

**KNX 6-channel touch push-button panel  
module with interchangeable symbols**



**GWA9421**

**Technical Manual**

# Contents

1	Introduction.....	3
2	Application .....	3
2.1	Association limits .....	3
3	“Main” menu .....	4
3.1	Parameters .....	4
4	“Channel x” menu (independent channels) .....	8
4.1	Parameters .....	8
5	“Edges/Sequence commands” function .....	11
5.1	Parameters .....	11
6	“1 push-button + stop dimmer” function .....	17
6.1	Parameters .....	17
7	“Cyclic sending 1 push-button dimmer” function .....	19
7.1	Parameters .....	19
8	“1 push-button shutter control” function.....	21
8.1	Parameters .....	21
9	“2 push-button + stop dimmer” function .....	22
9.1	Parameters .....	23
10	“2 push-button dimmer with cyclic sending” function.....	24
10.1	Parameters .....	25
11	“2 push-button shutter control” function.....	26
11.1	Parameters .....	26
12	“Scene management” function.....	27
12.1	Parameters .....	27
13	“Switching sequences” function .....	28
13.1	Parameters .....	28
14	“Multiple touches” function.....	32
14.1	Parameters .....	33
15	“Sound signalling” menu .....	55
15.1	Parameters .....	55
16	“Soft reduction” menu .....	58
16.1	Parameters .....	58
17	“LED X” menu .....	60
17.1	Parameters .....	60
18	Communication objects .....	65
19	Signalling of plate absence/presence and BUS voltage recovery.....	71

# 1 Introduction

The KNX 6-channel touch push-button panel module with interchangeable symbols (flush-mounting) is a command device with 6 channels used - on their own or combined - to perform the functions of ON/OFF, dimmer control, roller shutter control, scene management, priority and timed commands on a KNX BUS.

The device is powered from the BUS line, and each channel has RGB LEDs for night-time localisation and display of the commanded load status. The device is fitted with a buzzer that emits a signal when the plate is touched, and a proximity sensor. The push-button panel module is inside the standard flush-mounting boxes, mounted on the Chorus range supports in the space taken up by three modules.

## 2 Application

Each of the 6 channels of the push-button panel is configured with the ETS software to create one of the functions listed below.

Management of edge/command sequence:

- edge management touch/release with sequence sending
- short/long touch management with command transmission
- channel enabling/block

Scenes:

- management of scenes with 1-byte items
- sending of scene storing commands

Priority commands:

- sending of priority commands

Roller shutters/curtain command:

- with single or double push-button
- with sending of the percentage position (0%-100%)

Dimmer command:

- with single or double push-button
- with stop telegram or cyclical send
- with sending of the light intensity value (0%-100%)

Multiple touches:

- management of multiple touches on consecutive drives (maximum 4) with sending of commands

Switching sequences:

- with 1-bit objects on BUS (from 2 to 8)

Control of the output RGB LEDs:

- 5 lighting effects for each RGB LED, and colour selection

### 2.1 Association limits

Maximum number of group addresses:	254
Maximum number of associations:	254

This means that up to 254 group addresses can be defined, and up to 254 associations can be made (communication objects and group addresses).

### 3 “Main”menu

The **Main** menu contains the application parameters relating to the 6 channels implemented by the device (Fig. 3.1).

---1 KNX pushbutton module 6 ch. with replac. symbols > Main

Main		
Sound signalling	Channel 1	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Soft reduction	Channel 2	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Led 1	Channel 3	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Led 2	Channel 4	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Led 3	Channel 5	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Led 4	Channel 6	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Led 5		
Led 6	Proximity sensor sensitivity	<input checked="" type="radio"/> low <input type="radio"/> high
	Glass cleaning function	<input type="radio"/> disable <input checked="" type="radio"/> enable
	- Inhibition time [s]	30
	Backlight brightness with user in proximity	100%
	Backlight brightness in standby	30%
	Transmission delay after start	11.. 21 seconds (depending on physical address)

Fig. 3.1

#### 3.1 Parameters

##### ➤ 3.1.1 Channel X

Each of the 6 input channels implemented by the module can be managed autonomously, carrying out a function independent of the others; parameters “**Channel 1**”, “**Channel 2**”, “**Channel 3**”, “**Channel 4**”, “**Channel 5**” and “**Channel 6**” are used to enable the configuration of the relative input channels, making the configuration menus visible. The values that can be set are:

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

Selecting **enabled** displays the **Channel 1**, **Channel 2**, **Channel 3**, **Channel 4**, **Channel 5** or **Channel 6** configuration menu (see par. 4 - “Channel x” menu).

### ➤ 3.1.2 Proximity sensor sensitivity

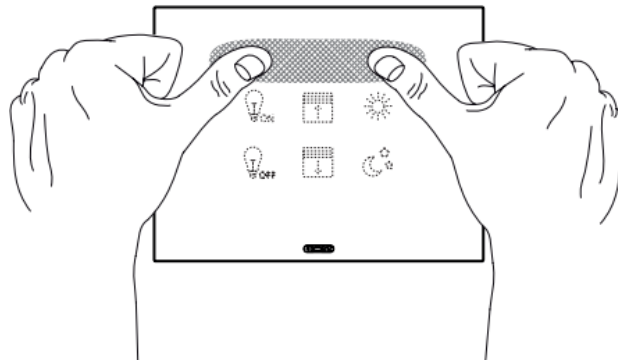
The device is fitted with a proximity sensor that is used to increase the level of brightness of the touch push-button backlighting when the user approaches the glass plate.

Using the “**Proximity sensor sensitivity**” parameter, you can set the level of sensitivity of the proximity sensor to increase/reduce the maximum distance at which the device is able to detect the approach of the user. The values that can be set are:

- **low** (default value)
- high

Setting the value **low**, the device can detect the user's presence at a maximum distance of 10 (TBD) cm. Setting the value **high**, the device can detect the user's presence at a maximum distance of 20 (TBD) cm. The sensitivity of the proximity sensor can always be modified using the following manual procedure:

#### Change mode input



1. touch the plate as shown in figure for at least 15 seconds, without touching the device channels
2. wait until you hear two long beeps, then remove your hand from the plate
3. wait until all six LEDs are flashing in a magenta colour
4. activation of LED 5 on the basis of the configuration that is active in that moment (all the other LEDs are OFF)

Ciclically change the sensitivity of the proximity sensor by touching channel CH5 repeatedly, as shown in the table:

LED 5	sensitivity of the proximity sensor
fixed blue	high
flashing blue	low

#### Change mode output

1. wait 10 seconds without touching any channels
2. wait until you hear a long beep
3. wait until all six LEDs are flashing quickly with a magenta colour
4. wait until the flashing stops and the current status or night-time localisation signals are reset

During this configuration phase, the messages from the BUS are ignored (they will be managed after you have quit the configuration phase).

### ➤ 3.1.3 Glass cleaning function

This parameter enables the inhibition of the capacitive sensors to allow the cleaning of the glass surface. The values that can be set are:

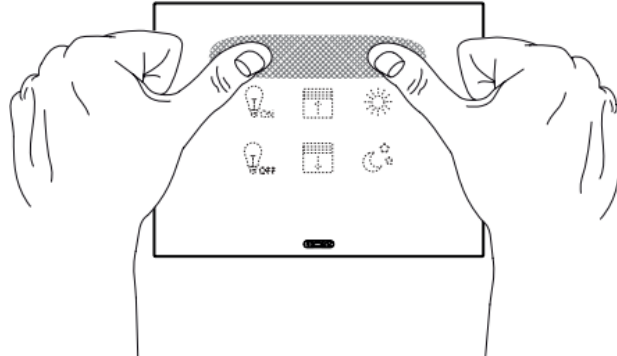
- Disabled
- **Enable (default value)**

Selecting **enable** displays the “**Inhibition time [s]**” parameter and the **Glass cleaning** communication object.

The “**Inhibition time [s]**” parameter allows you to define the sensor inhibition period. The values that can be set are:

- from 10 to 240 in steps of 1 (**default value 30**)

- manual activation:



### **Enabling**

1. touch the plate as shown for at least 10 seconds, without touching the device channels.
2. wait until you hear a long beep, then remove your hand from the plate
3. wait until all 6 LEDs are flashing blue
4. the countdown of the inhibition time set via ETS begins; this is indicated by the activation of the blue LEDs in the following sequence:

LED 1 with LED 2 for 500ms

LED 3 with LED 4 for 500ms

LED 5 with LED 6 for 500ms

### **Disabling**

1. wait until the end of the inhibition time set via ETS
2. wait until you hear a long beep
3. wait until all 6 LEDs are flashing quickly in a blue colour
4. wait until the flashing stops and the current status or night-time localisation signals are reset

During this configuration phase, the messages from the BUS are ignored (they will be managed after you have quit the configuration phase).

## ➤ **3.1.4 Backlight brightness**

In standby, the brightness of the backlighting is reduced to avoid disturbance and limit the consumption of the device on the BUS line.

The backlighting is activated when the user approaches the surface of the glass, and deactivates automatically after 30 seconds with no user activity on the device (no pressing of the touch push-buttons).

The “**Backlight brightness with user in proximity**” parameter is used to define the level of brightness of the backlighting when the proximity sensor detects the presence of the user in front of the device. The values that can be set are:

- **100%**                      (**default value**)
- 90%
- 80%
- 70%
- 60%
- 50%

The “**Backlight brightness in standby**” parameter is used to define the level of brightness of the backlighting when the device is in standby (after 30 seconds with no user activity on it). The values that can be set are:

- 40%
- **30%**                      (**default value**)

- 20%
- 10%
- 0%
- no reduction                    255

### ➤ **3.1.5 Transmission delay after start**

To ensure that, with multiple devices in the line, the telegrams sent by the various devices do not collide when the BUS voltage is recovered, it is possible 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. The parameter "**Transmission delay after start**" is used to set this delay; 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 a channel is enabled, a dedicated menu - called **Channel x** (x = 1 .. 6 is the input index) - is displayed for each input. 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. 4.1):

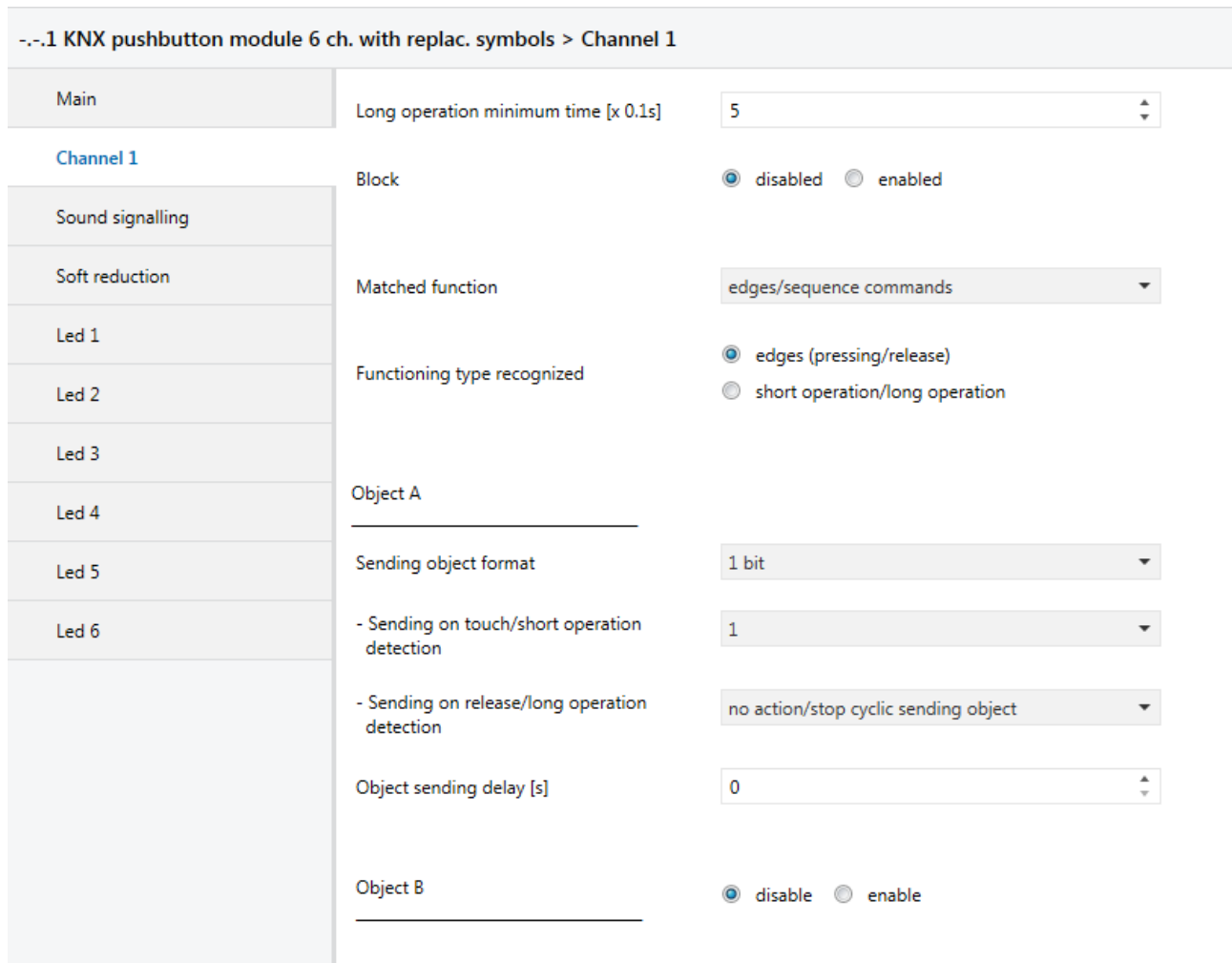


Fig 4.1

### 4.1 Parameters

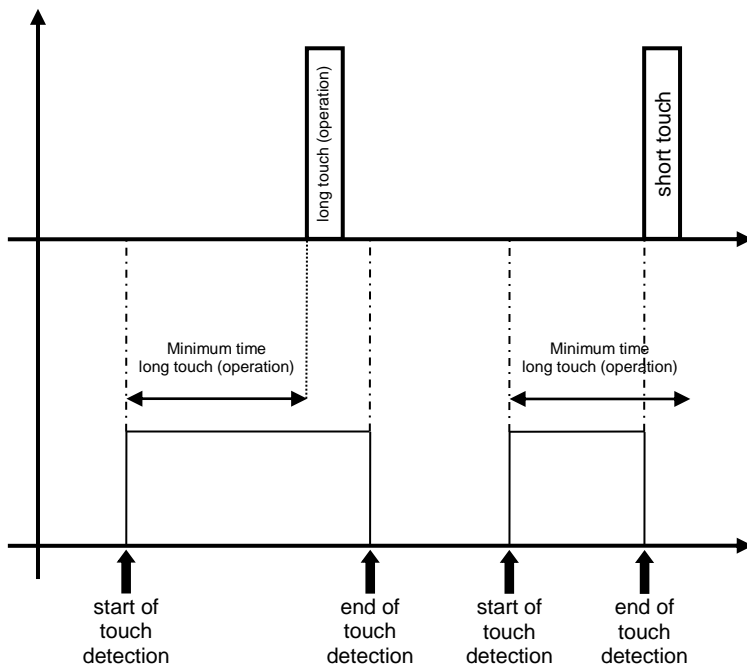
#### ➤ 4.1.1 Long operation minimum time [x x0.1s]

Many of the functions that the inputs can perform require differentiation between short and long touch (operation). The “**Long operation minimum time [x 0.1s]**” parameter is used to define the real minimum time in which the capacitive sensor must detect the touch, in order to differentiate between a long or short touch (operation) of the capacitive area associated with the channel. The possible values are:

- from 3 to 150 with steps of 1 (**default value 5**)

The following example shows the meaning of the above-mentioned parameter





### ➤ 4.1.2 Block

This inhibits the detection of the touch/release of the capacitive sensors, thereby preventing the device from sending - via the BUS - the telegrams associated with those events. If it is activated, any possible status variation will not be interpreted until a block deactivation command is received. The “**Block**” parameter can have the following values:

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

If **enabled** is selected, the following parameters will be visualised: “**Block activation value**” and “**Block function on BUS voltage recovery (= Block on BUS voltage recovery function)**”, along with the **Ch.x - Block** communication item, via which it is possible to activate the function by means of the 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 BUS voltage recovery; The values that can be set are:

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

### ➤ 4.1.3 Matched function

This determines the function associated with the general channel x. Depending on the value set with this parameter, the **Channel x** menu will appear differently. The values that can be set are:

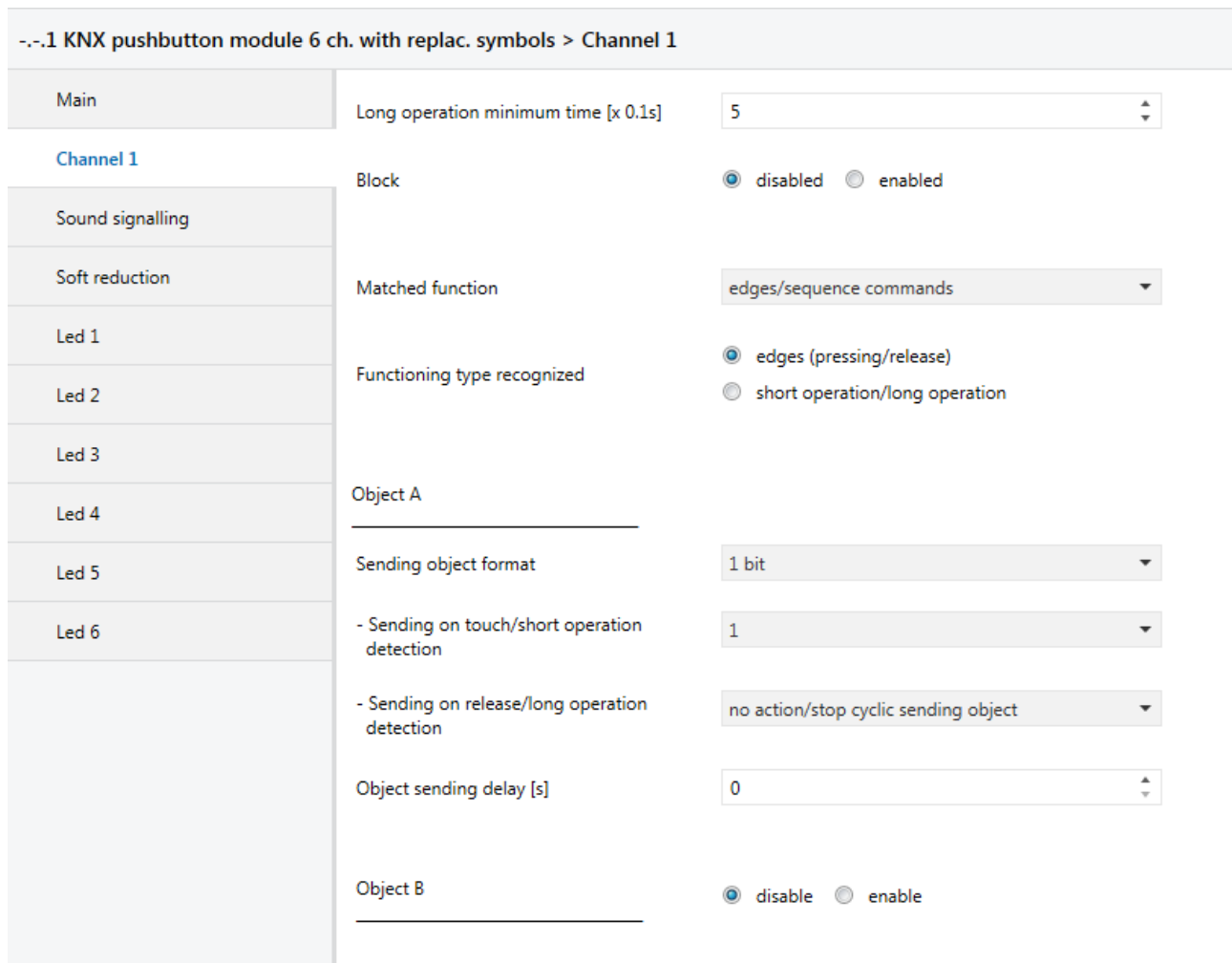
- **Edges/Sequence commands**  
See chapter 5 - “**Edges/Sequence commands**” function

- **1 push-button + stop dimmer**  
See chapter 6 - "**1 push-button + stop dimmer**" function
- **Cyclic sending 1 push-button dimmer**  
See chapter 7 - "**Cyclic sending 1 push-button dimmer**" function
- **1 push-button shutter control**  
See chapter 8 - "**1 push-button shutter control**" function
- **2 push-button + stop dimmer**  
See chapter 9 - "**2 push-button + stop dimmer**" function
- **2 push-button dimmer with cyclic sending**  
See chapter 10 - "**2 push-button dimmer with cyclic sending**" function
- **2 push-button shutter control**  
See chapter 11 - "**2 push-button shutter control**" function
- **Scene management**  
See chapter 12 - "**Scene management**" function
- **Switching sequences**  
See chapter 13 - "**Switching sequences**" function
- **Multiple touches**  
See chapter 14 - "**Multiple touches**" function

## 5 “Edges/Sequence commands” function

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 4 commands per channel. It is possible to differentiate the command values according to the event detected (touch/release or short/long touch), and delay the sending of commands with a settable fixed time.

