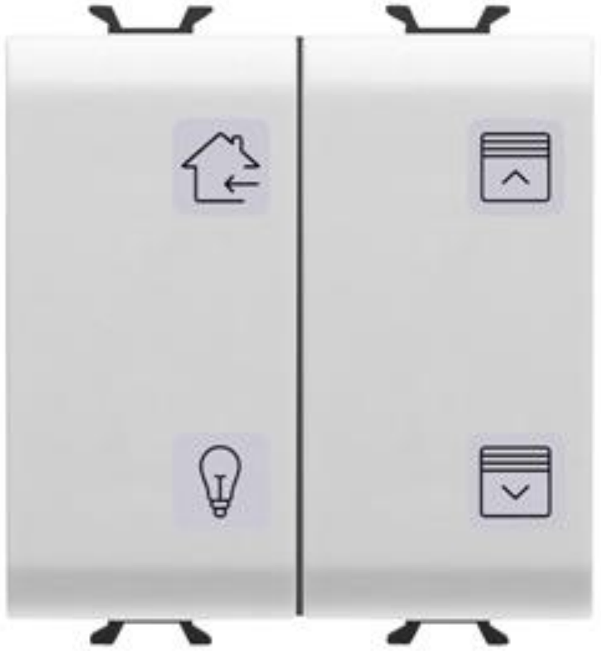
CHORUS

GEWISS

KNX 4-channel push-button panel with interchangeable symbols



GW10787
GW12787
GW14787

Technical Manual

Contents

1 Introduction

The KNX 4-channel push-button panel with interchangeable symbols (flush-mounting) is a command device with 4 channels that can be used individually or combined to implement the on/off command, dimmer control, roller shutter control, scene management, priority and timed commands on the KNX BUS. The device can be completed with push-buttons of 1 or 2 modules, tilting or non-tilting (as shown in figure B). One tilting push-button manages two channels (independent or combined).
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 push-button panel module is positioned inside a standard flush-mounting box, assembled in the supports of the Chorus range (in the space of two modules).

2 Application

Each of the 4 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
- management of brief/prolonged touch 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 presses:
- management of multiple presses on consecutive drives, up to a maximum of 4

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

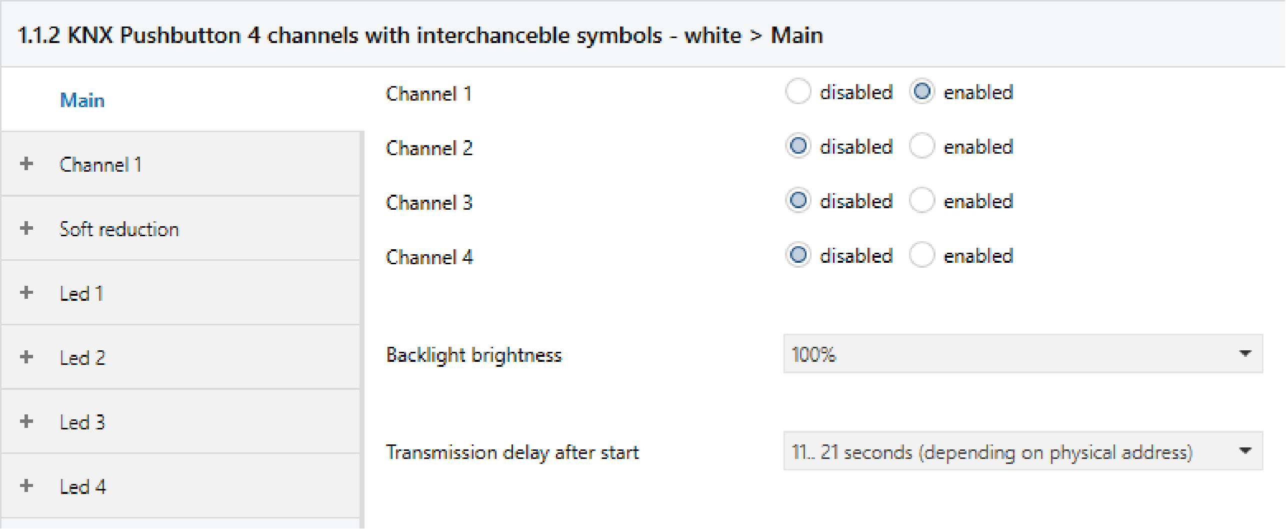
2.1 Association limits

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 (between communication objects and group addresses).

3 *“Main”* menu

The **Main** menu contains the application parameters relating to all 4 channels managed by the device (Fig. 3.1).



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Main

Channel 1 ○ disabled ● enabled

Channel 2 ● disabled ○ enabled

Channel 3 ● disabled ○ enabled

Channel 4 ● disabled ○ enabled

Backlight brightness 100%

Transmission delay after start 11.. 21 seconds (depending on physical address)

Fig. 3.1

3.1 Channel X

Each of the 4 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”** are used to configure the relative input channels, displaying the configuration menus. The values that can be set are:

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

If **enabled** is selected, the **Channel 1**, **Channel 2**, **Channel 3**, **Channel** configuration menu is displayed (see par. 4 “Channel x” menu).

3.2 Backlight brightness

Used to define the intensity level of the backlighting for all 4 LEDs on the device. The values that can be set are:

- **100%**	**(default value)**
- 90%	
- 80%	
- 70%	
- 60%	
- 50%	
- 40%	
- 30%	
- 20%	
- 10%	
- 5%	

3.3 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 **“Transmission delay after start”** parameter is used to define this delay. The values that can be set are:

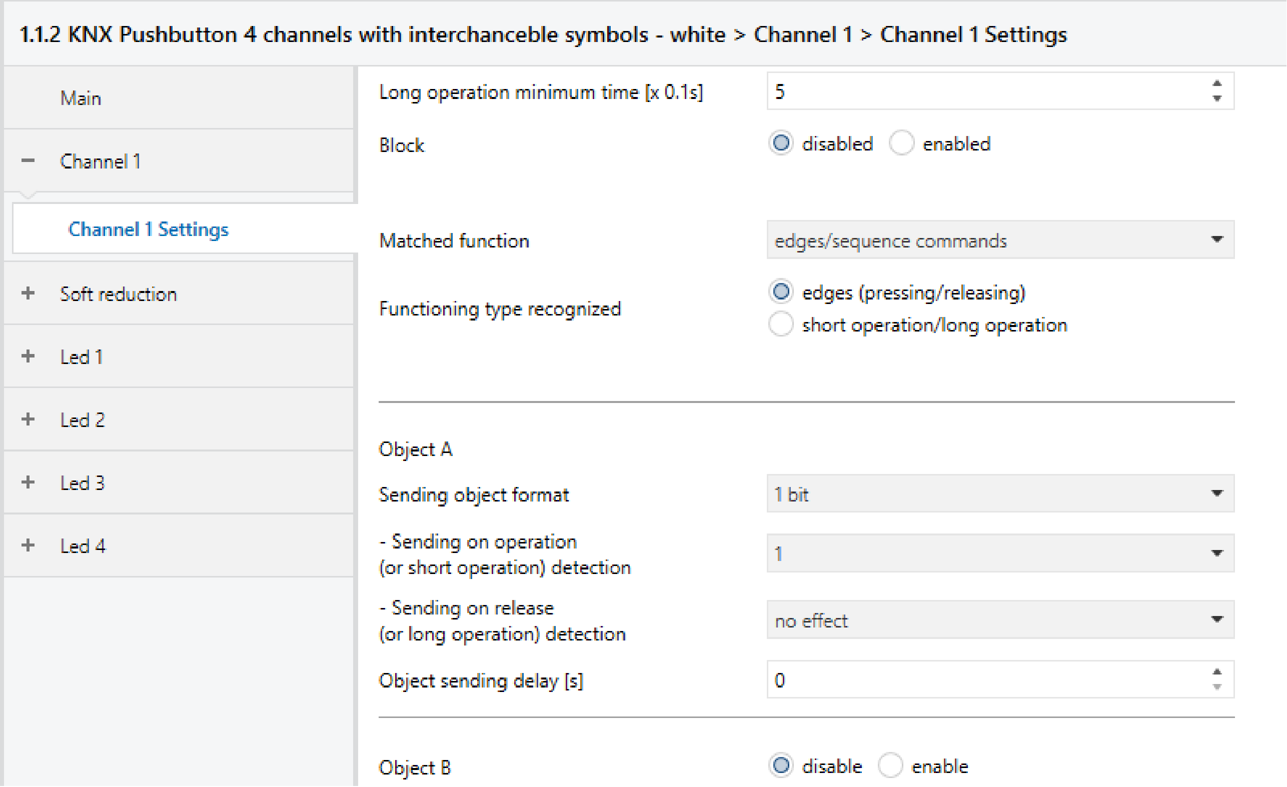
- **11.. 21 seconds (depending on physical address) (default value)**
- 5.. 9 seconds
- 11 seconds
- 13 seconds
- 15seconds
- 17 seconds
- 19 seconds
- 21 seconds
- no delay

If the value **11.. 21 seconds (depending on physical address)** or **5.. 9 seconds** is set, the device automatically calculates the transmission delay using 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

If a channel is enabled, a dedicated menu - called **Channel x** (x = 1 .. 4, is the input index) - is displayed for each input. The menu structure will change according to 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):



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s]: 5

Block: ● disabled ○ enabled

Matched function: edges/sequence commands

Functioning type recognized: ● edges (pressing/releasing) ○ short operation/long operation

Object A

Sending object format: 1 bit

- Sending on operation (or short operation) detection: 1

- Sending on release (or long operation) detection: no effect

Object sending delay [s]: 0

Object B: ● disable ○ enable

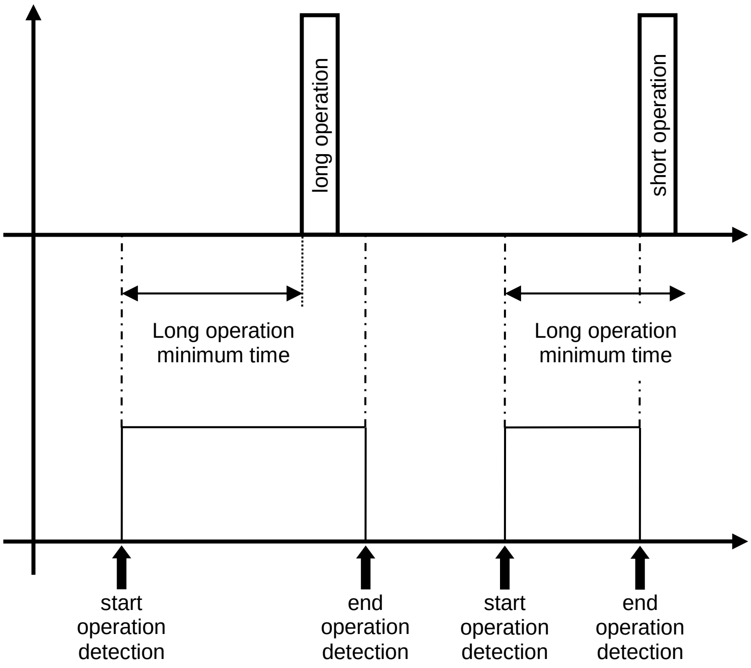
Fig 4.1

4.1 *Long operation minimum time [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 can be used to define the minimum time in which the device must detect the pressing of the command button key in order to distinguish between a long touch and a short one. The possible values are:

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

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


long operation
short operation
Long operation minimum time
Long operation minimum time
start operation detection
end operation detection
start operation detection
end operation detection

4.2 Block

This inhibits the detection of the touch/release of the command button key, 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 may have the following values:

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

If **enabled** is selected, the **"Block activation value"** and **"Block function on bus voltage recovery"** parameters, and the ***Ch.x - Block*** communication object, are displayed so that the function can be activated by means of a BUS command.
The **"Block activation value"** parameter is used to define which logic value the bit received via BUS telegram should assume in order to activate the block function. The values that can be set are:

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

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

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

4.3 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 - **Function *"edges/sequence commands"***

- ***1 push button dimmer + stop***

See chapter 6 - **Function *"1 push button dimmer + stop***

- ***1 push button dimmer with cyclic sending***

See chapter 7 - **Function** *“**1 push button dimmer with cyclic sending***

- ***1 push button shutter control***

See chapter 8 - **Function***“**1** push button shutter control*

- ***2 push button dimmer + stop***

See chapter 9 - **Function** *“2 push button dimmer + stop*

- ***2 push button dimmer with cyclic sending***

See chapter 10 - **Function** *“**2 push button dimmer with cyclic sending***

- ***2 push button shutter control***

See chapter 11 - **Function** *“*2 push button shutter control

- ***Scene management***

See chapter 12 - **Function** *“**scene management***

- ***Switching sequences***

See chapter 13 - **Function** *“**switching sequences***

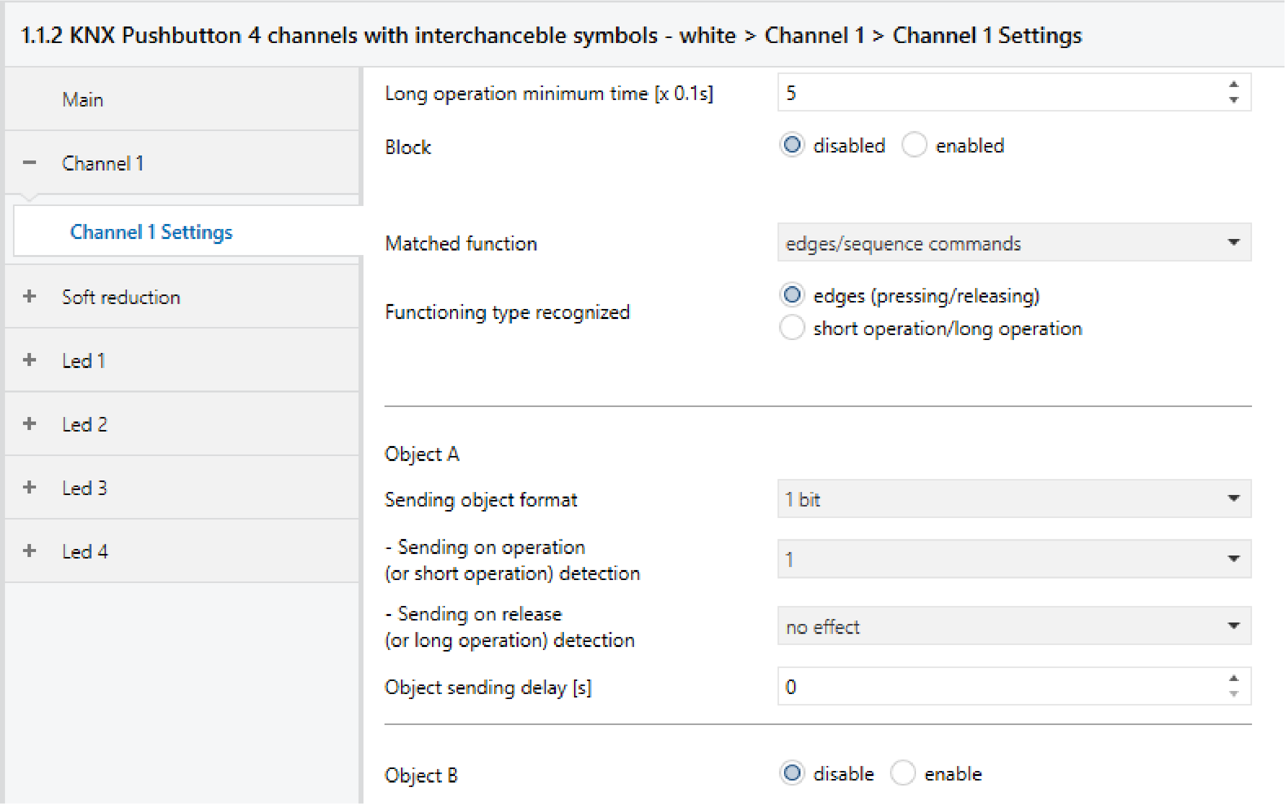
- ***Multiple press***

See chapter 14 - **Function** ***“multiple press***

5 Function “edges/sequence commands”

This function is used to set the type and number of commands to send after a status change has been detected, for up to a total of 4 commands per channel. It is possible to differentiate the command value 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):



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s]: 5

Block: ● disabled ○ enabled

Matched function: edges/sequence commands

Functioning type recognized: ● edges (pressing/releasing) ○ short operation/long operation

Object A

Sending object format: 1 bit

- Sending on operation (or short operation) detection: 1

- Sending on release (or long operation) detection: no effect

Object sending delay [s]: 0

Object B: ● disable ○ enable

Fig. 5.1

5.1 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/releasing)** **(default value)**
- short operation/long operation

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

For each input, up to 4 different objects can be sent (distinguished by the letters A, B, C and D) on the basis of the touch (or short operation) or release (or long operation) of the button key associated with the channel. Object A is always enabled, but 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 **“Sending object format”**, **“Sending on operation (or short operation) detection”**, **“Sending on release (or long operation) detection”** and **“Object sending delay [s]”** parameters will be displayed, grouped in the **Object z** sub-set (where z is the index of the object associated with the input, between **A** and **D**).

The **“Sending object format”** parameter is used 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
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC mode
- 2 byte unsigned
- 2 byte signed
- 3 byte RGB color
- 4 byte unsigned
- 4 byte signed
- 14 byte

Depending on the value set for this item, the values that can be set for the **“Sending on operation (or short operation) detection”** and **“Sending on release (or long operation) detection”** parameters will change.

The **“Sending on operation (or short operation) detection”** parameter is used to set the command or value to be sent following the detection of the pressing or short pressing (depending on the type of operation selected) on the button key associated with the channel.

The **“Sending on release (or long operation) detection”** parameter is used to set the command or value to be sent following the detection of the release or long pressing (depending on the type of operation selected) on the button key associated with the channel.

