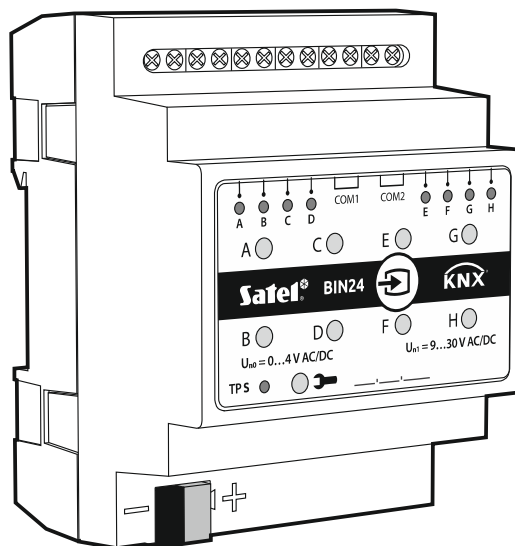


# Satel®

## KNX-BIN24

Universal binary input module



Firmware version 1.01

knx-bin24\_en 10/19

SATEL sp. z o.o. • ul. Budowlanych 66 • 80-298 Gdańsk • POLAND  
tel. +48 58 320 94 00  
[www.satel.eu](http://www.satel.eu)

## IMPORTANT

The device should be installed by qualified personnel.

Prior to installation, please read carefully this manual in order to avoid mistakes that can lead to malfunction or even damage to the equipment.

Changes, modifications or repairs not authorized by the manufacturer shall void your rights under the warranty.

SATEL aims to continually improve the quality of its products, which may result in changes in their technical specifications and software. Current information about the changes being introduced is available on our website.

Please visit us at:  
<http://www.satel.eu>

**The declaration of conformity may be consulted at [www.satel.eu/ce](http://www.satel.eu/ce)**

The following symbols may be used in this manual:



- note,



- caution.

## CONTENTS

1. Features .....	2
2. Description.....	3
2.1 Enclosure .....	4
3. Installation .....	4
3.1 Wiring diagram .....	5
4. Configuring the module .....	5
4.1 Schematic diagram of the module.....	7
4.2 Starting configuration .....	7
4.3 Configuring global parameters .....	10
4.4 Configuring channels.....	10
4.4.1 Physical channels.....	10
4.4.2 Virtual channels .....	12
4.4.3 Logic channels.....	13
4.4.4 Timer channels.....	15
4.5 Configuring function blocks .....	17
4.5.1 Switch .....	19
4.5.2 Edge (monostable switch) .....	21
4.5.3 Dimmer .....	23
4.5.4 Shutter .....	26
4.5.5 Sequence.....	29
4.5.6 Counter .....	34
4.5.7 Scene.....	37
4.6 Communication objects .....	37
4.6.1 Virtual inputs.....	37
4.6.2 Communication objects of function blocks .....	38
4.7 Configuration examples.....	45
4.7.1 One-channel control of several function blocks.....	45
4.7.2 Using timer channel.....	46
4.7.3 Using virtual and logic channels.....	47
5. Specifications .....	49
6. Appendix.....	50
6.1 4-bit telegrams for dimming DOWN/UP .....	50
6.2 4-bit telegrams for shutter control.....	50

KNX-BIN24 is a universal module of KNX binary inputs that enables signals from outside KNX installation to be converted into control telegrams for other devices on the bus. These signals may be generated by conventional ON/OFF buttons (to operate lighting, etc.), reed switches or sensors for registering various types of physical quantities (e.g. temperature).



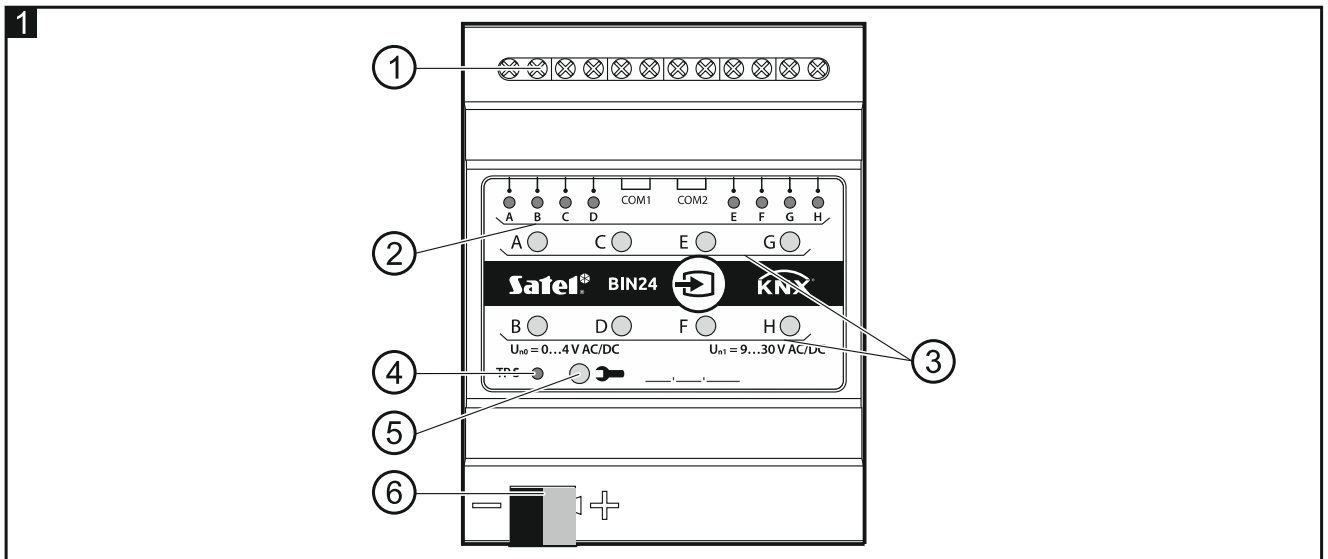
*The module has eight physical inputs which allow it to handle 8 independent signals ranging from 0 to 30 VDC/AC. Each physical input in the module corresponds to one physical channel (A – H).*


## 1. Features

---

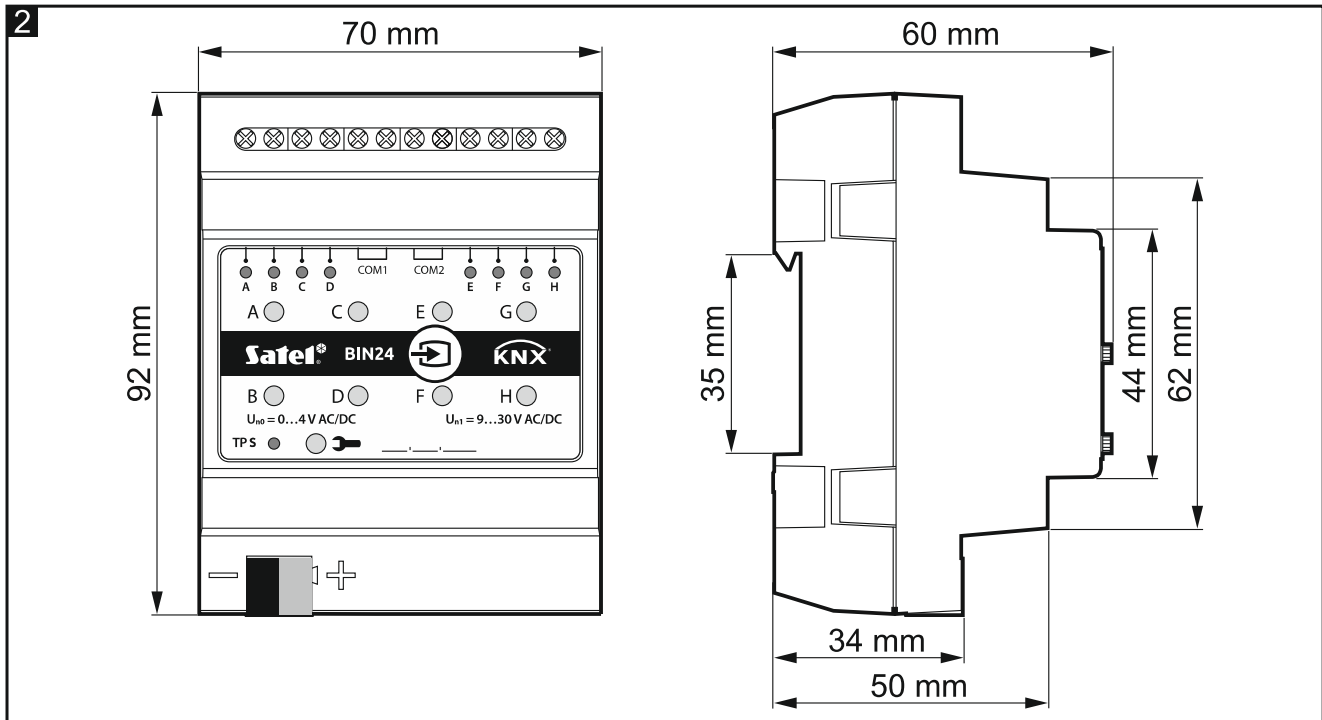
- Communication with the KNX bus via integrated bus connector.
- Definable channel polarity (NO / NC).
- Configurable time of short / long button press.
- Virtual channels for receiving telegrams from other KNX bus devices.
- Virtual logic channels for creating logical links between module channels.
- Virtual timer channels for creating time links.
- Possibility to define 20 function blocks that perform one of the available functions:
  - switch / value transmitter,
  - edge response,
  - dimmer,
  - shutter controller,
  - switching sequence,
  - counter,
  - scene controller.
- Possibility to control each function block using any channel.
- Control of several function blocks using one channel.
- Control of lighting and shutters using 1 or 2 buttons (channels).
- Ability to call a scene from any channel by using 8-bit commands.
- Manual operation of physical channels status by using buttons on the enclosure.
- LEDs to indicate status of physical channels.
- Module configuration by using ETS program.
- Mounted in distribution boxes, on DIN rail (35 mm).

## 2. Description



- ① physical input terminals.
  - ② green LEDs indicating the status of module physical channels. One channel status LED is assigned to each channel:  
ON – channel is ON,  
OFF – channel is OFF.
  - ③ buttons to manually switch the state of physical channels (to simulate changes on physical inputs).
- i** | Each button can work as monostable or bistable switch. How the button will operate can be defined separately for each channel in ETS program, “Physical channels” tab.
- ④ red LED – is ON when physical address is being set using the ETS program. Setting the address may be activated remotely from the ETS program or manually, using the  button on the enclosure.
  - ⑤ programming button (used when setting the physical address).
  - ⑥ terminal to connect KNX bus.

## 2.1 Enclosure



The module takes up 4 units on the 35 mm DIN rail.

## 3. Installation



**Disconnect power before making any electrical connections.**

The module is designed for indoor installation, in spaces with normal air humidity, e.g. in distribution boxes on 35 mm DIN rail.

1. Mount the module on the mounting rail.
2. Connect the devices to the terminals. For designations of the terminals see the front panel.



*All connections should be made as recommended in section "Wiring diagram".*

3. Use the connection terminal to connect the KNX bus cable to the module.



*The module is supplied with voltage from the KNX bus and requires no additional power supply.*

4. Connect a computer running ETS program to the KNX bus and configure the module.



*To configure the module, you will require a computer running the ETS program version 5.5 or newer, provided with USB or Ethernet (TCP/IP) connector. The SATEL ETS application file, which can be downloaded from [www.satel.eu/ets](http://www.satel.eu/ets), must be imported into the program.*

For detailed information on how to configure the module, refer to section "Configuring the module".

### 3.1 Wiring diagram

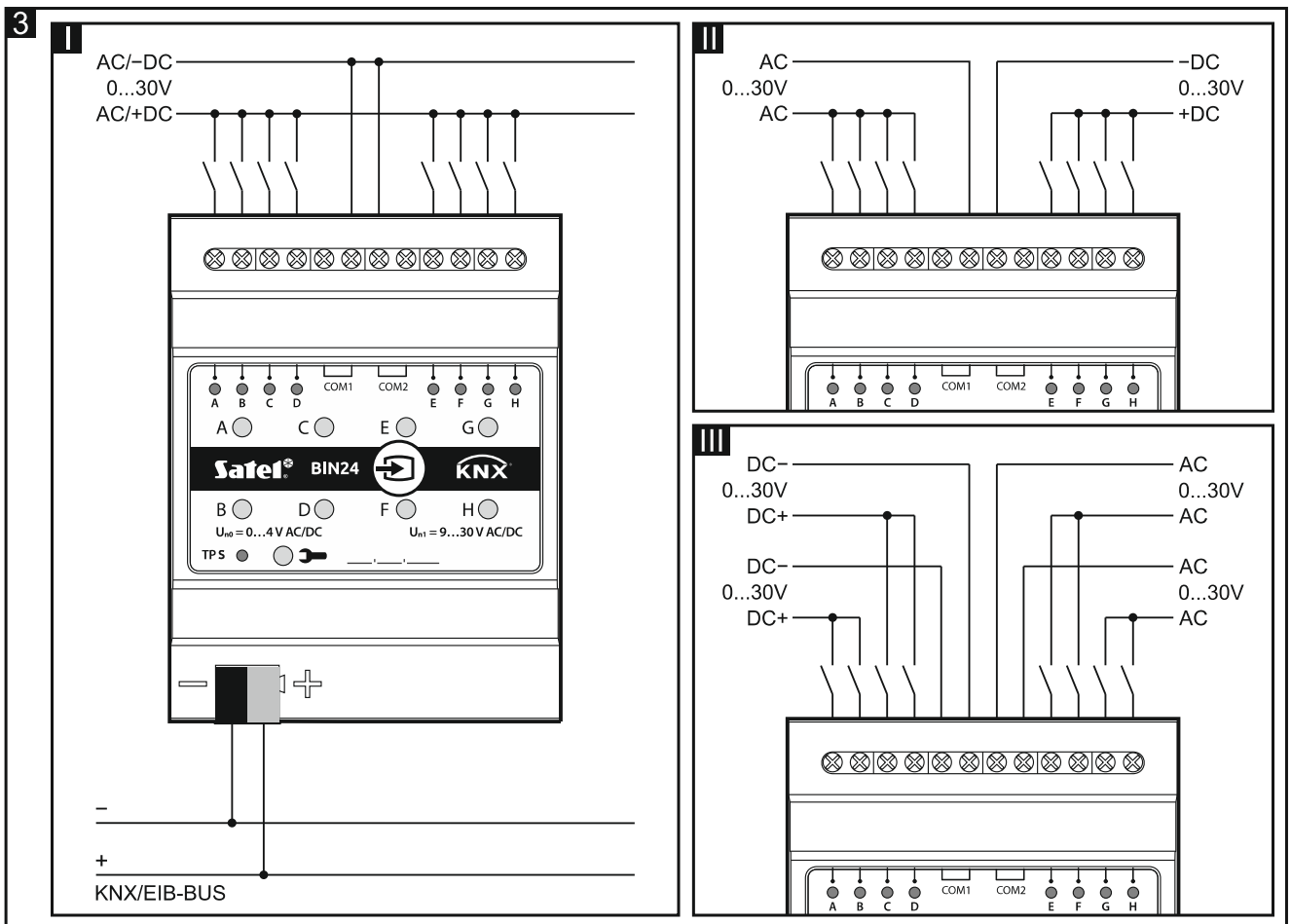
The KNX-BIN24 module has two independent circuits. Each circuit includes four inputs: A, B, C, D – circuit 1, and E, F, G, H – circuit 2. DC or AC voltage can be applied to each module input, allowable input voltage ranging from 0 to 30 V.

**i** *Input voltage from 0 to 4 V is interpreted as “0” and voltage from 9 to 30 V – as “1”. To avoid malfunctioning of the module, voltage ranging from 4 to 9 V must not be applied to any input.*

Each module input can be connected to a different current source, but one type of voltage, either DC or AC, must be applied to inputs belonging to one circuit.

**Fig. 3** shows some examples of module connection:

- I – voltage from one source, AC or DC, is applied to all inputs of the module,
- II – voltage from two different sources is applied to inputs of circuits 1 and 2 (DC voltage is applied to inputs of one circuit, and AC voltage to inputs of the other circuit),
- III – voltage from different sources is applied to inputs within one circuit (DC voltage is applied to inputs of one circuit, and AC voltage to inputs of the other circuit).



### 4. Configuring the module

Working parameters of the module are defined in the ETS program (version 5.5 or higher). The tabs for defining individual parameters are built dynamically. Depending on the selected options, further parameters may be displayed in the tabs or further tabs may be displayed when defining parameters. Moving the cursor over the selected parameter will display a tooltip with default value or value range for that parameter.

Configuring the module can be divided into three steps:

- Configuring global parameters.
- Configuring channels.
- Configuring function blocks.

There are four channel groups in the module:

- **physical** – convert electric signals from physical inputs (terminals) into events that can be handled by the module (8 channels), see “Physical channels”.
- **virtual** – convert telegrams received from other KNX bus devices and the module function blocks into events that can be handled by the module (8 channels), see “Virtual channels”.
- **logic** – special virtual channels that enable logical links to be created between the module channels (4 channels), see “Logic channels”.
- **timer** – special virtual channels that enable the status of channels of the module or KNX bus devices (4 channels) to be cyclically switched, see “Timer channels”.

The following events can be generated in each channel:

- rising edge,
- falling edge,
- button short pressing,
- button long pressing (holding).

Additionally, each channel store information about its own current state (Off / On).

You can define 20 function blocks in the module. The function block can execute one on the following functions:

- switch / value transmitter,
- edge response,
- dimmer,
- shutter controller,
- switching sequence,
- counter,
- scene controller.

Each function block can be controlled by using events from any module channel, and events from one channel can be used to control several function blocks at the same time.