The basic structure of the menu is as follows (Fig. 5.1):



The screenshot shows a configuration interface for a KNX pushbutton module. The title bar reads "--.1 KNX pushbutton module 6 ch. with replac. symbols > Channel 1". The interface is divided into a left sidebar and a main configuration area. The sidebar has a "Main" section and a "Channel 1" section with a list of items: "Sound signalling", "Soft reduction", "Led 1", "Led 2", "Led 3", "Led 4", "Led 5", and "Led 6". The main area is currently displaying the configuration for "Led 6".

Configuration details for "Led 6":

- Long operation minimum time [x 0.1s]:** 5
- Block:**  disabled  enabled
- Matched function:** edges/sequence commands
- Functioning type recognized:**  edges (pressing/release)  short operation/long operation
- Object A:** (indicated by a horizontal line)
- Sending object format:** 1 bit
- Sending on touch/short operation detection:** 1
- Sending on release/long operation detection:** no action/stop cyclic sending object
- Object sending delay [s]:** 0
- Object B:** (indicated by a horizontal line)  disable  enable

Fig. 5.1

### 5.1 Parameters

#### ➤ 5.1.1 Type of recognised activation (Functioning type recognized)

Used to define which type of activation performed on the capacitive sensor must generate the sending of the sequence commands. The values that can be set are:

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

#### ➤ 5.1.2 Parameters in the “Object z” section (with z= A,B,C,D)

For each input, up to 8 different objects can be sent (distinguished by the letters A, B, C and D) on the basis of the touch (or short touch) or release (or long operation) of the capacitive sensor associated with the channel. Object A is always enabled, whereas the “**Object z**” parameter (where z is the index of the object associated with the channel, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

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

If **enable** is selected, the following parameters will be visualised: “**Sending object format**”, “**Sending on touch/short operation detection**”, “**Sending on release/long operation detection**” and “**Object sending delay [s]**”. These are grouped together in the **Object z** sub-group (where z is the index of the object associated with the input, between **A** and **D**).

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 bit
- 1 byte unsigned value
- 1 byte signed value
- 1 byte percentage value
- 1 byte HVAC mode
- 2 bytes unsigned value
- 2 bytes signed value
- 3 bytes RGB colour
- 4 bytes unsigned value
- 4 bytes signed value
- 14 bytes
- 

The value set for this item will condition the values that can be set for the “**Sending on touch/short operation detection**” and “**Sending on release/long operation detection**” parameters.

The “**Sending on touch/short operation detection**” parameter is used to set the command or value to send following the detection of a touch or short operation (depending on the type of action selected) by the capacitive sensor associated with the channel.

The “**Sending on release/long operation detection**” parameter is used to set the command or value to send following the detection of a release or long touch (depending on the type of action selected) by the capacitive sensor associated with the channel.

- 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** (touch detection default value)
- cyclical switching

Selecting the value **cyclical switching**, the “**Object status feedback**” parameter will be shown, which makes it possible to enable and display the **Ch.x - z object status feedback** communication object (Data Point Type: 1.001 DPT\_Switch); 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 recovered, 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 **Ch.x 2 bit value z object** communication object will be visible (Data Point Type: 2001 DPT\_Switch\_Control). The values that can be set for the two parameters above are:

- **no effect** (release detection default value)
- **activate forcing ON (down)** (touch detection default value)
- activate forcing OFF (up)
- deactivate forcing
- cyclical switching - ON forcing /OFF forcing
- cyclical switching - ON forcing/forcing deactivation
- cyclical switching - OFF forcing/forcing deactivation

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 value**, the **Ch.x - 1 byte value z object** communication object will be visible (Data Point Type: 5.010 DPT\_Value\_1\_Ucount) and the values that can be set for the two parameters listed above are:

- **no effect** (release detection default value)
- **send value** (touch detection default value)

By setting **send value**, it is possible to define the value to be sent via the newly 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 value**, the **Ch.x - 1 byte value z object** communication object will be visible (Data Point Type: 6.010 DPT\_Value\_1\_Count) and the values that can be set for the two parameters listed above are:

- **no effect** (release detection default value)
- **send value** (touch detection default value)

By setting **send value**, it is possible to define the value to be sent via the newly 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 value**, the **Ch.x - 1 byte value z object** communication object will be visible (Data Point Type: 5.001 DPT\_Scaling) and the values that can be set for the two parameters listed above are:

- **no effect** (release detection default value)
- **send value** (touch detection default value)

By setting **send value**, it is possible to define the value to be sent via the newly 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 **Ch.x - 1 byte value z object** communication object will be visible (Data Point Type: 20.102 DPT\_HVACMode) and the values that can be set for the two parameters listed above are:

- **no effect** (release default value)
- auto
- **comfort** (touch default value)
- pre-comfort

- economy
- off (building protection)
- cyclic switching (thermostat)
- cyclic 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 **cyclic switching (thermostat)**, each time the associated event (touch/release) is detected, the device will send a new thermoregulation mode command (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Comfort* ...; by selecting **cyclic switching (timed thermostat)**, each time the associated event (touch/release) is detected, the device will send a new thermoregulation mode command (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Auto*→ *Comfort* ...

- If the format of the object to send is **2 bytes unsigned value**, the **Ch.x 2 byte value z object** communication object will be visible (Data Point Type: 7.001 DPT\_Value\_2\_Ucount) and the values that can be set for the two parameters listed above are:

- **no effect** (release detection default value)
- **send value** (touch detection default value)

By setting **send value**, it is possible to define the value to be sent via the newly 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 value**, the **Ch.x 2 byte value z object** communication object will be visible (Data Point Type: 8.001 DPT\_Value\_2\_Count) and the values that can be set for the two parameters listed above are:

- **no effect** (release detection default value)
- **send value** (touch detection default value)

By setting **send value**, it is possible to define the value to be sent via the newly 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 **3 bytes RGB colour**, the **Ch.x 3 byte value z object** communication object will be visible (Data Point Type: 232.600 DPT\_Colour\_RGB) and the values that can be set for the two parameters above are:

- **no effect** (release detection default value)
- **send value** (touch detection default value)

By setting **send value**, you can select the colour to be sent via the “**Colour**” dummy parameter; The values that can be set are:

- **white** (default value)
- yellow
- magenta
- red
- turquoise
- green
- blue
- customize

By selecting **customize**, the following parameters are made visible: “**Value of RED component (0 .. 255)**”, “**Value of GREEN component (0 .. 255)**” and “**Value of BLUE component (0 .. 255)**”; The combination of the three colour components determines the actual value sent on the BUS. If you select

any of the other values, these parameters will still be visible but with pre-set values that cannot be modified.

The values that can be set are:

- from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **4 bytes unsigned value**, the **Ch.x 4 byte value z object** communication object will be visible (Data Point Type: 12.001 DPT\_Value\_4\_Ucount) and the values that can be set for the two parameters listed above are:
  - **no effect** (release detection default value)
  - **send value** (touch detection default value)

By setting **send value**, it is possible to define the value to be sent via the newly 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 byte signed value**, the **Ch.x 4 byte value z object** communication object will be visible (Data Point Type: 13.001 DPT\_Value\_4\_Count) and the values that can be set for the two parameters listed above are:
  - **no effect** (release detection default value)
  - **send value** (touch detection default value)

By setting **send value**, it is possible to define the value to be sent via the newly 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 **Ch.x 14 byte value z object** communication object will be visible (Data Point Type: 16.001 DPT\_String\_8859\_1) and the values that can be set for the two parameters listed above are:
  - **no effect** (release detection default value)
  - **send value** (touch detection default value)

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

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

**NOTE:** to remedy the problem of coding and the method for inserting values with 2/4 byte floating point format (DPT 9.0xx and 14.0xx), there is an external transformation tool that makes it possible to enter a value in the floating format and obtain the corresponding value with “unsigned and signed value” coding, and vice versa. In this way, the user obtains the value to be entered in the ETS database, selecting the format “2/4 byte signed/unsigned value”.

The “**Object sending delay (0.. 255 seconds)**” parameter sets the delay between the detection of the event associated with the sending of the command, and the effective sending of the command/value 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 in these cases is calculated from the moment when the command/value associated with the object with the previous index (z-1) is sent, not from the moment of detection of the event that generated the sending (touch/release or short/long operation).

The set delay will only be executed if the event in progress, associated with 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 sequence of commands with delays - activated by the detection of a specific event (touch/release or short/long touch) - is being sent, then the detection of the opposite event will cause the termination of the sending of that sequence, only if at least one of the actions associated with the detection of the latter event is different from **no effect**; otherwise, the command/value sequence will be continue to be sent until the last command/value has been sent.



## 6 “1 push-button + stop dimmer” function

This is used to configure the channel in order to control a dimmer with a single push-button, increasing and decreasing dimmer brightness with the same input.

It is possible to send on/off telegrams and brightness dimming telegrams.

As there is only one channel to manage the On/Off and brightness dimming functions, the operation is managed by differentiating between short touches and long touches:

- a long touch (operation) is transformed into a brightness dimming command. When released, a regulation 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 regulation command was received. With the technology used, the maximum touch duration that the device can recognise is about 20 seconds.
- a short touch is transformed into an on/off command.

Using this type of function, brightness dimming depends on the so-called brightness dimming 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 device, but the latter regulates the brightness itself by stopping its increase/decrease based on the desired value. The communication objects enabled by this function are **Ch.x - Switching** (Data Point Type: 1.001 DPT\_Switch) and **Ch.x - Brightness dimming** (Data Point Type: 3.007 DPT\_Control\_Dimming).

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

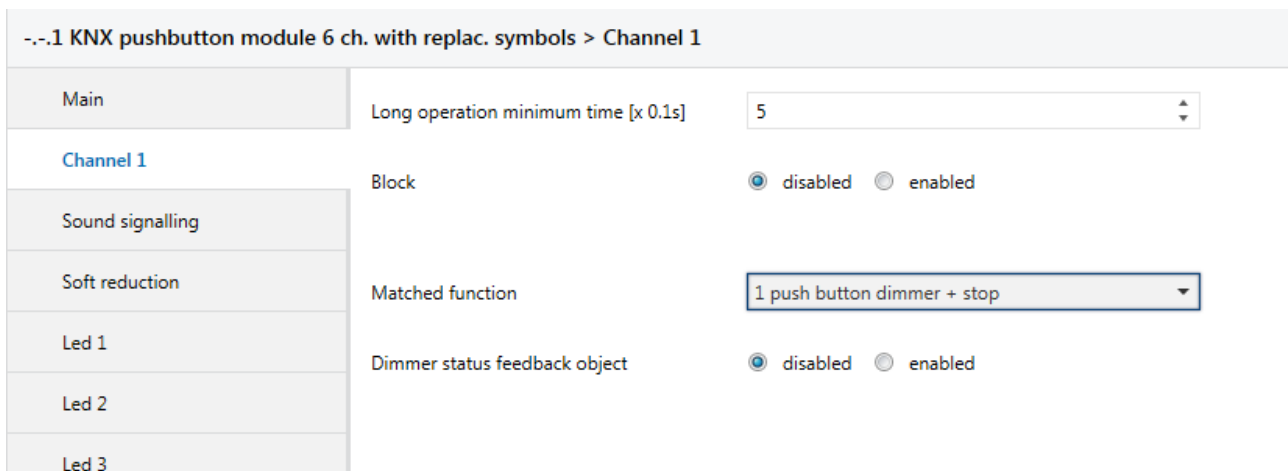


Fig. 6.1

### 6.1 Parameters

The push-button panel makes sure that the command to be sent is the opposite of the last command that was sent, which results as:

- long touch (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 command was an on command or an increase brightness command, the new one will be a decrease brightness command. In both cases, when released, a regulation 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 regulation command was received.
- short touch: 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 dimming 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.

### ➤ 6.1.1 Dimmer status feedback object

This parameter may have the following values:

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

If **enabled** is selected, the “**Brightness dimming commands with dimmer on**” parameter is visible along with the **Ch.x - Dimmer status feedback** communication object (Data Point Type: 1.001 DPT\_Switch), which is used to receive the status feedback from the controlled dimmer actuator. The behaviour of the push-button panel is modified as follows:

- long touch (operation): the command that the push-button panel sends depend on the parameter “**Brightness dimming 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 dimming command to be sent will be the opposite of the last sent command; when released, a regulation 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 regulation 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 touch: 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 recovered, the device sends a status reading command for this object to update the push-button panel about the status of the connected devices.

## 7 “Cyclic sending 1 push-button dimmer” function

This is used to configure the channel to control a dimmer with a single push-button, increasing and decreasing dimmer brightness always using the same push-button, with defined and settable regulation steps.

As there is only one push-button to manage the On/Off and brightness dimming functions, the operation is managed in the following way: with each touch, the command sent is the opposite of the last one sent. Furthermore, a differentiation is made between short touches and long touches:

a long touch (operation) is transformed into a brightness dimming command. No telegram is sent when released. With the technology used, the maximum touch duration that the device can recognise is about 20 seconds.

- a short touch is transformed into an on/off command.

Unlike the **1 push-button + stop dimmer**, it is possible to define both the brightness variation steps and the time that must elapse between the sending of one command and another when the long touch (operation) is drawn out over time; the sending of the regulation stop telegram at the end of the operation is not therefore necessary, because although the regulation does follow the characteristic power/brightness curve, it is the command sent from the push-button panel that determines the percentage variation. The communication objects enabled by this function are **Ch.x - Switching** (Data Point Type: 1.001 DPT\_Switch) and **Ch.x - Brightness dimming** (Data Point Type: 3.007 DPT\_Control\_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 touch (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 the touch is maintained, the device sends the command cyclically until it detects the release.

### ➤ **7.1.2 Cyclical sending period [x 0.1s]**

Used to set the time that must elapse between the sending of one increase/decrease command and another, when the operation is maintained. When released, no telegram is sent but only the cyclical sending of the brightness dimming commands is stopped.

The values that can be set for this parameter are:

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

To sum up, when a long touch (operation) is detected, the device sends the first increase/decrease command with the set percentage and, if the touch is maintained, it sends the command cyclically until it detects the release.

EXAMPLE: let's suppose we've set a value of **0.5 sec** for the item **Long operation minimum time** in the **Main** menu, a value of **12.5%** for the **Increase/decrease step** parameter, and a value of **3** (0.3 sec) for the **Cyclical sending period [x 0.1s]** parameter, and the operation is detected:

- 0.5 seconds after the initial detection of the touch (operation), a long touch (operation) is detected and so the first 12.5% brightness increase/decrease telegram is sent
- from this moment, for every 0.3 seconds that the touch (operation) is maintained, the device sends the 12.5% brightness increase/decrease command again, continuing until it detects the release
- when released, no telegram is sent but the cyclical sending is stopped.

### ➤ **7.1.3 Dimmer status feedback object**

Refer to paragraph 6.1.1 for the settings of this parameter

## 8 “1 push-button shutter control” function

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

As only one channel manages the louvre up/down and control functions, operation is managed so that with each activation, a command is sent that is the opposite to the last movement received by the actuator that manages the shutter. There is a difference between short and long touches:

- a long touch (operation) is transformed into an up/down movement command. The new value to be sent is the opposite of the last value sent via the **Ch.x - Shutter movement** object or of the movement feedback received via the **Ch.x - Movement feedback** object, depending on which of the two events occurred last. If the last event that occurred is “upward movement feedback reception” or “sending upward movement command”, the new command will be a “downward movement” command and vice versa.
- a short operation is transformed into a louvre control command. The new value to be sent depends on the last value sent via the **Ch.x - Shutter movement** object or the movement feedback received via the **Ch.x - Movement feedback** object, depending on which of the two events occurred last; if the last event that occurred is “upward movement feedback reception” or “send upward movement command”, the command will be a “closing louvres adjustment” command, and vice versa. If the shutter is moving, the louvre adjustment command will only stop the shutter up/down movement.

The communication objects enabled by this function are **Ch.x - Shutter movement** (Data Point Type: 1.008 DPT\_UpDown), **Ch.x - Shutter stop/louvres control** (Data Point Type: 1.007 DPT\_Step) and **Ch.x - Movement feedback** (Data Point Type: 1.008 DPT\_UpDown).

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

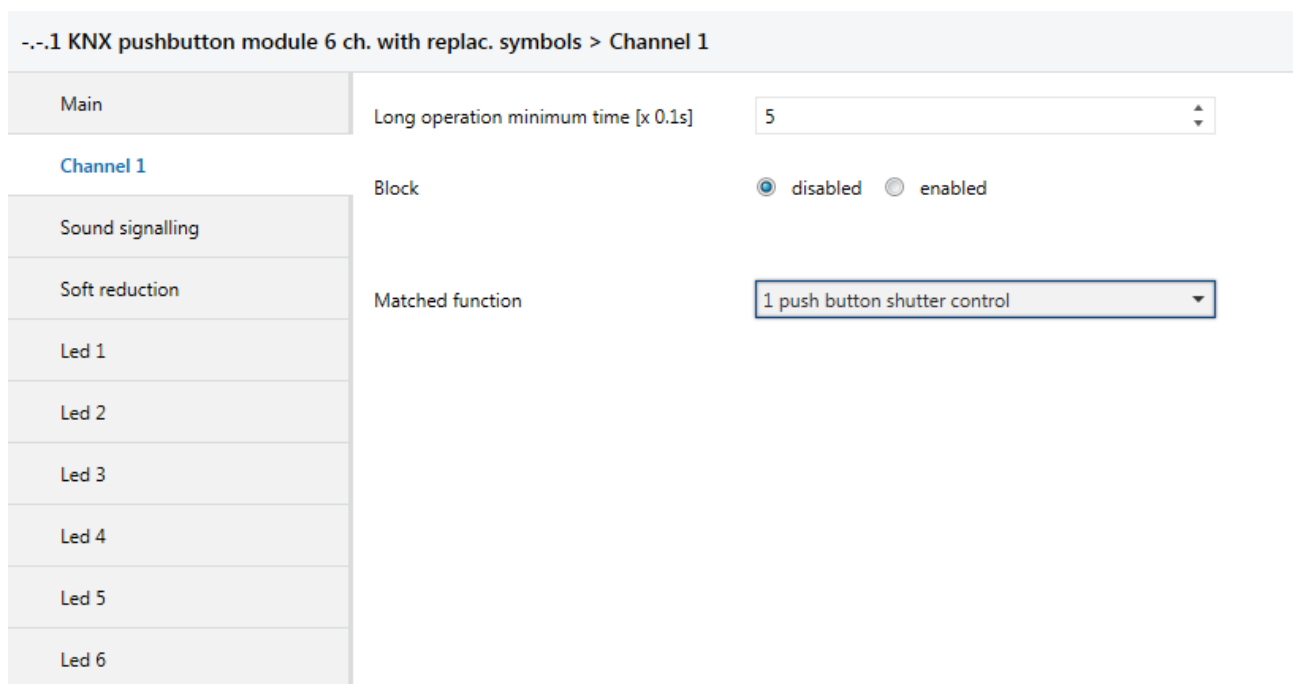


Fig. 8.1

### 8.1 Parameters

No new parameters are enabled with this function

## 9 “2 push-button + stop dimmer” function

This is used to configure the channel to control a dimmer with two push-buttons, managing in this case only one of the two regulation directions (brightness increase/decrease).

