CHORUS

GEWISS

KNX 6-channel push-button panel with 1-channel actuator

GW1x784A

Technical Manual

Contents

1 Introduction

This manual describes the functions for the device “**KNX 6-channel push-button panel with 1-channel actuator**” (GW10784A, GW12784A, GW14784A) and how they are set and configured using the ETS configuration software.

2 Application

The flush-mounted KNX 6-channel push-button panel is a command device with 6 channels, which can be used individually or combined and with an actuator with ON/OFF functions.

Each channel has two LEDs for light signalling, one amber and one green.
The buttons can perform the following functions:

- load ON / OFF commands
- timed commands
- dimmer management (single or double button)
- curtain / shutter management (single or double button)
- scene management

The actuator 1 channel can perform the following functions:

- load ON / OFF
- timed activation (stairs light)
- blinking activation
- activation through a priority command (forcing)
- scenes
- boolean logic
- safety
- block

A function can be associated to each channel by means of a specific parameter, as described below.

2.1 Association limits

The maximum number of communication objects available is 156.
The maximum number of associations that the device can store is 212.
The maximum number of group addresses is 212.

3 *“6-channel push-button panel”* menu

The **6-channel push-button panel** menu contains the application parameters for all the input channels implemented by the device.
The main operator parameters for the device (fig. 3.1).

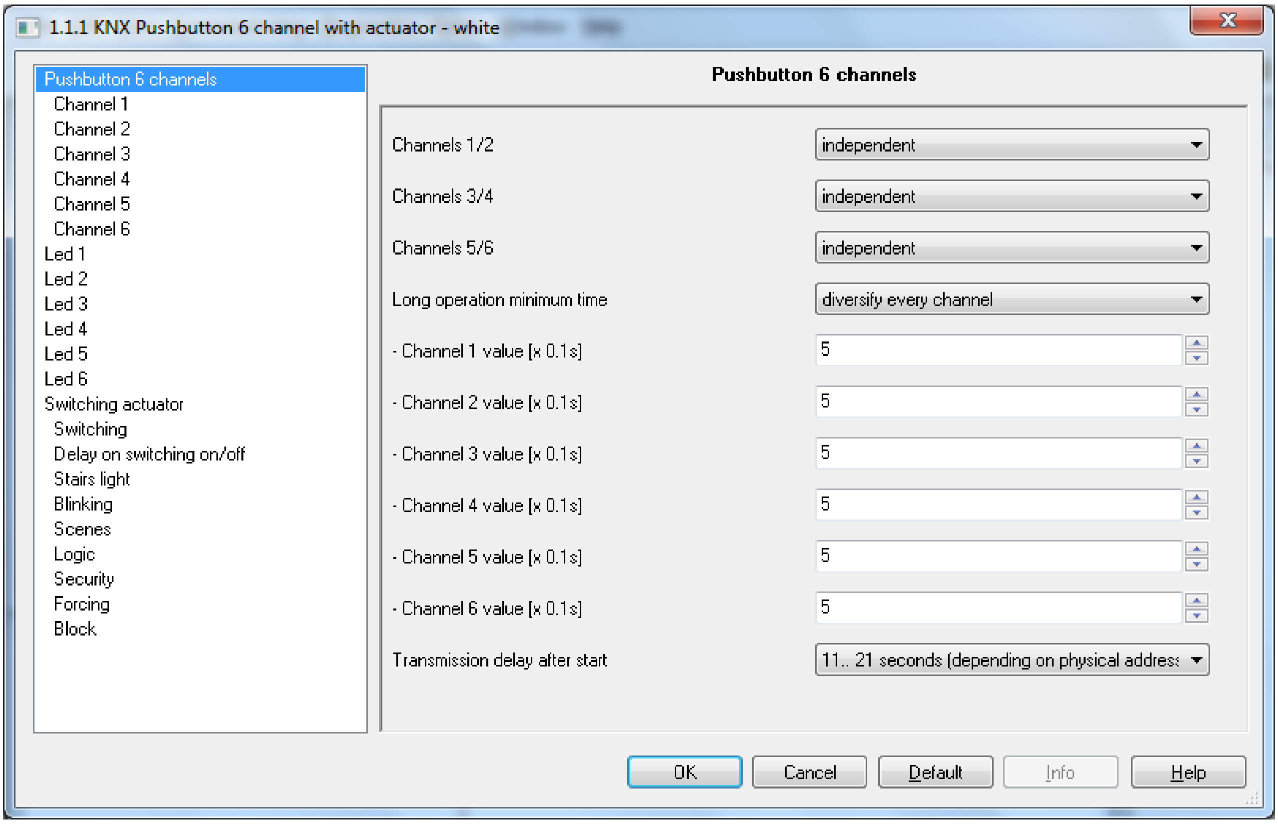


Fig. 3.1

3.1 Parameters

➢ *3.1.1. Channels X/Y*

Using these parameters, it is possible to select for each of the 6 input channels implemented by the push-button panel if they should carry out an independent function or if it should be combined two at a time to contribute towards carrying out a shared function. The values that can be set are:

- **independent (default value)**
- coupled

The database structure will vary depending on the values set for the above parameters: an independent setting menu for each channel if the set value is independent and a common menu if a combined value is set.

➢ *3.1.2 Long operation minimum time*

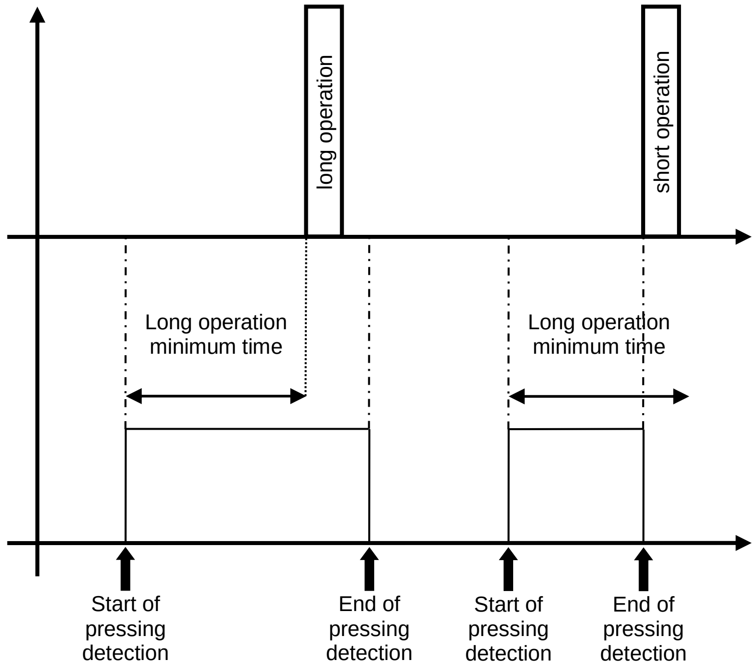
Many functions that the independent or combined channels can carry out foresee the differentiation between a short operation and a long operation. This parameter can be used to define a single time value for all channels or a different one for each of the channels; The values that can be set are:

- **the same for all channels (default value)**
- diversify every channel

The following parameters are displayed depending on the selected value: **“Value [x 0.1s]”** (if an identical value is set for all channels) or **“Channel 1 value [x 0.1s]”, “Channel 2 value [x 0.1s]”, “Channel 3 value [x 0.1s]”, “Channel 4 value [x 0.1s]”, “Channel 5 value [x 0.1s]” and “Channel 6 value [x 0.1s]”** (if a diversity value is set for each channel), which determine the minimum effective time during which the device must detect the pressing of the button to differentiate between the short operation and the long operation associated with the channel.
The possible values are:

- from 3 to 150 with step 1, **5 (default value)**

The following example shows the meaning of the above parameters



➢ *3.1.3 Transmission delay after start*

This parameter is used to define the time that must pass after which the device may transmit the telegrams on the bus following a drop/recovery of the bus supply voltage, to ensure that, with multiple devices in the line, the telegrams sent by the various devices do not collide when the bus voltage is restored.
The values that can be set are:

- **11.. 21 seconds (depending on physical address) (default value)**
- 5.. 9 seconds
- 11 seconds
- 13 seconds
- 15 seconds
- 17 seconds
- 19 seconds
- 21 seconds
- no delay

Setting the values **11.. 21 seconds (depending on physical address)** and **5.. 9 seconds**, the device automatically calculates the transmission delay according to an algorithm that examines the physical address of the device itself; the presented values (11/21 or 5/9) indicate the extremes of the value interval that can be calculated.

4 “Channel x” menu (independent channels)

If channel operation is independent, a dedicated menu is shown for each channel called **Channel x** (x is the channel index). The menu structure changes based on the value set for the **“Matched function”** parameter.
For the sake of simplicity, the parameters enabled according to the value set for the above parameter are listed in the following paragraphs.

Figure 4.1 shows the basic structure of the menu:

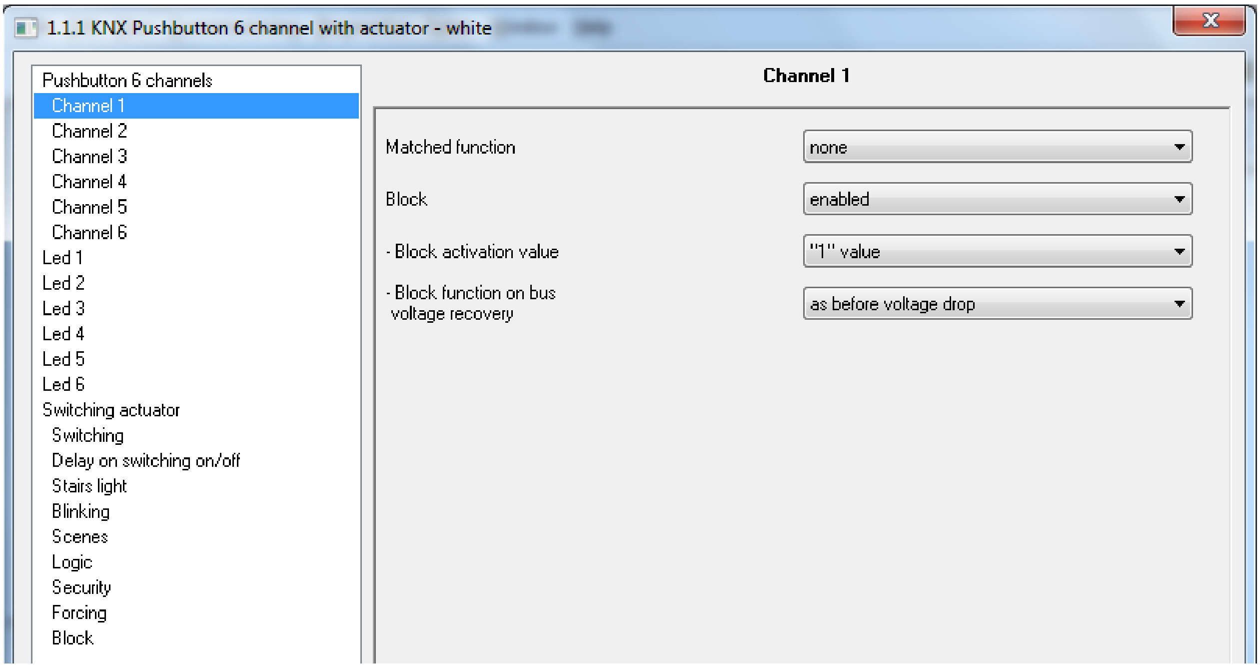


Fig 4.1

4.1 Parameters

➢ 4.1.1 Matched function

This is used to define the function implemented by the channel; The values that can be set are:

- ***none (default value)***
- ***edges/sequence commands***
  See chapter 5 Function *“**edges/sequence commands**”*
- ***1 push button + stop dimmer***
  See chapter 6 Function “***1 push button + stop dimmer***”
- ***cyclic sending 1 push button dimmer***
  See chapter 7 Function “***cyclic sending 1 push button dimmer***”
- ***1 push button shutter control***
  See chapter 8 Function “***1 push button shutter control***”
- ***scene management***
  See chapter 9 Function “***scene management***”
- ***switching sequences***
  See chapter 10 Function “***switching sequences***”
- ***actuator local command***
  See chapter 11 Function “***actuator local command***”

➢ *4.1.2 Block*

To inhibit the channel when sending commands associated with pressing/release the button, the block function must be activated: this function inhibits the detection of button pressing/releasing, preventing the device from sending the telegrams associated with these events on the bus; if activated, any change in status that occurs will not be interpreted until a block deactivation command is received.
This parameter is used to enable/disable the function and can have the following values:

- **disabled (default value)**
- enabled

Setting the value **enabled** displays the parameters **"Block activation value"** and **"Block function on bus voltage recovery"** and the communication object **Ch.x - Block** through which it is possible to activate the function via a bus command.

The parameter **"Block activation value"** makes it possible to set which logic value the bit received via bus telegram should assume to activate the block function; The values that can be set are:

- value "0"
- **value "1" (default value)**

The parameter **"Block function on bus voltage recovery"** is used to set the status of the block function on recovery; The values that can be set are:

- disabled
- enabled
- **as before voltage drop (default value)**

5 Function *“edges/sequence commands”*

This function is used to set the type and number of commands to send after a status change has been detected, for up to a total of eight commands per channel; it is possible to differentiate the type of command depending on the event that is detected (pressing and releasing) and delay the sending of the commands with a fixed settable time.
The basic structure of the menu is as follows (Fig. 5.1):

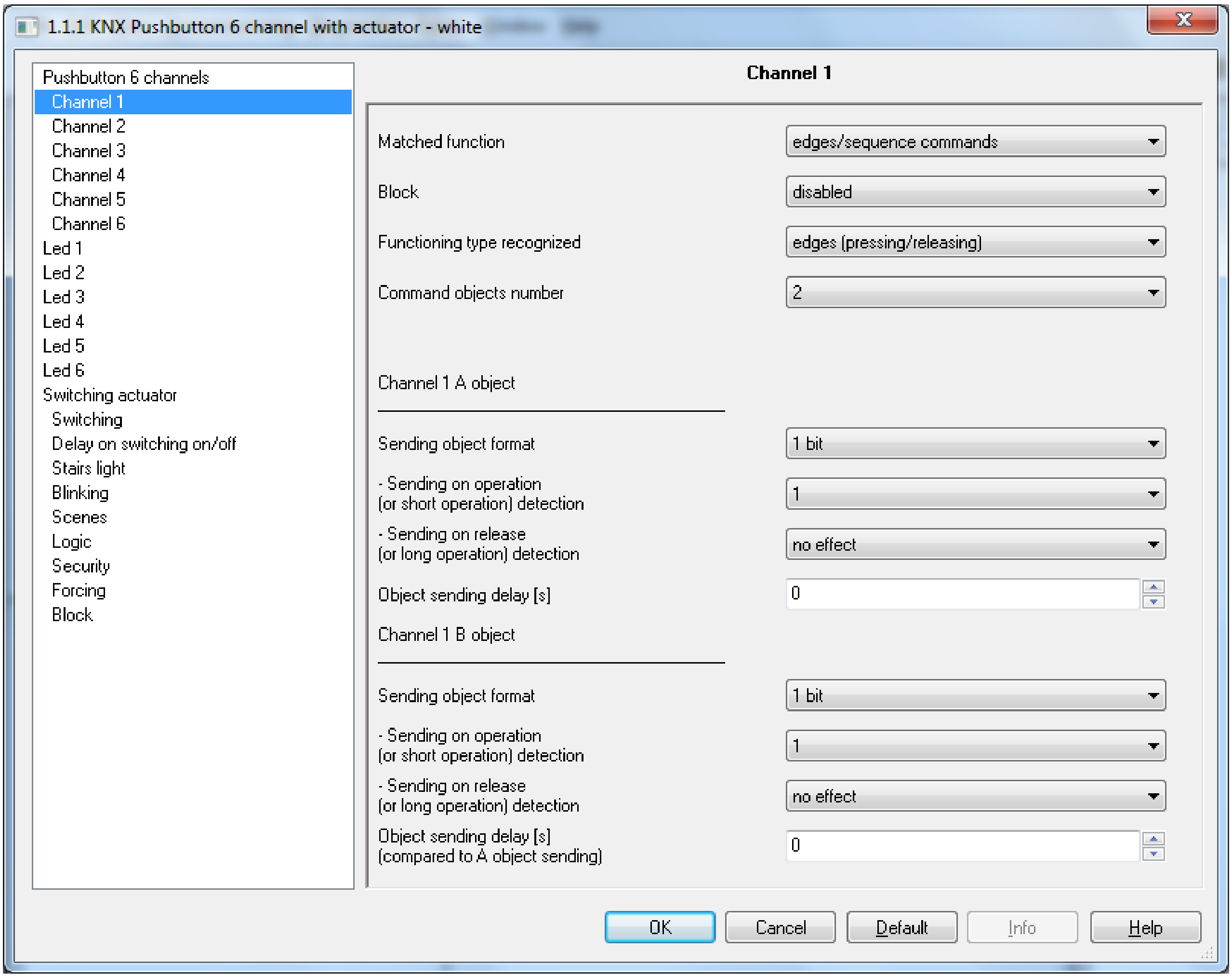


Fig. 5.1

5.1 *Parameters*

➢ *5.1.1 Functioning type recognized*

This is used to define which type of button operation generates the sending of the sequence commands; The values that can be set are:

- **edges (pressing/releasing)** **(default value)**
- short operation/long operation

➢ *5.1.2 Command objects number*

This is used to define how many communication objects are managed by the channel in question; The values that can be set are:

- **1 (default value)**, 2, …, 8

Depending on the set value, the following parameters appear for each of the selected object **“Sending object format”**, **“Sending on operation (or short operation) detection”**, **“Sending on release (or long operation) detection”** and **“Object sending delay [s]”** grouped in the subset **Channel x z object** (z is the index of the object associated with the channel, between **A** and **H)**.

The parameter **“Sending object format”** makes it possible to set the format and code of the bus telegram that will be sent by the device. The values that can be set are:

- **1 bit** **(default value)**
- 2 bits
- 1 byte unsigned value
- 1 byte signed value
- 1 byte percentage value
- 1 byte HVAC mode
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value
- 14 bytes

The value set for this item will change as a result the values set for the parameters **“Sending on operation (or short operation) detection”** and **“Sending on release (or long operation) detection”**.
The parameter **“Sending on operation (or short operation) detection”** is used to set the command or the value to send when pressing of the button associated with the channel is detected.
The parameter **“Sending on release (or long operation) detection”** is used to set the command or the value to send when the release of the button associated with the channel is detected.

– If the format of the object to send is **1 bit**, the communication object ***Ch.x - z object 1 bit value*** will be visible and the values that can be set for the two above parameters are:

  - **no effect** **(release detection default value)**
  - 0
  - **1** **(pressing detection default value)**
  - cyclic switching

  Selecting the value **cyclic switching**, the parameter **“Status feedback object”** will be shown, which makes it possible to enable and display the communication object ***Ch.x - z object feedback status***; by enabling this object, when the status feedback telegram is received for the object in question, the command that the push-button panel will send (via the object ***Ch.x - z object 1 bit value***) when the event associated with the cyclic sending is detected will be the opposite of the value generated by the most recent event between the bus value received on object ***Ch.x - z object status feedback*** and the last sent value (via the object ***Ch.x - z object 1 bit value***).

  The **“Status feedback object”** may have the following values:

  - **disabled** **(default value)**
  - enabled

  Selecting the value **enabled** displays the communication object ***Ch.x - z object status feedback***. Each time bus voltage is reset, the device sends a status reading command for this object to update the push-button panel about the status of the connected devices.

– If the format of the object to send is **2 bits**, the communication object ***Ch.x - z object 2 bits value*** will be visible and the values that can be set for the two above parameters are:

- **no effect** **(release default value)**
- **activate on (down) forcing** **(default pressing value)**
- activate off (up) forcing
- disable forced positioning
- cyclical switching forcing on/forcing off
- cyclical switching forcing on/deactivate forcing
- cyclical switching forcing off/deactivate forcing

By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.

– If the format of the object to send is **1 byte unsigned**, the communication object ***Ch.x - z object 1 byte value*** will be visible and the values that can be set for the two above parameters are:

- **no effect** **(release default value)**
- **value sending** **(pressing default value)**

By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **"Value (0 .. 255)"** which can assume the following values:

- from **0 (default value)** to 255

– If the format of the object to send is **1 byte signed**, the communication object ***Ch.x - z object 1 byte value*** will be visible and the values that can be set for the two above parameters are:

- **no effect** **(release default value)**
- **value sending** **(pressing default value)**

By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **"Value (-128 .. 127)"** which can assume the following values:

- from -128 to 127, **0 (default value)**

– If the format of the object to send is **1 byte percentage**, the communication object ***Ch.x - z object 1 byte value*** will be visible and the values that can be set for the two above parameters are:

- **no effect** **(release default value)**
- **value sending** **(pressing default value)**

By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **"Value (0% .. 100%)"** which can assume the following values:

- from **0 (default value)** to 100

– If the format of the object to send is **1 byte HVAC mode**, the communication object ***Ch.x - z object 1 byte value*** will be visible and the values that can be set for the two above parameters are:

- **no effect** **(release default value)**
- auto
- **comfort** **(pressing default value)**

- precomfort
- economy
- off (building protection)
- cyclical switching (thermostat)
- cyclical switching (timed thermostat)

By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.

By selecting the value **cyclical switching (thermostat)**, each time the associated event is detected (pressing/releasing) the device sends a new thermoregulation mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* …; By selecting the value **cyclical switching (chronothermostat),** each time the associated event is detected (pressing/releasing) the device sends a new thermoregulation mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* …

– If the format of the object to send is **2 bytes unsigned**, the communication object ***Ch.x - z object 2 bytes value*** will be visible and the values that can be set for the two above parameters are:

  - **no effect (release detection default value)**
  - **value sending (pressing detection default value)**

  By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **"Value (0 .. 65535)"** which can assume the following values:

  - from **0 (default value)** to 65535

– If the format of the object to send is **2 bytes signed**, the communication object ***Ch.x - z object 2 bytes value*** will be visible and the values that can be set for the two above parameters are:

  - **no effect (release detection default value)**
  - **value sending (pressing detection default value)**

  By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **"Value (-32768 .. +32767)"** which can assume the following values:

  - from -32768 to +32767, **0 (default value)**

– If the format of the object to send is **4 bytes unsigned**, the communication object ***Ch.x - z object 4 bytes value*** will be visible and the values that can be set for the two above parameters are:

  - **no effect (release detection default value)**
  - **value sending (pressing detection default value)**

  By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **"Value (0 .. 4294967295)"** which can assume the following values:

  - from **0 (default value)** to 4294967295

– If the format of the object to send is **4 bytes signed**, the communication object ***Ch.x - z object 4 bytes value*** will be visible and the values that can be set for the two above parameters are:

  - **no effect (release detection default value)**
  - **value sending (pressing detection default value)**

By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **“Value (-2147483648 .. 2147483647)”** which can assume the following values:

- from -2147483648 to 2147483647, **0 (default value)**

– If the format of the object to send is **14 bytes**, the communication object ***Ch.x - z object 14 bytes value*** will be visible and the values that can be set for the two above parameters are:

- **no effect (release detection default value)**
- **value sending (pressing detection default value)**

By setting **value sending**, it is possible to define the value to be sent via the new displayed parameter **“Value (ISO characters 8859-1)”** which can assume the following values:

- 14 alphanumeric characters with ISO/IEC coding 8859-1

➢ *5.1.3 Object sending delay [s] (0..255 seconds)*

This is used to set the delay between the detection of the event associated with sending the command and the effective sending of the command on the bus.
With regard to the objects that range from index B to index H, this parameter indicates the delay between sending the command/value associated with the object with the previous index (z-1) and sending the command/value associated with the object to which the parameter refers; The delay to which reference is made in these cases is calculated from the moment in which the command/value is sent that is associated with the object with the previous index (z-1) and not from the moment in which the event that generates sending is detected (pressing/releasing or short operation).
The set delay will only be executed if the event in progress, associated to the object to which the parameter refers, is associated with any value other than **no effect**; otherwise, the delay is ignored.
The parameter may assume the following values:

- from **0 (default value)** to 255 seconds

NOTE : If a command sequence is being sent with delays, activated by the detection of a certain event (pressing/releasing), the detection of the opposite event will stop the sending of the sequence only if at least one of the actions associated with the detection of the last event is different than **no effect**; otherwise, the command/value sequence will be continue to be sent until the last command/value is sent.

6 Function *"1 push button + stop dimmer"*

This is used to configure the channel to control a dimmer with a single button, increasing and decreasing dimmer brightness always using the same button.
For sending on/off telegrams and brightness control telegrams.
As there is only one button to manage the On/Off and brightness control functions, the operation is managed by differentiating between short operations and long operations:

– a long operation is transformed into a brightness control command. When released, an adjustment stop telegram is sent to stop the brightness increase/decrease operation for the dimmer and to fix the brightness value reached at the moment the stop control command was received.
– a short operation is transformed into an on/off command.

Using this type of function, brightness control depends on the so-called brightness control characteristic curve, which varies from actuator to actuator, based on how the manufacturer designed the curve that regulates power, and as a result brightness. This means that the speed with which brightness reaches its maximum and minimum value does not depend on the commands sent from the push-button panel, but the latter regulates the brightness itself by stopping its increase/decrease based on the desired value. The communication objects that this function enables are ***Ch.x - Switching*** and ***Ch.x - Brightness dimming***.