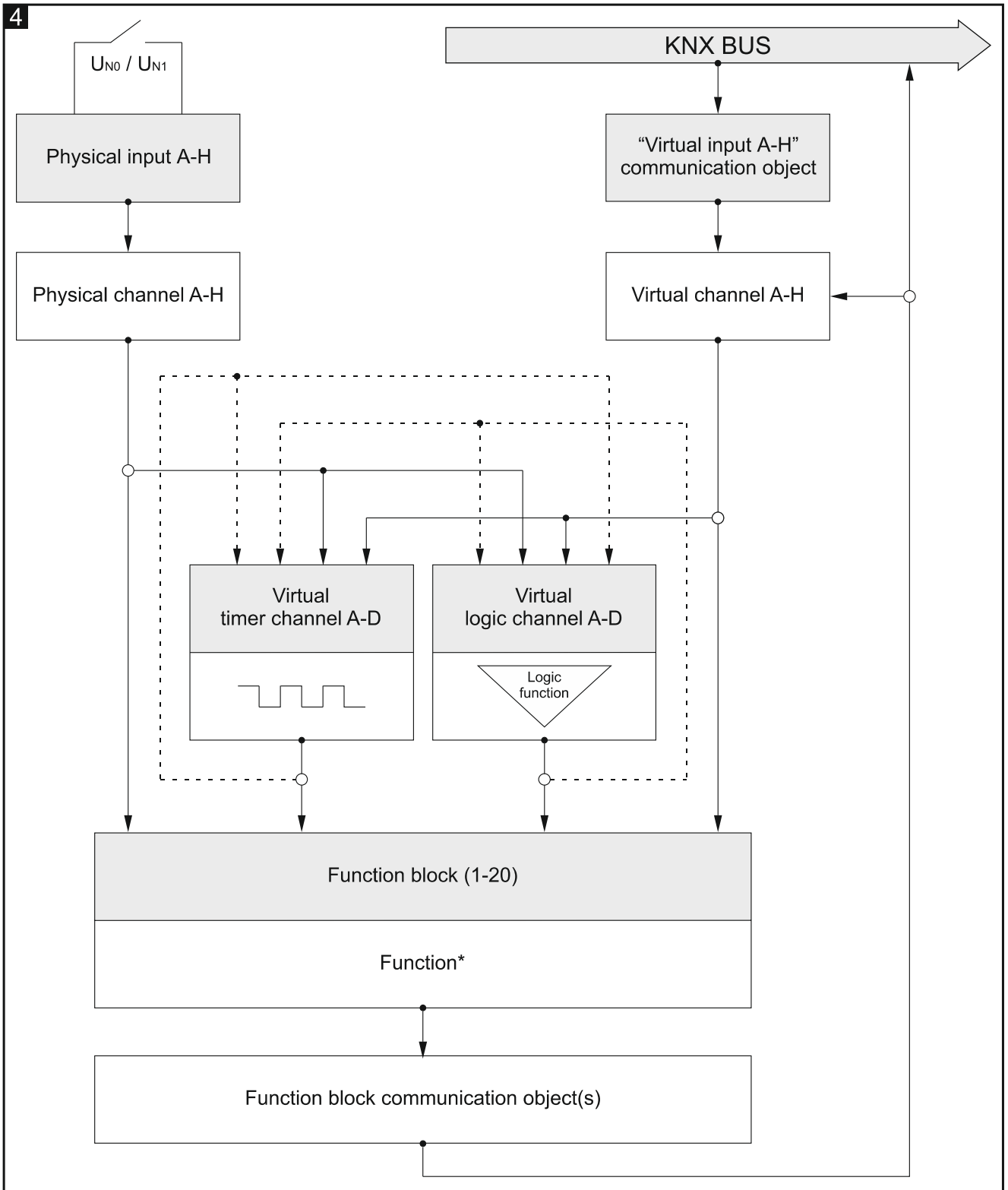


*Function blocks executing different functions can be controlled by events from one channel.*

Events generated in the module channels can also be control signals for the timer channels, and the state of channels can be linked in the logic function defined in the logic channel. Next sections present a schematic diagram of the module (**Fig. 4**) and a diagram of the starting configuration (**Fig. 5**) that was defined in the module ETS application. Section “Configuration examples” shows an example of configuration in which one channel is used to control several function blocks at the same time as well as examples of using the virtual, logic and timer channels.



### 4.1 Schematic diagram of the module



\* function: switch / value transmitter, edge response, dimmer, shutter controller, switching sequence, counter, scene controller.

### 4.2 Starting configuration

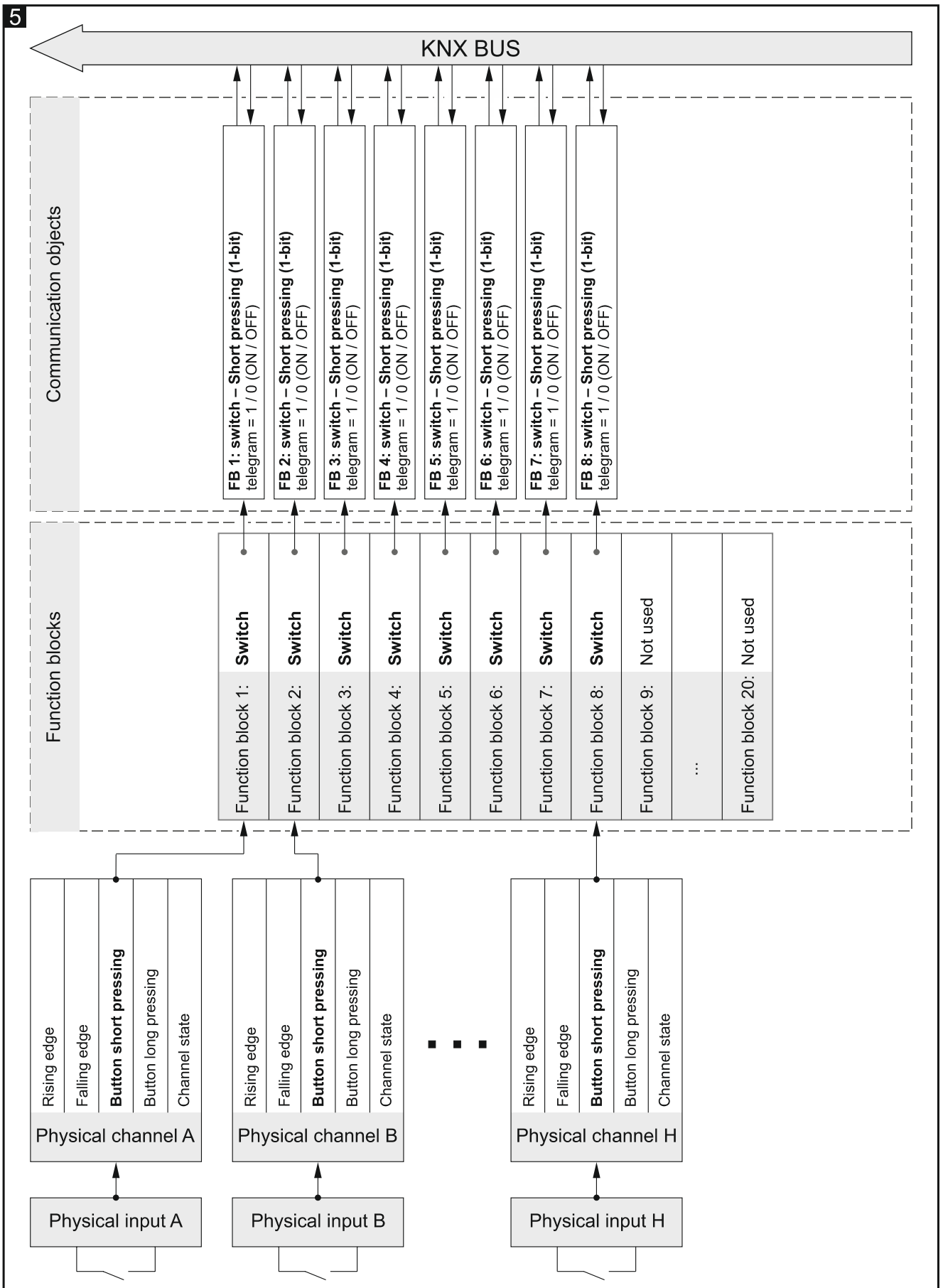


*In the ETS application for configuring the module working parameters, the starting configuration has been defined, including the working parameters of physical channels and 8 function blocks that execute the switch function (Function block 1...8: switch). If*

*the module is programmed without modifying the starting configuration, each physical channel of the module will control one function block executing the bistable switch function (ON / OFF).*

*The module starting configuration can be modified by the user at will.*

Presented below (**Fig. 5**) is a block diagram with starting configuration in which physical channels of the module receive electric signals from the physical inputs and convert them into events handled by the module. In the starting configuration, short button press events generated in each physical channel of the module (A – H) control one function block (Function block 1...8: switch), and thus “ON” / “OFF” telegrams are sent to the KNX bus.



See also other “Configuration examples”.

### 4.3 Configuring global parameters

The main working parameters of the module are defined in the “Global” tab, which is presented below.

Global	Startup delay	0	Seconds
+ Channels	Number of virtual inputs	0	
+ Function blocks			

**Startup delay** – time period by which the module startup will be delayed after power-on (0 - 65535 [seconds]). Value “0” will disable the delay.

During the delay period, the telegram handling is stopped and the channels do not change their state. The module sends no telegrams to the KNX bus. After the delay expires, the telegrams will be sent and the channel state will be set according to the parameters defined. If any telegrams from the bus with queries about the status of communication objects reach the module during the delay period, they will be stored. Replies to these telegrams are sent after expiry of the delay time.



*The startup delay may be used to reduce load on the KNX bus and supply circuit after power-on.*

**Number of virtual inputs** – the number of “Virtual input A...H” 1-bit communication objects that will be enabled in the module (0 – 8). The objects allow the module to receive telegrams from the KNX bus (see “Virtual channels”).

### 4.4 Configuring channels

You can configure the channels in the “Channels” tab. Physical, virtual, logic and timer channels are grouped in separate tabs. Configuring is described based on the first channel from each group, because it is done in the same way for all channels from the given group.

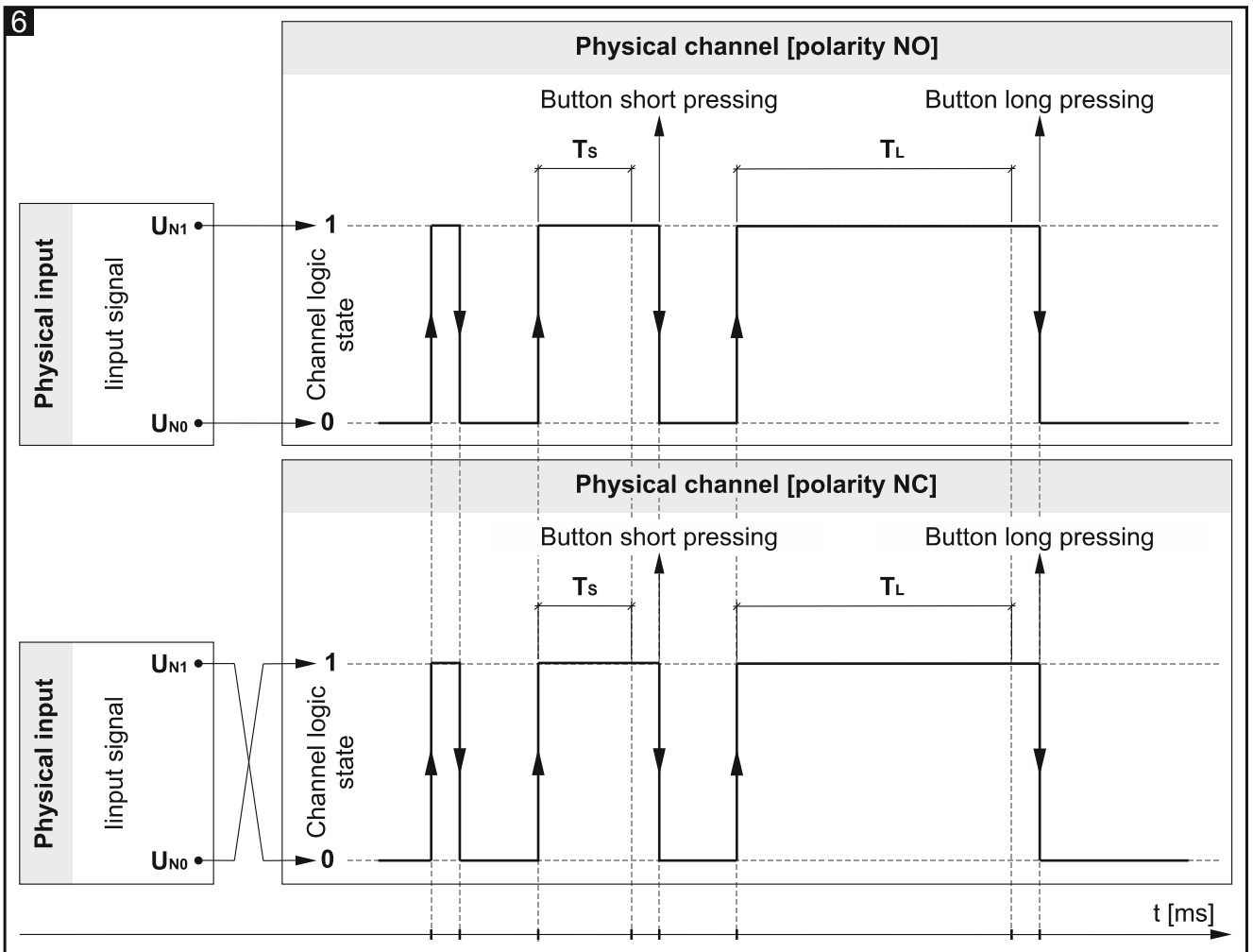
#### 4.4.1 Physical channels



*Working parameters of the physical channels have been defined in the starting configuration, which is saved in the ETS module configuring application. If the module has been programmed without modifying the starting configuration, the physical channels will have the following parameters: Polarity = NO; Time of button short pressing = 100 ms; Time of button long pressing = 1000 ms; Behavior of manual override button = Monostable.*

One physical channel (A – H) is assigned to each physical module input. A change of electric input signal ( $U_{N0} / U_{N1}$ ) will change the channel logic state (0 / 1). These changes in the channel will generate events supported by the module. The event generating scheme is shown in **Fig. 6**, where: change of the channel logic state from 0 to 1 = rising edge, from 1 to 0 = falling edge,  $T_S$  – short button press duration,  $T_L$  – long button press duration.

For each physical channel you can define its polarity (NO / NC), short and long button press duration and behavior of manual channel status override button.



### Description of parameters

Global	Polarity	<input checked="" type="radio"/> NO <input type="radio"/> NC
- Channels	Time of button short pressing (n * 10 ms)	10
- Physical Channels	Time of button long pressing (n * 10 ms)	100
Channel A	Behavior of manual override button	<input checked="" type="radio"/> Monostable <input type="radio"/> Bistable
Channel B	Channel events generating:	
Channel C	Button short pressing – when state on physical input changes from '0' to '1' and back to '0' between short & long pressing time	
Channel D	Button long pressing – when state on physical input changes from '0' to '1' and it stays on until the long pressing time expires	
Channel E	Rising edge – when state on physical input changes from '0' to '1'	
Channel F	Falling edge – when state on physical input changes from '1' to '0'	
Channel G		

**Polarity** – defines how the channel logic state will change in response to a signal change on physical input (**Fig. 6**):

Polarity	Input signal	Channel logic state
NO	$U_{N0}$	0
	$U_{N1}$	1
NC	$U_{N0}$	1
	$U_{N1}$	0

**Time of button short pressing ( $n * 10$  ms)** – time that must elapse during a change of the channel logic state from “0” to “1” and back to “0” for short button press to be generated in the channel – see **Fig. 6** (0 – 255; default = 10). Button press duration is calculated as product of  $n * 10$  ms, where:  $n$  = value entered in the field.

**Time of button long pressing ( $n * 10$  ms)** – time that must elapse during a change of the channel logic state from “0” to “1” and back to “0” for long button press to be generated in the channel – see **Fig. 6** (0 – 255; default = 100). Button press duration is calculated as product of  $n * 10$  ms, where:  $n$  = value entered in the field.

**Behavior of manual override button** – behavior of the channel state manual override button (to simulate changes on the physical input):

**Monostable** – pressing the button changes the channel state, releasing the button restores the previous state of the channel.

**Bistable** – pressing the button changes the channel state.

#### 4.4.2 Virtual channels

Enable events supported by the module to be generated based on the telegrams received from the KNX bus. The channel virtual input can be the “Virtual input A...H” 1-bit communication object or 1-bit function block object that has been activated in the program, e.g. the “FB 1: switch – Short pressing (1-bit)” object. Changing the state of the object (virtual input) generates events in the channel according to the scheme below:

- **rising edge** – when state changes from “0” to “1”.
- **falling edge** – when state changes from “1” to “0”.
- **button short pressing** – on receiving a telegram, regardless of whether or not it will change the object state.
- **button long pressing** – receiving a telegram by the object triggers countdown of that object monitoring time (“Time of button long pressing” field). If the object receives no other telegram within that time period, a long press event is generated in the channel.

For each virtual channel, you can select a virtual input (communication object) and define the channel polarity as well as long button press duration.

## Description of parameters

Global	Polarity	<input checked="" type="radio"/> NO <input type="radio"/> NC
- Channels	Time of button long pressing	0
+ Physical Channels	Channel input virtual	Virtual input A (Group Object 1)
- Virtual Channels	Channel events generating:	
Channel A	Button short pressing – when the object receives a telegram (regardles whether value changed or not)	
Channel B	Button long pressing – when the object receives not any telegram in time of button long pressing	
Channel C	Rising Edge – when value of object changes from '0' to '1'	
Channel D	Falling Edge – when value of object changes from '1' to '0'	
Channel E		

**Polarity** – defines how the channel logic state will change in response to a change of state of the communication object selected in the field “Channel virtual input”:

Polarity	Object (input) state	Channel logic state
NO	0	0
	1	1
NC	0	1
	1	0

**Time of button long pressing** – object monitoring time selected in the “Virtual input” field (0 - 65535 [seconds]). If value “0” is selected, the long button press event cannot be generated in the channel.