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

  - **no effect** **(release detection default value)**
  - 0
  - **1** **(operation detection default value)**
  - cyclical switching

  If **cyclical switching** is selected, the **“Status feedback object”** parameter is displayed for enabling the viewing of the ***Ch.x – z object status feedback*** (Data Point Type: 1.001 DPT_Switch) communication object. If this object is enabled, when a status notification telegram is received on the object in question, the command sent by the push-button panel (via the ***Ch.x –z object 1 bit value*** object) upon detection of the event associated with cyclical sending will be the opposite of the value generated by the most recent of two events - the arrival of the BUS value on the ***Ch.x – z object status feedback*** object, or the last value sent on the ***Ch.x –z object 1 bit value*** object.

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

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

  If **enabled** is selected, the ***Ch.x – z object status feedback*** communication object is displayed. 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 be sent is **2 bit**, the ***Ch.x – z object 2 bits value*** (Data Point Type: 2.001 DPT_Switch_Control) communication object is displayed and the values that can be set for the two parameters above are:

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

- cyclical switching forcing on/deactivate forcing
- cyclical switching forcing off/deactivate forcing

If **cyclical switching** is selected, no communication object is displayed because the device is always updated about the function activation status.

– If the format of the object to be sent is **1 byte unsigned**, the ***Ch.x – z object 1 byte value*** (Data Point Type: 5.010 DPT_Value_1_Ucount) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect (default value on detection of release)**
- **value sending (default value on detection of operation)**

If **value sending** is set, it is possible to define the value to be sent via the new parameter displayed (**"Value (0 .. 255)"**), which may have the following values:

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

– If the format of the object to be sent is **1 byte signed**, the ***Ch.x – z object 1 byte value*** (Data Point Type: 6.010 DPT_Value_1_Count) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect (default value on detection of release)**
- **value sending (default value on detection of operation)**

If **value sending** is set, it is possible to define the value to be sent via the new parameter displayed (**"Value (-128 .. 127)"**), which may have the following values:

- from -128 to 127 **(default value0)**

– If the format of the object to be sent is **1 byte percentage**, the ***Ch.x – z object 1 byte value*** (Data Point Type: 5.001 DPT_Scaling) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect (default value on detection of release)**
- **value sending (default value on detection of operation)**

If **value sending** is set, it is possible to define the value to be sent via the new parameter displayed (**"Value (0% .. 100%)"**), which may have the following values:

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

– If the format of the object to be sent is **1 byte HVAC mode**, the ***Ch.x – z object 1 byte value*** (Data Point Type: 20.102 DPT_HVACMode) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect (default value - release)**
- auto
- **comfort (default value - operation)**
- precomfort
- economy
- off (building protection)
- cyclical switching (thermostat)
- cyclical switching (chronothermostat)

If **cyclical switching** is selected, no communication object is displayed because the device is always updated about the function activation status.

If **cyclical switching (thermostat)** is selected, every time the associated event (operation/release) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→*

Precomfort→ Economy→ Off→ Comfort …. If **cyclical switching (chronothermostat** is selected, every time the associated event (operation/release) is detected, the device sends a new temperature adjustment mode (HVAC**)** in the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* ….

– If the format of the object to be sent is **2 byte unsigned**, the ***Ch.x – x object 2 bytes value*** (Data Point Type: 7.001 DPT_Value_2_Ucount) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect** **(default value on detection of release)**
- **value sending** **(default value on detection of operation)**

If **value sending** is set, you can define the value to be sent via the new parameter displayed (**“Value (0 .. 65535)”**), which may have the following values:

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

– If the format of the object to be sent is **2 byte signed**, the ***Ch.x – x object 2 bytes value*** (Data Point Type: 8.001 DPT_Value_2_Count) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect** **(default value on detection of release)**
- **value sending** **(default value on detection of operation)**

If **value sending** is set, you can define the value to be sent via the new parameter displayed (**“Value (-32768 .. +32767)”**), which may have the following values:

- from -32768 to +32767 **(default value0)**

– If the format of the object to be sent is **3 byte RGB color**, the ***Ch.x – z object 3 byte value*** (Data Point Type: 232.600 DPT_Colour_RGB) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect** **(default value on detection of release)**
- **value sending** **(default value on detection of operation)**

If **value sending** is selected, you can define the colour to be sent via the **“Color”** dummy parameter. The values that can be set are:

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

If **customize** is selected, the **“RED component value (0 .. 255)”**, **“GREEN component value (0 .. 255)”** and **“BLUE component value (0 .. 255)”** parameters are displayed. 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, in steps of 1

– If the format of the object to be sent is **4 byte unsigned**, the ***Ch.x – z object 4 bytes value*** (Data Point Type: 12.001 DPT_Value_4_Ucount) communication object is displayed and the values that can be set for the two parameters above are:

- **no effect** **(default value on detection of release)**

- **value sending** **(default value on detection of operation)**

  If **value sending** is set, you can define the value to be sent via the new parameter displayed (**“Value (0 .. 4294967295”**), which may have the following values:

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

- If the format of the object to be sent is **4 byte signed**, the ***Ch.x – z object 4 bytes value*** (Data Point Type: 13.001 DPT_Value_4_Count) communication object is displayed and the values that can be set for the two parameters above are:

  - **no effect** **(default value on detection of release)**
  - **value sending** **(default value on detection of operation)**

  If **value sending** is set, you can define the value to be sent via the new parameter displayed (**“Value (-2147483648 .. 2147483647”**), which may have the following values:

  - from -2147483648 to 2147483647 **(default value0)**

- If the format of the object to be sent is **14 byte**, the ***Ch.x – z object 14 bytes value*** (Data Point Type: 16.001 DPT_String_8859_1) communication object is displayed and the values that can be set for the two parameters above are:

  - **no effect** **(default value on detection of release)**
  - **value sending** **(default value on detection of operation)**

  If **value sending** is set, it is possible to define the value to be sent via the new parameter displayed (**“Value (characters ISO 8859-1)”**), which may have the following values:

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

The **“Object sending delay [s] (0.. 255 seconds)”** parameter is used to set the delay between the detection of the event associated with the sending of the command and the actual sending of the command/value on the BUS. With regard to the objects that range from index B to index D, this parameter indicates the delay between sending the command/value associated with the object with the previous index (z-1) and sending the command/value associated with the object to which the parameter refers; The delay to which reference is made in these cases is calculated from the moment in which the command/value is sent that is associated with the object with the previous index (z-1) and not from the moment in which the event that generates sending is detected (pressing/releasing or short operation).
The set delay is only implemented if a value other than **no effect** is associated with the event in progress (associated with the object to which the parameter refers); 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.

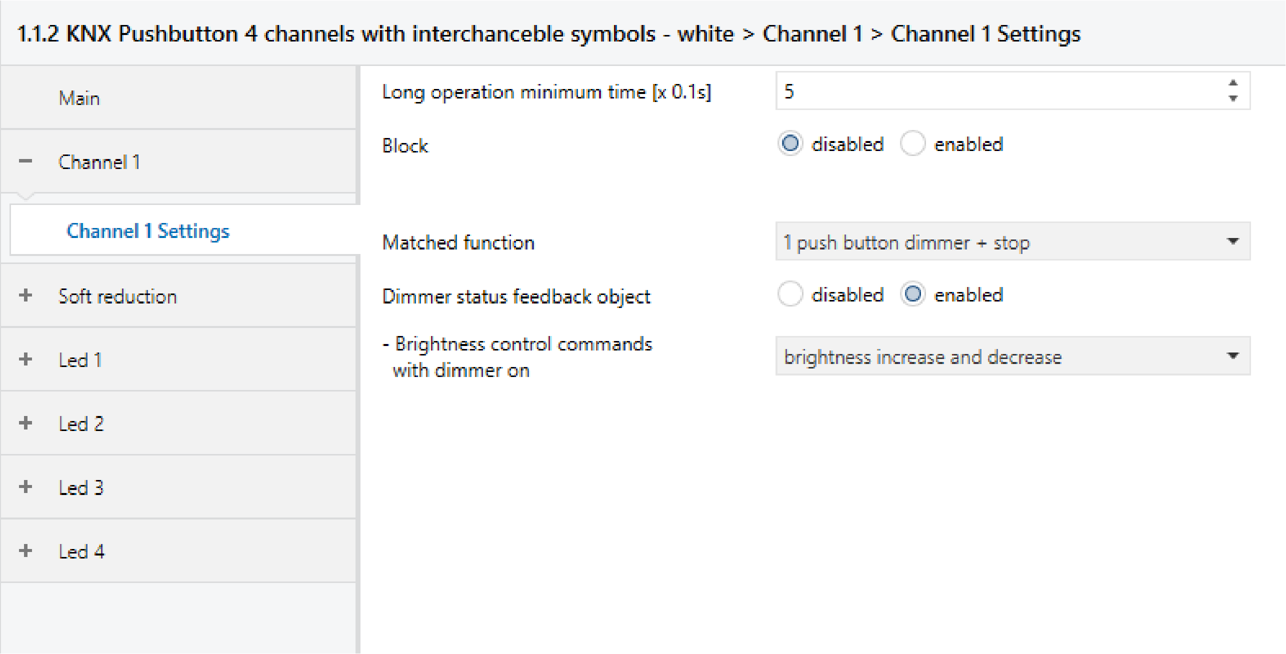
6 *“1 push button dimmer + stop”* function

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 regulation functions, operation is managed by differentiating between short touches and long touches:

- a long operation is transformed into a brightness control 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.
- a short operation 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 - Switch*** (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):



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s]: 5

Block: ● disabled ○ enabled

Matched function: 1 push button dimmer + stop

Dimmer status feedback object: ○ disabled ● enabled

- Brightness control commands with dimmer on: brightness increase and decrease

Fig. 6.1

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 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 operation: if the last sent command was an on command, the new command will be an off command; vice versa, if the last sent command was an off command, the new command will be an on command; the brightness increase/decrease 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 Dimmer status feedback object

This parameter may have the following values:

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

If **enabled** is selected, the **“Brightness control commands with dimmer on”** parameter and ***Ch.x – Dimmer status feedback*** (Data Point Type: 1.001 DPT_Switch) communication object are displayed, so status notifications can be received by the dimmer actuator controlled. The behaviour of the push-button panel is modified as follows:

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

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

  When **brightness increase and decrease** is set, if the value of the latest of the two events "last command sent" and "dimmer status notification" is ON, the new brightness regulation command to be sent will be the opposite of the last command sent; 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 operation: if the value of the last of the two events "last sent command" and "dimmer status feedback" is ON, the new command will be an off command. Vice versa, if the value of the last of the two events "last sent command" and "dimmer status feedback" is OFF, the new command will be an on command.

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

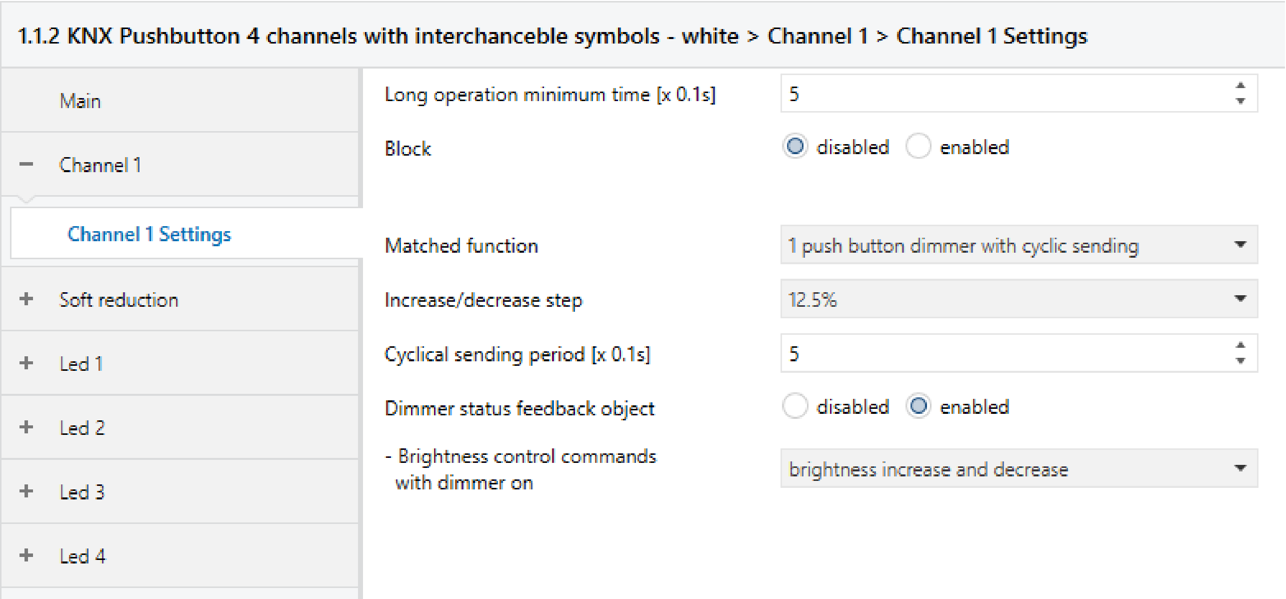
7 *“1 push button dimmer with cyclic sending”* function

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 button to manage the On/Off and brightness control functions, the operation is managed so that each time the button is pressed the opposite command is sent in comparison to the last sent command and by differentiating between short operations and long operations:

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

Unlike the **1 push button dimmer + stop** function, in this case it is possible to define both the brightness variation steps and the time that passes between the sending of one command and the next (if the long operation continues over time). This means it is not necessary to send a regulation stop telegram at the end of the operation, 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 - Switch*** (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):



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s]: 5
Block: ● disabled ○ enabled
Matched function: 1 push button dimmer with cyclic sending
Increase/decrease step: 12.5%
Cyclical sending period [x 0.1s]: 5
Dimmer status feedback object: ○ disabled ● enabled
- Brightness control commands with dimmer on: brightness increase and decrease

Fig. 7.1

7.1 *Increase/decrease step*

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

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

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

7.2 *Cyclical sending period [x 0.1s]*

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

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

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

EXAMPLE: let's suppose that for the **Long operation minimum time** item of the ***Main*** menu a value of ***0.5 sec*** has been set, with the **increase/decrease step** parameter set at **12.5%** and the **Cyclical sending period [x 0.1s]** parameter set at **3** (0.3 sec) and operation is detected.

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

7.3 *Dimmer status feedback object*

To set this parameter, refer to paragraph

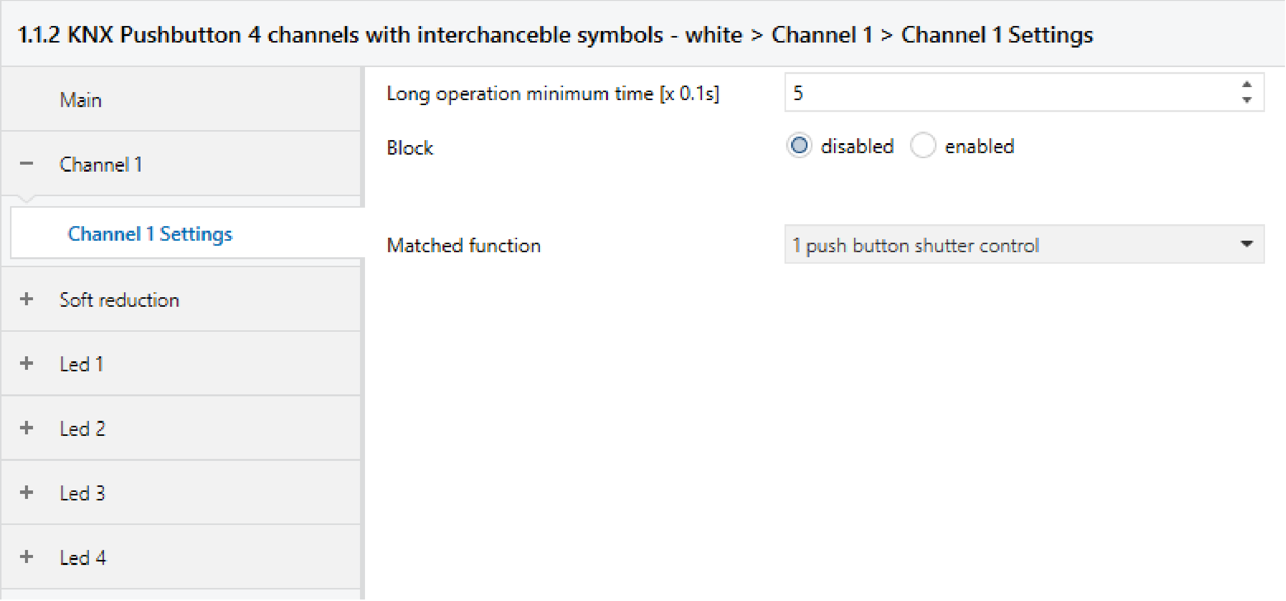
8 *“1 push button shutter control”* function

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 signal received by the actuator that manages the shutter. A differentiation is made between short and long operations:

- a long operation is transformed into an up/down movement command. The new value to be sent is the opposite of the last value sent via the ***Ch.x - Shutter movement*** object or the movement signal received via the ***Ch.x – Movement feedback*** object, depending on which of these two events occurred most recently. 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 louvres control command. The new value to be sent will depend on the last value sent via the ***Ch.x - Shutter movement*** object or the movement signal received via the ***Ch.x – Movement feedback*** object, depending on which of these two events occurred most recently; 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):