The basic structure of the menu is as follows (Fig. 6.1):

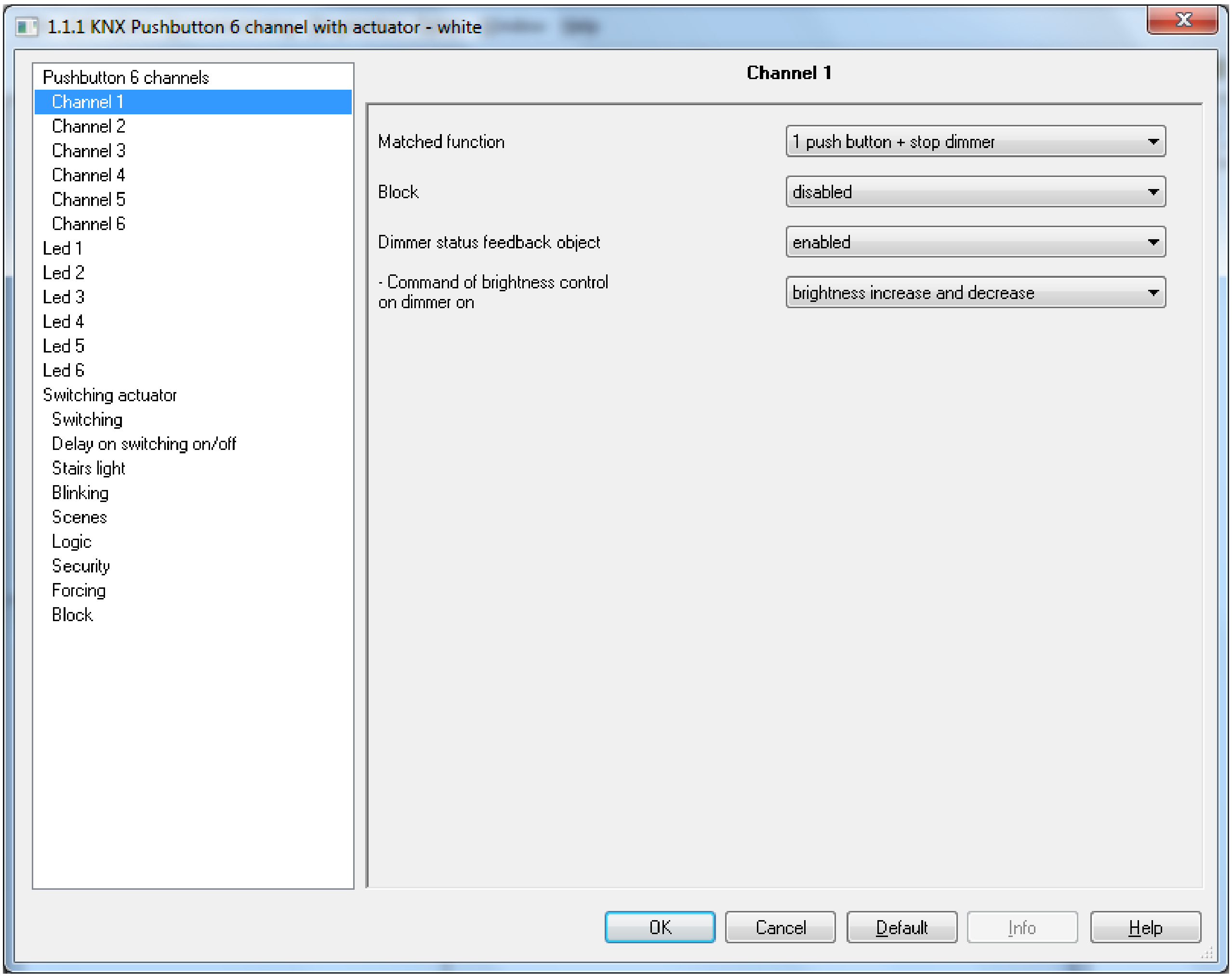


Fig. 6.1

The push-button panel foresees that if the command to be sent is the opposite of the last command sent, this is transformed into:

– long operation: if the last sent command was an off command or a decrease brightness command, the new command will be an increase brightness command; vice versa, if the last sent command was an on command or an increase brightness command, the new command will be a decrease brightness command. In both cases, when released, an adjustment stop telegram is sent to stop the brightness increase/decrease operation for the dimmer and to fix the brightness value reached at the moment the stop control command was received.
– short operation: if the last sent command was an on command, the new command will be an off command; vice versa, if the last sent command was an off command, the new command will be an on command; the brightness increase/decrease control commands in this case do not determine the value of the last command sent to distinguish the value of the new command to be sent.

This behaviour is changed if the user enables the communication object ***Ch.x - Dimmer status feedback***, via the parameter **“Dimmer status feedback object”**; this parameter may have the following values:

- **disabled (default value)**
- enable

Selecting the value **enable** displays the parameter **“Brightness control commands with dimmer on”** and the communication object ***Ch.x - Dimmer status feedback***, which makes it possible to receive status feedback from the controlled dimmer actuator; the behaviour of the push-button panel is modified as follows:

– long operation: the command that the push-button panel sends depend on the parameter **“Brightness control commands with dimmer on”**, which can assume the following values:

  - only brightness increase
  - only brightness decrease
  - **brightness increase and decrease (default value)**

  By setting **brightness increase and decrease**, if the value of the last two events "last sent command" and "dimmer status feedback" is ON, the new brightness adjustment command to be sent will be the opposite of the last sent command; when released, an adjustment stop telegram is sent to stop the brightness increase/decrease operation for the dimmer and to fix the brightness value reached at the moment the stop control command was received; if the value of the last of the two events "last sent command" and "dimmer status feedback" is OFF, the first command to be sent is increase brightness value, followed by sending the command opposite of the last one sent.
– short operation: if the value of the last of the two events "last sent command" and "dimmer status feedback" is ON, the new command will be an off command; vice versa, if the value of the last of the two events "last sent command" and "dimmer status feedback" is OFF, the new command will be an on command.

If the feedback object is enabled, each time bus voltage is reset, the device sends a status reading command for this object to update the push-button panel about the status of the connected devices.

7 Function *“cyclic sending 1 push button dimmer”*

This is used to configure the channel to control a dimmer with a single button, increasing and decreasing dimmer brightness always using the same button, with defined and settable control steps.
As there is only one button to manage the On/Off and brightness control functions, the operation is managed so that each time the button is pressed the opposite command is sent in comparison to the last sent command and by differentiating between short operations and long operations:

– a long operation is transformed into a brightness control command. No telegram is sent when released.
– a short operation is transformed into an on/off command.

Unlike the function ***1 push button + stop dimmer***, it is possible to define both the brightness various steps as well as the time that passes between sending different commands, if the long operation continues over time; therefore the sending of the control stop telegram is not necessary when releasing the button, even though the control follows the power/brightness characteristic curve, as it is the command that is sent from the push-button panel that determines the percentage variation. The communication objects that this function enables are ***Ch.x - Switching*** and ***Ch.x - Brightness dimming***.

The structure of the menu is as follows (Fig. 7.1):

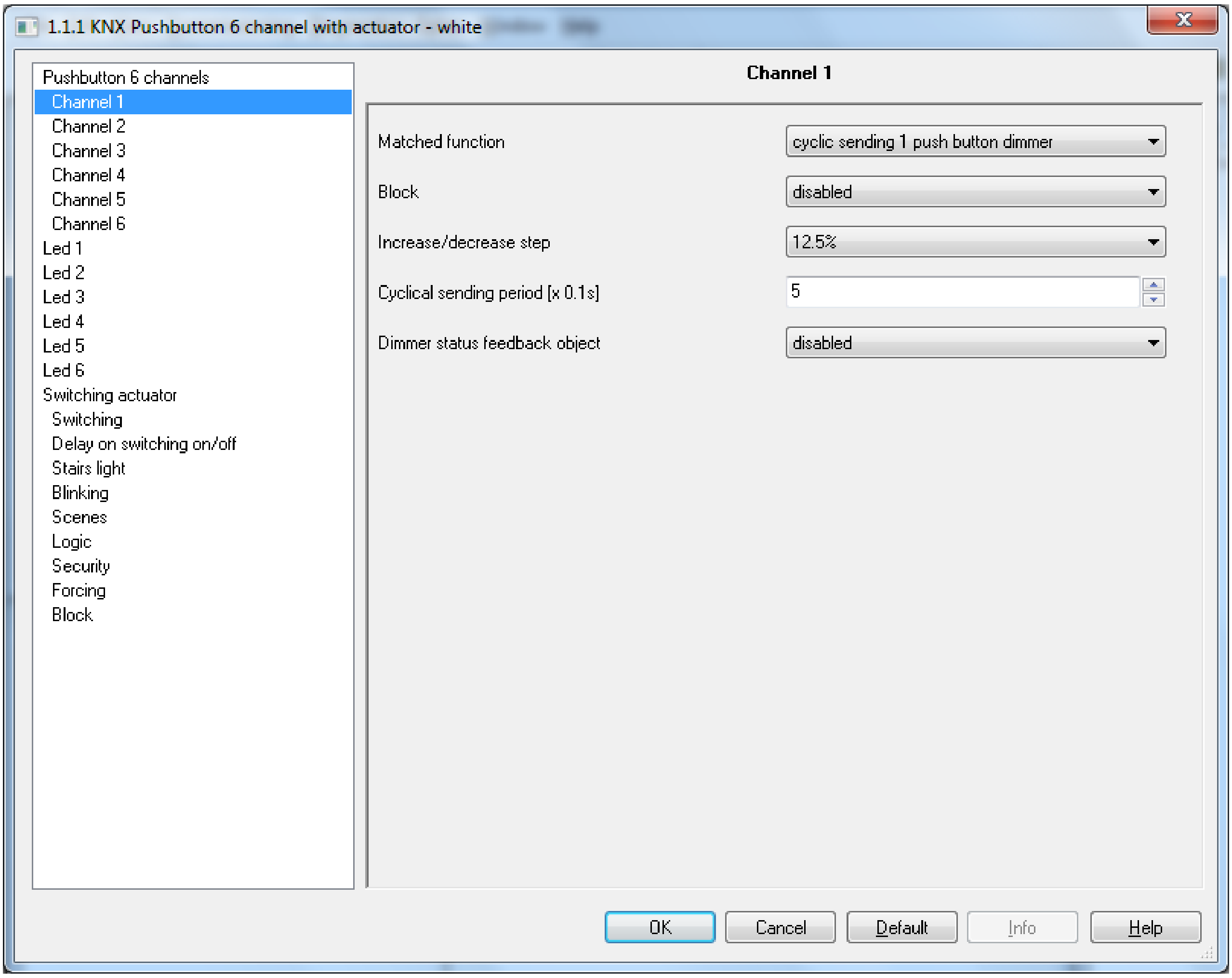


Fig. 7.1

7.1 Parameters

➢ *7.1.1 Increase/decrease step*

This is used to set the percentage value of the brightness variation associated with the brightness increase/decrease commands. In this way, as soon as a long operation is detected, the device sends the first increase/decrease command with the set percentage; The values that can be set are:

- 100%
- 50%
- 25%
- **12.5% (default value)**
- 6.25%
- 3.125%
- 1.56%

If pressing is maintained, the device will cyclically send the command until release is detected.

➢ *7.1.2 Cyclical sending period [x 0.1s]*

This is used to set the time that passes between sending subsequent increase/decrease commands if pressing is maintained. When released, no telegram is sent but only the cyclical sending of the brightness control commands is stopped.
The values that can be set are:

- from 3 to 50, **5 (default value)**

In summary, if a long operation is detected, the device sends the first increase/decrease command with the set percentage and, if the pressing is maintained, it will cyclically send the command until release is detected.

EXAMPLE: for example, if the value for item **Long operation minimum time** in the ***Main*** menu is set to ***0.5 sec***, the parameter **Increase/decrease step** is set to **12.5%** and parameter **Cyclical sending period [x 0.1s]** is set to **3** (0.3 sec) and pressing is detected:

– 0.5 seconds after detecting the pressing of the button, a long operation is recognized and as a result the first 12.5% brightness increase/decrease telegram is sent
– from this moment, for every 0.3 seconds that pressing is continued, the device will send a new 12.5% brightness increase/decrease command until the release of the button is detected
– when released, no telegram is sent but the cyclical sending is stopped

➢ *7.1.3 Dimmer status feedback object*

Refer to chapter 6 for the settings of this parameter

8 Function *“1 push button shutter control”*

This is used to configure the channel to control a shutter with a single button, regulating the upward and downward travel of the shutter and, depending on the device version, controlling louvres opening/closing.

As only one button manages the louvres up/down and control functions, the operation is managed so that each time the button is pressed, a command is sent that is the opposite of the last movement signal received by the actuator that manages the shutter; a differentiation is made between short and long operations:

– a long operation is transformed into an up/down movement command. If the last received movement signal was “up”, the new command will be a down command, and vice versa.
– a short operation is transformed into a louvres control command. If the last received movement signal was “up”, the new command will be a closing louvres control command however, if the last received movement signal was “down”, the new command will be an opening louvres control command. If the shutter is moving, the louvres control command will only stop the shutter up/down movement.

The communication objects this function enables are ***Ch.x - Shutter movement***, ***Ch.x - Shutter stop/Louvres control and Ch.x - Movement feedback***.

The structure of the menu is as follows (fig. 8.1):

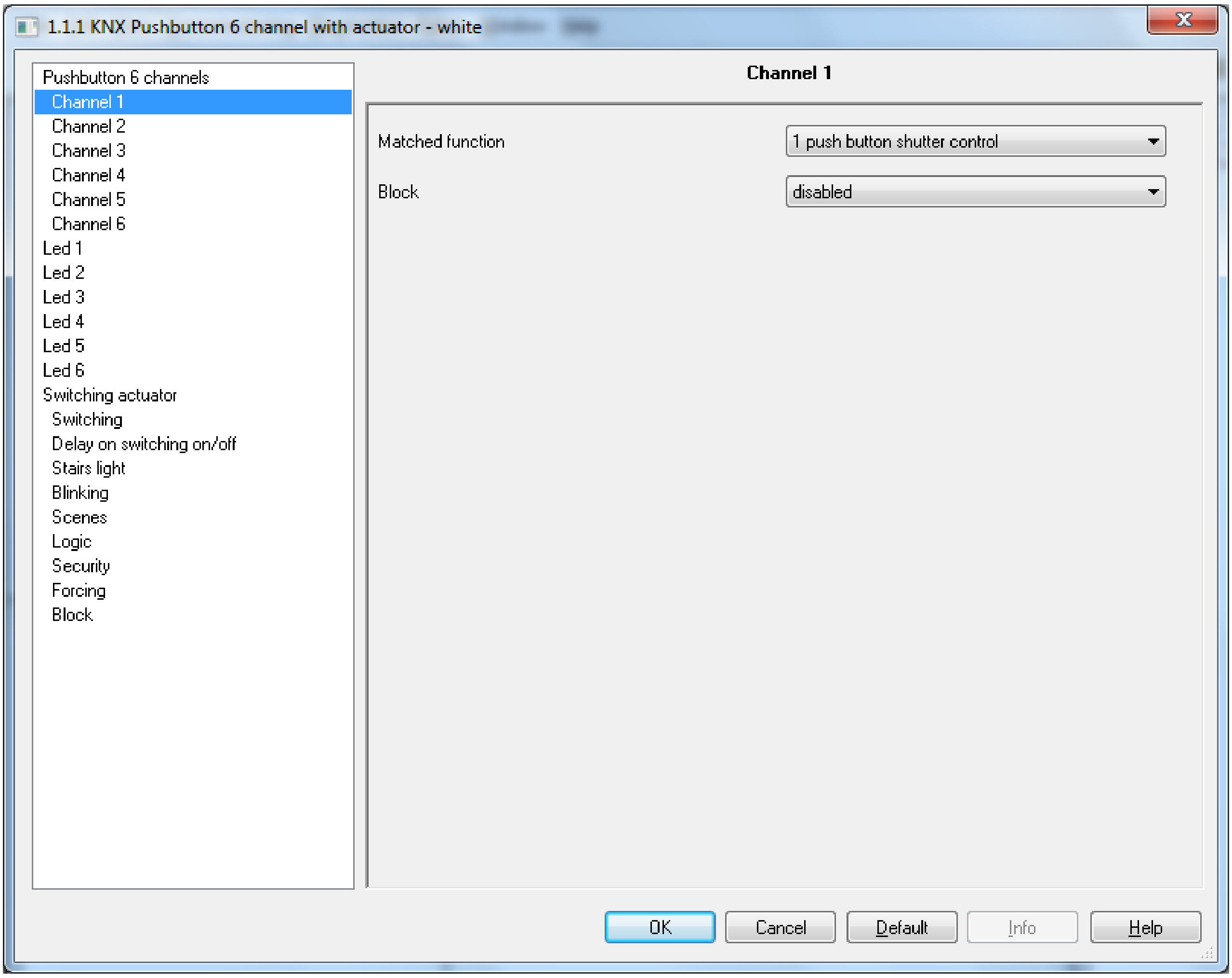


Fig. 8.1

9 Function *"Scene management"*

This is used to configure the channel to send scene memorizing and execution commands, with the possibility of sending the scene memorizing command following a command received from the bus. Only one scene can be managed for each channel, and a differentiation is made between short and long operations:

– a long operation is transformed into a scene storing command.
– a long operation is transformed into a scene execution command.

The communication objects that this function enables are ***Ch.x - Scene*** and ***Ch.x - Scene storing trigger***.

When a long operation is recognized, to provide the user with a visual confirmation that the scene storing command was sent, night signalling will be deactivated for a brief period (blink); If night lighting is disabled, when the long operation is detected, a light signalling (blink) is briefly activated with the currently selected color; this effect has priority over all the light effects activated by the bus (see chapter 13 "Led X menu"). The structure of the menu is as follows (fig. 9.1):

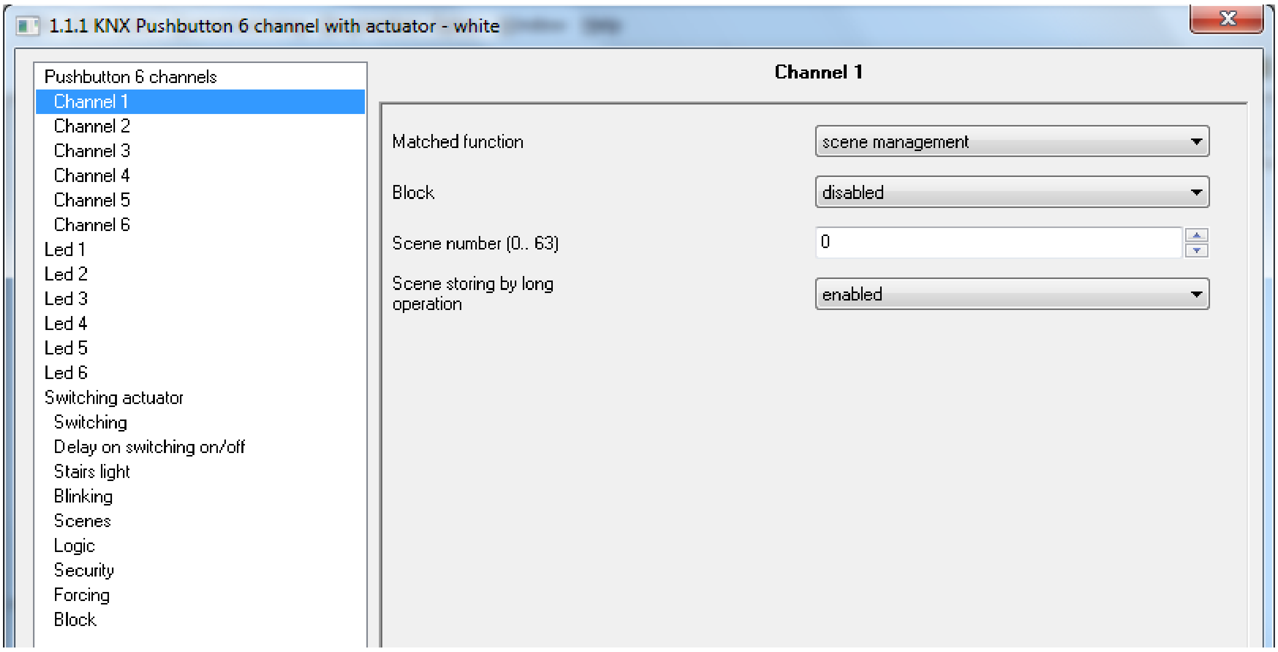


Fig. 9.1

9.1 Parameters

➢ *9.1.1 Scene number (0..63)*

This is used to set the value of the scene to be recalled/stored and as a result the relative values that are sent via the object ***Ch.x - Scene***. The possible values are:

- from **0 (default value)** to 63

➢ *9.1.2 Scene storing by long operation*

This enables the sending of a scene memorizing command when a long operation is recognized; The values that can be set are:

- disabled
- **enabled (default value)**

The device will send the scene storing command after a long operation is detected only if the value **enabled** is selected; by selecting the value **disabled**, a long operation is not recognized and the long operation triggers the sending of the scene execution command (as for a short operation).

Independently of the value set for the above parameter, it is possible to indirectly generate the sending of the scene memorizing command after receiving a bus telegram on the object ***Ch.x - Scene storing trigger*** (both with value "1" as well as with value "0"); each time the device receives a telegram on that object, a scene memorization telegram will be sent immediately.

10 Function *"switching sequences"*

This is used to send a command sequence after a certain pressing has been detected.
The structure of the menu is as follows (fig. 10.1):

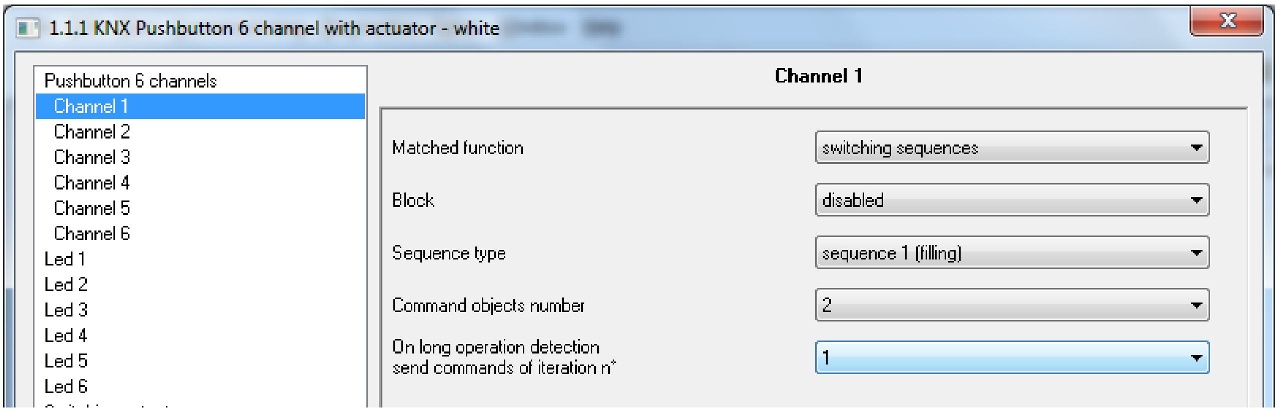


Fig. 10.1

10.1 Parameters

➢ *10.1.1 Command objects number*

This is used to set the number of commands that make up the sequence itself; based on the value set for this item, the communication objects ***Ch.x - Sequence z*** are enabled (with **z** between A and H).
The values that can be set are:

- from **2 (default value)** to 8

➢ *10.1.2 Sequence type*

This is used to set the type of sequence to be sent.
The values that can be set are:

- **sequence 1 (filling) (default value)**
- sequence 2 (sum)
- sequence 3 (random)

Sequence 1 (filling) consists in: each time pressing is detected (edge) the device sends a sequence that follows the filling progress on the communication objects. This sequence consists in activating one communication object a time, in cascade, until all the objects have the logical value "1" and in deactivating the objects in cascade until they again have the logical value "0". Taking into consideration a sequence that includes 3 commands, at each iteration, the sent commands will be:

Edge no.	Value sent on ***Ch.x - C sequence***	Value sent on ***Ch.x - B sequence***	Value sent on ***Ch.x - A sequence***
1st edge	0	0	1
2nd edge	0	1	1
3rd edge	1	1	1
4th edge	0	1	1
5th edge	0	0	1
6th edge	0	0	0

Once the 6th edge is detected, the sequence will start from the beginning

The table shows how, considering the increasing/decreasing trend of the sequence, the most significant bit of the sequence, in this particular case, is the one for the communication object ***Ch.x - C sequence*** whereas the least significant is always the one for the object ***Ch.x - A sequence***.

Sequence 2 (sum) consists in: each time pressing is detected (edge) the device sends a sequence that follows the sum progress on the communication objects. This sequence consists in counting the detected edges and converting this value into a binary format, distributing it on the enabled communication objects. Taking into consideration a sequence that includes 3 commands, at each iteration, the sent commands will be:

Edge no.	Value sent on ***Ch.x - C sequence***	Value sent on ***Ch.x - B sequence***	Value sent on ***Ch.x - A sequence***
1st edge	0	0	1
2nd edge	0	1	0
3rd edge	0	1	1
4th edge	1	0	0
5th edge	1	0	1
6th edge	1	1	0
7th edge	1	1	1
8th edge	0	0	0

Once the 8th edge is detected, the sequence will start from the beginning

The table shows how the trend of the sent commands depends on the count of the detected edge; in fact it starts with the binary coding of value 1 up to, in this specific case, the coding of value 7 and then the counting starts again for the next edge. Also in this case, the most significant bit in the sequence is the one for the communication object ***Ch.x - C sequence*** whereas the least significant is always the one for object ***Ch.x - A sequence***.

