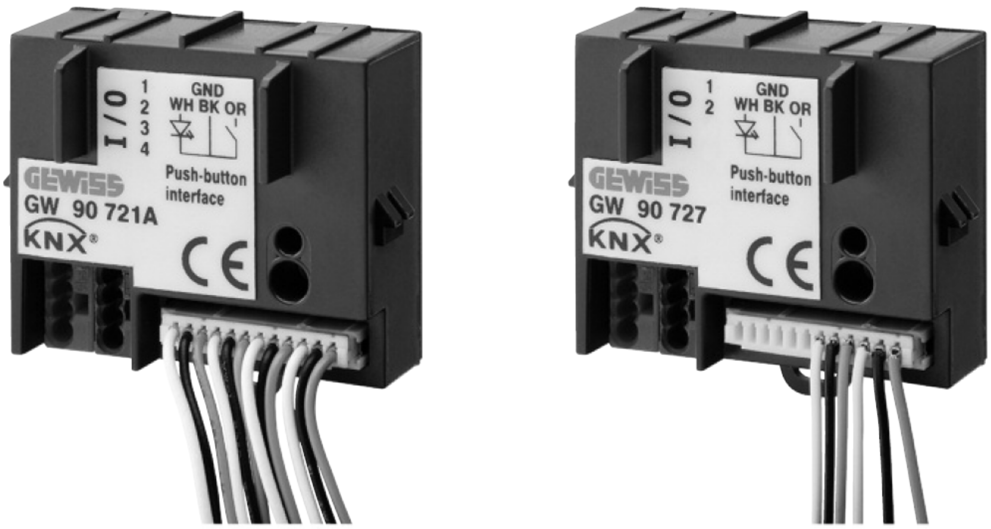
KNX 2-and 4-channel contact interfaces



GW90727 and GW 90 721A

Technical Manual

Contents

1 Introduction

This manual explains the functions of both the GW90727 **“Contact interface 2 channels”** and the GW90721A **“Contact interface 4 channels”**, and how they are set and configured using the ETS configuration software.

The only practical difference between the two devices lies in the fact that the GW90721A interface is able to send measured temperature values as well, if NTC temperature sensors (e.g. GW10800 or GW1x900) are connected to the input channels; given, therefore, that the devices are almost identical in practical terms, this manual speaks only of the GW90721A.

2 Application

The GW 90721A 4-input interface is an input device that can be inserted in flush-mounting boxes. It has 4 inputs to which potential-free contacts (push-buttons, one-way switches, sensors, etc.) can be connected, and a contact that can power a supplementary LED (of 3.3 Volt and with a maximum current of 1 mA). The interrogation voltage needed to establish the closure/opening of the contact is supplied directly by the device and is equal to 3.3 Volt. There are certain device functions which, to be managed, require the use of two linked channels (for example to command a roller shutter with an up button and a down button), and others for which the use of a single channel is sufficient.
The device can perform the following functions:

- ON / OFF commands for loads
- forcing management
- dimmer management (single or double push-button)
- curtain / roller shutter management (single or double push-button)
- scene management
- pulse counter
- sequences of commands with objects of different sizes (1,2 bits and 1,2,3,4,14 bytes)
- commands with multiple pressing and long pressing
- management of LEDs separately from the function fulfilled by the channels and with lighting effects
- temperature sensor (*available for GW90721A only, not for GW90727*)

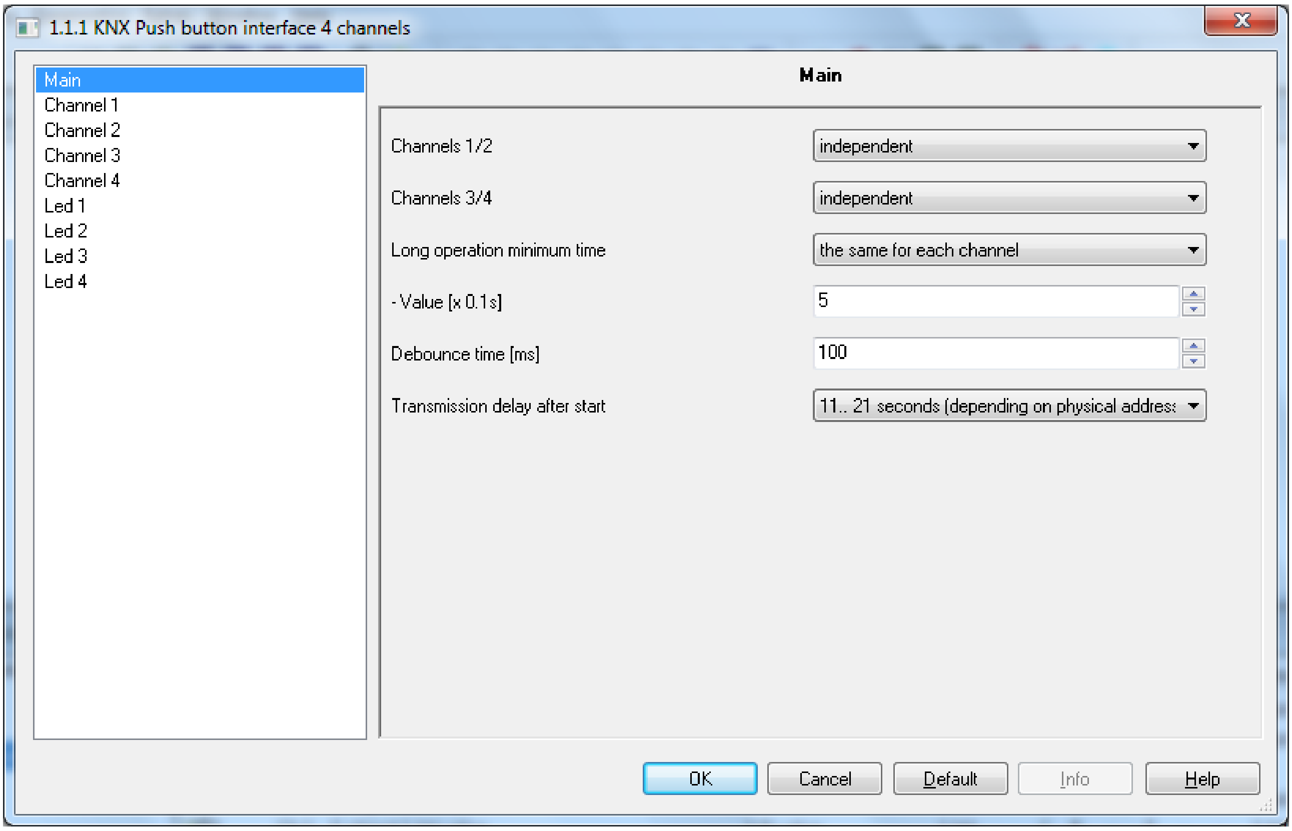
The pairs of combined inputs are pre-established. Respectively, they are: channel 1 with channel 2 and channel 3 with channel 4.

2.1 Association limits

The maximum number of logic associations that the device can store is 254. This means that the maximum number of logic connections between communication objects and group addresses is 254.
The maximum number of group addresses that the device can store is 254. This means that the communication objects can be associated with a maximum of 254 group addresses.

3 *"Main" menu*

The ***Main*** menu contains the parameters for configuring the behaviour of the device input contacts (Fig. 3.1).



Main

Channels 1/2: independent

Channels 3/4: independent

Long operation minimum time: the same for each channel

- Value [x 0.1s]: 5

Debounce time [ms]: 100

Transmission delay after start: 11.. 21 seconds (depending on physical addres:

Fig. 3.1

3.1 Parameters

3.1.1 Channels 1/2

Determines the management of channels 1 and 2. The values that can be set are:

- ***independent***
  The two channels are independent; a different function can be set for each one.

- ***combined***
  The channels are combined in order to manage a single function (e.g. for commanding a dimmer or roller shutter); each channel is associated with a different command, but they both act on the same communication objects to fulfil the configured function.

3.1.2 Channels 3/4

Determines the management of channels 3 and 4. The values that can be set, and their respective functions, are the same as described in paragraph **3.1.1**.

3.1.3 Minimum time long operation

This parameter can be used to define a single time value for all channels, or a different one for each of the channels; the values that can be set are:

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

Depending on the selected value, the following parameters are displayed: **"Value [x 0.1s]"** (if the value is **the same for each channel**) or **"Value channel 1 [x 0.1s]"**, **"Value channel 2 [x 0.1s]"**, **"Value channel 3 [x 0.1s]"**, **"Value channel 4 [x 0.1s]"**, **"Value channel 5 [x 0.1s]"** and **"Value channel 6 [x 0.1s]"** (if the value is **different for each channel**). These parameters determine the real minimum time in which the device must detect the closure of the contact in order to distinguish a short operation from a long one. The possible values are:

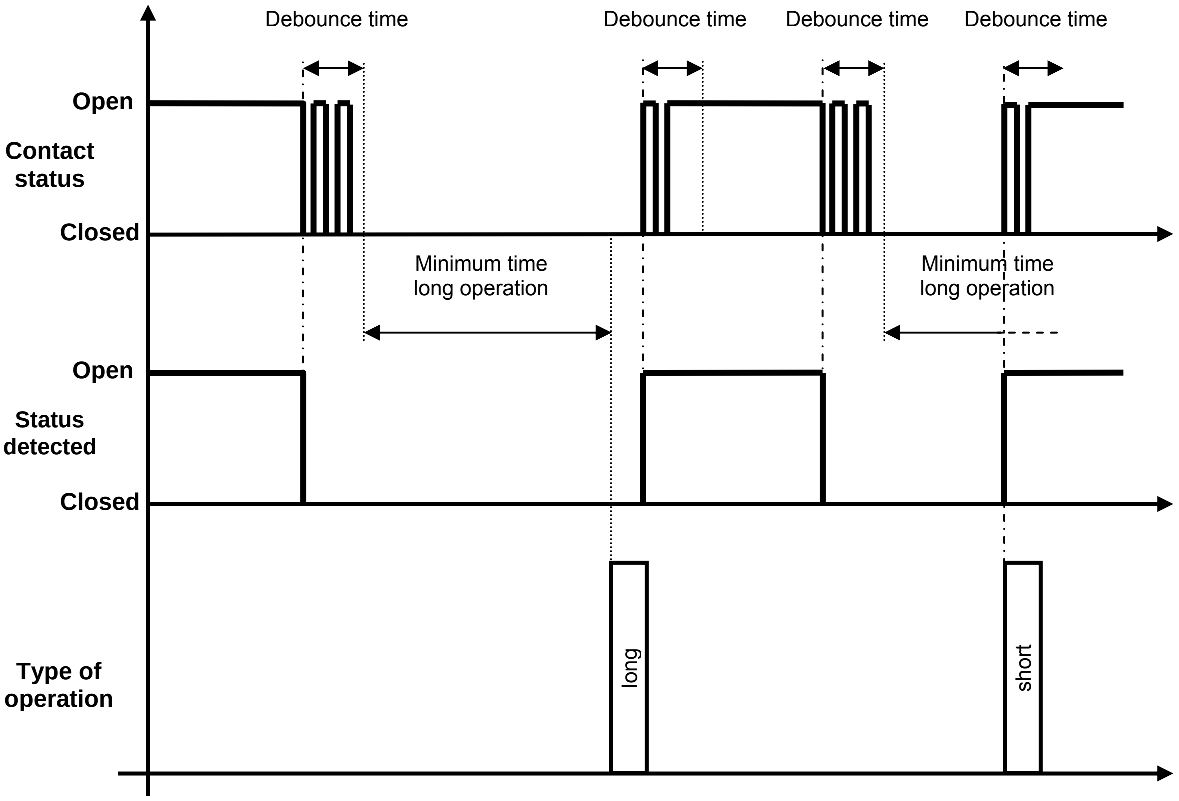
- from 3 to 150 with step 1, **5 (default value)**

3.1.4 Debounce time [ms]

When an electro-mechanical device such as a push-button is pressed, there is a series of brief bounces (quick closing and opening of the contact) before the contact shifts definitively to the open or closed status; if suitable precautions are not taken, these bounces may be detected by the application software and interpreted as multiple command activations, causing subsequent device malfunctioning.
Given that the duration of these bounces depends on the type of device used, a function has been added to the device software to avoid the problem; it basically involves inserting a delay time between the reading moments of the push-button contact status so that when a contact status variation is detected, a specific time must pass before the device can detect another variation. This value can be set in the **"Debounce time [ms]"** parameter. The values that can be set are:

- from 10 to 255 with steps of 1, **100 (default value)**

The following chart summarises the concepts of **"Value [x 0.1s]"** (long operation) and **"Debounce time [ms]"** explained above.



Starting from the top, the first chart shows a simulation of the time trend of the push-button status. The second chart shows the time trend of the push-button status detected by the device software that filters the contact disturbance (bounce) for a time equal to $\mathbf{T}_{\mathbf{debounce}}$ starting from the moment when the first variation is detected.
At the end of the debounce time, the software re-reads the contact status and, if it is the same as the last status detected and if the variation is from open status to closed status (push-button pressed), it activates a timer whose initial value is the one set in **"Value [x 0.1s]"** (long operation). If the timer expires before the status variation from closed to open is detected, the software interprets this as a long operation; otherwise, the timer is blocked and the action is considered a short operation, as shown in the third chart.

3.1.5 Delay time between switching on and first transmission

This defines the time that must pass before the device can transmit the telegrams on the BUS, following a BUS voltage drop/reset. This time is important because when the BUS voltage is reset, a large number of telegrams could be transmitted simultaneously if there are many devices in the system, and this would create collisions and hence the possible loss of some telegrams. The values that can be set are:

- ***11.. 21 seconds (depending on physical address)***
  If this value is set, the fixed value of the first transmission time is not determined directly; in this case it is calculated randomly by an algorithm on the basis of the physical address assigned to the specific device. The values 11 and 21 indicate the limits of the value range that the algorithm can assign.

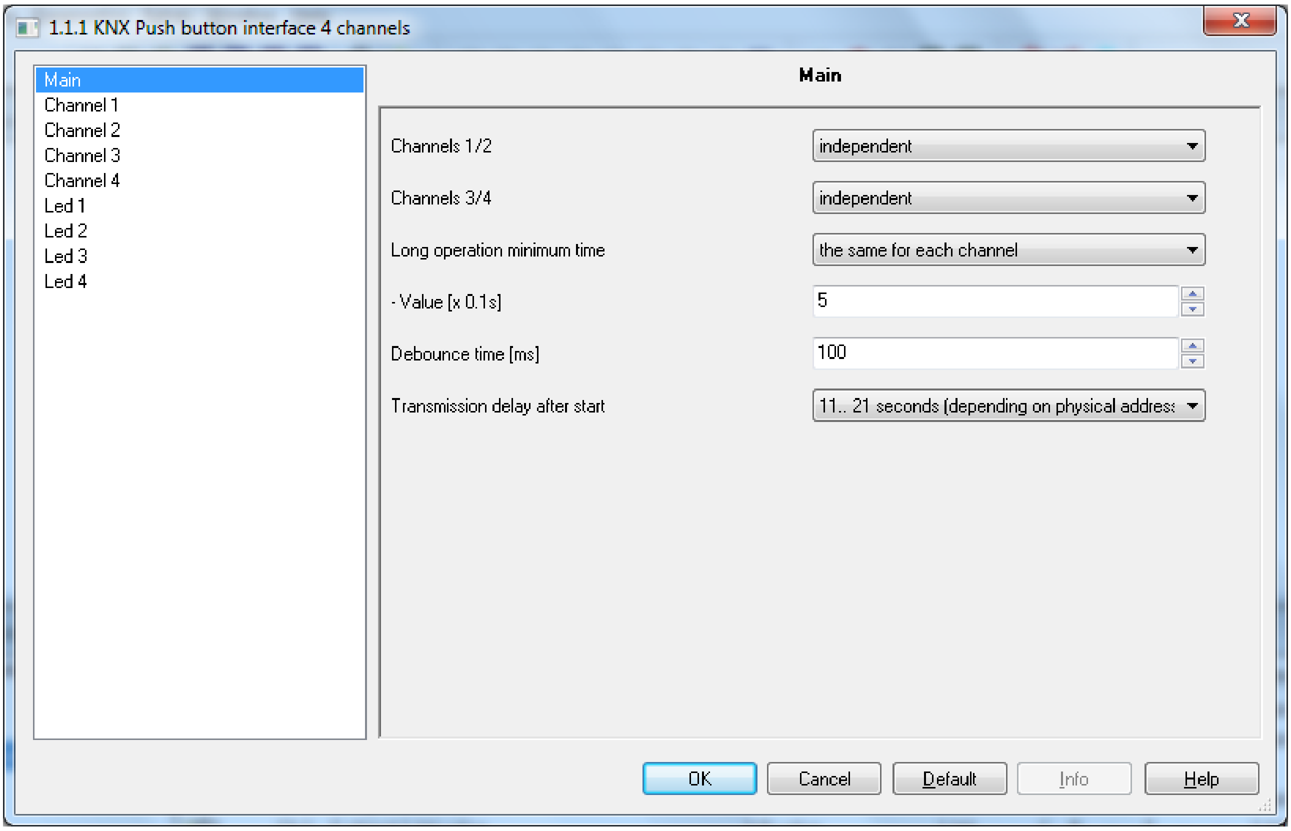
- ***5.. 9 seconds***
  As above, but with range limit values of 5 and 9 seconds.