1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s] 5

Block ● disabled ○ enabled

Matched function 1 push button shutter control

Fig. 8.1

No new parameters are enabled with this function

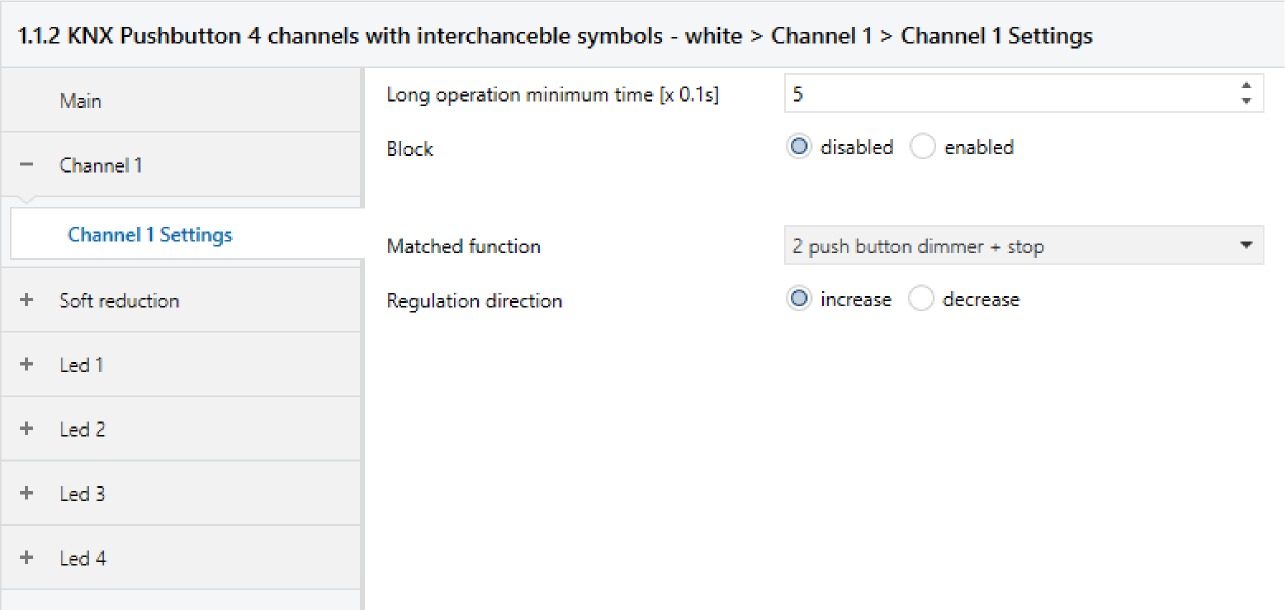
9 *“2 push button dimmer + stop”* function

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. In this case too, a differentiation is made between short and long operations:

- a long operation is transformed into a brightness control 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.
- a short operation is transformed in to an on or off command depending on the set control 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 - Switch*** (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:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s] 5

Block ● disabled ○ enabled

Matched function 2 push button dimmer + stop

Regulation direction ● increase ○ decrease

Fig. 9.1

9.1 Regulation direction

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

-	**Increase**	**(default value - odd channels)**
-	**Decrease**	**(default value - even channels)**

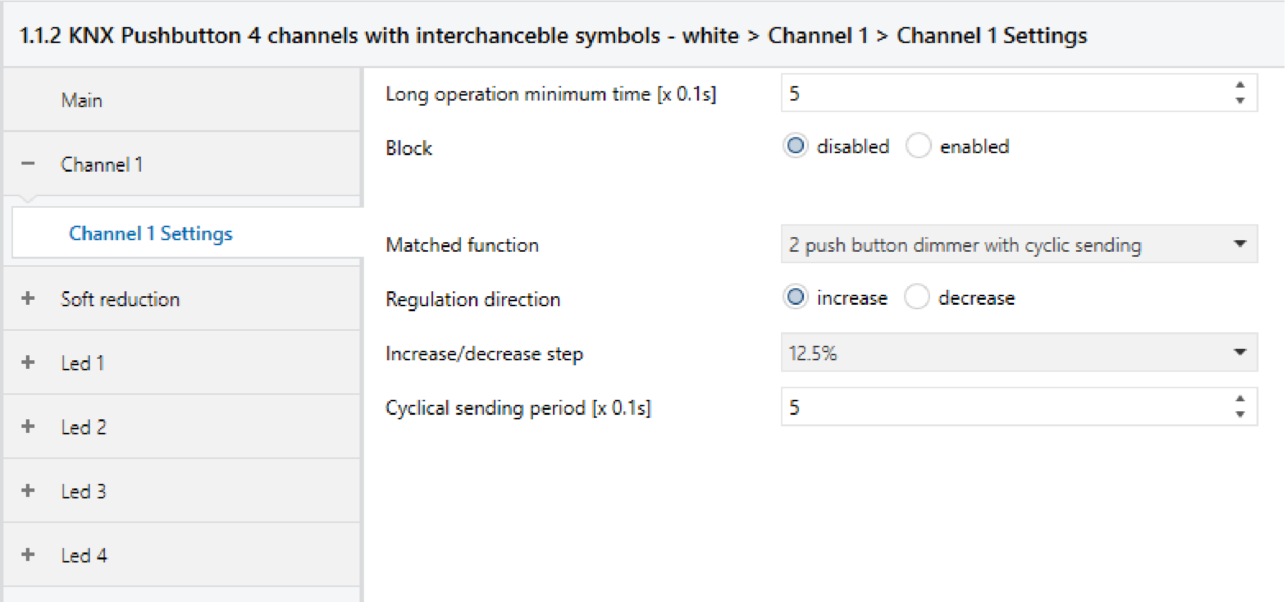
Selecting **increase**, the commands sent are "increase brightness by 100%" or "ON", depending on the operation recognised; selecting **decrease**, the commands sent are "decrease brightness by 100%" or "OFF".

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. In this case too, a differentiation is made between short and long operations:

– a long operation is transformed into a brightness control 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.
– a short operation is transformed in to an on or off command depending on the set control 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 dimmer + stop*** function, in this case it is possible to define both the brightness variation steps of the increase/decrease commands and the time that passes between the sending of one command and the next (if the long operation continues 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 - Switch*** (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:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s]: 5
Block: ● disabled ○ enabled
Matched function: 2 push button dimmer with cyclic sending
Regulation direction: ● increase ○ decrease
Increase/decrease step: 12.5%
Cyclical sending period [x 0.1s]: 5

Fig. 10.1

10.1 *Regulation direction*

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

-	**Increase**	**(default value - odd channels)**
-	**Decrease**	**(default value - even channels)**

Selecting **Increase**, the commands sent are "increase brightness by 100%" or "ON", depending on the operation recognised; selecting **Decrease**, the commands sent are "decrease brightness by 100%" or "OFF".

10.2 *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 operation is detected, the device sends the first increase/decrease command with the set percentage. The values that can be set are:

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

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

10.3 Cyclical sending period [x 0.1s]

The **"Cyclical sending period [x 0.1s]"** parameter sets the time that passes between the sending of one increase/decrease command and the next, if the button key is kept pressed. When it is released, no telegram is sent but the cyclical sending of the brightness regulation commands is stopped.
The values that can be set for the **"Cyclical sending period [x 0.1s]"** parameter are:

- from 3 to 50, in steps of 1 **(default value 5)**

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

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

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

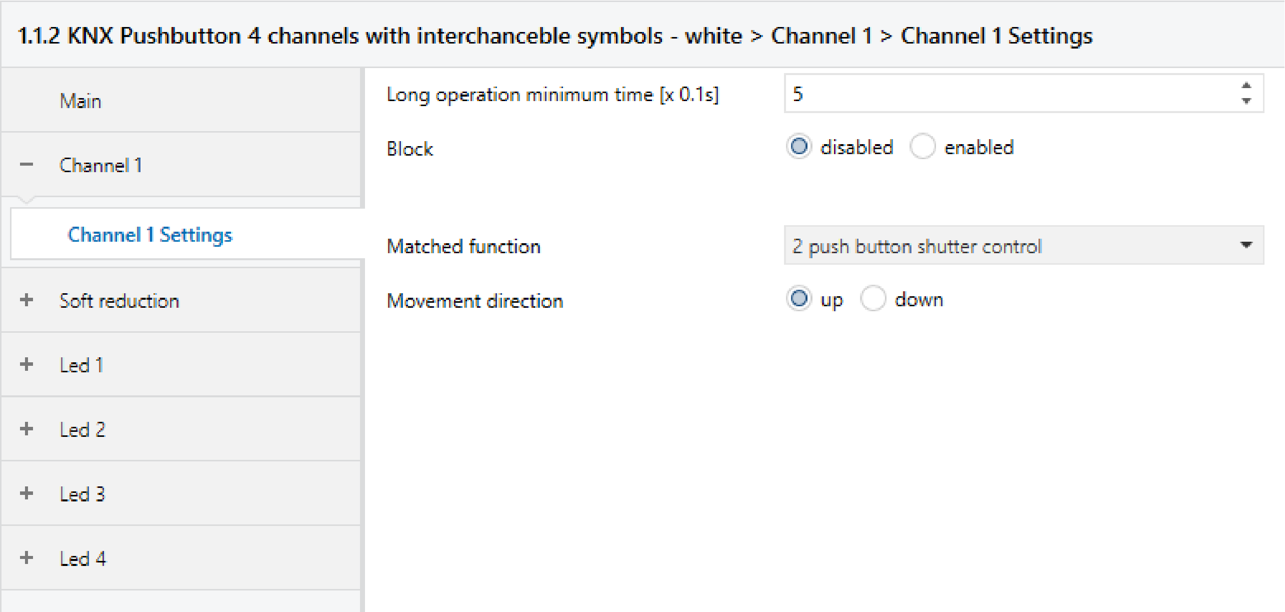
11 “*2 push button shutter control* function

This is used to configure the channel to control a Shutter/Venetian blind with two 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. In this case too, a differentiation is made between short and long operations:

- a long operation is considered as 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 operation is considered as a slat regulation command (or movement stop, if the shutter is moving) on opening or closure, depending on the set movement direction. If the set movement direction is "up", the sent command will only be a slat opening control command (or stop movement); If the set regulation direction is "down", the sent command will only be a slat 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/Louvres control*** (Data Point Type: 1.007 DPT_Step).

The structure of the menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s] 5

Block ● disabled ○ enabled

Matched function 2 push button shutter control

Movement direction ● up ○ down

Fig. 11.1

11.1 Movement direction

The **“Movement direction”** parameter configures the movement direction of the roller shutter controlled by the channel. The values that can be set are:

-	**up**	**(default value - odd channels)**
-	**down**	**(default value - even channels)**

Selecting **up**, the commands sent are "up movement" or "slat regulation on opening" (movement stop), depending on the operation recognised; selecting **down**, the commands sent are "down movement" or "slat regulation on closure" (movement stop).

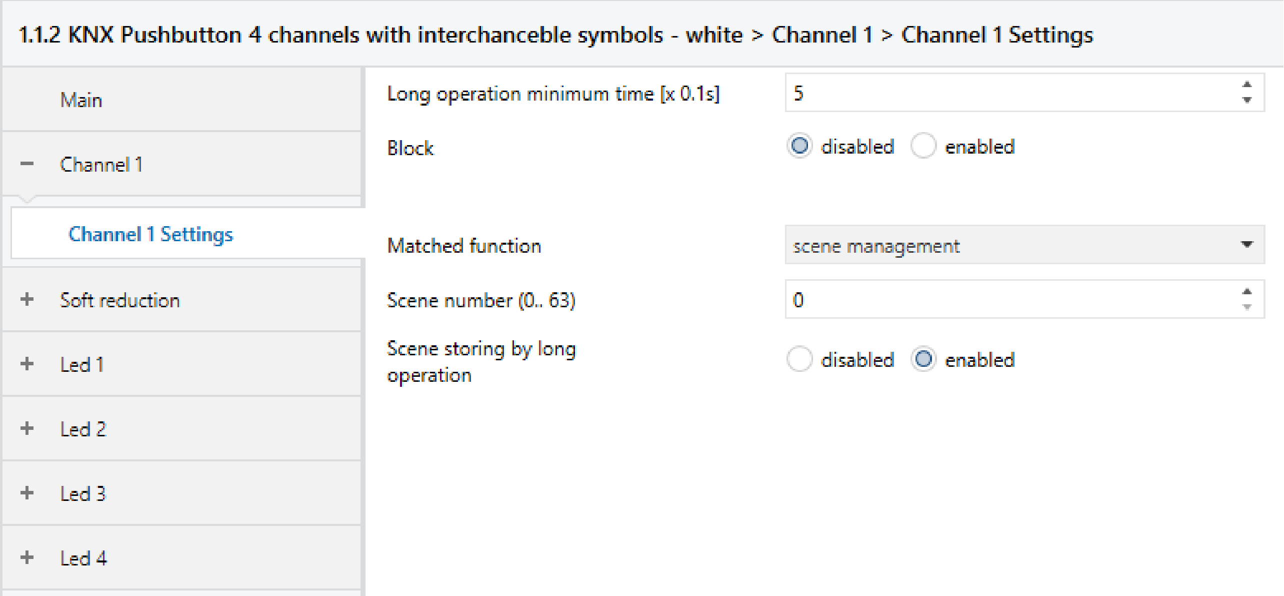
12 *“Scene management”* function

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.
A differentiation is made between short and long operations:

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

The communication objects 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):



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s] 5

Block ● disabled ○ enabled

Matched function scene management

Scene number (0.. 63) 0

Scene storing by long operation ○ disabled ● enabled

Fig. 12.1

12.1 Scene number (0..63)

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

- from **0 (default value)** to 63, in steps of 1

12.2 Scene storing by long operation

This enables the sending of a scene memorising command when a long operation is recognised. The values that can be set are:

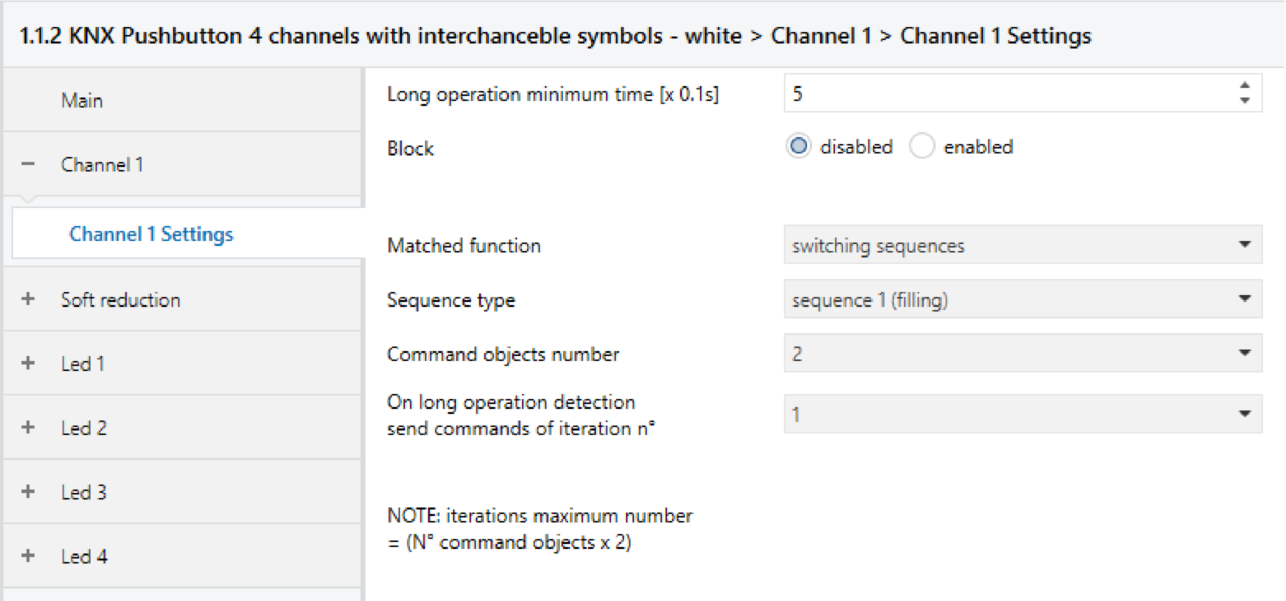
- disabled
- **enabled** **(default value)**

The device will only send the scene storage command if **enabled** is selected, and only when a long operation is detected; if **disabled**, is selected, a long operation is not recognised so a scene execution command is sent, as for a short operation.

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

13 *“Switches sequences”* function

This is used to send a command sequence after a certain pressing has been detected.
The structure of the menu is as follows (Fig. 13.1):



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

Long operation minimum time [x 0.1s]: 5

Block: ● disabled ○ enabled

Matched function: switching sequences

Sequence type: sequence 1 (filling)

Command objects number: 2

On long operation detection send commands of iteration n°: 1

NOTE: iterations maximum number = (N° command objects x 2)

Fig. 13.1

13.1 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 (random)

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

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

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

As you can see in the table showing the upward/downward trend of the sequence, the most significant bit of the sequence in this particular case is that of the ***Ch.x – C Sequence*** communication object, whereas the least significant is that of the ***Ch.x – A Sequence*** object.

sequence 2 (sum) acts as follows:

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

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

Once the 8th edge is detected, the sequence will start 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. In this case too, the most significant bit of the sequence is that of the ***Ch.x – C Sequence*** communication object, whereas the least significant is again that of the ***Ch.x – A Sequence*** object.

