a channel with a long keystroke on and with a short keystroke off. For both objects, a function can be set individually. Therefore the sub-functions on, off, toggle 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 input 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:



In this example the binary input 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	<ul style="list-style-type: none"> ▪ 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 33: 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.

4.5.5 One button Dimming

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

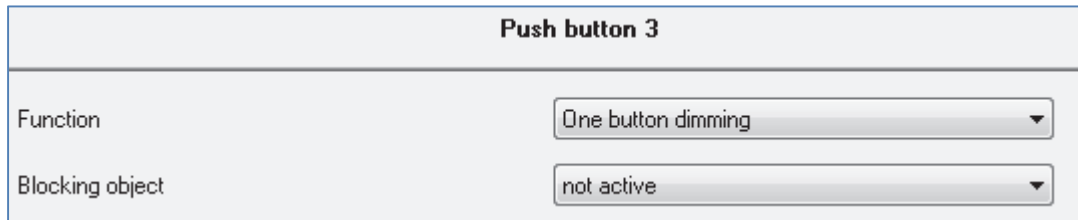


Figure 22: Parameter one-button dimming

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

Sub-function	Dynamic range [default value]	comment
Blocking object	<ul style="list-style-type: none"> ▪ Inactive ▪ Active 	have a look at 4.3.1 blocking object

Table 34: Sub function one-button dimming

The chart shows the available communication 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 35: 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 dimming 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%.

4.5.6 One-button Shutter

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

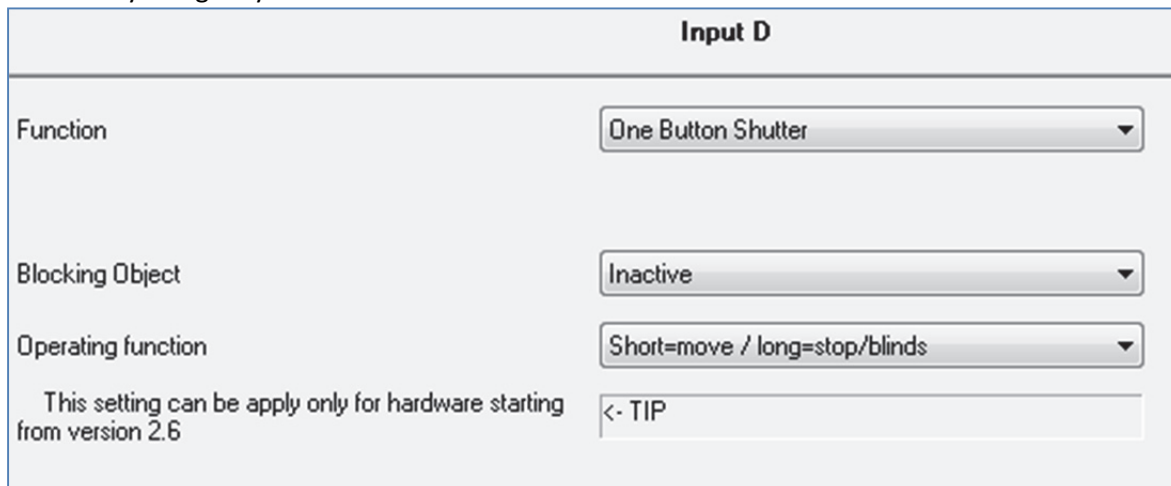


Figure 23: Parameter one-surface shutter

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

Sub-function	Dynamic range [default value]	comment
Blocking object	<ul style="list-style-type: none"> ▪ Inactive ▪ Active 	have a look at 4.3.1 blocking object

Table 36: Sub-functions one-surface 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

Table 37: Communication objects one-surface 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 2.6 (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 binary input sends always the complementary value as before.