- ***11 seconds***
- ***13 seconds***
- ***15 seconds***
- ***17 seconds***
- ***19 seconds***
- ***21 seconds***

- ***no delay***
  If this value is set, there is no delay between the moment of switch-on and the moment in which the device can send the first telegram; as soon as the device is reactivated, it can immediately send telegrams on the BUS.

4 *“Channel x”* menu (independent channels)

If channel operation is independent, a specific menu called **Channel x** is visualised for each channel (x = 1 .. 4 is the channel indicator). The menu structure changes based on the value set for the **“Matched function”** parameter. For the sake of simplicity, the parameters enabled according to the value set for the above parameter are listed in the following paragraphs.
The basic structure of the menu is as follows:



Channels 1/2: independent
Channels 3/4: independent
Long operation minimum time: the same for each channel
- Value [x 0.1s]: 5
Debounce time [ms]: 100
Transmission delay after start: 11.. 21 seconds (depending on physical addres:

Fig 4.1

4.1 Parameters

4.1.1 Matched function

This determines the function associated to the generic channel x; depending on the value set for this parameter, the ***Channel x*** menu will appear differently. The values that can be set are:

- ***none***

No function is associated with the general channel x, so it is disabled.

- ***temperature sensor***

See chapter 5 - **“*Temperature sensor*” function**

- ***edges/sequence commands***

See chapter 6 - “***Edges/sequence commands*” function**

- ***multiple press/closing contact***

See chapter 7 - **“*Multiple press/closing contact*” function**

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

See chapter 8 - “***1 push-button + stop dimmer***” **function**

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

See chapter 9 - ***"Cyclic sending 1 push-button dimmer" function***

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

See chapter 10 - **"*1 push-button shutter control*" function**

- ***scene management***

See chapter 11 - ***"Scene management"* function**

- ***pulse counter***

See chapter 12 - ***"Pulse counter"* function**

- ***switching sequences***

See chapter 13 - **"*Switching sequences*" function**

4.1.2 Block

To inhibit the channel when sending commands associated with the closure/opening or long/short enabling of the contact, the block function must be activated: this function inhibits the detection of the closure/opening or long/short enabling of the contact, thereby preventing the device from sending the telegrams associated with these events on the BUS. If it is activated, any change in status that occurs will not be interpreted until a block deactivation command is received. The parameter for enabling the function is the **"Block"** parameter, that can take the following values:

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

If **enabled** is selected, the following parameters will be visualised: **"Block activation value"** and **"Block function on BUS voltage recovery**, along with the ***Ch.x - Block*** communication item (Data Point Type: 1.003 DPT_Enable), with which you can activate the function via the BUS command.
In particular cases where a front (opening or closure) or operation (short or long) is associated with the cyclical sending of a command/value, the block works in the following way:

a. if the block is activated while the cyclical sending is active, the device continues to send cyclically throughout the period in which the block is active. When the block is deactivated, the activation condition of the cyclical sending will be checked again. If it continues to be checked, the cyclical sending will continue; otherwise, the cyclical sending will end (even if the variation occurred while the block was active, so the sending of the telegram on front detection was inhibited).
b. if the block is activated while the cyclical sending is not active, the device does not react. When the block is deactivated, the cyclical sending condition will be checked and the necessary actions will be taken (even if the variation occurred while the block was active).

4.1.3 Block activation value

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

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

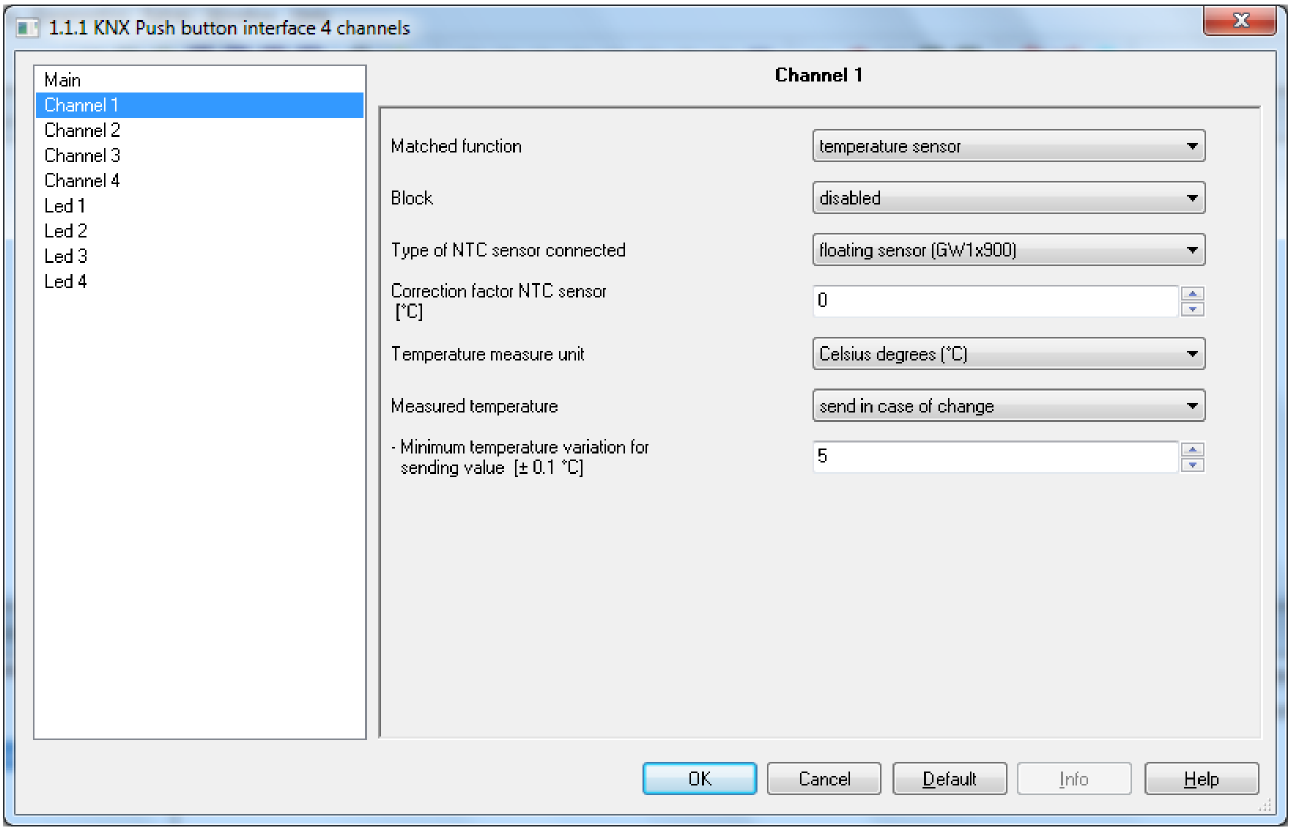
4.1.4 Block function on BUS voltage recovery

With the **"Block function on BUS voltage recovery (=Block on BUS tension recovery function)"** parameter, you can set the status of the block function when the BUS voltage is restored. the values that can be set are:

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

5 *“Temperature sensor” function (for GW90721A only)*

Various temperature sensors can be connected to the input contacts; given the different characteristics of each transducer, the **“Type of NTC sensor connected”** parameter is used to define which of the possible sensors will be connected to the device contacts, in order to interface correctly with the sensor itself. The basic menu structure is as follows:



Channel 1

Matched function: temperature sensor

Block: disabled

Type of NTC sensor connected: floating sensor (GW1x900)

Correction factor NTC sensor [°C]: 0

Temperature measure unit: Celsius degrees (°C)

Measured temperature: send in case of change

- Minimum temperature variation for sending value [± 0.1 °C]: 5

Fig. 5.1

5.1 Parameters

5.1.1 Type of NTC sensor connected

The values that can be set for this parameter are:

- **wired sensor (GW10800)** **(default value)**
- flush-mounting sensor, 1 module (GW1x900)

5.1.2 Correction factor NTC sensor [0.1°C]

This parameter is used to set the correction factor to be applied to the temperature value of the NTC sensor connected to the input, to eliminate the heat contribution generated by the installation site. The values that can be set are:

- from -20 to + 20 with steps of 1, **0 (default value)**

5.1.3 Measured temperature

This parameter is used to define the conditions for sending the value of the temperature measured by the device. the values that can be set are:

- send on demand only
- **send in case of change** **(default value)**
- send periodically
- send on change and periodically

Selecting the value **send in case of change** or **send on change and periodically**, also the parameter **"Minimum temperature variation for sending value [± 0.1°C]"** will be visible, whereas by selecting **send periodically** or **send on change and periodically** the parameter **"Temperature sending period [minutes]"** will be visible.
Selecting the value **send on demand only**, no new parameter will be enabled, as the temperature value is not sent spontaneously by the device; in the case of a status reading request, it sends the requester a telegram in response to the received command, which includes information about the measured temperature value.

5.1.4 Temperature measure unit

The **"Temperature measure unit"** parameter is used to set the measurement unit for coding and sending the information via the ***Ch.x – Temperature sensor*** communication object. The values that can be set are:

- **degrees Celsius (°C)** **(default value)**
- degrees Kelvin (°K)
- degrees Fahrenheit (°F)

The value set for this parameter affects the coding of the ***Ch.x - Temperature sensor*** communication object: *9.001 DPT_Value_Temp* if the value is **degrees Celsius (°C)**, *9.002 DPT_Value_Tempd* if the value is **degrees Kelvin (°K)** and *9.027 DPT_Value_Temp_F* if the value is **degrees Fahrenheit (°F)**.

5.1.5 Minimum temperature variation for sending value [± 0.1°C]

This parameter is visible if the temperature is sent on variation, and is used to define the minimum temperature variation (in relation to the last temperature value sent) that causes the new measured value to be spontaneously sent. The values that can be set are:

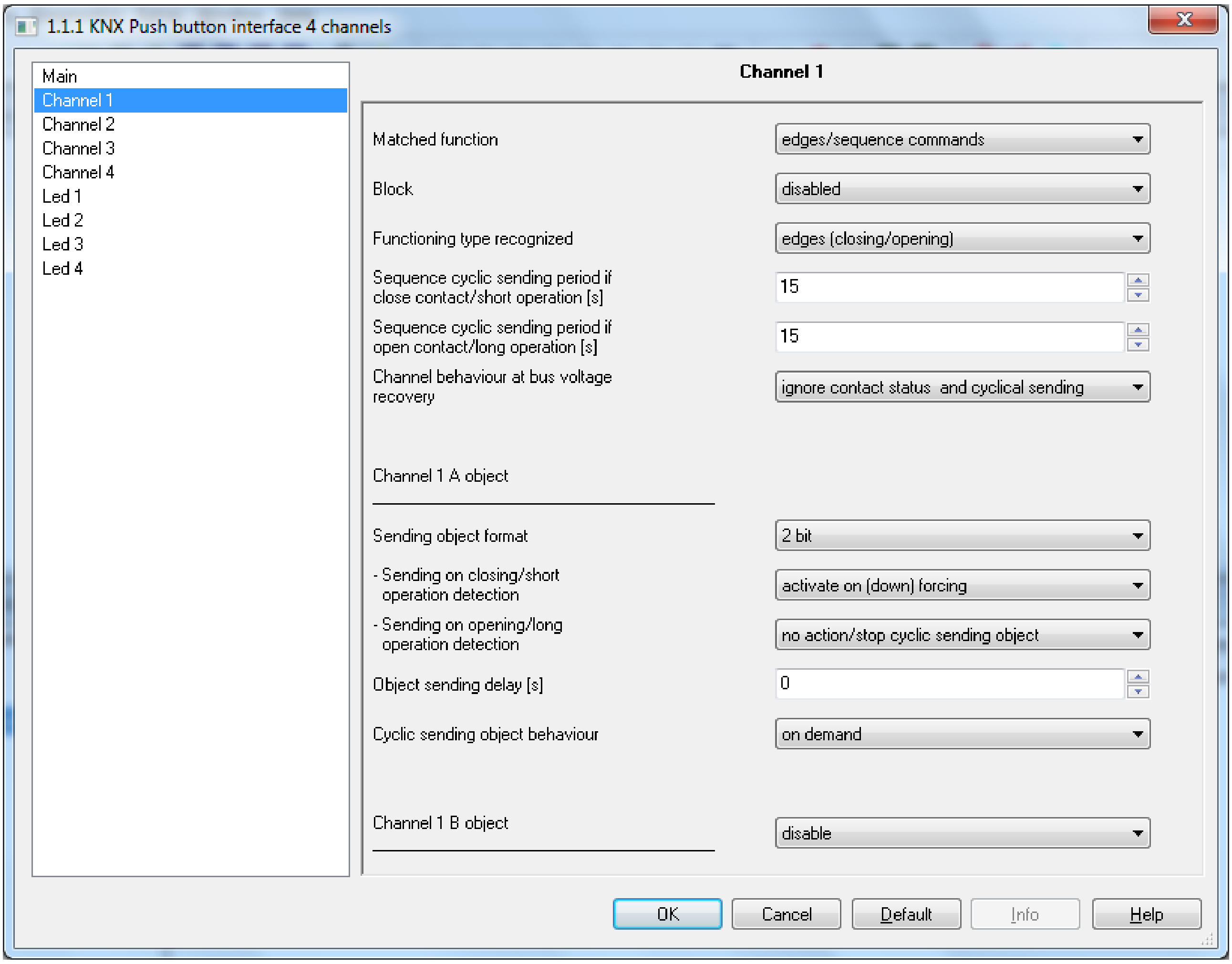
- from 1 to 10 with steps of 1, **5 (default value)**

The parameter **"Temperature sending period [minutes]"**, which is visible if the temperature is sent periodically, is used to define the period with which the measured temperature feedback telegrams are sent spontaneously; the values that can be set are:

- from 1 to 255 with steps of 1, **5 (default value)**

6 *"Edges/sequence commands"* function

This function is used to set the type and number of commands to send after a status change has been detected, for up to a total of eight commands per channel. The value of the command can be differentiated according to the event detected (closure/opening, or short/long operation). The sending of commands can also be delayed with a set fixed time, and the cyclical sending of command telegrams can be enabled.
The basic structure of the menu is as follows:



Channel 1

Matched function	edges/sequence commands
Block	disabled
Functioning type recognized	edges (closing/opening)
Sequence cyclic sending period if close contact/short operation [s]	15
Sequence cyclic sending period if open contact/long operation [s]	15
Channel behaviour at bus voltage recovery	ignore contact status and cyclical sending
Channel 1 A object	
Sending object format	2 bit
- Sending on closing/short operation detection	activate on (down) forcing
- Sending on opening/long operation detection	no action/stop cyclic sending object
Object sending delay [s]	0
Cyclic sending object behaviour	on demand
Channel 1 B object	disable

Fig. 6.1

6.1 Parameters

For each input, up to 8 different objects can be sent (distinguished by the letters A, B, C, D, E, F, G and H) on the basis of the closure (or short operation) or opening (or long operation) of the contact. Object A is always enabled, and the **"z object"** parameter (z indicates the object associated with the threshold, included between **A** and **H)** is used to enable a new object to be sent. The parameter may assume the following values:

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

If **enable** is selected, the following parameters will be visualised: **"Sending object format"**, **"Sending on closing/short operation detection"**, **"Sending on opening/long operation detection"** and **"Object sending delay [s]"**. These are grouped together in the **z object** sub-group (where z indicates the object associated with the binary input, included between **A** and **H)**.

6.1.1 Functioning type recognised

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

- **edges (closure/opening)** **(default value)**
- short operation/long operation

6.1.2 Sending object format

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

The value set for this item will alter the values that can be set for the **"Sending on closing/short operation detection"** and **"Sending on opening/long operation detection"** parameters.

6.1.3 Sending on closing/short operation detection

This parameter is used to set the command or value to be sent following the detection of the closure or short operation of the contact (depending on the type of operation selected) associated with the channel.

6.1.4 Sending on opening/long operation detection

This parameter is used to set the command or value to be sent following the detection of the opening or long operation of the contact (depending on the type of operation selected) associated with the channel.