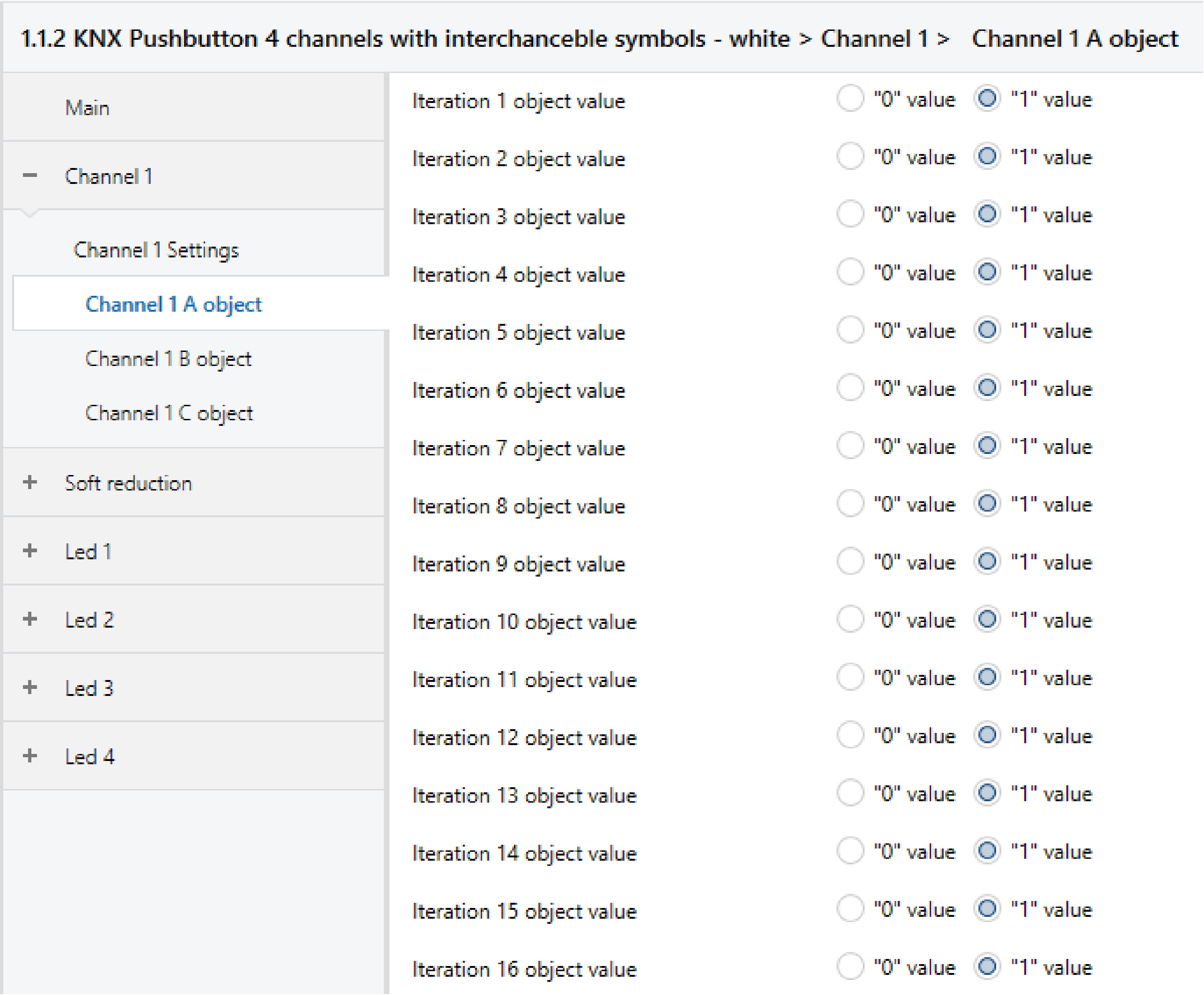
sequence 3 (random) allows the user to directly set the value of each command for each edge set. With this setting, the **“Number of iterations of the sequence”** parameter is enabled, along with the **Channel x z object** configuration menus (one for every command enabled). The **“Number of iterations of the sequence”** parameter sets the number of iterations (edges) that make up the sequence. The values that can be set are:

- from **2 (default value)**to 16, in steps of 1

Depending on the value set for this item, the **Channel x z object** menus may or may not display the **“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”** e **“iteration 16 object value”** parameters, that may have the following values:

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

The structure of the **Channel x z object**menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 A object

Iteration 1 object value ○ "0" value ● "1" value

Iteration 2 object value ○ "0" value ● "1" value

Iteration 3 object value ○ "0" value ● "1" value

Iteration 4 object value ○ "0" value ● "1" value

Iteration 5 object value ○ "0" value ● "1" value

Iteration 6 object value ○ "0" value ● "1" value

Iteration 7 object value ○ "0" value ● "1" value

Iteration 8 object value ○ "0" value ● "1" value

Iteration 9 object value ○ "0" value ● "1" value

Iteration 10 object value ○ "0" value ● "1" value

Iteration 11 object value ○ "0" value ● "1" value

Iteration 12 object value ○ "0" value ● "1" value

Iteration 13 object value ○ "0" value ● "1" value

Iteration 14 object value ○ "0" value ● "1" value

Iteration 15 object value ○ "0" value ● "1" value

Iteration 16 object value ○ "0" value ● "1" value

Fig. 13.2

13.2 Command object 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 – z Sequence*** (Data Point Type: 1.001 DPT_Switch) communication objects will be enabled, with **z** between A and D. The values that can be set are:

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

13.3 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 defines which sequence iteration should be sent if a long operation is detected. The values that can be set are:

- from 1 to 16, in 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 operation is detected, the device will send:

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

"Filling" sequence

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

"Sum" sequence

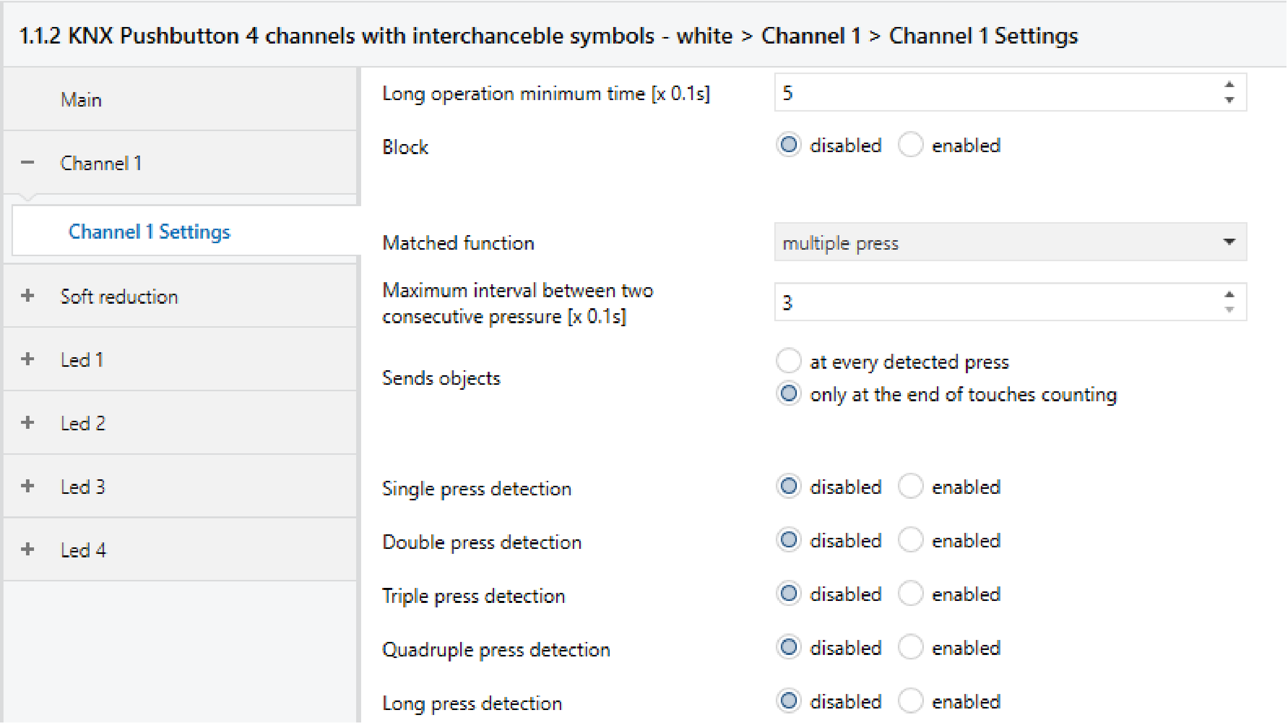
Once the long operation is detected and the sequence relative to the set iteration is sent, the next time a short operation is detected the sequence related to the iteration following the one associated with the long operation will be sent (in this example, the sequence association with iteration 4 will be sent).

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 for initialising the iteration counter when a long 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 press”* 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 (up to 4 commands per channel).

The structure of the menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 Settings

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

- single press detection → one operation detected
- double press detection → two consecutive operations detected
- triple press detection → three consecutive operations detected
- quadruple press detection → four consecutive operations detected
- long press detection → one long operation detected

Five consecutive operations or more are interpreted as a “quadruple press”.

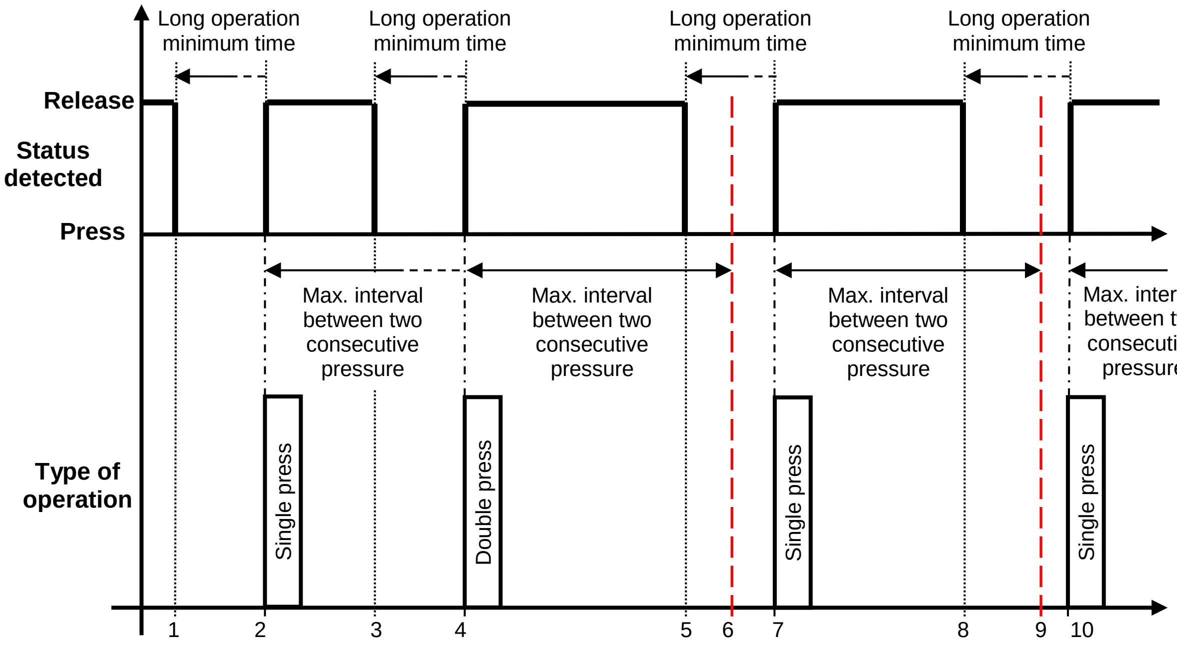
In order to recognise two consecutive operations, it's necessary to define the maximum gap between the detection of one press and the next; if the effective time is less than the maximum gap, the multiple press count is increased. When the time that elapses after the detection of an operation exceeds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

14.1 Maximum interval between two consecutive pressure [x 0.1s]

The **“Maximum interval between two consecutive pressure [x 0.1s]”** parameter defines the maximum gap between the detection of one operation and the next in order to recognise them as consecutive. The values that can be set are:

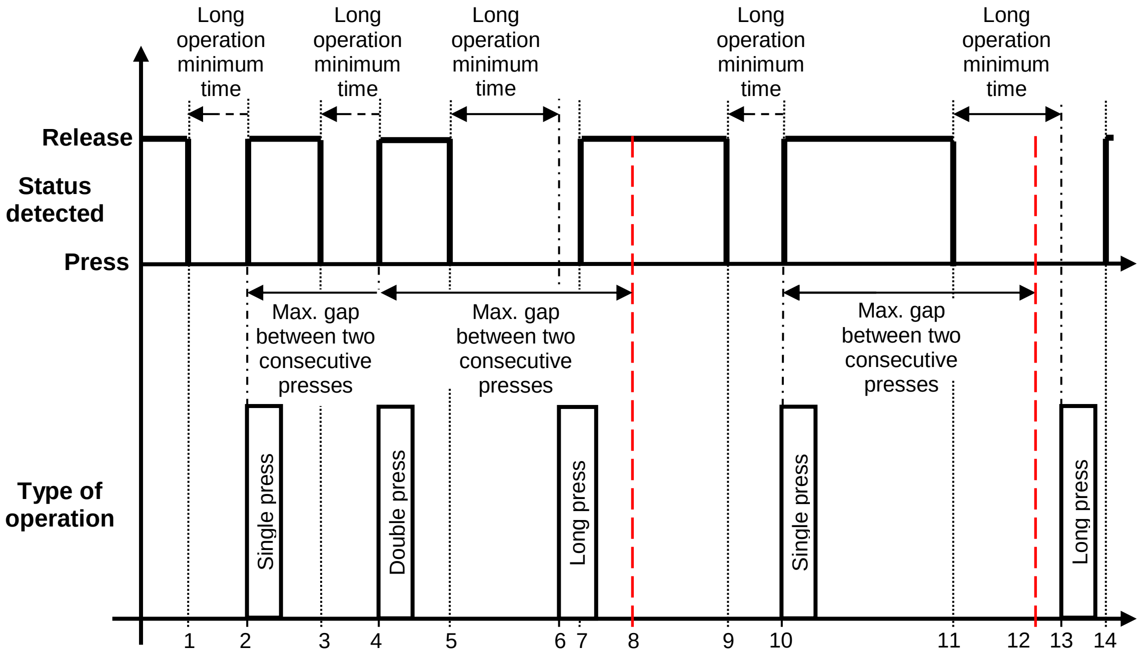
- from **3 (default value)** to 100, in steps of 1

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


Long operation minimum time
Long operation minimum time
Long operation minimum time
Long operation minimum time
Release
Status detected
Press
Max. interval between two consecutive pressure
Max. interval between two consecutive pressure
Max. interval between two consecutive pressure
Max. interval between two consecutive pressure
Type of operation
Single press
Double press
Single press
Single press
1
2
3
4
5
6
7
8
9
10

1. Once a press has been detected, the count of the press duration is initialised in order to distinguish between a short operation and a long one.
2. If the button key is released before the end of the long operation minimum time, a short press is recognised; the count of the gap between two consecutive presses is begun, and the multiple press count is increased.
3. The detection of a new press leads to the initialisation of the press duration time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
4. The release of the button key before the end of the long operation minimum time, and before reaching the maximum gap between two consecutive presses, means the detection of a new short press that increases the multiple press count and re-initialises the calculation of the gap between two consecutive presses.
5. The detection of a new press leads to the initialisation of the press duration time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
6. Once the maximum gap between two consecutive presses (dotted red line) has elapsed, the multiple press count is terminated and, after sending the KNX commands relating to this action, the counter is reset.
7. If the button key is released before the end of the long operation minimum time, a new short press is recognised; the count of the gap between two consecutive presses is begun, and the multiple press count is increased.
8. The detection of a new press leads to the initialisation of the press duration time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
9. Once the maximum gap between two consecutive presses (dotted red line) has elapsed, the multiple press count is terminated and, after sending the KNX commands relating to this action, the counter is reset.
10. If the button key is released before the end of the long operation minimum time, a new short press is recognised; the count of the gap between two consecutive presses is begun, and the multiple press count is increased.

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


Long operation minimum time
Long operation minimum time
Long operation minimum time
Long operation minimum time
Long operation minimum time
Release
Status detected
Press
Max. gap between two consecutive presses
Max. gap between two consecutive presses
Max. gap between two consecutive presses
Type of operation
Single press
Double press
Long press
Single press
Long press
1
2
3
4
5
6
7
8
9
10
11
12
13
14

1. Once a press has been detected, the count of the press duration is initialised in order to distinguish between a short operation and a long one.
2. If the button key is released before the end of the long operation minimum time, a short press is recognised; the count of the gap between two consecutive presses is begun, and the multiple press count is increased.
3. The detection of a new press leads to the initialisation of the press duration time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
4. The release of the button key before the end of the long operation minimum time, and before reaching the maximum gap between two consecutive presses, means the detection of a new short press that increases the multiple press count and re-initialises the calculation of the gap between two consecutive presses.
5. The detection of a new press leads to the initialisation of the press duration time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
6. If the pressing continues for a time greater than the minimum duration of a long operation, a long press is recognised and the KNX commands for that action are sent, but neither the calculation of the gap between two consecutive presses nor the multiple press count is modified in any way.
7. The release of the button key after the recognition of the long operation does not trigger any action.
8. Once the maximum gap between two consecutive presses (dotted red line) has elapsed, the multiple press count is terminated and, after sending the KNX commands relating to this action, the counter is reset.
9. The detection of a new press leads to the initialisation of the press duration time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
10. If the button key is released before the end of the long operation minimum time, a new short press is recognised; the count of the gap between two consecutive presses is begun, and the multiple press count is increased.
11. The detection of a new press leads to the initialisation of the press duration time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
12. Once the maximum gap between two consecutive presses (dotted red line) has elapsed, the multiple press count is terminated and, after sending the KNX commands relating to this action, the counter is reset.
13. If the pressing continues for a time greater than the minimum duration of a long operation, a long press is recognised and the KNX commands for that action are sent, but neither the calculation of the gap between two consecutive presses nor the multiple press count is modified in any way.
14. The release of the button key after the recognition of the long operation does not trigger any action.