Sequence 3 (random) allows the user to directly set the value for each command for each set edge; this setting enables the parameter **"Number of sequence iterations"** and the configuration menu **z object channel x** (one for each enabled command). The parameter **"Number of sequence iterations"** allows to set the number of iterations (edges) that make up the sequence; The values that can be set are:

- from **2 (default value)** to 16

Based on the value set for this item, the **Channel x z object** menu will display or hide the parameters **"Iteration 1 object value"**, **"Iteration 2 object value"**, **"Iteration 3 object value"**, **"Iteration 4 object value"**, **"Iteration 5 object value"**, **"Iteration 6 object value"**, **"Iteration 7 object value"**, **"Iteration 8 object value"**, **"Iteration 9 object value"**, **"Iteration 10 object value"**, **"Iteration 11 object value"**, **"Iteration 12 object value"**, **"Iteration 13 object value"**, **"Iteration 14 object value"**, **"Iteration 15 object value"** and **"Iteration 16 object value"**, which can assume the following values:

- value "0"
- **value "1" (default value)**

The structure of the menu **Channel x z object** is as follows(fig. 10.2):

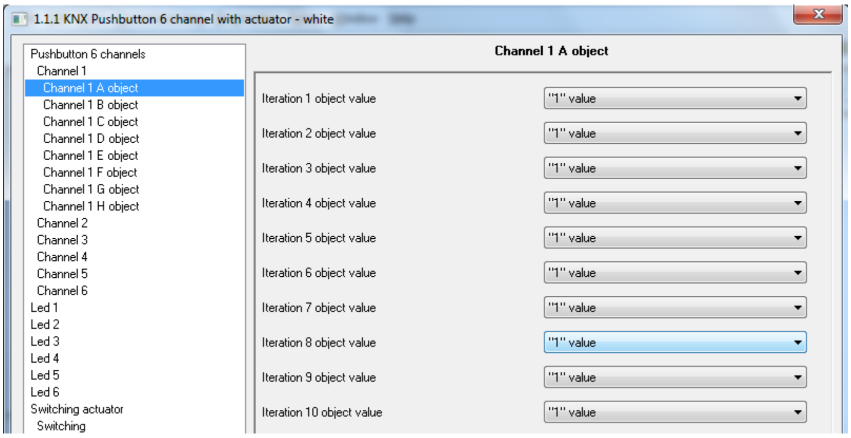


Fig. 10.2

➢ *10.1.3 On long operation detection, send commands of iteration n°*

This is used to define which iteration of the sequence is to be sent when a long operation is detected; The values that can be set are:

- from 1 to 256, **1 (default value)**

EXAMPLE: with reference to the previous table, supposing that the value set by the user is **3**, when a long operation is detected the device will send:

Edge no.	Value sent on ***Ch.x - C sequence***	Value sent on ***Ch.x - B sequence***	Value sent on ***Ch.x - A sequence***
1st edge	0	0	1
2nd edge	0	1	1
3rd edge	1	1	1
4th edge	0	1	1
5th edge	0	0	1
6th edge	0	0	0

"Filling" sequence

Edge no.	Value sent on ***Ch.x - C sequence***	Value sent on ***Ch.x - B sequence***	Value sent on ***Ch.x - A sequence***
1st edge	0	0	1
2nd edge	0	1	0
3rd edge	0	1	1
4th edge	1	0	0
5th edge	1	0	1
6th edge	1	1	0
7th edge	1	1	1
8th edge	0	0	0

"Sum" sequence

Once the long operation is detected and the sequence relative to the set iteration is sent, the next time a short operation is detected the sequence related to the iteration following the one associated with the long operation will be sent (in this example, the sequence association with iteration 4 will be sent).
In summary, the value set for the parameter **"On long operation detection, send commands of iteration n°"** defines both the sequence to be sent as well as the value with which the iteration counter should start when a long operation is detected.

11 *“Actuator local command”* function

This is used to dedicate the channel to controlling the actuator on the device without having to make additional logical connections via the ETS software. No communication object is enabled by this function.

The structure of the menu is as follows (fig. 11.1):

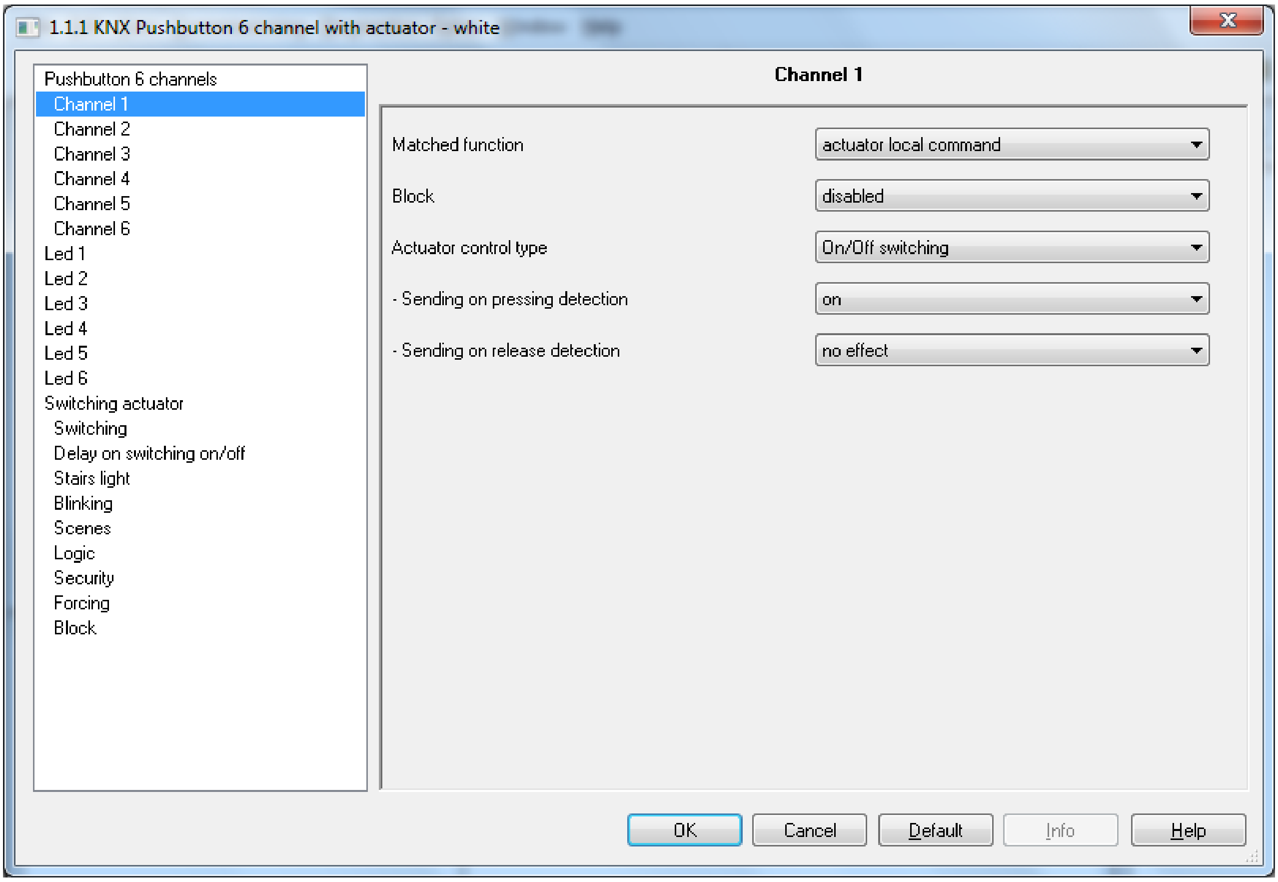


Fig. 11.1

11.1 Parameters

➢ *11.1.1 Actuator control type*

As the actuator on board the device implements various types of operations (On/Off switching, timed activation, activation/deactivation delay, blinking and scenes) and functions with different priorities (forcing and block), it is necessary to define which of those functions the push-button associated with the channel must perform using the parameter **“Actuator control type”**; the values that can be set are:

- **On/Off switching(default value)**
- Delay on switching on/off
- stairs light
- flashing
- scene
- forced positioning
- block
- local command pushbutton

The difference between the values **On/Off switching** and **local command button** is due to the fact that the first acts as a command received from the bus on the object ***Actuator switching*** (and as a result has lower priority than the safety, forcing and block functions of the actuator) whereas the second directly switches the relay, ignoring any active function, whose activation status is not changed in any manner.

– If the actuator control type is **On/Off switching** or **local command button**, the values that can be set for the two above parameters are:

  - off
  - **on (pressing default value)**
  - cyclical switching
  - **no effect (release default value)**

  by selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the on board relay status.

– If the type of actuator control is **delay on switching on/off**, the values that can be set for the two above parameters are:

  - deactivation
  - **activation (pressing default value)**
  - cyclical switching
  - **no effect (release default value)**

  By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.

– If the type of actuator control is **stairs light**, the values that can be set for the two above parameters are:

  - timing stop
  - **timing start (pressing default value)**
  - cyclical switching
  - **no effect (release default value)**

  By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.

– If the type of actuator control is **blinking**, the values that can be set for the two above parameters are:

  - deactivation
  - **activation (pressing default value)**
  - cyclical switching
  - **no effect (release default value)**

  By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.

– If the type of actuator control is **forced positioning**, the values that can be set for the two above parameters are:

  - **active on forcing (default pressing value)**
  - active off forcing
  - disable forced positioning
  - cyclical switching forcing on/forcing off

- cyclical switching forcing on/deactivate forcing
- cyclical switching forcing off/deactivate forcing
- **no effect (release default value)**

By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.

– If the type of actuator control is **block**, the values that can be set for the two above parameters are:

- deactivation
- **activation (pressing default value)**
- cyclical switching
- **no effect (release default value)**

By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.

– If the type of actuator control is **scenes**, the two above parameters are not displayed, but the parameter **"Scene number (0.. 63)"** is shown together with the parameter **"Scene storing by long operation"**.
The parameter **"Scene number (0.. 63)"** is used to set the value of the scene to be recalled/stored; if this value does not coincide with what is associated with the relative parameter in the **Scene** menu of the actuator channel, no scene will be recalled/memorized. The possible values are:

- from **0 (default value)** to 63

Scene storing by long operation enables the sending of a scene memorizing command when a long operation is recognized. The values that can be set are:

- disabled
- **enabled (default value)**

The device will send the scene storing command after a long operation is detected only if the value **enabled** is selected; by selecting the value **disabled**, a long operation is not recognized and the long operation triggers the sending of the scene execution command (as for a short operation).

➢ *11.1.2 Sending on pressing detection*

If any other value other than **scene** is selected for the previously described parameter, this parameter will be displayed that makes it possible to set the command to be sent to the on-board actuator after the pressing of the button associated with the channel has been detected.

➢ *11.1.3 Sending on release detection*

If any other value other than **scene** is selected for the parameter described in 11.1.1, this parameter will be displayed that makes it possible to set the command to be sent to the on-board actuator after the release of the button associated with the channel has been detected.

12 *“Channel X/Y” (coupled channels)* menu

If the channel operation is matched, a dedicated menu is displayed for each channel pair, called **Channel x/y**. The menu structure changes based on the value set for the **“Matched function”** parameter. For the sake of simplicity, the parameters enabled according to the value set for the above parameter are listed in the following paragraphs.
The basic structure of the menu is as follows (Fig. 12.1):

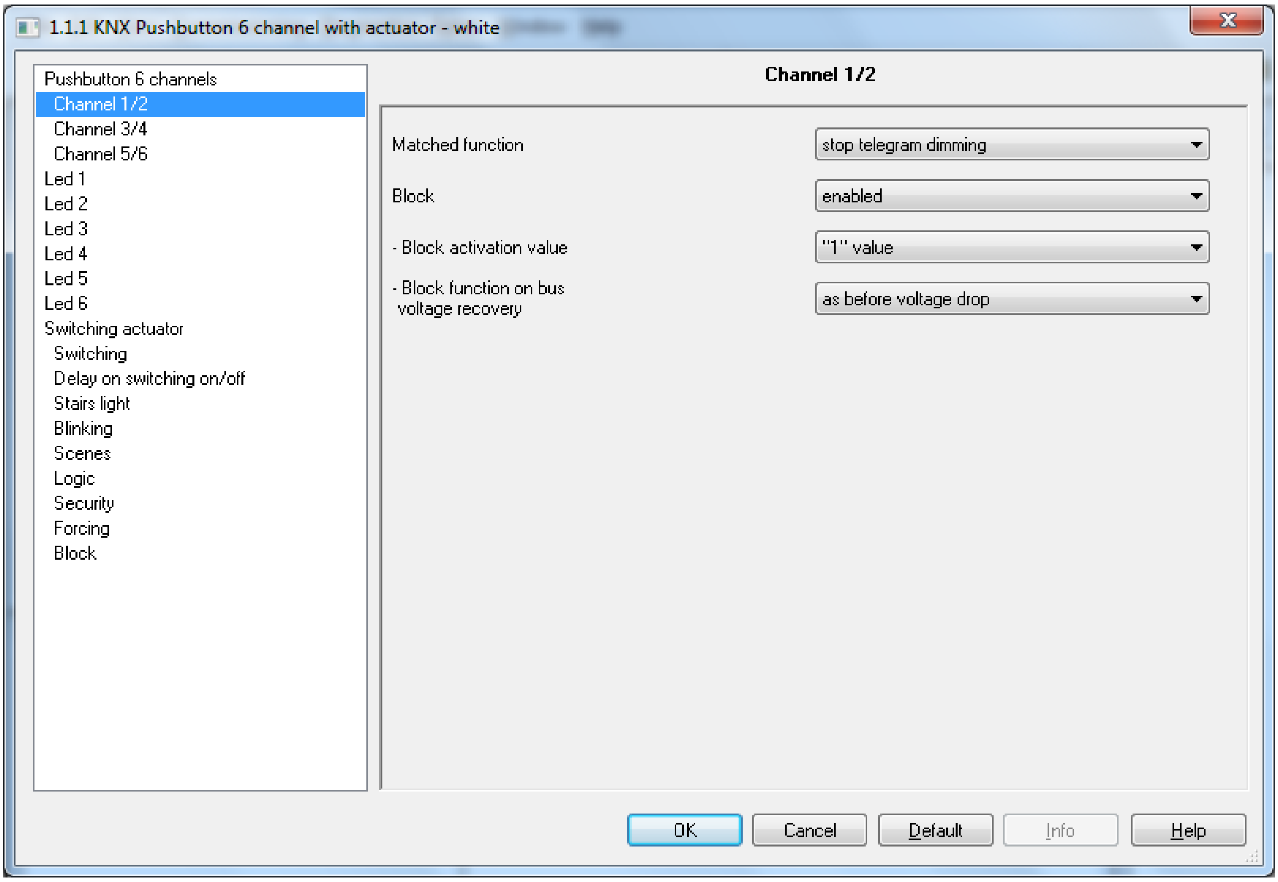


Fig. 12.1

12.1 Parameters

➢ *12.1.1 Matched function*

This is used to define the function implemented by the combined channels; The values that can be set are:

- **stop telegram dimming (default value)**
  (See chapter 6 Function “1 push button + STOP dimmer”)

- cyclic telegram dimming
  (See chapter 7 Function “cyclic sending 1 push button dimmer”)

- shutter control
  (See paragraph 8 Function “1 push button shutter control”)

➢ *12.1.2 Block*

Refer to paragraph 4.1.2 for the settings of this parameter.

13 *"Led X"* menu

This is used to define and personalize the operation of the signalling LEDs associated with the channel.

The signalling LED can assume an amber or green color, one of them can be used for the night lighting function or signalling of motor control actuator movement in progress or both can be independently managed via the relative communication objects. The communication objects enabled by this function are ***Led x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x - Effect 4*** and ***Led x - Effect 5***.

The basic structure of the menu is as follows (fig. 13.1):

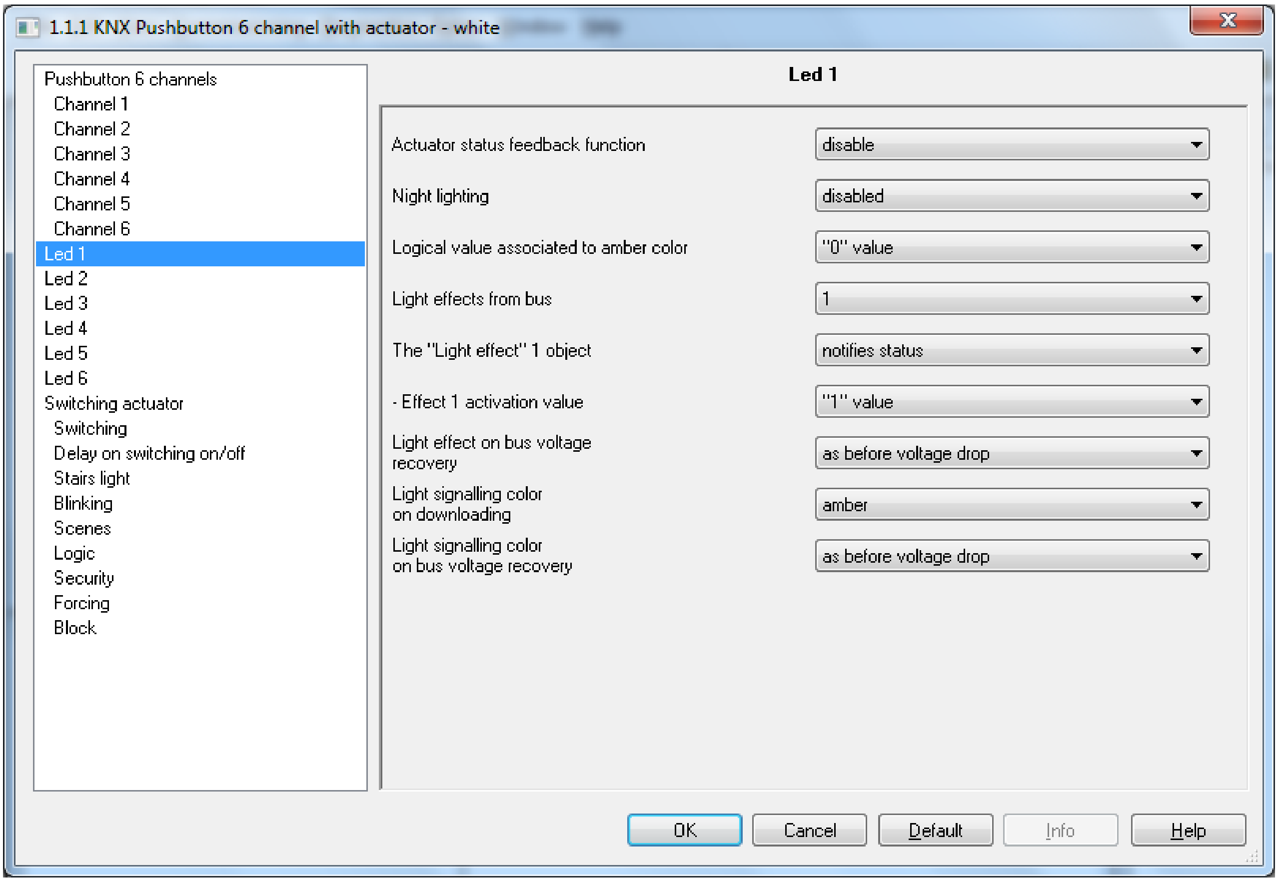
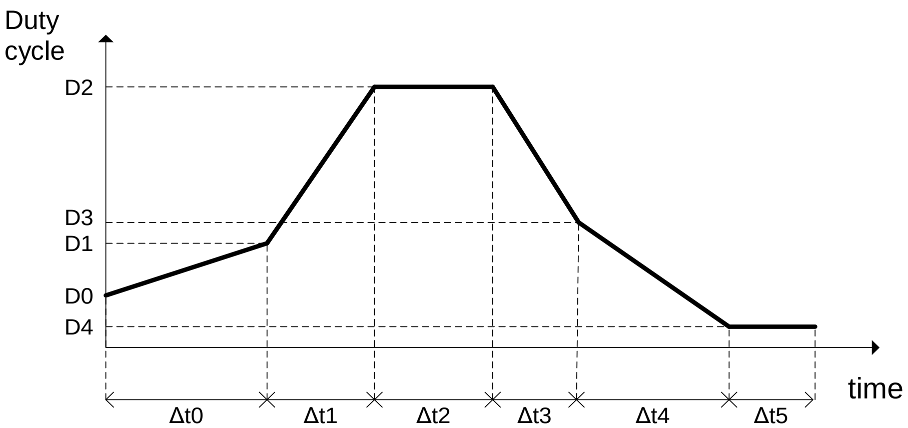


Fig. 13.1

Given how complex it is for the final user to understand all the parameters that are theoretically configurable to obtain the desired light effect (see figure below), a predefined set of light effects is defined, limiting as far as possible the number of parameters that can be configured for each effect.



13.1 Parameters

➢ 13.1.1 Actuator status feedback function

As the actuator on board the device does not have its own status light signalling LED, the LED associated with channel x can carry out this function; this parameter therefore makes it possible to configure the actuator status signalling led and can assume the following values:

- **Disabled (default value)**
- enable

➢ 13.1.2 Night lighting

This is used to enable and define the night lighting color associated with channel x; The values that can be set are:

- disabled
- **enable amber signalling (default value)**
- enable green signalling

Selecting the value **disabled**, displays the parameter **”LED color for signalling active load status”**; selecting a value other than **disabled** displays the parameter **"LED brightness percentage value for night lighting"**.

The parameter **”LED color for signalling active load status”** is used to associate a certain color to the light signalling that indicates the activation status of the load controlled by the activator on board the device; The values that can be set are:

- amber
- **green (default value)**

The parameter **”LED brightness percentage value for night lighting”** is used to define the LED brightness intensity percentage with the night lighting function; The values that can be set are:

- from 5% to **100% (default value)**

When the load is activated (NO contact closed/NC contact open), the LED used for status signalling is on fixed; vice versa, with the load deactivated (NO contact open/NC contact closed) the signalling LED is off (localisation LED on, if enabled).

With the parameter **”Night lighting”**, it is possible to enable and define the color of the night lighting associated with channel x; The values that can be set are:

- disabled
- **enable amber signalling (default value)**
- enable green signalling

Selecting the value **disabled**, displays the communication object ***Led x - color choice*** that allows to change from the bus the signalling LED color and the parameters **“Logical value associated to amber color”, “Light signalling color on downloading"** and **“Light signalling color on bus voltage recovery”**.
Selecting a value other than **disabled** displays the parameter **"LED brightness percentage value for night lighting"**.

➢ 13.1.3 Logical value associated to amber color

This defines which logical value received on the communication object ***Led x - color choice*** is associated with the amber color; as a result, the opposite logical value is associated with the green color. The values that can be set are:

- **value “0” (default value)**
- value “1”

➢ *13.1.4 LED brightness percentage value for night lighting*
This is used to define the LED brightness intensity percentage with the night lighting function; The values that can be set are:

from 5% to **100% (default value)**

➢ *13.1.5 Light effects from bus*
This is used to enable various communication objects for the activation of the light signalling via bus telegram; The values that can be set are:

- **none (default value)**
- 1
- 2
- 3
- 4
- 5

Based on the number of selected effects, this will display the parameters **“The “Light effect” 1 object”**, **“Effect 1 activation value”**, **“The “Light effect” 2 object”**, **“Effect 2 activation value”**, **“The “Light effect” 3 object”**, **“Effect 3 activation value”**, **“The “Light effect” 4 object”**, **“Effect 4 activation value”** and **“The “Light effect” 5 object”**, **“Effect 5 activation value”**.

Parameters **“The “Light effect” 1 object”**, **“The “Light effect” 2 object”**, **“The “Light effect” 3 object”**, **“The “Light effect” 4 object”** and **“The “Light effect” 5 object”** are used to associate the luminous effect to display via the bus communication objects Led ***x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x - Effect 4*** and ***Led x - Effect 5***; via these communication objects, it is possible to activate/deactivate the set light signalling from the bus. The values that can be set for this parameter are:

- **notifies status** **(default value)**
- switching night signalling off (visible only if green/amber signalling is enabled)
- activate fast cyclic blinking
- activate slow cyclic blinking
- executes down ramp
- executes heartbeat 1
- executes heartbeat 2
- executes jellyfish
- executes blinking
- executes heartbeat 3
- executes heartbeat 4
- executes fast blinking
- executes slow blinking
- executes very slow blinking
- executes 3 blinkings
- executes personalized effect

If the selected value is **personalized effect**, this will display the new configuration menu **Personalize effect y**, with 1≤Y≤5 (see paragraph 13.2)

The parameters **“Effect 1 activation value”**, **“Effect 2 activation value”**, **“Effect 3 activation value”**, **“Effect 4 activation value”** and **“Effect 5 activation value”** are used to define which logical value received via the objects ***Led x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x -***

Effect 4 and ***Led x - Effect 5*** activates the associated light signalling. The values that can be set for this parameter are:

- value “0”
- **value “1” (default value)**

Via the communication objects ***Led x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x - Effect 4* and *Led x - Effect 5***, it is possible to activate/deactivate the associated light effect via bus commands; by activating a light effect that is different than the one already active, the new effect will be implemented and the old effect will be deactivated. This means that only one effect may be active and, once it is deactivated, the signalling led will deactivate and the night lighting will activate without having to deactivate the previously activated light effects; to deactivate the led, the active light effect must be deactivated.