– If the format of the object to send is **1 bit**, the ***Ch.x - 1 bit value z object*** communication object will be visible (Data Point Type: 1.002 DPT_Bool) and the values that can be set for the two parameters above are:

  - **no action/stop cyclic sending object** **(default value on detection of opening)**
  - 0
  - **1** **(closing detection default value)**
  - cyclical switching

  Selecting the value **cyclical switching**, the **"Object status feedback"** parameter will be shown, which makes it possible to enable and display the ***Ch.x - z object status feedback*** communication object (Data Point Type: 1.001 DPT_Switch); by enabling this object, when the status feedback telegram is received for the object in question, the command that the interface will send (via the ***Ch.x - z object 1 bit value*** object) when the event associated with the cyclical switching detected will be the opposite of the value generated by the most recent event between the BUS value received on the ***Ch.x - z object status feedback*** object and the last value sent (via the ***Ch.x - z object 1 bit value*** object). The **"Status feedback object"** parameter may have the following values:

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

Selecting the value **enabled** displays the ***Ch.x - z object status feedback*** communication object.

– If the format of the object to send is **2 bits**, the ***Ch.x 2 bit value z object*** communication object will be visible (Data Point Type: 2.001 DPT_Switch_Control) and the values that can be set for the two parameters listed above are:

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- **on forcing active (down)** **(default closing value)**
- forcing active off (up)
- deactivate forcing [=forcing deactivation]
- forcing ON / forcing OFF - cyclical switching
- forcing ON / deactivate forcing - cyclical switching
- forcing OFF / deactivate forcing - cyclical switching

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

– If the format of the object to send is **1 byte value without sign**, the ***Ch.x - 1 byte value z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the two parameters listed above are:

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- **send value** **(default closing value)**

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

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

– If the format of the object to send is **1 byte value with sign**, the ***Ch.x - 1 byte value z object*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the two parameters listed above are:

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- **send value** **(default closing value)**

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

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

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

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- **send value** **(default closing value)**

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

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

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

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- auto
- **comfort** **(closure default value)**
- pre-comfort
- economy
- off (building protection)
- cyclical switching (thermostat)
- cyclical switching (timed thermostat)

By selecting **cyclical switching**, in this case no communication object will be displayed as the device is always updated about the function activation status.
By selecting **cyclical switching (thermostat)**, each time the associated event (closing/opening or short/long operation) is detected, the device will send a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* … By selecting **cyclical switching (timed thermostat)**, each time the associated event (closing/opening or short/long operation) is detected, the device will send a new temperature adjustment mode (HVAC) in the order *Comfort→ Precomfort→ Economy→ Off→ Auto → Comfort …*

– If the format of the object to send is **2 bytes value without sign**, the ***Ch.x 2 byte value z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the two parameters listed above are:

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- **send value** **(closing detection default value)**

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

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

– If the format of the object to send is **2 bytes value with sign**, the ***Ch.x 2 byte value z object*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the two parameters listed above are:

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- **send value** **(closing detection default value)**

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

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

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

- **no action/stop cyclic sending object** **(default value on detection of opening)**
- **send value** **(closing detection default value)**

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

- **white** **(default value)**
- yellow
- magenta
- red

- turquoise
- green
- blue
- customise

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

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

– If the format of the object to send is **4 bytes value without sign**, the ***Ch.x 4 byte value z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the two parameters listed above are:

- **no action/stop cyclic sending object**	**(default value on detection of opening)**
- **send value**	**(closing detection default value)**

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

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

– If the format of the object to send is **4 bytes value with sign**, the ***Ch.x 4 byte value z object*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the two parameters listed above are:

- **no action/stop cyclic sending object**	**(default value on detection of opening)**
- **send value**	**(closing detection default value)**

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

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

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

- **no action/stop cyclic sending object**	**(default value on detection of opening)**
- **send value**	**(closing detection default value)**

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

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

6.1.5 Object sending delay (0.. 255 seconds)

This 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 H, this parameter indicates the delay between sending the command/value associated with the object with the previous index (z-1) and sending the command/value associated with the object to which the parameter refers; the delay in these cases is calculated from the moment when the command/value associated with the object with the previous index (z-1) is sent, not from the moment of detection of the event that generated the sending (closure/opening or short/long operation).

The set delay will only be executed if the event in progress, associated with the object to which the parameter refers, is associated with any value other than **no action**; otherwise, the delay is ignored.
The parameter may assume the following values:

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

NB: if a sequence of commands with delays - activated by the detection of a specific event (closure/opening or short/long operation) - is being sent, then the detection of the opposite event will cause the termination of the sending of that sequence, but only if at least one of the actions associated with the detection of the latter event is different from **no action**; otherwise, the command/value sequence will be continue to be sent until the last command/value has been sent.

6.1.6 Cyclic sending object condition

Given the possibility to interface various devices with the interface input contacts, it may be useful to repeat the command telegrams at pre-set intervals (especially if there is a sensor interface). The **"Cyclic sending object condition"** parameter defines the conditions for the cyclical sending of the command telegrams. The values that can be set are:

- **never** **(default value)**
- in the case of an open contact/long operation
- in the case of a closed contact/short operation
- always

By selecting **never**, the device will only send the telegram with the set value on the BUS when the contact changes from closed to open or vice versa (or when a short/long operation is detected on the contact).
By selecting **in the case of an open contact/long operation**, the device will only send the telegram with the set value on the BUS when the contact changes from closed to open (or when a long operation is detected on the contact). As long as the contact remains open (or no other operation is recognised), the device will occasionally send the value associated with the event; if a new long operation is recognised, this cyclical sending is interrupted and the sending of the sequence associated with the detected operation restarts.
By selecting **in the case of a closed contact/short operation**, the device will only send the telegram with the set value on the BUS when the contact changes from open to closed (or when a short operation is detected on the contact). As long as the contact remains closed (or no other operation is recognised), the device will occasionally send the value associated with the event; if a new short operation is recognised, the sending of the sequence associated with the detected operation restarts.
By selecting **always**, the device will only send the telegram with the set value on the BUS when the contact changes from closed to open or vice versa (or when a short/long operation is detected on the contact). The command telegram associated with the detected event is repeated at regular intervals. If a short/long operation is recognised, this cyclical sending is interrupted and the sending of the sequence associated with the detected operation restarts.
If the value **no action/stop cyclic sending object** is associated with a specific operation for all the objects enabled, then the cyclical condition will be ignored even if it is enabled. If cyclical sending is active (determined by the setting of the other operation), this is terminated.

6.1.7 Sequence cyclic sending period if close contact/short operation [s]

This parameter is used to set the repeat period for the sequence commands associated with the closed contact (or short operation) event. The values that can be set are:

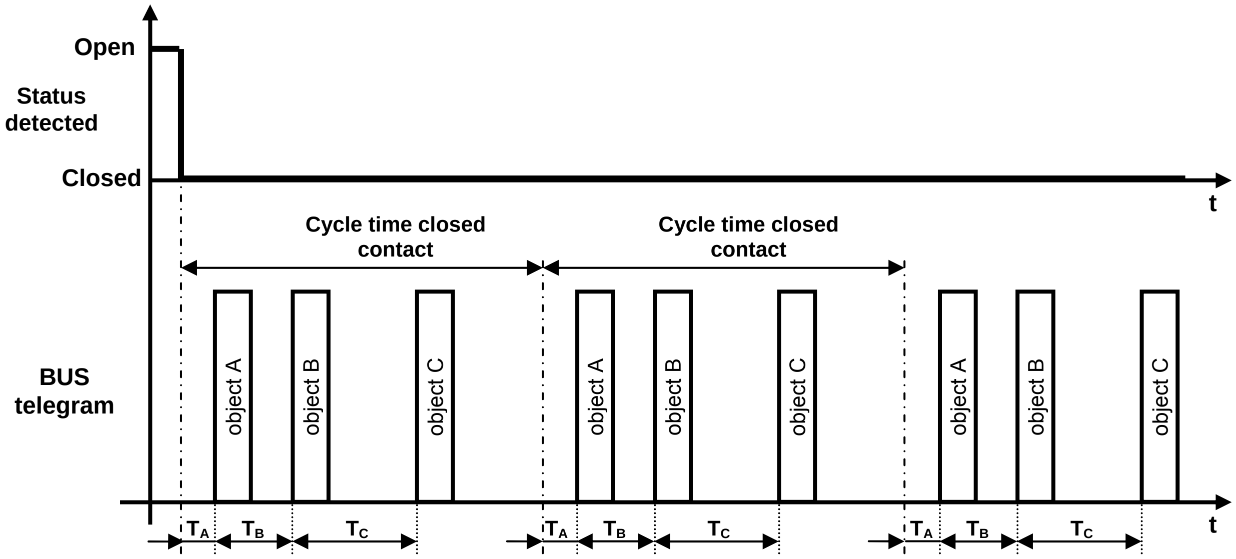
- from 1 to 65535 with steps of 1, **15 (default value)**

6.1.8 Sequence cyclic sending period if open contact/long operation [s]

This parameter is used to set the repeat period for the sequence commands associated with the open contact (or long operation) event. The values that can be set are:

- from 1 to 65535 with steps of 1, **15 (default value)**

The count of the cyclical sending time is initialised in the moment when the operation associated with cyclical sending is detected. The commands are repeated at the end of the cycle time, on the basis of the delays set (the entire command sequence is repeated). The following chart summarises the concept.



T_A= Delay on sending object A
T_B= Delay on sending object B
T_C= Delay on sending object C

The chart shows that, once contact closure has been detected, the cycle time counter is initialised along with the delay on the sending of the first object (in this case, object A). At the end of the cycle time, the whole sequence (including delays) is repeated. Throughout the repeat, the contact remains closed.

When the BUS voltage is restored, the behaviour of channel x (with regards the sending of the sequence and the cyclical sending of telegrams) can be defined via the **"Channel x behaviour at BUS voltage recovery"** parameter. The values that can be set are:

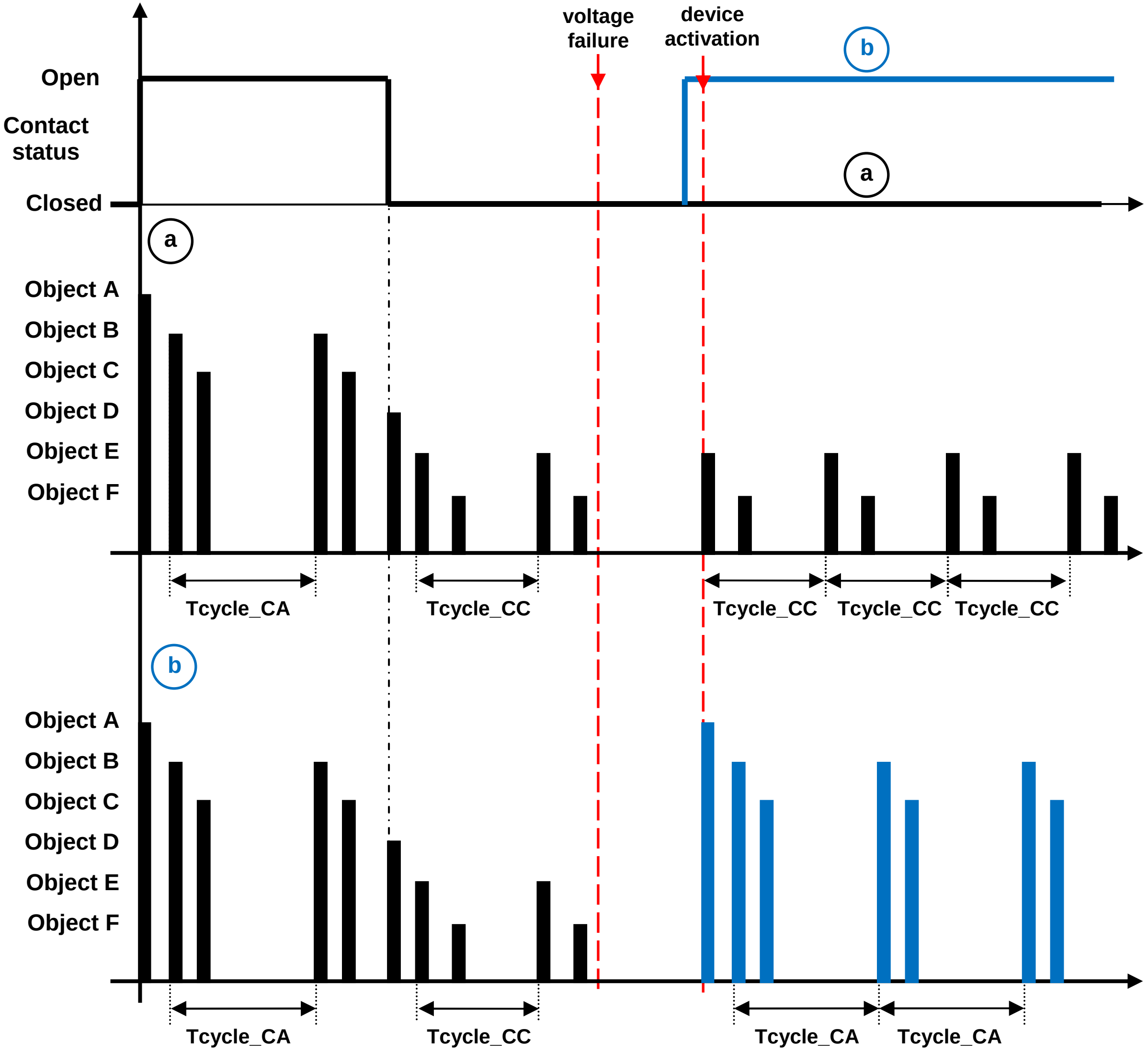
- **ignore contact status and cyclical sending** **(default value)**
- evaluate contact status and cyclical sending

By selecting **evaluate contact status and cyclical sending**, the device behaves in the following way:

- if the recognised type of operation is **edges (closing/opening)**, the device checks the contact status and:
  a) if the current status is the same as before the voltage failure, the device evaluates the value set in the **"Sending object condition**" items of all the objects of the sequence, and sends only those telegrams for which cyclical sending is enabled (as if the voltage failure had not occurred).
  b) if the current status is different from the one before the voltage failure, the device interprets the event as a new edge (occurring at switch-on) and consequently initialises the sending of the entire sequence.
- if the recognised type of operation is **short operation/long operation**, the device checks the last operation recognised before the voltage failure and, after evaluating the value set for the **"Sending object condition**" items of all the objects of the sequence, it sends only those telegrams for which cyclical sending is enabled (as if the voltage failure had not occurred).

If the value **ignore contact status and cyclical sending** is selected, no telegram is sent when the BUS voltage is restored; the status variation or a short/long operation must be detected in order to reactivate the sending of the sequence.

The following chart helps you to understand the behaviour of the device upon BUS recovery if the value "evaluate contact status and cyclical sending" is selected and the type of operation recognised is "edges" (closure/opening).



In the example above, objects A, B, C are sent on the contact opening edge and objects B and C are also sent cyclically. Objects D, E, F are sent on the contact closure edge and objects E and F are also sent cyclically. Chart “a” shows the condition in which the contact status when the device is activated following BUS voltage failure is the same as before that failure; vice versa, in chart “b” the contact status when the device is activated is different from that prior to the failure.

Chart “a”

- On the opening of the contact, the device sends the sequence of telegrams A, B and C on the basis of the set sending delays
- after a period of time equal to the period of cyclical telegram sending with an open contact (Tcycle_CA), the device again sends objects B, C for which cyclical sending is enabled
- on the closure of the contact, the device sends the sequence of telegrams D, E and F on the basis of the set sending delays
- after a period of time equal to the period of cyclical telegram sending with a closed contact (Tcycle_CC), the device again sends objects E, F for which cyclical sending is enabled
- upon recovery after a BUS voltage failure, the device detects that the contact status is “closed”, as it was prior to the failure. At this point, it sends telegrams E, F for which cyclical sending is enabled. Object D is not sent
- After a period of time equal to the period of cyclical telegram sending with a closed contact (Tcycle_CC), the device again sends objects E, F for which cyclical sending is enabled. This condition continues until contact opening is detected.

Chart “b”

- on the opening of the contact, the device sends the sequence of telegrams A, B and C on the basis of the set sending delays
- after a period of time equal to the period of cyclical telegram sending with an open contact (Tcycle_CA), the device again sends objects B, C for which cyclical sending is enabled
- on the closure of the contact, the device sends the sequence of telegrams D, E and F on the basis of the set sending delays
- after a period of time equal to the period of cyclical telegram sending with a closed contact (Tcycle_CC), the device again sends objects E, F for which cyclical sending is enabled
- upon recovery after a BUS voltage failure, the device detects that the contact status is “open”, unlike the condition prior to the failure. At this point, it sends telegrams A, B and C on the basis of the set sending delays, as if it had detected an opening edge at the time of activation
- after a period of time equal to the period of cyclical telegram sending with an open contact (Tcycle_CA), the device again sends objects B, C for which cyclical sending is enabled. This condition continues until contact closure is detected.