14.2 Sends objects

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

a) the device waits for the gap between two consecutive presses to exceed the maximum value, therefore interrupting the multiple press count and sending the commands associated with the number of presses detected;
b) every time the multiple press count is increased, the device sends the telegrams associated with the number of presses detected

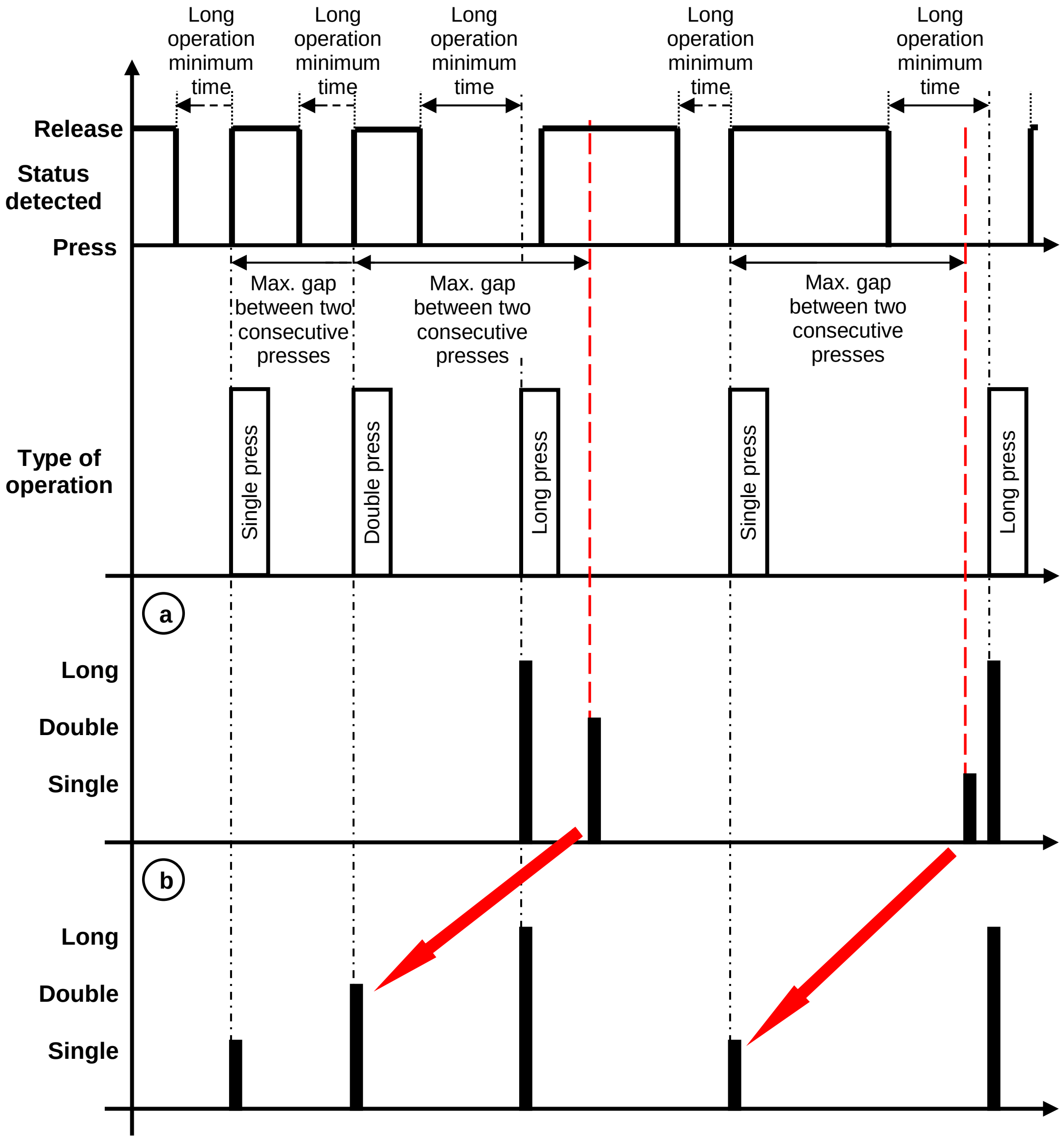
The commands associated with a "long press" are always sent as soon as the long press is detected.

The **"Sends objects"** parameter defines the conditions for sending objects associated with multiple presses. The values that can be set are:

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

If **only at the end of touches counting** is selected, the device behaves as explained in point "a". If **at every detected press** is selected, the device behaves as explained in point "b".

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



Long operation minimum time
Long operation minimum time
Long operation minimum time
Long operation minimum time
Long operation minimum time
Release
Status detected
Press
Max. gap between two consecutive presses
Max. gap between two consecutive presses
Max. gap between two consecutive presses
Type of operation
Single press
Double press
Long press
Single press
Long press
a
Long
Double
Single
b
Long
Double
Single

The chart resumes the situation shown previously, introducing the long press and its effect on counters and timers. The two lower sections show the commands sent on the KNX BUS if sending is **only at the end of touches counting** (case "a") or **at every detected press** (case "b"). The main difference between the two cases is that, in case "b", the relative telegrams are sent every time a multiple press is counted, whereas in case "a" it's necessary to wait until the time between two consecutive presses exceeds the maximum value defined for ending the count and sending the telegrams associated with the value actually reached.

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

14.3 Single press detection

The **"Single press detection"** parameter enables the recognition of a single press, and displays the **Channel x - Single press** menu for enabling and configuring the commands that will be sent following the recognition of the single press. The values that can be set are:

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

If **enable** is selected, the **Channel x - Single press** menu is displayed (see "Channel x - Single press" menu).

14.4 Double press detection

The **"Double press detection"** parameter enables the recognition of a double press, and displays the **Channel x - Double press** menu for enabling and configuring the commands that will be sent following the recognition of the double press. The values that can be set are:

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

If **enable** is selected, the **Channel x - Double press** menu is displayed (see "Channel x - Double press" menu).

14.5 Triple press detection

The **"Triple press detection"** parameter enables the recognition of a triple press, and displays the **Channel x - Triple press** menu for enabling and configuring the commands that will be sent following the recognition of the triple press. The values that can be set are:

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

If **enable** is selected, the **Channel x - Triple press** menu is displayed (see "Channel x - Triple press" menu).

14.6 Quadruple press detection

The **"Quadruple press detection"** parameter enables the recognition of a quadruple press, and displays the **Channel x - Quadruple press** menu for enabling and configuring the commands that will be sent following the recognition of the quadruple press. The values that can be set are:

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

If **enable** is selected, the **Channel x - Quadruple press** menu is displayed (see "Channel x - Quadruple press" menu).

14.7 Long press detection

The **"Long press detection"** parameter enables the recognition of a long press, and displays the **Channel x – Long press** menu for enabling and configuring the commands that will be sent following the recognition of the long press. The values that can be set are:

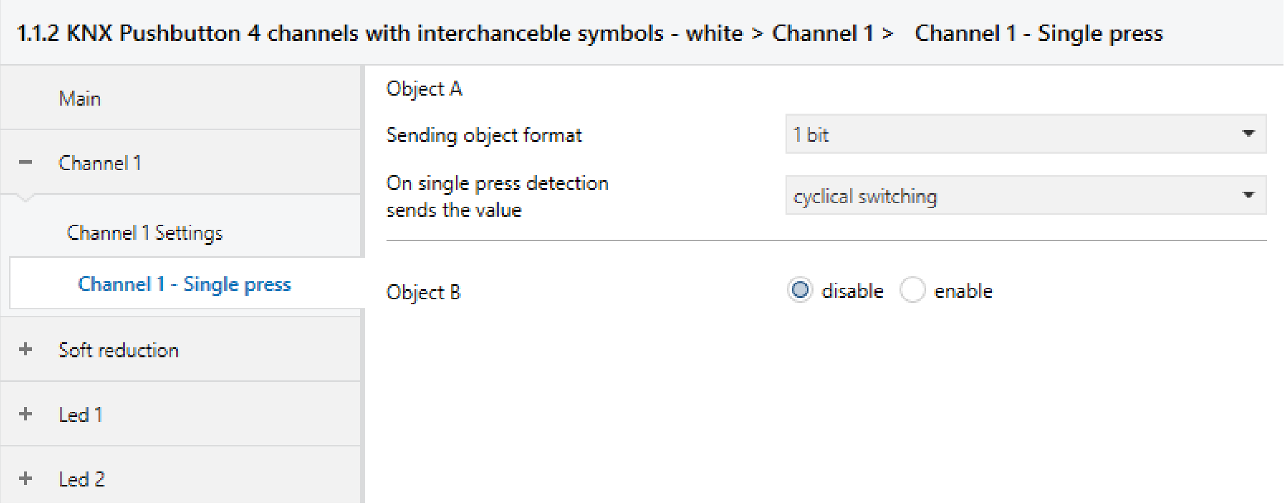
- **disable** **(default value)**
- enable

If **enable** is selected, the **Channel x – Long press** menu is displayed (see "Channel x – Long press" menu).

14.8 Menù Channel x – Single press

This menu, visible if the **"Single press detection"** parameter of the **Channel x** menu is set at **enabled**, configures the communication objects and the relative values (that the device must send on the BUS) associated with the "single press" event.

The structure of the menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 - Single press

Object A

Sending object format: 1 bit

On single press detection sends the value: cyclical switching

Object B: ● disable ○ enable

Fig. 14.2

Upon detection of the single press, 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, but 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 **"Sending object format"** e **"On single press detection sends the value"** parameters will be displayed, grouped in the **Object z** sub-set (where z is the index of the object associated with the channel, between **A** and **D**).

The **"Sending object format"** parameter sets the format and code of the object "z" of input "x" that will be sent by the device. The values that can be set are:

- **1 bit** **(default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC mode
- 2 byte unsigned
- 2 byte signed
- 3 byte RGB color
- 4 byte unsigned
- 4 byte signed

Depending on the value set for this item, the values that can be set for the **"On single press detection sends the value"** parameter will change.
The **"On single press detection sends the value"** parameter sets the command or value (to be sent following the detection of a single press, depending on the sending conditions set) associated with the channel. The values that can be set are:

- If the format of the object to be sent is **1 bit**, the ***Ch.x - Single press 1 bit z object*** (Data Point Type: 1.002 DPT_Bool) communication object is displayed and the values that can be set for the above parameter are:

  - 0
  - 1
  - **ciclical switching (default value)**

  If **ciclical switching** is selected, the command sent by the device (via the ***Ch.x - Single press 1 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the value generated by the most recent of two events - the value received by the BUS on the ***Ch.x - Single press 1 bit z object***, or the last value sent.
  In this configuration, there is no communication object dedicated to receiving the status notification from the BUS so ***Ch.x - Single press 1 bit 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 be sent is **2 bit**, the ***Ch.x - Single press 2 bit z object*** (Data Point Type: 2.001 DPT_Switch_Control communication object is displayed and the values that can be set for the above parameter are:

  - sends activate on (down) forcing
  - sends activate off (up) forcing
  - sends deactivate forcing
  - cyclical switching forcing on/forcing off
  - **cyclical switching forcing on/deactivate forcing (default value)**
  - **cyclical switching forcing off/deactivate forcing**

  If **cyclical switching** is selected, the command sent by the device (via the ***Ch.x - Single press 2 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via the ***Ch.x - Single press 2 bit z object*** object).

- If the format of the object to be sent is **1 byte unsigned**, the ***Ch.x - Single press 1 byte z object***(Data Point Type: 5.010 DPT_Value_1_Ucount communication object is displayed and the values that can be set for the above parameter are:
  - from **0 (default value)** to 255, in steps of 1

- If the format of the object to be sent is **1 byte signed**, the ***Ch.x - Single press 1 byte z object***Data Point Type: 6.010 DPT_Value_1_Count communication object is displayed and the values that can be set for the above parameter are:

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

- If the format of the object to be sent is **1 byte percentage**, the ***Ch.x - Single press 1 byte z object***(Data Point Type: 5.001 DPT_Scaling communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 100, in steps of 1

– If the format of the object to be sent is **1 byte HVAC mode**, the ***Ch.x - Single press 1 byte z object***(Data Point Type: 20.102 DPT_HVACMode communication object is displayed and the values that can be set for the above parameter are:

- sends auto mode
- sends comfort mode
- sends precomfort mode
- sends economy mode
- sends off (building protection) mode
- **cyclical switching (thermostat)** **(default value)**
- cyclical switching (chronothermostat)

If **cyclical switching (thermostat)** is selected, every time the associated event (single press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* … If **cyclical switching (chronothermostat)** is selected, every time the associated event (single press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* …

– If the format of the object to be sent is **2 byte unsigned**, the ***Ch.x - Single press 2 byte z object***Data Point Type: 7.001 DPT_Value_2_Ucount communication object is displayed and the values that can be set for the above parameter are:

- from **0 (default value)** to 65535, in steps of 1

– If the format of the object to be sent is **2 byte signed**, the ***Ch.x - Single press 2 byte z object*** (Data Point Type: 8.001 DPT_Value_2_Count communication object is displayed and the values that can be set for the above parameter are:

- from -32768 to +32767, in steps of 1 **(default value 0)**

– If the format of the object to be sent is **3 byte RGB color**, the **"On single press detection sends value"** parameter is a dummy one and is used to select the colour to be sent, whereas the effective value downloaded in the memory will depend on the three parameters that represent the colour components (see below). In addition, the ***Ch.x - Single press 3 byte z object*** (Data Point Type: 232.600 DPT_Colour_RGB) communication object is displayed and the values that can be set for the above parameter are:

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

If **customize** is selected, the **"RED component value (0 .. 255)"**, **"GREEN component value (0 .. 255)"** and "**"BLUE component value (0 .. 255)"** parameters are displayed. 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, in steps of 1

– If the format of the object to be sent is **4 byte unsigned**, the ***Ch.x - Single press 4 byte z object*** (Data Point Type: 12.001 DPT_Value_4_Ucount communication object is displayed and the values that can be set for the above parameter are:

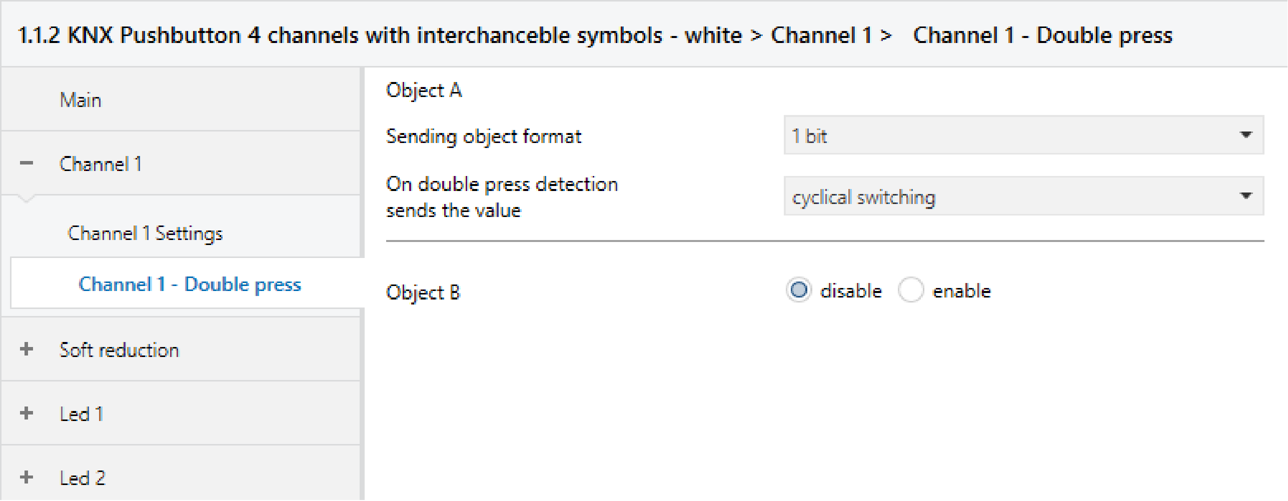
- from **0 (default value)** to 4294967295, in steps of 1

- If the format of the object to be sent is **4 byte signed**, the ***Ch.x - Single press 4 byte z object***(Data Point Type: 13.001 DPT_Value_4_Count) communication object is displayed and the values that can be set for the above parameter are:
  - from -2147483648 to 2147483647, in steps of 1 **(default value 0)**

14.9 Channel x – Double press menu

This menu, visible if the **"Double press detection"** parameter of the **Channel x** menu is set at **enabled**, configures the communication objects and the relative values (that the device must send on the BUS) associated with the "double press" event.

The structure of the menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 - Double press

Object A

Sending object format: 1 bit

On double press detection sends the value: cyclical switching

Object B: ● disable ○ enable

Fig. 14.3

Upon detection of the double press, 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, but 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 **"Sending object format"** and **"On double press detection sends the value"** parameters will be displayed, grouped in the **Object z** sub-set (where z is the index of the object associated with the channel, between **A** and **D**).

The **"Sending object format"** parameter sets the format and code of the object "z" of input "x" that will be sent by the device. The values that can be set are:

- **1 bit** **(default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC mode
- 2 byte unsigned
- 2 byte signed
- 3 byte RGB color
- 4 byte unsigned

- 4 byte signed

Depending on the value set for this item, the values that can be set for the **"On double press detection sends the value"** parameter will change.
The **"On double press detection sends the value"** parameter sets the command or value (to be sent following the detection of a double press, depending on the sending conditions set) associated with the channel. The values that can be set are:

– If the format of the object to be sent is **1 bit**, the ***Ch.x - Double press 1 bit z object***(Data Point Type: 1.002 DPT_Bool) communication object is displayed and the values that can be set for the above parameter are:

  - 0
  - 1
  - **ciclical switching (default value)**

  If **ciclical switching** is selected, the command sent by the device (via the ***Ch.x - Double press 1 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the value generated by the most recent of two events - the value received by the BUS on the ***Ch.x - Double press 1 bit z object*** object, or the last value sent.
  In this configuration, there is no communication object dedicated to receiving the status notification from the BUS so ***Ch.x - Double press 1 bit 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 be sent is **2 bit**, the ***Ch.x - Double press 2 bit z object***(Data Point Type: 2.001 DPT_Switch_Control communication object is displayed and the values that can be set for the above parameter are:

  - sends activate on (down) forcing
  - sends activate off (up) forcing
  - sends deactivate forcing
  - cyclical switching forcing on/forcing off
  - **cyclical switching forcing on/deactivate forcing (default value)**
  - cyclical switching forcing off/deactivate forcing

  If **ciclical switching** is selected, the command sent by the device (via the ***Ch.x - Double press 2 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via the ***Ch.x - Double press 2 bit z object*** object).