➢ *13.1.6 Light effect on bus voltage recovery*

Makes it possible to set the light signalling effect that is activated when the bus voltage is recovered. The values that can be set are:

- no effect
- light effect 1
- light effect 2
- light effect 3
- light effect 4
- light effect 5
- **as before voltage drop (default value)**

Selecting the value **no effect**, if night lighting is activate, when bus voltage is recovered the night lighting will be activated. By selecting any value other than **no effect**, if night lighting is active, the selected effect will be replicated when the bus voltage is recovered with coloring that is NOT dedicated to night lighting.

➢ *13.1.7 Light signalling color on downloading*

This is visible if night lighting is disabled, and is used to preset the color of the light signalling color on downloading the application parameters via ETS.
The values that can be set are:

- **amber (default value)**
- green

➢ *13.1.8 Light signalling color on bus voltage recovery*

This is visible if night lighting is disabled, and is used to preset the color of the light signalling color on bus voltage recovery. The values that can be set are:

- amber
- green
- **as before voltage drop (default value)**

In summary, if night lighting is active, the coloring of the light effects activated by the bus command (via objects ***Led x - Effect 1*** etc.) is associated to a LED not used for signalling; vice versa, if night lighting is disabled, the coloring of these effects depends on the communication object ***Led x - color choice*** and on the parameter value **“Light signalling color on bus voltage recovery”**.

13.2 “Personalize effect y” menu

This menu is displayed if the value for parameter **“"Light effect” y object”** in menu **Led x** is set to **execute personalized effect**.

In this menu, all the parameters used for creating the light effects are visible and configurable; in this way, the user can create a personalized light effect. These light effects are activated/deactivated by the objects ***Led x - Effect y***.

The basic structure of the menu is as follows (fig. 13.2):

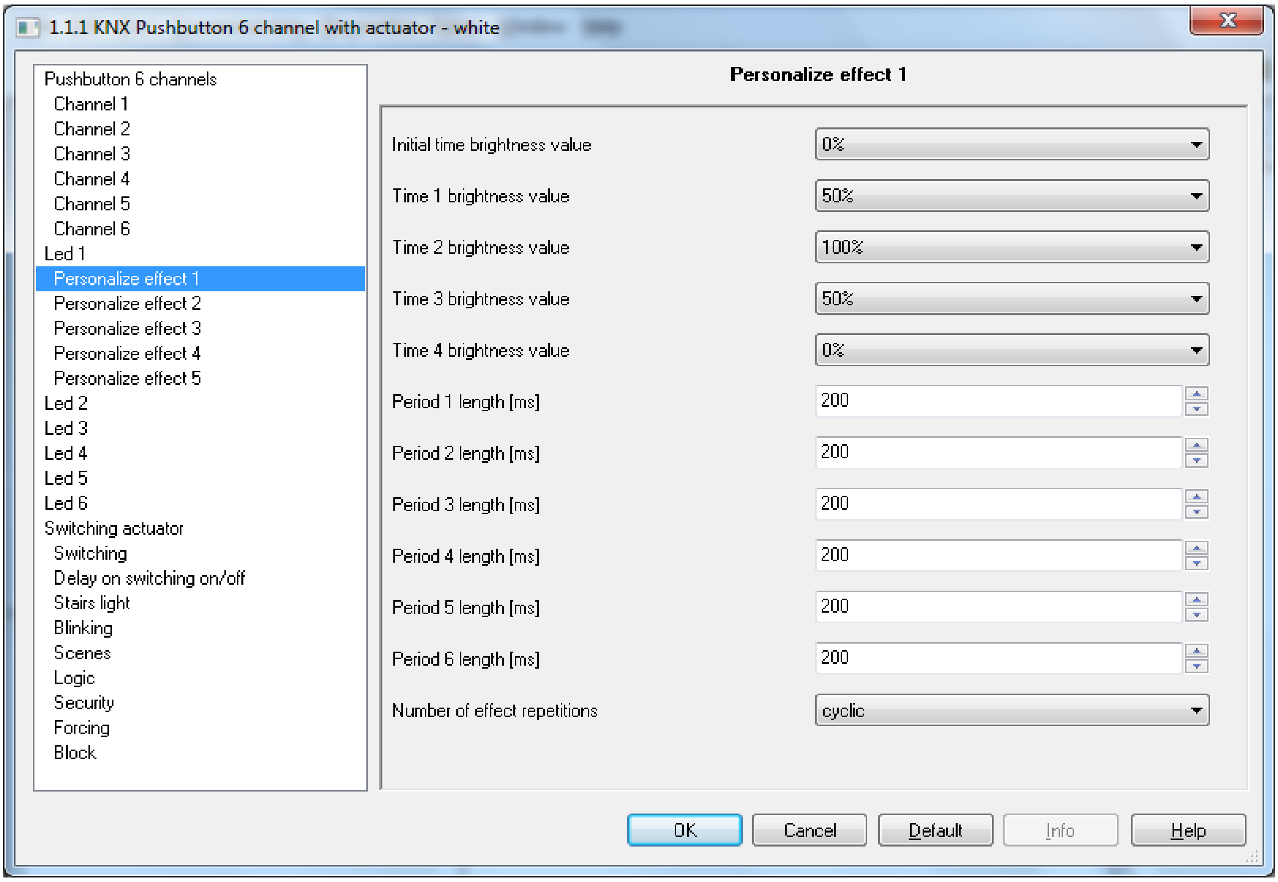
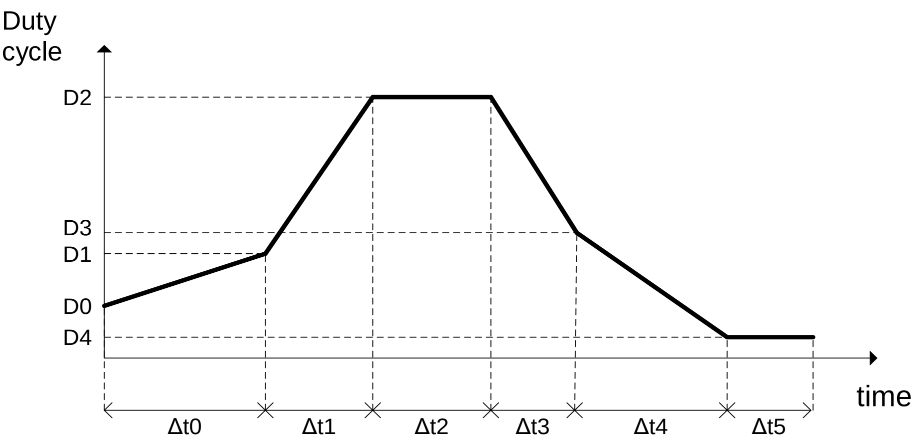


Fig. 13.2

Using the figure shown below as a reference, all the variables that create the light effect can be set by the user in this configuration menu.



The available variables are:

- 6 parameters $\Delta_t 0$, $\Delta_t 1$, $\Delta_t 2$, $\Delta_t 3$, $\Delta_t 4$, $\Delta_t 5$ expressed in ms (0-65535) that define the duration of the brightness control ramp between the value $\Delta_t(n)$ and $\Delta_t(n+1)$ or, in this case $\Delta_t 2$, the interval for which the brightness D2 will be maintained
- 5 parameters D0, D1, D2, D3, D4 (0-255) that define LED brightness values (duty-cycle). The values that can be set for these parameters will be displayed to the user as a percentage value between 0% and 100%, according to the proportion D(n)= Parameter*255/100
- 1 parameter that defines the number of cycles for **repeating** the effect (1 .. 254);

The parameters used to define the brightness values to reproduce are **"Initial time brightness value"** (D0), **"Time 1 brightness value"** (D1), **"Time 2 brightness value"** (D2), **"Time 3 brightness value"** (D3) and **"Time 4 brightness value"** (D4), which may be assigned the following values:

- from 0% to 100%, **0 % (default value D0-D4)**, **50% (default value D1-D3)** and **100% (default value D2)**

The parameters used to define the duration of the control ramp between a brightness value and the next are **"Period 1 length [ms]"** ($\Delta_t 0$), **"Period 2 length [ms]"** ($\Delta_t 1$), **"Period 3 length [ms]"** ($\Delta_t 2$), **"Period 4 length [ms]"** ($\Delta_t 3$), **"Period 5 length [ms]"** ($\Delta_t 4$) and **"Period 6 length [ms]"** ($\Delta_t 5$), which can assume the following values:

from 0 to 65535, **200 (default value)**

➢ *13.2.1 Number of effect repetitions*

This defines how many times the set light effect must be repeated when an activation command is received from the bus; The values that can be set are:

- 1, 2, .. 254, **cyclic (default value)**

14 *“Switching actuator”* menu

The **Switching actuator** menu contains parameters that define the behaviour of the relay with the change-over contact on board the device beyond the specific functions implemented by the actuator channel (paragraph 14.1). The priority of the functions implemented by the on/off actuator channel (paragraphs 14.2 to 14.10) is shown in the following table:

Function	Priority	
On/off switching	1	Low
Timed switching	1	
Delayed switching	1	
Blinking	1	
Scene	1	
Relay status after security	1	
Relay status after forcing	1	
Relay status on block deactivation	1	
Logic function (if used for enabling of commands)	2	
Relay status after bus voltage recovery	3	
Safety status when BUS voltage is reset	4	
Forced positioning status on bus voltage recovery	4	
Safety	5	
Forcing	6	
Lock	7	
Actuator local command (if “local command button” function)	8	
Block function on downloading ("Block on download function)/bus voltage recovery (if value = active)	9	
Relay status at bus voltage failure	10	High

The basic structure of the menu is as follows (fig. 14.1):

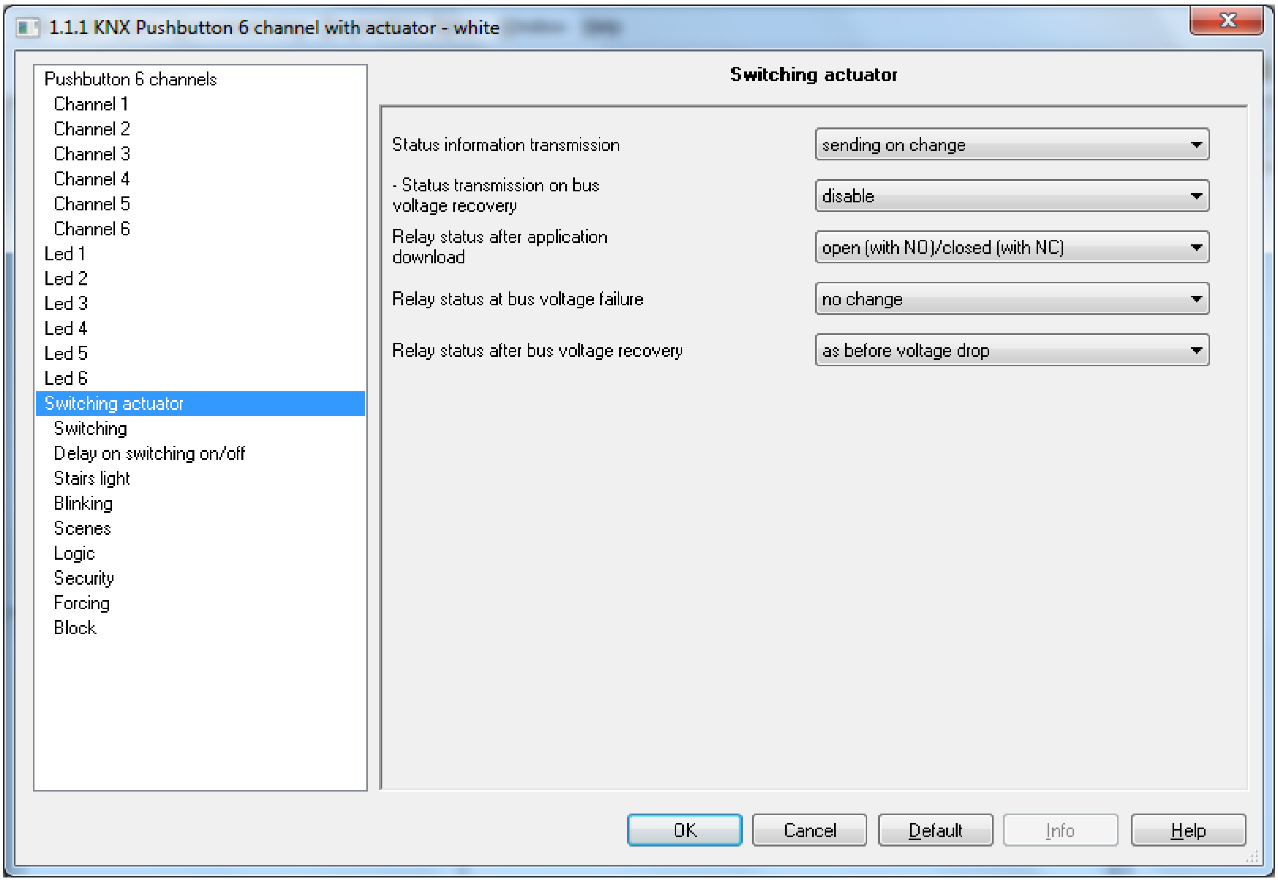


Fig. 14.1

14.1 Parameters

➢ *14.1.1 Status information transmission*

This parameter is used to enable the communication object for transmitting the status of the on board relay and as a result the connected load.
The parameter may have the following values:

- disabled
- on demand only
- **sending on change (default value)**

Selecting any value other than **disabled**, displays the communication object ***Actuator status*** that allows transmitting the status information on the bus concerning the load connected to the device.
If the status signalling takes place **sending on change** the communication object is sent spontaneously when the status switches from ON to OFF or vice versa; If the set value is **on demand only**, the status will never be sent spontaneously by the device, but only after a status reading request is received from the bus will the device send a response telegram with the current status of the load.
The communication object assumes the value 1 = ON when the NO contact (normally open) is closed and the NC contact (normally closed) is open, and the value 0 = OFF when the NO contact is open and the NC contact is closed.

➢ *14.1.2 Status transmission on bus voltage recovery*

Selecting the value **sending on change** at 14.1.1, displays this parameter that allows enabling the transmission of load status information on recovery of the bus power supply voltage. The parameter may have the following values:

- **Disabled (default value)**
- enable

By selecting the value **enable**, the status signalling is sent after a certain period of time has passed after switching on, as defined by the parameter **“Delay time between switching on and first transmission”** in the **6-channel push-button panel** menu (see paragraph 3.1.3).

➢ *14.1.3 Relay status after application download*

This parameter is used to set the status that the relay change-over contact must assume after the application parameters are downloaded from the ETC software, which can assume the following values:

- **open (with NO)/closed (with NC) (default value)**
- closed (with NO)/open (with NC)

➢ *14.1.4 Relay status at bus voltage failure*

This parameter is used to set the status of the relay change-over contact after the bus voltage drop, which can assume the following values:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)
- **no change (default value):**

➢ *14.1.5 Relay status after bus voltage recovery*

This parameter is used to set the status of the relay change-over contact on bus voltage recovery, which can assume the following values:

- open (with NO)/closed (with NC)

- closed (with NO)/open (with NC)
- **as before voltage drop (default value)**

14.2 "Switching" function

One of the actuator operating modes is on/off switching, which involves switching the relay status according to the received commands; from the bus, it is possible to control this operating mode via the communication object ***Actuator switching***.
This function has the same priority of the activation/deactivation delay, stairs light and flashing functions; this means that when one of the functions is activated while another is already active, it is executed, ending the previously active one.
The basic structure of the menu is as follows (fig. 14.2):

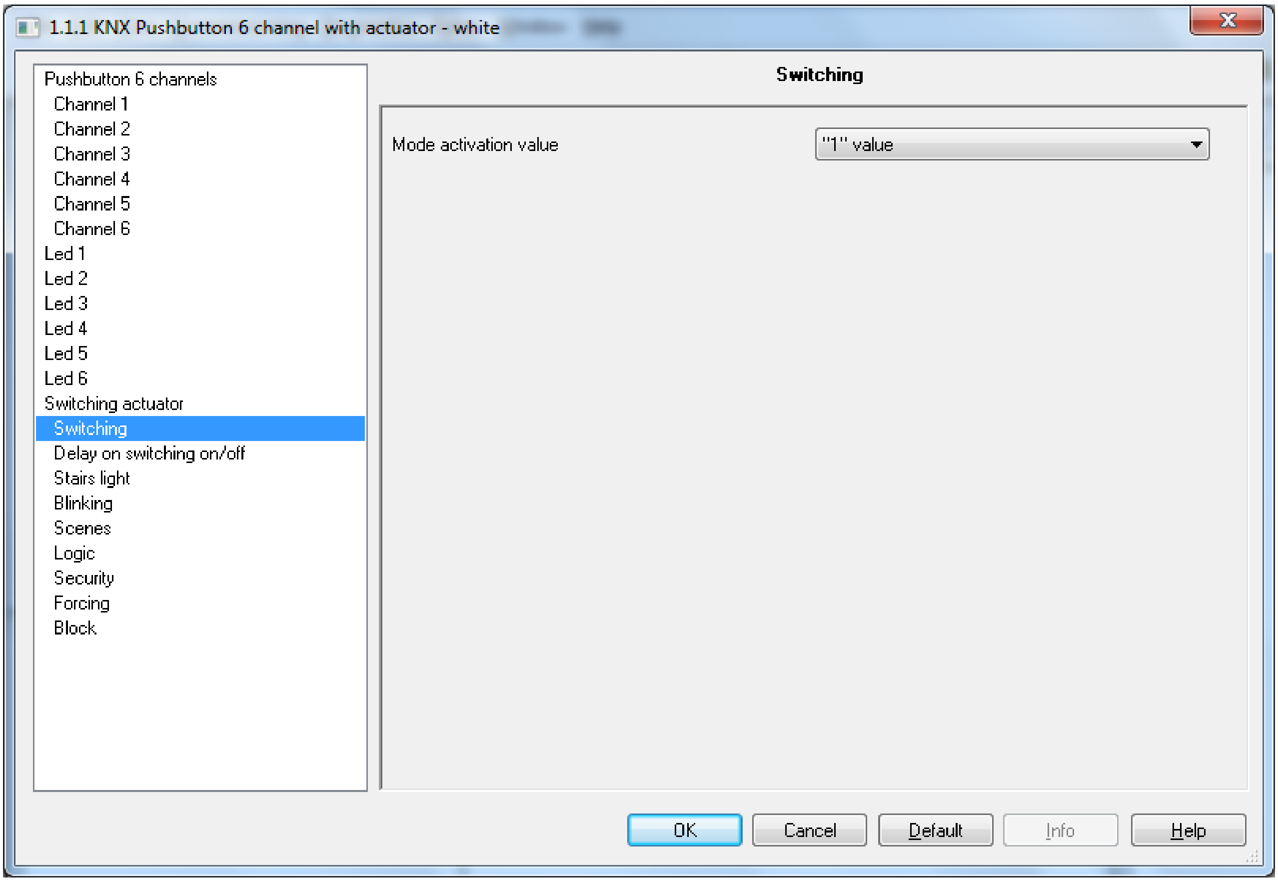


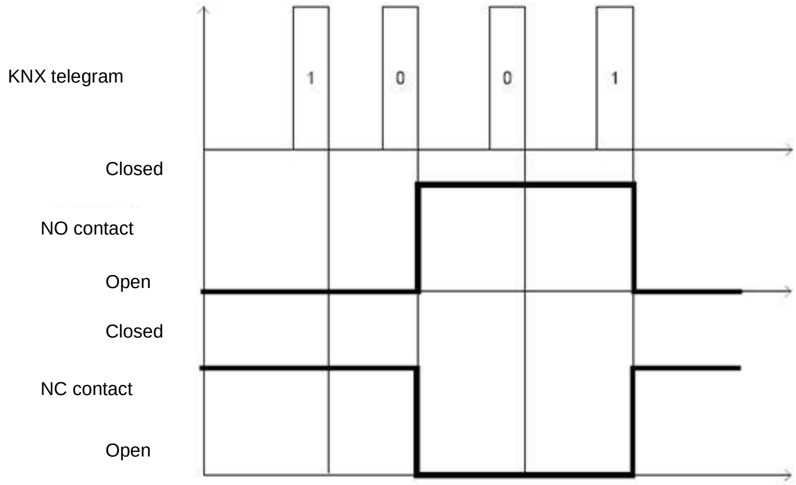
Fig. 14.2

➢ *14.2.1 Mode activation value*

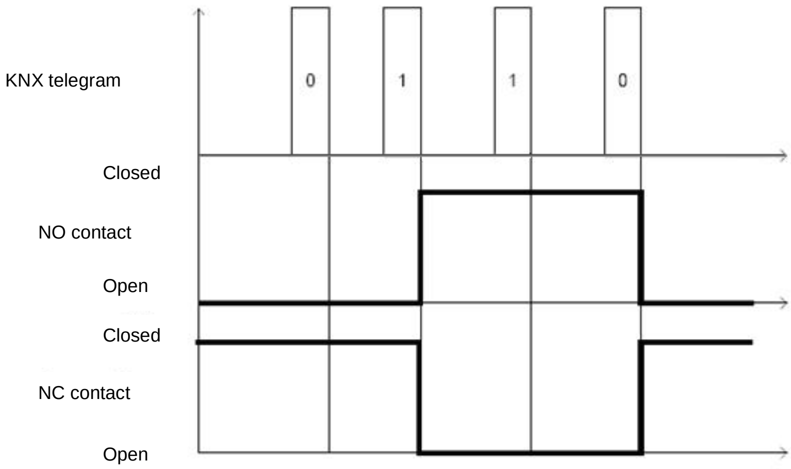
This determines which logical value received on the communication object ***Actuator switching*** switches the relay to the ON status (NO contact closed/NC open); the possible values are:

- value "0"
- **value "1" (default value)**

Selecting **value "0"**, when the device receives a telegram from the bus with a logical value equal to "0", it switches the relay with change-over contact to the status → NO contact closed/NC contact open; vice versa, when the logical value of "1" is received, the device switches the change-over contact to the conditions → NO contact open/NC contact closed. See figure below.



Selecting **value "1"**, when the device receives a telegram from the bus with a logical value equal to "1", it switches the relay with change-over contact to the status → NO contact closed/NC contact open; vice versa, when the logical value of "0" is received, the device switches the change-over contact to the conditions → NO contact open/NC contact closed. See figure below.