4.6. Logic

The binary input contains 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	not automatic ▼
Output inverted	No ▼
Settings for logic 2	Or ▼
objecttype 2	Scene ▼
Scene Number	2 ▼

Figure 24: 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	<ul style="list-style-type: none"> ▪ 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 38: 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 binary input 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
<ul style="list-style-type: none"> ▪ disabled ▪ And ▪ Or 	<ul style="list-style-type: none"> ▪ 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 39: 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 40: 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 inputs 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".

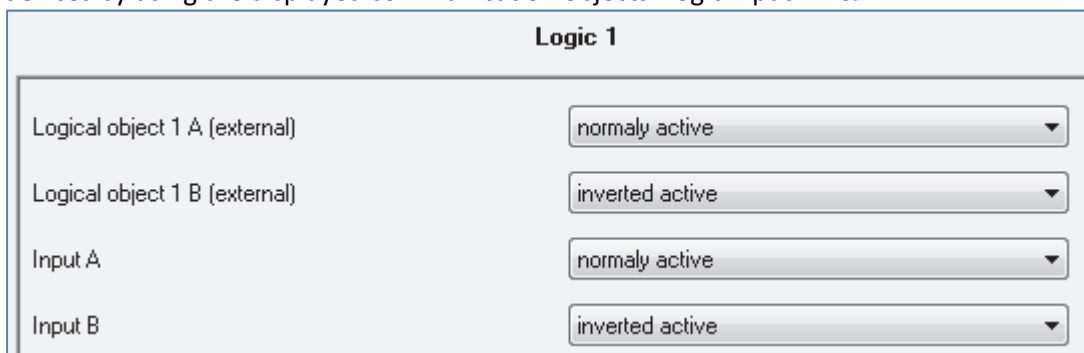


Figure 25: Setting logic

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:

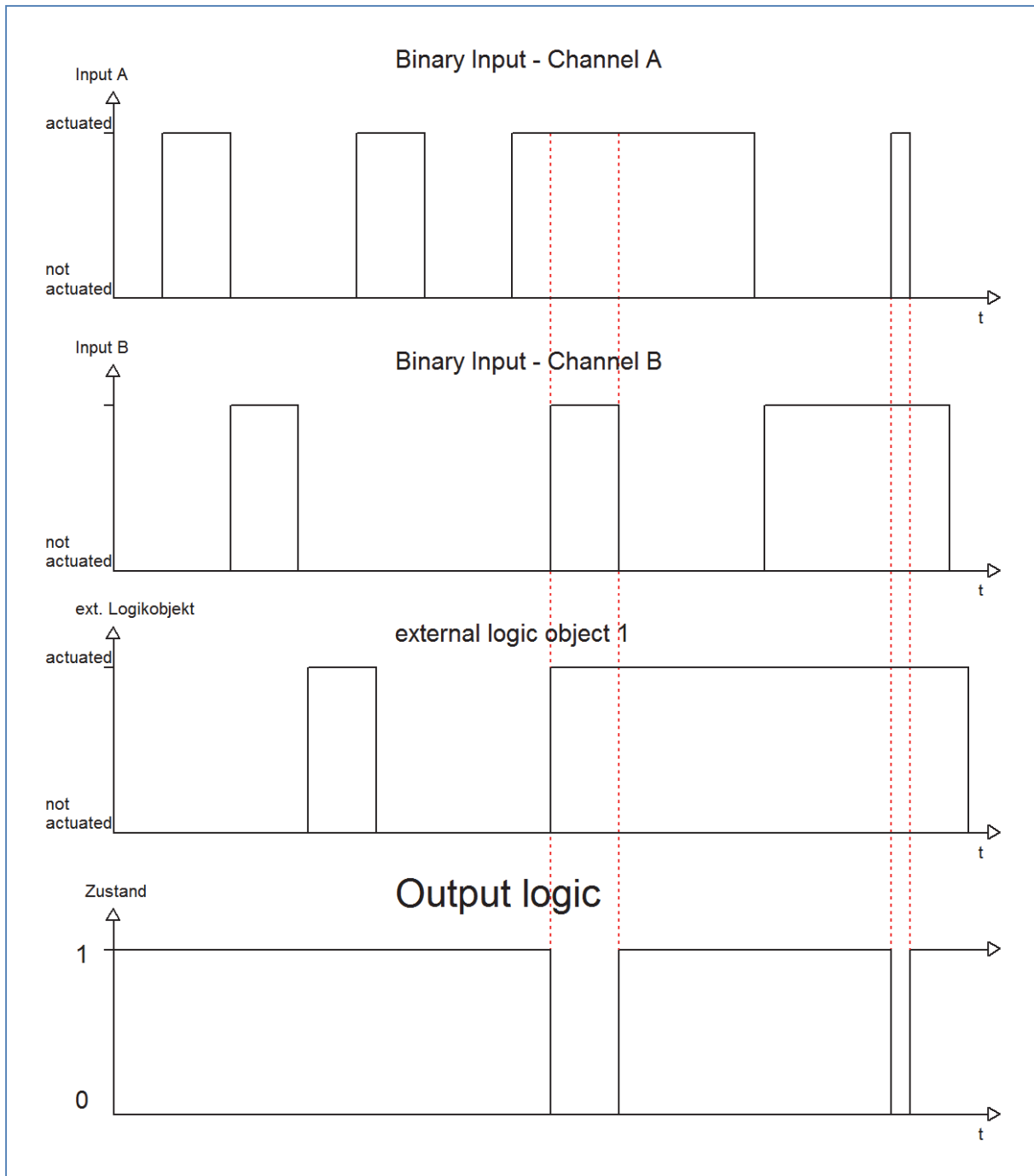
Sub-function	Dynamic range [default value]	comment
Sending condition	<ul style="list-style-type: none"> ▪ not automatic ▪ change of input ▪ change of output 	Adjustment indicates, when the state of the logic block should be sent
Output inverted	<ul style="list-style-type: none"> ▪ No ▪ Yes 	Adjustment indicates, whether the output should be inverted or not