7 "Multiple press/closing contact" function

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

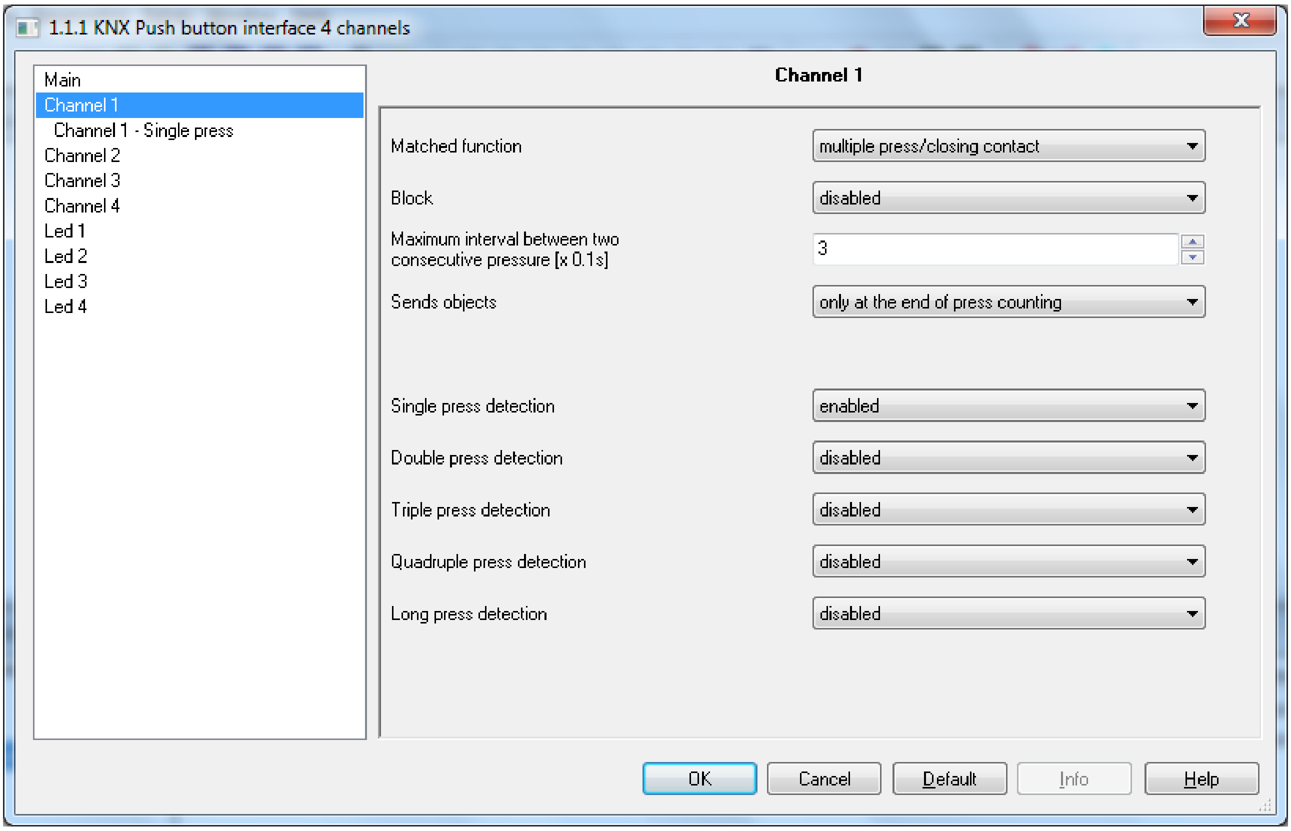
In this mode, every contact can send a series of KNX telegrams following the detection of several consecutive contact pressing operations; a pressing is recognised when the contact re-opens after a closure (open→closed→open). In particular, the device is able to distinguish the following consecutive pressings:

- single press → one pressing of the push-button
- double press → two consecutive pressings of the push-button
- triple press → three consecutive pressings of the push-button
- quadruple press → four consecutive pressings of the push-button
- long press → long contact closure

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

In order to recognise two consecutive presses, it is necessary to define the maximum gap between the detection of one press and the next; if the time between two presses (not counting the debounce time) is less than the maximum gap, the count of multiple presses is increased. When the time that elapses after the detection of a pressing (not counting the debounce time) 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 their counter.

The structure of the menu is as follows:



Channel 1

Matched function	multiple press/closing contact
Block	disabled
Maximum interval between two consecutive pressure [x 0.1s]	3
Sends objects	only at the end of press counting
Single press detection	enabled
Double press detection	disabled
Triple press detection	disabled
Quadruple press detection	disabled
Long press detection	disabled

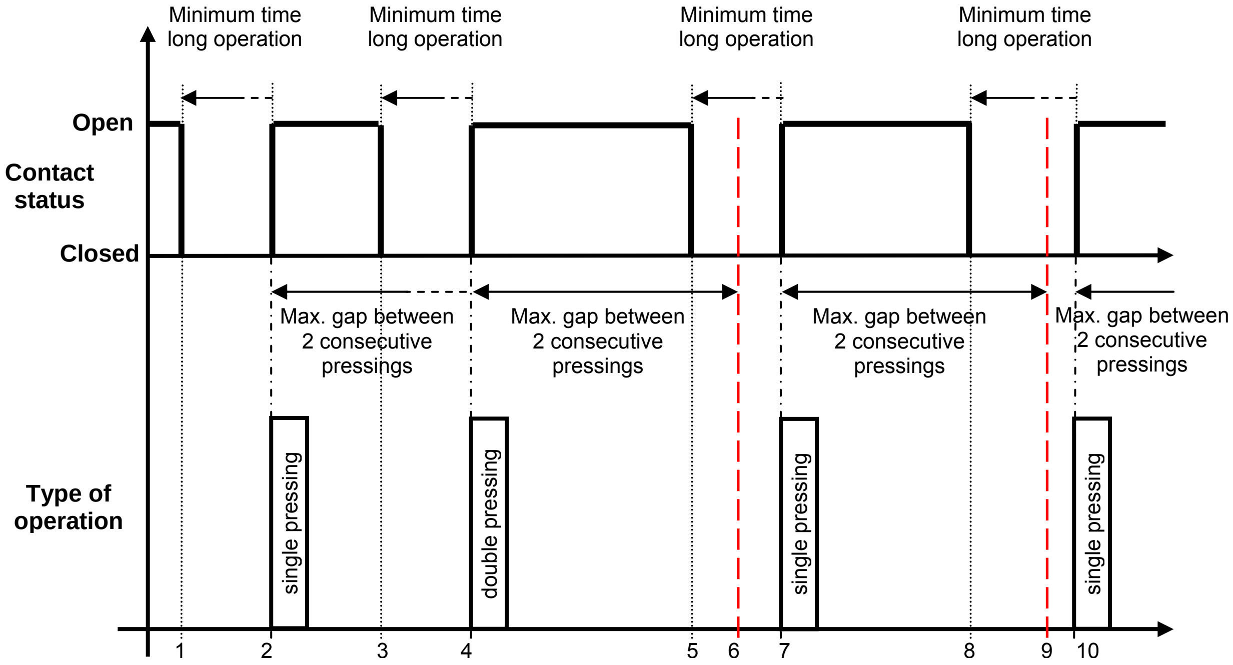
Fig. 7.1

7.1 *Maximum interval between two consecutive pressure [x 0.1s]*

This parameter is used to define the maximum gap between the detection of one press and the next, so that they are recognised as consecutive presses. The values that can be set are:

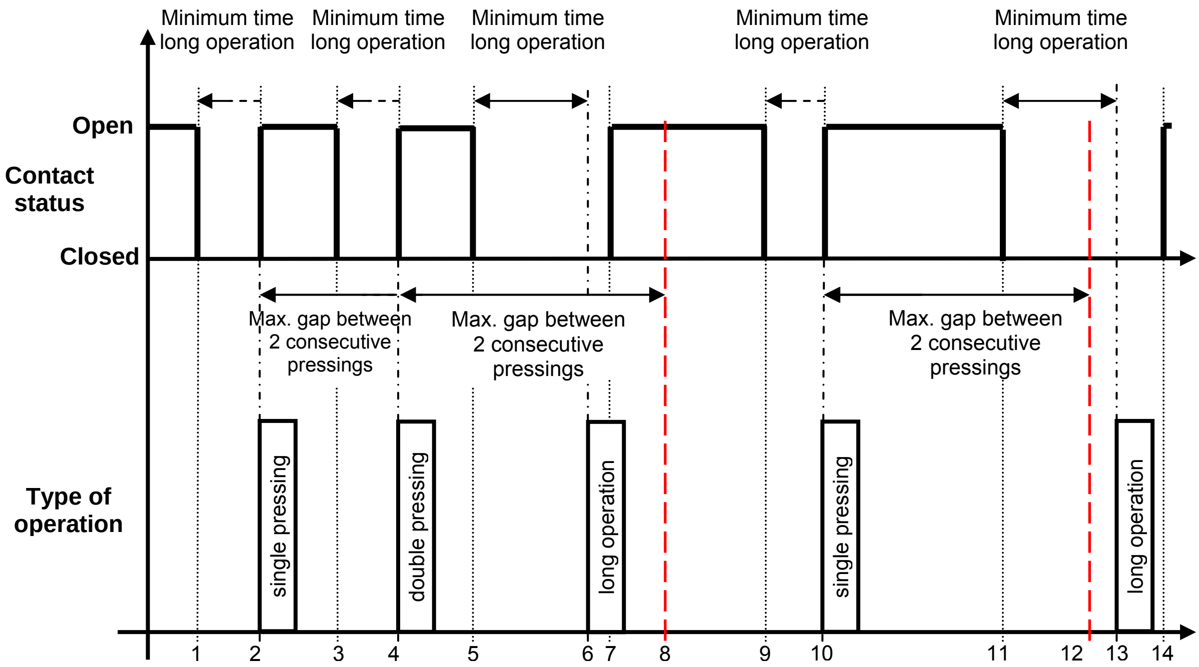
- from **3 (default value)** to 100 seconds, with steps of 1.

The following chart shows some situations that summarise the concept of multiple presses (the debounce time is not shown).


Minimum time long operation
Minimum time long operation
Minimum time long operation
Minimum time long operation
Open
Contact status
Closed
Max. gap between 2 consecutive pressings
Max. gap between 2 consecutive pressings
Max. gap between 2 consecutive pressings
Max. gap between 2 consecutive pressings
Type of operation
single pressing
double pressing
single pressing
single pressing
1
2
3
4
5
6
7
8
9
10

1. Once the closure of the contact has been detected, the contact closure time is calculated in order to distinguish a short press from a long one.
2. When the re-opening of the contact is detected before the long operation time, a short press is recognised and the count of the gap between two consecutive presses is started. The multiple press count is increased.
3. A new contact closure leads to the initialisation of the contact closure 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 re-opening of the contact before the long operation 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. A new contact closure leads to the initialisation of the contact closure 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. The re-opening of the contact before the long operation time means the detection of a new short press that increases the multiple press count and initialises the count of the gap between two consecutive presses.
8. A new contact closure leads to the initialisation of the contact closure 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. The re-opening of the contact before the long operation time means the detection of a new short press that increases the multiple press count and initialises the count of the gap between two consecutive presses.

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 (the debounce time is not shown).


Minimum time long operation
Minimum time long operation
Minimum time long operation
Minimum time long operation
Minimum time long operation
Open
Contact status
Closed
Max. gap between 2 consecutive pressings
Max. gap between 2 consecutive pressings
Max. gap between 2 consecutive pressings
Type of operation
single pressing
double pressing
long operation
single pressing
long operation
1
2
3
4
5
6
7
8
9
10
11
12
13
14

1. Once the closure of the contact has been detected, the contact closure time is calculated in order to distinguish a short press from a long one.
2. When the re-opening of the contact is detected before the long operation time, a short press is recognised and the count of the gap between two consecutive presses is started. The multiple press count is increased.
3. A new contact closure leads to the initialisation of the contact closure 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 re-opening of the contact before the long operation 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. A new contact closure leads to the initialisation of the contact closure 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 contact remains closed 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 calculation of the gap between two consecutive presses nor the multiple press count is modified in any way.
7. The re-opening of the contact following the recognition of a long press does not lead to 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. A new contact closure leads to the initialisation of the contact closure 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. The re-opening of the contact before the long operation time means the detection of a new short press that increases the multiple press count and initialises the count of the gap between two consecutive presses.
11. A new contact closure leads to the initialisation of the contact closure 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 contact remains closed 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 calculation of the gap between two consecutive presses nor the multiple press count is modified in any way.
14. The re-opening of the contact following the recognition of a long press does not lead to any action.

7.2 Single press detection

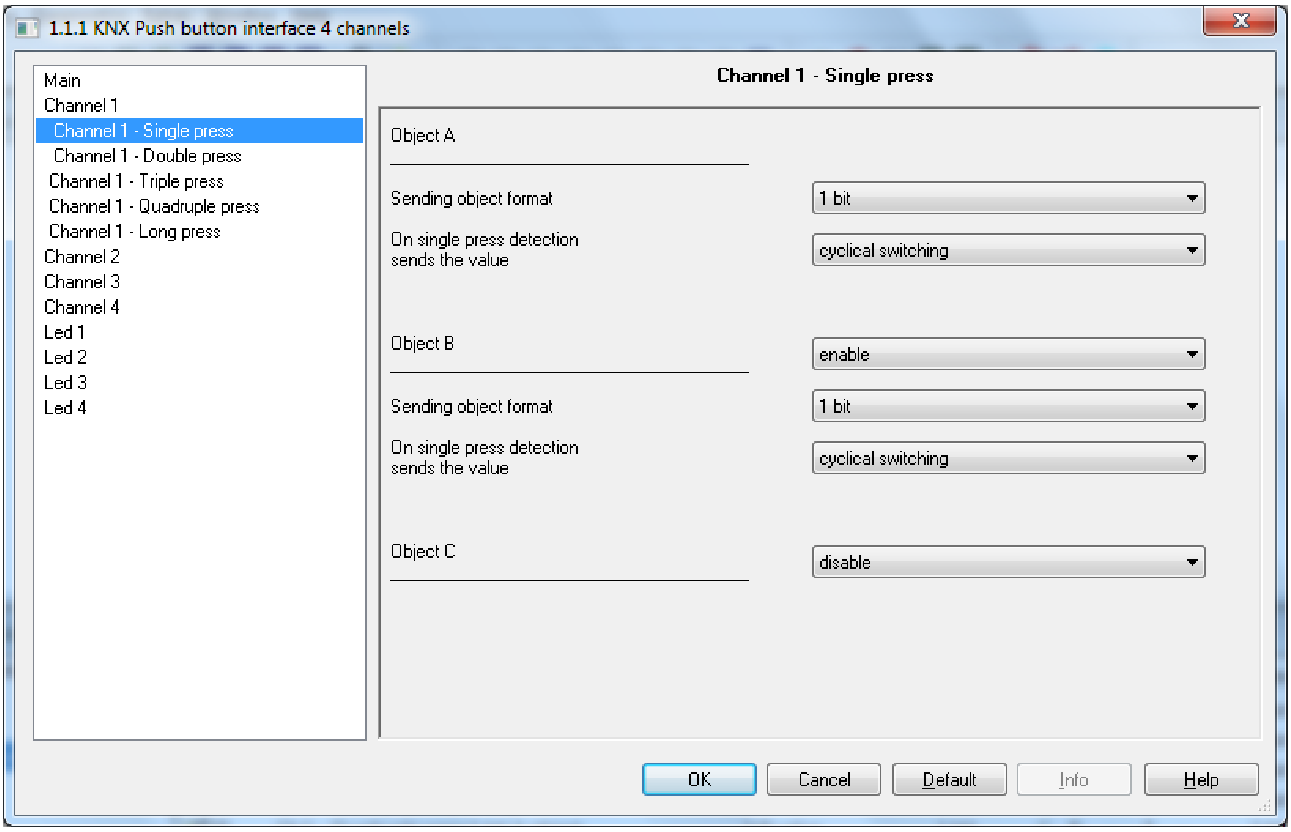
This parameter is used to enable the recognition of a single press, and to visualise the **Channel x - Single press** menu for enabling and configuring the commands that will be sent following the recognition of a single press. The values that can be set are:

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

By selecting **enabled**, the **Channel x - Single press** menu is made visible (see "Channel x - Single press" menu).