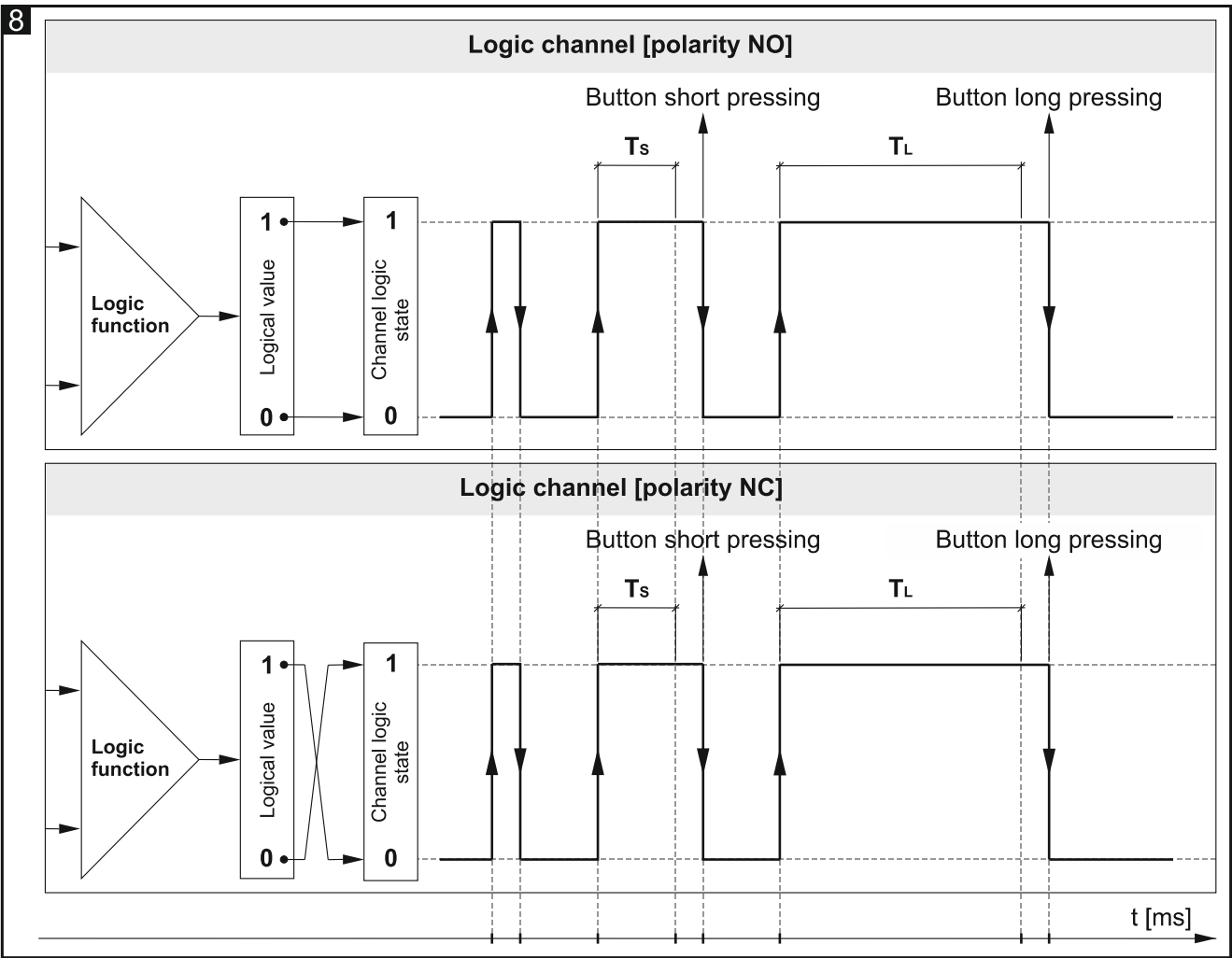
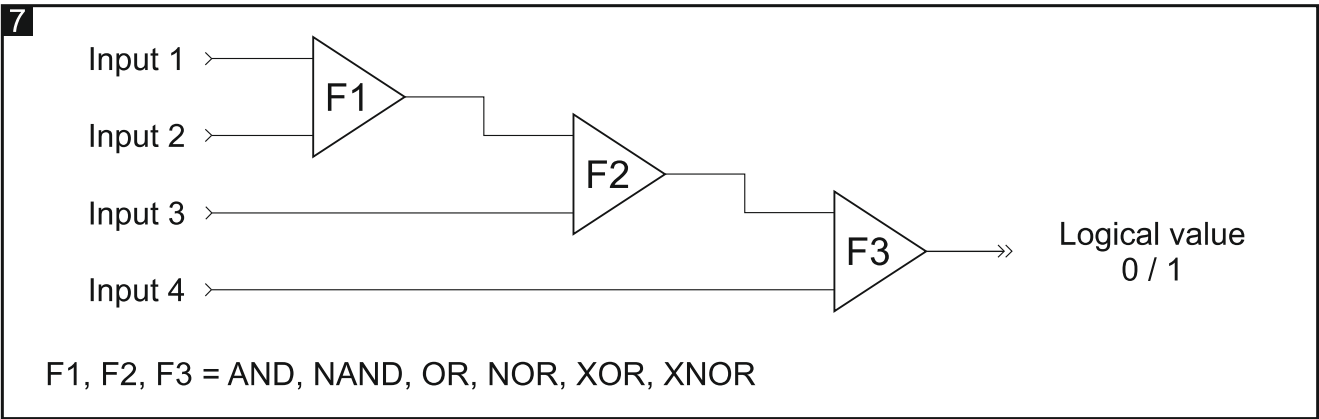
**Channel virtual input** – select the communication block that will be a virtual input of the channel (changing the object state will generate channel event). You can select the “Virtual input A...H” object or communication object of function block which has been activated in the program (e.g. “FB 1...20: switch”).



Only a 1-bit communication object may be a virtual channel input.

### 4.4.3 Logic channels

You can define 3 logical links in each logic channel. Using the available logic functions AND, NAND, OR, NOR, XOR, XNOR, you can create a combination function in which the state of different module channels can be linked (see **Fig. 7**, where: “Input 1...4” = state of physical / virtual / logical / timer channel). Changing the logical value (0 / 1) resulting from these links changes the channel logic state. These changes generate the channel events which are handled by the module. The event generating scheme is shown in **Fig. 8**, where: change in channel logic state from 0 to 1 = rising edge, from 1 to 0 = falling edge, **T<sub>S</sub>** – short button press duration, **T<sub>L</sub>** – long button press duration.



*Because the channel logic states instead of events generated in the channels are the input signals for logic functions, the logic channel can respond more quickly to changes occurring in these channels.*



## Description of parameters

Global	Polarity	<input checked="" type="radio"/> NO <input type="radio"/> NC
- Channels	Time of button short pressing (n * 10 ms)	5
+ Physical Channels	Time of button long pressing (n * 10 ms)	100
+ Virtual Channels	Number of logic functions	1
- Logic Channels	Logic function 1	Not used
Channel A	Input 1	Physical Channel A
Channel B	Input 2	Physical Channel A
Channel C		
Channel D		
+ Timer Channels	Channel events generating:	
	Button short pressing – when state on physical input changes from '0' to '1' and back to '0' between short & long pressing time	
	Button long pressing – when state on physical input changes from '0' to '1' and it stays on until the long pressing time expires	
	Rising edge – when value of logic operations on channel changes from '0' to '1'	
	Falling edge – when value of logic operations on channel changes from '1' to '0'	
+ Function blocks		

**Polarity** – defines how the channel logic state will change in response to the logic function result (Fig. 8):

Polarity	Logic function result	Channel logic state
NO	0	0
	1	1
NC	0	1
	1	0

**Time of button short pressing (n \* 10 ms)** – time that must elapse when changing the channel logic state from “0” to “1” and back to “0” for the short button press event to be generated in the channel – see Fig. 8 (0 – 255; default = 5). The press duration is calculated as product n \* 10 ms, where: n = value entered in the field.

**Time of button long pressing (n \* 10 ms)** – time that must elapse when changing the channel logic state from “0” to “1” and back to “0” for the long button press event to be generated in the channel – see Fig. 8 (0 – 255; default = 100). The press duration is calculated as product n \* 10 ms, where: n = value entered in the field.

**Number of logic functions** – the number of logic functions used in the channel (default = 1, maximum = 3). Selecting values 2 and 3 will display in the tab further fields for defining the logic functions 2 and 3.

**Logic function 1/2/3** – type of function (AND / NAND / OR / NOR / XOR / XNOR).

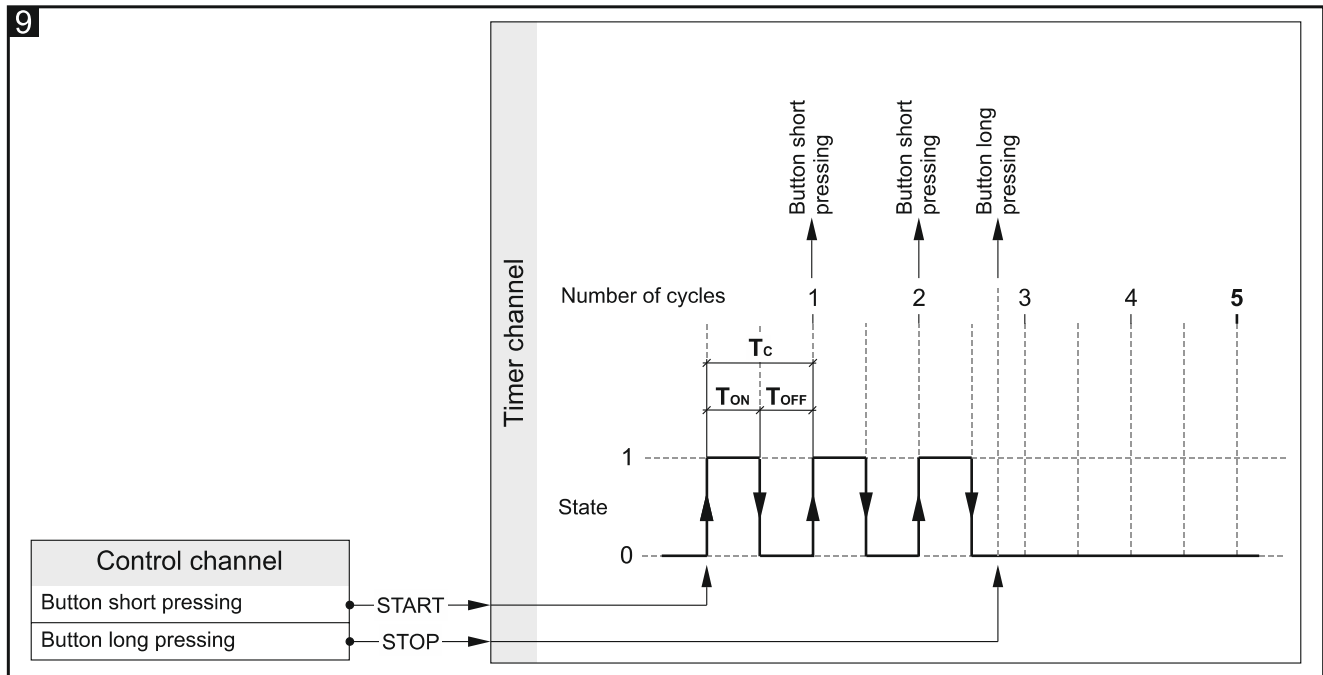
**Input 1/2/3/4** – channel whose state will be input value for logic function (Physical channel A...H / Virtual channel A...H / Logic channel A...D / Timer channel A...D).

### 4.4.4 Timer channels

They generate events in defined time intervals which makes it possible to cyclically switch the state of other devices on the KNX bus. Each timer channel switches its state “0” / “1” (ON / OFF) cyclically. The time of one cycle  $T_c$  is defined in the program as a sum of times defined in the fields “Timer ON duration”  $T_{ON}$  and “Timer OFF duration”  $T_{OFF}$ , whereas the maximum number of cycles  $L_c$  which will turn off the timer is defined in the “Timer cycles count” field. You can select in the program the channel and a few events to control the timer channel.

Occurrence of the first event of this pair in the control channel will enable the timer channel, and the second event may disable the timer channel before the maximum number of cycles  $L_c$  is reached (see **Fig. 9**, where:  $L_c = 5$ ; control pair = short and long press of button). In the “Timer autostart on bus recovery” field, you can define whether the timer channel is to be enabled immediately on starting the module, or only after the selected event is generated in the control channel.

Changing the timer channel state will generate in the channel events that are handled by the module cyclically. The event generating diagram is shown in **Fig. 9**, where: state change from 0 to 1 = rising edge, and from 1 to 0 = falling edge.



A long button press event can only be generated in the channel, when the “Yes” option is selected for the “Long pressing at the end” parameter, and the value defined in the field “Timer cycles count” field is higher than zero.



*To enable synchronization with the other module channels and other devices on the KNX bus, the timer channels work in much the same way as the physical channels, changing their state from OFF to ON etc. The difference between the physical channels and the timer channels is that the latter change their state in defined time intervals.*

## Description of parameters

Global	Timer ON duration	1	Seconds
- Channels	Timer OFF duration	1	Seconds
+ Physical Channels	Timer cycles count	0	
+ Virtual Channels	Timer control event pair	<input checked="" type="radio"/> button short pressing / button long pressing <input type="radio"/> rising edge / falling edge	
+ Logic Channels	Timer trigger channel	Physical Channel A	
- Timer Channels	Timer autostart on bus recovery	<input checked="" type="radio"/> No <input type="radio"/> Yes	
Channel A	Long pressing at the end	<input checked="" type="radio"/> No <input type="radio"/> Yes	
Channel B	Channel events generating:		
Channel C	Button short pressing – on every timer cycle		
Channel D	Button long pressing – at the end of timer cycles		
+ Function blocks	Rising Edge – on internal's clock rising edge		
	Falling Edge – on internal's clock falling edge		

**Timer ON duration** – the time for which the channel will be enabled during the cycle (0 - 30000 [seconds]).

**Timer OFF duration** – the time for which the channel will be disabled during the cycle (0 - 30000 [seconds]).

**Timer cycles count** – the number of cycles after which the timer will be disabled (0 – 255). If you select “0”, the timer will not be disabled (no long button press event will be generated in the channel).

**Timer control event pair** – you can select the pair of events to control the timer. Occurrence of the first event from the pair in the channel selected in the “Timer trigger channel” field will enable the timer, and the second one will disable it. You can select:

**Button short pressing / Button long pressing**

**Rising edge / Falling edge**

**Timer trigger channel** – the channel whose events selected in the “Timer control event pair” field will enable / disable the timer channel (Physical channel A...H / Virtual channel A...H / Logic channel A...D / Timer channel A...D).

**Timer autostart on bus recovery** – response of the channel to the bus power supply restoration (No / Yes). If you select “Yes”, the timer channel will be enabled immediately on starting the module.

**Long pressing at the end** – if you select “Yes”, a long button press event can be generated in the channel after the last cycle – disabling the timer (No / Yes). The event will only be generated, if the value defined in the “Timer cycles count” field is higher than zero.

## 4.5 Configuring function blocks

20 function blocks with unique numbers from 1 to 20 can be defined in the module. The function block can be enabled in the “General” tab (that can be seen after opening the “Function blocks” tab), by selecting one of the following functions: switch/value transmitter, edge response, dimmer, shutter controller, switching sequence, counter, scene controller, for the block with the given number (“Function block 1...20” field). For each enabled function

block, communication objects are enabled (the number of objects depends on the selected function). A tab is displayed for configuration of the block (e.g. “1: Switch”) and linking it to the channel or channels that will be used to control it (each function block can be controlled by using events from any channel, and events from one channel can be used to control several function blocks at the same time – see “Configuration examples”).



*In the starting configuration of the module (ETS application file), 8 function blocks have been defined that execute the switch function and are controlled by the short button press events, activated in the physical channels of the module (A...H) - see “Starting configuration”.*

Shown below is the “General” tab for enabling functions in the module.

Global	Number of function blocks	8
+ Channels	Function block 1	Switch
- Function blocks	Function block 2	Switch
	Function block 3	Switch
General	Function block 4	Switch
1: Switch	Function block 5	Switch
2: Switch	Function block 6	Switch
3: Switch	Function block 7	Switch
4: Switch	Function block 8	Switch

**Number of function blocks** – number of the function blocks that can be enabled in the module (0...20, default = 8, maximum = 20). The number of displayed “Function block 1...20” fields depends on selected value.

**Function block 1...20** – you can select the function to be executed by the block:

**Not used** – the function block will not be used.

**Switch** – functionality of bistable switch or value transmitter. Selecting this function will enable the “FB 1...20: switch – Short press (1-bit)” communication object and display the “1...20: Switch” tab.

**Edge** – the rising edge / falling edge events are used to control switching. Selecting this function will enable the “FB 1...20: edge – Rising / falling (1-bit)” communication object and display the “1...20: Edge” tab.

**Dimmer control** – for stepless control of lighting. Selecting this function will enable the “FB 1...20: dimmer – Switch” and “FB 1...20: dimmer – Dim relative” communication objects and display the “1...20: Dimmer” tab.

**Shutter control** – for stepless control of roller shutter. Selecting this function will enable the “FB 1...20: shutter – Move up/down” and “FB 1...20: shutter – Stop/step up/down” communication objects and display the “1...20: Shutter” tab.

**Switching sequence** – sequentially switches the 1-bit communication objects on the KNX bus. Selecting this function will enable the “FB 1...20: sequence – Object 1” and “FB 1...20: sequence - Object 2” communication objects and display the “1...20: Sequence” tab.

**Counter** – counts the events generated in the module channels. Selecting this function will enable the communication objects “FB 1...20: counter: - Current value”, “FB 1...20:

counter – Set value” and “FB 1...20: counter – Indicates overflow” and display the “1...20: Counter” tab.

**Scene** – calls one of the 20 scenes to which the module can be linked. Selecting this function will enable the “FB 1...20: scene – Scene trigger” communication object and display the “1...20: Scene” tab.



The next sections describe how to configure the function blocks enabled in the module. As the configuring method depends not on the number of the block, but on its function, configuring is discussed based on the function block number “1” (tab “1: Name”, where: Name = Function implemented by the given block, e.g. “Switch”). The names of communication objects enabled for the block begin with the “FB 1” symbol.