On or off telegrams and brightness increase or decrease dimming telegrams can be sent, based on the configured regulation direction. Also in this case, there is a difference between short and long touches:

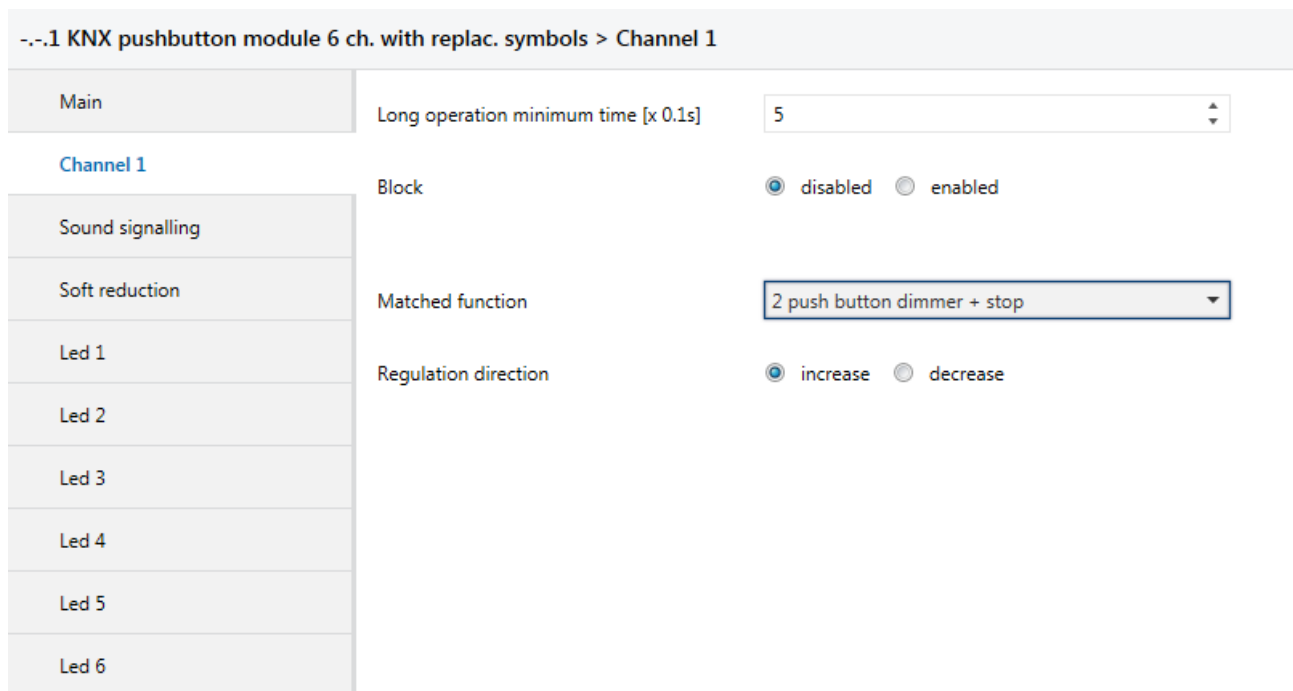
- a long touch (operation) is transformed into a brightness dimming command. If the set regulation direction is "increase", the regulation will only be increasing, otherwise if the set regulation direction is "decrease" the regulation will be decreasing. In both cases, when released, a regulation stop telegram is sent to stop the brightness increase or decrease operation for the dimmer and to fix the brightness value reached at the moment the stop regulation command was received.

With the technology used, the maximum touch duration that the device can recognise is about 20 seconds.

- a short touch is transformed in to an on or off command depending on the set regulation direction. If the set regulation direction is "increase" the sent command will only be an ON command. If the set regulation direction is "decrease" the sent command will only be an OFF command.

Using this type of function, brightness dimming depends on the so-called brightness dimming characteristic curve, which varies from device to device, based on how the manufacturer designed the curve that regulates power, and as a result brightness. The communication objects enabled by this function are **Ch.x - Switching** (Data Point Type: 1.001 DPT\_Switch) and **Ch.x - Brightness dimming** (Data Point Type: 3.007 DPT\_Control\_Dimming).

The structure of the menu is as follows:



--.1 KNX pushbutton module 6 ch. with replac. symbols > Channel 1		
Main	Long operation minimum time [x 0.1s]	5
Channel 1	Block	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Sound signalling		
Soft reduction	Matched function	2 push button dimmer + stop
Led 1	Regulation direction	<input checked="" type="radio"/> increase <input type="radio"/> decrease
Led 2		
Led 3		
Led 4		
Led 5		
Led 6		

Fig. 9.1

## 9.1 Parameters

### ➤ 9.1.1 Regulation direction

The parameter “**Regulation direction**” configures the regulation direction of the brightness that the channel controls; The values that can be set are:

- **increase** (uneven channel default value)
- **decrease** (even channel default value)

Selecting **increase**, the sent commands will be 'increase brightness 100%' or ON, depending on the recognized activation; otherwise, selecting **decrease** the sent commands will be 'decrease brightness 100%' or OFF.

## 10 “2 push-button dimmer with cyclic sending” function

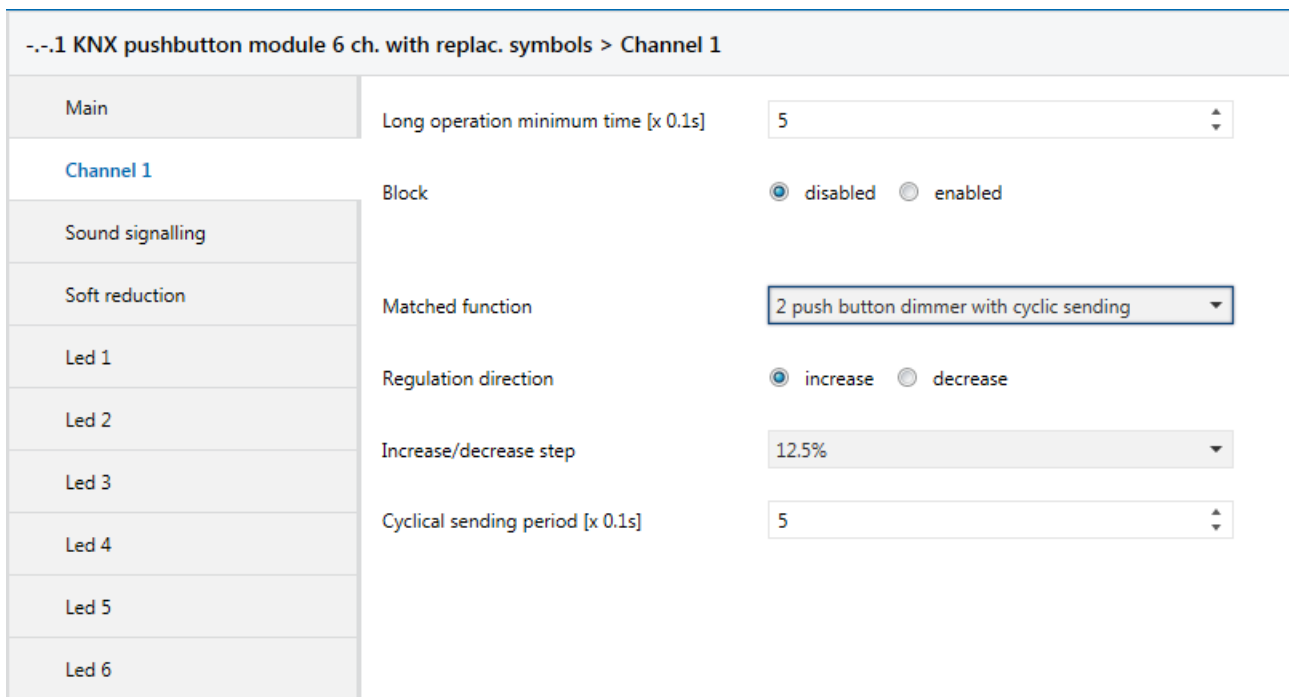
This is used to configure the channel to control a dimmer with two push-buttons, managing in this case only one of the two regulation directions (brightness increase/decrease).

On or off telegrams and brightness increase or decrease dimming telegrams can be sent, based on the configured regulation direction. Also in this case, there is a difference between short and long touches:

- a long touch (operation) is transformed into a brightness dimming command. If the set regulation direction is "increase", the regulation will only be increasing, otherwise if the set regulation direction is "decrease" the regulation will be decreasing. In both cases, no telegram is sent upon release. With the technology used, the maximum touch duration that the device can recognise is about 20 seconds.
- a short touch is transformed in to an on or off command depending on the set regulation direction. If the set regulation direction is "increase" the sent command will only be an ON command. If the set regulation direction is "decrease" the sent command will only be an OFF command.

Unlike the **2 push-button + stop dimmer** function, it is possible to define both the brightness variation steps of the increase/decrease commands and the time that must elapse between the sending of one command and another when the long touch (operation) is drawn out over time. The sending of the regulation stop telegram on push-button release is not therefore necessary, because although the regulation does follow the characteristic power/brightness curve, it is the command sent by the device that determines the percentage variation. The communication objects enabled by this function are **Ch.x - Switching** (Data Point Type: 1.001 DPT\_Switch) and **Ch.x - Brightness dimming**(Data Point Type: 3.007 DPT\_Control\_Dimming).

The structure of the menu is as follows:



-.-.1 KNX pushbutton module 6 ch. with replac. symbols > Channel 1		
Main	Long operation minimum time [x 0.1s]	5
Channel 1	Block	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Sound signalling		
Soft reduction	Matched function	2 push button dimmer with cyclic sending
Led 1	Regulation direction	<input checked="" type="radio"/> increase <input type="radio"/> decrease
Led 2	Increase/decrease step	12.5%
Led 3		
Led 4	Cyclical sending period [x 0.1s]	5
Led 5		
Led 6		

Fig. 10.1



## 10.1 Parameters

### ➤ 10.1.1 Regulation direction

The parameter “**Regulation direction**” configures the regulation direction of the brightness that the channel controls; The values that can be set are:

- **Increase** (uneven channel default value)
- **Decrease** (even channel default value)

Selecting **increase**, the sent commands will be 'increase brightness 100%' or ON, depending on the recognized activation; otherwise, selecting **decrease** the sent commands will be 'decrease brightness 100%' or OFF.

### ➤ 10.1.2 Increase/decrease step

The “**Increase/decrease step**” parameter 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 touch (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 the touch is maintained, the device sends the command cyclically until it detects the release.

### ➤ 10.1.3 Cyclical sending period [x 0.1s]

The “**Cyclical sending period [x 0.1s]**” parameter defines the time that passes between the sending of one increase/decrease command and the next, if the pressure on the push-button is maintained. When released, no telegram is sent but only the cyclical sending of the brightness dimming commands is stopped.

The values that can be set for the “**Cyclical sending period [x 0.1s]**” parameter are:

- from 3 to 50 with steps of 1 (default value 5)

To sum up, when a long touch (operation) is detected, the device sends the first increase/decrease command with the set percentage and, if the touch is maintained, it sends the command cyclically until it detects the release.

EXAMPLE: let's suppose we've set a value of **0.5 sec** for the **Long operation minimum time** item in the **Channel x** menu, a value of **12.5%** for the **Increase/decrease step** parameter, and a value of **3** (0.3 sec) for the **Cyclical sending period [x 0.1s]** parameter, and an operation is then detected.

- 0.5 seconds after the initial detection of the touch (operation), a long touch (operation) is detected and so the first 12.5% brightness increase/decrease telegram is sent
- from this moment, for every 0.3 seconds that the touch (operation) is maintained, the device sends the 12.5% brightness increase/decrease command again, continuing until it detects the release
- when released, no telegram is sent but the cyclical sending is stopped

## 11 “2 push-button shutter control” function

This is used to configure the channel to control a shutter/Venetian blind with two push-buttons, managing in this case only one of the two movement directions (down or up).

Up or down movement telegrams or louvres open or close control telegrams can be sent. Also in this case, there is a difference between short and long touches:

- a long touch (operation) is transformed into a movement command. If the set movement direction is "up", the movement will only be up; vice versa if the set direction is "down" the movement will be down. When released, the device will not perform any action.
- a short touch is transformed into a louvres control command (stop movement if the shutter is moving), depending on the set movement direction. If the set movement direction is "up" the sent command will only be a louvres opening control command (or stop movement); If the set regulation direction is "down" the sent command will only be a louvres closing control command (or stop movement).

The communication objects enabled by this function are **Ch.x - Shutter movement** (Data Point Type: 1.008 DPT\_UpDown) and **Ch.x - Shutter stop/adjustment** (Data Point Type: 1.007 DPT\_Step).

The structure of the menu is as follows:

-.1 KNX pushbutton module 6 ch. with replac. symbols > Channel 1

Main	Long operation minimum time [x 0.1s]	5
Channel 1	Block	<input checked="" type="radio"/> disabled <input type="radio"/> enabled
Sound signalling		
Soft reduction	Matched function	2 push button shutter control
Led 1	Movement direction	<input checked="" type="radio"/> up <input type="radio"/> down
Led 2		
Led 3		
Led 4		
Led 5		
Led 6		

Fig. 11.1

### 11.1 Parameters

#### ➤ 11.1.1 Movement direction

The “**Movement direction**” parameter is used to configure the direction of movement of the shutter controlled by the channel. The values that can be set are:

- **up** (uneven channel default value)
- **down** (even channel default value)

selecting **up**, the sent commands will be up movement or louvres opening control (stop movement), depending on the recognised activation; vice versa, selecting **down**, the sent commands will be down movement or louvres closing control (stop movement).

## 12 “Scene management” function

This is used to configure the channel to send scene memorising and execution commands, with the possibility of sending the scene memorising command following a command received from the BUS. Only one scene can be managed for each channel.

There is a difference between short and long touches:

- a long touch (operation) is transformed into a scene storing command.
- a short operation is transformed into a scene execution command.

The communication objects enabled by this function are **Ch.x - Scene** (Data Point Type: 18.001 DPT\_SceneControl) and **Ch.x - Scene storing trigger** (Data Point Type: 1.017 DPT\_Trigger).

The structure of the menu is as follows (Fig. 12.1):

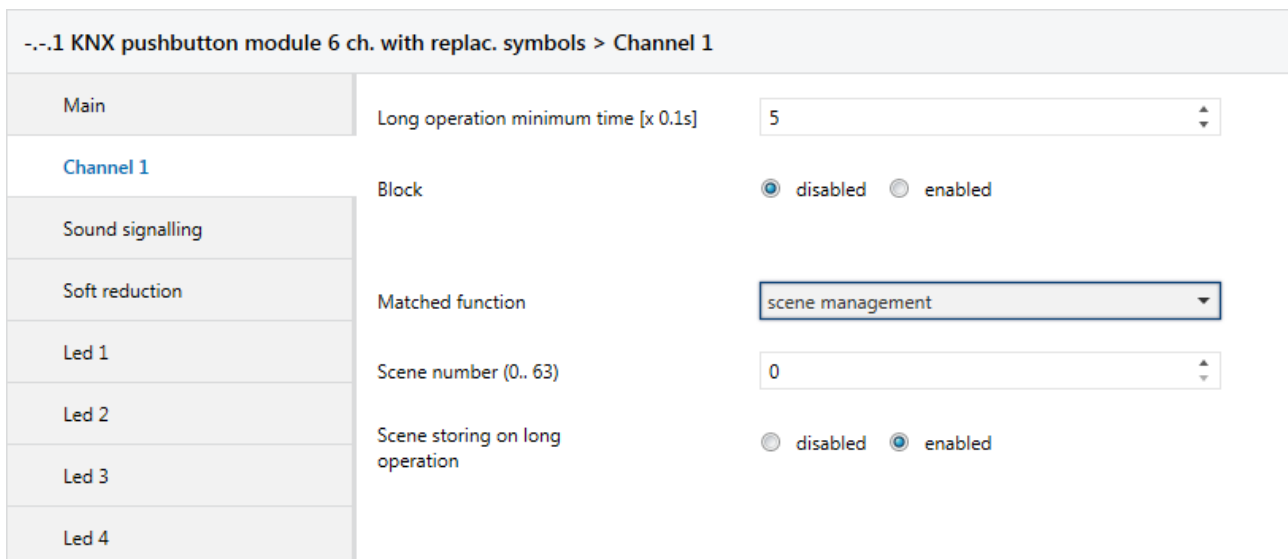


Fig. 12.1

### 12.1 Parameters

#### ➤ 12.1.1 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, with steps of 1

#### ➤ 12.1.2 Scene storing on long operation

Used to enable the sending of the scene storing command when a long touch (operation) is detected. The values that can be set are:

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

If **enabled** is selected, the device will send the scene storing command when a long touch (operation) is detected; if **disabled** is selected, a long touch (operation) is not recognised and only causes the sending of the scene execution command (like the short operation).

Regardless of the value set for the parameter above, it is possible to indirectly generate the sending of the scene storing command following the arrival of a BUS telegram on the object **Ch.x - Scene storing trigger** (both with a value of “1” and with a value of “0”); each time the device receives a telegram on this object, the scene storing telegram will immediately be sent.

## 13 “Switching sequences” function

Used to send a sequence of commands following the detection of a specific touch.  
The structure of the menu is as follows (Fig. 13.1):

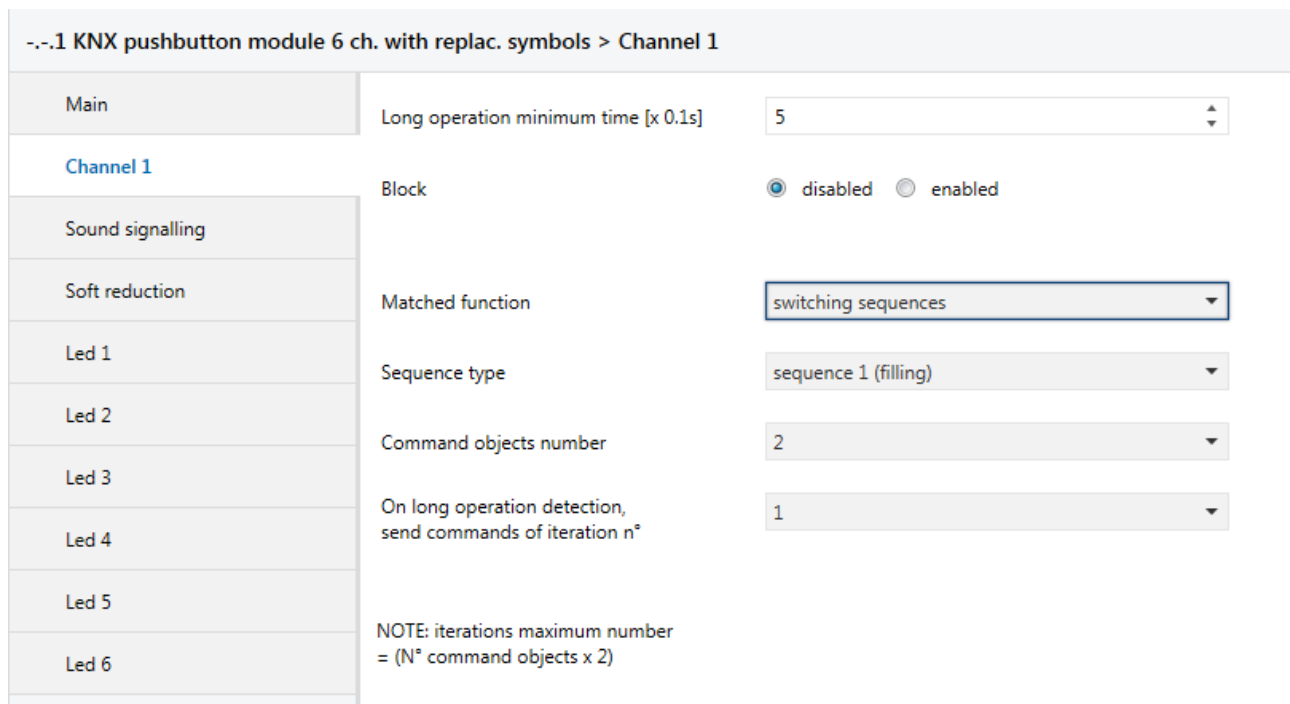


Fig. 13.1

### 13.1 Parameters

#### ➤ 13.1.1 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 (free)

#### Sequence 1 (filling):

with each operation (edge) detected, the device sends a sequence to the communication object enabled. This sequence follows the filling progress. It 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 <b>Ch.x - C sequence</b>	Value sent on <b>Ch.x - B sequence</b>	Value sent on <b>Ch.x - A sequence</b>
<b>1st edge</b>	0	0	1
<b>2nd edge</b>	0	1	1
<b>3rd edge</b>	1	1	1
<b>4th edge</b>	0	1	1
<b>5th edge</b>	0	0	1
<b>6th edge</b>	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):**

with each operation (edge) detected, the device sends a sequence to the communication object enabled. This sequence follows the sum progress. 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 <i>Ch.x - C sequence</i>	Value sent on <i>Ch.x - B sequence</i>	Value sent on <i>Ch.x - A sequence</i>
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 again from the beginning

The table shows how the trend of the sent commands depends on the count of the detected edge; it begins with the binary code of value 1 and moves up (in this specific case) to the code of value 7 then, from the next edge, the count begins again. 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 (free)** 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 with steps of 1

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 **Channel x z object** menu is as follows:

--.1 KNX pushbutton module 6 ch. with replac. symbols > Channel 1 A object		
Main	Iteration 1 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Channel 1	Iteration 2 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
<b>Channel 1 A object</b>		
Channel 1 B object	Iteration 3 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Sound signalling	Iteration 4 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Soft reduction	Iteration 5 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Led 1	Iteration 6 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Led 2	Iteration 7 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Led 3	Iteration 8 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Led 4	Iteration 9 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Led 5	Iteration 10 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Led 6	Iteration 11 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
	Iteration 12 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
	Iteration 13 object value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value

Fig. 13.2

### ➤ 13.1.2 Command objects number

This is used to set the number of commands that make up the sequence itself; Depending on the value set for this item, the **Ch.x - Sequence z** communication objects are enabled (Data Point Type: 1.001 DPT\_Switch), with **z** included between A and H. The values that can be set are:

from **2 (default value)** to 4, with steps of 1

### ➤ 13.1.3 On long operation detection, send commands of iteration n°

Regardless of the type of sequence selected, the “**On long operation detection, send commands of iteration n°**” parameter is used to define which sequence iteration to send if a long touch (operation) is detected. The values that can be set are:

- from 1 to 16 with steps of 1, (**default value 1**)

EXAMPLE: with reference to the above tables, let's suppose that the value set by the user is **3**. When a long touch (operation) is detected, the device will send:

Edge no.	Value sent on <i>Ch.x - C sequence</i>	Value sent on <i>Ch.x - B sequence</i>	Value sent on <i>Ch.x - A sequence</i>
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 <i>Ch.x - C sequence</i>	Value sent on <i>Ch.x - B sequence</i>	Value sent on <i>Ch.x - A sequence</i>
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 a long touch (operation) has been detected and the sequence relating to the set iteration has been sent, then when the next short operation is detected, the sequence relating to the iteration immediately after the one associated with the long touch (operation) will be sent (in the example given here, the sequence associated with iteration no. 4 will be sent).