14.3 "Delay on switching on/off" function

An operating mode of the actuator is on/off switching with activation/deactivation delay, which switches the relay status depending on the received commands, creating a delay between the moment of receiving the command and the effective moment in which the relay is switched; from the bus, it is possible to control this operating mode via the communication object ***Actuator delayed switching***. This function has the same priority of the on/off switching, stairs light and blinking functions; this means that when one of the functions is activated while another is already active, it is executed, ending the previously active one.
The basic structure of the menu is as follows (fig. 14.3):

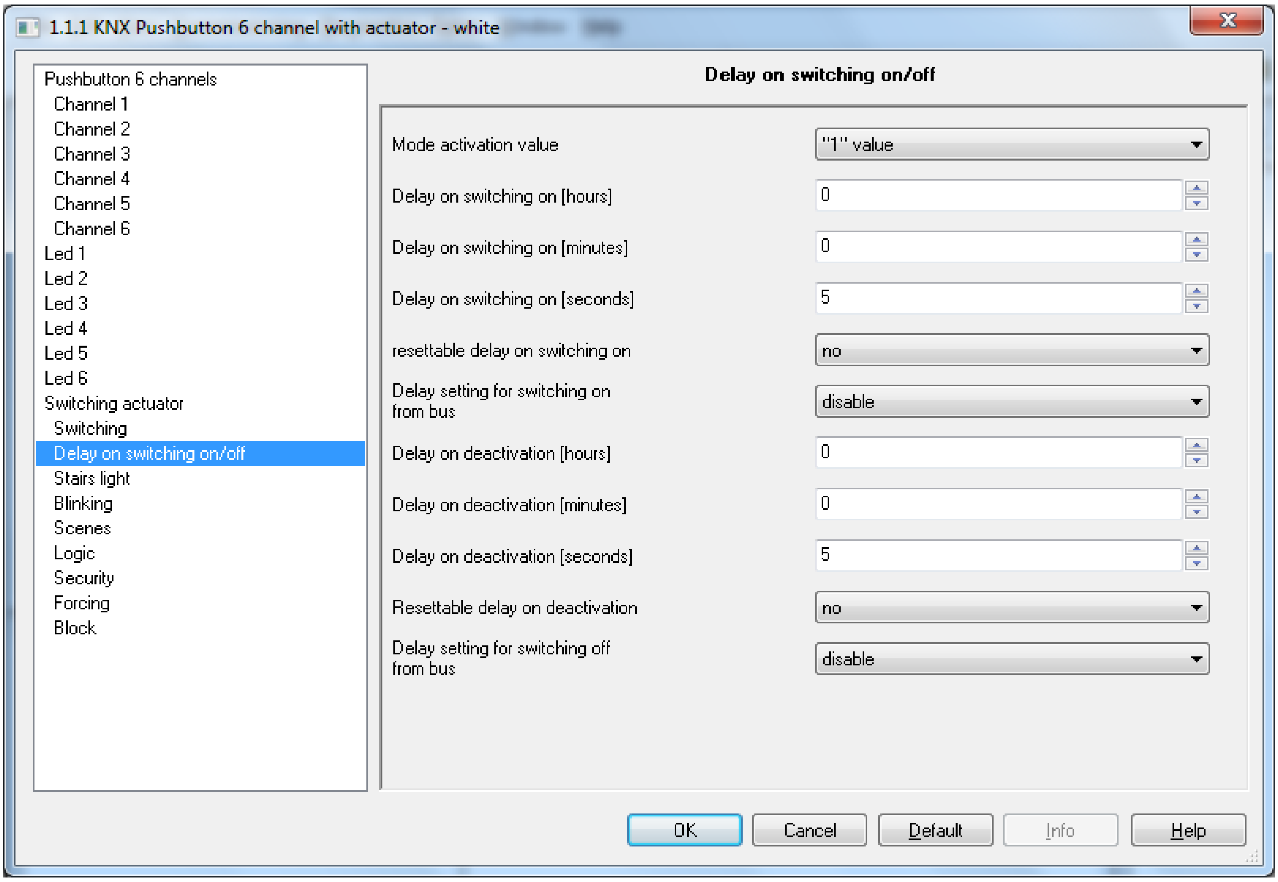


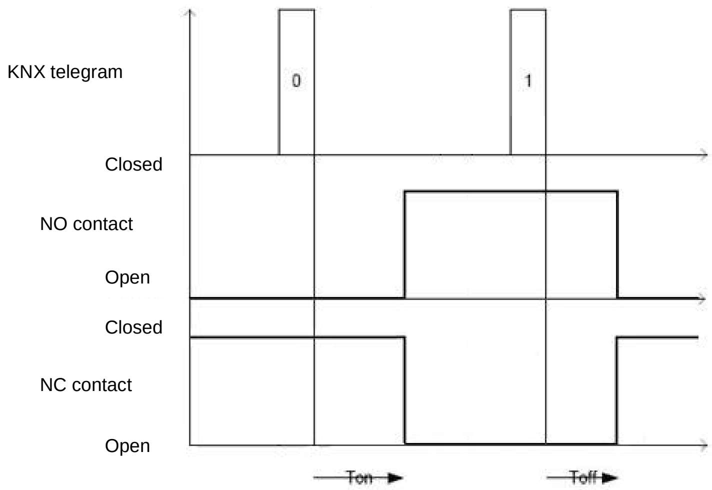
Fig. 14.3

➢ *14.3.1 Mode activation value*

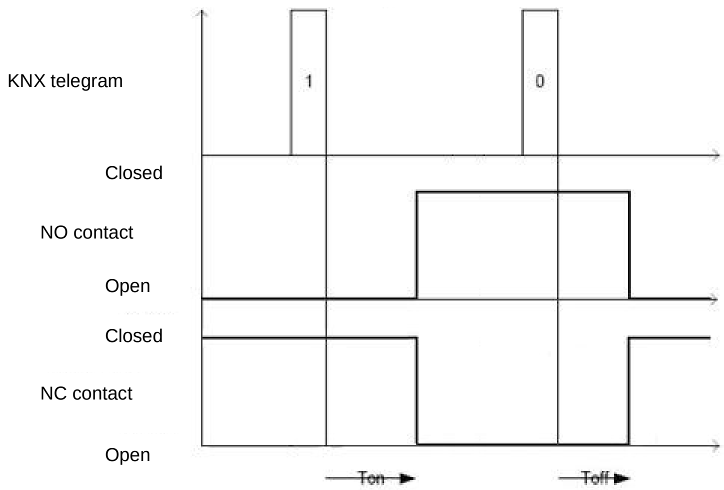
This determines which logical value received on the communication object ***Actuator delayed switching*** switches the relay to the ON status (NO contact closed/NC open); the possible values are:

- value "0"
- **value "1" (default value)**

Selecting **value "0"**, when the device receives a telegram from the bus with a logical value equal to "0", after the set activation delay time (Ton) has passed, it switches the relay with change-over contact to the status → NO contact closed/NC contact open; vice versa, when the logical value of "1" is received, after the set deactivation delay time (Toff) has passed, the device switches the change-over contact to the conditions → NO contact open/NC contact closed. See figure below.



Selecting **value "1"**, when the device receives a telegram from the bus with a logical value equal to "1", after the set activation delay time (Ton) has passed, it switches the relay with change-over contact to the status → NO contact closed/NC contact open; vice versa, when the logical value of "0" is received, after the set deactivation delay time (Toff) has passed, the device switches the change-over contact to the conditions → NO contact open/NC contact closed. See figure below.



➢ *14.3.2 Delay on switching on [hours]*
This is used to set the first of the three values (hours) that make up the activation delay time (hours, minutes, seconds); the settable values range from **0 (default value)** to 24.

➢ *14.3.3 Delay on switching on [minutes]*
This is used to set the second of the three values (minutes) that make up the activation delay time (hours, minutes, seconds); the settable values range from **0 (default value)** to 59.

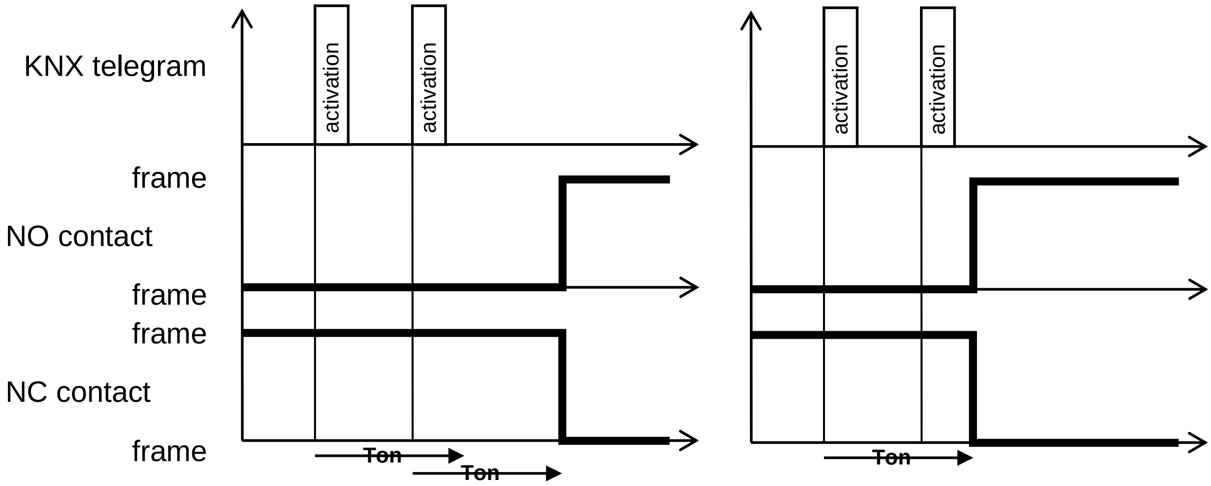
➢ *14.3.4 Delay on switching on [seconds]*
This is used to set the last of the three values (seconds) that make up the activation delay time (hours, minutes, seconds); the settable values range from 0 to 59, **5** is the default value.

➢ *14.3.5 Resettable delay on switching on*
This is used to enable the reset of the activation delay time each time that a delayed activation bus telegram is received with the delay count already active; The values that can be set are:

- **No (default value)**

- yes

by selecting **yes**, if a new delayed activation telegram is received during the activation delay count, the counter is reinitialized; otherwise, the count continues without changes. See figure below (to the left with reset enabled, to the right without reset).



➢ *14.3.6 Delay setting for switching on from bus*

This is used to enable the communication object through which a new activation delay value is received, which overwrites the one configured in ETS; The values that can be set are:

- **disable (default value)**
- enable

Selecting the value **enable**, displays the communication object ***Delay on actuator switching on*** which is used to receive the value of the delay upon activation from the bus.
If the new value is received while an activation delay time count is already in progress, it will become operative when the subsequent activation command is received.

➢ *14.3.7 Delay on deactivation [hours]*

This is used to set the first of the three values (hours) that make up the deactivation delay time (hours, minutes, seconds); the settable values range from **0 (default value)** to 24.

➢ *14.3.8 Delay on deactivation [minutes]*

This is used to set the second of the three values (minutes) that make up the deactivation delay time (hours, minutes, seconds); the settable values range from **0 (default value)** to 59.

➢ *14.3.9 Delay on deactivation [seconds]*

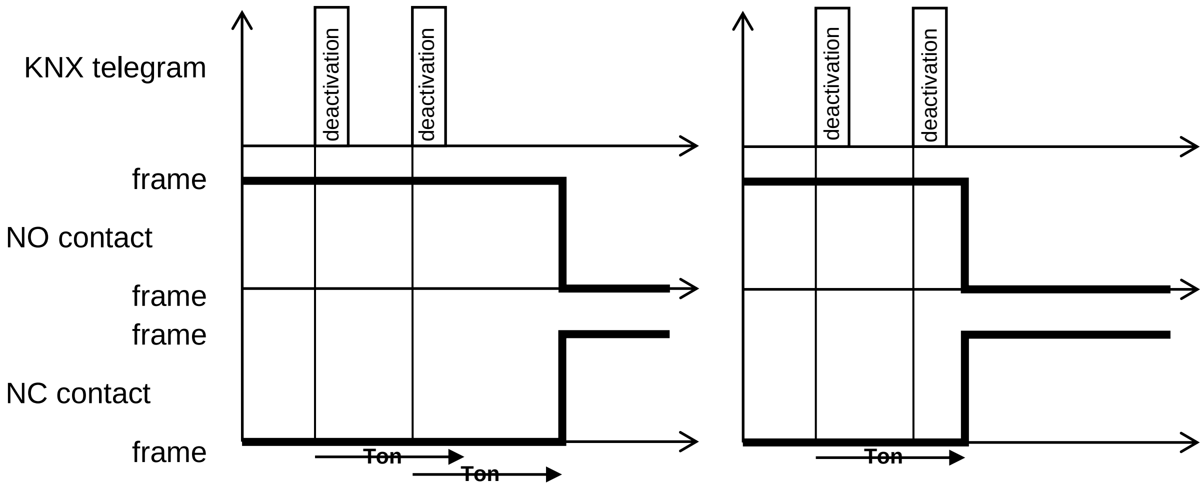
This is used to set the last of the three values (seconds) that make up the deactivation delay time (hours, minutes, seconds); the settable values range from 0 to 59, **5** is the default value.

➢ *14.3.10 Resettable delay on deactivation*

This is used to enable the reset of the deactivation delay time each time that a delayed activation bus telegram is received with the delay count already active; The values that can be set are:

- **no (default value)**
- yes

By selecting **yes**, if a new delayed deactivation telegram is received during the deactivation delay count, the counter is reinitialized; otherwise, the count continues without changes. See figure below (to the left with reset enabled, to the right without reset).



➢ *14.3.11 Delay setting for switching off from bus*

This is used to enable the communication object through which a new deactivation delay value is received, which overwrites the one configured in ETS; The values that can be set are:

- **disable (default value)**
- enable

Selecting the value **enable**, displays the communication object ***Actuator deactivation delay*** which is used to receive the value of the delay upon deactivation from the bus.
If the new value is received while a deactivation delay time count is already in progress, it will become operative when the subsequent deactivation command is received.

14.4 “Stairs light” function

One of the actuator operating modes is timed activation or stairs light function, which involves activating the load for a certain period of time and then deactivating it automatically without receiving a command. Furthermore, it is possible to enter a certain delay between the moment the timed start command is received and the effective instant in which the relay is switched; from the bus, it is possible to control this operating mode via the communication object ***Actuator timed switching***. This function has the same priority as the on/off switching, activation/deactivation delay and flashing functions; this means that when one of the functions is activated while another is already active, it is executed, ending the previously active one.

The basic structure of the menu is as follows (fig. 14.4):

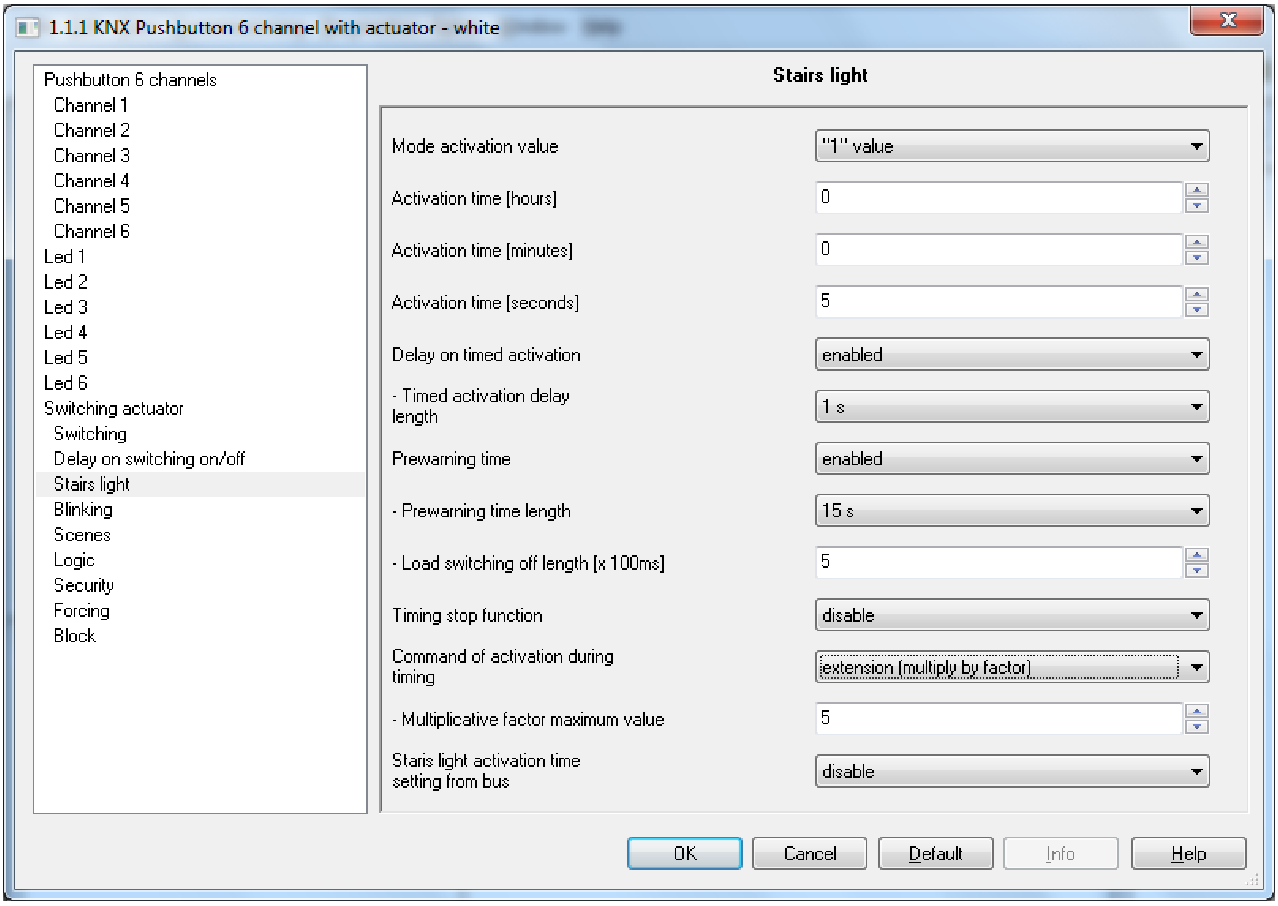


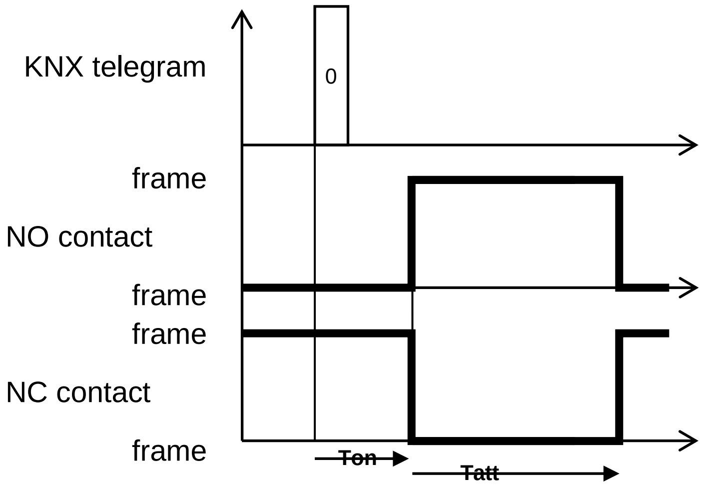
Fig. 14.4

➢ *14.4.1 Mode activation value*

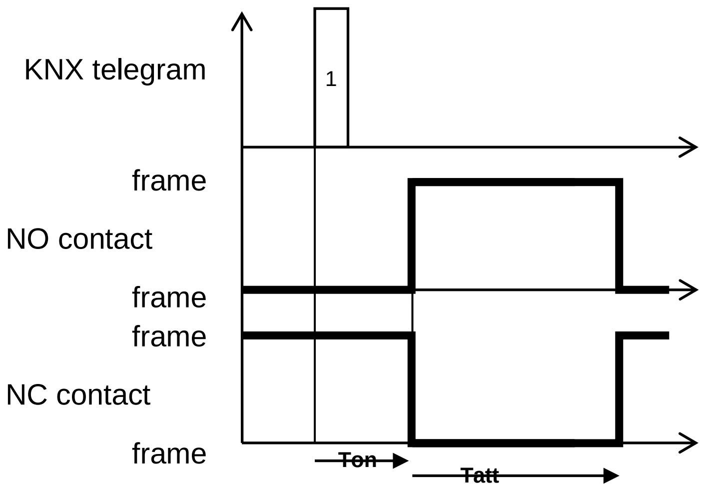
This determines which logical value received on the communication object ***Actuator timed switching*** switches the relay to the ON status (NO contact closed/NC open) and activates the timing; the possible values are:

- value “0”
- **value “1” (default value)**

Selecting **value “0”**, when the device receives a telegram from the bus with a logical value equal to “0”, after the set activation delay time (Ton) has passed, it switches the relay with change-over contact to the status → NO contact closed/NC contact open and starts the activation time count; See figure below.



Selecting **value “1”**, when the device receives a telegram from the bus with a logical value equal to “1”, after the set activation delay time (Ton) has passed, it switches the relay with change-over contact to the status → NO contact closed/NC contact open; See figure below.



➢ *14.4.2 Activation time [hours]*

This is used to set the first of three values (hours) that make up the load activation time (Tatt); the settable values range from **0 (default value)** to 24.

➢ *14.4.3 Activation time [minutes]*

This is used to set the second of three values (minutes) that make up the load activation time (Tatt); the settable values range from **0 (default value)** to 59.

➢ *14.4.4 Activation time [seconds]*

This is used to set the last of the three values (seconds) that make up the load activation time (Tatt); the settable values range from 0 to 59, **5** is the default value.

➢ *14.4.5 Delay on time activation*

This is used to enter a delay between the moment in which the communication object ***Actuator timed switching*** is received and the moment in which the command is actually executed (NO contact closed/NC contact open); the possible values are:

- **disabled (default value)**
- enabled

If the delay is **enabled**, the parameter **Px "Timed activation delay length"** will be displayed, which is used to set the value of the delay in seconds. The parameter may have the following values:

- **1 s (default value)**, 2 s, 3 s, 5 s, 10 s, 15 s, 20 s, 30 s, 45 s, 1 min, 1 min 15 s, 1 min 30 s, 2 min, 2 min 30 s, 3 min, 5 min, 15 min, 20 min, 30 min, 1 h, 2 h, 3 h, 5 h, 12 h, 24 h.

The activation delay cannot be reset.

➢ *14.4.6 Prewarning time*

This parameter can be used to enable signalling when the load will soon be automatically switched off, by deactivating and reactivating the load for a brief period of time (blink); The prewarning time is applied after the expiration of the activation time. The parameter may assume the following values:

- **disabled (default value)**
- enabled

Selecting **enabled**, displays the parameters **"Prewarning time length"** and **"Load switching off length [x 100ms]"**.

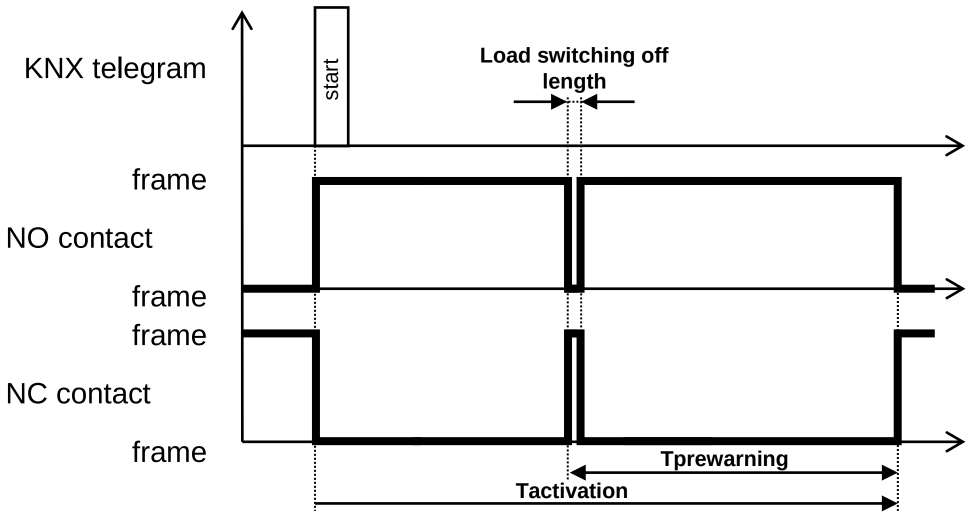
The **"Prewarning time length"** is used to set the time that passes between the signalling that the deactivation will take place soon and the deactivation itself of the load; the possible values are:

- **15 s (default value)**, 30 s, 1 min.

Parameter **Px "Load switching off length [x 100ms]"** is used to set the time interval during which the load is deactivated to perform the prewarning function; The values that can be set are:

- from **5 (default value)** to 15

The below figure shows the operating principle of the prewarning function.



➢ *14.4.7 Timing stop function*

Via this parameter **"Timing stop function"** it is possible to enable the possibility of ending the timed activation by means of the bus command on the communication object ***Actuator timed switching*** with the value opposite to the value set for **"Mode activation value"** analyzed previously in paragraph 14.4.1. T he possible values are:

- **Disabled (default value)**
- Enable

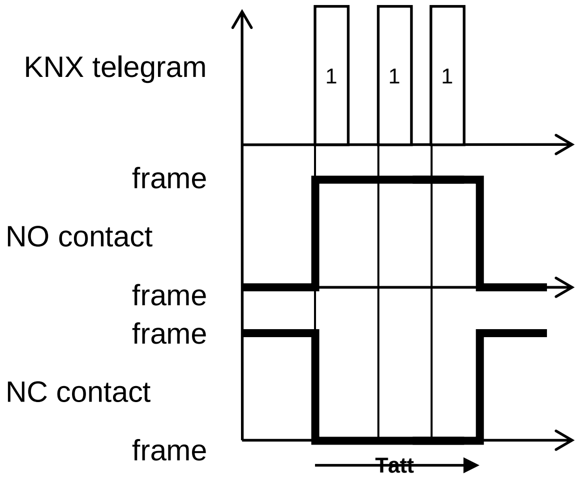
If the function is enabled, when the value opposite to the mode activation value is received, the device ends the timing and deactivates the load.

➢ *14.4.8 Command of activation during timing*

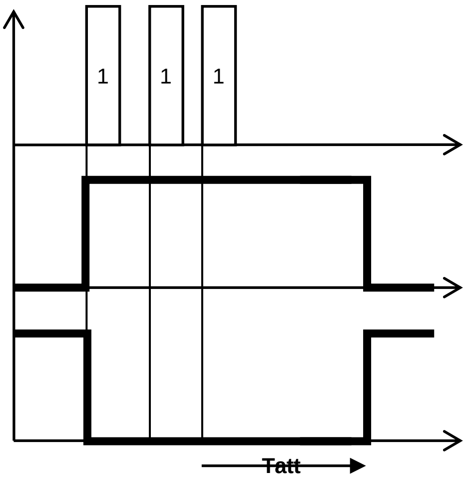
this is used to define the behaviour of the device if a timed activation command is received while timing is in progress; the possible values are:

- no effect
- **reset (default value)**
- extend (multiply by factor)

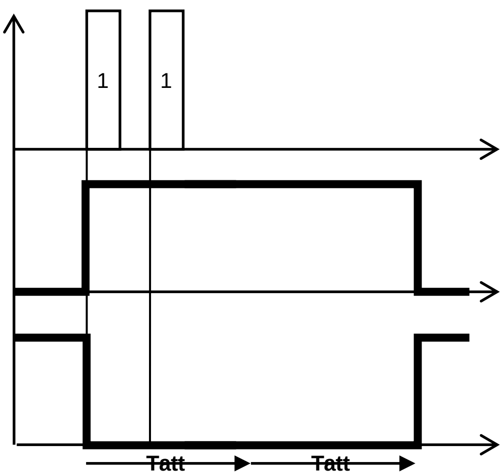
By selecting **no effect**, the subsequent commands are ignored; by selecting **reset**, each timed activation command received during the activation time count causes the count to reinitialize. By selecting **extension**, each received command results in an extension equal to the count activation time. The below figure shows an example of each of the three configurations.



no effect



reset



extension

If the value **extension** is selected, it is possible to set a maximum number of consecutive extensions of the activation time via the new displayed parameter **“Multiplicative factor maximum value”**. The parameter value can range between 2 and **5 (default value)**.