Table 41: Logic sub-function switch

The sending condition adjusts, when the binary input should send a signal on the bus. By adjusting the sending condition “change of input”, the binary input 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 binary input 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 channel 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 42: 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 binary input and additional 2 external logic objects to the logic function.

5 Index

5.1 Register of illustrations

Figure 1: Exemplary circuit diagram BE16000.01.....	4
Figure 2: Exemplary circuit diagram BE08000.01.....	5
Figure 3: Exemplary circuit diagram BE16024.01.....	5
Figure 4: Exemplary circuit diagram BE08024.01.....	6
Figure 5: Exemplary circuit diagram BE16230.01.....	6
Figure 6: Exemplary circuit diagram BE08230.01.....	7
Figure 7: Overview hardware module binary input	8
Figure 8: Communication objects per channel.....	11
Figure 9: communication objects logic	12
Figure 10: General settings.....	14
Figure 11: Usage of the channels	16
Figure 12: Parameter two-button dimming	18
Figure 13: Two-button shutter function	20
Figure 14: Two-button switching function.....	21
Figure 15: Parameter switch.....	22
Figure 16: Sub-function send status	26
Figure 17: Sub-function send value.....	28
Figure 18: Send value with on-delay.....	31
Figure 19: Parameter Scene.....	32
Figure 20: Sub-function counter.....	34
Figure 21: Parameter switch short/long.....	36
Figure 22: Parameter one-button dimming.....	39
Figure 23: Parameter one-surface shutter.....	40
Figure 24: Activation logic blocks	41
Figure 25: Setting logic	42