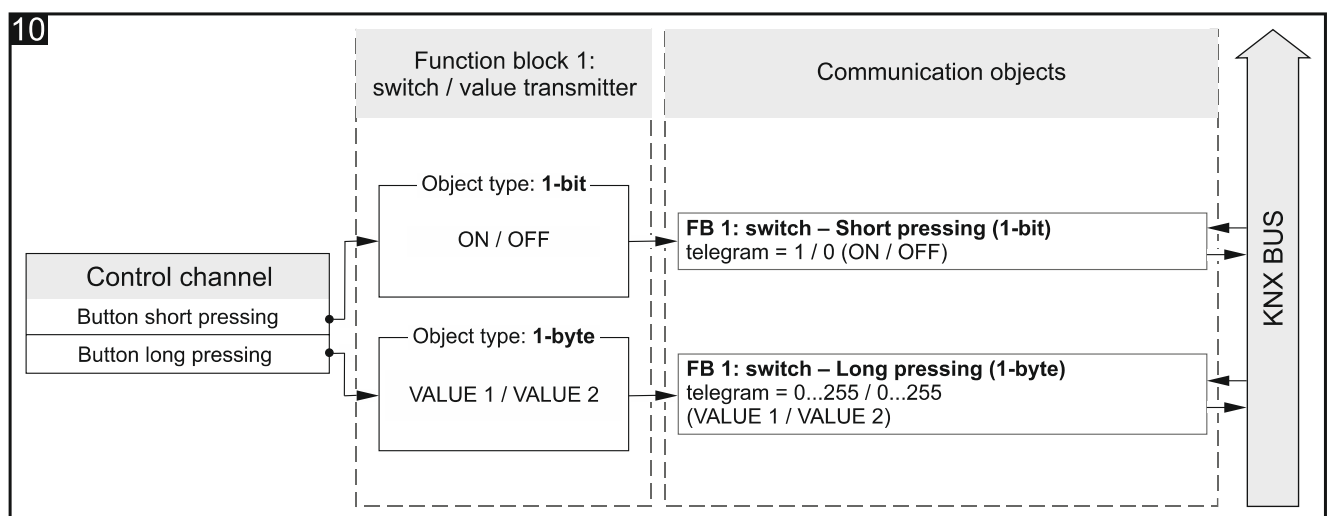
### 4.5.1 Switch

Enables implementation of the bistable switch or value transmitter functionality. By default, the function block is controlled by the short button press events, where switching is done by using the “FB 1: switch – Short pressing (1-bit)” communication object. If you select “Yes” for the “Event on button long pressing” parameter, the function block can be controlled by using the long button press events – “FB 1: switch – Long pressing (1-byte)” object. Each object can implement the functionality of switch or value transmitter. If in the “Short pressing: object type” and “Long pressing: object type” fields, the “1-byte – percentage”, “4-bit – dimming control” or “4-bit – shutter control” option is selected, the objects can implement the value transmitter functionality by sending telegrams to the bus with values defined in the program.

**Fig. 10** shows an example of function block which is controlled by short / long button press events. In the example, the “FB 1: switch – Short pressing (1-bit)” object, defined as 1-bit object, sends to the bus “on” / “off” telegrams, whereas the “FB 1: switch – Long pressing (1-byte)” object, defined as 1-byte object, sends telegrams with values defined in the program (0...255).



Value of the “FB 1: switch - Short pressing” and “FB 1: switch - Long pressing” communication object can be changed by a telegram from the KNX bus. This allows the value of these objects to be updated each time when the switching is triggered by another communication object from the same address group (e.g. another button).



Configuration for the example on **Fig. 10** in ETS: Event on button long pressing = yes; Short pressing: object type = 1-bit; Long pressing: object type = 1-byte.

## Description of parameters

Global	Control channel	Physical Channel A
+ Channels	Event on button long pressing	<input checked="" type="radio"/> no <input type="radio"/> yes
- Function blocks	Interval of value cyclic sending (short pressing)	0 Seconds
General	Short pressing: object type	1-bit
1: Switch	Short pressing: value 1	<input type="radio"/> off <input checked="" type="radio"/> on
2: Switch	Short pressing: value 2	<input checked="" type="radio"/> off <input type="radio"/> on
3: Switch		

**Control channel** – events generated in the channel control switching / setting the value (Physical Channel 1...8 / Virtual Channel 1...8 / Logic Channel 1...4 / Timer Channel 1...4).

**Event on button long pressing** – defines the control method:

**No** – control by using short button press. Default value.

**Yes** – control by using short and long button press.

**Interval of value cyclic sending (short pressing)** – frequency with which the “FB 1: switch – Short pressing” object will send a telegram to the bus with information about its current value (0 - 65535 [seconds]). Information contained in the telegram enables the object state to be monitored by other devices on the KNX bus. Value “0” disables the cyclic sending of information.

**Interval of value cyclic sending (long pressing)** – frequency with which the “FB 1: switch – Long pressing” object will send a telegram to the bus with information about its current value (0 65535 [seconds]). Information contained in the telegram enables the object state to be monitored by other devices on the KNX bus. Value “0” disables the cyclic sending of information.

**Short pressing: object type** – data type for the “FB 1: switch – Short pressing” communication object. It defines the range in which the object values can be defined:

**1-bit** – object can assume value 1 or 0.

**1-byte** – object can assume values from 0 to 255 range.

**1-byte – percentage** – object can assume values from 0 to 100 % range.

**4-bit – dimming control** – object can assume values ranging from maximum dimming up (100% Brighter) to maximum dimming down (100% Darker) and value “Stop dimming” – see Appendix “4-bit telegrams for dimming DOWN/UP”.

**4-bit – shutter control** – object can assume values ranging from shutter fully open (100% Up) to shutter fully closed (100% Down) and value “Stop” – see Appendix “4-bit telegrams for shutter control”.

**Short pressing: value 1** – value that the “FB 1: switch – Short pressing” communication object will assume in response to the first / odd short press of button in the “Control channel” (1-bit: Off / On (0 / 1); 1-byte: 0-255; 1-byte - percentage: 0...100 %; 4-bit - dimming control: 100...1% Brighter, Stop dimming, 1...100% Darker; 4-bit – shutter control: 100...1% Up, Stop, 1...100% Down).

**Short pressing: value 2** – value that the “FB 1: switch – Short pressing” communication object will assume in response to the second / even short press of button in the “Control channel” (1-bit: Off / On (0 / 1); 1-byte: 0-255; 1-byte - percentage: 0...100 %; 4-bit –

dimming control: 100...1% Brighter, Stop dimming, 1...100% Darker; 4-bit – shutter control: 100...1% Up, Stop, 1...100% Down).



*If the function block is to implement the functionality of a typical On / Off switch (1-bit object type), then the first and second value of the “FB 1: switch – Short pressing” communication object should be set as follows: “Short pressing: value 1” = On, “Short pressing: value 2” = Off.*

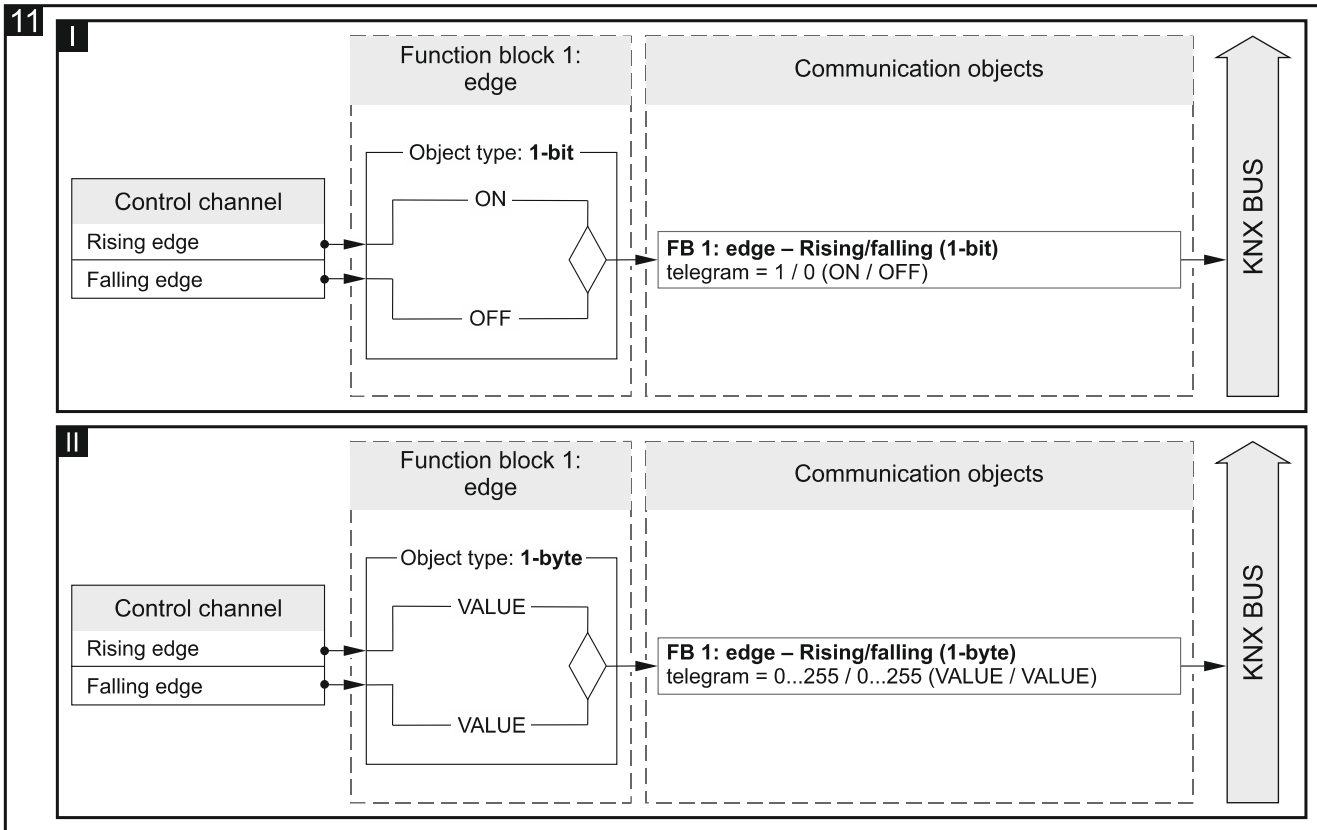
**Long pressing: object type** – data type for the “FB 1: switch – Long pressing” communication object. It defines the range within which the object values can be defined (1-bit / 1-byte / 1-byte – percentage / 4-bit – dimming control / 4-bit – shutter control; the same as for short press).

**Long pressing: value 1** – value that the “FB 1: switch – Long pressing” communication object will assume in response to the first / odd long press of button in the “Control channel” (1-bit: Off / On (0 / 1); 1-byte: 0-255; 1-byte – percentage: 0...100 %; 4-bit - dimming control: 100...1% Brighter, Stop dimming, 1...100% Darker; 4-bit – shutter control: 100...1% Up, Stop, 1...100% Down).

**Long pressing: value 2** – value that the “FB 1: switch – Long pressing” communication object will assume in response to the second / even long button press in the “Control channel” (1-bit: Off / On (0 / 1); 1-byte: 0-255; 1-byte – percentage: 0...100 %; 4-bit - dimming control: 100...1% Brighter, Stop dimming, 1...100% Darker; 4-bit – shutter control: 100...1% Up, Stop, 1...100% Down).

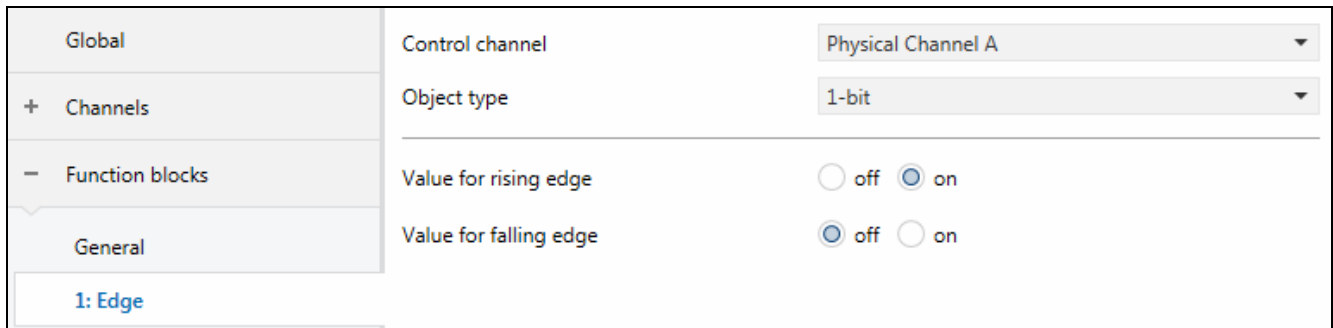
#### 4.5.2 Edge (monostable switch)

Makes it possible to control switching / value transmitting by using the rising edge / falling edge events. In response to occurrence of these events, the “FB 1: edge – Rising/falling (1-bit)” communication object can send an “ON” / “OFF” telegram to the KNX bus. If the “1-byte – percentage”, “4-bit – dimming control” or “4-bit – shutter control” option is selected in the “Object type” field, the object can send telegrams with values defined in the “Value for rising edge” and “Value for falling edge” fields. Depending on the selected type, the object implement the functionality of monostable switch (1-bit object - **Fig. 11-I**) or value transmitter (e.g. 1-byte object - **Fig. 11-II**) controlled by the rising / falling edge events.



Configuration for the example on **Fig. 11-I** in ETS: Object type = 1-bit. Configuration for the example shown on **Fig. 11-II** in ETS: Object type = 1-byte.

**Description of parameters**



**Control channel** – rising edge and falling edge events generated in the channel control the switching / value transmitting (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4).

**Object type** – data type for the “FB 1: edge - Rising / falling” communication object. It defines the range of values the object can assume:

**1-bit** – object can assume value 1 or 0.

**1-byte** – object can assume values from 0 to 255 range.

**1-byte – percentage** – object can assume values from 0 to 100 % range.

**4-bit – dimming control** – object can assume values from the range of maximum dimming up (100% Brighter) to maximum dimming down (100% Darker) and value “Stop dimming” – see Appendix “4-bit telegrams for dimming DOWN/UP”.

**4-bit – shutter control** – object can assume values within the range of shutter fully open (100% Up) to shutter full closed (100% Down) and value “Stop” – see Appendix “4-bit telegrams for shutter control”.



**Value for rising edge** – value that the “FB 1: edge - Rising/falling” communication object will assume in response to the rising edge event in the “Control channel” (1-bit: Off / On (0 / 1); 1-byte: 0-255; 1-byte – percentage: 0...100 %; 4-bit - dimming control: 100...1% Brighter, Stop dimming, 1...100% Darker; 4-bit – shutter control: 100...1% Up, Stop, 1...100% Down).

**Value for falling edge** – value that the “FB 1: edge - Rising/falling” communication object will assume in response to the falling edge event in the “Control channel” (1-bit: Off / On (0 / 1); 1-byte: 0-255; 1-byte – percentage: 0...100 %; 4-bit - dimming control: 100...1% Brighter, Stop dimming, 1...100% Darker; 4-bit – shutter control: 100...1% Up, Stop, 1...100% Down).

### 4.5.3 Dimmer

It allows you to turn lighting ON/OFF and to regulate its brightness. You can define in the program (“Two-channel mode” parameter) whether lighting control is to be implemented by using one or two channels. Turning on/off is implemented by the “FB 1: dimmer – Switch” 1-bit communication object, whereas the process of dimming brighter / darker is implemented by the “FB 1: dimmer – Dim relative” 4-bit object, where the “BRIGHTER” / “DARKER” telegram will change brightness by the value defined in the “Dimming up: step value” and “Dimming down: step value” fields. The “STOP” telegram terminates the process at the value reached.

The function block can be controlled by short/long press events and falling edge events.



*Value of the “FB 1: dimmer – Switch” communication object can be changed by telegram from the KNX bus. This enables its value to be updated each time when turning the light on/off is triggered by another communication object from the same address group (e.g. another button in the staircase).*

### Description of parameters

Global	Two-channel mode	<input checked="" type="radio"/> no <input type="radio"/> yes
+ Channels		Type A:
- Function blocks	Control scheme	<input checked="" type="radio"/> button short pressing = step; button long pressing = on / off
General		Type B:
1: Dimmer		<input type="radio"/> button short pressing = on / off; button long pressing = step; falling edge = stop
2: Switch	Control channel	Physical Channel A
3: Switch	Dimming up: step value	100% Brighter
4: Switch	Dimming down: step value	100% Darker

**Two-channel mode** – defines how the lighting is to be controlled:

**No** – two-channel mode disabled i.e. the lighting can be controlled by using one channel – see Fig. 12. Default value.

**Yes** – two-channel mode enabled i.e. the lighting can be controlled by using two channels – see Fig. 13.

**Control scheme** – selects the control scheme:

**Type A: Button short pressing = Step; Button long pressing = On / Off** – depending on the selected control method:

one-channel control – long button press will turn the lighting on /off. When the lighting is on, each short button press will dim the light down. When the lighting is off, each short button press will dim the light up.

two-channel control – long button press in the first channel (“Dimming up: control channel”) will turn the lighting on. Long button press in the second channel (“Dimming down: control channel”) will turn the lighting off. Short button press in the first channel (“Dimming up: control channel”) will dim the light up. Short button press in the second channel (“Dimming down: control channel”) will dim the light down.

**Type B: Button short pressing = On / Off; Button long pressing = Step; Falling edge = Stop** – depending on the selected control method:

one-channel control – short button press will turn the lighting on /off. When the lighting is on, the long button press will dim the light down. When the lighting is off, the long button press will dim the light up. Dimming up / down direction will change after each long press. Falling edge (button release) will stop dimming the light up / down.

two-channel control – short button press in the first channel (“Dimming up: control channel”) will turn the lighting on. Short button press in the second channel (“Dimming down: control channel”) will turn the lighting off. Long button press in the first channel (“Dimming up: control channel”) will dim the light up. Long button press in the second channel (“Dimming down: control channel”) will dim the light down. Falling edge (button release) in the channel will stop dimming the light up / down.