7.2.1 "Channel x - Single press" menu

This menu is visible if the value of the **"Single press detection"** parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the "single press" event.
The structure of the menu is as follows:



1.1.1 KNX Push button interface 4 channels

Main
Channel 1
Channel 1 - Single press
Channel 1 - Double press
Channel 1 - Triple press
Channel 1 - Quadruple press
Channel 1 - Long press
Channel 2
Channel 3
Channel 4
Led 1
Led 2
Led 3
Led 4

Channel 1 - Single press

Object A
Sending object format: 1 bit
On single press detection sends the value: cyclical switching
Object B: enable
Sending object format: 1 bit
On single press detection sends the value: cyclical switching
Object C: disable

Fig. 7.2

Using the **"Object A"** parameter, you can enable the parameters of the relative object and, at the same time, visualise the parameter for enabling **"Object B"** and so on until you reach the eighth (and last) object, **"Object H"**.

Depending on the value set, the **"Sending object format"** and **"On single press detection sends the value"** parameters - located in the **Object z** sub-group (where z indicates the object associated with the channel, included between **A** and **H**) - will appear for each of the selected objects.

7.2.2 “Sending object format” parameter

The **“Sending object format”** parameter is used to set the format and code of the “z” object of the “x” channel that will be sent by the device. The values that can be set are:

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

7.2.3 “On single press detection sends the value” parameter

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

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

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

  If **cyclical switching** is selected, the command that the interface will send (via ***Ch.x - Single press 1 bit z object***) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via ***Ch.x - Single press 1 bit z object***).

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

  - forcing active on (down)
  - forcing active off (up)
  - deactivate forcing [=forcing deactivation]
  - forcing ON / forcing OFF - cyclical switching
  - **cyclical switching - forcing ON / deactivate forcing (default value)**
  - forcing OFF / deactivate forcing - cyclical switching

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

– If the format of the object to send is **1 byte value without sign**, the ***Ch.x - Single press 1 byte z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **1 byte value with sign**, the ***Ch.x - Single press 1 byte z object*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

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

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

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

  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

  By selecting **cyclical switching (thermostat)**, each time the associated event is detected (single press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* … By selecting **cyclical switching (timed thermostat)**, each time the associated event is detected (single press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* …

– If the format of the object to send is **2 bytes value without sign**, the ***Ch.x - Single press 2 byte z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **2 bytes value with sign**, the ***Ch.x - Single press 2 byte z object*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **3 bytes RGB colour**, the **“On single press detection sends the value”** parameter is a dummy one, used to select the colour to be sent. The real value, downloaded from 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*** communication object will be visible (Data Point Type: 232.600 DPT_Colour_RGB) and the values that can be set for the above parameter are:

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

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

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

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

– If the format of the object to send is **4 bytes value without sign**, the ***Ch.x - Single press 4 byte z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **4 bytes value with sign**, the ***Ch.x - Single press 4 byte z object*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the above parameter are:

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

7.3 Double press detection

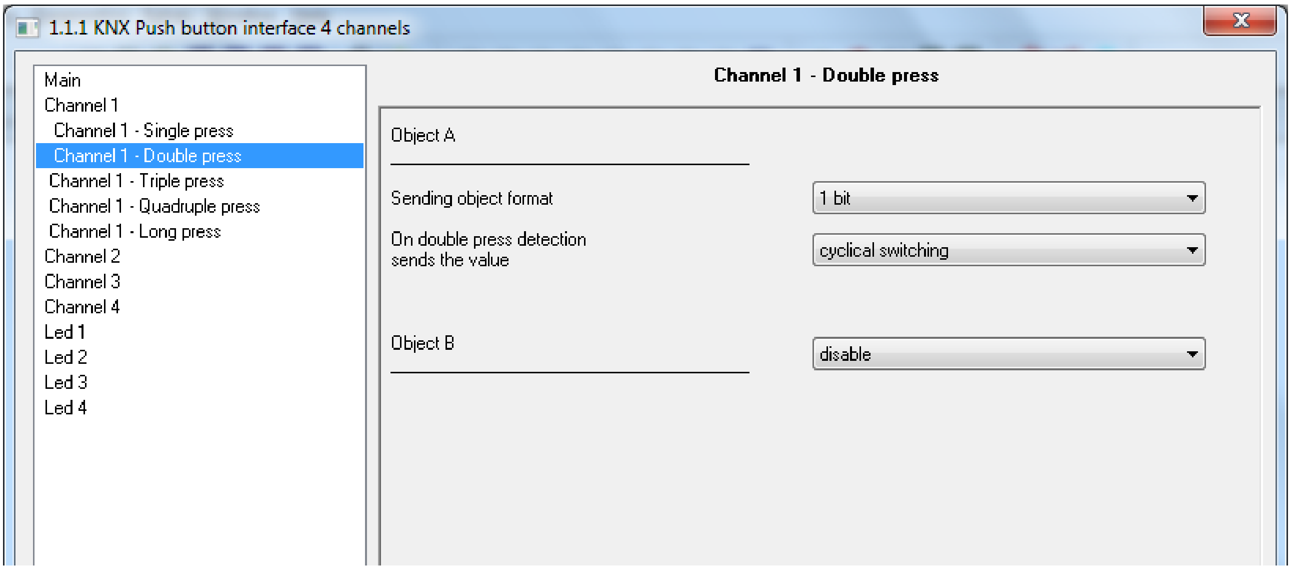
This parameter is used to enable the recognition of a double press, and to visualise the **Channel x - Double press** menu for enabling and configuring the commands that will be sent following the recognition of a double press. The values that can be set are:

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

By selecting **enabled**, the **Channel x - Double press** menu is made visible (see “Channel x - Double press” menu).

7.3.1 “Channel x - Double press” menu

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


1.1.1 KNX Push button interface 4 channels

Channel 1 - Double press

Object A

Sending object format: 1 bit

On double press detection sends the value: cyclical switching

Object B: disable

Fig. 7.3

Using the **“Object A”** parameter, you can enable the parameters of the relative object and, at the same time, visualise the parameter for enabling **“Object B”** and so on until you reach the eighth (and last) object, **“Object H”**.

Depending on the value set, the **“Sending object format”** and **“On double press detection sends the value”** parameters - located in the **Object z** sub-group (where z indicates the object associated with the channel, included between **A** and **H**) - will appear for each of the selected objects.

7.3.2 “Sending object format” parameter

The **“Sending object format”** parameter is used to set the format and code of the “z” object of the “x” channel that will be sent by the device. The values that can be set are:

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

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 be different.

7.3.3 “On double press detection sends the value” parameter

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

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

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

  If **cyclical switching** is selected, the command that the interface will send (via ***Ch.x - Double press 1 bit z object***) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via ***Ch.x - Double press 1 bit z object***).

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

  - forcing active on (down)
  - forcing active off (up)
  - deactivate forcing [=forcing deactivation]
  - forcing ON / forcing OFF - cyclical switching
  - **cyclical switching - forcing ON / deactivate forcing (default value)**
  - forcing OFF / deactivate forcing - cyclical switching

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

- If the format of the object to send is **1 byte value without sign**, the ***Ch.x - Double press 1 byte z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **1 byte value with sign**, the ***Ch.x - Double press 1 byte z object*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

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

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

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

  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

  By selecting **cyclical switching (thermostat)**, each time the associated event is detected (double press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* … By selecting **cyclical switching (timed thermostat)**, each time the associated event is detected (double press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* …

- If the format of the object to send is **2 bytes value without sign**, the ***Ch.x - Double press 2 byte z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **2 bytes value with sign**, the ***Ch.x - Double press 2 byte z object*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **3 bytes RGB colour**, the **"On double press detection sends the value"** parameter is a dummy one, used to select the colour to be sent. The real value, downloaded from 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*** communication object will be visible (Data Point Type: 232.600 DPT_Colour_RGB) and the values that can be set for the above parameter are:

  - **white (default value)**

- yellow
- magenta
- red
- turquoise
- green
- blue
- customise

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

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

– If the format of the object to send is **4 bytes value without sign**, the ***Ch.x - Double press 4 byte z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **4 bytes value with sign**, the ***Ch.x - Double press 4 byte z object*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the above parameter are:

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

7.4 Triple press detection

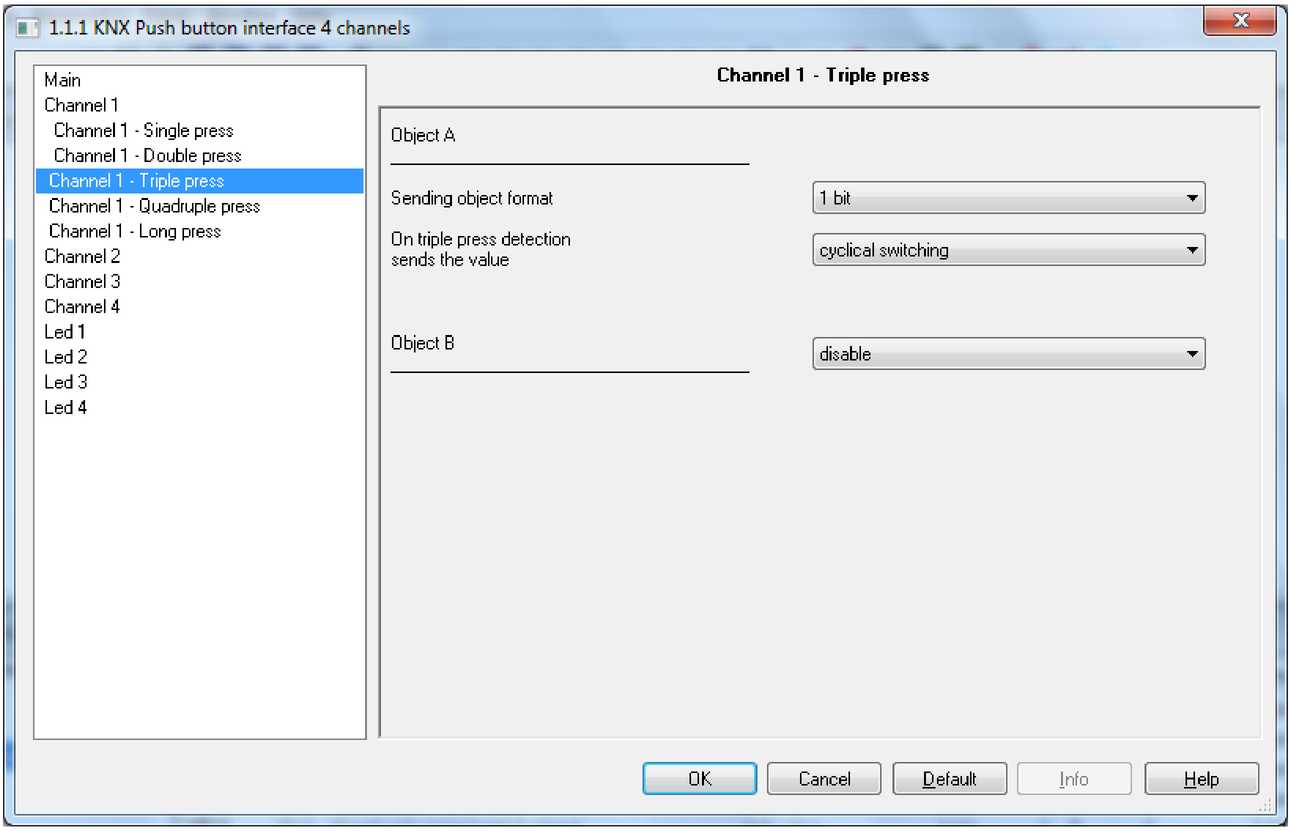
This parameter is used to enable the recognition of a triple press, and to visualise the **Channel x - Triple press** menu for enabling and configuring the commands that will be sent following the recognition of a triple press. The values that can be set are:

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

By selecting **enabled**, the **Channel x - Triple press** menu is made visible (see "Channel x - Triple press" menu).

7.4.1 "Channel x - Triple press" menu

This menu is visible if the value of the **"Triple press detection"** parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the "triple press" event.
The structure of the menu is as follows:



1.1.1 KNX Push button interface 4 channels

Channel 1 - Triple press

Object A

Sending object format: 1 bit

On triple press detection sends the value: cyclical switching

Object B: disable

Fig. 7.4

Using the **"Object A"** parameter, you can enable the parameters of the relative object and, at the same time, visualise the parameter for enabling **"Object B"** and so on until you reach the eighth (and last) object, **"Object H"**.

Depending on the value set, the **"Sending object format"** and **"On triple press detection sends the value"** parameters - located in the **Object z** sub-group (where z indicates the object associated with the channel, included between **A** and **H**) - will appear for each of the selected objects.

7.4.2 "Sending object format" parameter

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

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

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 be different.

7.4.3 “On triple press detection sends the value” parameter

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

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

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

  If **cyclical switching** is selected, the command that the interface will send (via ***Ch.x - Triple press 1 bit z object***) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via ***Ch.x - Triple press 1 bit z object***).

- If the format of the object to send is **2 bits**, the ***Ch.x - Triple press 2 bit z object*** communication object will be visible (Data Point Type: 2.001 DPT_Switch_Control) and the values that can be set for the above parameter are:

  - forcing active on (down)
  - forcing active off (up)
  - deactivate forcing [=forcing deactivation]
  - forcing ON / forcing OFF - cyclical switching
  - **cyclical switching - forcing ON / deactivate forcing (default value)**
  - forcing OFF / deactivate forcing - cyclical switching

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

- If the format of the object to send is **1 byte value without sign**, the ***Ch.x - Triple press 1 byte z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **1 byte value with sign**, the ***Ch.x - Triple press 1 byte z object*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

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

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

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

  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)

- **cyclical switching (thermostat) (default value)**
- cyclical switching (timed thermostat)

By selecting **cyclical switching (thermostat)**, each time the associated event is detected (triple press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Comfort* … By selecting **cyclical switching (timed thermostat)**, each time the associated event is detected (triple press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort* …

– If the format of the object to send is **2 bytes value without sign**, the ***Ch.x - Triple press 2 byte z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **2 bytes value with sign**, the ***Ch.x - Triple press 2 byte z object*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **3 bytes RGB colour**, the **"On triple press detection sends the value"** parameter is a dummy one, used to select the colour to be sent. The real value, downloaded from 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*** communication object will be visible (Data Point Type: 232.600 DPT_Colour_RGB) and the values that can be set for the above parameter are:

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

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

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

– If the format of the object to send is **4 bytes value without sign**, the ***Ch.x - Triple press 4 byte z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **4 bytes value with sign**, the ***Ch.x - Triple press 4 byte z object*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the above parameter are:

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

7.5 *Quadruple press detection*

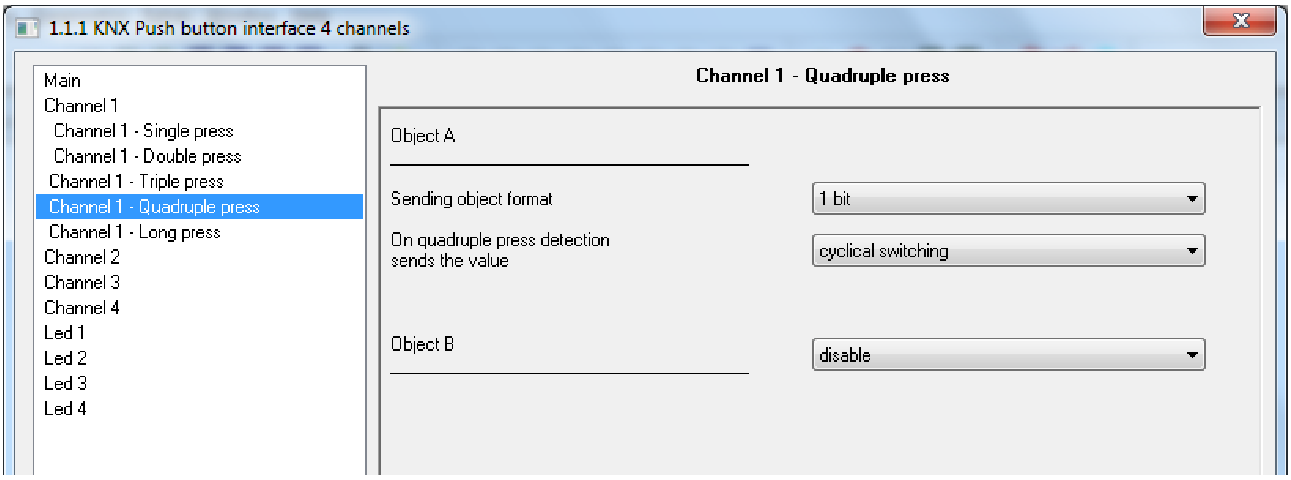
This parameter is used to enable the recognition of a quadruple press, and to visualise the **Channel x - Quadruple press** menu for enabling and configuring the commands that will be sent following the recognition of a quadruple press. The values that can be set are:

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

By selecting **enabled**, the **Channel x - Quadruple press** menu is made visible (see "Channel x - Quadruple press" menu).