To sum up, the value set for the **"On long operation detection, send commands of iteration n"** parameter defines both the sequence to be sent and the value with which to initialise the iterations counter when a long touch (operation) is detected.

Make sure the selected iteration number associated with the sequence to be sent with a long operation is less than - or equal to - the maximum number of iterations associated with the sequence; otherwise, the iteration to be taken into consideration is the limit one.

## 14 “Multiple touches” function

This function is used to set the type and number of commands to send after a series of consecutive pressing operations has been detected, for up to 4 commands per channel.

The structure of the menu is as follows:

The screenshot shows the configuration interface for a KNX pushbutton module. The title bar reads "--.1 KNX pushbutton module 6 ch. with replac. symbols > Channel 1". The left sidebar contains a navigation menu with the following items: Main, Channel 1, Sound signalling, Soft reduction, Led 1, Led 2, Led 3, Led 4, Led 5, and Led 6. The main configuration area for Channel 1 includes the following settings:

- Long operation minimum time [x 0.1s]: 5
- Block:  disabled  enabled
- Matched function: multiple touches
- Maximum interval between two consecutive touches [x 0.1s]: 3
- Sends objects:  at every detected touch  only at the end of touches counting
- Single touch detection:  disabled  enabled
- Double touch detection:  disabled  enabled
- Triple touch detection:  disabled  enabled
- Quadruple touch detection:  disabled  enabled
- Long touch detection:  disabled  enabled

Fig. 14.1

In this mode, every channel can send a series of KNX telegrams following the detection of several consecutive pressing operations. In particular, the device is able to distinguish between the following consecutive pressings:

- single touch → one touch detected
- double touch → two consecutive touches detected
- triple touch → three consecutive touches detected
- quadruple touch → four consecutive touches detected
- long touch (operation) → long touch (operation) detected

Five consecutive touches or more are interpreted as a “quadruple touch”.

In order to recognise two consecutive touches, it is necessary to define the maximum interval between the detection of one touch and the next; if the time between two consecutive touches is less than the maximum interval, the multiple touch count is increased. When the time that elapses after the detection of a touch exceeds the maximum interval, the device recognises a number of consecutive multiple touches equal to the value counted and, after sending the telegrams associated with this action, it resets the counter.



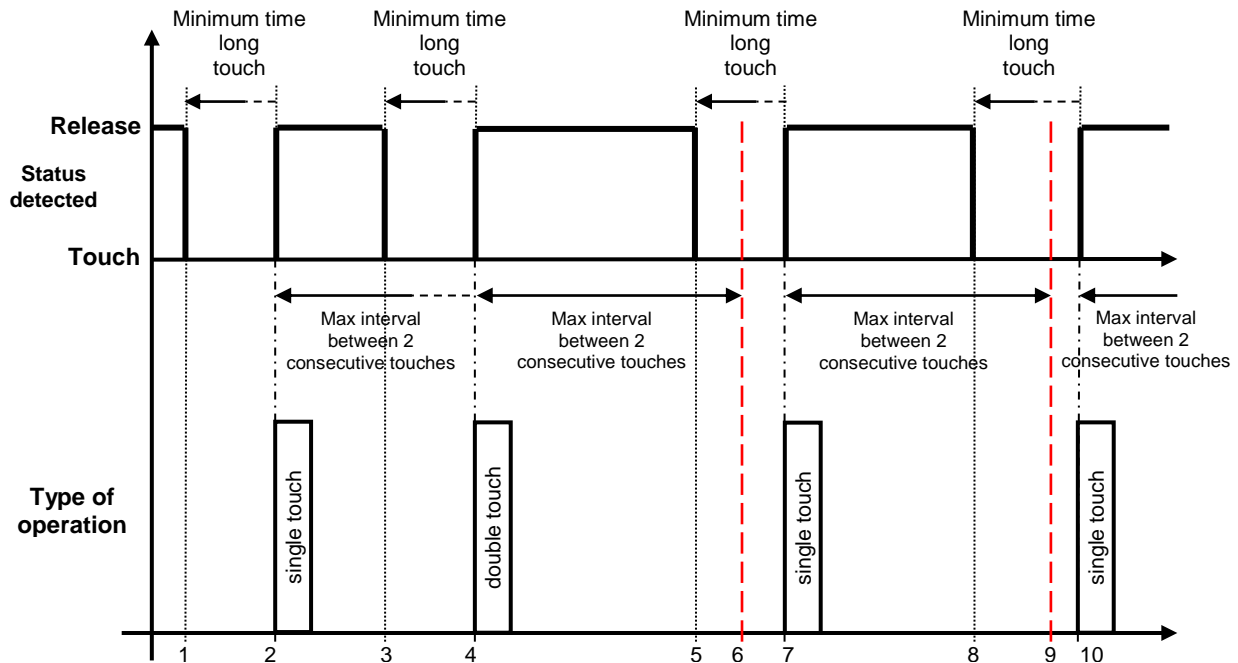
## 14.1 Parameters

### ➤ 14.1.1 Maximum interval between two consecutive touches [x 0.1s]

The “Maximum interval between two consecutive touches [x 0.1s]” parameter defines the maximum interval between the detection of one touch and the next, so that they are recognised as consecutive touches. The values that can be set are:

- from **3 (default value)** to 100, with steps of 1

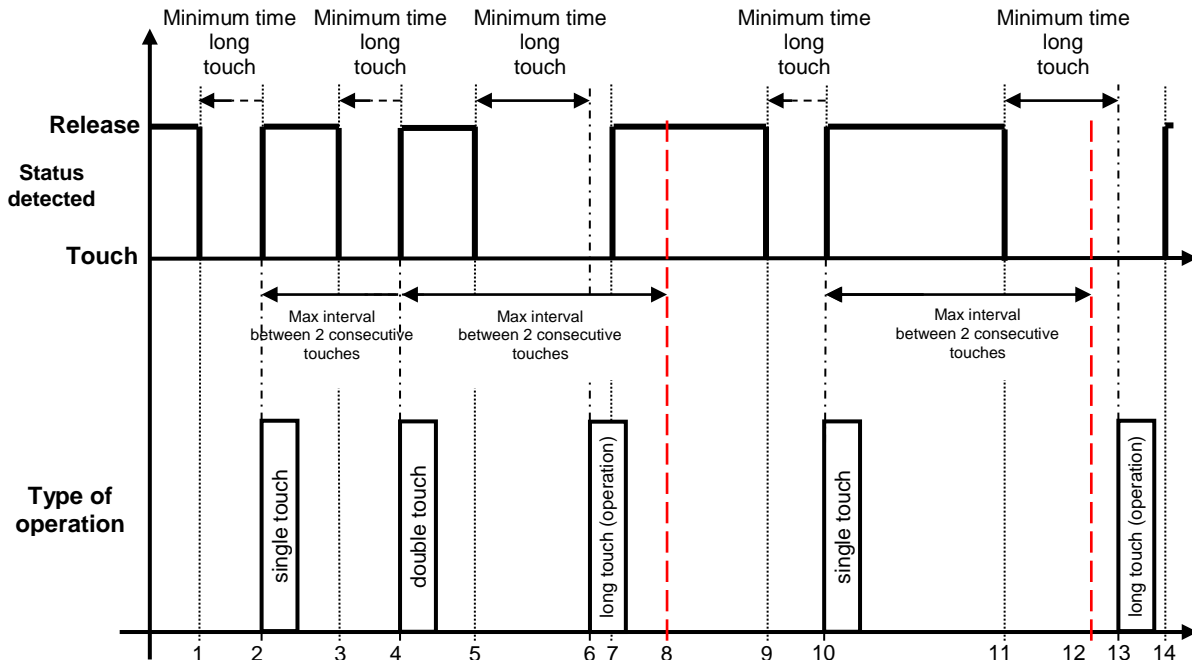
The following chart shows some situations that summarise the concept of multiple touches.



1. Once the touch has been detected, the count of the touch duration is begun in order to differentiate between a short touch and a long one
2. If the button is released before the end of the long touch (operation) minimum time, a short touch is recognised; the count of the interval between two consecutive touches is begun, and the multiple touch count is increased
3. The detection of a new touch triggers the start of the touch duration count in order to differentiate between a short touch and a long one, but does not alter the count of the interval between two consecutive touches or the multiple touch counter in any way
4. If the button is released before the end of the long touch (operation) minimum time and before reaching the maximum interval between two consecutive touches, a new short touch is detected; this increases the multiple touch counter and restarts the count of the interval between two consecutive touches
5. A new touch triggers the start of the touch duration count in order to differentiate between a short touch and a long one, but does not alter the count of the interval between two consecutive touches or the multiple touch counter in any way
6. Once the maximum interval between two consecutive touches (dotted red line) has elapsed, the multiple touch count is terminated and, after sending the KNX commands relating to this action, the counter is reset
7. If the button is released before the end of the long touch (operation) minimum time, a new short touch is recognised; the multiple touch count is increased and the count of the interval between two consecutive touches is begun
8. A new touch triggers the start of the touch duration count in order to differentiate between a short touch and a long one, but does not alter the count of the interval between two consecutive touches or the multiple touch counter in any way
9. Once the maximum interval between two consecutive touches (dotted red line) has elapsed, the multiple touch count is terminated and, after sending the KNX commands relating to this action, the counter is reset

- If the button is released before the end of the long touch (operation) minimum time, a new short touch is recognised; the multiple touch count is increased and the count of the interval between two consecutive touches is begun

The detection of a long touch (operation) in no way alters the multiple press count or any calculation of the interval between two consecutive touches, even if the minimum duration of the long touch (operation) is less than the maximum interval between two consecutive touches See below.



- Once the touch has been detected, the count of the touch duration is begun in order to differentiate between a short touch and a long one
- If the button is released before the end of the long touch (operation) minimum time, a short touch is recognised; the count of the interval between two consecutive touches is begun, and the multiple touch count is increased
- The detection of a new touch triggers the start of the touch duration count in order to differentiate between a short touch and a long one, but does not alter the count of the interval between two consecutive touches or the multiple touch counter in any way
- If the button is released before the end of the long touch (operation) minimum time and before reaching the maximum interval between two consecutive touches, a new short touch is detected; this increases the multiple touch counter and restarts the count of the interval between two consecutive touches
- A new touch triggers the start of the touch duration count in order to differentiate between a short touch and a long one, but does not alter the count of the interval between two consecutive touches or the multiple touch counter in any way
- If the touch is maintained for a time greater than the minimum duration of a long touch (operation), a long touch (operation) is recognised and the KNX commands for that action are sent, but neither the calculation of the interval between two consecutive touches nor the multiple touch count is modified in any way
- The release of the button after the recognition of the long touch (operation) does not trigger any action
- Once the maximum interval between two consecutive touches (dotted red line) has elapsed, the multiple touch count is terminated and, after sending the KNX commands relating to this action, the counter is reset
- The detection of a new touch triggers the start of the touch duration count in order to differentiate between a short touch and a long one, but does not alter the count of the interval between two consecutive touches or the multiple touch counter in any way
- If the button is released before the end of the long touch (operation) minimum time, a new short touch is recognised; the multiple touch count is increased and the count of the interval between two consecutive touches is begun
- A new touch triggers the start of the touch duration count in order to differentiate between a short touch and a long one, but does not alter the count of the interval between two consecutive touches or the multiple touch counter in any way

12. Once the maximum interval between two consecutive touches (dotted red line) has elapsed, the multiple touch count is terminated and, after sending the KNX commands relating to this action, the counter is reset
13. If the touch is maintained for a time greater than the minimum duration of a long touch (operation), a long touch (operation) is recognised and the KNX commands for that action are sent, but neither the calculation of the interval between two consecutive touches nor the multiple touch count is modified in any way
14. The release of the button after the recognition of the long touch (operation) does not trigger any action

### ➤ 14.1.2 Send objects

The commands associated with the “multiple touch” function can be sent in two different ways:

- a) the device waits for the interval between two consecutive touches to exceed the maximum value, consequently interrupting the multiple touch count and sending the commands associated with the number of touches detected;
- b) every time the multiple touch count is increased, the device sends the telegrams associated with the number of touches detected.

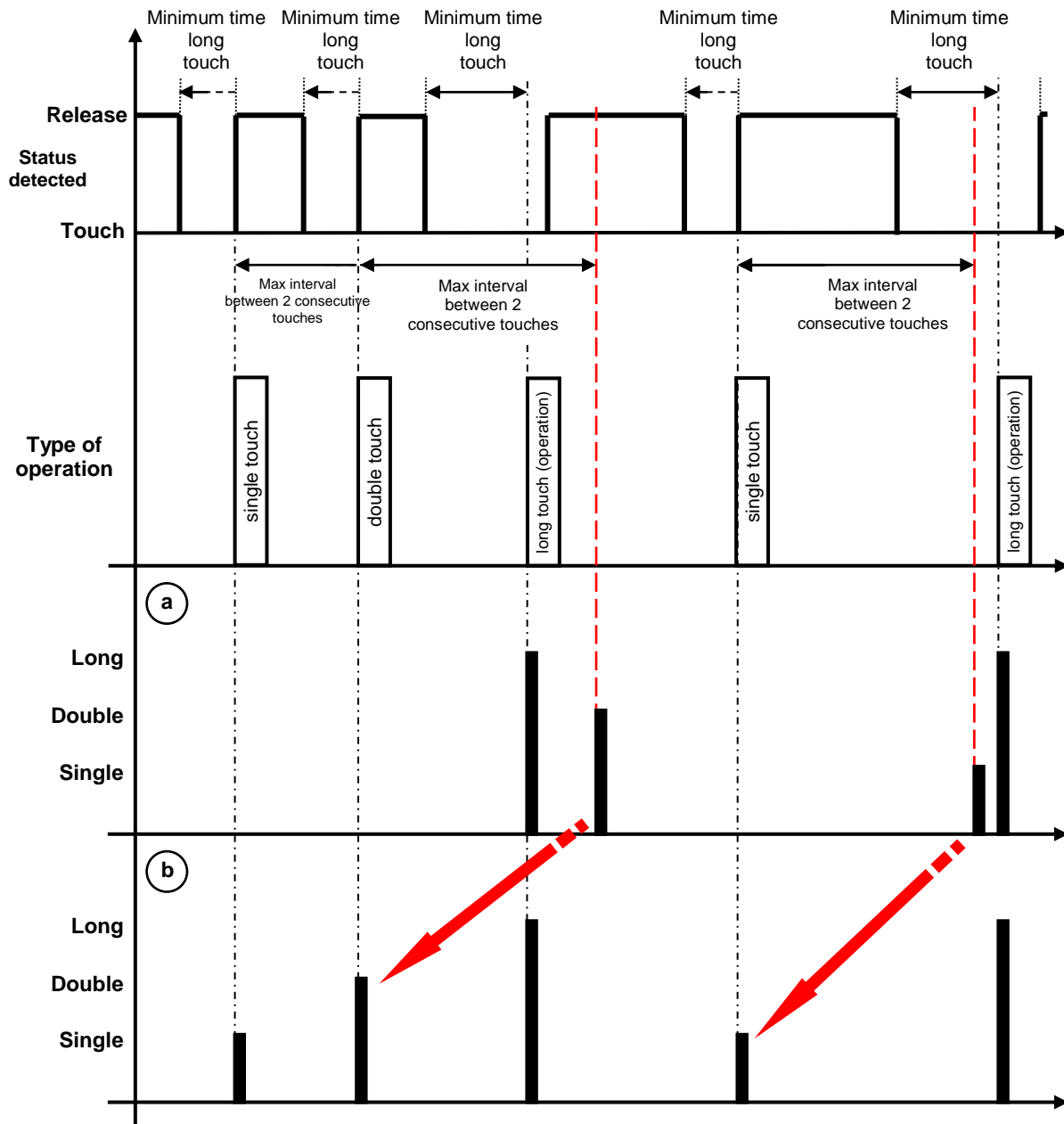
The commands associated with a “long touch” (operation) are always sent as soon as the long touch is detected.

The “**Send objects**” parameter defines the sending conditions for the objects associated with multiple touches. The values that can be set are:

- at every detected touch
- **only at the end of touch counting (default value)**

Selecting **only at the end of touch counting**, the device behaves as described in point “a” above; selecting **at every detected touch**, the device behaves as described in point “b” above.

The following chart summarises the behaviour of the device on the basis of the set sending condition.



The chart resumes the situation shown previously, introducing the long touch (operation) and its effect on counters and timers. The two sections at the bottom show the commands sent on the KNX BUS if the sending is **only at the end of touch counting** (case “a”) or **at every detected touch** (case “b”). The main difference between the two cases is that in case “b”, every time a multiple touch is counted the associated telegrams are sent, whereas in case “a” it is necessary to wait until the time between two consecutive touches exceeds the maximum value in order to terminate the count and send the telegrams associated with the value it has reached.

The red arrows highlight the differences between the moments when the telegrams associated with the multiple touches are actually sent.

### ➤ 14.1.3 Single touch detection

The “**Single touch detection**” parameter is used to enable the recognition of a single touch and visualise the **Channel x - Single touch** menu for enabling and configuring the commands that will be sent following the recognition of a single touch. The values that can be set are:

- disabled (default value)
- enabled

Selecting **enabled** displays the **Channel x - Single touch** menu (see the “Channel x - Single touch” menu).

➤ **14.1.4 Double touch detection**

The “**Double touch detection**” parameter is used to enable the recognition of a double touch and visualise the **Channel x - Double touch** menu for enabling and configuring the commands that will be sent following the recognition of a double touch. The values that can be set are:

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

Selecting **enabled** displays the **Channel x - Double touch** menu (see the “Channel x - Double touch” menu).

➤ **14.1.5 Triple touch detection**

The “**Triple touch detection**” parameter is used to enable the recognition of a triple touch and visualise the **Channel x - Triple touch** menu for enabling and configuring the commands that will be sent following the recognition of a triple touch. The values that can be set are:

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

Selecting **enabled** displays the **Channel x - Triple touch** menu (see the “Channel x - Triple touch” menu).

➤ **14.1.6 Quadruple touch detection**

The “**Quadruple touch detection**” parameter is used to enable the recognition of a quadruple touch and visualise the **Channel x - Quadruple touch** menu for enabling and configuring the commands that will be sent following the recognition of a quadruple touch. The values that can be set are:

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

Selecting **enabled** displays the **Channel x - Quadruple touch** menu (see the “Channel x - Quadruple touch” menu).

➤ **14.1.7 Long touch detection**

The “**Long touch detection**” parameter is used to enable the recognition of a long touch and visualise the **Channel x - Long touch** menu for enabling and configuring the commands that will be sent following the recognition of a long touch. The values that can be set are:

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

Selecting **enabled** displays the **Channel x - Long touch** menu (see the “Channel x - Long touch” menu).

### ➤ 14.1.8 Channel x – Single touch menu

This menu is visible if the value of the “**Single touch detection**” parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the “single touch” event.