– If the format of the object to be sent is **1 byte unsigned**, the ***Ch.x - Double press 1 byte z object*** (Data Point Type: 5.010 DPT_Value_1_Ucount communication object is displayed and the values that can be set for the above parameter are:

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

– If the format of the object to be sent is **1 byte signed**, the ***Ch.x - Double press 1 byte z object*** (Data Point Type: 6.010 DPT_Value_1_Count) communication object is displayed and the values that can be set for the above parameter are:

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

– If the format of the object to be sent is **1 byte percentage**, the ***Ch.x - Double press 1 byte z object***(Data Point Type: 5.001 DPT_Scaling communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 100, in steps of 1

- If the format of the object to be sent is **default value**, the ***Ch.x - Double press 1 byte z object***Data Point Type: 20.102 DPT_HVACMode communication object is displayed and the values that can be set for the above parameter are:

  - sends auto mode
  - sends comfort mode
  - sends precomfort mode
  - sends economy mode
  - sends off (building protection) mode
  - **cyclical switching (thermostat)** **(default value)**
  - cyclical switching (chronothermostat)

  If **cyclical switching (thermostat)** is selected, every time the associated event (double press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* … If **cyclical switching (chronotermostato) ,** is selected, every time the associated event (double press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort*…

- If the format of the object to be sent is **2 byte unsigned**, the ***Ch.x - Double press 2 byte z object*** (Data Point Type: 7.001 DPT_Value_2_Ucount) communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 65535, in steps of 1

- If the format of the object to be sent is **2 byte signed**, the ***Ch.x - Double press 2 byte z object***(Data Point Type: 8.001 DPT_Value_2_Count) communication object is displayed and the values that can be set for the above parameter are:

  - from -32768 to +32767, in steps of 1 **(default value 0)**

- If the format of the object to be sent is **3 byte RGB color**, the **"On double press detection sends the value"** parameter is a dummy one and is used to select the colour to be sent, whereas the effective value downloaded in the memory will depend on the three parameters that represent the colour components (see below). In addition, the ***Ch.x - Double press 3 byte z object*** ((Data Point Type: 232.600 DPT_Colour_RGB) communication object is displayed and the values that can be set for the above parameter are:

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

  If **customize** is selected, the **"RED component value (0 .. 255)"**, **"GREEN component value (0 .. 255)"** and "**"BLUE component value (0 .. 255)"** parameters are displayed. 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, in steps of 1

- If the format of the object to be sent is **4 byte unsigned**, the ***Ch.x - Double press 4 byte z object*** (Data Point Type: 12.001 DPT_Value_4_Ucount) communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 4294967295, in steps of 1

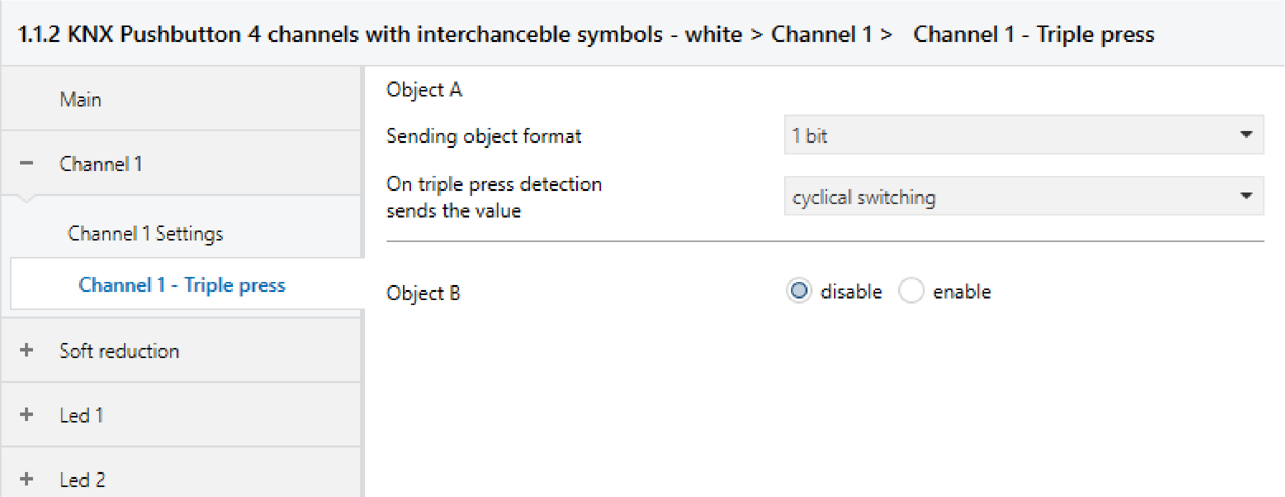
– If the format of the object to be sent is **4 byte signed**, the ***Ch.x - Double press 4 byte z object***(Data Point Type: 13.001 DPT_Value_4_Count) communication object is displayed and the values that can be set for the above parameter are:

  - from -2147483648 to 2147483647, in steps of 1 **(default value 0)**

14.10 Channel x – Triple press menu

This menu, visible if the **"Triple press detection"** parameter of the **Channel x** menu is set at **enable**, configures the communication objects and the relative values (that the device must send on the BUS) associated with the "triple press" event.

The structure of the menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 - Triple press

Object A

Sending object format: 1 bit

On triple press detection sends the value: cyclical switching

Object B: ● disable ○ enable

Fig. 14.4

Upon detection of the triple press, 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, but 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

If **enable** is selected, the **"Sending object format"** and **"On triple press detection sends the value"** parameters will be displayed, grouped in the **Object z** sub-set (where z is the index of the object associated with the input, between **A** and **D**).

The **"Sending object format"** parameter sets the format and code of the object "z" of input "x" that will be sent by the device. The values that can be set are:

- **1 bit** **(default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC mode
- 2 byte unsigned
- 2 byte signed
- 3 byte RGB color
- 4 byte unsigned
- 4 byte signed

Depending on the value set for this item, the values that can be set for the **“On triple press detection sends the value”** parameter will change.
The **“On triple press detection sends the value”** parameter sets the command or value (to be sent following the detection of a triple press, depending on the sending conditions set) associated with the channel. The values that can be set are:

- If the format of the object to be sent is **1 bit**, the ***Ch.x - Triple press 1 bit z object***(Data Point Type: 1.002 DPT_Bool) communication object is displayed and the values that can be set for the above parameter are:

  - 0
  - 1
  - **ciclical switching (default value)**

  If **ciclical switching** is selected, the command sent by the device (via the ***Ch.x - Triple press 1 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the value generated by the most recent of two events - the value received from the BUS on the ***Ch.x - Triple press 1 bit z object*** object, or the last value sent.
  In this configuration, there is no communication object dedicated to receiving the status notification from the BUS so ***Ch.x - Triple press 1 bit 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 be sent is **2 bit**, the ***Ch.x - Triple press 2 bit z object***(Data Point Type: 2.001 DPT_Switch_Control) communication object is displayed and the values that can be set for the above parameter are:

  - sends activate on (down) forcing
  - sends activate off (up) forcing
  - sends deactivate forcing
  - cyclical switching forcing on/forcing off
  - **cyclical switching forcing on/deactivate forcing (default value)**
  - cyclical switching forcing off/deactivate forcing

  If **cyclical switching** is selected, the command sent by the device (via the ***Ch.x - Triple press 2 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via the ***Ch.x - Triple press 2 bit z object*** object).

- If the format of the object to be sent is **1 byte unsigned**, the ***Ch.x - Triple press 1 byte z object*** (Data Point Type: 5.010 DPT_Value_1_Ucount) communication object is displayed and the values that can be set for the above parameter are:

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

- If the format of the object to be sent is **1 byte signed**, the ***Ch.x - Triple press 1 byte z object*** (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, in steps of 1 **(default value 0)**

- If the format of the object to be sent is **1 byte percentage**, the ***Ch.x - Triple press 1 byte z object*** (Data Point Type: 5.001 DPT_Scaling communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 100, in steps of 1

- If the format of the object to be sent is **1 byte HVAC mode**, the ***Ch.x - Triple press 1 byte z object*** (Data Point Type: 20.102 DPT_HVACMode communication object is displayed and the values that can be set for the above parameter are:

  - sends auto mode
  - sends comfort mode
  - sends precomfort mode
  - sends economy mode
  - sends off (building protection) mode
  - **cyclical switching (thermostat)** **(default value)**
  - cyclical switching (chronothermostat)

  If **cyclical switching (thermostat)** is selected, every time the associated event (triple press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* … If **cyclical switching (chronothermostat)** is selected, every time the associated event (triple press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* …

- If the format of the object to be sent is **2 byte unsigned**, the ***Ch.x - Triple press 2 byte z object*** (Data Point Type: 7.001 DPT_Value_2_Ucount) communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 65535, in steps of 1

- If the format of the object to be sent is **2 byte signed**, the ***Ch.x - Triple press 2 byte z object*** (Data Point Type: 8.001 DPT_Value_2_Count) communication object is displayed and the values that can be set for the above parameter are:

  - from -32768 to +32767, in steps of 1 **(default value 0)**

- If the format of the object to be sent is **3 byte RGB color**, the **"On triple press detection sends the value"** parameter is a dummy one and is used to select the colour to be sent, whereas the effective value downloaded in the memory will depend on the three parameters that represent the colour components (see below). In addition, the ***Ch.x - Triple press 3 byte z object*** ((Data Point Type: 232.600 DPT_Colour_RGB) communication object is displayed and the values that can be set for the above parameter are:

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

  If **customize** is selected, the **"RED component value (0 .. 255)"**, **"GREEN component value (0 .. 255)"** and "**"BLUE component value (0 .. 255)"** parameters are displayed. 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, in steps of 1

- If the format of the object to be sent is **4 byte unsigned**, the ***Ch.x - Triple press 4 byte z object*** (Data Point Type: 12.001 DPT_Value_4_Ucount) communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 4294967295, in steps of 1

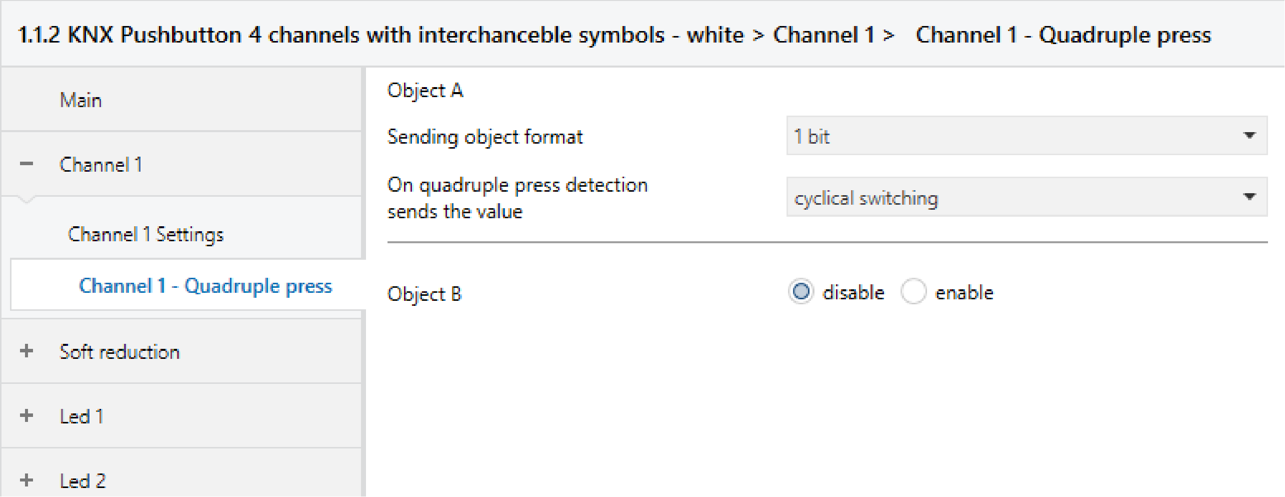
– If the format of the object to be sent is **4 byte signed**, the ***Ch.x - Triple press 4 byte z object*** (Data Point Type: 13.001 DPT_Value_4_Count) communication object is displayed and the values that can be set for the above parameter are:

  - from -2147483648 to 2147483647, in steps of 1 **(default value 0)**

14.11 Channel x – Quadruple press menu

This menu, visible if the **"Quadruple press detection"** parameter of the **Channel x** menu is set at **enable**, configures the communication objects and the relative values (that the device must send on the BUS) associated with the "quadruple press" event.

The structure of the menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 - Quadruple press

Object A

Sending object format: 1 bit

On quadruple press detection sends the value: cyclical switching

Object B: ● disable ○ enable

Fig. 14.5

Upon detection of the quadruple press, 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, but 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

If **enable** is selected, the **"Sending object format"** and **"On Quadruple press detection sends the value"** parameters will be displayed, grouped in the **Object z** sub-set (where z is the index of the object associated with the input, between **A** and **D**).

The **"Sending object format"** parameter sets the format and code of the object "z" of input "x" that will be sent by the device. The values that can be set are:

- **1 bit** **(default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC mode
- 2 byte unsigned
- 2 byte signed
- 3 byte RGB color
- 4 byte unsigned
- 4 byte signed

Depending on the value set for this item, the values that can be set for the **"On Quadruple press detection sends the value"** parameter will change.
The **"On Quadruple press detection sends the value"** parameter sets the command or value (to be sent following the detection of a quadruple press, depending on the sending conditions set) associated with the input. The values that can be set are:

– If the format of the object to be sent is **1 bit**, the ***Ch.x - Quadruple press 1 bit z object***(Data Point Type: 1.002 DPT_Bool communication object is displayed and the values that can be set for the above parameter are:

- 0
- 1
- **ciclical switching (default value)**

If **ciclical switching** is selected, the command sent by the device (via the ***Ch.x - Quadruple press 1 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the value generated by the most recent of two events - the value received from the BUS on the ***Ch.x - Quadruple press 1 bit z object*** object, or the last value sent.
In this configuration, there is no communication object dedicated to receiving the status notification from the BUS so ***Ch.x - Quadruple press 1 bit 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 be sent is **2 bit**, the ***Ch.x - Quadruple press 2 bit z object*** (Data Point Type: 2.001 DPT_Switch_Control) communication object is displayed and the values that can be set for the above parameter are:

- sends activate on (down) forcing
- sends activate off (up) forcing
- sends deactivate forcing
- cyclical switching forcing on/forcing off
- **cyclical switching forcing on/deactivate forcing (default value)**
- cyclical switching forcing off/deactivate forcing

If **cyclical switching** is selected, the command sent by the device (via the ***Ch.x - Quadruple press 2 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via the ***Ch.x - Quadruple press 2 bit z object*** object).

– If the format of the object to be sent is **1 byte unsigned**, the ***Ch.x - Quadruple press 1 byte z object*** (Data Point Type: 5.010 DPT_Value_1_Ucount) communication object is displayed and the values that can be set for the above parameter are:

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

– If the format of the object to be sent is **1 byte signed**, the ***Ch.x - Quadruple press 1 byte z object*** (Data Point Type: 6.010 DPT_Value_1_Count) communication object is displayed and the values that can be set for the above parameter are:

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

– If the format of the object to be sent is **1 byte percentage**, the ***Ch.x - Quadruple press 1 byte z object*** (Data Point Type: 5.001 DPT_Scaling) communication object is displayed and the values that can be set for the above parameter are:

- from **0 (default value)** to 100, in steps of 1

- If the format of the object to be sent is **1 byte HVAC mode**, the ***Ch.x - Quadruple press 1 byte z object***(Data Point Type: 20.102 DPT_HVACMode) communication object is displayed and the values that can be set for the above parameter are:

  - sends auto mode
  - sends comfort mode
  - sends precomfort mode
  - sends economy mode
  - sends off (building protection) mode
  - **cyclical switching (thermostat)** **(default value)**
  - cyclical switching (chronothermostat)

  If **cyclical switching (thermostat)** is selected, every time the associated event (quadruple press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* …. If **cyclical switching (chronothermostat)** is selected, every time the associated event (quadruple press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* ….

- If the format of the object to be sent is **2 byte unsigned**, the ***Ch.x - Quadruple press 2 byte z object***(Data Point Type: 7.001 DPT_Value_2_Ucount) communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 65535, in steps of 1

- If the format of the object to be sent is **2 byte signed**, the ***Ch.x - Quadruple press 2 byte z object*** (Data Point Type: 8.001 DPT_Value_2_Count) communication object is displayed and the values that can be set for the above parameter are:

  - from -32768 to +32767, in steps of 1 **(default value 0)**

- If the format of the object to be sent is **3 byte RGB color**, the **"On Quadruple press detection sends the value"** parameter is a dummy one and is used to select the colour to be sent, whereas the effective value downloaded in the memory will depend on the three parameters that represent the colour components (see below). In addition, the ***Ch.x - Quadruple press 3 byte z object*** ((Data Point Type: 232.600 DPT_Colour_RGB) communication object is displayed and the values that can be set for the above parameter are:

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

  If **customize** is selected, the **"RED component value (0 .. 255)"**, **"GREEN component value (0 .. 255)"** and "**"BLUE component value (0 .. 255)"** parameters are displayed. 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, in steps of 1

- If the format of the object to be sent is **4 byte unsigned**, the ***Ch.x - Quadruple press 4 byte z object*** (Data Point Type: 12.001 DPT_Value_4_Ucount) communication object is displayed and the values that can be set for the above parameter are:

  - from **0 (default value)** to 4294967295, in steps of 1

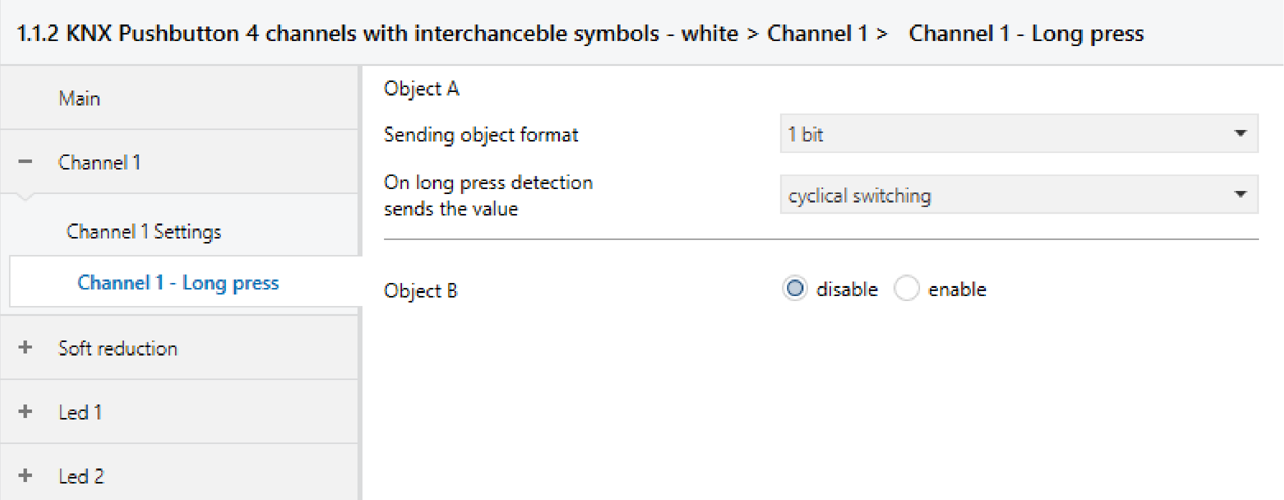
– If the format of the object to be sent is **4 byte signed**, the ***Ch.x - Quadruple press 4 byte z object*** (Data Point Type: 13.001 DPT_Value_4_Count) communication object is displayed and the values that can be set for the above parameter are:

  - from -2147483648 to 2147483647, in steps of 1 **(default value 0)**

14.12 Menù Channel x – Long Press

This menu, visible if the **"Long press detection"** parameter of the **Cannel x** menu is set at **enabled**, configures the communication objects and the relative values (that the device must send on the BUS) associated with the "long press" event.