7.5.1 "Channel x - Quadruple press" menu

This menu is visible if the value of the **"Quadruple press detection"** parameter of the **Channel x** menu is **enabled**. It is used to configure the communication objects and the relative values - that the device must send on the BUS - associated with the "quadruple press" event.
The structure of the menu is as follows:



Sending object format: 1 bit

On quadruple press detection sends the value: cyclical switching

Object B: disable

Fig. 7.5

Using the **"Object A"** parameter, you can enable the parameters of the relative object and, at the same time, visualise the parameter for enabling **"Object B"** and so on until you reach the eighth (and last) object, **"Object H"**.

Depending on the value set, the **"Sending object format"** and **"On quadruple press detection sends the value"** parameters - located in the **Object z** sub-group (where z indicates the object associated with the channel, included between **A** and **H**) - will appear for each of the selected objects.

7.5.2 "Sending object format" parameter

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

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

- 3 bytes RGB colour

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 be different.

7.5.3 “On quadruple press detection sends the value” parameter

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

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

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

  If **cyclical switching** is selected, the command that the interface will send (via ***Ch.x - Quadruple press 1 bit z object***) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via ***Ch.x - Quadruple press 1 bit z object***).

- If the format of the object to send is **2 bits**, the ***Ch.x - Quadruple press 2 bit z object*** communication object will be visible (Data Point Type: 2.001 DPT_Switch_Control) and the values that can be set for the above parameter are:

  - forcing active on (down)
  - forcing active off (up)
  - deactivate forcing [=forcing deactivation]
  - forcing ON / forcing OFF - cyclical switching
  - **cyclical switching - forcing ON / deactivate forcing (default value)**
  - forcing OFF / deactivate forcing - cyclical switching

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

- If the format of the object to send is **1 byte value without sign**, the ***Ch.x - Quadruple press 1 byte z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **1 byte value with sign**, the ***Ch.x - Quadruple press 1 byte z object*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

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

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

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

  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

  By selecting **cyclical switching (thermostat)**, each time the associated event is detected (quadruple press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Comfort …* By selecting **cyclical switching (timed thermostat)**, each time the associated event is detected (quadruple press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort …*

- If the format of the object to send is **2 bytes value without sign**, the ***Ch.x - Quadruple press 2 byte z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **2 bytes value with sign**, the ***Ch.x - Quadruple press 2 byte z object*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **3 bytes RGB colour**, the **"On quadruple press detection sends the value"** parameter is a dummy one, used to select the colour to be sent. The real value, downloaded from 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*** communication object will be visible (Data Point Type: 232.600 DPT_Colour_RGB) and the values that can be set for the above parameter are:

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

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

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

- If the format of the object to send is **4 bytes value without sign**, the ***Ch.x - Quadruple press 4 byte z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **4 bytes value with sign**, the ***Ch.x - Quadruple press 4 byte z object*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the above parameter are:

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

7.6 Long press detection

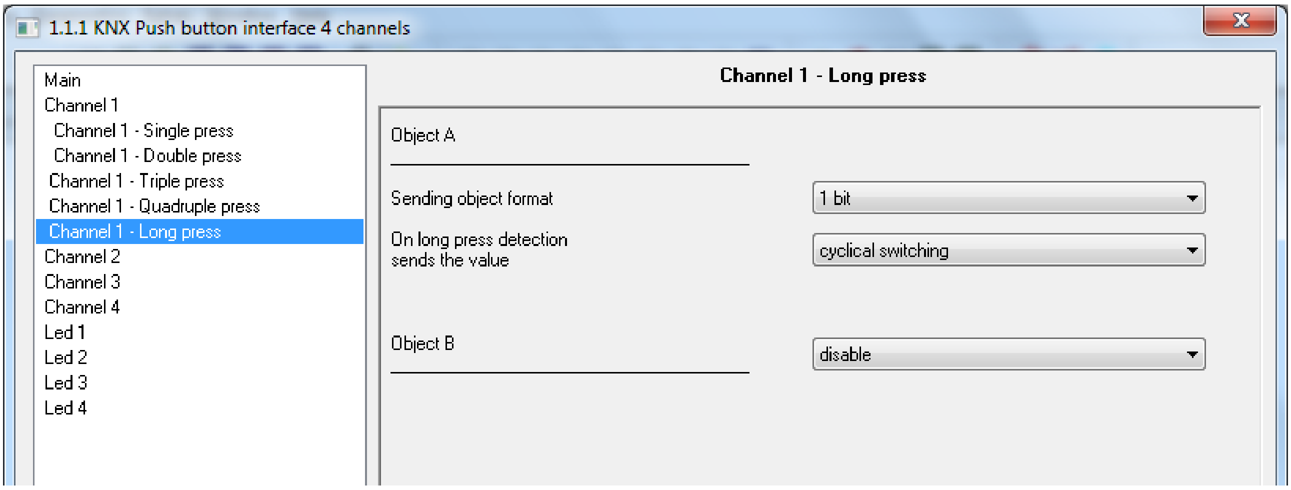
This parameter is used to enable the recognition of a long press, and to visualise the **Channel x - Long press** menu for enabling and configuring the commands that will be sent following the recognition of a long press. The values that can be set are:

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

By selecting **enabled**, the **Channel x - Long press** menu is made visible (see “Channel x - Long press” menu).

7.6.1 “Channel x - Long press” menu

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



Channel 1 - Long press

Object A

Sending object format: 1 bit

On long press detection sends the value: cyclical switching

Object B: disable

Fig. 7.6

Using the **“Object A”** parameter, you can enable the parameters of the relative object and, at the same time, visualise the parameter for enabling **“Object B”** and so on until you reach the eighth (and last) object, **“Object H”**.

Depending on the value set, the **“Sending object format”** and **“On long press detection sends the value”** parameters - located in the **Object z** sub-group (where z indicates the object associated with the channel, included between **A** and **H**) - will appear for each of the selected objects.

7.6.2 “Sending object format” parameter

The **“Sending object format”** parameter is used to set the format and code of the “z” object of the “x” channel that will be sent by the device. The values that can be set are:

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

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

7.6.3 “On long press detection sends the value” parameter

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

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

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

  If **cyclical switching** is selected, the command that the interface will send (via ***Ch.x - Long press 1 bit z object***) when the event associated with cyclical switching is detected will be the opposite of the last value sent (via ***Ch.x - Long press 1 bit z object***).

– If the format of the object to send is **2 bits**, the ***Ch.x - Long press 2 bit z object*** communication object will be visible (Data Point Type: 2.001 DPT_Switch_Control) and the values that can be set for the above parameter are:

  - forcing active on (down)
  - forcing active off (up)
  - deactivate forcing [=forcing deactivation]
  - forcing ON / forcing OFF - cyclical switching
  - **cyclical switching - forcing ON / deactivate forcing (default value)**
  - forcing OFF / deactivate forcing - cyclical switching

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

– If the format of the object to send is **1 byte value without sign**, the ***Ch.x - Long press 1 byte z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **1 byte value with sign**, the ***Ch.x - Long press 1 byte z object*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

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

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

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

  - auto mode
  - comfort mode
  - pre-comfort mode
  - economy mode
  - off mode (building protection)
  - **cyclical switching (thermostat) (default value)**
  - cyclical switching (timed thermostat)

  By selecting **cyclical switching (thermostat)**, each time the associated event is detected (long press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Comfort …* By selecting **cyclical switching (timed thermostat)**, each time the associated event is detected (long press) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort→ Precomfort→ Economy→ Off→ Auto→ Comfort …*

- If the format of the object to send is **2 bytes value without sign**, the ***Ch.x - Long press 2 byte z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **2 bytes value with sign**, the ***Ch.x - Long press 2 byte z object*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

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

- If the format of the object to send is **3 bytes RGB colour**, the **"On long press detection sends the value"** parameter is a dummy one, used to select the colour to be sent. The real value, downloaded from 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*** communication object will be visible (Data Point Type: 232.600 DPT_Colour_RGB) and the values that can be set for the above parameter are:

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

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

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

– If the format of the object to send is **4 bytes value without sign**, the ***Ch.x - Long press 4 byte z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the object to send is **4 bytes value with sign**, the ***Ch.x - Long press 4 byte z object*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the above parameter are:

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

7.7 Sending 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, consequently 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

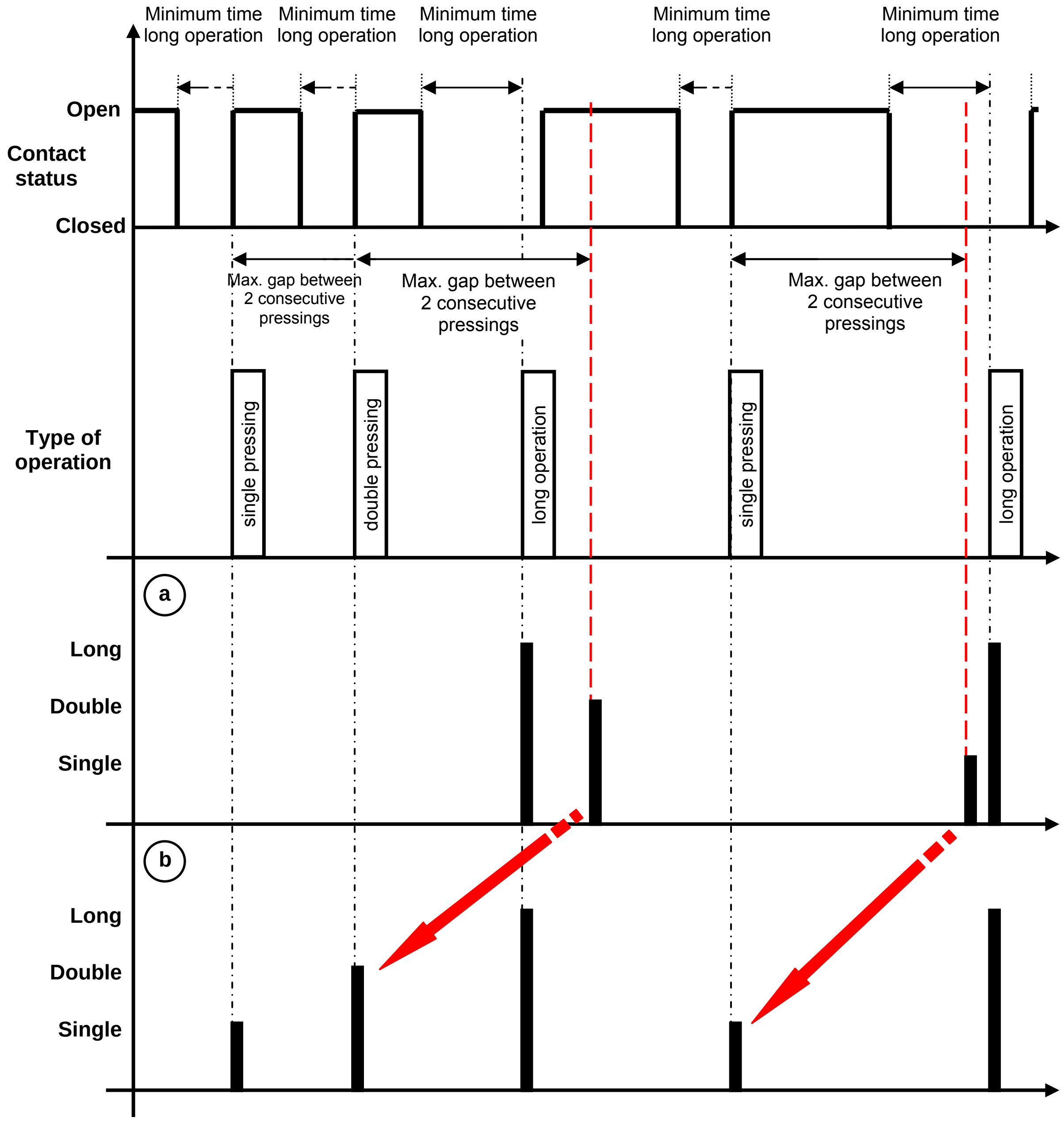
The commands associated with a "long press" are always sent as soon as the long press is detected.

The **"Send objects"** parameter is used to define the sending conditions of the objects associated with multiple presses. The values that can be set are:

- with every press detected
- **only at the end of the press count (default value)**

Selecting **only at the end of the press count**, the device behaves as described in point "a". Selecting **with every press detected**, the device behaves as described in point "b".

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


Minimum time long operation
Minimum time long operation
Minimum time long operation
Minimum time long operation
Minimum time long operation
Open
Contact status
Closed
Max. gap between 2 consecutive pressings
Max. gap between 2 consecutive pressings
Max. gap between 2 consecutive pressings
Type of operation
single pressing
double pressing
long operation
single pressing
long operation
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 sections at the bottom show the commands sent on the KNX BUS if the sending is **only at the end of the press count** (case “a”) or **with every press detected** (case “b”). The main difference between the two cases is that in case “b”, every time a multiple press is counted, the associated telegrams are sent, while in case “a” it is necessary to wait until the time between two consecutive presses exceeds the maximum value in order to end the multiple press count, and the telegrams sent are only those associated with the last press detected.

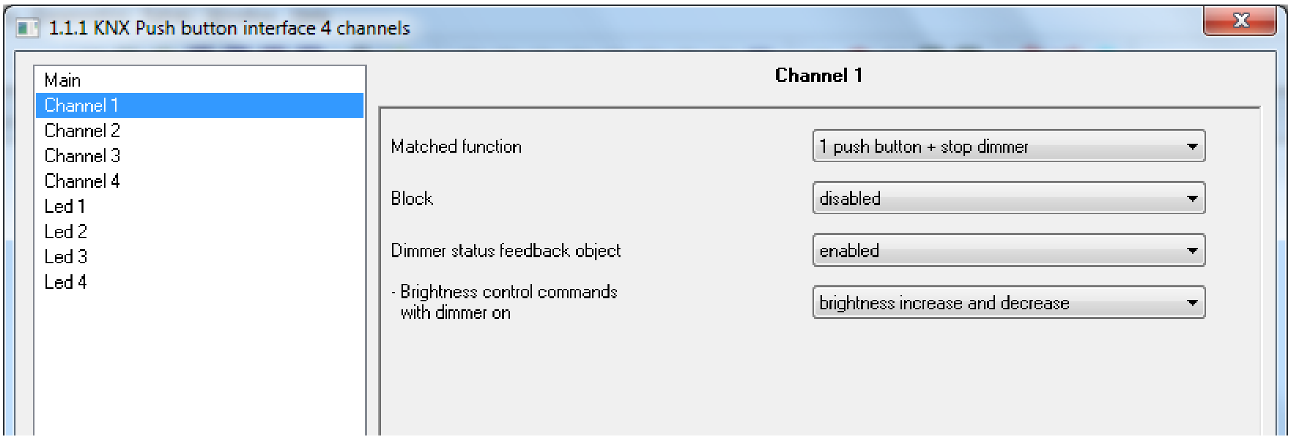
The red arrows highlight the differences between the moments when the telegrams associated with the same long presses are actually sent.