The structure of the menu is as follows:

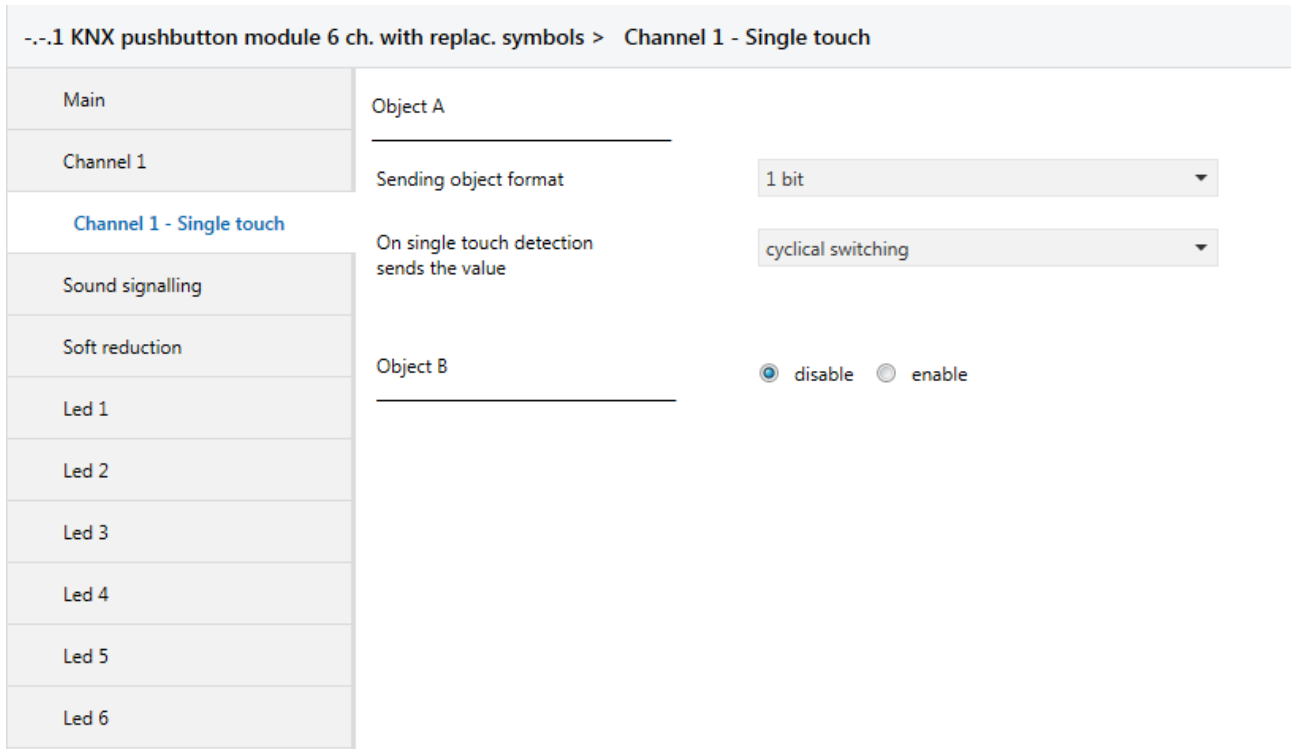


Fig. 14.2

Upon detection of the single touch, it is possible to send up to 4 different objects (which are distinguished by the letters A, B, C and D). Object A is always enabled, whereas the “**Object z**” parameter (where z is the index of the object associated with the channel, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

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

Selecting **enable** displays the “**Sending object format**” and “**On single touch detection sends the value**” parameters found in the sub-group **Object z** (where z is the index of the object associated with the channel, included between **A** and **D**).

The parameter “**Sending object format**” makes it possible to set the format and code of the object “z” of input “x” that is sent by the device. The values that can be set are:

- **1 bit (default value)**
- 2 bit
- 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

- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the “**On single touch detection sends the value**” parameter will be different.

The “**On single touch detection sends the value**” parameter is used to set the command or value to send following the detection of a single touch (on the basis of the set sending conditions) associated with the channel. The values that can be set are:

- If the format of the object to send is **1 bit**, the **Ch.x - Single touch 1 bit object z** communication object will be visible (Data Point Type: 1.002 DPT\_Bool) and the values that can be set for the above parameter are:

- 0
- 1
- **cyclical switching 3 (default value)**

Selecting the value **cyclical switching**, the command that the device sends (via the **Ch.x - Single touch 1 bit object z** object) upon detection of the event associated with cyclical switching will be the opposite of the value generated by the most recent event - the value received via BUS on the **Ch.x - Single touch 1 bit object z** object or the last value sent.

In this configuration, as there is no specific communication object for receiving the status feedback from the BUS, the **Ch.x - Single touch 1 bit object z** object is an input/output object. With KNX technology, a communication object is sent to a single destination group address, so if this object is associated with more than one group address, the device will send the BUS telegram to the group address where the object has the “S” (sending) flag. Vice versa, the device will update its value when a BUS telegram is received on any group address associated with the object, regardless of the “S” flag.

In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the status of the devices connected.

- If the format of the object to send is **2 bit**, the **Ch.x - Single touch 2 bit object z** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control) and the values that can be set for the above parameter are:

- activate forcing ON (down)
- activate forcing OFF (up)
- deactivate forcing
- cyclical switching - ON forcing /OFF forcing
- **cyclical switching - forcing ON / deactivate forcing (default value)**
- cyclical switching - OFF forcing/forcing deactivation

Selecting the value **cyclical switching**, the command that the device sends (via the object **Ch.x - Single touch 2 bit object z**) upon detection of the event associated with cyclical switching will be the opposite of the last value sent (via the **Ch.x - Single touch 2 bit object z** object).

- If the format of the object to send is **1 byte unsigned value**, the **Ch.x - Single touch 1 byte object z** communication object will be visible (Data Point Type: 5.010 DPT\_Value\_1\_Ucount) and the values that can be set for the above parameter are:

- from **0 (default value)** to 255, with steps of 1

- If the format of the object to send is **1 byte signed value**, the **Ch.x - Single touch 1 byte object z** communication object will be visible (Data Point Type: 6.010 DPT\_Value\_1\_Count) and the values that can be set for the above parameter are:

- from -128 to 127 with steps of 1, **0 (default value)**

- If the format of the object to send is **1 byte percentage value**, the **Ch.x - Single touch 1 byte object z** communication object will be visible (Data Point Type: 5.001 DPT\_Scaling) and the values that can be set for the above parameter are:

- from **0 (default value)** to 100, with steps of 1
- If the format of the object to send is **1 byte HVAC mode**, the **Ch.x - Single touch 1 byte object z** communication object will be visible (Data Point Type: 20.102 DPT\_HVACMode) and the values that can be set for the above parameter are:
  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

By selecting **cyclical switching (thermostat)**, each time the associated event (single touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Comfort* ...; by selecting **cyclical switching (timed thermostat)**, each time the associated event (single touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Auto*→ *Comfort* ...

- If the format of the object to send is **2 byte unsigned value**, the **Ch.x - Single touch 2 byte object z** communication object will be visible (Data Point Type: 7.001 DPT\_Value\_2\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 65535, with steps of 1
- If the format of the object to send is **2 byte signed value**, the **Ch.x - Single touch 2 byte object z** communication object will be visible (Data Point Type: 8.001 DPT\_Value\_2\_Count) and the values that can be set for the above parameter are:
  - from -32768 to +32767 with steps of 1, **0 (default value)**
- If the format of the object to send is **3 byte RGB colour**, the **“On single touch detection sends the value”** parameter is a dummy one, used to select the colour to be sent. The real value, downloaded into the memory, will depend on the three parameters that represent the colour components (see below). In addition, the **Ch.x - Single touch 3 byte object z** communication object will be visible (Data Point Type: 232.600 DPT\_Colour\_RGB) and the values that can be set for the above parameter are:
  - **white (default value)**
  - yellow
  - magenta
  - red
  - turquoise
  - green
  - blue
  - customize

By selecting **customize**, the following parameters are made visible: **“Value of RED component (0 .. 255)”**, **“Value of GREEN component (0 .. 255)”** and **“Value of BLUE component (0 .. 255)”**; The combination of the three colour components determines the actual value sent on the BUS. If you select any of the other values, these parameters will still be visible but with pre-set values that cannot be modified.

The values that can be set are:

- from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **4 byte unsigned value**, the **Ch.x - Single touch 4 byte object z** communication object will be visible (Data Point Type: 12.001 DPT\_Value\_4\_Ucount) and the values that can be set for the above parameter are:



- from **0 (default value)** to 4294967295, with steps of 1
- If the format of the object to send is **4 byte signed value**, the **Ch.x - Single touch 4 byte object z** communication object will be visible (Data Point Type: 13.001 DPT\_Value\_4\_Count) and the values that can be set for the above parameter are:
  - from -2147483648 to 2147483647 with steps of 1 (**default value 0**)

**NOTE:** to remedy the problem of coding and the method for inserting values with 2/4 byte floating point format (DPT 9.0xx and 14.0xx), there is an external transformation tool that makes it possible to enter a value in the floating format and obtain the corresponding value with “unsigned and signed value” coding, and vice versa. In this way, the user obtains the value to be entered in the ETS database, selecting the format “2/4 byte signed/unsigned value”.

### ➤ 14.1.9 Channel x – Double touch menu

This menu is visible if the value of the “**Double touch detection**” parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the “double touch” event.

The structure of the menu is as follows:

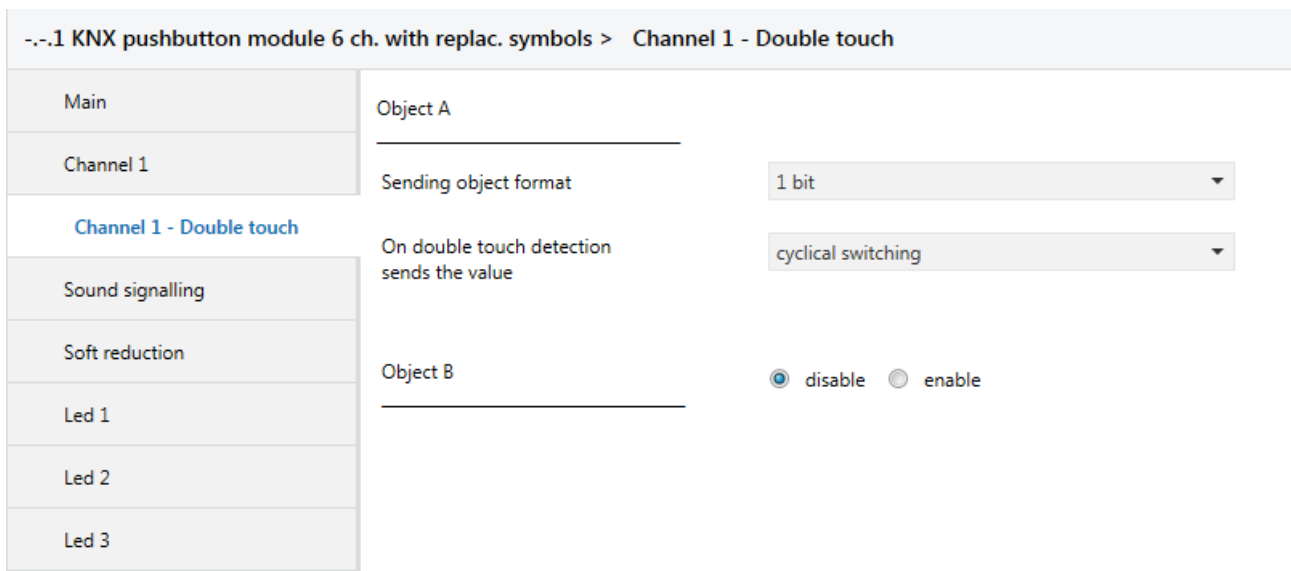


Fig. 14.3

Upon detection of the double touch, it is possible to send up to 4 different objects (which are distinguished by the letters A, B, C and D). Object A is always enabled, whereas the “**Object z**” parameter (where z is the index of the object associated with the channel, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

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

Selecting **enable** displays the “**Sending object format**” and “**On double touch detection sends the value**” parameters found in the sub-group **Object z** (where z is the index of the object associated with the channel, included between **A** and **D**).

The parameter “**Sending object format**” makes it possible to set the format and code of the object “z” of input “x” that is sent by the device. The values that can be set are:

- **1 bit (default value)**
- 2 bit
- 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
- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the “**On double touch detection sends the value**” parameter will be different.

The “**On double touch detection sends the value**” parameter is used to set the command or value to send following the detection of a double touch (on the basis of the set sending conditions) associated with the channel. The values that can be set are:

- If the format of the object to send is **1 bit**, the **Ch.x - Double touch 1 bit object z** communication object will be visible (Data Point Type: 1.002 DPT\_Bool) and the values that can be set for the above parameter are:

- 0 1
- 1 2
- **cyclical switching 3 (default value)**

Selecting the value **cyclical switching**, the command that the device sends (via the **Ch.x - Double touch 1 bit object z** object) upon detection of the event associated with cyclical switching will be the opposite of the value generated by the most recent event - the value received via BUS on the **Ch.x - Double touch 1 bit object z** object or the last value sent.

In this configuration, as there is no specific communication object for receiving the status feedback from the BUS, the **Ch.x - Double touch 1 bit object z** object is an input/output object. With KNX technology, a communication object is sent to a single destination group address, so if this object is associated with more than one group address, the device will send the BUS telegram to the group address where the object has the “S” (sending) flag. Vice versa, the device will update its value when a BUS telegram is received on any group address associated with the object, regardless of the “S” flag.

In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the status of the devices connected.

- If the format of the object to send is **2 bit**, the **Ch.x - Double touch 2 bit object z** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control) and the values that can be set for the above parameter are:

- activate forcing ON (down)
- activate forcing OFF (up)
- deactivate forcing
- cyclical switching - ON forcing /OFF forcing
- **cyclical switching - forcing ON / deactivate forcing (default value)**
- cyclical switching - OFF forcing/forcing deactivation

Selecting the value **cyclical switching**, the command that the device sends (via the object **Ch.x - Double touch 2 bit object z**) upon detection of the event associated with cyclical switching will be the opposite of the last value sent (via the **Ch.x - Double touch 2 bit object z** object).

- If the format of the object to send is **1 byte unsigned value**, the **Ch.x - Double touch 1 byte object z** communication object will be visible (Data Point Type: 5.010 DPT\_Value\_1\_Ucount) and the values that can be set for the above parameter are:

- from **0 (default value)** to 255, with steps of 1

- If the format of the object to send is **1 byte signed value**, the **Ch.x - Double touch 1 byte object z** communication object will be visible (Data Point Type: 6.010 DPT\_Value\_1\_Count) and the values that can be set for the above parameter are:
  - from -128 to 127 with steps of 1, **0 (default value)**
- If the format of the object to send is **1 byte percentage value**, the **Ch.x - Double touch 1 byte object z** communication object will be visible (Data Point Type: 5.001 DPT\_Scaling) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 100, with steps of 1
- If the format of the object to send is **1 byte HVAC mode**, the **Ch.x - Double touch 1 byte object z** communication object will be visible (Data Point Type: 20.102 DPT\_HVACMode) and the values that can be set for the above parameter are:
  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

By selecting **cyclical switching (thermostat)**, each time the associated event (double touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Comfort* ...; by selecting **cyclical switching (timed thermostat)**, each time the associated event (double touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Auto*→ *Comfort* ...

- If the format of the object to send is **2 byte unsigned value**, the **Ch.x - Double touch 2 byte object z** communication object will be visible (Data Point Type: 7.001 DPT\_Value\_2\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 65535, with steps of 1
- If the format of the object to send is **2 byte signed value**, the **Ch.x - Double touch 2 byte object z** communication object will be visible (Data Point Type: 8.001 DPT\_Value\_2\_Count) and the values that can be set for the above parameter are:
  - from -32768 to +32767 with steps of 1, **0 (default value)**
- If the format of the object to send is **3 byte RGB colour**, the “**On double touch detection sends the value**” parameter is a dummy one, used to select the colour to be sent. The real value, downloaded into the memory, will depend on the three parameters that represent the colour components (see below). In addition, the **Ch.x - Double touch 3 byte object z** communication object will be visible (Data Point Type: 232.600 DPT\_Colour\_RGB) and the values that can be set for the above parameter are:
  - **white (default value)**
  - yellow
  - magenta
  - red
  - turquoise
  - green
  - blue
  - customize

By selecting **customize**, the following parameters are made visible: “**Value of RED component (0 .. 255)**”, “**Value of GREEN component (0 .. 255)**” and “**Value of BLUE component (0 .. 255)**”; The

combination of the three colour components determines the actual value sent on the BUS. If you select any of the other values, these parameters will still be visible but with pre-set values that cannot be modified.

The values that can be set are:

- from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **4 byte unsigned value**, the **Ch.x - Double touch 4 byte object z** communication object will be visible (Data Point Type: 12.001 DPT\_Value\_4\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 4294967295, with steps of 1
- If the format of the object to send is **4 byte signed value**, the **Ch.x - Double touch 4 byte object z** communication object will be visible (Data Point Type: 13.001 DPT\_Value\_4\_Count) and the values that can be set for the above parameter are:
  - from -2147483648 to 2147483647 with steps of 1 (**default value 0**)

**NOTE:** to remedy the problem of coding and the method for inserting values with 2/4 byte floating point format (DPT 9.0xx and 14.0xx), there is an external transformation tool that makes it possible to enter a value in the floating format and obtain the corresponding value with “unsigned and signed value” coding, and vice versa. In this way, the user obtains the value to be entered in the ETS database, selecting the format “2/4 byte signed/unsigned value”.

#### ➤ 14.1.10 Channel x – Triple touch menu

This menu is visible if the value of the “**Triple touch detection**” parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the “triple touch” event.

The structure of the menu is as follows:

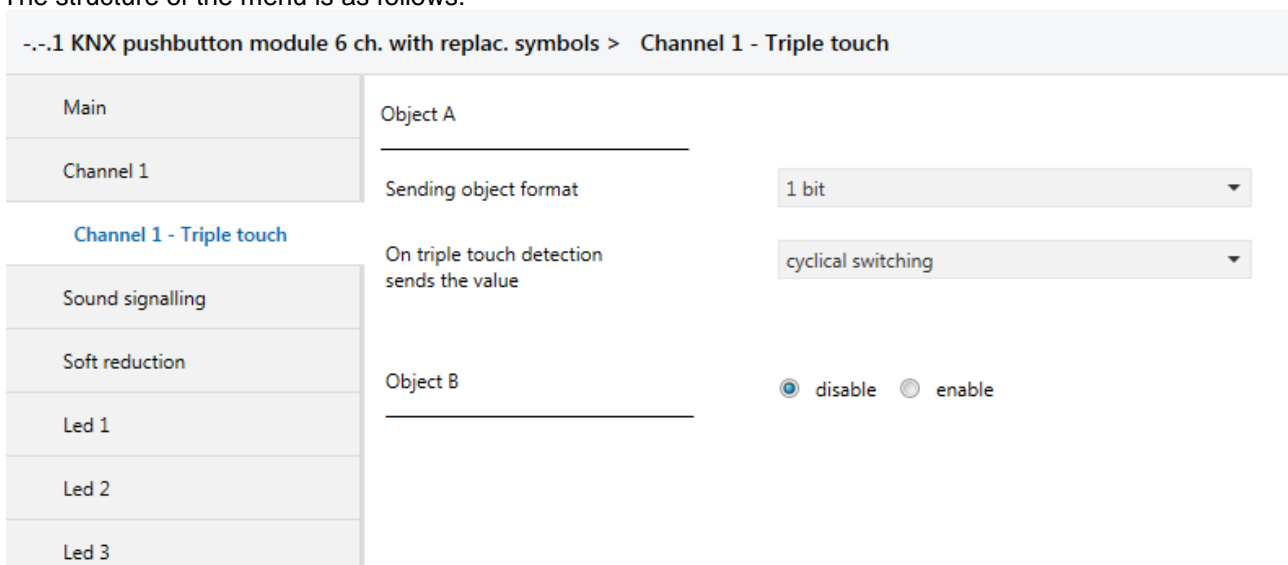


Fig. 14.4

Upon detection of the triple touch, it is possible to send up to 4 different objects (which are distinguished by the letters A, B, C and D). Object A is always enabled, whereas the “**Object z**” parameter (where z is the index of the object associated with the threshold, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

- **disable** (default value)

- enable

Selecting **enable** displays the “**Sending object format**” and “**On triple touch detection sends the value**” parameters found in the sub-group **Object z** (where z is the index of the object associated with the input, included between **A** and **D**).

The parameter “**Sending object format**” makes it possible to set the format and code of the object “z” of input “x” that is sent by the device. The values that can be set are:

- **1 bit (default value)**
- 2 bit
- 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
- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the “**On triple touch detection sends the value**” parameter will be different.

The “**On triple touch detection sends the value**” parameter is used to set the command or value to send following the detection of a triple touch (on the basis of the set sending conditions) associated with the channel. The values that can be set are:

- If the format of the object to send is **1 bit**, the **Ch.x - Triple touch 1 bit object z** communication object will be visible (Data Point Type: 1.002 DPT\_Bool) and the values that can be set for the above parameter are:
  - 0
  - 1
  - **cyclical switching** (default value)

Selecting the value **cyclical switching**, the command that the device sends (via the **Ch.x - Triple touch 1 bit object z** object) upon detection of the event associated with cyclical switching will be the opposite of the value generated by the most recent event - the value received via BUS on the **Ch.x - Triple touch 1 bit object z** object or the last value sent.

In this configuration, as there is no specific communication object for receiving the status feedback from the BUS, the **Ch.x - Triple touch 1 bit object z** object is an input/output object. With KNX technology, a communication object is sent to a single destination group address, so if this object is associated with more than one group address, the device will send the BUS telegram to the group address where the object has the “S” (sending) flag. Vice versa, the device will update its value when a BUS telegram is received on any group address associated with the object, regardless of the “S” flag.