The structure of the menu is as follows:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Channel 1 > Channel 1 - Long press

Object A

Sending object format: 1 bit

On long press detection sends the value: cyclical switching

Object B: ● disable ○ enable

Fig. 14.6

Upon detection of the long press, 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, but 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 **"Sending object format"** " and " **"On long press detection sends value"** parameters will be displayed, grouped in the **Object z** sub-set (where z is the index of the object associated with the input, between **A** and **D**).

The **"Sending object format"** parameter sets the format and code of the object "z" of input "x" that will be sent by the device. The values that can be set are:

- **1 bit** **(default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC mode
- 2 byte unsigned
- 2 byte signed
- 3 byte RGB color

- 4 byte unsigned
- 4 byte signed

Depending on the value set for this item, the values that can be set for the **“On long press detection sends value”** parameter will change.
The **“On long press detection sends value”** parameter sets the command or value (to be sent following the detection of a long press, depending on the sending conditions set) associated with the channel. The values that can be set are:

- If the format of the object to be sent is **1 bit**, the ***Ch.x - Long press 1 bit z object*** (Data Point Type: 1.002 DPT_Bool) communication object is displayed and the values that can be set for the above parameter are:

  - 0
  - 1
  - **ciclical switching (default value)**

  If **ciclical switching** is selected, the command sent by the device (via the ***Ch.x - Long press 1 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the value generated by the most recent of two events - the value received from the BUS on the ***Ch.x - Long press 1 bit z object*** object, or the last value sent.
  In this configuration, there is no communication object dedicated to receiving the status notification from the BUS so ***Ch.x - Long press 1 bit 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 be sent is **2 bit**, the ***Ch.x - Long press 2 bit z object*** (Data Point Type: 2.001 DPT_Switch_Control) communication object is displayed and the values that can be set for the above parameter are:

  - sends activate on (down) forcing
  - sends activate off (up) forcing
  - sends deactivate forcing
  - cyclical switching forcing on/forcing off
  - **cyclical switching forcing on/deactivate forcing (default value)**
  - cyclical switching forcing off/deactivate forcing

  If **ciclical switching** is selected, the command sent by the device (via the ***Ch.x - Long press 2 bit z object*** object) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via the ***Ch.x - Long press 2 bit z object*** object).

- If the format of the object to be sent is **1 byte unsigned**, the ***Ch.x - Long press 1 byte z object*** (Data Point Type: 5.010 DPT_Value_1_Ucount) communication object is displayed and the values that can be set for the above parameter are:

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

- If the format of the object to be sent is **1 byte signed**, the ***Ch.x - Long press 1 byte z object*** (Data Point Type: 6.010 DPT_Value_1_Count) communication object is displayed and the values that can be set for the above parameter are:

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

- If the format of the object to be sent is **1 byte percentage**, the ***Ch.x - Long press 1 byte z object*** (Data Point Type: 5.001 DPT_Scaling) communication object is displayed and the values that can be set for the above parameter are:

- from **0 (default value)** to 100, in steps of 1

– If the format of the object to be sent is **1 byte HVAC mode**, the ***Ch.x - Long press 1 byte z object*** (Data Point Type: 20.102 DPT_HVACMode communication object is displayed and the values that can be set for the above parameter are:

- sends auto mode
- sends comfort mode
- sends precomfort mode
- sends economy mode
- sends off (building protection) mode
- **cyclical switching (thermostat)** **(default value)**
- cyclical switching (chronothermostat)

If **cyclical switching (thermostat)** is selected, every time the associated event (long press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* …. If **cyclical switching (chronothermostat)** is selected, every time the associated event (long press) is detected, the device sends a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* ….

– If the format of the object to be sent is **2 byte unsigned**, the ***Ch.x - Long press 2 byte z object*** (Data Point Type: 7.001 DPT_Value_2_Ucount) communication object is displayed and the values that can be set for the above parameter are:

- from **0 (default value)** to 65535, in steps of 1

– If the format of the object to be sent is **2 byte signed**, the ***Ch.x - Long press 2 byte z object*** (Data Point Type: 8.001 DPT_Value_2_Count) communication object is displayed and the values that can be set for the above parameter are:

- from -32768 to +32767, in steps of 1 **(default value 0)**

– If the format of the object to be sent is **3 byte RGB color**, the **"On Long press detection sends the value"** parameter is a dummy one and is used to select the colour to be sent, whereas the effective value downloaded in the memory will depend on the three parameters that represent the colour components (see below). In addition, the ***Ch.x - Long press 3 byte z object*** (Data Point Type: 232.600 DPT_Colour_RGB) communication object is displayed and the values that can be set for the above parameter are:

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

– If **customize** is selected, the **"RED component value (0 .. 255)"**, **"GREEN component value (0 .. 255)"** and "**"BLUE component value (0 .. 255)"** parameters are displayed. 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, in steps of 1

– If the format of the object to be sent is **4 byte unsigned**, the ***Ch.x - Long press 4 byte z object*** (Data Point Type: 12.001 DPT_Value_4_Ucount) communication object is displayed and the values that can be set for the above parameter are:

- from **0 (default value)** to 4294967295, in steps of 1

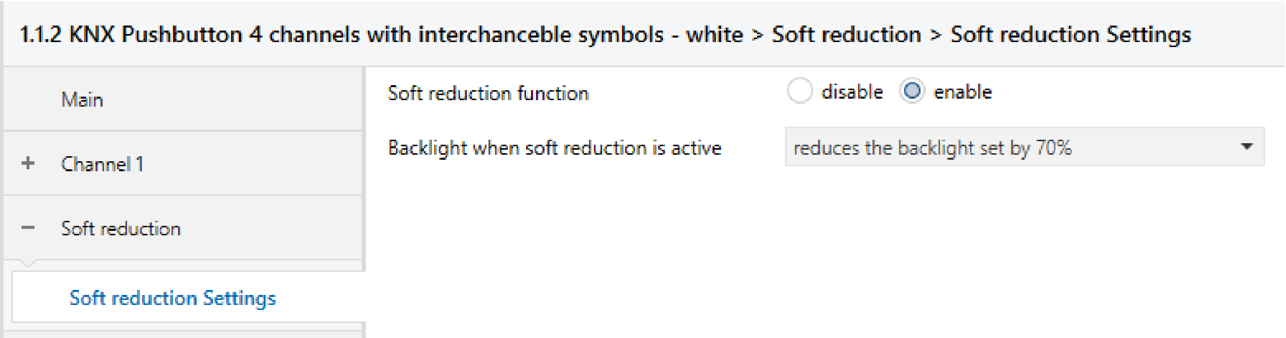
- If the format of the object to be sent is **4 byte signed**, the ***Ch.x - Long press 4 byte z object*** (Data Point Type: 13.001 DPT_Value_4_Count) communication object is displayed and the values that can be set for the above parameter are:

    - from -2147483648 to 2147483647, in steps of 1 **(default value 0)**

15 *“Soft reduction”* menu

Given its function, the device may be installed in places where it's necessary to 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 light signalling settings to minimise any possible 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 15.1 shows the basic structure of the menu:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Soft reduction > Soft reduction Settings

Soft reduction function ○ disable ● enable

Backlight when soft reduction is active: reduces the backlight set by 70%

Fig. 15.1

15.1 Soft reduction

The **“Soft reduction function”** parameter enables this function and displays the configuration parameters. The values that can be set are:

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

If **enable** is selected, the function configuration parameters and the ***Soft reduction*** (Data Point Type: 1.001 DPT_Switch) communication object are displayed.
Every time the BUS voltage is reset, a status read command (read request) must be sent on this object in order to update the device about the function activation status. When the BUS voltage is restored, 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.

15.2 Backlight when soft reduction is active

The **“Backlight when soft reduction is active”** parameter defines the backlighting intensity level of all 4 LEDs on the device when the soft reduction function is activated (e.g. via the timing of a remote device connected to the KNX system). The values that can be set are:

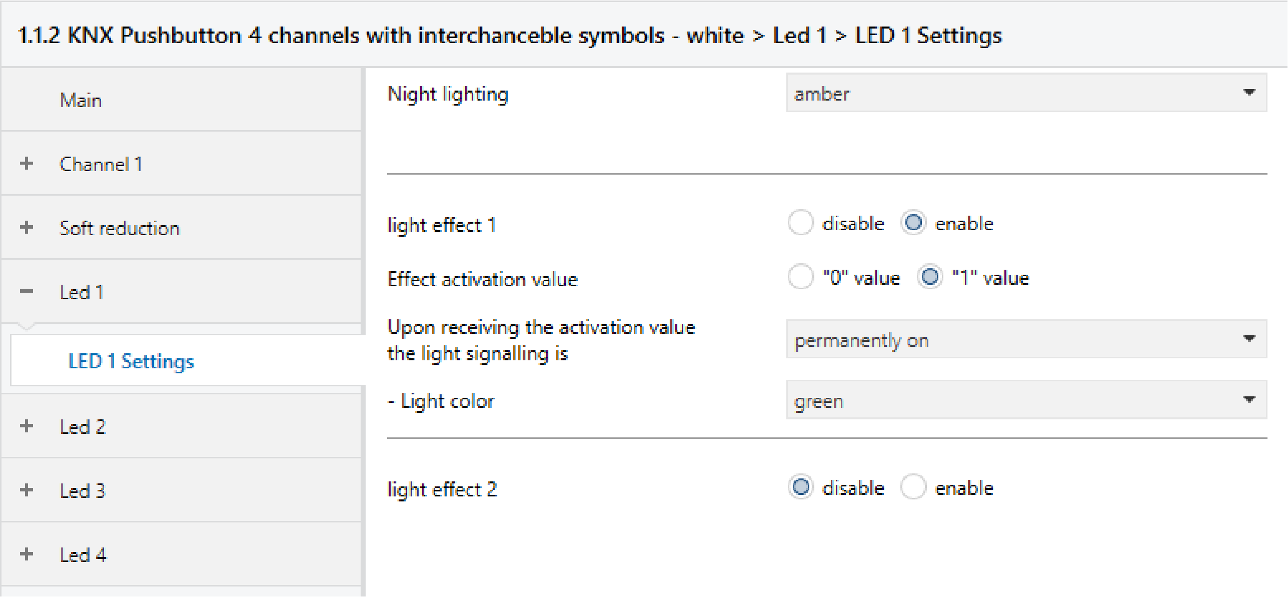
- Riduces the backlight set by 10%
- Riduces the backlight set by 20%
- Riduces the backlight set by 30%
- Riduces the backlight set by 40%
- Riduces the backlight set by 50%
- Riduces the backlight set by 60%
- **Riduces the backlight set by 70%** **(default value)**
- Riduces the backlight set by 80%
- Riduces the backlight set by 90%
- Riduces the backlight set by 95%
- Switches off the backlight

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

16 “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-time localisation function, or it can be autonomously managed via the relative communication objects. The communication objects enabled by this function are ***Led x – Effect 1***, ***Led x – Effect 2***, ***Led x – Effect 3***, ***Led x – Effect 4*** and e ***Led x – Effect 5***..

Figure 16.1 shows the basic structure of the menu:



1.1.2 KNX Pushbutton 4 channels with interchanceble symbols - white > Led 1 > LED 1 Settings

Main
+ Channel 1
+ Soft reduction
− Led 1
LED 1 Settings
+ Led 2
+ Led 3
+ Led 4

Night lighting: amber

light effect 1: ○ disable ● enable

Effect activation value: ○ "0" value ● "1" value

Upon receiving the activation value the light signalling is: permanently on

- Light color: green

light effect 2: ● disable ○ enable

Fig. 17.1

16.1 Night lighting

The **“Night lighting”** parameter enables and defines the colour of the night-time localisation associated with channel x. The values that can be set are:

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

If **customize** is selected, the **“RED component value (0 .. 255)”**, **“GREEN component value (0 .. 255)”** and **“BLUE component value (0 .. 255)”** parameters, and the ***Led x – Customize night signalling*** (Data Point Type: 232.600 DPT_Colour_RGB) communication object, are displayed. The combination of the three colour components determines the colour associated with night-time localisation. The values that can be set are:

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

The ***Led x – Customize night signalling*** communication object allows to receive, via the BUS, the colour you want to associate with night-time localisation. 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-time localisation of LED x is active.
The same object - ***Led x – Customize night signalling*** - is also used to send the signalling of the colour currently associated with night-time localisation. 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 signalling*** 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.

16.2 Light effect X

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

The **“Light effect 1”**, **“Light effect 2”**, **“Light effect 3”**, **“Light effect 4”** e **“Light effect 5”** parameters enable various communication objects for activating 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 displayed, along with the associated BUS communication object - ***Led x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x - Effect 4*** e ***Led x - Effect 5*** (Data Point Type: 1.002 DPT_Bool).
Every time the BUS voltage is reset, a status read command (read request) must be sent 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*** e ***Led x - Effect 5*** - will activate the set brightness effect. The values that can be set for this parameter are:

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

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

The ***Led x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x - Effect 4*** e ***Led x - Effect 5*** communication objects can be used to activate/deactivate the associated brightness effect, thanks to 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-time localisation 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 when an activation command is received for the brightness effect that the parameter refers to. The values that can be set are:

- off
- **permanently on (default value)**

- blinking 1 Hz
- blinking 2 Hz

If you select any value other than **Off**, the **"Light color"** parameter is displayed and can be used to personalise the colour of the signalling associated with the defined effect. The values that can be set are:

- current color
- white
- yellow
- magenta
- red
- turquoise
- **green** **(default value)**
- blue
- 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 color**, if the LED is disabled when the effect is activated, the effect will not be implemented.
If **customize** is selected, the **"RED component value (0 .. 255)"**, **"GREEN component value (0 .. 255)"** and **"BLUE component value (0 .. 255)"** parameters are displayed, along with the ***Led x - Customize effect 1*** (***Led x - Customize effect 2***, ***Led x - Customize effect 3***, ***Led x - Customize effect 4***, ***Led x - Customize effect 5***, communication object, depending on the effect that the parameter refers to (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, in steps of 1

The ***Led x - Customize effect 1 (2/3/4/5)*** communication object allows to receive, via the BUS, the colour you want to associate with the brightness 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 colour currently associated with the brightness 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, ***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 ***TEST RGB color*** (Data Point Type: 232.600 DPT_Colour_RGB) and ***TEST RGB color brightness*** (Data Point Type: 5.001 DPT_Scaling) communication objects can be used to select the colour and percentage regulation of the signal brightness so that the selected colour can be checked against the brightness levels set for proximity backlighting and standby. The colour test mode is activated when a telegram is received on the ***TEST RGB color*** object; if a telegram is received on the ***TEST RGB color brightness*** object and the test function isn't active, nothing will happen.
When the TEST function is active, all 4 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 ***TEST RGB color*** object, the brightness of the light signalling returns to the 100% value (even if it was 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.

17 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.