5.2 List of tables

Table 1: Functional overview binary input	9
Table 2: communication objects per channel	12
Table 3: communication objects logic	12
Table 4: Communication objects – default settings	13
Table 5: General settings	14
Table 6: Parameter channel-configuration	16
Table 7: Communication object blocking object	17
Table 8: Parameter Channels grouped	17
Table 9: Communication objects two-button dimming	18
Table 10: Dimming function	19
Table 11: Communication objects two-button shutter function	20
Table 12: shutter function	20
Table 13: Communication object two-button switching function	21
Table 14: Parameter switch rising/falling edge	23
Table 15: Communication object switch rising/falling edge	23
Table 16: Communication objects toggle rising/falling edge	24
Table 17: Parameter Send status	26
Table 18: Communication object send status	26
Table 19: Parameter send value	28
Table 20: Parameter send value, 1 Byte object	28
Table 21: Communication object Parameter Send value-1 Byte object	28
Table 22: Dynamic range send value-forced setting	29
Table 23: Communication object Send value-forced setting	29
Table 24: Parameter Send value with delay	30
Table 25: Communication object send value with delay	31
Table 26: Sub-function scene	32
Table 27: Communication object Parameter scene	32
Table 28: Calling and saving scenes	33
Table 29: Sub-function counter	34
Table 30: Communication object Parameter Counter	34
Table 31: Sub-functions parameter switch short/long	36
Table 32: Communication object parameter switch short/long	36
Table 33: Sub function Send value at switch short/long	38
Table 34: Sub function one-button dimming	39
Table 35: Communication objects one-button dimming	39
Table 36: Sub-functions one-surface shutter	40
Table 37: Communication objects one-surface dimming	40
Table 38: Common Parameter logic blocks	41
Table 39: Dynamic range logic	42
Table 40: Communication objects logic	42
Table 41: Logic sub-function switch	43
Table 42: Logic sub-function scene	45

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

MDT Binary Inputs 4/8/16-fold, MDRC

Version		
BE-04000.01	Binary Input 4-fold	2SU MDRC, Inputs for floating contacts
BE-04024.01	Binary Input 4-fold	2SU MDRC, Inputs 24VAC/DC
BE-04230.01	Binary Input 4-fold	2SU MDRC, Inputs 230VAC
BE-08000.01	Binary Input 8-fold	4SU MDRC, Inputs for floating contac
BE-08024.01	Binary Input 8-fold	4SU MDRC, Inputs 24VAC/DC
BE-08230.01	Binary Input 8-fold	4SU MDRC, Inputs 230VAC
BE-16000.01	Binary Input 16-fold	8SU MDRC, Inputs for floating contacts
BE-16024.01	Binary Input 16-fold	8SU MDRC, Inputs 24VAC/DC
BE-16230.01	Binary Input 16-fold	8SU MDRC, Inputs 230VAC

The MDT Binary Input is available in 3 versions, input detection of 24VAC/DC signals, input detection of 230VAC signals and reading out of floating contacts. The Binary Input reacts depending on its programmed parameters and sends a telegram on the KNX bus. You can connect conventional push-buttons or auxiliary contacts (e.g. door and window contacts) to the device.

The MDT Binary Inputs include four integrated logical modules to implement logical operations and logical control. These logical modules interpret all the inputs plus two external objects. So you can easily create KNX/EIB telegrams which are required in daily practice (e.g. fault telegrams or „all windows closed“).

Each Input is parameterized individually via ETS3f/4. The device provides extensive functions like switching of lighting, operation of blinds and shutters, counting of pulses, debounce time, contact type and telegram rate limitation. The command for rising and falling edge can be defined independently and with the block communication object each channel can be blocked or released. The sending of an second object is possible by the logical modules.

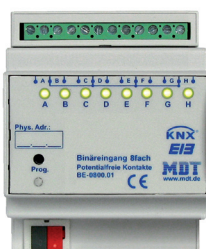
The MDT Binary Input is a modular installation device for fixed installations in dry rooms. It fits on DIN 35mm rails in power distribution boards or closed compact boxes.