In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the status of the devices connected.

- If the format of the object to send is **2 bit**, the **Ch.x - Triple touch 2 bit object z** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control) and the values that can be set for the above parameter are:
  - activate forcing ON (down)
  - activate forcing OFF (up)
  - deactivate forcing
  - cyclical switching - ON forcing /OFF forcing
  - **cyclical switching - forcing ON / deactivate forcing** (default value)
  - cyclical switching - OFF forcing/forcing deactivation

Selecting the value **cyclical switching**, the command that the device sends (via the object **Ch.x - Triple touch 2 bit object z**) upon detection of the event associated with cyclical switching will be the opposite of the last value sent (via the **Ch.x - Triple touch 2 bit object z** object).

- If the format of the object to send is **1 byte unsigned value**, the **Ch.x - Triple touch 1 byte object z** communication object will be visible (Data Point Type: 5.010 DPT\_Value\_1\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **1 byte signed value**, the **Ch.x - Triple touch 1 byte object z** communication object will be visible (Data Point Type: 6.010 DPT\_Value\_1\_Count) and the values that can be set for the above parameter are:
  - from -128 to 127 with steps of 1, **0 (default value)**
- If the format of the object to send is **1 byte percentage value**, the **Ch.x - Triple touch 1 byte object z** communication object will be visible (Data Point Type: 5.001 DPT\_Scaling) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 100, with steps of 1
- If the format of the object to send is **1 byte HVAC mode**, the **Ch.x - Triple touch 1 byte object z** communication object will be visible (Data Point Type: 20.102 DPT\_HVACMode) and the values that can be set for the above parameter are:
  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

By selecting **cyclical switching (thermostat)**, each time the associated event (triple touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Comfort* ...; by selecting **cyclical switching (timed thermostat)**, each time the associated event (triple touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Auto*→ *Comfort* ...

- If the format of the object to send is **2 byte unsigned value**, the **Ch.x - Triple touch 2 byte object z** communication object will be visible (Data Point Type: 7.001 DPT\_Value\_2\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 65535, with steps of 1
- If the format of the object to send is **2 byte signed value**, the **Ch.x - Triple touch 2 byte object z** communication object will be visible (Data Point Type: 8.001 DPT\_Value\_2\_Count) and the values that can be set for the above parameter are:
  - from -32768 to +32767 with steps of 1, **0 (default value)**
- If the format of the object to send is **3 byte RGB colour**, the “**On triple touch detection sends the value**” parameter is a dummy one, used to select the colour to be sent. The real value, downloaded into the memory, will depend on the three parameters that represent the colour components (see below). In addition, the **Ch.x - Triple touch 3 byte object z** communication object will be visible (Data Point Type: 232.600 DPT\_Colour\_RGB) and the values that can be set for the above parameter are:
  - **white (default value)**

- yellow
- magenta
- red
- turquoise
- green
- blue
- customize

By selecting **customize**, the following parameters are made visible: “**Value of RED component (0 .. 255)**”, “**Value of GREEN component (0 .. 255)**” and “**Value of BLUE component (0 .. 255)**”; The combination of the three colour components determines the actual value sent on the BUS. If you select any of the other values, these parameters will still be visible but with pre-set values that cannot be modified.

The values that can be set are:

- from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **4 byte unsigned value**, the **Ch.x - Triple touch 4 byte object z** communication object will be visible (Data Point Type: 12.001 DPT\_Value\_4\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 4294967295, with steps of 1
- If the format of the object to send is **4 byte signed value**, the **Ch.x - Triple touch 4 byte object z** communication object will be visible (Data Point Type: 13.001 DPT\_Value\_4\_Count) and the values that can be set for the above parameter are:
  - from -2147483648 to 2147483647 with steps of 1 (**default value 0**)

**NOTE:** to remedy the problem of coding and the method for inserting values with 2/4 byte floating point format (DPT 9.0xx and 14.0xx), there is an external transformation tool that makes it possible to enter a value in the floating format and obtain the corresponding value with “unsigned and signed value” coding, and vice versa. In this way, the user obtains the value to be entered in the ETS database, selecting the format “2/4 byte signed/unsigned value”.

### ➤ 14.1.11 Channel x – Quadruple touch menu

This menu is visible if the value of the “**Quadruple touch detection**” parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the “quadruple touch” event.

The structure of the menu is as follows:

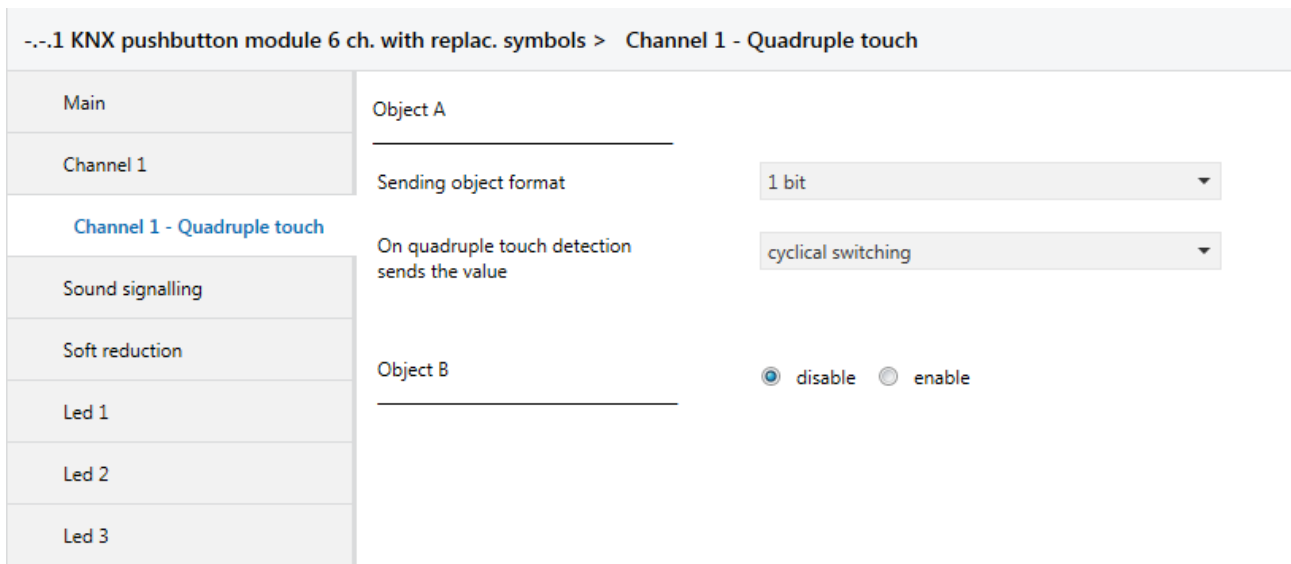


Fig. 14.5

Upon detection of the quadruple touch, it is possible to send up to 4 different objects (which are distinguished by the letters A, B, C and D). Object A is always enabled, whereas the “**Object z**” parameter (where z is the index of the object associated with the threshold, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

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

Selecting **enable** displays the “**Sending object format**” and “**On quadruple touch detection sends the value**” parameters found in the sub-group **Object z** (where z is the index of the object associated with the input, included between **A** and **D**).

The parameter “**Sending object format**” makes it possible to set the format and code of the object “z” of input “x” that is sent by the device. The values that can be set are:

- **1 bit (default value)**
- 2 bit
- 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
- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the “**On quadruple touch detection sends the value**” parameter will be different.

The “**On quadruple touch detection sends the value**” parameter is used to set the command or value to send following the detection of a quadruple touch (on the basis of the set sending conditions) associated with the input. The values that can be set are:

- If the format of the object to send is **1 bit**, the **Ch.x - Quadruple touch 1 bit object z** communication object will be visible (Data Point Type: 1.002 DPT\_Bool) and the values that can be set for the above parameter are:
  - 0
  - 1
  - **cyclical switching 3** (default value)



Selecting the value **cyclical switching**, the command that the device sends (via the **Ch.x - Quadruple touch 1 bit object z** object) upon detection of the event associated with cyclical switching will be the opposite of the value generated by the most recent event - the value received via BUS on the **Ch.x - Quadruple touch 1 bit object z** object or the last value sent.

In this configuration, as there is no specific communication object for receiving the status feedback from the BUS, the **Ch.x - Quadruple touch 1 bit object z** object is an input/output object. With KNX technology, a communication object is sent to a single destination group address, so if this object is associated with more than one group address, the device will send the BUS telegram to the group address where the object has the "S" (sending) flag. Vice versa, the device will update its value when a BUS telegram is received on any group address associated with the object, regardless of the "S" flag. In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the status of the devices connected.

- If the format of the object to send is **2 bit**, the **Ch.x - Quadruple touch 2 bit object z** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control) and the values that can be set for the above parameter are:
  - activate forcing ON (down)
  - activate forcing OFF (up)
  - deactivate forcing
  - cyclical switching - ON forcing /OFF forcing
  - **cyclical switching - forcing ON / deactivate forcing** (default value)
  - cyclical switching - OFF forcing/forcing deactivation

Selecting the value **cyclical switching**, the command that the device sends (via the object **Ch.x - Quadruple touch 2 bit object z**) upon detection of the event associated with cyclical switching will be the opposite of the last value sent (via the **Ch.x - Quadruple touch 2 bit object z** object).

- If the format of the object to send is **1 byte unsigned value**, the **Ch.x - Quadruple touch 1 byte object z** communication object will be visible (Data Point Type: 5.010 DPT\_Value\_1\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **1 byte signed value**, the **Ch.x - Quadruple touch 1 byte object z** communication object will be visible (Data Point Type: 6.010 DPT\_Value\_1\_Count) and the values that can be set for the above parameter are:
  - from -128 to 127 with steps of 1, **0 (default value)**
- If the format of the object to send is **1 byte percentage value**, the **Ch.x - Quadruple touch 1 byte object z** communication object will be visible (Data Point Type: 5.001 DPT\_Scaling) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 100, with steps of 1
- If the format of the object to send is **1 byte HVAC mode**, the **Ch.x - Quadruple touch 1 byte object z** communication object will be visible (Data Point Type: 20.102 DPT\_HVACMode) and the values that can be set for the above parameter are:
  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat)** (default value)
  - cyclical switching (timed thermostat)

By selecting **cyclical switching (thermostat)**, each time the associated event (quadruple touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Comfort* ...; by selecting **cyclical switching (timed thermostat)**, each time the associated event (quadruple touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Auto*→ *Comfort* ...

- If the format of the object to send is **2 byte unsigned value**, the **Ch.x - Quadruple touch 2 byte object z** communication object will be visible (Data Point Type: 7.001 DPT\_Value\_2\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 65535, with steps of 1
- If the format of the object to send is **2 byte signed value**, the **Ch.x - Quadruple touch 2 byte object z** communication object will be visible (Data Point Type: 8.001 DPT\_Value\_2\_Count) and the values that can be set for the above parameter are:
  - from -32768 to +32767 with steps of 1, **0 (default value)**
- If the format of the object to send is **3 byte RGB colour**, the “**On quadruple touch detection sends the value**” parameter is a dummy one, used to select the colour to be sent. The real value, downloaded into the memory, will depend on the three parameters that represent the colour components (see below). In addition, the **Ch.x - Quadruple touch 3 byte object z** communication object will be visible (Data Point Type: 232.600 DPT\_Colour\_RGB) and the values that can be set for the above parameter are:
  - **white** (default value)
  - yellow
  - magenta
  - red
  - turquoise
  - green
  - blue
  - customize

By selecting **customize**, the following parameters are made visible: “**Value of RED component (0 .. 255)**”, “**Value of GREEN component (0 .. 255)**” and “**Value of BLUE component (0 .. 255)**”; The combination of the three colour components determines the actual value sent on the BUS. If you select any of the other values, these parameters will still be visible but with pre-set values that cannot be modified.

The values that can be set are:

- from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **4 byte unsigned value**, the **Ch.x - Quadruple touch 4 byte object z** communication object will be visible (Data Point Type: 12.001 DPT\_Value\_4\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 4294967295, with steps of 1
- If the format of the object to send is **4 byte signed value**, the **Ch.x - Quadruple touch 4 byte object z** communication object will be visible (Data Point Type: 13.001 DPT\_Value\_4\_Count) and the values that can be set for the above parameter are:
  - from -2147483648 to 2147483647 with steps of 1 (**default value 0**)

**NOTE:** to remedy the problem of coding and the method for inserting values with 2/4 byte floating point format (DPT 9.0xx and 14.0xx), there is an external transformation tool that makes it possible to enter a value in the floating format and obtain the corresponding value with “unsigned and signed value” coding, and vice versa. In this way, the user obtains the value to be entered in the ETS database, selecting the format “2/4 byte signed/unsigned value”.

### ➤ 14.1.12 Channel x – Long touch menu

This menu is visible if the value of the “**Long touch detection**” parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the “long touch” event.

The structure of the menu is as follows:

Fig. 14.6

Upon detection of the long touch, it is possible to send up to 4 different objects (which are distinguished by the letters A, B, C and D). Object A is always enabled, whereas the “**Object z**” parameter (where z is the index of the object associated with the channel, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

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

Selecting **enable** displays the “**Sending object format**” and “**On long touch detection sends the value**” parameters found in the sub-group **Object z** (where z is the index of the object associated with the input, included between **A** and **D**).

The parameter “**Sending object format**” makes it possible to set the format and code of the object “z” of input “x” that is sent by the device. The values that can be set are:

- **1 bit (default value)**
- 2 bit
- 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
- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the “**On long touch detection sends the value**” parameter will be different.

The “**On long touch detection sends the value**” parameter is used to set the command or value to send following the detection of a long touch (on the basis of the set sending conditions) associated with the channel. The values that can be set are:

- If the format of the object to send is **1 bit**, the **Ch.x - Long touch 1 bit object z** communication object will be visible (Data Point Type: 1.002 DPT\_Bool) and the values that can be set for the above parameter are:
  - 0
  - 1
  - **cyclical switching 3 (default value)**

Selecting the value **cyclical switching**, the command that the device sends (via the **Ch.x - Long touch 1 bit object z** object) upon detection of the event associated with cyclical switching will be the opposite of the value generated by the most recent event - the value received via BUS on the **Ch.x - Long touch 1 bit object z** object or the last value sent.

In this configuration, as there is no specific communication object for receiving the status feedback from the BUS, the **Ch.x - Long touch 1 bit object z** object is an input/output object. With KNX technology, a communication object is sent to a single destination group address, so if this object is associated with more than one group address, the device will send the BUS telegram to the group address where the object has the “S” (sending) flag. Vice versa, the device will update its value when a BUS telegram is received on any group address associated with the object, regardless of the “S” flag.

In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the status of the devices connected.

- If the format of the object to send is **2 bit**, the **Ch.x - Long touch 2 bit object z** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control) and the values that can be set for the above parameter are:
  - activate forcing ON (down)
  - activate forcing OFF (up)
  - deactivate forcing
  - cyclical switching - ON forcing /OFF forcing
  - **cyclical switching - forcing ON / deactivate forcing (default value)**
  - cyclical switching - OFF forcing/forcing deactivation

Selecting the value **cyclical switching**, the command that the device sends (via the object **Ch.x - Long touch 2 bit object z**) upon detection of the event associated with cyclical switching will be the opposite of the last value sent (via the **Ch.x - Long touch 2 bit object z** object).

- If the format of the object to send is **1 byte unsigned value**, the **Ch.x - Long touch 1 byte object z** communication object will be visible (Data Point Type: 5.010 DPT\_Value\_1\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **1 byte signed value**, the **Ch.x - Long touch 1 byte object z** communication object will be visible (Data Point Type: 6.010 DPT\_Value\_1\_Count) and the values that can be set for the above parameter are:
  - from -128 to 127 with steps of 1, **0 (default value)**
- If the format of the object to send is **1 byte percentage value**, the **Ch.x - Long touch 1 byte object z** communication object will be visible (Data Point Type: 5.001 DPT\_Scaling) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 100, with steps of 1

- If the format of the object to send is **1 byte HVAC mode**, the **Ch.x - Long touch 1 byte object z** communication object will be visible (Data Point Type: 20.102 DPT\_HVACMode) and the values that can be set for the above parameter are:
  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

By selecting **cyclical switching (thermostat)**, each time the associated event (long touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Comfort* ...; by selecting **cyclical switching (timed thermostat)**, each time the associated event (long touch) is detected, the device will send a new thermoregulation mode (HVAC), in the order *Comfort*→ *Pre-comfort*→ *Economy*→ *Off*→ *Auto*→ *Comfort* ...

- If the format of the object to send is **2 byte unsigned value**, the **Ch.x - Long touch 2 byte object z** communication object will be visible (Data Point Type: 7.001 DPT\_Value\_2\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 65535, with steps of 1
- If the format of the object to send is **2 byte signed value**, the **Ch.x - Long touch 2 byte object z** communication object will be visible (Data Point Type: 8.001 DPT\_Value\_2\_Count) and the values that can be set for the above parameter are:
  - from -32768 to +32767 with steps of 1, **0 (default value)**
- If the format of the object to send is **3 byte RGB colour**, the “**On long touch detection sends the value**” parameter is a dummy one, used to select the colour to be sent. The real value, downloaded into the memory, will depend on the three parameters that represent the colour components (see below). In addition, the **Ch.x - Long touch 3 byte object z** communication object will be visible (Data Point Type: 232.600 DPT\_Colour\_RGB) and the values that can be set for the above parameter are:
  - **white (default value)**
  - yellow
  - magenta
  - red
  - turquoise
  - green
  - blue
  - customize

By selecting **customize**, the following parameters are made visible: “**Value of RED component (0 .. 255)**”, “**Value of GREEN component (0 .. 255)**” and “**Value of BLUE component (0 .. 255)**”; The combination of the three colour components determines the actual value sent on the BUS. If you select any of the other values, these parameters will still be visible but with pre-set values that cannot be modified.

The values that can be set are:

- from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **4 byte unsigned value**, the **Ch.x - Long touch 4 byte object z** communication object will be visible (Data Point Type: 12.001 DPT\_Value\_4\_Ucount) and the values that can be set for the above parameter are:
  - from **0 (default value)** to 4294967295, with steps of 1

- If the format of the object to send is **4 byte signed value**, the **Ch.x - Long touch 4 byte object z** communication object will be visible (Data Point Type: 13.001 DPT\_Value\_4\_Count) and the values that can be set for the above parameter are:
  - from -2147483648 to 2147483647 with steps of 1 (**default value 0**)

**NOTE:** to remedy the problem of coding and the method for inserting values with 2/4 byte floating point format (DPT 9.0xx and 14.0xx), there is an external transformation tool that makes it possible to enter a value in the floating format and obtain the corresponding value with “unsigned and signed value” coding, and vice versa. In this way, the user obtains the value to be entered in the ETS database, selecting the format “2/4 byte signed/unsigned value”.

## 15 “Sound signalling” menu

The device is equipped with an electronic buzzer that can be configured to emit sound signals following BUS commands or touch events. With regards touch events, the configuration of this function is valid for all six channels; it is not possible to differentiate the sound effects associated with each individual channel.

Figure 15.1 shows the basic structure of the menu:

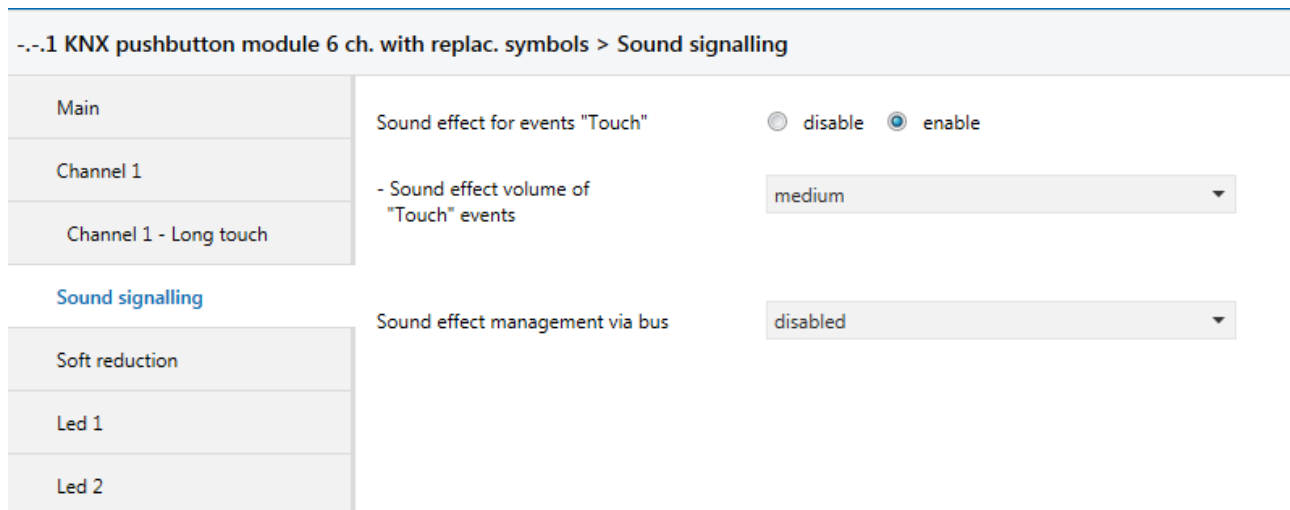


Fig. 15.1

### 15.1 Parameters

#### ➤ 15.1.1 Sound effect for events “Touch”

The “**Sound effect for events “Touch”**” parameter is used to enable the reproduction of certain sound effects indicating the detection of certain touch events by the capacitive sensor. The values that can be set are:

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

Selecting the value **enable** displays the “**Sound effect volume of “Touch” events**” parameter. Each touch event is associated with a particular sound effect:

- Touch → **click**
- Maintained touch → **click 300ms**
- Long touch → **short beep**

#### ➤ 15.1.2 Sound effect volume of “Touch” events

The “**Sound effect volume of “Touch” events**” parameter defines the power level of the sounds associated with the Touch events. The values that can be set are:

- low
- **medium (default value)**
- high

#### ➤ 15.1.3 Sound effect management via bus

The “**Sound effect management via bus**” parameter is used to enable a communication object for the activation/deactivation of a certain sound effect. The values that can be set are:

- **disabled (default value)**

- click 1
- beep 2
- long beep 3
- double click 4
- double beep 5
- triple click 6
- triple beep 7
- cyclical click (300 ms) 8
- cyclical beep (1 s) 9

Selecting any value other than **disabled** displays the “**Volume of signalling associated with the effect via bus**” parameter and the **Sound effect** (Data Point Type: 1.001 DPT\_Switch) communication object, for activating/deactivating the reproduction of the selected effect.