17.1 Communication objects with output functions

#				Object name	Object function	Description	Datapoint type
Ch 1	Ch 2	Ch 3	Ch 4				
1	26	51	76	Ch.x - Switch	On/Off	Dimmer switching on/off commands	1.001 DPT_Switch
1	26	51	76	Ch.x – Shutter movement	Up/Down	Moves up/down the shutter	1.008 DPT_UpDown
1	26	51	76	Ch.x - Scene	Execute/Store	Sends learn/execute scene commands	18.001 DPT_SceneControl
1	26	51	76	Ch.x – Sequence A	On/Off	Sends On/Off commands associated with object A of the sequence	1.001 DPT_Switch
1	26	51	76	Ch.x – A object 1 bit value	1/0 value	Sends 1/0 values associated with object A	1.002 DPT_Bool
1	26	51	76	Ch.x – A object 2 bit value	On/Off forced position	Sends priority commands associated with object A	2.001 DPT_Switch_Control
1	26	51	76	Ch.x – A object 1 byte value	Unsigned value	Sends unsigned value (0..255) associated with object A	5.010 DPT_Value_1_Ucount
1	26	51	76	Ch.x – A object 1 byte value	Signed value	Sends signed value (-128..127) associated with object A	6.010 DPT_Value_1_Count
1	26	51	76	Ch.x – A object 1 byte value	% value	Sends percent values (0%..100%) associated with object A	5.001 DPT_Scaling
1	26	51	76	Ch.x – A object 1 byte value	HVAC Mode	Sends HVAC Mode (auto/comfort/precomfort/economy/off) associated with object A	20.102 DPT_HVACMode
1	26	51	76	Ch.x – A object 2 bytes value	Unsigned value	Sends unsigned value (0..65535) associated with object A	7.001 DPT_Value_2_Ucount
1	26	51	76	Ch.x – A object 2 bytes value	Signed value	Sends signed value (-32768..32767) associated with object A	8.001 DPT_Value_2_Count
1	26	51	76	Ch.x – A object 3 bytes value	RGB Color	Sends RGB color components value associated with object A	232.600 DPT_Colour_RGB
1	26	51	76	Ch.x – A object 4 bytes value	Unsigned value	Sends unsigned value (0.. 4294967295) associated with object A	12.001 DPT_Value_4_Ucount
1	26	51	76	Ch.x – A object 4 bytes value	Signed value	Sends signed value (-2147483648..2147483647) associated with object A	13.001 DPT_Value_4_Count
1	26	51	76	Ch.x – A object 14 bytes value	ISO 8859-1 Characters	Sends characters codified with ISO 8859-1 standard associated with object A	16.001 DPT_String_8859_1
1	26	51	76	Ch.x - Single press 1 bit A object	1/0 value	Sends 1/0 values associated with object A of the single press	1.002 DPT_Bool
1	26	51	76	Ch.x - Single press 2 bit A object	On/Off forced position	Sends priority commands associated with object A of the single press	2.001 DPT_Switch_Control
1	26	51	76	Ch.x - Single press 1 byte A object	Unsigned value	Sends unsigned value (0..255) associated with object A of the single press	5.010 DPT_Value_1_Ucount
1	26	51	76	Ch.x - Single press 1 byte A object	Signed value	Sends signed value (-128..127) associated with object A of the single press	6.010 DPT_Value_1_Count
1	26	51	76	Ch.x - Single press 1 byte A object	% value	Sends percent values (0%..100%) associated with object A of the single press	5.001 DPT_Scaling
1	26	51	76	Ch.x - Single press 1 byte A object	HVAC Mode	Sends HVAC Mode (auto/comfort/precomfort/economy/off) associated with object A of the single press	20.102 DPT_HVACMode

1	26	51	76	Ch.x - Single press 2 byte A object	Unsigned value	Sends unsigned value (0..65535) associated with object A of the single press	7.001 DPT_Value_2_Ucount
1	26	51	76	Ch.x - Single press 2 byte A object	Signed value	Sends signed value (-32768 ..32767) associated with object A of the single press	8.001 DPT_Value_2_Count
1	26	51	76	Ch.x - Single press 3 byte A object	RGB Color	Sends RGB color components value associated with object A of the single press	232.600 DPT_Colour_RGB
1	26	51	76	Ch.x - Single press 4 byte A object	Unsigned value	Sends unsigned value (0.. 4294967295) associated with object A of the single press	12.001 DPT_Value_4_Ucount
1	26	51	76	Ch.x - Single press 4 byte A object	Signed value	Sends signed value (-2147483648..2147483647) associated with object A of the single press	13.001 DPT_Value_4_Count
2	27	52	77	Ch.x – Brightness dimming	Increase/ Decrease	Dimmer brightness increasing / decreasing commands	3.007 DPT_Control_Dimming
2	27	52	77	Ch.x – Shutter stop / Louvres control	Stop/Step	Sends stop movement / slat regulation commands	1.007 DPT_Step
2	27	52	77	Ch.x – Sequence B	On/Off	Sends On/Off commands associated with object B of the sequence	1.001 DPT_Switch
2	27	52	77	Ch.x – B object 1 bit value	1/0 value	Sends 1/0 values associated with object B	1.002 DPT_Bool
2	27	52	77	Ch.x - Single press 1 bit B object	1/0 value	Sends 1/0 values associated with object B of the single press	1.002 DPT_Bool
3	28	53	78	Ch.x – Sequence C	On/Off	Sends On/Off commands associated with object C of the sequence	1.001 DPT_Switch
3	28	53	78	Ch.x – C object 1 bit value	1/0 value	Sends 1/0 values associated with object C	1.002 DPT_Bool
3	28	53	78	Ch.x - Single press 1 bit C object	1/0 value	Sends 1/0 values associated with object C of the single press	1.002 DPT_Bool
4	29	54	79	Ch.x – Sequence D	On/Off	Sends On/Off commands associated with object D of the sequence	1.001 DPT_Switch
4	29	54	79	Ch.x – D object 1 bit value	1/0 value	Sends 1/0 values associated with object D	1.002 DPT_Bool
4	29	54	79	Ch.x - Single press 1 bit D object	1/0 value	Sends 1/0 values associated with object D of the single press	1.002 DPT_Bool
5	30	55	80	Ch.x - Double press 1 bit A object	1/0 value	Sends 1/0 values associated with object A of the double press	1.002 DPT_Bool
5	30	55	80	Ch.x - Double press 2 bit A object	On/Off forced position	Sends priority commands associated with object A of the double press	2.001 DPT_Switch_Control
5	30	55	80	Ch.x - Double press 1 byte A object	Unsigned value	Sends unsigned value (0..255) associated with object A of the double press	5.010 DPT_Value_1_Ucount
5	30	55	80	Ch.x - Double press 1 byte A object	Signed value	Sends signed value (-128..127) associated with object A of the double press	6.010 DPT_Value_1_Count
5	30	55	80	Ch.x - Double press 1 byte A object	% value	Sends percent values (0%..100%) associated with object A of the double press	5.001 DPT_Scaling
5	30	55	80	Ch.x - Double press 1 byte A object	HVAC Mode	Sends HVAC Mode (auto/comfort/precomfort/ economy/off) associated with object A of the double press	20.102 DPT_HVACMode
5	30	55	80	Ch.x - Double press 2 byte A object	Unsigned value	Sends unsigned value (0..65535) associated with object A of the double press	7.001 DPT_Value_2_Ucount
5	30	55	80	Ch.x - Double press 2 byte A object	Signed value	Sends signed value (-32768..32767) associated with object A of the double press	8.001 DPT_Value_2_Count
5	30	55	80	Ch.x - Double press 3 byte A object	RGB Color	Sends RGB color components value associated with object A of the double press	232.600 DPT_Colour_RGB

5	30	55	80	Ch.x - Double press 4 byte A object	Unsigned value	Sends unsigned value (0.. 4294967295) associated with object A of the double press	12.001 DPT_Value_4_Ucount
5	30	55	80	Ch.x - Double press 4 byte A object	Signed value	Sends signed value (-2147483648.. 2147483647) associated with object A of the double press	13.001 DPT_Value_4_Count
6	31	56	81	Ch.x - Double press 1 bit B object	1/0 value	Sends 1/0 values associated with object B of the double press	1.002 DPT_Bool
7	32	57	82	Ch.x - Double press 1 bit C object	1/0 value	Sends 1/0 values associated with object C of the double press	1.002 DPT_Bool
8	33	58	83	Ch.x - Double press 1 bit D object	1/0 value	Sends 1/0 values associated with object D of the double press	1.002 DPT_Bool
9	34	59	84	Ch.x - Triple press 1 bit A object	1/0 value	Sends 1/0 values associated with object A of the triple press	1.002 DPT_Bool
9	34	59	84	Ch.x - Triple press 2 bit A object	On/Off forced position	Sends priority commands associated with object A of the triple press	2.001 DPT_Switch_Control
9	34	59	84	Ch.x - Triple press 1 byte A object	Unsigned value	Sends unsigned value (0..255) associated with object A of the triple press	5.010 DPT_Value_1_Ucount
9	34	59	84	Ch.x - Triple press 1 byte A object	Signed value	Sends signed value (-128..127) associated with object A of the triple press	6.010 DPT_Value_1_Count
9	34	59	84	Ch.x - Triple press 1 byte A object	% value	Sends percent values (0%..100%) associated with object A of the triple press	5.001 DPT_Scaling
9	34	59	84	Ch.x - Triple press 1 byte A object	HVAC Mode	Sends HVAC Mode (auto/ comfort/precomfort/economy/off) associated with object A of the triple press	20.102 DPT_HVACMode
9	34	59	84	Ch.x - Triple press 2 byte A object	Unsigned value	Sends unsigned value (0..65535) associated with object A of the triple press	7.001 DPT_Value_2_Ucount
9	34	59	84	Ch.x - Triple press 2 byte A object	Signed value	Sends signed value (-32768..32767) associated with object A of the triple press	8.001 DPT_Value_2_Count
9	34	59	84	Ch.x - Triple press 3 byte A object	RGB Color	Sends RGB color components value associated with object A of the triple press	232.600 DPT_Colour_RGB
9	34	59	84	Ch.x - Triple press 4 byte A object	Unsigned value	Sends unsigned value (0.. 4294967295) associated with object A of the triple press	12.001 DPT_Value_4_Ucount
9	34	59	84	Ch.x - Triple press 4 byte A object	Signed value	Sends signed value (-2147483648.. 2147483647) associated with object A of the triple press	13.001 DPT_Value_4_Count
10	35	60	85	Ch.x - Triple press 1 bit B object	1/0 value	Sends 1/0 values associated with object B of the triple press	1.002 DPT_Bool
11	36	61	86	Ch.x - Triple press 1 bit C object	1/0 value	Sends 1/0 values associated with object C of the triple press	1.002 DPT_Bool
12	37	62	87	Ch.x - Triple press 1 bit D object	1/0 value	Sends 1/0 values associated with object D of the triple press	1.002 DPT_Bool
13	38	63	88	Ch.x - Quadruple press 1 bit A object	1/0 value	Sends 1/0 values associated with object A of the quadruple press	1.002 DPT_Bool
13	38	63	88	Ch.x - Quadruple press 2 bit A object	On/Off forced position	Sends priority commands associated with object A of the quadruple press	2.001 DPT_Switch_Control
13	38	63	88	Ch.x - Quadruple press 1 byte A object	Unsigned value	Sends unsigned value (0..255) associated with object A of the quadruple press	5.010 DPT_Value_1_Ucount
13	38	63	88	Ch.x - Quadruple press 1 byte A object	Signed value	Sends signed value (-128..127) associated with object A of the quadruple press	6.010 DPT_Value_1_Count
13	38	63	88	Ch.x - Quadruple press 1 byte A object	% value	Sends percent values (0%..100%) associated with object A of the quadruple press	5.001 DPT_Scaling

13	38	63	88	Ch.x - Quadruple press 1 byte A object	HVAC Mode	Sends HVAC Mode (auto/ comfort/precomfort/economy/off) associated with object A of the quadruple press	20.102 DPT_HVACMode
13	38	63	88	Ch.x - Quadruple press 2 byte A object	Unsigned value	Sends unsigned value (0..65535) associated with object A of the quadruple press	7.001 DPT_Value_2_Ucount
13	38	63	88	Ch.x - Quadruple press 2 byte A object	Signed value	Sends signed value (-32768..32767) associated with object A of the quadruple press	8.001 DPT_Value_2_Count
13	38	63	88	Ch.x - Quadruple press 3 byte A object	RGB Color	Sends RGB color components value associated with object A of the quadruple press	232.600 DPT_Colour_RGB
13	38	63	88	Ch.x - Quadruple press 4 byte A object	Unsigned value	Sends unsigned value (0.. 4294967295) associated with object A of the quadruple press	12.001 DPT_Value_4_Ucount
13	38	63	88	Ch.x - Quadruple press 4 byte A object	Signed value	Sends signed value (-2147483648.. 2147483647) associated with object A of the quadruple press	13.001 DPT_Value_4_Count
14	39	64	89	Ch.x - Quadruple press 1 bit B object	1/0 value	Sends 1/0 values associated with object B of the quadruple press	1.002 DPT_Bool
15	40	65	90	Ch.x - Quadruple press 1 bit C object	1/0 value	Sends 1/0 values associated with object C of the quadruple press	1.002 DPT_Bool
16	41	66	91	Ch.x - Quadruple press 1 bit D object	1/0 value	Sends 1/0 values associated with object D of the quadruple press	1.002 DPT_Bool
17	42	67	92	Ch.x - Long press 1 bit A object	1/0 value	Sends 1/0 values associated with object A of the long press	1.002 DPT_Bool
17	42	67	92	Ch.x - Long press 2 bit A object	On/Off forced position	Sends priority commands associated with object A of the long press	2.001 DPT_Switch_Control
17	42	67	92	Ch.x - Long press 1 byte A object	Unsigned value	Sends unsigned value (0..255) associated with object A of the long press	5.010 DPT_Value_1_Ucount
17	42	67	92	Ch.x - Long press 1 byte A object	Signed value	Sends signed value (-128..127) associated with object A of the long press	6.010 DPT_Value_1_Count
17	42	67	92	Ch.x - Long press 1 byte A object	% value	Sends percent values (0%..100%) associated with object A of the long press	5.001 DPT_Scaling
17	42	67	92	Ch.x - Long press 1 byte A object	HVAC Mode	Sends HVAC Mode (auto/ comfort/precomfort/economy/off) associated with object A of the long press	20.102 DPT_HVACMode
17	42	67	92	Ch.x - Long press 2 byte A object	Unsigned value	Sends unsigned value (0..65535) associated with object A of the long press	7.001 DPT_Value_2_Ucount
17	42	67	92	Ch.x - Long press 2 byte A object	Signed value	Sends signed value (-32768..32767) associated with object A of the long press	8.001 DPT_Value_2_Count
17	42	67	92	Ch.x - Long press 3 byte A object	RGB Color	Sends RGB color components value associated with object A of the long press	232.600 DPT_Colour_RGB
17	42	67	92	Ch.x - Long press 4 byte A object	Unsigned value	Sends unsigned value (0.. 4294967295) associated with object A of the long press	12.001 DPT_Value_4_Ucount
17	42	67	92	Ch.x - Long press 4 byte A object	Signed value	Sends signed value (-2147483648.. 2147483647) associated with object A of the long press	13.001 DPT_Value_4_Count
18	43	68	93	Ch.x - Long press 1 bit B object	1/0 value	Sends 1/0 values associated with object B of the long press	1.002 DPT_Bool
19	44	69	94	Ch.x - Long press 1 bit C object	1/0 value	Sends 1/0 values associated with object C of the long press	1.002 DPT_Bool
20	45	70	95	Ch.x - Long press 1 bit D object	1/0 value	Sends 1/0 values associated with object D of the long press	1.002 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), C (objects 3/28/53/78) and D (object 4/29/54/79) 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), C (objects 7/32/57/82) and D (object 8/33/58/83) 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), C (objects 11/36/61/86) and D (object 12/37/62/87) 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), C (objects 15/40/65/90) and D (object 16/41/66/91) 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), C (objects 19/44/69/94) and D (object 20/45/70/95) due to space problems, but they are present.

17.2 Communication objects with input functions

#				Object name	Object function	Description	Datapoint type
Ch 1	Ch 2	Ch 3	Ch 4				
0	25	50	75	Ch.x - Block	Switching On/Off	Enables the activation/ deactivation of the block function	1.003 DPT_Enable
2	27	52	77	Ch.x – Scene storing trigger	Store	Receives the sending learn scene message request (trigger)	1.017 DPT_Trigger
21	46	71	96	Ch.x – Dimmer status feedback	On/off status	Receives the dimmer activation status feedbacks	1.001 DPT_Switch
21	46	71	96	Ch.x – A object status feedback	On/off status	Receives status feedback from actuator for cyclical swithcing function of object A	1.001 DPT_Switch
21	46	71	96	Ch.x – Movement feedback	Increase/Decrease	Receives increase/decrease movement status from roller shutter actuator	1.008 DPT_UpDown
22	47	72	97	Ch.x – B object status feedback	On/off status	Receives status feedback from actuator for cyclical swithcing function of object B	1.001 DPT_Switch
23	48	73	98	Ch.x – C object status feedback	On/off status	Receives status feedback from actuator for cyclical swithcing function of object C	1.001 DPT_Switch
24	49	74	99	Ch.x – D object status feedback	On/off status	Receives status feedback from actuator for cyclical swithcing function of object D	1.001 DPT_Switch
100	111	122	133	Led x - Effect 1	1/0 Value	Plays the associated light effect	1.002 DPT_Bool
101	112	123	134	Led x - Effect 2	1/0 Value	Plays the associated light effect	1.002 DPT_Bool
102	113	124	135	Led x - Effect 3	1/0 Value	Plays the associated light effect	1.002 DPT_Bool
103	114	125	136	Led x - Effect 4	1/0 Value	Plays the associated light effect	1.002 DPT_Bool
104	115	126	137	Led x - Effect 5	1/0 Value	Plays the associated light effect	1.002 DPT_Bool
105	116	127	138	Led x – Customize night signalling	Set RGB color	Enables to customize the color associated with night signalling	232.600 DPT_Colour_RGB
106	117	128	139	Led x - Customize effect 1	Set RGB color	Enables to customize the color associated with night signalling	232.600 DPT_Colour_RGB
107	118	129	140	Led x - Customize effect 2	Set RGB color	Enables to customize the color associated with night signalling	232.600 DPT_Colour_RGB
108	119	130	141	Led x - Customize effect 3	Set RGB color	Enables to customize the color associated with night signalling	232.600 DPT_Colour_RGB
109	120	131	142	Led x - Customize effect 4	Set RGB color	Enables to customize the color associated with night signalling	232.600 DPT_Colour_RGB
110	121	132	143	Led x - Customize effect 5	Set RGB color	Enables to customize the color associated with night signalling	232.600 DPT_Colour_RGB
144				Soft reduction	Switching On/Off	Receives function activation/ deactivation commands	1.001 DPT_Switch
145				TEST RGB color	Set custom color	Receives the RGB color components to be tested	232.600 DPT_Colour_RGB
146				TEST RGB color brightness	Set color brightness	Receives the personalized color brightness to be tested	5.001 DPT_Scaling

18 BUS voltage recovery

The start-up of the device following a BUS failure is indicated by the sequential activation of all four 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

SAT

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

24h +39 035 946 260

@ sat@gewiss.com
www.gewiss.com