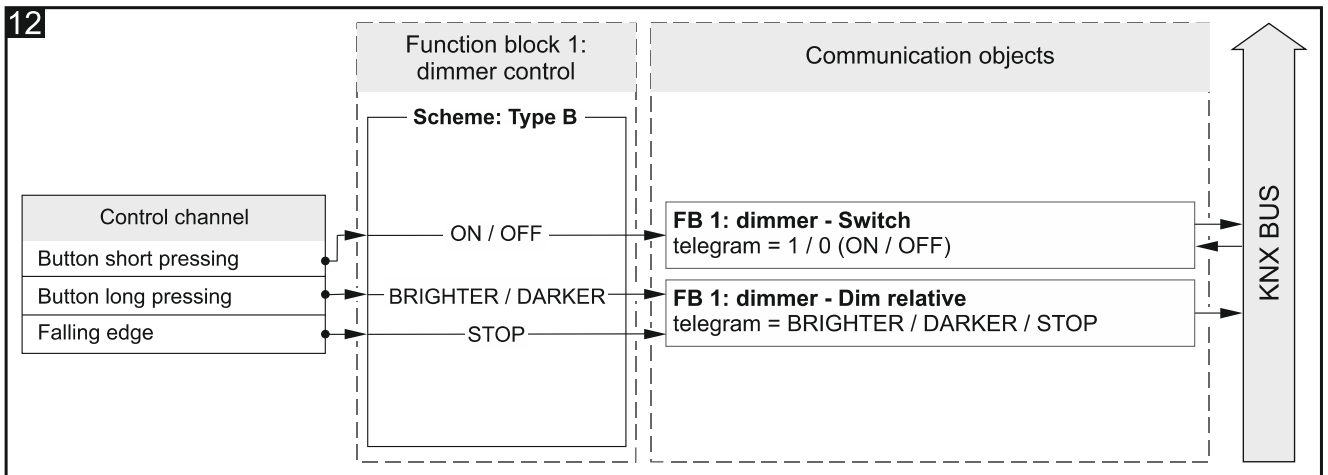


*Lighting control by using two channels can only be correctly implemented when two different channels are selected in the “Dimming up: control channel” and “Dimming down: control channel” fields.*

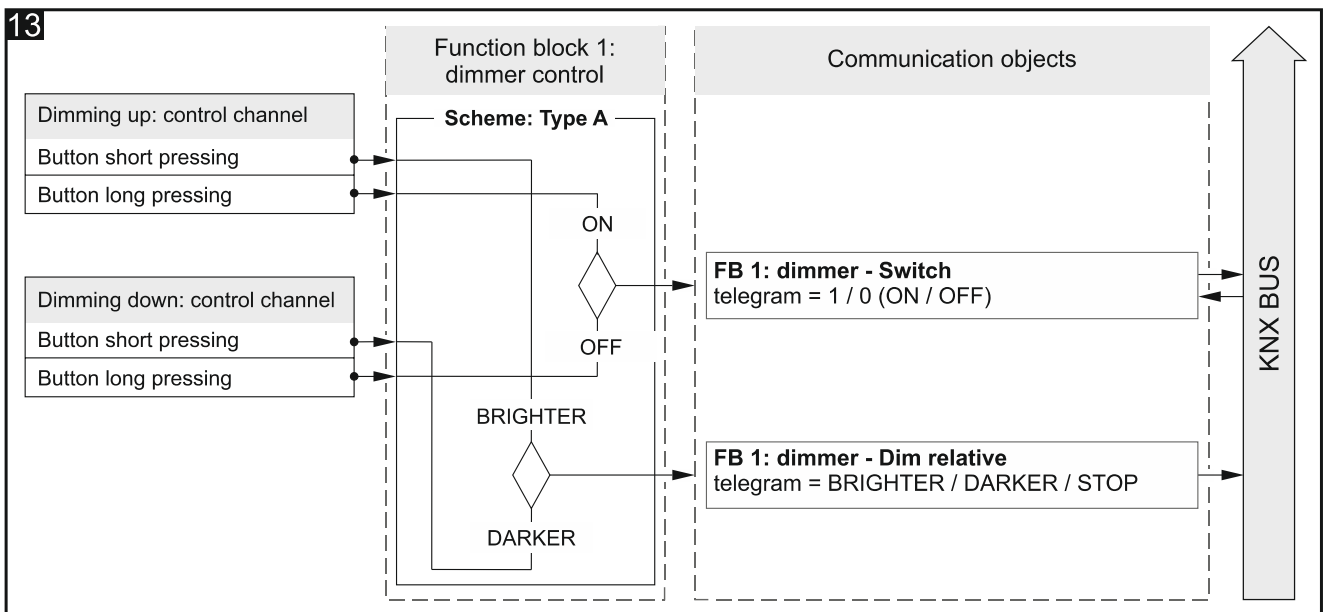
*“Type B” is the dedicated scheme to control lighting by using physical channels. It is recommended for this scheme that the dimming up/down step value be set at 100% (“Dimming up: step value” and “Dimming down: step value” fields). Thus the light dimming up / dimming down will only be stopped when the button is released by the user.*

*Please remember that the binary input module enables the signals from outside of the KNX system to be changed into control telegrams for other devices on the bus (e.g. for dimming actuators), but interpretation of these telegrams depends on how these actuators are configured.*

**Fig. 12** shows an example of lighting control using one channel (“Type B” control scheme) and **Fig. 13** – using two channels (“Type A” control scheme).



Configuration for example shown in **Fig. 12** in ETS: Two-channel mode = No; Control scheme = Type B.



Configuration for example shown in **Fig. 13** in ETS: Two-channel mode = Yes; Control scheme = Type A.

**Control channel** – events generated in the channel control lighting (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of one-channel control.

**Dimming up: control channel** – events generated in the channel turn the lighting on and dim it up (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of two-channel control.

**Dimming down: control channel** – events generated in the channel turn the lighting off and dim it down (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of two-channel control.



*Which events from the above channels change the value of “FB 1: dimmer – Switch” object and which change the value of “FB 1: dimmer – Dim relative” object is defined by the scheme selected for the “Control scheme” parameter.*

**Dimming up: step value** – the value by which lighting will be brightened when an event is generated in the control channel (“Control channel” or “Dimming up: control channel”) that changes the state of the “FB 1: dimmer – Dim relative” communication object (100%...1% Brighter, Stop - see Appendix “4-bit telegrams for dimming DOWN/UP”).

**Dimming down: step value** – the value by which lighting will be darkened when an event is generated in the control channel (“Control channel” or “Dimming down: control channel”) that changes the state of the “FB 1: dimmer – Dim relative” communication object (100%...1% Darker; Stop, - see Appendix “4-bit telegrams for dimming DOWN/UP”).

#### 4.5.4 Shutter

It allows you to control the movement of window coverings, such as roller shutters, blinds or shades. You can define in the program (“Two-channel mode” parameter) whether the shutter control is to be implemented by using one or two channels. The Up / Down movement of the shutter is implemented by the “FB 1: shutter - Move up/down” 1-bit communication object. The object starts the shutter movement, which can be stopped by the “STOP” telegram or when the shutter reaches its end point (after it is fully opened or closed). Stepwise adjustment of slats or stepwise movement of the whole shutter in the case of roller blind and shade is implemented by the “FB 1: shutter - Stop/step up/down” object.

The function block can be controlled by events of short button press / long press and falling edge.



*Value of the “FB 1: shutter – Move up/down” communication object can be changed by a telegram from the KNX bus. It allows its value to be updated each time when movement of the shutter is triggered by another communication object from the same address group (e.g. another button).*

#### Description of parameters

Global	Two-channel mode	<input checked="" type="radio"/> no <input type="radio"/> yes
+ Channels	Control scheme	Type A - 1-bit: button short pressing = step/stop; button long pressing = open/close
- Function blocks	Control channel	Physical Channel A
General		
1: Shutter		

**Two-channel mode** – defines how to control the shutter:

**No** – two-channel mode disabled, i.e. the shutter can be controlled by using one channel – see Fig. 14. Default value.

**Yes** – two-channel mode enabled, i.e. the shutter can be controlled by using two channels – see Fig. 15.

**Control scheme** – contents of the list changes depending on whether one-channel or two-channel mode has been selected.

In the case of one-channel mode, you can select:

**Type A – 1-bit: button short pressing = step/stop; button long pressing = open/close**  
– long button press will start shutter movement Up / Down. When the shutter/blind is fully closed, each short press will adjust the slats by one step up. When the shutter/blind is fully open, each short press will adjust the slats by one step down. When the shade (without slats) is fully open, each short press will move the shade by one step up. When the shade is fully open, each short press will move the shade by one step down. If the slates or shutter are moving, a short button press will stop them.

**Type B – 1-bit: button short pressing = open/close; button long pressing = step/stop**  
– the role of short / long button press events is reversed as compared to the “Type A – 1-bit” scheme.

**Type C – 4-bit: button short pressing = open/close; button long pressing = step; falling edge = stop** – this scheme differs from the “Type A – 1-bit” scheme in that each long press changes direction to which slats are adjusted / shutters are moved, and the falling edge (button release) stops them. The “FB 1: shutter – Stop/step up/down” communication object may assume 4-bit values, and in the “Open: step value” and “Close: step value” fields you can define values by which slats are adjusted / shutters are moved during the step up / down.

In the case of two-channel mode, you can select:

**Type A – 1-bit: button short pressing = step/stop; button long pressing – open/close**

**Type B – 1-bit: button short pressing = open/close; button long pressing = step/stop**

**Type C – 4-bit: button short pressing = step; button long pressing = open/close**

**Type D – 4-bit: button short pressing = open/close; button long pressing = step; falling edge = stop**

Events from the first channel (“Open: control channel”) will start the shutter moving up and trigger stepwise slat adjustment / shutter movement in the same direction. Events from the second channel (“Close: control channel”) will start the shutter moving down and trigger stepwise slat adjustment / shutter movement in that direction. The scheme defines which event will trigger movement and which will adjust the slats / move the shutter by one step up / down, depending on the channel in which it is generated. Falling edge (button release) in the channel will stop movement of the slats / shutter. If you select the “Type C” or “Type D” scheme, the “FB 1: shutter – Stop/step up/down” communication object will be a 4-bit object. The fields will be displayed in the tab in which you can define the step value up (“Open: step value”) and down (“Close: step value”).

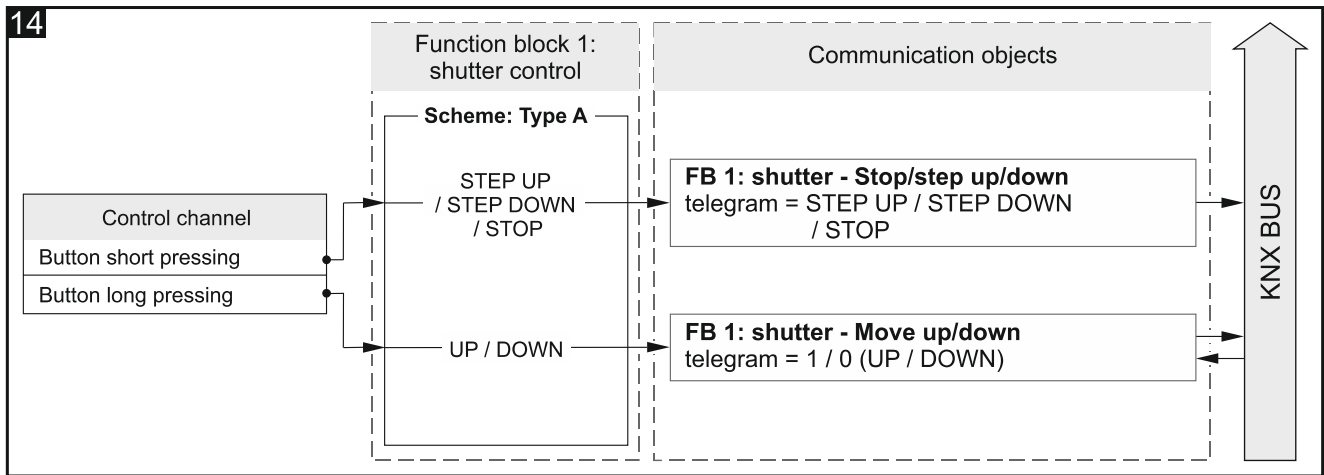


*The two-channel control can only be properly implemented when two different channels are selected in the “Open: control channel” and “Close: control channel” fields.*

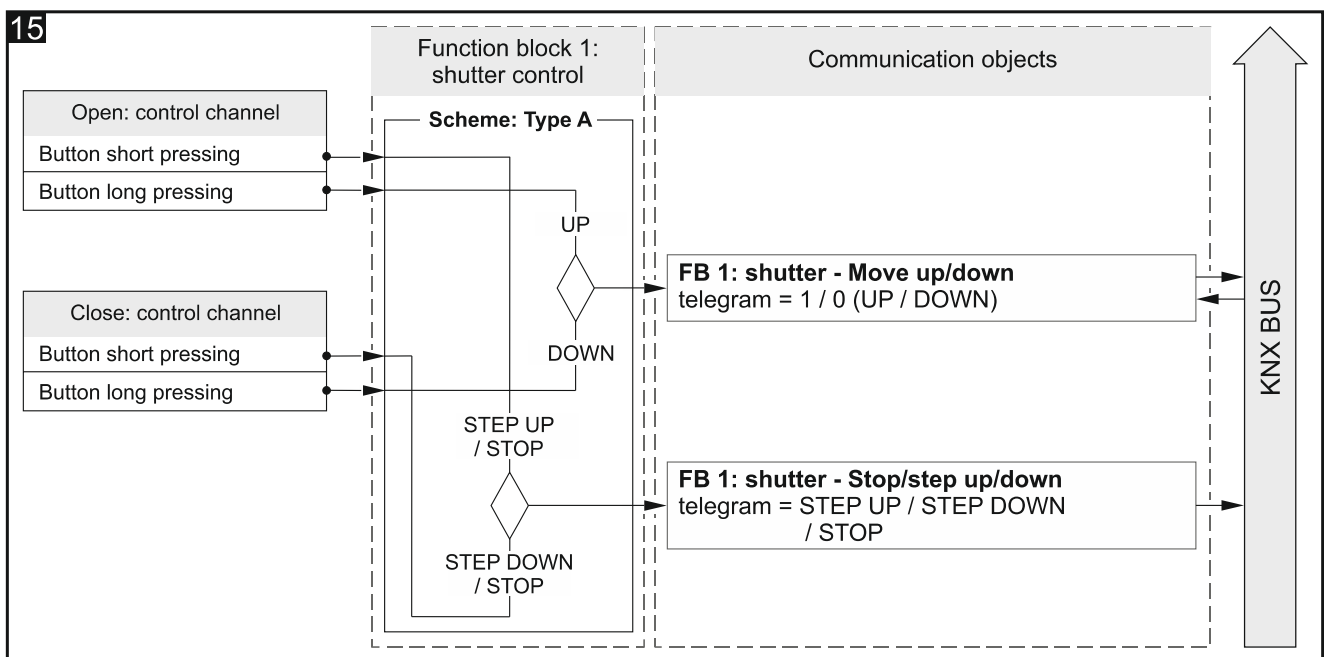
*“Type C – 4-bit” (one-channel control) and “Type D – 4-bit” (two-channel control) are dedicated schemes for shutter movement control by using physical channels. It is recommended for these schemes that the step value for movement up and down be set at 100% (“Open: step value” and “Close: step value” fields). Thus the stepwise movement of slats or shutter will only be stopped after the button is released by the user.*

*Please remember that the binary input module enables the signals from outside of the KNX system to be changed into control telegrams for other devices on the bus (e.g. for roller shutter/blind actuators), but interpretation of these telegrams depends on how these devices/actuators are configured.*

**Fig. 14** shows an example of one-channel roller shutter control (“Type A” control scheme) and **Fig. 15** – an example of two-channel roller shutter control (“Type A” control scheme).



Configuration for example shown in **Fig. 14** in ETS: Two-channel mode = No; Control scheme = Type A.



Configuration for example shown in **Fig. 15** in ETS: Two-channel mode = Yes; Control scheme = Type A.

**Control channel** – events generated in the channel control the shutter movement (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of one-channel control.

**Open: control channel** – events generated in the channel start shutter movement and slat adjustment up (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of two-channel control.

**Close: control channel** – events generated in the channel start shutter movement and slat adjustment down (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of two-channel control.

**i** Which events from the above channels will change the value of “FB 1: shutter - Move up/down” object and which the value of “FB 1: shutter - Stop/step up/down” object is defined by the control scheme selected in the “Control scheme” field.

**Open: step value** – value by which slats will be adjusted / shutter moved up when an event is generated in the control channel (“Control channel” or “Open: control channel”) that changes the state of the “FB 1: shutter - Stop/step up/down” communication object

(100%...1% Up, Stop - see Appendix "4-bit telegrams for shutter control"). The field is displayed when the "Type C" or "Type D" scheme is selected in the "Control scheme" field.

**Close: step value** – value by which slats will be adjusted / shutter moved down when an event is generated in the control channel ("Control channel" or "Close: control channel") that changes the state of the "FB 1: shutter - Stop/step up/down" communication object (100%...1% Down, Stop - see Appendix "4-bit telegrams for shutter control"). The field is displayed when the "Type C" or "Type D" scheme is selected in the "Control scheme" field.



*In the case of louver type of shutter, changing the state of "FB 1: shutter - Stop/step up/down" communication object results in stepwise adjustment of slats, and in the case of roller blind/shade – in stepwise movement of the whole shutter.*

### 4.5.5 Sequence

Used for stepwise switching of the state of 1-bit communication objects according to predefined order, enabling devices or device groups on the KNX bus to be sequentially switched on/off. Within one sequence you can switch over the state of up to 5 communication objects ("FB 1: sequence - Object 1...5"). The switching order is defined by selecting the type of switching sequence in the "Type of sequence" field. You can define in the program ("Two-channel mode" field) whether the switching control in sequence will be implemented by using one or two channels. You can also enable the option to repeat the sequence after completion of the last step. All sequence types available in the program are described below.

The function block can be controlled by short button press events when using one-button control or by falling edge events and channel state using two-button control.



*Restart of the module sets to "0" the value of all "FB 1: sequence - Object 1...5" communication objects enabled in sequence and sends it to the KNX bus.*

### Binary code

In this sequence, each occurrence of the event that activates the switching will change the object state according to the scheme based on binary code. In the table, state changes for sequences using 2, 3, 4 and 5 objects are separated with a thick line.

00000->00001->00010->00011->00100->00101->00110->00111...