When the BUS voltage is recovered, the device only reactivates the sound effect if it is a cyclical effect and was already active prior to the BUS failure, and it sends the status reading on the **Sound effect** object in order to realign itself with the field.

#### ➤ 15.1.4 Volume of signalling associated with the effect via bus

The “**Volume of signalling associated with the effect via bus**” parameter defines the power level of the sounds associated with the sound effect via the BUS. The values that can be set are:

- low
- medium
- **high** (default value)

The sound effect associated with Touch events has a higher priority than the active effect (activated by the BUS command) in the moment when the sensor detects the operation; at the end of the reproduction of the effect associated with the current event, a certain time period (not configurable by the user) must pass before the effect activated previously by the BUS command is again reproduced, and continues until it is deactivated by the relative BUS command. The effect activated by the BUS event is therefore “suspended” when a “Touch” event occurs.

The table below shows which sound effects are reproduced, according to the function associated with the channel.

		Touch events		
		Touch	Operation	Long touch (operation)
Matched function	No function			
	Edge/command sequence - touch/release	YES		
	Edge/command sequence - short/long operation	YES		YES if at least one action is associated with the long touch (operation)
	1 push-button + stop dimmer	YES	YES	
	Cyclic sending 1 push-button dimmer	YES	YES	



	1 push-button shutter control	YES		YES
	2 push-button + stop dimmer	YES	YES	
	2 push-button dimmer with cyclic sending	YES	YES	
	2 push-button shutter control	YES		YES
	Scene management	YES		YES if scene storing is enabled; cannot be deactivated by user
	Switching sequences	YES		YES
	Multiple touches	YES		YES if at least one action is associated with the long touch (operation)
	Associated effect	click	click 300 ms	short beep

The signalling associated with Touch events can be enabled/disabled via the BUS using the **Enable sound effects on Touch** (Data Point Type: 1.003 DPT\_Enable) communication object, which is only visible if the **“Sound effect for “Touch” events”** parameter has the value **enable**. When the signalling is deactivated via this communication object, any sound effects associated with Touch events are not reproduced until the signalling is reactivated. Any sound effect activated via the BUS using the **Sound effect** object will be reproduced regardless of the activation of the signalling associated with Touch events. In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the function activation status.

## 16 “Soft reduction” menu

Given its function, the device could be installed in places where it is necessary to temporarily inhibit the sound signalling and minimise the intensity of the backlighting, so as not to cause any disturbance. The “Soft reduction” function uses a specific communication object to temporarily modify the sound/light signalling settings to minimise the disturbance for the user during the night-time. This function is activated/deactivated from a remote device connected to the KNX system (e.g. an hourly timer, a supervisor/control panel, a light sensitive sensor, etc.).

Figure 16.1 shows the basic structure of the menu:

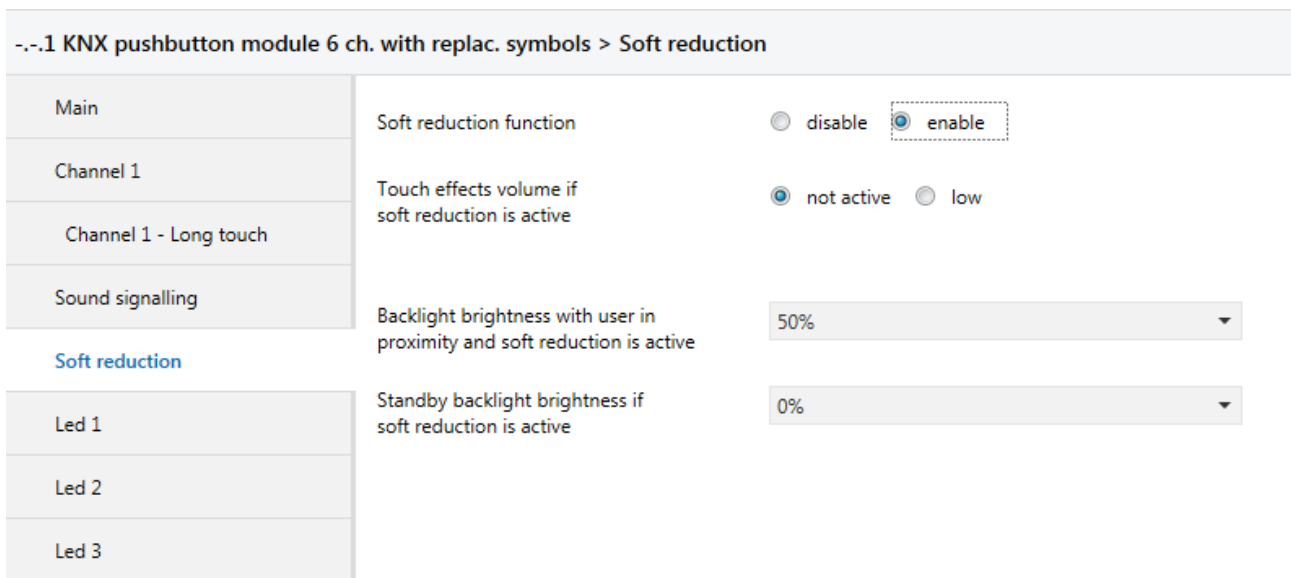


Fig. 16.1

### 16.1 Parameters

#### ➤ 16.1.1 Soft reduction function

The “**Soft reduction function**” parameter is used to enable the function and display the configuration parameters. The values that can be set are:

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

Selecting **enable** displays the function configuration parameters and the **Soft reduction** (Data Point Type: 1.001 DPT\_Switch) communication object.

In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the function activation status. When the BUS voltage is recovered, the function status is the one that was active prior to the failure. If necessary, it will then be updated according to the status read request.

#### ➤ 16.1.2 Touch effect volume if soft reduction is active

The “**Touch effect volume if soft reduction is active**” parameter, visible if the “**Sound effect for “Touch” events**” parameter of the **Sound signalling** menu assumes the value **enable**, defines the power level of the sounds associated with the touch effect (Touch, Maintained, Long touch, etc.) when the Soft reduction function is active. The values that can be set are:

- **not active** (default value)
- low

This setting does not concern the possible reproduction of a sound effect activated via the BUS.  
When the function is deactivated, the volume depends on the setting of the “**Sound signalling volume**” parameter of the **Sound signalling** menu.

➤ **16.1.3 Backlight brightness with user in proximity and soft reduction active**

The “**Backlight brightness with user in proximity and soft reduction active**” parameter defines the level of brightness of the backlighting when the proximity sensor detects the presence of the user in front of the device and the soft reduction function is active. The values that can be set are:

- 100%
- 90%
- 80%
- 70%
- 60%
- **50%** (default value)

➤ **16.1.4 Standby backlight brightness if soft reduction is active**

The “**Standby backlight brightness if soft reduction is active**” parameter defines the level of brightness of the backlighting when the device is in standby and the soft reduction function is active. The values that can be set are:

- 40%
- 30%
- 20%
- 10%
- **0%** (default value)
- no reduction

When the function is disabled, the backlight brightness levels will depend on the setting of the relative parameters in the **Main** menu.

## 17 “LED X” menu

This is used to define and customize the operation of the signalling LEDs associated with the channel. The signalling LED can assume different colours for the night lighting function, or it can be autonomously managed via the relative communication objects. The "light signalling upon touch/operation/long touch/release" function may be enabled or disabled with various preconfigured lighting effects. The effects associated with the events detected by the capacitive sensor are performed, even if prior to the touch detection phase (standby) a light effect enabled by a BUS command is active (and is then reactivated at the end of the effect associated with the release). 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**.

Figure 17.1 shows the basic structure of the menu:

Navigation	Parameter	Value
Main	Night lighting	amber
Channel 1		
Channel 1 - Long touch	Light effect for "Touch" events	<input type="radio"/> disable <input checked="" type="radio"/> enable
Sound signalling		
Soft reduction	Light effect 1	<input type="radio"/> disable <input checked="" type="radio"/> enable
<b>Led 1</b>	Effect activation value	<input type="radio"/> "0" value <input checked="" type="radio"/> "1" value
Led 2	Upon receiving the activation value the light signalling is	permanently on
Led 3		
Led 4	- Light color	blue
Led 5		
Led 6	Light effect 2	<input checked="" type="radio"/> disable <input type="radio"/> enable

Fig. 17.1

### 17.1 Parameters

#### ➤ 17.1.1 Night lighting

The “**Night lighting**” parameter is used to enable and define the colour of the night lighting associated with channel x. The values that can be set are:

- disabled
- white
- yellow
- magenta
- red
- turquoise
- green
- blue
- **amber (default value)**
- customize

Selecting **customize** displays the following parameters: “**Value of RED component (0 .. 255)**”, “**Value of GREEN component (0 .. 255)**” and “**Value of BLUE component (0 .. 255)**” and the **Led x - Customize night lighting** (Data Point Type: 232.600 DPT\_Colour\_RGB) communication object. The combination of the three colour components determines the colour associated with night lighting. The values that can be set are:

- from **0 (default value)** to 255, with steps of 1

The **Led x - Customize night lighting** communication object is used to receive, via the BUS, the colour you want to associate with night lighting. Every time a new value is received via this object, the device associates the new colour with night-time signalling. This doesn't mean, however, that every time a new colour is received the light signalling of LED x assumes that colour; it is only visualised immediately if the night lighting of LED x is active.

The same object (**Led x - Customize night lighting**) is also used to send the signalling of the current colour associated with night lighting. This feedback is sent upon demand and spontaneously when the BUS voltage is recovered and there is a colour variation.

In this configuration, the **Led x - Customize night lighting** object is an input/output object. With KNX technology, a communication object is sent to a single destination group address, so if this object is associated with more than one group address, the device will send the BUS telegram (signalling of the current colour) to the group address where the object has the “S” (sending) flag. Vice versa, the device will update its value when a BUS telegram (customized colour setting) is received on any group address associated with the object, regardless of the “S” flag.

### ➤ 17.1.2 Light effects for “Touch” events

The “**Light effects for “Touch” events**” parameter is used to enable the reproduction of certain light effects indicating the detection of certain touch events by the capacitive sensor. The values that can be set are:

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

Selecting the value **enable**, the reproduction of a certain light effect is associated with each Touch event:

- Touch → **blink**
- Maintained touch → **quick blinking**
- Long touch (operation) → **3 blinks**

The effects associated with Touch events do not alter the current colour of the light signalling; if the light signalling is disabled, the effect associated with the Touch event is not reproduced.

The light effect associated with Touch events has a higher priority than the active effect (e.g. blinking activated by the BUS command) in the moment when the sensor detects the operation; at the end of the reproduction of the effect associated with the current event, a certain time period (not configurable by the user) must pass before the effect activated previously by the BUS command is again reproduced. The effect activated by the BUS event is therefore “suspended” when a “Touch” event occurs.

The table below shows which light effects are reproduced, according to the function associated with the channel.

		Touch events		
		Touch	Operation	Long touch (operation)
Matched function	No function			
	Edge/command sequence - touch/release	YES		
	Edge/command sequence - short/long operation	YES		YES if at least one action is associated with the long touch (operation)
	1 push-button + stop dimmer	YES	YES	
	Cyclic sending 1 push-button dimmer	YES	YES	
	1 push-button shutter control	YES		YES
	2 push-button + stop dimmer	YES	YES	
	2 push-button dimmer with cyclic sending	YES	YES	
	2 push-button shutter control	YES		YES
	Scene management	YES		YES if scene storing is enabled; cannot be deactivated by user
	Switching sequences	YES		YES
	Multiple touches	YES		YES if at least one action is associated with the long touch (operation)
Associated effect	blink	quick blinking	3 blinks	

After downloading the application, the light signalling of the channel will be the one defined for night lighting. When the BUS voltage is recovered, the effect that was active prior to the failure is reactivated. If necessary, it will then be updated according to the status read requests on the objects **Led x - Effect 1**, **Led x - Effect 2**, **Led x - Effect 3**, **Led x - Effect 4** and **Led x - Effect 5**.

### ➤ 17.1.3 Light effect X

The brightness of the LED associated with the channel depends on the setting of the parameters of the **Main** menu.

The “**Light effect 1**”, “**Light effect 2**”, “**Light effect 3**”, “**Light effect 4**” and “**Light effect 5**” parameters are used to enable various communication objects for the activation of the light signalling via a BUS telegram; when a light effect is enabled, the parameter for enabling the next one will appear. The values that can be set are:

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

For each effect enabled, the “**Effect activation value**” and “**Upon receiving the activation value the light signalling is**” parameters are made visible, along with the associated BUS communication object (**Led x - Effect 1, Led x - Effect 2, Led x - Effect 3, Led x - Effect 4 or Led x - Effect 5**) (Data Point Type: 1.002 DPT\_Bool).

In this case, every time the BUS voltage is recovered, you must send a status read request on the above objects in order to update the push-button panel about the status of the devices connected.

The “**Effect activation value**” parameter defines which logic value received via the associated object (**Led x - Effect 1, Led x - Effect 2, Led x - Effect 3, Led x - Effect 4 or Led x - Effect 5**) will activate that specific light effect. The values that can be set for this parameter are:

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

The opposite value to the one set for activation will deactivate the associated effect.

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. For the 5 light effects associated with each LED, there is a priority order: effect 5 takes priority over all the others.

Function/object	Priority	
Light effect 1	1	low
Light effect 2	2	
Light effect 3	3	
Light effect 4	4	
Light effect 5	5	high

Only one effect can be reproduced at a time. Once that effect has been deactivated, the activation status of those with a lower priority ranking is evaluated, and night lighting is activated if there is no other effect active. In particular, the activation of an effect with a priority ranking higher than the active one causes the reproduction of the new effect without actually deactivating the old one (the activation of a higher priority effect does not deactivate the lower priority one); the old effect will be reproduced when the higher priority one has been deactivated. The activation of an effect with a lower priority than the active one is stored and will be implemented when all the higher priority effects have been deactivated.

The “**Upon receiving the activation value the light signalling is**” parameter defines the behaviour of the light signalling upon receiving the activation command for the light effect to which the parameter refers. The values that can be set are:

- off
- **permanently on** (default value)
- blinking 1 Hz
- blinking 2 Hz

Selecting any value other than **off** displays the “**Light colour**” parameter for customizing the colour of the signal associated with that effect. The values that can be set are:

- current colour
- white
- yellow
- magenta
- red
- turquoise
- green
- **blue** (default value)
- amber
- customize (only visible for effect 1)
- customize (only visible for effect 2)
- customize (only visible for effect 3)

- customize (only visible for effect 4)
- customize (only visible for effect 5)

Selecting **current colour**, if the LED is disabled when the effect is activated, the set light effect will not be reproduced.

Selecting **customize** displays the following parameters: “**Value of RED component (0 .. 255)**”, “**Value of GREEN component (0 .. 255)**” and “**Value of BLUE component (0 .. 255)**”, along with the **Led x - Customize effect 1** (**Led x - Customize effect 2**, **Led x - Customize effect 3**, **Led x - Customize effect 4** or **Led x - Customize effect 5** communication object (depending on the effect to which the parameter refers) (Data Point Type: 232.600 DPT\_Colour\_RGB). The combination of the three colour components determines the colour associated with the light effect via the BUS. The values that can be set are:

- from **0 (default value)** to 255, with steps of 1

The **Led x - Customize effect 1 (2/3/4/5)** communication object is used to receive (via the BUS) the colour you want to associate with the light effect that the object refers to. Every time a new value is received via this object, the device associates the new colour with the associated light effect. This doesn't mean, however, that every time a new colour is received the light signalling of LED x assumes that colour; it is only visualised immediately if the light effect of LED x is active.

The same object (**Led x - Customize effect 1 (2/3/4/5)**) is also used to send the signalling of the current colour associated with the light effect that the object refers to. This feedback is sent upon demand and spontaneously when the BUS voltage is recovered and there is a colour variation.

In this configuration, the **Led x - Customize effect 1 (2/3/4/5)** object is an input/output object. With KNX technology, a communication object is sent to a single destination group address, so if this object is associated with more than one group address, the device will send the BUS telegram (signalling of the current colour) to the group address where the object has the “S” (sending) flag. Vice versa, the device will update its value when a BUS telegram (customized colour setting) is received on any group address associated with the object, regardless of the “S” flag.

NOTE: the customized colour resulting from the combination of the three fundamental RGB components is interpreted by the device as a colour with 100% brightness. This means that if the backlighting brightness value set in the main menu is different from 100%, the final effect may not be as expected.

To help the installer make the best selection of the RGB components of the colour to be associated with the light signalling, the **RGB colour TEST** (Data Point Type: 232.600 DPT\_Colour\_RGB) and the **RGB colour brightness TEST** (Data Point Type: 5.001 DPT\_Scaling) communication objects can be used to select respectively the colour and the brightness percentage of the signalling, so that the selected colour can be verified with the brightness levels set for the backlighting both in proximity and in standby. The colour test mode is activated when a telegram is received on the **RGB colour TEST** object. If a telegram is received on the **RGB colour brightness TEST** object and the test function is not active, there is no action.

When the TEST function is active, all 6 LEDs light up with the colour received via the BUS. The TEST function is automatically deactivated after 30 seconds of no telegrams received on the test objects. Any light/sound effects received when the TEST function is active are managed after the TEST function has been deactivated.

Every time the colour is modified via the **RGB colour TEST** object, the brightness of the light signalling returns to the 100% value (even if it has been previously modified).

Once the required values have been established, they should be indicated in the ETS parameters or sent to the device via the BUS, using the communication objects given over to light signalling customization, so the customized colour can be used during normal device operation.



## 18 Communication objects

The following tables summarise all the communication objects with their specific ID numbers, names and functions displayed in ETS, plus a brief description of the function and the type of Datapoint used.