For project design and commissioning of the MDT Binary Input it is recommended to use the ETS3f/ETS4 or later. Please download the application software at www.mdt.de/Downloads.html

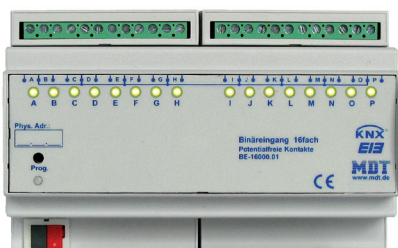
BE-04000.01



BE-08000.01



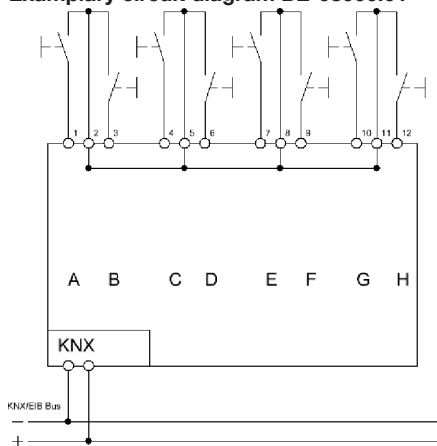
BE-16000.01



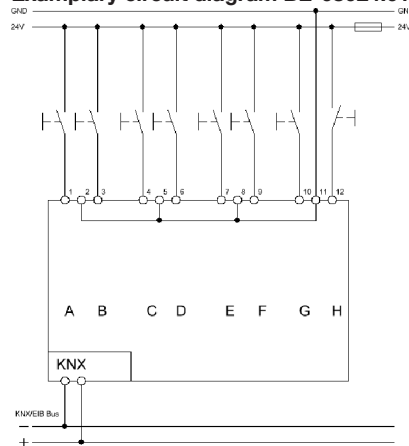
- Production in Germany, certified according to ISO 9001
- LED indicator for each channel
- To connect conventional push buttons or window/auxiliary contacts
- **4 integrated logical modules**, e.g. Interpretation of the inputs (e.g. window closed or sending of an second object)
- Telegram rate limitation
- NO or NC contact operation, sending of contact state
- Operation of blinds and shutters, 1 and 2 button operation
- Forced setting function for each output
- Scenes
- Block communication object for each channel
- Common object via logical modules
- Operation with short/long button push and 2 objects
- Counting of pulses, suitable for 30ms S0 pulses
- Cyclical sending, sending after reset
- Internal contact voltage 12V
- Modular installation device for DIN 35mm rails
- Integrated bus coupling unit
- 3 years warranty

Technical Data	BE-04000.01 BE-08000.01 BE-16000.01	BE-04230.01 BE-08230.01 BE-16230.01	BE-04230.01 BE-08024.01 BE-16024.01
Number of inputs	4/8/16	4/8/16	4/8/16
Signal voltage	For floating contacts	230VAC	12-24VDC / 24VAC
Internal contact voltage	12V	--	--
Permitted input cable length	100m	100m	100m
Permitted wire gauge			
Screw terminal	0,5 - 4,0mm ² solid core 0,5 - 2,5mm ² finely stranded	0,5 - 4,0mm ² solid core 0,5 - 2,5mm ² finely stranded	0,5 - 4,0mm ² solid core 0,5 - 2,5mm ² finely stranded
KNX busconnection terminal	0,8mm Ø, solid core	0,8mm Ø, solid core	0,8mm Ø, solid core
Power Supply	KNX bus	KNX bus	KNX bus
Power consumption KNX bus typ.	<0,3W	<0,3W	<0,3W
Operation temperature range	0 bis + 45°C	0 bis + 45°C	0 bis + 45°C
Enclosure	IP 20	IP 20	IP 20
Dimensions MDRC (Space Units)	2/4/8SU	2/4/8SU	2/4/8SU

Exemplary circuit diagram BE-08000.01



Exemplary circuit diagram BE-08024.01



Exemplary circuit diagram BE-08230.01