Event No.	Sequence step	Communication object value				
		Object 5	Object 4	Object 3	Object 2	Object 1
0	00000	0	0	0	0	0
1	00001	0	0	0	0	1
2	00010	0	0	0	1	0
3	00011	0	0	0	1	1
4	00100	0	0	1	0	0
5	00101	0	0	1	0	1
6	00110	0	0	1	1	0
7	00111	0	0	1	1	1
8	01000	0	1	0	0	0
9	01001	0	1	0	0	1
10	01010	0	1	0	1	0
11	01011	0	1	0	1	1
12	01100	0	1	1	0	0
13	01101	0	1	1	0	1
14	01110	0	1	1	1	0
15	01111	0	1	1	1	1
16	10000	1	0	0	0	0
17	10001	1	0	0	0	1

Event No.	Sequence step	Communication object value				
		Object 5	Object 4	Object 3	Object 2	Object 1
18	10010	1	0	0	1	0
19	10011	1	0	0	1	1
20	10100	1	0	1	0	0
21	10101	1	0	1	0	1
22	10110	1	0	1	1	0
23	10111	1	0	1	1	1
24	11000	1	1	0	0	0
25	11001	1	1	0	0	1
26	11010	1	1	0	1	0
27	11011	1	1	0	1	1
28	11100	1	1	1	0	0
29	11101	1	1	1	0	1
30	11110	1	1	1	1	
31	11111	1	1	1	1	1

### Gray code

The sequence, which is based on the Gray code, makes it possible to go through all achievable combinations of communication object states. In each sequence step, the state of just one object changes, the first and the last sequence steps differing in the state of just one object. In the table, state changes for sequences using 2, 3, 4 and 5 objects are separated with a thick line.

00000->00001->00011->00010->00110->00111->00101->00100...

Event No.	Sequence step	Communication object value				
		Object 5	Object 4	Object 3	Object 2	Object 1
0	00000	0	0	0	0	0
1	00001	0	0	0	0	1
2	00011	0	0	0	1	1
3	00010	0	0	0	1	0
4	00110	0	0	1	1	0
5	00111	0	0	1	1	1
6	00101	0	0	1	0	1
7	00100	0	0	1	0	0
8	01100	0	1	1	0	0
9	01101	0	1	1	0	1
10	01111	0	1	1	1	1
11	01110	0	1	1	1	0
12	01010	0	1	0	1	0
13	01011	0	1	0	1	1
14	01001	0	1	0	0	1
15	01000	0	1	0	0	0
16	11000	1	1	0	0	0
17	11001	1	1	0	0	1
18	11011	1	1	0	1	1
19	11010	1	1	0	1	0
20	11110	1	1	1	1	0
21	11111	1	1	1	1	1
22	11101	1	1	1	0	1



Event No.	Sequence step	Communication object value				
		Object 5	Object 4	Object 3	Object 2	Object 1
23	11100	1	1	1	0	0
24	10100	1	0	1	0	0
25	10101	1	0	1	0	1
26	10111	1	0	1	1	1
27	10110	1	0	1	1	0
28	10010	1	0	0	1	0
29	10011	1	0	0	1	1
30	10001	1	0	0	0	1
31	10000	1	0	0	0	0

**Queue**

In each sequence step (except for the first one), the state of two communication objects is switched over. In one step, the state of the next object changes to “ON” and the state of the object switched on in the previous step changes to “OFF”.

00000->00001->00010->00100->01000->10000

Event No.	Sequence step	Communication object value				
		Object 5	Object 4	Object 3	Object 2	Object 1
0	00000	0	0	0	0	0
1	00001	0	0	0	0	1
2	00010	0	0	0	1	0
3	00100	0	0	1	0	0
4	01000	0	1	0	0	0
5	10000	1	0	0	0	0

**Queue with pause**

In this sequence, the steps in which the state of consecutive objects changes to “ON”, are divided with pauses, i.e. steps in which all objects are switched off.

00000->00001->00000->00010->00000->00100->00000->01000->00000->10000

Event No.	Sequence step	Communication object value				
		Object 5	Object 4	Object 3	Object 2	Object 1
0	00000	0	0	0	0	0
1	00001	0	0	0	0	1
2	00000	0	0	0	0	0
3	00010	0	0	0	1	0
4	00000	0	0	0	0	0
5	00100	0	0	1	0	0
6	00000	0	0	0	0	0
7	01000	0	1	0	0	0
8	00000	0	0	0	0	0
9	10000	1	0	0	0	0

**Queue with fill**

In each sequence step, the state of consecutive object is changed to “ON”, until all objects are switched on.

00000->00001->00011->00111->01111->11111

Event No.	Sequence step	Communication object value				
		Object 5	Object 4	Object 3	Object 2	Object 1
0	00000	0	0	0	0	0
1	00001	0	0	0	0	1
2	00011	0	0	0	1	1
3	00111	0	0	1	1	1
4	01111	0	1	1	1	1
5	11111	1	1	1	1	1

## Description of parameters

Global	Two-channel mode	<input checked="" type="radio"/> no <input type="radio"/> yes
+ Channels	Control channel	Physical Channel A
- Function blocks	Repeat sequence at the end	<input checked="" type="radio"/> off <input type="radio"/> on
General	For sequence description please refer to user manual.	
1: Sequence	Type of sequence	binary code
2: Switch	Number of objects in sequence	2

**Two-channel mode** – defines how to control the switching sequence:

**No** – the two-channel mode disabled i.e. the switching sequence can be controlled by using one channel. Short button press in the control channel will start the sequence (switch the object state according to the selected type of sequence). Long press in the control channel will change the switching direction (next step / previous step) - see **Fig. 16**. Default value.

**Yes** – two-channel mode enabled i.e. the switching sequence can be controlled by using two channels – see **Fig. 17**.

**Control scheme** – in the case of two-channel mode, you can select:

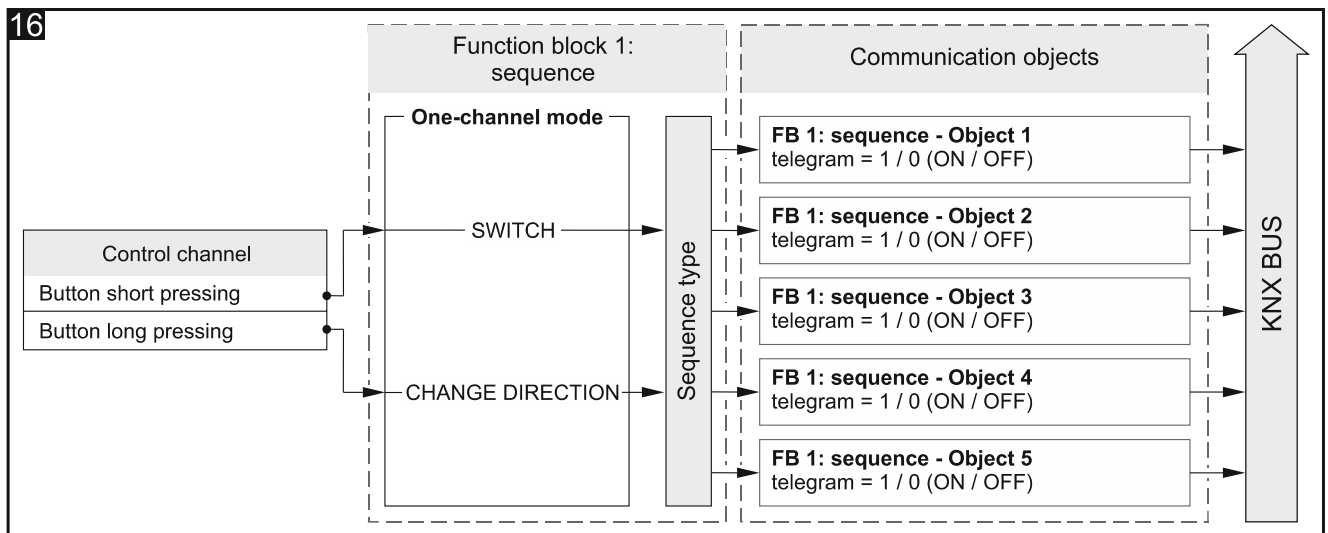
**Type A: button short pressing = change sequence** – short button press in the first channel (“Control channel: previous step”) will trigger next step in the sequence. Short button press in the second channel (“Control channel: next step”) will trigger previous step in the sequence (see **Fig. 17-I**).

**Type B: falling edge = change sequence state = change direction (1 = next / 0 = previous)** – falling edge in the first channel (“Control channel: switching”) will start the sequence (will trigger next / previous step in the sequence). State of the second channel (“Control channel: direction”) defines switching direction in the sequence, where: “1” = NEXT, “0” = PREVIOUS (see **Fig. 17-II**).

Depending on the selected scheme, another pair of fields that allows you to select control channels is displayed in the tab (Type A: “Control channel: previous step” and “Control channel: next step”; Type B: “Control channel: switching” and “Control channel: direction”).



*The two-channel control can only be properly implemented when two different control channels are selected.*



Configuration for example shown in **Fig. 16** in ETS: Two-channel mode = No, Number of objects in sequence = 5.

**Control channel** – events generated in the channel control the switching sequence (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of one-channel control.

**Control channel: previous step** – short button press in the channel will trigger previous step in the sequence (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). See **Fig. 17-I**. The field is available if you enable the two-channel mode and select “Type A” control scheme.

**Control channel: next step** – short button press in the channel will trigger next step in the sequence (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). See **Fig. 17-I**. The field is available if you enable the two-channel mode and select “Type A” control scheme.

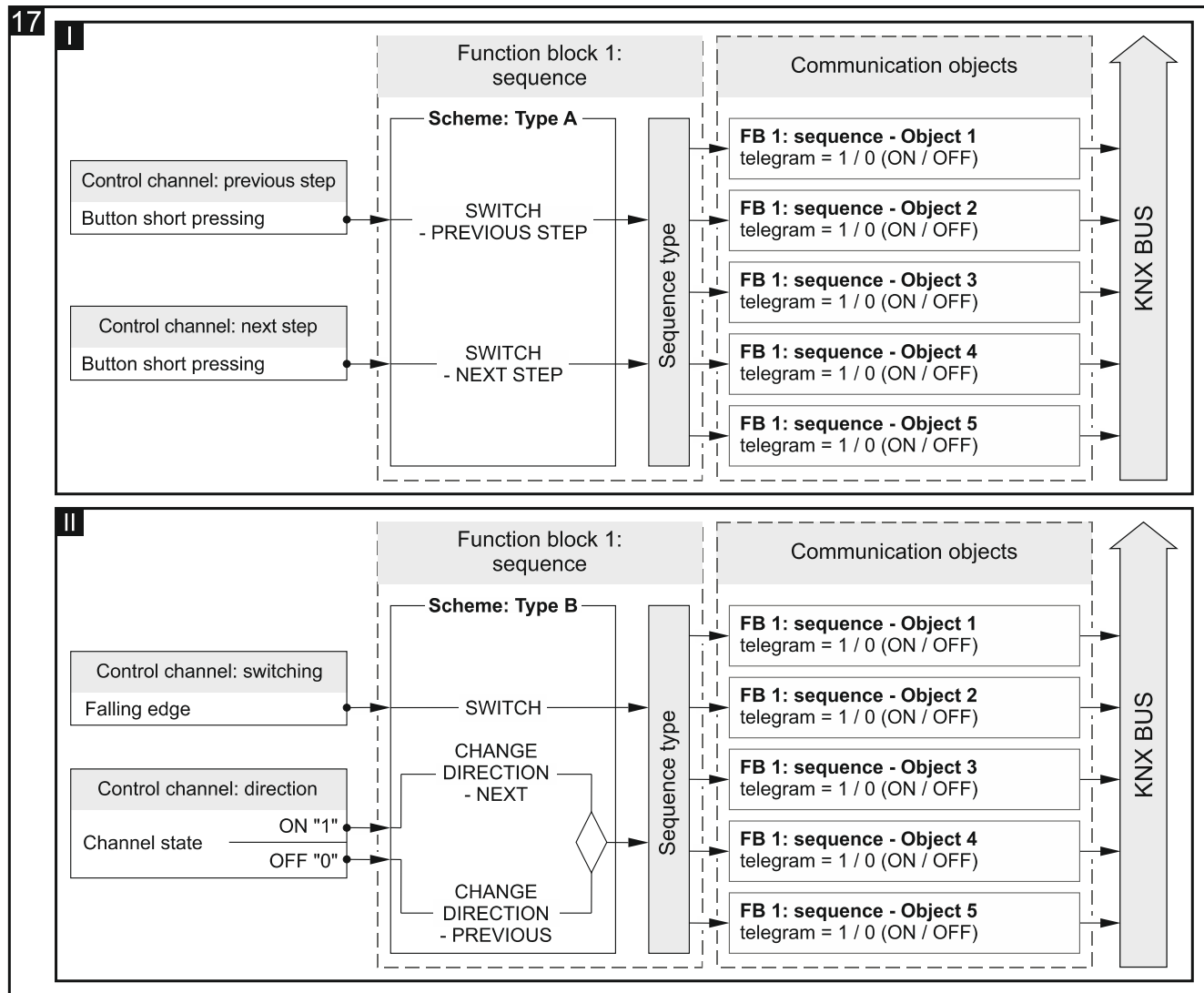
**Control channel: switching** – falling edge in the channel will trigger next / previous step in the sequence (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). See **Fig. 17-II**. The field is available if you enable the two-channel mode and select “Type B” control scheme.

**Control channel: direction** – channel state defines switching direction in the sequence, where: state of channel “1” = NEXT, state of channel “0” = PREVIOUS (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). See **Fig. 17-II**. The field is available if you enable the two-channel mode and select “Type B” control scheme.

**Repeat sequence at the end** – you can select:

**Off** – the option to repeat the sequence after the last step is disabled i.e. after reaching the last step of sequence, only the previous step in the sequence can be triggered. Default value.

**On** – the option to repeat the sequence after the last step is enabled i.e. after the last step of the sequence all objects will be switched off and switching can start from the first step in the sequence.



Configuration for example shown in **Fig. 17-I** in ETS: Two-channel mode = Yes; Control scheme = Type A; Number of objects in sequence = 5. Configuration for example shown in **Fig. 17-II** in ETS: Two-channel mode = Yes; Control scheme = Type B; Number of objects in sequence = 5.

**Type of sequence** – you can select the type of sequence (description – see “Sequence”):

binary code

Gray code

queue

queue with pause

queue with fill

**Number of objects in sequence** – number of objects whose state can be switched in sequence (minimum = 2; maximum = 5). Selecting a value higher than 2 will enable the “FB 1: sequence – Object 3...5” communication objects.



*Restart of the module will set to “0” the value of all “FB 1: sequence - Object 1...5” communication objects.*

#### 4.5.6 Counter

It counts events both from the physical channels and from other channels of the module. You can define in the program (“Two-channel mode” parameter) whether the counter control will be implemented by using one or two channels. Their current value of the counter is saved to the “FB 1: counter – Current value” communication object. You can define in the program the

initial as well as the minimum and maximum counter value. The “FB 1: counter – Indicates overflow” communication object sends to the KNX bus a telegram with information, when the counter has reached one of the limit values (0 = counter reached minimum value, 1 = counter reached maximal value). You can use the “FB 1: counter – Set value” communication object to force counter value by sending a telegram from the bus.

The function block can be controlled by events of short / long button press (one-channel control) or any pair of events defined in the program (two-channel control).

## Description of parameters

Global	Two-channel mode	<input checked="" type="radio"/> no <input type="radio"/> yes
+ Channels	Control channel	Physical Channel A
- Function blocks	Reset counter at the end	<input checked="" type="radio"/> off <input type="radio"/> on
General	Initial value of counter	100
1: Counter	Minimum value of counter	0
2: Switch	Maximum value of counter	255

**Two-channel mode** – defines how to control the counter:

**No** – two-channel mode disabled i.e. the counter can be controlled by using one channel.

Short button press in the control channel will increase (+1) / decrease (-1) the counter (depending on the current direction of counting). Long press in the control channel will change the direction of counting - see **Fig. 18-I**. Default value.

**Yes** – two-channel mode enabled i.e. the counter can be controlled by using two channels.

Event in the first channel (“Counter incrementing channel (+1)”) will increase the counter value. Event in the second channel (“Counter decrementing channel (-1)”) will decrease the counter value. The “Counter incrementing event (+1)” and “Counter decrementing event (-1)” fields allow you to select the control events. **Fig. 18-II** shows an example in which the short button press is the event that increases/decreases the counter.

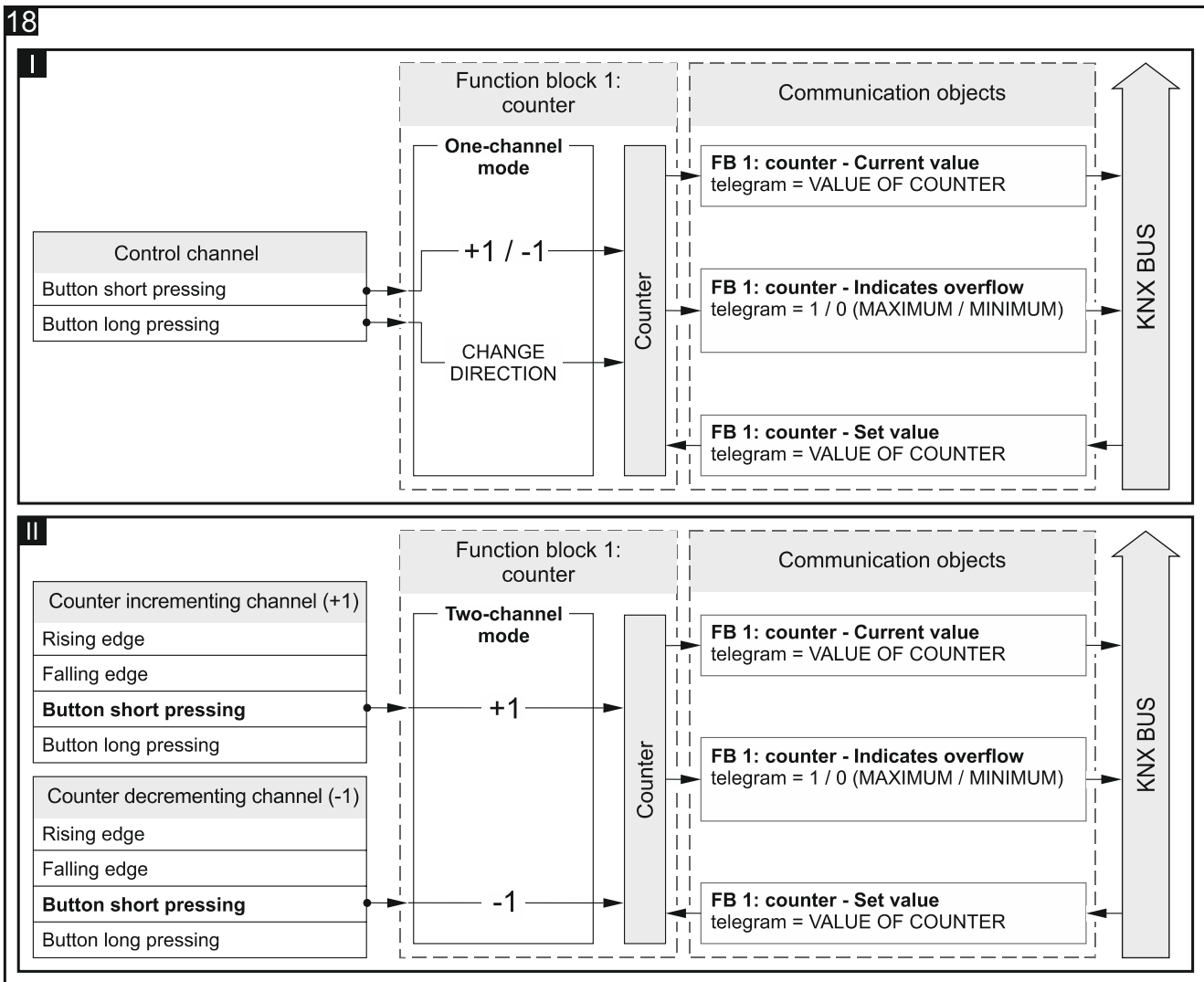
**Control channel** – the counter is controlled by events generated in the channel (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of one-channel control.