### Communication objects with output functions

#						Object name	Object function	Description	Datapoint type
Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6				
1	26	51	76	101	126	Ch.x - Switch	On/Off	Dimmer switching on/off commands	[1.1] DPT_Switch
1	26	51	76	101	126	Ch.x - Shutter movement	Up/Down	Moves up/down the shutter	[1.8] DPT_UpDown
1	26	51	76	101	126	Ch.x - Scene	Execute/Store	Send learn/execute scene commands	[18.1] DPT_SceneControl
1	26	51	76	101	126	Ch.x - Sequence A	On/Off	Send On/Off commands associated with object A of the sequence	[1.1] DPT_Switch
1	26	51	76	101	126	Ch.x - A object 1 bit value	1/0 value	Send 1/0 values associated with object A	[1.2] DPT_Bool
1	26	51	76	101	126	Ch.x - A object 2 bits value	On/Off forced positioning	Send priority commands associated with object A of the sequence	[2.1] DPT_Switch_Control
1	26	51	76	101	126	Ch.x - A object 1 byte value	Unsigned value	Send unsigned values (0..255) associated with object A	[5.10] DPT_Value_1_Ucount
1	26	51	76	101	126	Ch.x - A object 1 byte value	Signed value	Send signed values (-128..127) associated with object A	[6.10] DPT_Value_1_Count
1	26	51	76	101	126	Ch.x - A object 1 byte value	% Value	Send percent values (0%..100%) associated with object A	[5.1] DPT_Scaling
1	26	51	76	101	126	Ch.x - A object 1 byte value	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated with object A	[20.102] DPT_HVACMode
1	26	51	76	101	126	Ch.x - A object 2 bytes value	Unsigned value	Send unsigned values (0..65535) associated with object A	[7.1] DPT_Value_2_Ucount
1	26	51	76	101	126	Ch.x - A object 2 bytes value	Signed value	Send signed values (-32768..32767) associated with object A	[8.1] DPT_Value_2_Count
1	26	51	76	101	126	Ch.x - A object 3 byte value	RGB colour	Send RGB colour components value associated with object A	[232.600] DPT_Colour_RGB
1	26	51	76	101	126	Ch.x - A object 4 bytes value	Unsigned value	Send unsigned values (0..4294967295) associated with object A	[12.1] DPT_Value_4_Ucount
1	26	51	76	101	126	Ch.x - A object 4 bytes value	Signed value	Send signed values (-2147483648..2147483647) associated with object A	[13.1] DPT_Value_4_Count
1	26	51	76	101	126	Ch.x - A object 14 bytes value	Characters ISO 8859-1	Send characters codified with ISO 8859-1 standard associated with object A	[16.1] DPT_String_8859_1
1	26	51	76	101	126	Ch.x - Single touch 1 bit object A	1/0 value	Send 1/0 values associated with object A of the single touch	[1.2] DPT_Bool
1	26	51	76	101	126	Ch.x - Single touch 2 bit object A	On/Off forced positioning	Send priority commands associated with object A of the single touch	[2.1] DPT_Switch_Control
1	26	51	76	101	126	Ch.x - Single touch 1 byte object A	Unsigned value	Send unsigned values (0.. 255) associated with object A of the single touch	[5.10] DPT_Value_1_Ucount
1	26	51	76	101	126	Ch.x - Single touch 1 byte object A	Signed value	Send signed values (-128..127) associated with object A of the single touch	[6.10] DPT_Value_1_Count

							touch		
1	26	51	76	101	126	Ch.x - Single touch 1 byte object A	% Value	Send percent values (0%..100%) associated with object A of the single touch	[5.1] DPT_Scaling
1	26	51	76	101	126	Ch.x - Single touch 1 byte object A	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated with object A of single touch	[20.102] DPT_HVACMode
1	26	51	76	101	126	Ch.x - Single touch 2 byte object A	Unsigned value	Send unsigned values (0.. 65535) associated with object A of the single touch	[7.1] DPT_Value_2_Ucount
1	26	51	76	101	126	Ch.x - Single touch 2 byte object A	Signed value	Send signed values (-32768..32767) associated with object A of the single touch	[8.1] DPT_Value_2_Count
1	26	51	76	101	126	Ch.x - Single touch 3 byte object A	RGB colour	Send RGB colour components value associated with object A of the single touch	[232.600] DPT_Colour_RGB
1	26	51	76	101	126	Ch.x - Single touch 4 byte object A	Signed value	Send signed values (-2147483648..2147483647) associated with object A of the single touch	[13.1] DPT_Value_4_Count
2	27	52	77	102	127	Ch.x – Brightness dimming	Increase/Decrease	Dimmer brightness increasing/decreasing commands	[3.7] DPT_Control_Dimming
2	27	52	77	102	127	Ch.x – Shutter stop/Louvres control	Stop/Step	Send stop movement/slat regulation commands	[1.7] DPT_Step
2	27	52	77	102	127	Ch.x – Sequence B	On/Off	Send On/Off commands associated with object B of the sequence	[1.1] DPT_Switch
2	27	52	77	102	127	Ch.x – B object 1 bit value	1/0 value	Send 1/0 values associated with object B	[1.2] DPT_Bool
2	27	52	77	102	127	Ch.x - Single touch 1 bit object B	1/0 value	Send 1/0 values associated with object B of the single touch	[1.2] DPT_Bool
3	28	53	78	103	128	Ch.x – Sequence C	On/Off	Send On/Off commands associated with object C of the sequence	[1.1] DPT_Switch
3	28	53	78	103	128	Ch.x – Object C 1-bit value	1/0 value	Send 1/0 values associated with object C	[1.1] DPT_Switch
3	28	53	78	103	128	Ch.x - Single touch 1 bit object C	1/0 value	Send 1/0 values associated with object C of the single touch	[1.2] DPT_Bool
4	29	54	79	104	129	Ch.x – Sequence D	On/Off	Send On/Off commands associated with object D of the sequence	[1.1] DPT_Switch
4	29	54	79	104	129	Ch.x – D object 1 bit value	1/0 value	Send 1/0 values associated with object D	[1.1] DPT_Switch
4	29	54	79	104	129	Ch.x - Single touch 1 bit object D	1/0 value	Send 1/0 values associated with object D of the single touch	[1.2] DPT_Bool
5	30	55	80	105	130	Ch.x - Double touch 1 bit object A	1/0 value	Send 1/0 values associated with object A of the double touch	[1.2] DPT_Bool
5	30	55	80	105	130	Ch.x - Double touch 2 bit object A	On/Off forced positioning	Send priority commands associated with object A of the double touch	[2.1] DPT_Switch_Control
5	30	55	80	105	130	Ch.x - Double touch 1 byte object A	Unsigned value	Send unsigned values (0.. 255) associated with object A of the double touch	[5.10] DPT_Value_1_Ucount
5	30	55	80	105	130	Ch.x - Double touch 1 byte object A	Signed value	Send signed values (-128..127) associated with object A of the double touch	[6.10] DPT_Value_1_Count
5	30	55	80	105	130	Ch.x - Double touch 1 byte	% Value	Send percent values	[5.1] DPT_Scaling

						object B		(0%..100%) associated with object B of the double touch	
5	30	55	80	105	130	Ch.x - Double touch 1 byte object A	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated with object A of double touch	[20.102] DPT_HVACMode
5	30	55	80	105	130	Ch.x - Double touch 2 byte object A	Unsigned value	Send unsigned values (0.. 65535) associated with object A of the double touch	[7.1] DPT_Value_2_Ucount
5	30	55	80	105	130	Ch.x - Double touch 2 byte object A	Signed value	Send signed values (-32768..32767) associated with object A of the double touch	[8.1] DPT_Value_2_Count
5	30	55	80	105	130	Ch.x - Double touch 3 byte object A	RGB colour	Send RGB colour components value associated with object A of the double touch	[232.600] DPT_Colour_RGB
5	30	55	80	105	130	Ch.x - Double touch 4 byte object A	Unsigned value	Send unsigned values (0.. 4294967295) associated with object A of the double touch	[12.1] DPT_Value_4_Ucount
5	30	55	80	105	130	Ch.x - Double touch 4 byte object A	Signed value	Send signed values (-2147483648..2147483647) associated with object A of the double touch	[13.1] DPT_Value_4_Count
6	31	56	81	106	131	Ch.x - Double touch 1 bit object B	1/0 value	Send 1/0 values associated with object B of the double touch	[1.2] DPT_Bool
7	32	57	82	107	132	Ch.x - Double touch 1 bit object C	1/0 value	Send 1/0 values associated with object C of the double touch	[1.2] DPT_Bool
8	33	58	83	108	133	Ch.x - Double touch 1 bit object D	1/0 value	Send 1/0 values associated with object D of the double touch	[1.2] DPT_Bool
9	34	59	84	109	134	Ch.x - Triple touch 1 bit object A	1/0 value	Send 1/0 values associated with object A of the triple touch	[1.2] DPT_Bool
9	34	59	84	109	134	Ch.x - Triple touch 2 bit object A	On/Off forced positioning	Send priority commands associated with object A of the double touch	[2.1] DPT_Switch_Control
9	34	59	84	109	134	Ch.x - Triple touch 1 byte object A	Unsigned value	Send unsigned values (0.. 255) associated with object A of the triple touch	[5.10] DPT_Value_1_Ucount
9	34	59	84	109	134	Ch.x - Triple touch 1 byte object A	Signed value	Send signed values (-128..127) associated with object A of the triple touch	[6.10] DPT_Value_1_Count
9	34	59	84	109	134	Ch.x - Triple touch 1 byte object A	% Value	Send percent values (0%..100%) associated with object A of the triple touch	[5.1] DPT_Scaling
9	34	59	84	109	134	Ch.x - Triple touch 1 byte object A	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated with object A of triple touch	[20.102] DPT_HVACMode
9	34	59	84	109	134	Ch.x - Triple touch 2 byte object A	Unsigned value	Send unsigned values (0.. 65535) associated with object A of the triple touch	[7.1] DPT_Value_2_Ucount
9	34	59	84	109	134	Ch.x - Triple touch 2 byte object A	Signed value	Send signed values (-32768..32767) associated with object A of the triple touch	[8.1] DPT_Value_2_Count
9	34	59	84	109	134	Ch.x - Triple touch 3 byte object A	RGB colour	Send RGB colour components value associated with object A	[232.600] DPT_Colour_RGB

											of the triple touch	
9	34	59	84	109	134	Ch.x - Triple touch 4 byte object A	Unsigned value	Send unsigned values (0.. 4294967295) associated with object A of the triple touch	[12.1] DPT_Value_4_Ucount			
9	34	59	84	109	134	Ch.x - Triple touch 4 byte object A	Signed value	Send signed values (-2147483648.. 2147483647) associated with object A of the triple touch	[13.1] DPT_Value_4_Count			
10	35	60	85	110	135	Ch.x - Triple touch 1 bit object B	1/0 value	Send 1/0 values associated with object B of the triple touch	[1.2] DPT_Bool			
11	36	61	86	111	136	Ch.x - Triple touch 1 bit object C	1/0 value	Send 1/0 values associated with object C of the triple touch	[1.2] DPT_Bool			
12	37	62	87	112	137	Ch.x - Triple touch 1 bit object D	1/0 value	Send 1/0 values associated with object D of the triple touch	[1.2] DPT_Bool			
13	38	63	88	113	138	Ch.x - Quadruple touch 1 bit object A	1/0 value	Send 1/0 values associated with object A of the quadruple touch	[1.2] DPT_Bool			
13	38	63	88	113	138	Ch.x - Quadruple touch 2 bit object A	On/Off forced positioning	Send priority commands associated with object A of the quadruple touch	[2.1] DPT_Switch_Control			
13	38	63	88	113	138	Ch.x - Quadruple touch 1 byte object A	Unsigned value	Send unsigned values (0.. 255) associated with object A of the quadruple touch	[5.10] DPT_Value_1_Ucount			
13	38	63	88	113	138	Ch.x - Quadruple touch 1 byte object A	Signed value	Send signed values (-128..127) associated with object A of the quadruple touch	[6.10] DPT_Value_1_Count			
13	38	63	88	113	138	Ch.x - Quadruple touch 1 byte object A	% Value	Send percent values (0%..100%) associated with object A of the quadruple touch	[5.1] DPT_Scaling			
13	38	63	88	113	138	Ch.x - Quadruple touch 1 byte object A	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated with object A of quadruple touch	[20.102] DPT_HVACMode			
13	38	63	88	113	138	Ch.x - Quadruple touch 2 byte object A	Unsigned value	Send unsigned values (0.. 65535) associated with object A of the quadruple touch	[7.1] DPT_Value_2_Ucount			
13	38	63	88	113	138	Ch.x - Quadruple touch 2 byte object A	Signed value	Send signed values (-32768..32767) associated with object A of the quadruple touch	[8.1] DPT_Value_2_Count			
13	38	63	88	113	138	Ch.x - Quadruple touch 3 byte object A	RGB colour	Send RGB colour components value associated with object A of the quadruple touch	[232.600] DPT_Colour_RGB			
13	38	63	88	113	138	Ch.x - Quadruple touch 4 byte object A	Unsigned value	Send unsigned values (0.. 4294967295) associated with object A of the quadruple touch	[12.1] DPT_Value_4_Ucount			
13	38	63	88	113	138	Ch.x - Quadruple touch 4 byte object A	Signed value	Send signed values (-2147483648.. 2147483647) associated with object A of the quadruple touch	[13.1] DPT_Value_4_Count			
14	39	64	89	114	139	Ch.x - Quadruple touch 1 bit object B	1/0 value	Send 1/0 values associated with object B of the quadruple touch	[1.2] DPT_Bool			
15	40	65	90	115	140	Ch.x - Quadruple touch 1 bit object C	1/0 value	Send 1/0 values associated with object C of the quadruple touch	[1.2] DPT_Bool			
16	41	66	91	116	141	Ch.x - Quadruple touch 1 bit object D	1/0 value	Send 1/0 values associated with object D of the quadruple touch	[1.2] DPT_Bool			

17	42	67	92	117	142	Ch.x - Long touch 1 bit object A	1/0 value	Send 1/0 values associated with object A of the long touch	[1.2] DPT_Bool
17	42	67	92	117	142	Ch.x - Long touch 2 bit object A	On/Off forced positioning	Send priority commands associated with object A of the long touch	[2.1] DPT_Switch_Control
17	42	67	92	117	142	Ch.x - Long touch 1 byte object A	Unsigned value	Send unsigned values (0.. 255) associated with object A of the long touch	[5.10] DPT_Value_1_Ucount
17	42	67	92	117	142	Ch.x - Long touch 1 byte object A	Signed value	Send signed values (-128..127) associated with object A of the long touch	[6.10] DPT_Value_1_Count
17	42	67	92	117	142	Ch.x - Long touch 1 byte object A	% Value	Send percent values (0%..100%) associated with object B of the long touch	[5.1] DPT_Scaling
17	42	67	92	117	142	Ch.x - Long touch 1 byte object A	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated with object A of long touch	[20.102] DPT_HVACMode
17	42	67	92	117	142	Ch.x - Long touch 2 byte object A	Unsigned value	Send unsigned values (0.. 65535) associated with object A of the long touch	[7.1] DPT_Value_2_Ucount
17	42	67	92	117	142	Ch.x - Long touch 2 byte object A	Signed value	Send signed values (-32768..32767) associated with object A of the long touch	[8.1] DPT_Value_2_Count
17	42	67	92	117	142	Ch.x - Long touch 3 byte object A	RGB colour	Send RGB colour components value associated with object A of the long touch	[232.600] DPT_Colour_RGB
17	42	67	92	117	142	Ch.x - Long touch 4 byte object A	Unsigned value	Send unsigned values (0.. 4294967295) associated with object A of the long touch	[12.1] DPT_Value_4_Ucount
17	42	67	92	117	142	Ch.x - Long touch 4 byte object A	Signed value	Send signed values (-2147483648.. 2147483647) associated with object A of the long touch	[13.1] DPT_Value_4_Count
18	43	68	93	118	143	Ch.x - Long touch 1 bit object B	1/0 value	Send 1/0 values associated with object B of the long touch	[1.2] DPT_Bool
19	44	69	94	119	144	Ch.x - Long touch 1 bit object C	1/0 value	Send 1/0 values associated with object C of the long touch	[1.2] DPT_Bool
20	45	70	95	120	145	Ch.x - Long touch 1 bit object D	1/0 value	Send 1/0 values associated with object D of the long touch	[1.2] DPT_Bool

The variations of the objects highlighted in light blue in the above table are not shown for objects B (objects 2/27/52/77/102/127), C (objects 3/28/53/78/103/128) and D (objects 4/29/54/79/104/129) due to space problems, but they are present.

The variations of the objects highlighted in red in the above table are not shown for objects B (objects 6/31/56/81/106/131), C (objects 7/32/57/82/107/132) and D (objects 8/33/58/83/108/133) due to space problems, but they are present.

The variations of the objects highlighted in orange in the above table are not shown for objects B (objects 10/35/60/85/110/135), C (objects 11/36/61/86/111/136) and D (objects 12/37/62/87/112/137) due to space problems, but they are present.

The variations of the objects highlighted in green in the above table are not shown for objects B (objects 14/39/64/89/114/139), C (objects 15/40/65/90/115/140) and D (objects 16/41/66/91/116/141) due to space problems, but they are present.

The variations of the objects highlighted in grey in the above table are not shown for objects B (objects 18/43/68/93/118/143), C (objects 19/44/69/94/119/144) and D (objects 20/45/70/95/120/145) due to space problems, but they are present.

## Communication objects with input functions

#						Object name	Object function	Description	Datapoint type
Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6				
0	25	50	75	100	125	Ch.x - Block	Switching On/Off	Allows to activate/deactivate the block function	[1.3] DPT_Enable
2	27	52	77	102	127	Ch.x – Scene storing trigger	Store	Receive the sending learn scene message request (trigger)	[1.17] DPT_Trigger
21	46	71	96	121	146	Ch.x – Dimmer status feedback	On/Off status	Dimmer activation status feedbacks	[1.1] DPT_Switch
21	46	71	96	121	146	Ch.x – Dimmer status feedback	On/Off status	Dimmer activation status feedbacks	[1.1] DPT_Switch
21	46	71	96	121	146	Ch.x – A object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object A	[1.1] DPT_Switch
21	46	71	96	121	146	Ch.x – Movement feedback	Increase/Decrease	Receive movement direction feedback from shutter actuator	[1.8] DPT_UpDown
22	47	72	97	122	147	Ch.x – B object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object B	[1.1] DPT_Switch
23	48	73	98	123	148	Ch.x – C object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object C	[1.1] DPT_Switch
24	49	74	99	124	149	Ch.x – D object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object D	[1.1] DPT_Switch
150	161	172	183	194	205	Led x - Effect 1	1/0 value	Plays the associated light effect	[1.2] DPT_Bool
151	162	173	184	195	206	Led x - Effect 2	1/0 value	Plays the associated light effect	[1.2] DPT_Bool
152	163	174	185	196	207	Led x - Effect 3	1/0 value	Plays the associated light effect	[1.2] DPT_Bool
153	164	175	186	197	208	Led x - Effect 4	1/0 value	Plays the associated light effect	[1.2] DPT_Bool
154	165	176	187	198	209	Led x - Effect 5	1/0 value	Plays the associated light effect	[1.2] DPT_Bool
155	166	177	188	199	210	Led x - Customize night signalling	Set RGB colour	Personalises the colour of the night signalling	[232.600] DPT_Colour_RGB
156	167	178	189	200	211	Led x - Cutomize effect 1	Set RGB colour	Personalises the colour of the light effect	[232.600] DPT_Colour_RGB
157	168	179	190	201	212	Led x - Customize effect 2	Set RGB colour	Personalises the colour of the light effect	[232.600] DPT_Colour_RGB
158	169	180	191	202	213	Led x - Customize effect 3	Set RGB colour	Personalises the colour of the light effect	[232.600] DPT_Colour_RGB
159	170	181	192	203	214	Led x - Customize effect 4	Set RGB colour	Personalises the colour of the light effect	[232.600] DPT_Colour_RGB
160	171	182	193	204	215	Led x - Customize effect 5	Set RGB colour	Personalises the colour of the light effect	[232.600] DPT_Colour_RGB
216						Touch sound effects enabling	Enable/Disable	Receives activation/deactivation commands of Touch sound effect playback	[1.3] DPT_Enable
217						Sound effect	Switching On/Off	Receives sound effect activation/deactivation commands	[1.1] DPT_Switch
218						Soft reduction	Switching On/Off	Receives function activation/deactivation commands	[1.1] DPT_Switch
219						Glass cleaning	Switching On/Off	Enable/Disable sensor inhibition for glass cleaning function	[1.10] DPT_Start
220						TEST RGB colour	Set personalised colour	Receives the RGB colour components to be tested	[232.600] DPT_Colour_RGB
221						TEST RGB colour brightness	Set colour brightness	Receives the personalised colour brightness to be tested	[5.1] DPT_Scaling

## 19 Signalling of plate absence/presence and BUS voltage recovery

Even if the device is powered, it will not launch the application if the glass plate or installation plate is not fitted.

If the device is powered when there is no plate, all six LEDs will flash red with a frequency of 3hZ. Once the plate has been fitted, the device recognises this fact and begins calibrating the capacitive sensors; this is indicated by the simultaneous green flashing of all six LEDs at a frequency of 3hZ. When the calibration has been completed, the application is launched; Initialisation is indicated by the activation of all six LEDs in the sequence red →green→ blue.

If the plate is removed during normal operation, the device will stop the application and all six LEDs will flash red at a frequency of 3hZ. Once the plate has been fitted again, the device recognises this fact and begins calibrating the capacitive sensors; this is indicated by the simultaneous green flashing of all six LEDs at a frequency of 3hZ. When the calibration has been completed, the application is relaunched; Initialisation is indicated by the activation of all six LEDs in the sequence red →green→ blue. Any BUS signals received while there is no plate are processed when the application is relaunched.

The start-up of the device (with the plate fitted) following a BUS failure is indicated by the sequential activation of all six LEDs - red→green→blue.

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](mailto:qualitymarks@gewiss.com)**



**+39 035 946 111**

8.30 - 12.30 / 14.00 - 18.00  
lunedì ÷ venerdì - monday ÷ friday



**+39 035 946 260**



**[sat@gewiss.com](mailto:sat@gewiss.com)**  
**[www.gewiss.com](http://www.gewiss.com)**