8 *“1 push button + stop dimmer”* function

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

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

Using this type of function, brightness control depends on the so-called brightness control characteristic curve, which varies from actuator to actuator, based on how the manufacturer designed the curve that regulates power, and as a result brightness. This means that the speed with which brightness reaches its maximum and minimum values does not depend on the commands sent by the interface; the latter regulates the brightness itself by halting its increase/decrease on the base of the required value. The communication objects enabled by this function are ***Ch.x - Switch*** (Data Point Type: 1.001 DPT_Switch) and ***Ch.x - Brightness control*** (Data Point Type: 3.007 DPT _Control_Dimming).
The structure of the menu is as follows:



Channel 1

Matched function: 1 push button + stop dimmer
Block: disabled
Dimmer status feedback object: enabled
- Brightness control commands with dimmer on: brightness increase and decrease

Fig. 8.1

8.1 *Parameters*

With normal interface behaviour, the command to be sent is the opposite of the last command sent. This means:

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

This behaviour is modified if the user enables the ***Ch.x - Dimmer status feedback*** communication object (Data Point Type: 1.001 DPT_Switch), via the **“Dimmer status feedback object”** parameter. This parameter may have the following values:

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

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

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

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

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

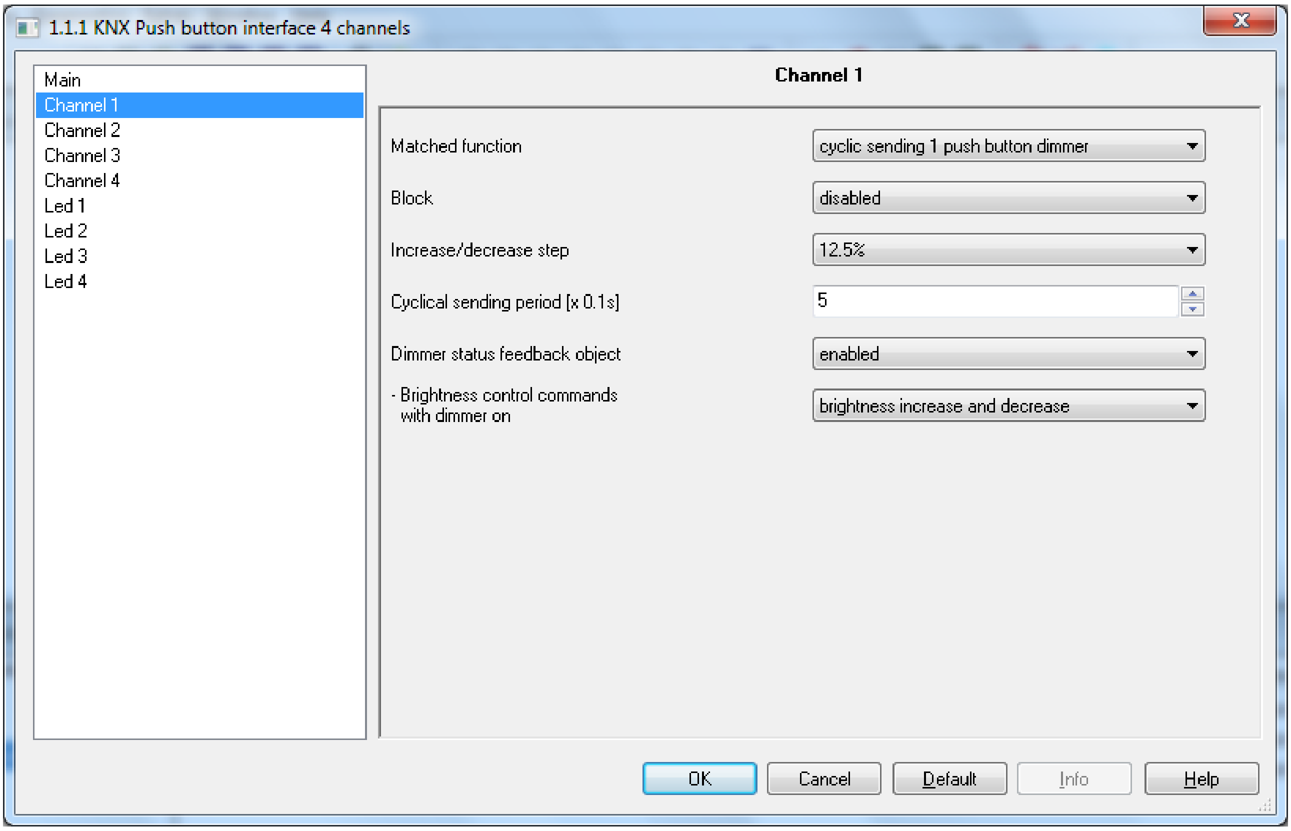
– short operation: if the value of the last of the two events "last sent command" and "dimmer status feedback" is ON, the new command will be an off command; vice versa, if the value of the last of the two events "last sent command" and "dimmer status feedback" is OFF, the new command will be an on command.

9 *“Cyclic sending 1 push-button dimmer”* function

This is used to configure the channel to control a dimmer with a single button, increasing and decreasing dimmer brightness always using the same button, with defined and settable control steps.
As there is only one channel to manage the On/Off and brightness control functions, the operation is managed in the following way: with each activation, the command sent is the opposite to the last one sent. Furthermore, a distinction is made between short operations and long operations:

- a long operation is interpreted as a brightness control command. No telegram is sent when the contact is opened.
- a short operation is interpreted as an on/off command.

Unlike the ***1 push-button + stop dimmer*** function, it is possible to define both the brightness variation steps and the time that must elapse between the sending of one command and another when the long operation is drawn out over time. The sending of the "regulation stop" telegram on contact opening is not therefore necessary, because although the regulation does follow the characteristic power/brightness curve, it is the command sent by the interface 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 control*** (Data Point Type: 3.007 DPT _Control_Dimming).
The structure of the menu is as follows:



Channel 1

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

Fig. 9.1

9.1 Parameters

9.1.1 Increase/decrease step

This 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%

9.1.2 Cyclical sending period [x 0.1s]

If the contact remains closed after recognising the long operation, the device sends the command cyclically until contact opening is detected. The **"Cyclical sending period [x 0.1s]"** parameter is used to set the time that passes between sending one increase/decrease command and another, if the contact remains closed after the recognition of a long operation. When the contact is opened, no telegram is sent; the cyclical sending of the brightness control commands is merely stopped.
The values that can be set for the parameter **"Cyclical sending period [x 0.1s]"** are:

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

To sum up, when a long operation is detected, the device sends the first increase/decrease command with the set percentage and, if the contact remains closed, it sends the command cyclically until it detects the opening of the contact.

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

- 0.5 seconds after the detection of the contact closure, a long operation is detected and so the first 12.5% brightness increase/decrease telegram is sent
- from this moment, for every 0.3 seconds that contact remains closed, the device will send the 12.5% brightness increase/decrease command again and again until opening is detected
- when the contact is opened, no telegram is sent; the cyclical sending is merely stopped

9.1.3 Dimmer status feedback object

As for the ***1 push-button + stop dimmer*** function, it is possible to enable the dimmer status feedback object by changing the behaviour of the switching and control commands as described in the "1 push-button + stop dimmer" function paragraph.

The parameter used to enable the feedback object is **"Dimmer status feedback object"** which can have the following values

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

If **enable** is selected, the **"Brightness control commands with dimmer on"** parameter is visible along with the ***Ch.x - Dimmer status feedback*** communication object (Data Point Type: 1.001 DPT_Switch), which is used to receive the status feedback from the controlled dimmer actuator.
The parameter **"Brightness control commands with dimmer on"** can have the following values:

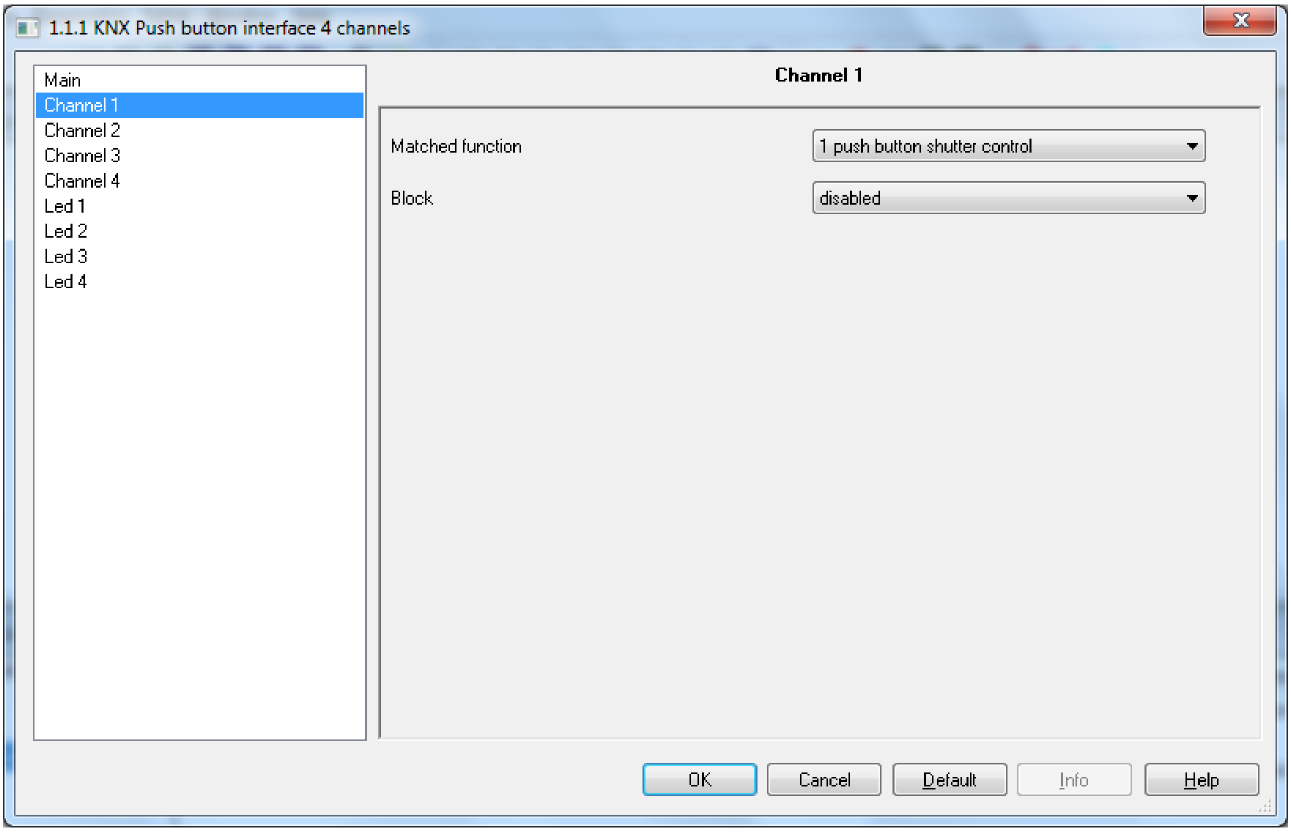
- only brightness increase
- only brightness decrease
- **brightness increase and decrease (default value)**

10 “*1 push-button shutter control*” function

This is used to configure the channel to control a shutter with a single button, regulating the upward and downward travel of the shutter and, depending on the device version, controlling louvres opening/closing.
As only one 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. There is a difference between short and long operations:

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

The communication objects enabled by this function are ***Ch.x - Shutter movement*** (Data Point Type: 1.008 DPT_UpDown), ***Ch.x - Louvre stop/adjustment*** (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:


1.1.1 KNX Push button interface 4 channels

Channel 1

Matched function: 1 push button shutter control

Block: disabled

Fig. 10.1

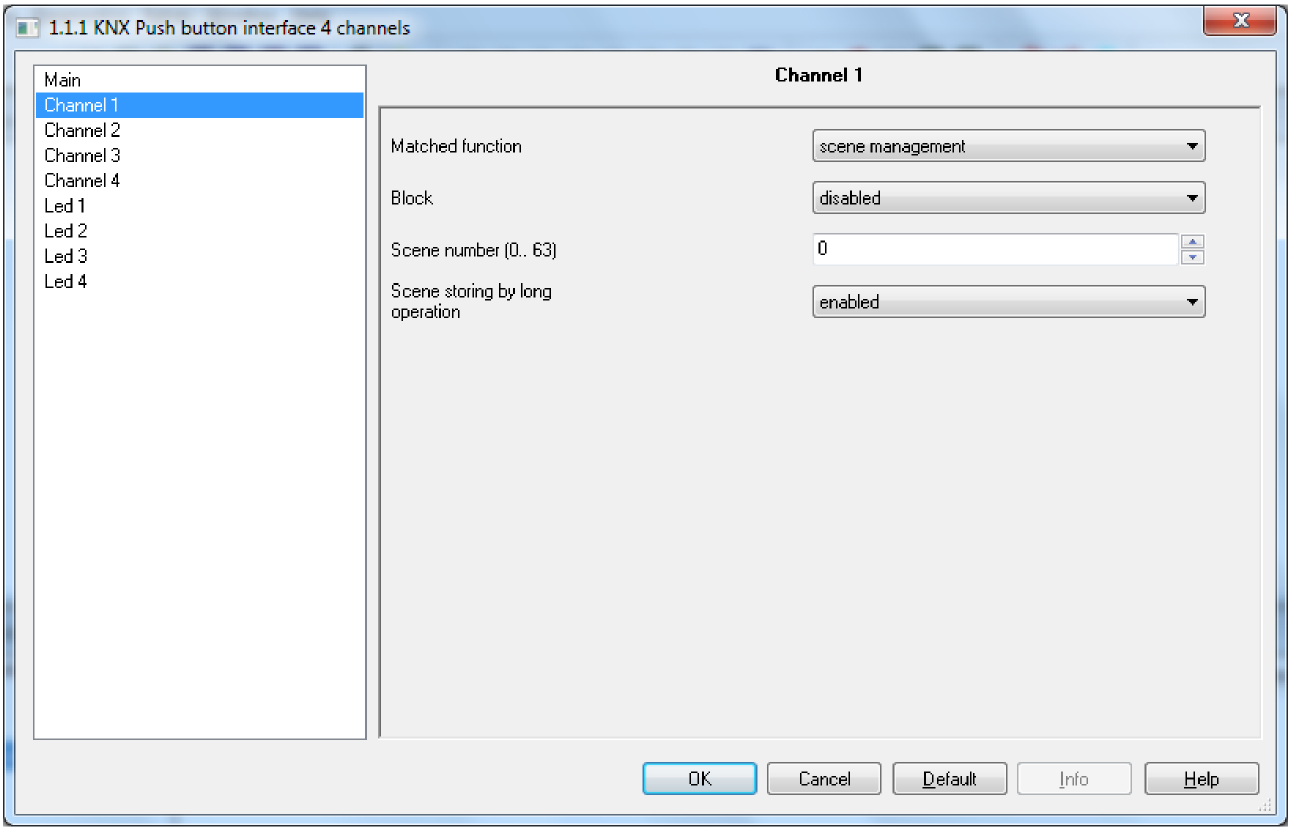
No new parameters are enabled with this function.

11 “*Scene management*” function

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

– a long operation is interpreted as a scene storing command.
– a short operation is interpreted as 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).
When a long operation is recognised, to provide the user with visual confirmation that the scene storing command was sent, night-time signalling is deactivated for a moment (blink). If night lighting is disabled, the light signal is activated for a moment (blink) when a long operation is detected; this effect takes priority over all the light effects activated by the BUS (see the “Led X menu” paragraph).
The structure of the menu is as follows:



Matched function: scene management
Block: disabled
Scene number (0.. 63): 0
Scene storing by long operation: enabled

Fig. 11.1

11.1 Parameters

11.1.1 Scene number (0.. 63)

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

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

11.1.2 Scene storing by long operation

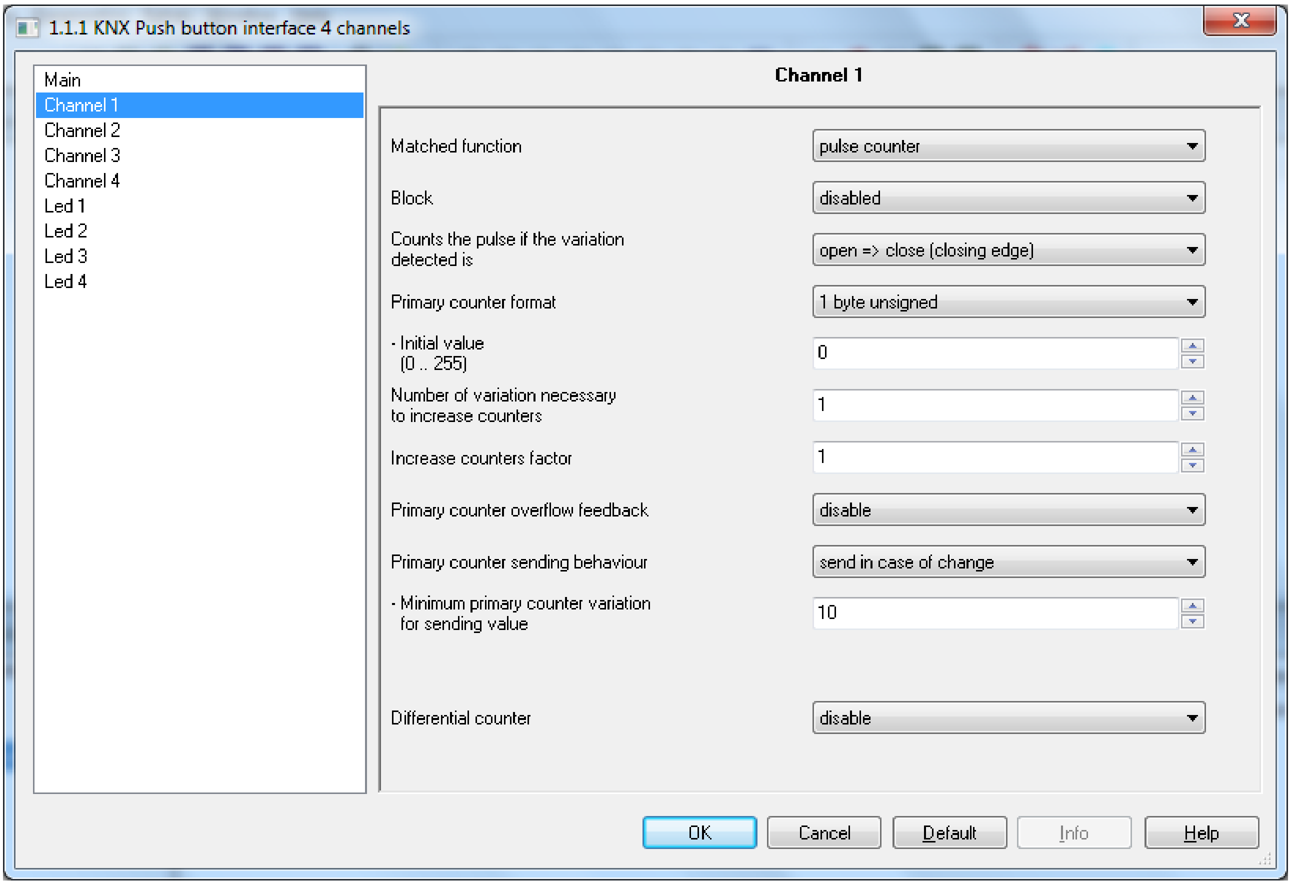
This parameter is used to enable the sending of a scene storage command when a long operation is recognised. The values that can be set are:

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

Only if **enabled** is selected, the device will send the scene storing command when a long operation is detected; if **disabled** is selected, a long operation is not recognised and only causes the sending of the scene execution command (like the short operation). Regardless of the value set for the parameter above, it is possible to indirectly generate the sending of the scene storing command following the arrival of a BUS telegram on the object ***Ch.x - Scene storing trigger*** (both with a value of “1” and with a value of “0”); each time the device receives a telegram on that object, it must immediately send a scene memorisation telegram.