**Counter incrementing channel (+1)** – events generated in the channel increase the counter value (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of two-channel control.

**Counter decrementing channel (-1)** – events generated in the channel decrease the counter value (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4). The field is available in the case of two-channel control.

**Counter incrementing event (+1)** – event whose occurrence in the “Counter incrementing channel (+1)” channel will increase the value of the “FB 1: counter - Current value” object by one (button short pressing / button long pressing / rising edge / falling edge). The field is available in the case of two-channel control.

**Counter decrementing event (-1)** – event whose occurrence in the “Counter decrementing channel (-1)” channel will decrease the value of the “FB 1: counter - Current value” object by one (button short pressing / button long pressing / rising edge / falling edge). The field is available in the case of two-channel control.



Configuration for example shown in **Fig. 18-I** in ETS: Two-channel mode = No. Configuration for example shown in **Fig. 18-II** in ETS: Two-channel mode = Yes; Counter incrementing event (+1) = button short pressing; Counter decrementing event (-1) = button short pressing.

**Reset counter at the end** – you can select:

**Off** – counter reset at the end of counting is disabled i.e. after reaching the upper or lower limit value (“Maximum value of counter” and “Minimum value of counter” fields), the counter will be stopped. Default value.

**On** – counter reset at the end of counting is enabled i.e. reaching the upper limit value will set in the counter the value from the “Minimum value of counter” field, and reaching the lower limit value will set in the counter the value from the “Maximum value of counter” field and the counting will restart.

**Initial value of counter** – initial value of the “FB 1: counter – Current value” communication object (0 – 255, default = 100).

**Minimum value of counter** – minimum value of the “FB 1: counter – Current value” communication object (0 - 255, default = 0).

**Maximum value of counter** – maximum value of the “FB 1: counter – Current value” communication object (0 – 255, default = 255).



*If the value entered in the “Minimum value of counter” field will be higher than the value in the “Maximum value of counter” field, the meaning of the fields will be reversed to ensure correct functioning of the counter.*

### 4.5.7 Scene

Triggers one of the 20 scenes to which the module can be linked. The scene makes it possible to switch the state of several communication objects (scene participants) when using one group address. Communication with all participants linked in the scene takes place via the same group address. Thus all that is needed to trigger a scene is one telegram with the number of the scene to be triggered (“FB 1: scene – Scene trigger” communication object).

The function block is controlled by short button press events.



*Using the scenes eases the telegram traffic and relieves the bus.*

#### Description of parameters

Global	Control channel	Physical Channel A
+ Channels	Scene number	1
- Function blocks		
General		
1: Scene		

**Control channel** – short button press event generated in the channel will trigger the scene having the number defined in the “Scene number” field (Physical channel 1...8 / Virtual channel 1...8 / Logic channel 1...4 / Timer channel 1...4).

**Scene number** – number of the scene that will be triggered when a short button press event, is generated in the channel selected in the “Control channel” field (1 - 64).

## 4.6 Communication objects

### 4.6.1 Virtual inputs

For the “Virtual input 1...8” communication objects, the numbers from 1 to 8 are reserved in the program.

	Number ^	Name	Object Function	Length	C	R	W	T	U
■ ↕	1	Virtual input A	Input A	1 bit	C	-	W	-	-
■ ↕	2	Virtual input B	Input B	1 bit	C	-	W	-	-
■ ↕	3	Virtual input C	Input C	1 bit	C	-	W	-	-
■ ↕	4	Virtual input D	Input D	1 bit	C	-	W	-	-
■ ↕	5	Virtual input E	Input E	1 bit	C	-	W	-	-
■ ↕	6	Virtual input F	Input F	1 bit	C	-	W	-	-
■ ↕	7	Virtual input G	Input G	1 bit	C	-	W	-	-
■ ↕	8	Virtual input H	Input H	1 bit	C	-	W	-	-

Communication objects of the virtual channel				
No.	Object name	Object function	Data type	Flags
1... 8	Virtual input A...H	Input A...H	1 bit, DPT 1.001	C, W
<p>You can enable objects in the “Global” tab by selecting the number of objects to be enabled in the “Number of virtual inputs” field. The objects can receive 1-bit telegrams sent by other devices from the KNX bus. You can enable up to 8 “Virtual input A...H” communication objects in the module. Each object can be a virtual input of any virtual channel. To link the object to the channel, you must select this object in the tab for defining the channel operating parameters, in the “Channel virtual input” field. Changing the object value will generate in the channel events handled by the module.</p> <p>See “Virtual channels”</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

### 4.6.2 Communication objects of function blocks

For each function block (number 1 to 20) activated in the module, up to 5 communication objects can be enabled (e.g. the “Sequence” function block). 100 numbers from 10 to 109 are reserved in the program for all communication objects that can be enabled for individual function blocks. 5 consecutive numbers are reserved for these objects in each block.

**i** *It is assumed that all the objects discussed below have been enabled for the function block number 1 that takes up numbers from 10 to 14.*

#### Switch / Value transmitter

	Number ^	Name	Object Function	Length	C	R	W	T	U
■ ↕	10	FB 1: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-
■ ↕	11	FB 1: switch	Long pressing (1-bit)	1 bit	C	R	W	T	-

Communication objects: Switch / Value transmitter				
No.	Object name	Object function	Data type	Flags
10	FB 1: switch	Short pressing (1-bit) / (1-byte) / (percentage) / (dimming control) / (shutter control)	1 bit, DPT 1.001  1 byte DPT 5.* 8-bit unsigned value  4 bit, DPT 3.007  4 bit, DPT 3.008	C, R, W, T
<p><b>i</b> <i>A starting configuration has been prepared for the module in the ETS program application, in which 8 function blocks are defined to execute the switch function. The “FB 1...8: switch – Short pressing (1-bit)” communication objects that allow implementation of these function blocks are enabled always the application is imported into the ETS program.</i></p>				



Communication objects: Switch / Value transmitter										
No.	Object name		Object function	Data type				Flags		
	Number *	Name	Object Function	Length	C	R	W	T	U	
	10	FB 1: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
	15	FB 2: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
	20	FB 3: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
	25	FB 4: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
	30	FB 5: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
	35	FB 6: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
	40	FB 7: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
	45	FB 8: switch	Short pressing (1-bit)	1 bit	C	R	W	T	-	
<p>During configuration of the module, the object can be enabled in the “General” tab (that can be seen after opening the “Function blocks” tab), when the “Switch” function is selected in the “Function block 1” field. The object value can be changed by a short button press event or a telegram from the KNX bus. The possibility to change object value by a telegram from the bus enables the object value to be updated each time when switching is triggered by another communication object from the same address group. If you do not want the object value to be changed by telegrams from the bus, disable the “W – Write” option in the object properties. The data type and value that can be assumed by the object are defined in the “1: Switch” tab:</p> <ul style="list-style-type: none"> <li>• 1-bit value [0/1] – DPT 1.001 (on / off).</li> <li>• 1-byte value [0...255] – DPT 5.* (set value).</li> <li>• 1-byte value [0...100%] – DPT 5.001 (set percentage value).</li> <li>• 4-bit value [1...100% Darker; Stop; 1...100% Brighter] – DPT 3.007 (set brightness value).</li> <li>• 4-bit value [1...100% Up; Stop; 1...100% Down] – DPT 3.008 (set shutter value).</li> </ul> <p>The object can send telegrams cyclically to the KNX bus with information on the current value. Entering value “0” in the “Interval of value cyclic sending (short pressing)” field will disable the cyclic sending.</p> <p>See “Switch”</p>										
11	FB 1: switch		Long pressing (1-bit) / (1-byte) / (percentage) / (dimming control) / (shutter control)	1 bit, DPT 1.001 1 byte DPT 5.* 8-bit unsigned value 4 bit, DPT 3.007 4 bit, DPT 3.008	C, R, W, T					
<p>The object will be enabled, if you select “Yes” for the “Event on button long pressing” option in the “1: Switch” tab. The object value can be changed by a long button press event or a telegram from the KNX bus. The possibility to change object value by a telegram from the bus enables the object value to be updated each time when switching is triggered by another communication object from the same address group. If you do not want the object value to be changed by telegrams from the bus, disable the “W – Write” option in the object properties. The data type and value that can be assumed by the object can be defined in</p>										

Communication objects: Switch / Value transmitter				
No.	Object name	Object function	Data type	Flags
the tab:				
<ul style="list-style-type: none"> <li>• 1-bit value [0/1] – DPT 1.001 (on / off).</li> <li>• 1-byte value [0...255] – DPT 5.* (set value).</li> <li>• 1-byte value [0...100%] – DPT 5.001 (set percentage value).</li> <li>• 4-bit value [1...100% Darker; Stop; 1...100% Brighter] – DPT 3.007 (set brightness value).</li> <li>• 4-bit value [1...100% Up; Stop; 1...100% Down] – DPT 3.008 (set shutter value).</li> </ul>				
<p>The object can send telegrams cyclically to the KNX bus with information on the current value. Entering value “0” in the “Interval of value cyclic sending (long pressing)” field will disable the cyclic sending.</p> <p>See “Switch”</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

**Edge**

Number ^	Name	Object Function	Length	C	R	W	T	U
10	FB 1: edge	Rising/falling (1-bit)	1 bit	C	R	W	T	-

Communication objects: Edge				
No.	Object name	Object function	Data type	Flags
10	FB 1: edge	Rising / falling (1-bit) / (1-byte) / (percentage) / (dimming control) / (shutter control)	1 bit, DPT 1.001 1 byte, DPT 5.* 8-bit unsigned value 4 bit, DPT 3.007 4 bit, DPT 3.008	C, R, T
<p>The object will be enabled, if you select the “Edge” function in the “General” tab (which can be seen after opening the “Function blocks” tab), “Function block 1” field. The object value can be changed by a rising / falling edge event. The data type and value that can be assumed by the object in response to occurrence of each event are defined in the “1: Edge” tab:</p> <ul style="list-style-type: none"> <li>• 1-bit value [0/1] – DPT 1.001 (On / Off).</li> <li>• 1-byte value [0...255] – DPT 5.* (set value).</li> <li>• 1-byte value [0...100%] – DPT 5.001 (set percentage value).</li> <li>• 4-bit value [1...100% Darker; Stop; 1...100% Brighter] – DPT 3.007 (set brightness value).</li> <li>• 4-bit value [1...100% Up; Stop; 1...100% Down] – DPT 3.008 (set shutter value).</li> </ul> <p>See “Edge”</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

## Dimmer

	Number ^	Name	Object Function	Length	C	R	W	T	U
■ ↕	10	FB 1: dimmer	Switch	1 bit	C	R	W	T	-
■ ↕	11	FB 1: dimmer	Dim relative	4 bit	C	R	-	T	-

Communication objects: Dimmer				
No.	Object name	Object function	Data type	Flags
<b>10</b>	<b>FB 1: dimmer</b>	<b>Switch</b>	<b>1 bit, DPT 1.001</b>	<b>C, R, W, T</b>
<p>The object will be enabled, if you select the “Dimmer control” function in the “General” tab (which can be seen after opening the “Function blocks” tab), “Function block 1” field. The communication object allows you to turn the light on / off. Depending on which mode and control scheme is selected in the “1: Dimmer” tab, the object value change can be triggered by a short / long button press. The object value can also be changed by sending a telegram from the KNX bus. This makes it possible to update the object value each time when turning the light on / off is triggered by another communication object from the same address group (e.g. another button in the staircase). If you do not want the object value to be changed by telegrams from the bus, disable the “W – Write” option in the object properties.</p> <p>Telegram: 1 = On; 0 = Off See “Dimmer”</p>				
<b>11</b>	<b>FB 1: dimmer</b>	<b>Dim relative</b>	<b>4 bit, DPT 3.007</b>	<b>C, R, T</b>
<p>The object will be enabled, if you select the “Dimmer control” function in the “General” tab (which can be seen after opening the “Function blocks” tab), “Function block 1” field. The object makes it possible to send telegrams of the “BRIGHTER”, “DARKER”, “STOP” type to the bus (see Appendix “4-bit telegrams for dimming DOWN/UP”). The value of “BRIGHTER” or “DARKER” telegram can be defined in the “1: Dimmer” tab, “Dimming up: step value” and “Dimming down: step value” fields.</p> <p>Depending on which mode and control scheme are selected in the “1: Dimmer” tab, the object value change can be triggered by a short / long button press.</p> <p>See “Dimmer”</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

## Shutter

	Number ^	Name	Object Function	Length	C	R	W	T	U
■ ↕	10	FB 1: shutter	Move up/down	1 bit	C	R	W	T	-
■ ↕	11	FB 1: shutter	Stop/step up/down (1-bit)	1 bit	C	R	-	T	-

Communication objects: Shutter				
No.	Object name	Object function	Data type	Flags
10	FB 1: shutter	Move up/down	1 bit, DPT 1.008	C, R, W, T
<p>The object will be enabled, if you select the “Shutter control” function in the “General” tab (that can be seen after opening the “Function blocks” tab), “Function block 1” field. The object enables the shutter to be moved in the Up / Down direction. The shutter movement started by the object can be stopped by a “STOP” telegram or after the shutter reaches its end point. Depending on which mode and control scheme are selected in the “1: Shutter” tab, the object value can be changed by a short / long button press event. The object value can also be changed by a telegram from the KNX bus. This enables the object value to be updated each time when the shutter movement is triggered by another communication object from the same address group (another button). If you do not want the object value to be changed by telegrams from the bus, disable the “W – Write” option in the object properties.</p> <p>Telegram:                      0 = Up,                      1 = Down.                      See “Shutter”</p>				
11	FB 1: shutter	Stop/step up/down (1-bit) / (4-bit)	1 bit, DPT 1.007 4 bit, DPT 3.008	C, R, T
<p>The object will be enabled, if you select the “Shutter control” function in the “General” tab (that can be seen after opening the “Function blocks” tab), “Function block 1” field. If the object is defined as a 4-bit one (the “4-bit” type scheme is selected in the “Control scheme” field), it will send telegrams of “STEP UP”, “STEP DOWN” and “STOP” type to the bus (see Appendix “4-bit telegrams for shutter control”). In such a case, the step value can be defined in the “Open: step value” and “Close: step value” fields.</p> <p>Telegrams for 1-bit object:                      0 = step Up,                      1 = step Down,                      0 and 1 = stop, when the shutter or slats are moving.</p> <p>Depending on which mode and control scheme are selected in the “1: Shutter” tab, the object value can be changed by short / long button press or falling edge events.                      See “Shutter”</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

**Sequence**

	Number ^	Name	Object Function	Length	C	R	W	T	U
■ ↕	10	FB 1: sequence	Object 1	1 bit	C	R	-	T	-
■ ↕	11	FB 1: sequence	Object 2	1 bit	C	R	-	T	-
■ ↕	12	FB 1: sequence	Object 3	1 bit	C	R	-	T	-
■ ↕	13	FB 1: sequence	Object 4	1 bit	C	R	-	T	-
■ ↕	14	FB 1: sequence	Object 5	1 bit	C	R	-	T	-

Communication objects: Sequence				
No.	Object name	Object function	Data type	Flags
10 ...1 4	FB 1: sequence	Object 1...5	1 bit, DPT 1.001	C, R, T
<p>Objects number 1 and 2 will be enabled, if the “Sequence” function is selected in the “General” tab (that can be seen after opening the “Function blocks” tab), “Function block 1” field. Objects with numbers from 3 to 5 can be enabled in the “1: Sequence” tab by selecting the number of objects to be used in the sequence in the “Number of objects in sequence” field. Depending on which mode and control scheme is selected in the tab, the value of objects can be changed by falling edge or short button press events. Values of individual objects change in accordance with the switching sequence selected in the “Type of sequence” field.</p> <p>Telegram:                      1 = On                      0 = Off                      See “Sequence”</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

**Counter**

	Number ^	Name	Object Function	Length	C	R	W	T	U
■ ↕	10	FB 1: counter	Current value	1 byte	C	R	-	T	-
■ ↕	11	FB 1: counter	Set value	1 byte	C	-	W	T	-
■ ↕	12	FB 1: counter	Indicates overflow	1 bit	C	R	-	T	-