➢ *14.4.9 - Stairs light activation time setting from the bus*

This parameter displays the input communication object ***Actuator stairs light activation time*** which makes it possible to receive the value of the stairs light function activation time via the bus communication object; the possible values are:

- **Disabled (default value)**
- enable

As the activation time is between 0h:0min:1sec and 24h:59min:59sec, when the bus receives a value that lies outside this interval, the value set for the deactivation delay time is the limit value of the interval that is closest to the received value.

If a new activation time value is received, this becomes the new stairs light time, overwriting the old value, which will be deleted; if the new value is received while the timing is already active, it will become operative upon the subsequent activation of the timing.

14.5 “Blinking” function

One of the actuator operating modes is timed activation or stairs light function, which involves activating the load for a certain period of time and then deactivating it automatically without receiving a command. Furthermore, it is possible to enter a certain delay between the moment the timed start command is received and the effective instant in which the relay is switched; from the bus, it is possible to control this operating mode via the communication object ***Actuator timed switching***. This function has the same priority of the on/off switching, activation/deactivation delay and blinking functions; this means that when one of the functions is activated while another is already active, it is executed, ending the previously active one.
The basic structure of the menu is as follows (fig. 14.5):

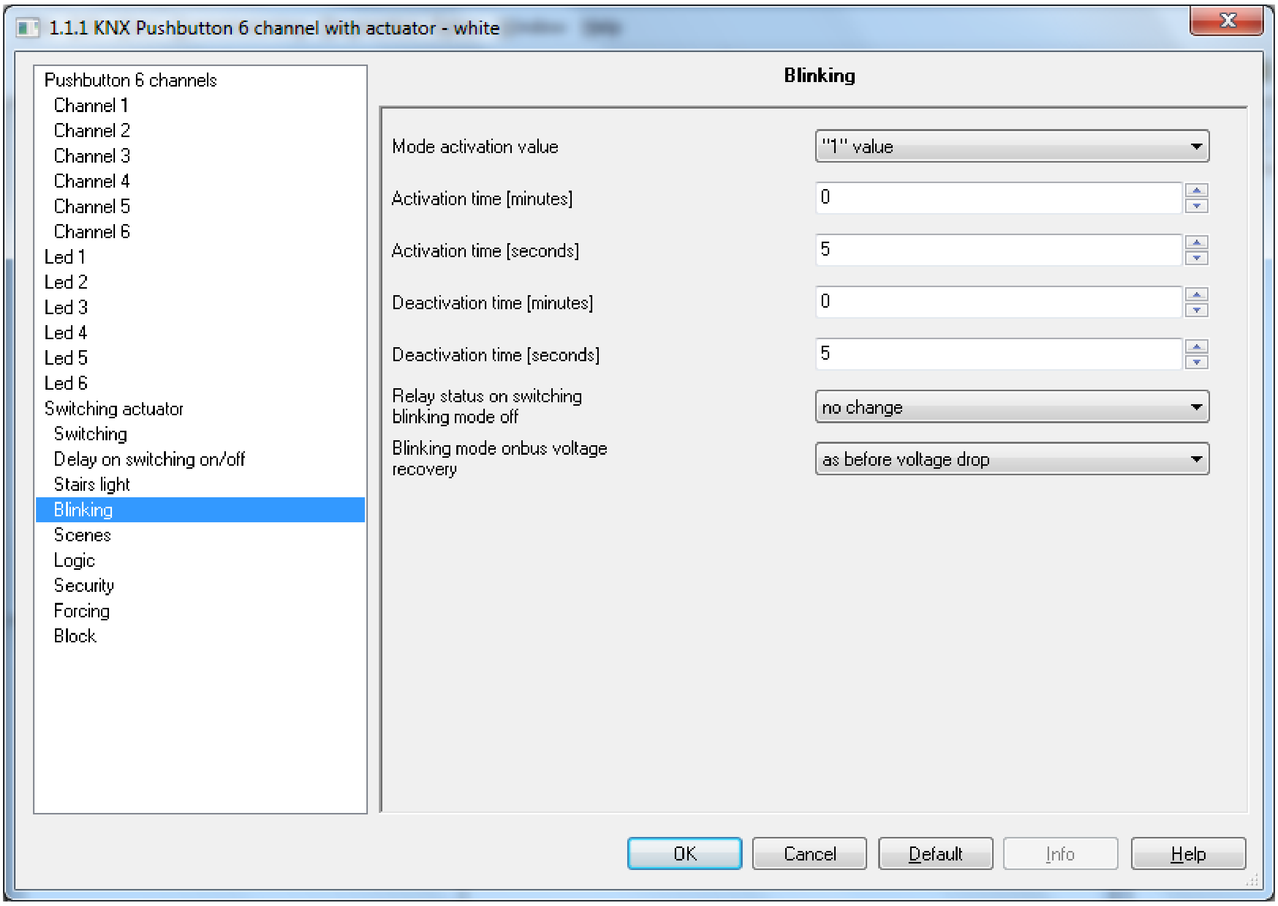


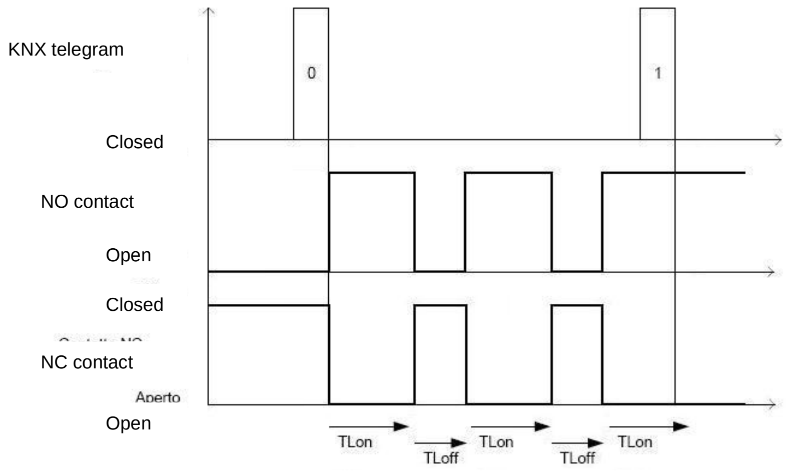
Fig. 14.5

➢ *14.5.1 Mode activation value*

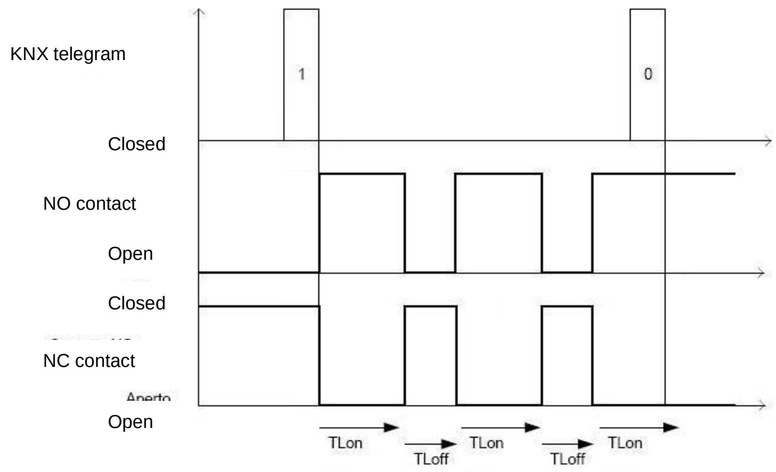
This determines which logical value received on the communication object ***Actuator blinking*** activates the load activation/deactivation process; the possible values are:

- value “0”
- **value “1” (default value)**

Selecting **value “0”**, when the device receives a telegram from the bus with a logical value equal to “0”, it switches the relay with change-over contact to the status → NO contact closed/NC contact open and starts the activation time count; at the end of the activation time, the device deactivates the load (NO contact open/NC contact closed) for a period of time equal to the deactivation time, and then reactivates the load and restarts the process. See figure below.



Selecting **value “1”**, when the device receives a telegram from the bus with a logical value equal to “1”, it switches the relay with change-over contact to the status → NO contact closed/NC contact open and starts the activation time count; at the end of the activation time, the device deactivates the load (NO contact open/NC contact closed) for a period of time equal to the deactivation time, and then reactivates the load and restarts the process. See figure below.



➢ *14.5.2 Activation time [minutes]*

This is used to set the first of two values (minutes) that make up the load activation time (TLon); the settable values range from **0 (default value)** to 59.

➢ *14.5.3 Activation time [seconds]*

This is used to set the last of two values (seconds) that make up the load activation time (TLon); the settable values range from 0 to 59, **5 (default value)**.

➢ *14.5.4 Deactivation time [minutes]*

This is used to set the first of two values (minutes) that make up the load deactivation time (TLoff); the settable values range from **0 (default value)** to 59.

➢ *14.5.5 Deactivation time [seconds]*

This is used to set the last of two values (seconds) that make up the load deactivation time (TLoff); the settable values range from 0 to 59, **5 (default value)**.

➢ *14.5.6 Relay status on switching blinking mode off*

It is possible to define the status of the relay change-over contact upon receipt of the blinking mode deactivation command via this parameter, which can assume the following values:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)
- **no change (default value):**

By selecting **no change**, the status of the contact remains the one assumed when the mode deactivation command was received.

➢ *14.5.7 Blinking mode on bus voltage recovery*

This is used to define the blinking mode status on bus voltage recovery; The values that can be set are:

- deactivated
- active
- **as before voltage drop (default value)**

By selecting **active**, if no function with a higher priority than the blinking mode is active, the device will start the blinking phase, ignoring the value set for **"Relay status after bus voltage recovery"** in the **Actuator** menu.

14.6 “Scenes” function

The scenes function is used to replicate a certain preset or previously memorized status upon receipt of the scene execution command; from the bus, it is possible to control this function with the communication object ***Actuator scene***. The device is able to memorize and execute 8 scenes; the basic structure of the menu is as follows (fig. 14.6):

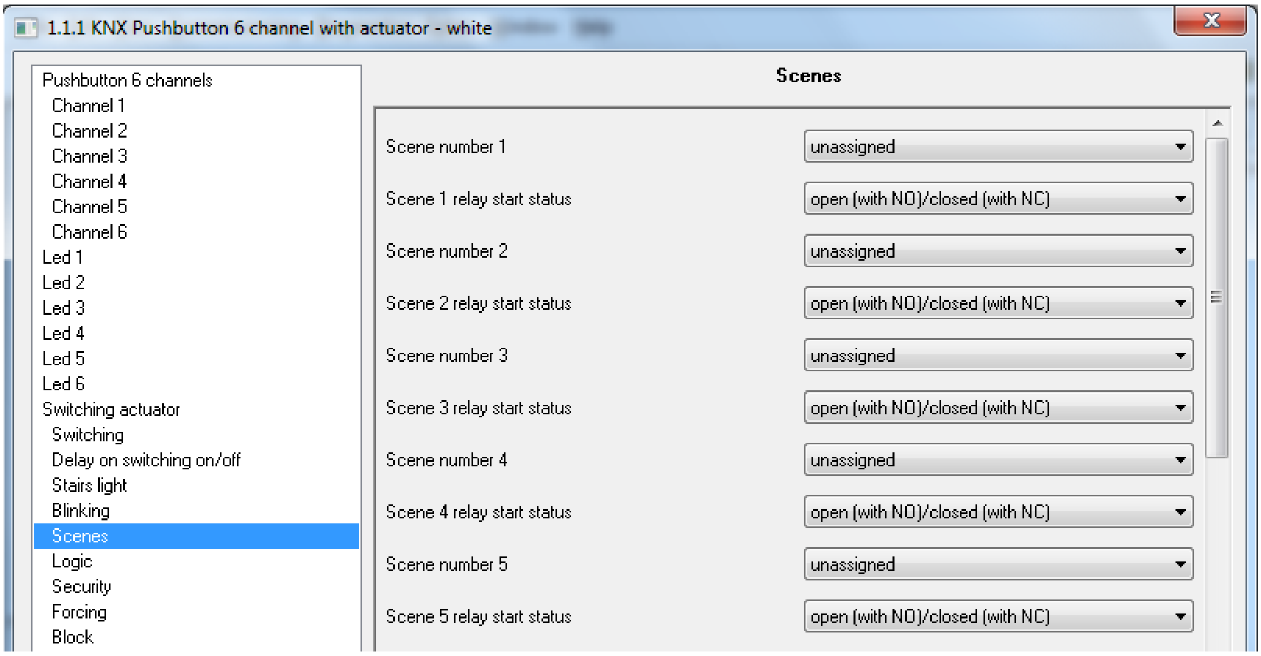


Fig. 14.6

➢ *14.6.1 “Scene number i”*

Using this parameter, where 1 ≤ i ≤ 8 it is possible to set the numerical value that is used to identify and therefore execute/memorize the i-th scene; the possible values are:

- **not assigned (unassigned) (default value)**
- 0, 1.. 63

➢ *14.6.2 Scene i relay start status*

These parameters where 1 ≤ i ≤ 8 are used to preset the status of the change-over contact that the device must replicate after receiving a telegram for the execution of the i-th scene; the possible values are:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)

➢ *14.6.3 Scenes storing enabling*

This parameter is used to enable/disable the scene storing possibility via the communication object ***Actuator scene***; the parameter may assume the following values:

- disabled
- **enabled (default value)**

Selecting **enable** displays the communication object ***Actuator scenes storing enabling*** which makes it possible to enable/disable via the bus the possibility to store scenes via the communication object ***Actuator scene***.

14.7 *“Logic” function*

It is possible to subordinate load activation/deactivation according to the result of the logic operations, which have dedicated communication objects as inputs; it is possible to enable the logic function via the parameter **”Logic function”** which can assume the following values:

- **Disabled (default value)**
- enabled

If the function is enabled, it is possible to set the number of logic inputs via the parameter **”Logic inputs number”** which can assume the following values:

- **1** (default value), 2, 3, 4

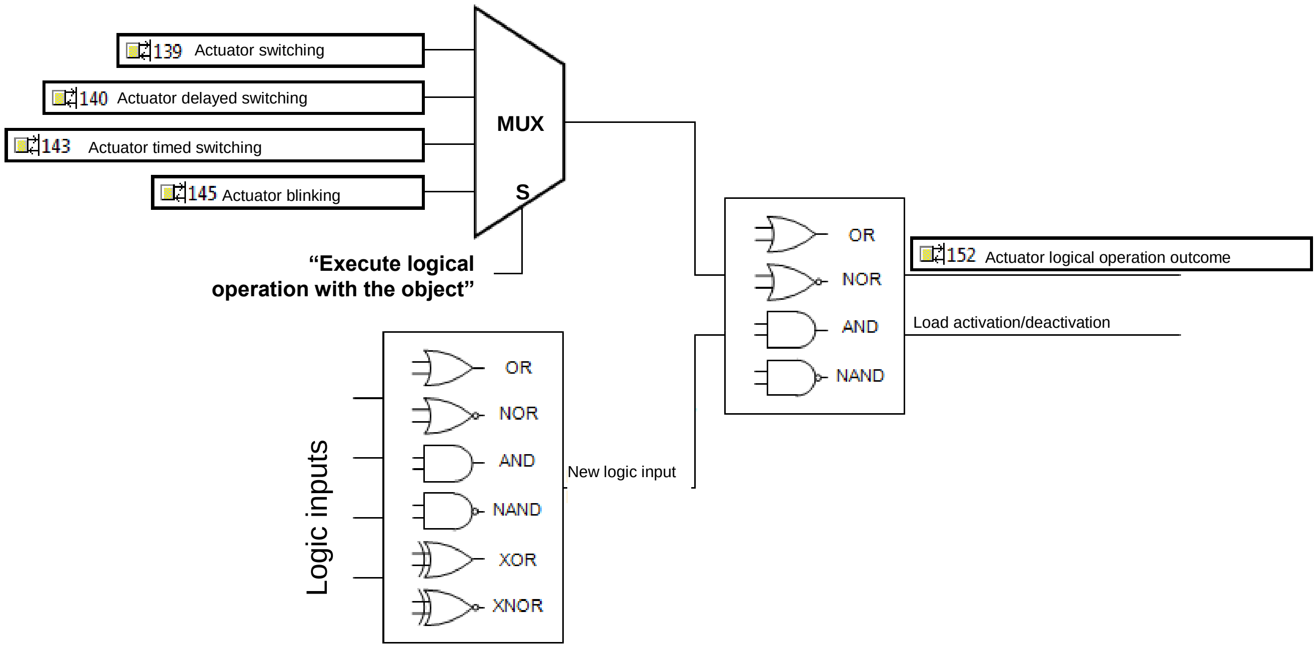
Depending on the selected value, the communication objects, ***Logic input 1 actuator***, ***Logic input 2 actuator***, ***Logic input 3 actuator*** and ***Logic input 4 actuator*** will be displayed.

If the set value is not **1**, it is possible to set the logic operation to be executed between the logic inputs. The operation is selected using the parameter **“Operation between logic inputs”** which can assume the following values:

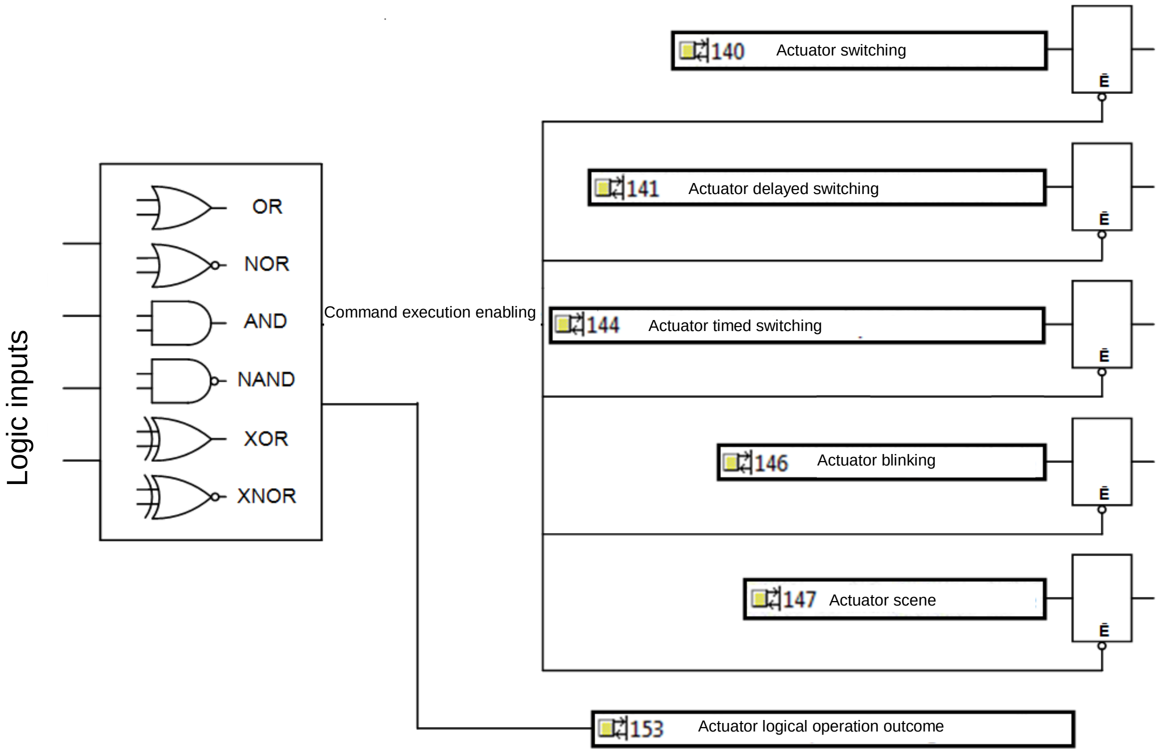
- **AND**	**0 (default value)**
- OR	1
- NAND	2
- NOR	3
- XOR	4
- XNOR	5

The outcome of the operations between logic inputs (or the value of the individual logic input if only one logic input was set) can be used as follows:

1. As input for an additional logic operation that is executed, selecting one of the following objects ***Actuator switching***, ***Actuator timed switching***, ***Actuator delayed switching*** and ***Blinking***



2. Used for enabling the execution of the commands received from the bus on the objects ***Actuator switching***, ***Actuator timed switching***, ***Actuator delayed switching***, ***Blinking*** and ***Scene***.



➢ *14.7.1 The logic input value stands for*

This is used to select the function of the outcome of the operation between logic inputs, which in the case of a single logic input is replaced by the parameter **"The logic input value stands for"**; these parameters can assume the following values:

- **new logic input (default value)**
- bus commands execution enabling

If the value **new logic input** was selected (case 1), it is possible to define which object should be used to execute the new logic operation via the parameter **"Execute logical operation with the object"** and the logic operation to execute with the selected object via the parameter **"Logical operation to execute"**.

The basic structure of the menu is as follows (fig. 14.7a):

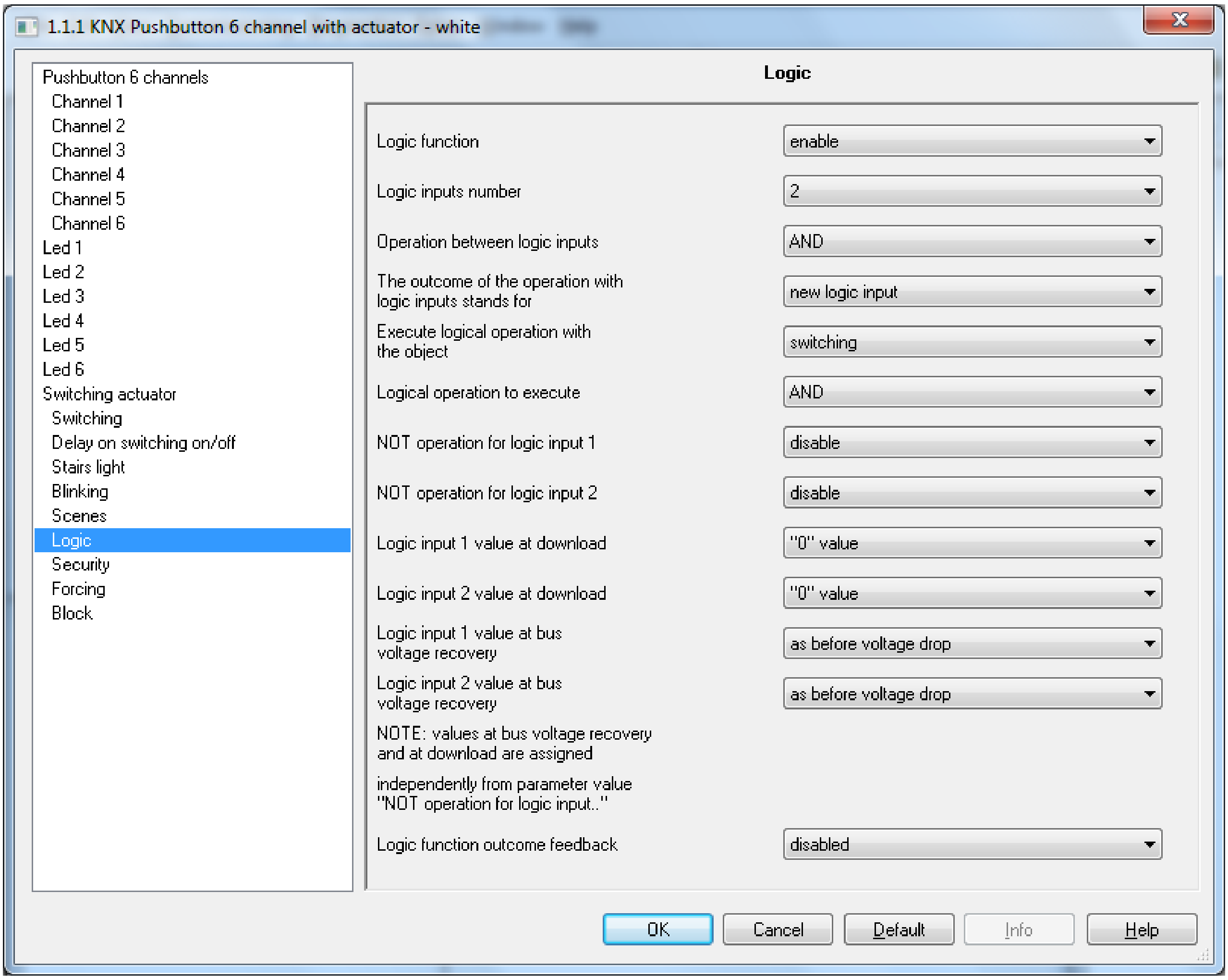


Fig. 14.7a

If the value **bus commands execution enabling** is selected (case 2), a series of parameters appear that are used to set which commands received from the bus require enabling to be executed; these parameters are **“Switching (on/off) commands”**, **“Delayed switching commands”**, **“Timed activation commands”**, **“Blinking switching on /off commands”** and **“Scene commands”**, which may be assigned the following values:

- **independent from logic function (default value)**
- enabled by logic function

The commands enabled by the logic function are only executed if the outcome of the logic operation is true. If the outcome of the logic operation changes from false to true, the commands received after the status change will be executed. The commands received when the outcome of the logic function is false are ignored.

The basic structure of the menu is as follows (fig. 14.7b):

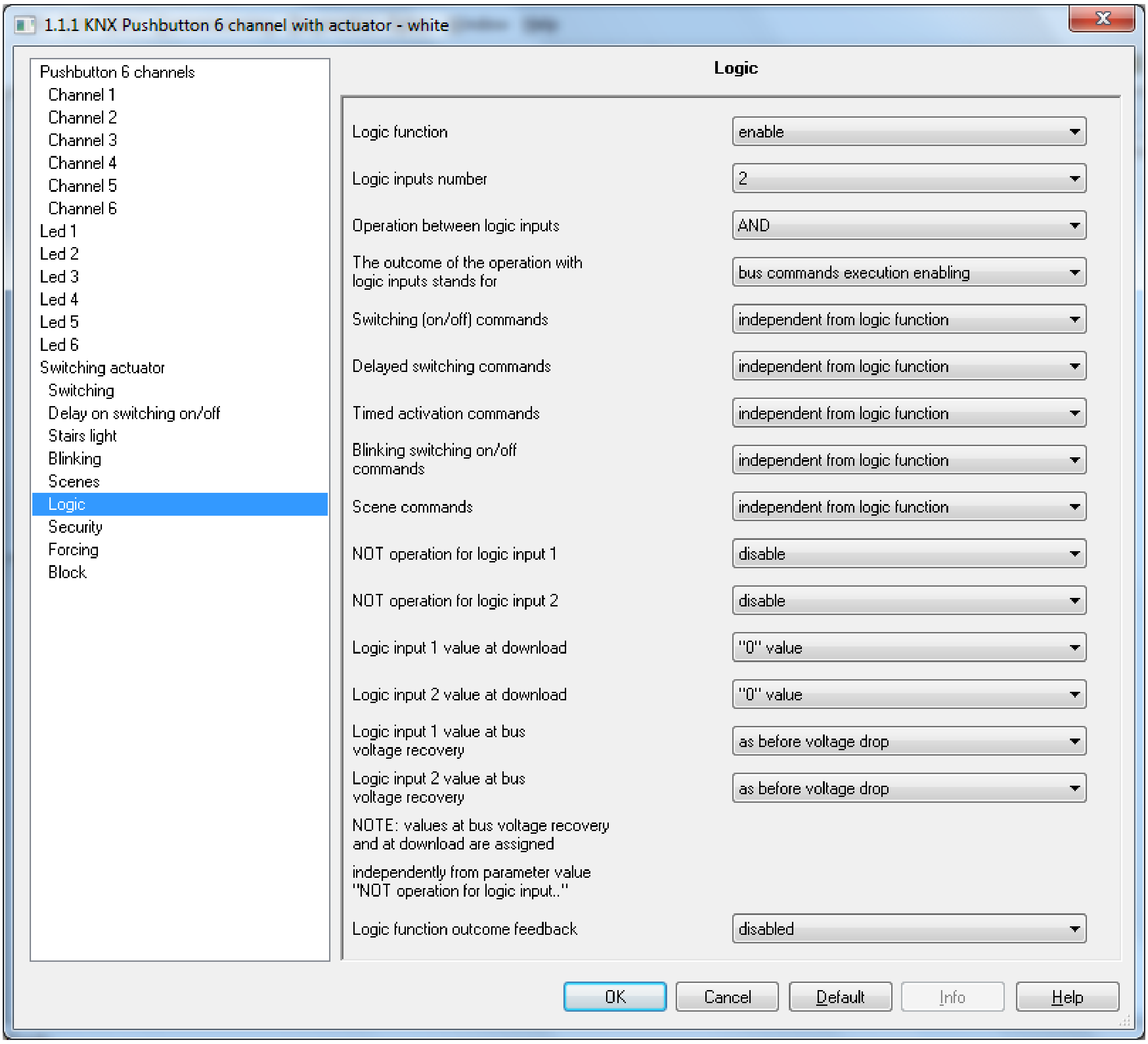


Fig. 14.7b

➢ *14.7.2 Execute logical operation with the object*

If the value **new logic input** is selected (case 1), as described in the previous paragraph, the following values become available:

- **Switching (default value)**
- Delayed switching
- Timed switching
- Flashing

➢ *14.7.3 Logical operation to execute*

If the value **new logic input** is selected (case 1), as described in paragraph 14.7.1, the following values become available:

- **AND (default value)**
- OR
- NAND
- NOR

➢ *14.7.4 NOT operation for logic input x*

If the value **enabled from logic function** is selected (case 2), as described in paragraph 14.7.1, the following values become available which can be used to negate the value received from the bus on the communication objects associated with the logic inputs:

- **disabled** (default value)
- active

➢ *14.7.5 Logic input x value at download*

If the value **enabled from logic function** is selected (case 2), as described in paragraph 14.7.1, the following values become available which can be used to reset the value of the logic inputs on ETC downloading:

- **value “0” (default value)**
- value “1”

➢ *14.7.6 Logic input x value at bus voltage recovery*

If the value **enabled from logic function** is selected (case 2), as described in paragraph 14.7.1, the following values become available which can be used to set the value of the logic inputs in the case of voltage recovery:

- value “0”
- value “1”
- **as before voltage drop (default value)**

NOTE : The values on bus voltage recovery and downloading are assigned to logic objects independently of the value of the parameters **“NOT operation for logic input i”** (1<i<4).

➢ *14.7.7 Logic function outcome feedback*

If the value **enabled from logic function** is selected (case 2), as described in paragraph 14.7.1, the following values become available for enabling the sending of the logic function outcome on the bus and if this signalling must always be sent on change of an input or only in the case if the outcome of the logic function changes:

- **disabled (default value)**
- only if outcome changes
- even if outcome doesn’t change

Setting a value other than **disabled** displays the output communication object ***Actuator logical operation outcome***.

The value transmitted on the bus is the outcome of the logic operation of the logical and object inputs selected with parameter **“Execute logical operation with the object”** if the parameter **“The logic input value stands for”** has the value **new logic input** or the outcome of the operation between logic inputs if the parameter has the value **bus commands execution enabling**.

14.8 “Security" function

The security function allows the device to function under normal conditions until certain set conditions occur (no periodic reception, reception of particular data from the bus), after which the actuator forces the status of the relay to a specific condition; to deactivate the security function, the normal operation conditions must be reset. Any command that is received (excluding the block activation and forcing activation command) during a period when the security is activated will not be executed as it has priority over any other bus command, with the exception of the block and forcing functions.
The communication object used to monitor the operating conditions is the object ***Actuator security***.
The basic structure of the menu is as follows (fig. 14.8):

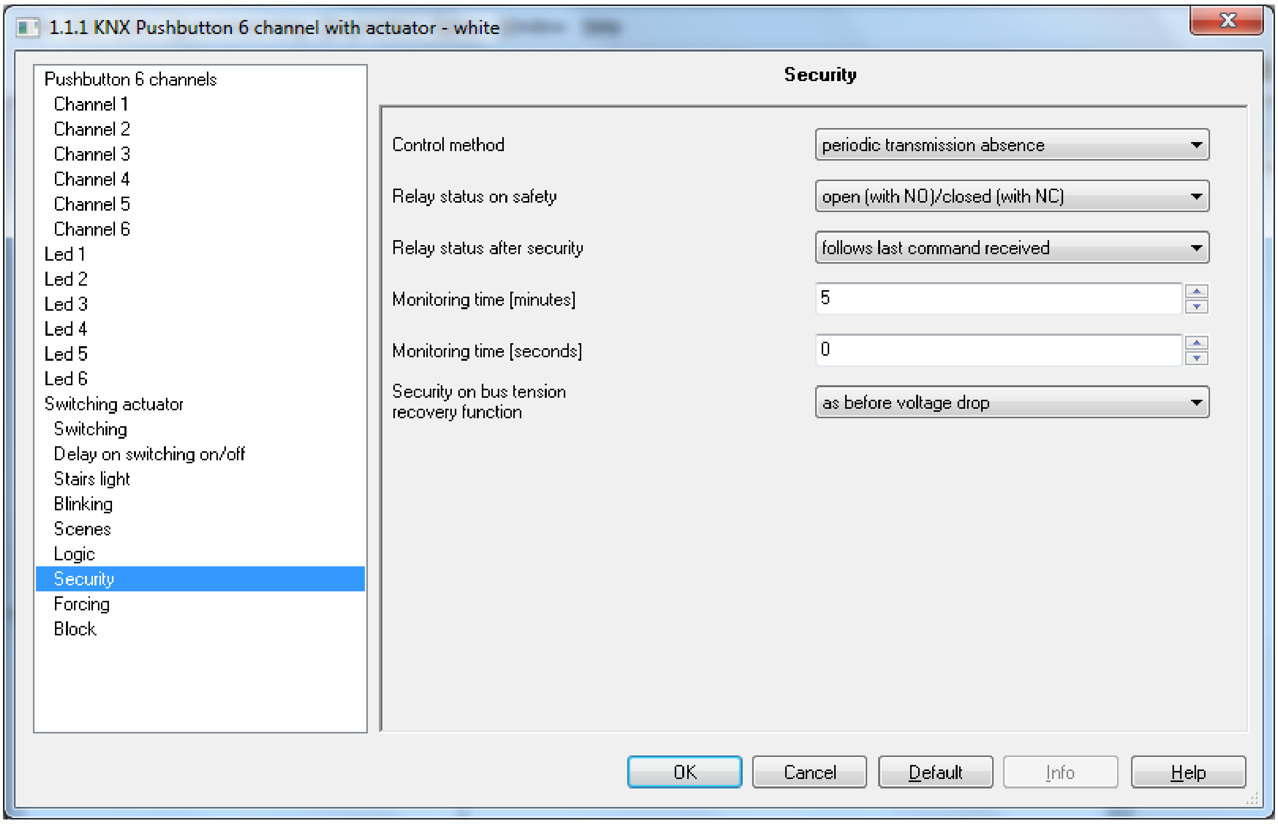


Fig. 14.8

➢ *14.8.1 Control method*

Allows to define the conditions for which the device activates the safety function; unlike the process for the ***Block*** and ***Command* priority** commands, which are activated via a bus command, the safety function is enabled by the device when the conditions set in the reference parameter occur. The values that can be set are:

- value “1” or periodic transmission absence
- value “0” or periodic transmission absence
- **periodic transmission absent (default value)**

By selecting **value “1” or no periodic transmission**, the safety function is activated following two events:

- the communication object ***Actuator security*** no longer receives the telegram with logic value “0” (no periodic transmission) for a period of time equal to the time represented by the values set in parameter **“Monitoring time [minutes]”** and **Px “Monitoring time [seconds]”**.

- the communication object ***Actuator security*** receives a telegram with logic value “1” (reception of value “1”).

In both cases, the safety function is deactivated when the communication object ***Actuator security*** receives a telegram with logic value “0”; once safety is deactivated, the monitoring time is restarted.

By selecting **value “0” or no periodic transmission**, the safety function is activated following two events:

- the communication object ***Actuator security*** no longer receives the telegram with logic value “1” (no periodic transmission) for a period of time equal to the time represented by the values set in parameters **“Monitoring time [minutes]”** and **Px “Monitoring time [seconds]”**.
- the communication object **Actuator security** receives a telegram with logic value “0” (reception of value “0”).

In both cases, the safety function is deactivated when the communication object ***Actuator security*** receives a telegram with logic value “1”; once safety is deactivated, the monitoring time is restarted.

By selecting the value **no periodic transmission**, the safety function is activated when the communication object ***Actuator security*** does not receive any telegram for a period of time equal to the time represented by the values set in parameters **“Monitoring time [minutes]”** and **“Monitoring time [seconds]”**, independently of the value of the telegram itself.
The safety function is deactivated when the communication object ***Actuator security*** receives a telegram with logic value “0” or "1"; once safety is deactivated, the monitoring time is restarted.

➢ *14.8.2 Relay status on safety*

This makes it possible to set the status of the change-over contact when the safety function is active; The values that can be set are:

- **open (with NO)/closed (with NC) (default value)**
- closed (with NO)/open (with NC)
- No change

➢ *14.8.3 Relay status after security*

When normal operating conditions are restored (safety deactivation), the status to which the actuator switches the relay is defined by the parameter **“Relay status after security”**; the possible values are:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)
- no change
- **follows last command received (default value)**
- as before security activation

If the parameter assumes the value **follows last command received,** the actuator follows the dynamics determined by the last command as if the execution of the command was initiated at the moment in which it was effectively received. Essentially, the command is executed in the background and is applied to the output in the moment in which safety is ended. This behaviour applies, for example, to timed actuation commands with timing that has a duration that goes beyond the moment of safety deactivation or commands with delayed activation/deactivation.

➢ *14.8.4 Monitoring time [minutes]*

This is used to set the first of two values (minutes) that make up the time that must pass after which the device will activate the safety if it does not receive the expected telegram (no periodic transmission); the settable values range from 0 to 59, **5** is the **default value**.

➢ *14.8.5 Monitoring time [seconds]*

This is used to set the second of two values (seconds) that make up the time that must pass after which the device will activate the safety if it does not receive the expected telegram (no periodic transmission); the settable values range from **0 (default value)** to 59.

➢ *14.8.6 Security on bus tension recovery function*

This parameter is used to determine the status of the safety function on bus voltage recovery. This parameter is useful if the function is active when the bus voltage drops and you want to have the actuator behaviour not be changed after voltage failure. The parameter may assume the following values:

- deactivated
- **as before voltage drop (default value)**

If the value **deactivated** is selected (and safety was activated before the bus voltage drop), when the bus voltage is recovered the safety function will be deactivated and the relay will take on the value determined by the parameter **"Relay status after security"**. If the value set for this last parameter is **follows last command received**, the actuator will execute the last command received before the bus voltage drop that, as a result, must be stored to the non-volatile memory. If the last command received before voltage drop is a timed activation of activation delay command, when the bus voltage is recovered the command will not be executed and the relay will switch to the open (with NO)/closed (with NC) status.
If the value **as before voltage drop** is selected (and safety was activated before the bus voltage drop), when the bus voltage is recovered the safety function will be reactivated and the relay will be set to the conditions set in the parameter **"Relay status on safety"**.

14.9 “Forcing” function

It is possible to force the relay status in a certain condition (settable) after receiving the communication object ***Actuator priority command*** that activates the forcing function; until this is deactivated, any command received on all other input communication objects will not be executed, with the exception of commands received on the object ***Actuator block***. The forcing function has the highest priority over all others with the exception of the Block function.

The semantics of the command received from the bus follows what is shown in the following table:

bit1	bit 0	
0	0	Deactivates forcing
0	1	Deactivates forcing
1	0	Forcing OFF
1	1	Forcing ON

When receiving the priority command with the forcing ON activation value, the actuator switches the relay, closing the NO contact and opening the NC contact; vice versa, when receiving the priority command with the forcing OFF activation value, the actuator switches the relay, opening the NO contact and closing the NC contact.

The basic structure of the menu is as follows (fig. 14.9):

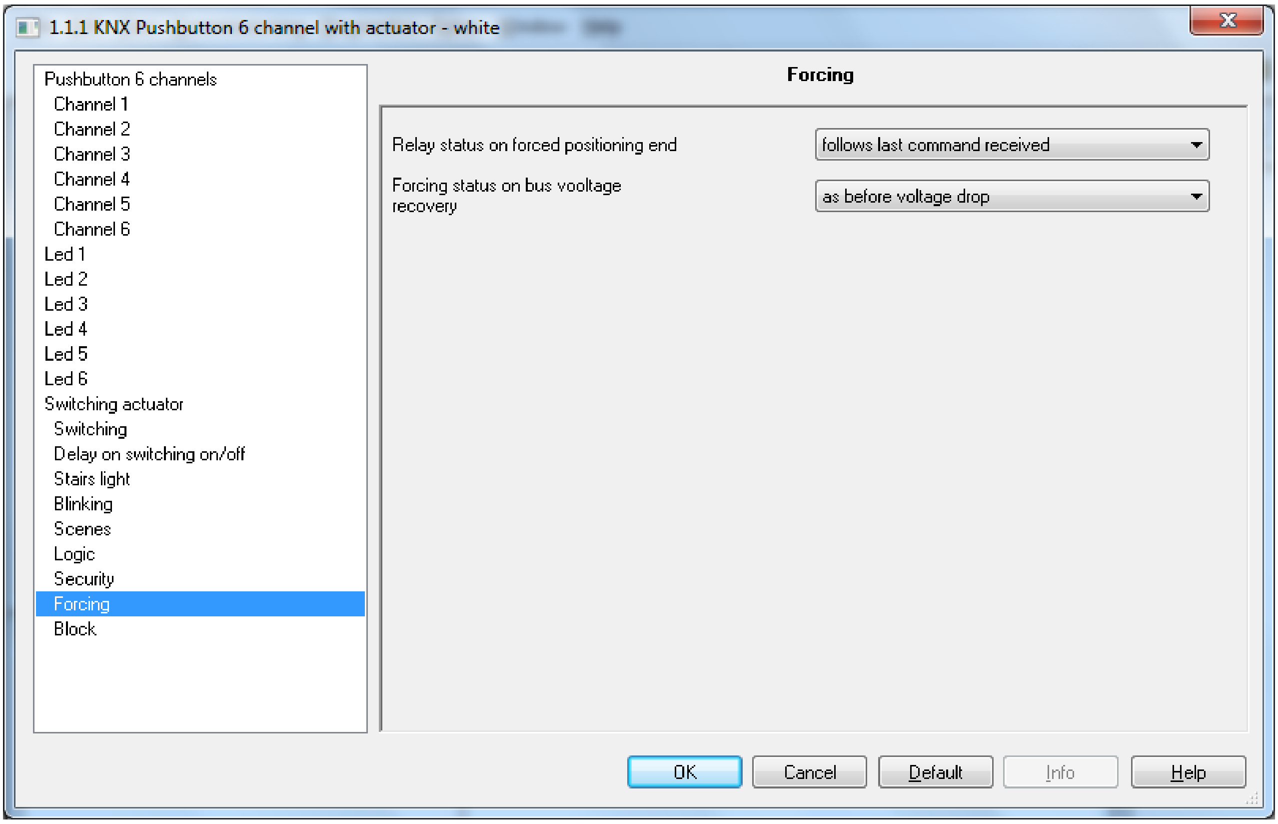


Fig. 14.9

➢ *14.9.1 Relay status on forced positioning end*

Upon receipt of the forcing deactivation command, the status to which the actuator switches the relay is defined by this parameter; the possible values are:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)
- no change

- **follows last command received (default value)**
- as before forcing activation

If the parameter assumes the value **follows last command received,** the actuator follows the dynamics determined by the last command as if the execution of the command was initiated at the moment in which it was effectively received. Essentially, the command is executed in the background and is applied to the output in the moment in which forcing is ended. This behaviour applies, for example, to timed actuation commands with timing that has a duration that goes beyond the moment of forcing deactivation or commands with delayed activation/deactivation.

➢ *14.9.2 Forcing status on bus voltage recovery*

This parameter is used to determine the status of the forcing function on bus voltage recovery. This parameter is useful if the function is active when the bus voltage drops and you want to have the actuator behaviour not be changed after voltage failure. The parameter may assume the following values:

- deactivated
- **as before voltage drop (default value)**

If the value **deactivated** is selected (and forcing was activated before the bus voltage drop), when the bus voltage is recovered the forcing function will be deactivated and the relay will be set to the value determined by this parameter. If the value set for this last parameter is **follows last command received**, the actuator will execute the last command received before the bus voltage drop that, as a result, must be stored to the non-volatile memory. If the last command received before voltage drop is a timed activation of activation delay command, when the bus voltage is recovered the command will not be executed and the relay will switch to the open (with NO)/closed (with NC) status.
If the value **as before voltage drop** is selected (and forcing was activated before bus voltage drop), when the bus voltage is recovered the forcing function is reactivated and the relay switches to the status prior to the voltage drop.
If a forcing deactivation command is received, if the parameter **Relay status on forced positioning end** takes on the value **follows last command received**, the actuator executes the last command received before the bus voltage drop that, as a result, must be stored to the non-volatile memory.. If the last command received before voltage drop is a timed activation of activation delay command, when the bus voltage is recovered the command will not be executed and the relay will switch to the open (with NO)/closed (with NC) status.

14.10 “Block” function

It is possible to block the device in a certain (settable) condition after receiving the communication object ***Actuator block*** that activates the block function; until it is deactivated, any command received on all other input communication objects will not be executed. The block function is the function with the highest priority.
The basic structure of the menu is as follows (fig. 14.10):

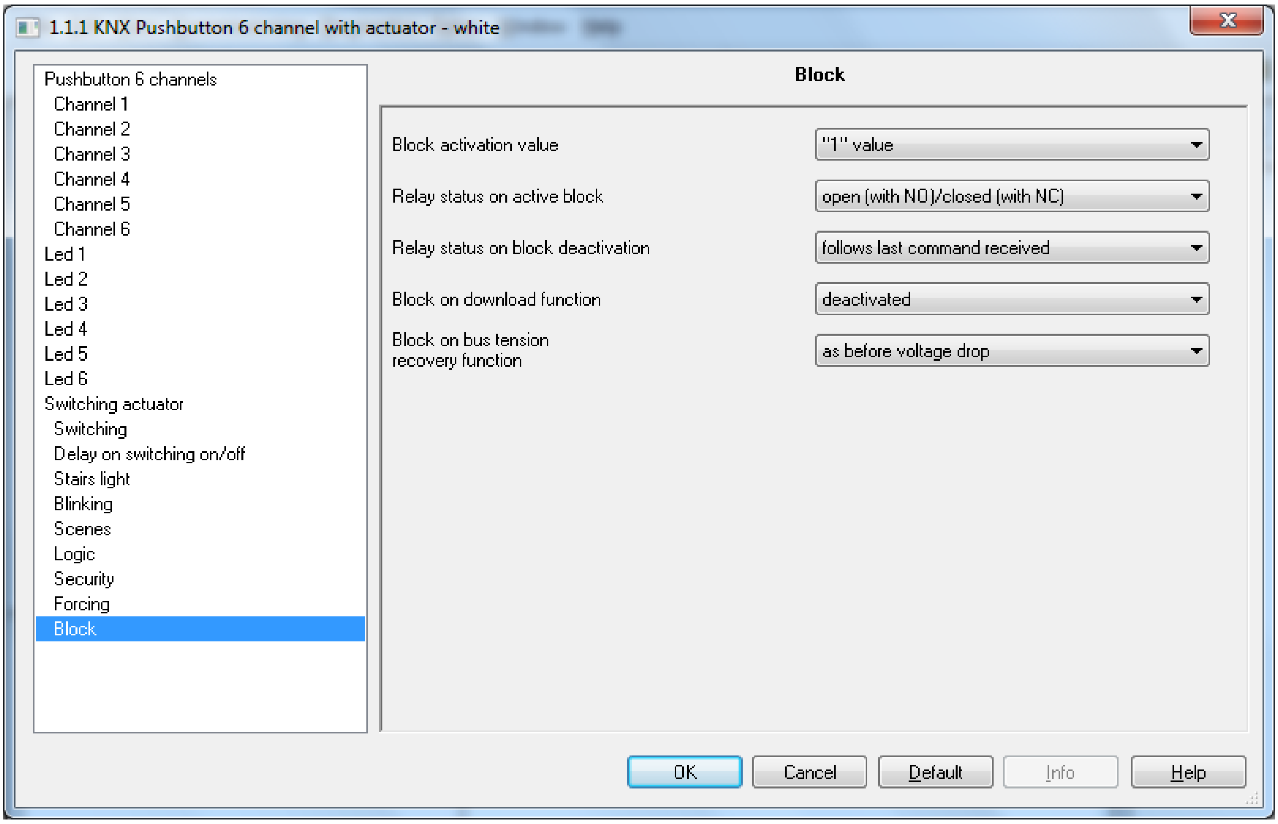


Fig. 14.10

➢ *14.10.1 Block activation value*

This determines which logic value activates the actuator block function; the possible values are:

- value “0”
- **value “1” (default value)**

➢ *14.10.2 Relay status on active block*

This makes it possible to set the status that the change-over contact must assume if the block function is activated; the possible values are:

- **open (with NO)/closed (with NC) (default value)**
- closed (with NO)/open (with NC)
- no change

➢ *14.10.3 Relay status on block deactivation*

This makes it possible to set the status that the change-over contact must assume if the block function is deactivated; the possible values are:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)

- no change
- **follows last command received (default value)**
- as before block activation

If the parameter assumes the value **follows last command received,** the actuator follows the dynamics determined by the last command as if the execution of the command was initiated at the moment in which it was effectively received. Essentially, the command is executed in the background and is applied to the output in the moment in which the block is deactivated. This behaviour applies, for example, to timed actuation commands with timing that has a duration that goes beyond the moment of block deactivation or commands with delayed activation/deactivation.