Communication objects: Counter				
No.	Object name	Object function	Data type	Flags
10	FB 1: counter	Current value	1 byte DPT 5.* 8-bit unsigned value	C, R, T
<p>The object will be enabled, if the “Counter” function is selected in the “General” tab (that can be seen after opening the “Function blocks” tab), “Function block 1” field. The object makes it possible to read the current value of the counter.</p> <p>See “Counter”</p>				
11	FB 1: counter	Set value	1 byte DPT 5.* 8-bit unsigned value	C, W, T
<p>The object will be enabled, if the “Counter” function is selected in the “General” tab (that can be seen after opening the “Function blocks” tab), “Function block 1” field. The object enables the value counter settings to be changed by a telegram from the KNX bus. If the value to be set by the telegram is lower than the minimum value defined for the counter, the minimum value will be set in the counter. If however the value to be set is higher than the maximum value that can be set in the counter, the maximum counter value will be set. The minimum and maximum counter values can be defined in the “1: Counter” tab.</p>				

Communication objects: Counter				
No.	Object name	Object function	Data type	Flags
See "Counter"				
12	FB 1: counter	Indicates overflow	1 bit, DPT 1.001	C, R, T
<p>The object will be enabled, if the "Counter" function is selected in the "General" tab (that can be seen after opening the "Function blocks" tab), "Function block 1" field. The object allows information to be sent when the counter reaches the minimum and maximum values defined for the counter. The minimum and maximum counter values can be defined in the "1: Counter" tab.</p> <p>Telegrams:                      1 = counter reached maximum value                      0 = counter reached minimum value                      See "Counter"</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

**Scene**

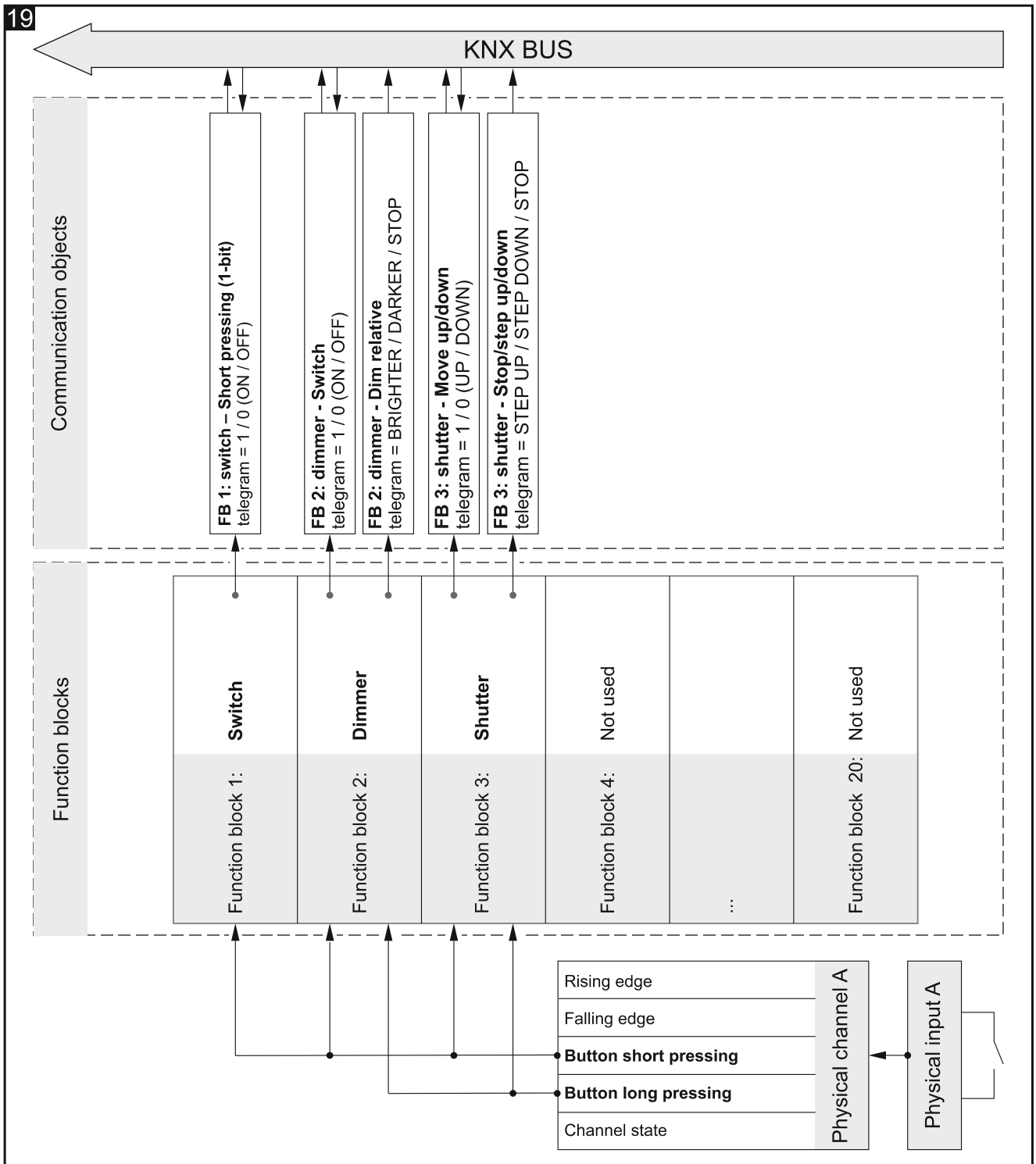
Number ^	Name	Object Function	Length	C	R	W	T	U
10	FB 1: scene	Scene trigger	1 byte	C	R	-	T	-

Communication objects: Scene				
No.	Object name	Object function	Data type	Flags
10	FB 1: scene	Scene trigger	1 byte DPT 17.001 DPT 18.001	C, R, T
<p>The object will be enabled, if the "Scene" function is selected in the "General" tab (that can be seen after opening the "Function blocks" tab), "Function block 1" field. The object enables sending of a 1-byte telegram that can trigger a scene in response to short button press events generated in the channel. The number of scene to be triggered (1-64), you can define in the "1: Scene" tab.</p> <p>See "Scene"</p>				

Flag: C – communication, R – read, W – write, T – transmit, U – update.

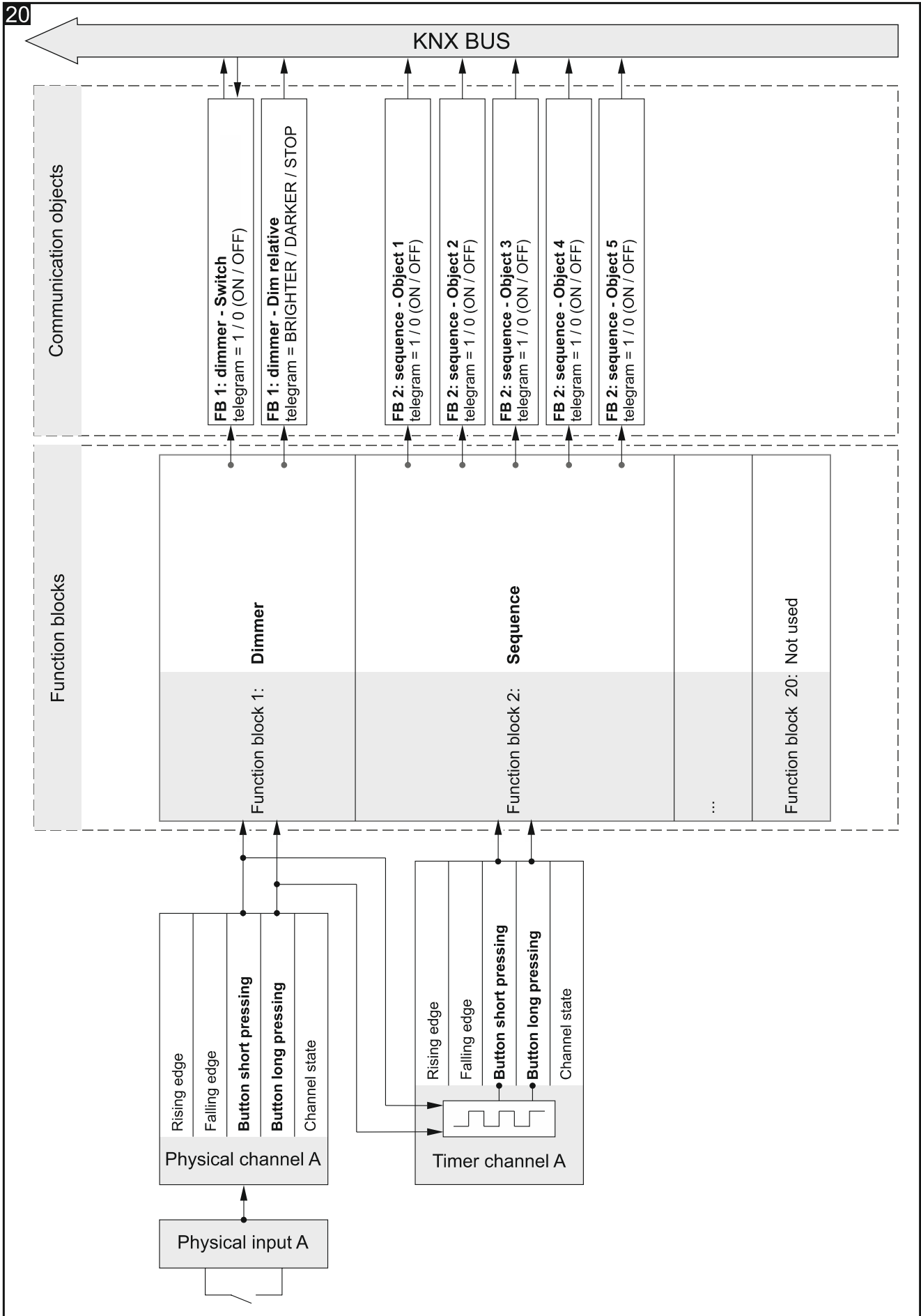
## 4.7 Configuration examples

### 4.7.1 One-channel control of several function blocks



The example shows a solution, where short and long button press events from the physical channel A are used to control three function blocks at the same time. The function block that enables switching (Function block 1: Switch) is controlled by short presses, and the function blocks that control light (Function block 2: Dimmer) and shutter movement (Function block 3: Shutter) – by short and long button presses. Which events change the state of communication objects that control lighting and shutter depends on the mode and control scheme selected when configuring individual function blocks.

### 4.7.2 Using timer channel



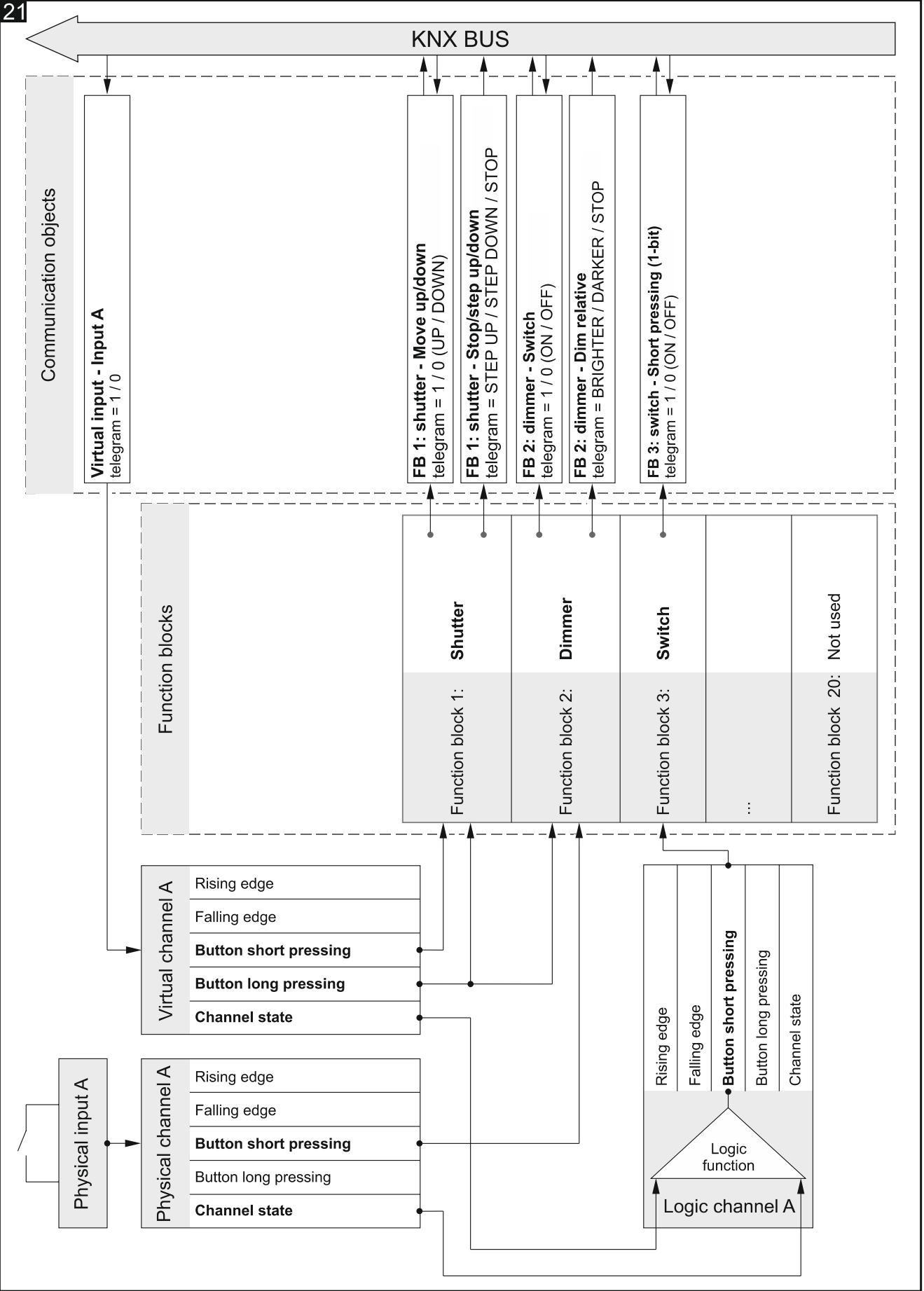


The example shows a configuration in which the short and long button press events from the physical channel A control the lighting (which events change the state of individual communication objects depends on the mode and control scheme selected when configuring the function block 1). At the same time, these events are the pair of events to control the timer channel A, where the short button press enables and the long button press disables this channel. When enabled, the timer channel is changing its state cyclically within the defined time intervals (see “Timer channels”). Thus, the channel generates events that control the switching sequence (Function block 2: Sequence), where short presses trigger consecutive steps in the sequence (switch the state of consecutive objects) and long presses change switching direction in the sequence.

### **4.7.3 Using virtual and logic channels**

The example shows a solution in which telegrams received from the KNX bus by the “Virtual input – Input A” communication object are converted in the virtual channel A into events that are handled by the module. This enables these telegrams to be linked to signals from outside of the bus. Short and long button press events from the virtual channel A control the shutter (Function block 1: Shutter). At the same time, long button press events from this, together with short button press events from the physical channel A control lighting (Function block 2: Dimmer). Which events change the state of individual communication objects depends on the mode and control scheme selected when configuring individual function blocks. The example shows also how the state of virtual channel A and the state of physical channel A are linked to each other in the logic function defined in the logic channel A. As a result of this link, short button press events that control switching (Function block 3: Switch) are generated in the logic channel.

21



## 5. Specifications

---

### Power supply

Supply voltage (KNX bus).....	20...30 VDC
Current consumption from KNX bus .....	<15 mA

### Inputs

Number of inputs .....	8
Input current $I_n$ .....	1.5 mA
Permitted voltage range $U_n$ .....	0...30 VAC/DC
Signal level for $U_{n0}$ signal.....	0...4 VAC/DC
Signal level for $U_{n1}$ signal.....	9...30 VAC/DC

### Input current

Maximum wire cross-section.....	2.5 mm <sup>2</sup>
Maximum tightening torque.....	0.5 Nm

### KNX parameters

Maximum time of reaction to telegram .....	<20 ms
Maximum number of communication objects.....	108
Maximum number of group addresses.....	256
Maximum number of associations .....	256

### Other parameters

Operating temperature range.....	0°C...+45°C
Storage/transport temperature range.....	-25°C...+70°C
IP code.....	IP20
Number of units on DIN rail.....	4
Enclosure dimensions.....	70 x 92 x 60 mm
Weight.....	144 g



**Exceeding the limit values of the module working parameters may damage the module and pose hazard to human health or life.**

## 6. Appendix

### 6.1 4-bit telegrams for dimming DOWN/UP

Values of BRIGHTER / DARKER telegrams				
fractional	percentage	binary	hexadecimal	decimal
STOP	STOP	0000	0	0
1/1 DARKER	100% DARKER	0001	1	1
1/2 DARKER	50% DARKER	0010	2	2
1/4 DARKER	25% DARKER	0011	3	3
1/8 DARKER	12% DARKER	0100	4	4
1/16 DARKER	6% DARKER	0101	5	5
1/32 DARKER	3% DARKER	0110	6	6
1/64 DARKER	1% DARKER	0111	7	7
STOP	STOP	1000	8	8
1/1 BRIGHTER	100% BRIGHTER	1001	9	9
1/2 BRIGHTER	50% BRIGHTER	1010	A	10
1/4 BRIGHTER	25% BRIGHTER	1011	B	11
1/8 BRIGHTER	12% BRIGHTER	1100	C	12
1/16 BRIGHTER	6% BRIGHTER	1101	D	13
1/32 BRIGHTER	3% BRIGHTER	1110	E	14
1/64 BRIGHTER	1% BRIGHTER	1110	F	15

### 6.2 4-bit telegrams for shutter control

Values of UP / DOWN telegrams				
fractional	percentage	binary	hexadecimal	decimal
STOP	STOP	0000	0	0
1/1 DOWN	100% DOWN	0001	1	1
1/2 DOWN	50% DOWN	0010	2	2
1/4 DOWN	25% DOWN	0011	3	3
1/8 DOWN	12% DOWN	0100	4	4
1/16 DOWN	6% DOWN	0101	5	5
1/32 DOWN	3% DOWN	0110	6	6
1/64 DOWN	1% DOWN	0111	7	7
STOP	STOP	1000	8	8
1/1 UP	100% UP	1001	9	9
1/2 UP	50% UP	1010	A	10
1/4 UP	25% UP	1011	B	11
1/8 UP	12% UP	1100	C	12
1/16 UP	6% UP	1101	D	13
1/32 UP	3% UP	1110	E	14
1/64 UP	1% UP	1110	F	15