➢ *14.10.4 Block on download function*

This makes it possible to set the block function after downloading the application from ETS; the possible values are:

- **deactivated (default value)**
- active

➢ *14.10.5 Block on bus tension recovery function*

This makes it possible to set the status of the block function after bus voltage recovery; the possible values are:

- deactivated
- active
- **as before voltage drop (default value)**

If the value **deactivated** is selected (and the block function was activated before the bus voltage drop), when the bus voltage is recovered the block function will be deactivated and the relay will take on the value determined by the parameter **"Relay status on block deactivation"**. If the value set for this last parameter is **follows last command received**, the actuator will execute the last command received before the bus voltage drop that, as a result, must be stored to the non-volatile memory. If the last command received before voltage drop is a timed activation of activation delay command, when the bus voltage is recovered the command will not be executed and the relay will switch to the open (with NO)/closed (with NC) status.
If the value **as before bus voltage drop** is selected (and the block function was activated before the bus voltage drop), when the bus voltage is recovered the block function will be reactivated and the relay will be set to the conditions set in the parameter **"Relay status with active block"**.

15 Communication objects

The communication objects are listed in the following table (this example shows only general objects and those relative to channel 1 and 1/2 of the push-button panel and those of the actuator) (Fig. 15.1):

Number	Name	Object Function	Leng...	C	R	W	T	U	Data Type	Priority
0	Ch.1 - Block	Switching On/Off	1 bit	C	-	W	-	-		Low
0	Ch.1/2 - Block	Switching On/Off	1 bit	C	-	W	-	-		Low
1	Ch.1 - A Sequence	On/Off	1 bit	C	R	-	T	-		Low
1	Ch.1 - Shutter movement	Up/Down	1 bit	C	R	-	T	-		Low
1	Ch.1 - Switch	On/Off	1 bit	C	R	-	T	-		Low
1	Ch.1/2 - Switch	On/Off	1 bit	C	R	-	T	-		Low
1	Ch.1/2 - Shutter movement	Up/Down	1 bit	C	R	-	T	-		Low
1	Ch.1 - Scene	Execute/Store	1 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 14 bytes value	Characters ISO 8859-1	14 B...	C	R	-	T	-		Low
1	Ch.1 - A object 4 bytes value	Signed value	4 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 4 bytes value	Unsigned value	4 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 2 bytes value	Signed value	2 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 2 bytes value	Unsigned value	2 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 1 byte value	HVAC mode	1 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 1 byte value	% Value	1 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 1 byte value	Signed value	1 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 1 byte value	Unsigned value	1 Byte	C	R	-	T	-		Low
1	Ch.1 - A object 2 bits value	On/Off forced positioning	2 bit	C	R	-	T	-		Low
1	Ch.1 - A object 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
1	Ch.1 - A Sequence	On/Off	1 bit	C	R	-	T	-		Low
2	Ch.1 - Shutter stop/Louvres control	Stop/step	1 bit	C	R	-	T	-		Low
2	Ch.1 - Brightness dimming	Increase/Decrease	4 bit	C	R	-	T	-		Low
2	Ch.1/2 - Shutter stop/Louvres control	Stop/step	1 bit	C	R	-	T	-		Low
2	Ch.1/2 - Brightness dimming	Increase/Decrease	4 bit	C	R	-	T	-		Low
2	Ch.1 - Scene storing trigger	Store	1 bit	C	-	W	-	-		Low
2	Ch.1 - B object 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
2	Ch.1 - B sequence	On/Off	1 bit	C	R	-	T	-		Low
3	Ch.1 - C object 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
3	Ch.1 - C sequence	On/Off	1 bit	C	R	-	T	-		Low
4	Ch.1 - D object 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
4	Ch.1 - D sequence	On/Off	1 bit	C	R	-	T	-		Low
5	Ch.1 - E object 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
5	Ch.1 - E sequence	On/Off	1 bit	C	R	-	T	-		Low
6	Ch.1 - F object 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
6	Ch.1 - F sequence	On/Off	1 bit	C	R	-	T	-		Low
7	Ch.1 - G object 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
7	Ch.1 - G sequence	On/Off	1 bit	C	R	-	T	-		Low
8	Ch.1 - Hobject 1 bit value	1/0 value	1 bit	C	R	-	T	-		Low
8	Ch.1 - H sequence	On/Off	1 bit	C	R	-	T	-		Low
9	Ch.1 - Movement feedback	Increase/Decrease	1 bit	C	-	W	-	-		Low
9	Ch.1 - Dimmer status feedback	On/Off status	1 bit	C	-	W	T	U		Low
9	Ch.1 - A object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
10	Ch.1 - B object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
11	Ch.1 - C object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
12	Ch.1 - D object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
13	Ch.1 - E object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
14	Ch.1 - F object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
15	Ch.1 - G object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
16	Ch.1 - H object status feedback	On/Off status	1 bit	C	-	W	T	U		Low
102	Led 1 - Effect 1	Switching On/Off	1 bit	C	-	W	-	U		Low
102	Led 1 - Effect 1	Switching On/Off	1 bit	C	-	W	-	U		Low
103	Led 1 - Effect 2	Switching On/Off	1 bit	C	-	W	-	U		Low
104	Led 1 - Effect 3	Switching On/Off	1 bit	C	-	W	-	U		Low
105	Led 1 - Effect 4	Switching On/Off	1 bit	C	-	W	-	U		Low
106	Led 1 - Effect 5	Switching On/Off	1 bit	C	-	W	-	U		Low
107	Led 1 - Color choice	1=green/0=amber	1 bit	C	-	W	-	-		Low
138	Actuator status	On/Off status	1 bit	C	R	-	T	-		Low

Number	Name	Object Function	Leng...	C	R	W	T	U	Data Type	Priority
139	Actuator switching	On/Off	1 bit	C	-	W	-	-		Low
140	Actuator delayed switching	On/Off	1 bit	C	-	W	-	-		Low
141	Delay on actuator switching on	Set value	2 Byte	C	-	W	-	-		Low
142	Delay on actuator switching off	Set value	2 Byte	C	-	W	-	-		Low
143	Actuator timed switching	Start/Stop	1 bit	C	-	W	-	-		Low
144	Actuator stairs light activation time	Set value	2 Byte	C	-	W	-	-		Low
145	Actuator blinking	Switching On/Off	1 bit	C	-	W	-	-		Low
146	Actuator scene	Execute/Store	1 Byte	C	-	W	-	-		Low
147	Actuator scenes storing enabling	Enable/Disable	1 bit	C	-	W	-	-		Low
148	Actuator logic input 1	Logic function input	1 bit	C	-	W	-	-		Low
149	Actuator logic input 2	Logic function input	1 bit	C	-	W	-	-		Low
150	Actuator logic input 3	Logic function input	1 bit	C	-	W	-	-		Low
151	Actuator logic input 4	Logic function input	1 bit	C	-	W	-	-		Low
152	Actuator logical operation outcome	Logic	1 bit	C	R	-	T	-		Low
153	Actuator security	Monitoring	1 bit	C	-	W	-	-		Low
154	Actuator priority command	On/Off forced positioning	2 bit	C	-	W	-	-		Low
155	Actuator block	Switching On/Off	1 bit	C	-	W	-	-		Low

Fig. 15.1

15.1 Communication object table

The following tables summarise all the communication objects with their ID number, the name and function displayed in ETS, plus a brief description of the function performed and the type of Datapoint used.

➢ 15.1.1 Communication objects with input functions

The following table contains all the objects with an input function:

Communication object no.						Object name	Object function	Description	Datapoint type
Ch.1	Ch.2	Ch.3	Ch.4	Ch.5	Ch.6				
0	17	34	51	68	85	Ch.x - Block	Switching On /Off	Used to activate/deactivate the block function	1.003 DPT_Enable
0	0	34	34	68	68	Ch.x/y - Block	Switching On /Off	Used to activate/deactivate the block function	1.003 DPT_Enable
2	19	36	53	70	87	Ch.x - Scene storing trigger	Store	Receives the request (trigger) to send a scene storing message	1.017 DPT_Trigger
9	26	43	60	77	94	Ch.x - Dimmer status feedback	On/Off status	Receives the dimmer status feedback	1.001 DPT_Switch

9	26	43	60	77	94	Ch.x - A status feedback	On/Off status	Receives the actuator status feedback for A object cyclic switching	1.001 DPT_Switch
9	26	43	60	77	94	Ch.x - Movement feedback	Increase/Decrease	Receives the feedback about the current movement direction of the motor command actuator	1.008 DPT_UpDown
10	27	44	61	78	95	Ch.x - B status feedback	On/Off status	Receives the actuator status feedback for B object cyclic switching	1.001 DPT_Switch
11	28	45	62	79	96	Ch.x - C status feedback	On/Off status	Receives the actuator status feedback for C object cyclic switching	1.001 DPT_Switch
12	29	46	63	80	97	Ch.x - D status feedback	On/Off status	Receives the actuator status feedback for D object cyclic switching	1.001 DPT_Switch
13	30	47	64	81	98	Ch.x - E status feedback	On/Off status	Receives the actuator status feedback for E object cyclic switching	1.001 DPT_Switch
14	31	48	65	82	99	Ch.x - F status feedback	On/Off status	Receives the actuator status feedback for F object cyclic switching	1.001 DPT_Switch

15	32	49	66	83	100	Ch.x - G status feedback	On/Off status	Receives the actuator status feedback for G object cyclic switching	1.001 DPT_Switch
16	33	50	67	84	101	Ch.x - H status feedback	On/Off status	Receives the actuator status feedback for H object cyclic switching	1.001 DPT_Switch
102						Led 1 - Effect 1	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
103						Led 1 - Effect 2	Switching On /Off	Switching On /Off light effect 2	1.001 DPT_Switch
104						Led 1 - Effect 3	Switching On /Off	Switching On /Off light effect 3	1.001 DPT_Switch
105						Led 1 - Effect 4	Switching On /Off	Switching On /Off light effect 4	1.001 DPT_Switch
106						Led 1 - Effect 5	Switching On /Off	Switching On /Off light effect 5	1.001 DPT_Switch
108						Led 2 - Effect 1	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
109						Led 2 - Effect 2	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
110						Led 2 - Effect 3	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
111						Led 2 - Effect 4	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
112						Led 2 - Effect 5	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
114						Led 3 - Effect 1	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
115						Led 3 - Effect 2	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
116						Led 3 - Effect 3	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch

117	Led 3 - Effect 4	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
118	Led 3 - Effect 5	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
120	Led 4 - Effect 1	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
121	Led 4 - Effect 2	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
122	Led 4 - Effect 3	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
123	Led 4 - Effect 4	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
124	Led 4 - Effect 5	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
126	Led 5 - Effect 1	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
127	Led 5 - Effect 2	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
128	Led 5 - Effect 3	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
129	Led 5 - Effect 4	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
130	Led 5 - Effect 5	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
132	Led 6 - Effect 1	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
133	Led 6 - Effect 2	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
134	Led 6 - Effect 3	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
135	Led 6 - Effect 4	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
136	Led 6 - Effect 5	Switching On /Off	Switching On /Off light effect 1	1.001 DPT_Switch
107	Led 1 - Color choice	1=green/0=amber	Selects the light signalling color	1.001 DPT_Switch

113	Led 2 - Color choice	1=green/0=amber	Selects the light signalling color	1.001 DPT_Switch
119	Led 3 - Color choice	1=green/0=amber	Selects the light signalling color	1.001 DPT_Switch
125	Led 4 - Color choice	1=green/0=amber	Selects the light signalling color	1.001 DPT_Switch
131	Led 5 - Color choice	1=green/0=amber	Selects the light signalling color	1.001 DPT_Switch
137	Led 6 - Color choice	1=green/0=amber	Selects the light signalling color	1.001 DPT_Switch
139	Actuator switching	On/Off	Receives the load activation/deactivation commands	1.001 DPT_Switch
140	Actuator delayed switching	On/Off	Receives the load activation/deactivation with delay commands	1.001 DPT_Switch
143	Actuator timed switching	Start/Stop	Receives the timed activation start/stop commands	1.001 DPT_Switch
145	Actuator blinking	Switching On /Off	Receives the load blinking mode activation/deactivation commands	1.001 DPT_Switch
146	Actuator scene	Execute/Store	Makes it possible to store/execute scenes	18.001 DPT_SceneControl
148	Actuator logic input 1	Logic function input	Logic function input	1.002 DPT_Bool
149	Actuator logic input 2	Logic function input	Logic function input	1.002 DPT_Bool
150	Actuator logic input 3	Logic function input	Logic function input	1.002 DPT_Bool
151	Actuator logic input 4	Logic function input	Logic function input	1.002 DPT_Bool

153	Actuator security	Monitoring	Makes it possible to monitor a sensor for the safety function	1.001 DPT_Switch
154	Actuator priority command	On/Off forced positioning	Forces the load value to an on/off value	2.001 DPT_Switch_Control
155	Actuator block	Switching On /Off	Blocks the status of a load in a parameterizable condition	1.003 DPT_Enable

➢ *15.1.2 Communication objects with output functions*

The following table contains all the objects with an output function:

Communication object no.						Object name	Object function	Description	Datapoint type
Ch.1	Ch.2	Ch.3	Ch.4	Ch.5	Ch.6				
1	18	35	52	69	86	Ch.x - Switching	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
1	1	35	35	69	69	Ch.x/y - Switching	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
1	18	35	52	69	86	Ch.x - Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
1	1	35	35	69	69	Ch.x/y - Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
1	18	35	52	69	86	Ch.x - Scene	Execute/Store	Sends scene memorising/execution commands	18.001 DPT_SceneControl
1	18	35	52	69	86	Ch.x - A Sequence	On/Off	Sends On/Off commands associated with A object of the sequence	1.001 DPT_Switch
1	18	35	52	69	86	Ch.x - A object 1 bit value	1/0 value	Sends values 1/0 associated with A object	1.002 DPT_Bool
1	18	35	52	69	86	Ch.x - A object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with A object	2.001 DPT_Switch_Control
1	18	35	52	69	86	Ch.x - A object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with A object	5.010 DPT_Value_1_Ucount
1	18	35	52	69	86	Ch.x - A object 1 byte value	Signed value	Sends signed values (-128..127) associated with A object	6.010 DPT_Value_1_Count

1	18	35	52	69	86	Ch.x - A object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with A object	5.001 DPT_Scaling
1	18	35	52	69	86	Ch.x - A object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
1	18	35	52	69	86	Ch.x - A object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with A object	7.001 DPT_Value_2_Ucount
1	18	35	52	69	86	Ch.x - A object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with A object	8.001 DPT_Value_2_Count
1	18	35	52	69	86	Ch.x - A object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with A object	12.001 DPT_Value_4_Ucount
1	18	35	52	69	86	Ch.x - A object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with A object	13.001 DPT_Value_4_Count
1	18	35	52	69	86	Ch.x - A object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
2	19	36	53	70	87	Ch.x - Brightness dimming	Increase/decrease	Sends brightness control commands	3.007 DPT_Control_Dimming
2	2	36	36	70	70	Ch.x/y - Brightness dimming	Increase/decrease	Sends brightness control commands	3.007 DPT_Control_Dimming
2	19	36	53	70	87	Ch.x - Shutter stop/Louvres control	Stop/Step	Sends stop movement/louvres control commands	1.007 DPT_Step
2	2	36	36	70	70	Ch.x/y - Shutter stop/Louvres control	Stop/Step	Sends stop movement/louvres control commands	1.007 DPT_Step
2	19	36	53	70	87	Ch.x - B sequence	On/Off	Sends On/Off commands associated with B object of the sequence	1.001 DPT_Switch
2	19	36	53	70	87	Ch.x - B object 1 bit value	1/0 value	Sends values 1/0 associated with B object	1.002 DPT_Bool

2	19	36	53	70	87	Ch.x - B object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with B object	2.001 DPT_Switch_Control
2	19	36	53	70	87	Ch.x - B object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with B object	5.010 DPT_Value_1_Ucount
2	19	36	53	70	87	Ch.x - B object 1 byte value	Signed value	Sends signed values (-128..127) associated with B object	6.010 DPT_Value_1_Count
2	19	36	53	70	87	Ch.x - B object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with B object	5.001 DPT_Scaling
2	19	36	53	70	87	Ch.x - B object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
2	19	36	53	70	87	Ch.x - B object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with B object	7.001 DPT_Value_2_Ucount
2	19	36	53	70	87	Ch.x - B object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with B object	8.001 DPT_Value_2_Count
2	19	36	53	70	87	Ch.x - B object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with B object	12.001 DPT_Value_4_Ucount
2	19	36	53	70	87	Ch.x - B object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with B object	13.001 DPT_Value_4_Count
2	19	36	53	70	87	Ch.x - B object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
3	20	37	54	71	88	Ch.x - C sequence	On/Off	Sends On/Off commands associated with C object of the sequence	1.001 DPT_Switch
3	20	37	54	71	88	Ch.x - C object 1 bit value	1/0 value	Sends values 1/0 associated with C object	1.002 DPT_Bool
3	20	37	54	71	88	Ch.x - C object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with C object	2.001 DPT_Switch_Control

3	20	37	54	71	88	Ch.x - C object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with C object	5.010 DPT_Value_1_Ucount
3	20	37	54	71	88	Ch.x - C object 1 byte value	Signed value	Sends signed values (-128..127) associated with C object	6.010 DPT_Value_1_Count
3	20	37	54	71	88	Ch.x - C object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with C object	5.001 DPT_Scaling
3	20	37	54	71	88	Ch.x - C object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
3	20	37	54	71	88	Ch.x - C object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with C object	7.001 DPT_Value_2_Ucount
3	20	37	54	71	88	Ch.x - C object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with C object	8.001 DPT_Value_2_Count
3	20	37	54	71	88	Ch.x - C object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with C object	12.001 DPT_Value_4_Ucount
3	20	37	54	71	88	Ch.x - C object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with C object	13.001 DPT_Value_4_Count
3	20	37	54	71	88	Ch.x - C object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
4	21	38	55	72	89	Ch.x - D sequence	On/Off	Sends On/Off commands associated with D object of the sequence	1.001 DPT_Switch
4	21	38	55	72	89	Ch.x - D object 1 bit value	1/0 value	Sends values 1/0 associated with object D	1.002 DPT_Bool
4	21	38	55	72	89	Ch.x - D object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with object D	2.001 DPT_Switch_Control

4	21	38	55	72	89	Ch.x - D object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with D object	5.010 DPT_Value_1_Ucount
4	21	38	55	72	89	Ch.x - D object 1 byte value	Signed value	Sends signed values (-128..127) associated with D object	6.010 DPT_Value_1_Count
4	21	38	55	72	89	Ch.x - D object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with D object	5.001 DPT_Scaling
4	21	38	55	72	89	Ch.x - D object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
4	21	38	55	72	89	Ch.x - D object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with D object	7.001 DPT_Value_2_Ucount
4	21	38	55	72	89	Ch.x - D object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with D object	8.001 DPT_Value_2_Count
4	21	38	55	72	89	Ch.x - D object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with D object	12.001 DPT_Value_4_Ucount
4	21	38	55	72	89	Ch.x - D object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with D object	13.001 DPT_Value_4_Count
4	21	38	55	72	89	Ch.x - D object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
5	22	39	56	73	90	Ch.x - E sequence	On/Off	Sends On/Off commands associated with E object of the sequence	1.001 DPT_Switch
5	22	39	56	73	90	Ch.x - E object 1 bit value	1/0 value	Sends values 1/0 associated with E object	1.002 DPT_Bool
5	22	39	56	73	90	Ch.x - E object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with E object	2.001 DPT_Switch_Control

5	22	39	56	73	90	Ch.x - E object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with E object	5.010 DPT_Value_1_Ucount
5	22	39	56	73	90	Ch.x - E object 1 byte value	Signed value	Sends signed values (-128..127) associated with E object	6.010 DPT_Value_1_Count
5	22	39	56	73	90	Ch.x - E object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with E object	5.001 DPT_Scaling
5	22	39	56	73	90	Ch.x - E object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
5	22	39	56	73	90	Ch.x - E object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with E object	7.001 DPT_Value_2_Ucount
5	22	39	56	73	90	Ch.x - E object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with E object	8.001 DPT_Value_2_Count
5	22	39	56	73	90	Ch.x - E object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with E object	12.001 DPT_Value_4_Ucount
5	22	39	56	73	90	Ch.x - E object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with E object	13.001 DPT_Value_4_Count
5	22	39	56	73	90	Ch.x - E object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
6	23	40	57	74	91	Ch.x - F sequence	On/Off	Sends On/Off commands associated with F object of the sequence	1.001 DPT_Switch
6	23	40	57	74	91	Ch.x - F object 1 bit value	1/0 value	Sends values 1/0 associated with F object	1.002 DPT_Bool
6	23	40	57	74	91	Ch.x - F object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with F object	2.001 DPT_Switch_Control

6	23	40	57	74	91	Ch.x - F object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with F object	5.010 DPT_Value_1_Ucount
6	23	40	57	74	91	Ch.x - F object 1 byte value	Signed value	Sends signed values (-128..127) associated with F object	6.010 DPT_Value_1_Count
6	23	40	57	74	91	Ch.x - F object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with F object	5.001 DPT_Scaling
6	23	40	57	74	91	Ch.x - F object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
6	23	40	57	74	91	Ch.x - F object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with F object	7.001 DPT_Value_2_Ucount
6	23	40	57	74	91	Ch.x - F object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with F object	8.001 DPT_Value_2_Count
6	23	40	57	74	91	Ch.x - F object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with F object	12.001 DPT_Value_4_Ucount
6	23	40	57	74	91	Ch.x - F object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with F object	13.001 DPT_Value_4_Count
6	23	40	57	74	91	Ch.x - F object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
7	24	41	58	75	92	Ch.x - G sequence	On/Off	Sends On/Off commands associated with G object of the sequence	1.001 DPT_Switch
7	24	41	58	75	92	Ch.x - G object 1 bit value	1/0 value	Sends values 1/0 associated with G object	1.002 DPT_Bool
7	24	41	58	75	92	Ch.x - G object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with G object	2.001 DPT_Switch_Control

7	24	41	58	75	92	Ch.x - G object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with G object	5.010 DPT_Value_1_Ucount
7	24	41	58	75	92	Ch.x - G object 1 byte value	Signed value	Sends signed values (-128..127) associated with G object	6.010 DPT_Value_1_Count
7	24	41	58	75	92	Ch.x - G object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with G object	5.001 DPT_Scaling
7	24	41	58	75	92	Ch.x - G object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
7	24	41	58	75	92	Ch.x - G object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with G object	7.001 DPT_Value_2_Ucount
7	24	41	58	75	92	Ch.x - G object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with G object	8.001 DPT_Value_2_Count
7	24	41	58	75	92	Ch.x - G object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with G object	12.001 DPT_Value_4_Ucount
7	24	41	58	75	92	Ch.x - G object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with G object	13.001 DPT_Value_4_Count
7	24	41	58	75	92	Ch.x - G object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
8	25	42	59	76	93	Ch.x - H sequence	On/Off	Sends On/Off commands associated with H Object of the sequence	1.001 DPT_Switch
8	25	42	59	76	93	Ch.x - H object 1 bit value	1/0 value	Sends values 1/0 associated with H object	1.002 DPT_Bool
8	25	42	59	76	93	Ch.x - H object 2 bits value	On/Off forced positioning	Sends values 1/0 associated with H object	2.001 DPT_Switch_Control

8	25	42	59	76	93	Ch.x - H object 1 byte value	Unsigned value	Sends unsigned values (0..255) associated with H object	5.010 DPT_Value_1_Ucount
8	25	42	59	76	93	Ch.x - H object 1 byte value	Signed value	Sends signed values (-128..127) associated with H object	6.010 DPT_Value_1_Count
8	25	42	59	76	93	Ch.x - H object 1 byte value	% Value	Sends the percentage values (0%..100%) associated with H object	5.001 DPT_Scaling
8	25	42	59	76	93	Ch.x - H object 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
8	25	42	59	76	93	Ch.x - H object 2 bytes value	Unsigned value	Sends unsigned values (0..65535) associated with H object	7.001 DPT_Value_2_Ucount
8	25	42	59	76	93	Ch.x - H object 2 bytes value	Signed value	Sends signed values (-32768..32767) associated with H object	8.001 DPT_Value_2_Count
8	25	42	59	76	93	Ch.x - H object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with H object	12.001 DPT_Value_4_Ucount
8	25	42	59	76	93	Ch.x - H object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with H object	13.001 DPT_Value_4_Count
8	25	42	59	76	93	Ch.x - H object 14 bytes value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
138						Actuator status	On/Off status	Sends the status of the load connected to the actuator	1.001 DPT_Switch
152						Actuator logical operation outcome	Logic	Logic function output	1.002 DPT_Bool

➢ *15.1.3 Communication objects with parameter setting functions*

The following table contains all the objects with a bus parameter setting function:

Communication object no.	Object name	Object function	Description	Datapoint type
141	Delay on actuator switching on	Set value	Activation delay value	7.005 DPT_TimePeriodSec
142	Actuator deactivation delay	Set value	Deactivation delay value	7.005 DPT_TimePeriodSec
144	Actuator stairs light activation time	Set value	Stairs light timing value	7.005 DPT_TimePeriodSec
147	Actuator scenes storing enabling	Enable/disable	Scene learning enable/disable	1.003 DPT_Enable

Punto di contatto indicato in adempimento ai fini delle direttive e regolamenti UE applicabili:
Contact details according to the relevant European Directives and Regulations:
GEWISS S.p.A. Via A.Volta, 1 IT-24069 Cenate Sotto (BG) Italy tel: +39 035 946 111 E-mail: qualitymarks@gewiss.com

SAT

+39 035 946 111
8.30 - 12.30 / 14.00 - 18.00
lunedì ÷ venerdì - monday ÷ friday

24h **+39 035 946 260**

@ **sat@gewiss.com**
www.gewiss.com