12 ”*Pulse counter*” function

Used to configure the channel for counting the number of contact status variations (edges) by setting the parameters that characterise the count.
The structure of the menu is as follows:



Channel 1

Matched function	pulse counter
Block	disabled
Counts the pulse if the variation detected is	open => close (closing edge)
Primary counter format	1 byte unsigned
- Initial value (0 .. 255)	0
Number of variation necessary to increase counters	1
Increase counters factor	1
Primary counter overflow feedback	disable
Primary counter sending behaviour	send in case of change
- Minimum primary counter variation for sending value	10
Differential counter	disable

Fig. 12.1

12.1 Parameters

12.1.1 Counts the pulse if the variation detected is

In this mode, each contact can count the incoming pulses. The count is based on the detection of the edges of the input signal. There are 2 edges that can be detected: contact closure and opening. The **“Counts the pulse if the variation detected is”** parameter is used to set the type of contact status variation to be considered for increasing the count of the primary and differential counters. The values that can be set are:

- **open => close (closing edge)** **(default value)**
- close => open (opening edge)
- both

By selecting **open => close (closing edge)**, only the variation from open contact to closed contact (closing edge) will be considered by the device as a pulse, so it is this variation that produces an increase in the count value; the opposite status variation will have no effect.
By selecting **close => open (opening edge)**, only the variation from closed contact to open contact (opening edge) will be considered by the device as a pulse, so it is this variation that produces an increase in the count value; the opposite status variation will have no effect.
By selecting **both**, the variation from closed contact to open contact (opening edge) and the variation from open contact to closed contact (closing edge) will both be considered by the device as a pulse, producing an increase in the count value.

12.1.2 Primary counter format

The primary counter used for the pulse count must be of a sufficient capacity to count the maximum required number of pulses. With the **“Primary counter format”** parameter, you can define the size and code of the communication object used to communicate the value of the primary counter. The values that can be set are:

- **1 byte value without sign** **(default value)**
- 1 byte signed value
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value

Depending on the value set for this item, the values that can be set for the **“Initial value”** parameter will be different.

12.1.3 Initial value

This parameter is used to set the initial value of the primary counter. When the primary counter reaches its overflow - or maximum value - point (or minimum value, depending on the counter increase factor set), it is re-initialised to the set initial value.
Depending on the value set for the **Primary counter format** parameter, the values that can be set for this item will be different.

- If the format of the primary counter is **1 byte value without sign**, the ***Ch.x - Primary counter*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the primary counter is **1 byte value with sign**, the ***Ch.x - Primary counter*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

– If the format of the primary counter is **2 bytes value without sign**, the ***Ch.x - Primary counter*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the primary counter is **2 bytes value with sign**, the ***Ch.x - Primary counter*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

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

– If the format of the primary counter is **4 bytes value without sign**, the ***Ch.x - Primary counter*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

– If the format of the primary counter is **4 bytes value with sign**, the ***Ch.x - Primary counter*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the above parameter are:

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

12.1.4 Number of variations necessary to increase counters

This parameter is used to set the number of edges necessary to increase the counters (both primary and differential). This means that, if a value of 2 is set (for example), two edges are needed to increase the value of the counters (both primary and differential). The values that can be set are:

- from **1 (default value)** to 32767 with step of 1

12.1.5 Increase counters factor

This parameter is used to establish by how many units the counters (both primary and differential) must increase when counter increase conditions occur (number of edges detected equal to the number of variations needed for a counter increase). This means that, if a value of 2 is set (for example), the counters (both primary and differential) will be increased by two units every time increase conditions occur.

- from - 32768 to +32767 with step of 1, **1 (default value)**

if a negative value is selected, the counters are decreased and the overflow value of the primary counter is the minimum value of the range defined by the selected format.

To better understand the meaning of the **“Number of variations necessary to increase counters”** and **“Increase counters factor”** parameters, let's consider the case where the increase factor is 2 and the number of variations necessary to increase the counter is 5; with this configuration, the value of the counters (both primary and differential) will be increased by two units for every five count increase edges detected. Of course, the counter value is not modified until 5 increase edges are detected.

12.1.6 Overflow feedback

This parameter is used to enable the display - and hence the use - of the communication objects that indicate when the primary counter has exceeded its maximum (or minimum) value. The values that can be set are:

- **disabled (default value)**
- enable object of 1 bit
- enable objects of 1 bit and 1 byte

Selecting a value other than **disabled**, the ***Ch.x – Primary counter bit overflow*** communication object is made visible (Data Point Type: 1.002 DPT_Bool) via which the device indicates the overflow of the primary counter. When the overflow occurs, a value of "1" is sent; a value of "0" is never sent.
Selecting **enable objects of 1 bit and 1 byte**, the ***Ch.x – Primary counter byte overflow*** communication object is made visible (Data Point Type: 5.010 DPT_Value_1_Ucount) via which the device indicates the overflow of the primary counter. When the overflow occurs, the value defined by the new **"Send the value with primary counter overflow"** parameter is sent. This parameter may assume the following values:

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

Once the maximum (or minimum) value has been reached, the primary counter restarts from the value set in **"Initial value"**.
If the value set in **"Increase counters factor"** is greater than 1, the number of units needed to trigger the overflow may be less than the increase factor; as the primary counter is circular, it is re-initialised when the overflow value is exceeded and the supplementary units are calculated. Example: increase counters factor of *7*, the counter is *1 unsigned byte* and the initial value is 50. If the counter value is 253 and the counter increase condition is detected, the overflow telegram is sent and the new counter value is 54 (the initial value is also counted).

12.1.7 Primary counter sending behaviour

This parameter is used to define the conditions for sending the current value of the primary counter. The values that can be set are:

- send on demand only
- **send in case of change (default value)**
- send periodically
- send on change and periodically

Selecting a value other than **send on demand only**, the ***Ch.x – Primary counter sending trigger*** communication object is made visible (Data Point Type: 1.017 DPT_Trigger). Selecting **send in case of change** or **send on change and periodically**, the **"Minimum primary counter variation for sending value"** parameter will be visible, whereas by selecting **send periodically** or **send on change and periodically** the **"Primary counter sending period (seconds)"** parameter will be visible.
Selecting the value **send on demand only**, no new parameter will be enabled because the primary counter value is not sent spontaneously by the device; only in the case of a status read request will it send the user a telegram in response to the command received, giving information about the current value of the primary counter.
If the primary counter sending condition is different from **on demand only**, there is the possibility of indirectly generating the sending of the current counter value following receipt of a BUS telegram on the ***Ch.x - Primary counter sending trigger*** object (with both a value of "1" and a value of "0"). Every time the device receives a telegram on that object, it must immediately send the current value of the primary counter. After a BUS voltage recovery, the value of the primary counter should be sent in order to update any connected devices.

12.1.8 Minimum primary counter variation for sending value

This parameter is visible if the primary counter value is sent with a change. It is used to define the minimum count variation (in relation to the last value sent) that causes the new measured value to be spontaneously sent. The values that can be set are:

- from 1 to 100 with steps of 1, **10 (default value)**

12.1.9 Primary counter sending period (seconds)

This parameter is visible if the primary counter value is sent periodically. It is used to define the period with which telegrams indicating the current primary counter value are spontaneously sent. The values that can be set are:

- from 1 to 255 with steps of 1, **15 (default value)**

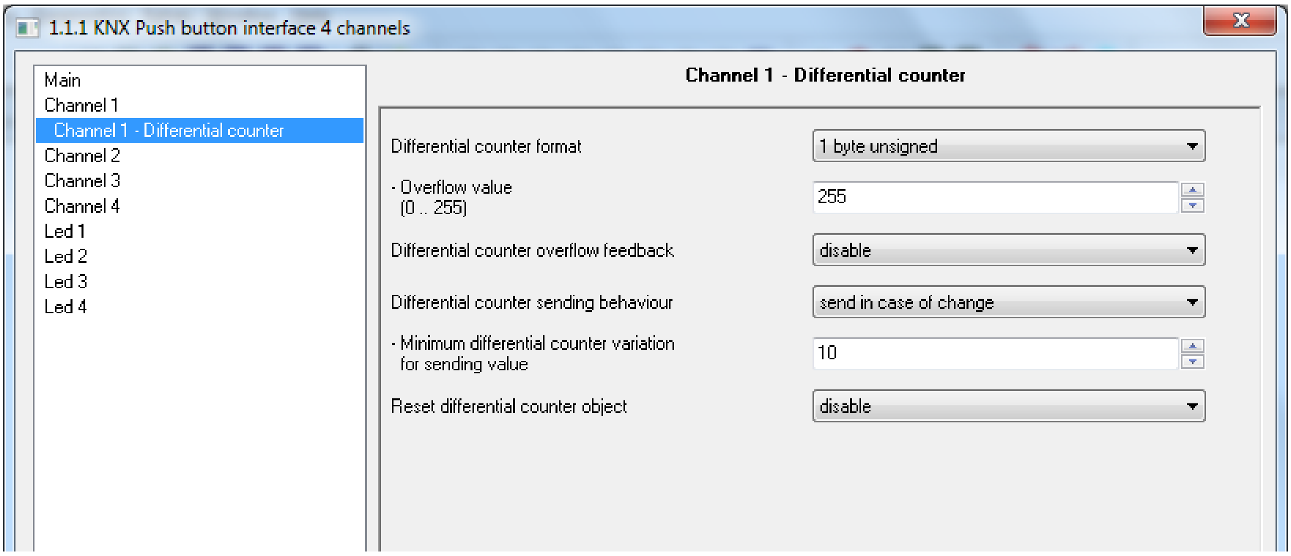
In the event of a BUS voltage failure, the primary counter value must be saved in a non-volatile memory and restored when the BUS voltage is recovered.

The **“Differential counter”** parameter is used to enable the display - and hence the use - of the ***Ch.x - Differential counter*** communication object, and to view the ***Channel x - Differential counter*** configuration menu (see “Channel x - Differential counter menu” paragraph).
Unlike the primary counter, the differential counter: can be reset, can indicate an overflow value different from the maximum coded value, and has an initial value of 0. The two counters both have: a counter increase edge, an increase factor, and a number of variations for counter increase.
The values that can be set are:

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

12.2 “Channel x – Differential counter” parameters

If ***enabled*** is set for the ***Differential counter*** item of the general ***Channel x*** menu, you will see the sub-menu that we're going to analyse: ***Channel x – Differential counter***.
The menu appears as shown in fig. 12.2.


1.1.1 KNX Push button interface 4 channels
Main
Channel 1
Channel 1 - Differential counter
Channel 2
Channel 3
Channel 4
Led 1
Led 2
Led 3
Led 4

Channel 1 - Differential counter

Differential counter format: 1 byte unsigned

- Overflow value (0 .. 255): 255

Differential counter overflow feedback: disable

Differential counter sending behaviour: send in case of change

- Minimum differential counter variation for sending value: 10

Reset differential counter object: disable

Fig. 12.2

12.3 Parameters

12.3.1 Primary counter sending period (seconds)

The differential counter used for the pulse count must be of a sufficient capacity to count the maximum required number of pulses. With the **"Differential counter format"** parameter, it is possible to define the size and code of the communication object used to communicate the value of the primary counter. The values that can be set are:

- **1 byte value without sign (default value)**
- 1 byte signed value
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value

The initial value is always 0, regardless of the format selected.
Depending on the value set for this item, the values that can be set for the **"Overflow value"** parameter will be different.

12.3.2 Overflow value

The **"Overflow value"** parameter is used to set the maximum value of the differential counter; in fact, unlike the primary counter, it is possible to set the maximum count value - i.e. the value beyond which the differential counter is in an overflow condition.
Depending on the value set for the **Differential counter format** parameter, the values that can be set for this item will be different.

- If the format of the differential counter is **1 byte unsigned**, the ***Ch.x - Differential counter*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the differential counter is **1 byte signed**, the ***Ch.x - Differential counter*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

- If the format of the differential counter is **2 byte unsigned**, the ***Ch.x - Differential counter*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the differential counter is **2 byte signed**, the ***Ch.x - Differential counter*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

  - from -32768 to **+32767 (default value)** with steps of 1

- If the format of the differential counter is **4 bytes unsigned**, the ***Ch.x - Differential counter*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

- If the format of the differential counter is **4 bytes signed**, the ***Ch.x - Differential counter*** communication object will be visible (Data Point Type: 13.001 DPT_Value_4_Count) and the values that can be set for the above parameter are:

  - from -2147483648 to **2147483647 (default value)** with steps of 1

12.3.3 Differential counter overflow feedback

This parameter is used to enable the display - and hence the use - of the communication objects that indicate when the differential counter has exceeded its maximum value. The values that can be set are:

- **disabled (default value)**
- enable object of 1 bit
- enable objects of 1 bit and 1 byte

Selecting a value other than **disabled**, the ***Ch.x – Differential counter bit overflow*** communication object is made visible (Data Point Type: 1.002 DPT_Bool) via which the device indicates the overflow of the differential counter. When the overflow occurs, a value of "1" is sent; a value of "0" is never sent.
Selecting **enable objects of 1 bit and 1 byte**, the ***Ch.x – Differential counter byte overflow*** communication object is made visible (Data Point Type: 5.010 DPT_Value_1_Ucount) via which the device indicates the overflow of the differential counter. When the overflow occurs, the value defined by the new **"Send the value with differential counter overflow"** parameter is sent. This parameter may assume the following values:

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

Once the maximum value has been reached, the differential counter restarts from 0.

12.3.4 Increase counters factor

If the value set in **"Increase counters factor"** of the **Channel x** menu is greater than 1, the number of units needed to trigger the overflow may be less than the increase factor; as the differential counter is circular, it is re-initialised when the overflow value is exceeded and the supplementary units are calculated. Example: increase counters factor of *7* and the counter is *1 byte unsigned*. If the differential counter value is 253 and the counter increase condition is detected, the overflow telegram is sent and the new counter value is 4 (the initial value is also counted).

12.3.5 Differential counter sending behaviour

This parameter **"Differential counter sending behaviour"** is used to define the conditions for sending the current value of the differential counter. The values that can be set are:

- **send on demand only (default value)**
- send on change
- send periodically
- send on change and periodically

Selecting a value other than **send on demand only**, the ***Ch.x – Differential counter sending trigger*** communication object is made visible (Data Point Type: 1.017 DPT_Trigger). Selecting **send in case of change** or **send on change and periodically**, the **"Minimum differential counter variation for sending value"** parameter will be visible, whereas by selecting **send periodically** or **send on change and periodically** the **"Differential counter sending period"** parameter will be visible.
Selecting the value **send on demand only**, no new parameter will be enabled because the differential counter value is not sent spontaneously by the device; only in the case of a status read request will it send the user a telegram in response to the command received, giving information about the current value of the differential counter.
If the differential counter sending condition is different from **on demand only** , there is the possibility of indirectly generating the sending of the current counter value following receipt of a BUS telegram on the

Ch.x - Differential counter sending trigger object (with both a value of “1” and a value of “0”). Every time the device receives a telegram on that object, it must immediately send the current value of the differential counter. After a BUS voltage recovery, the value of the differential counter should be sent in order to update any connected devices.

12.3.6 Minimum differential counter variation for sending value

This parameter is visible if the differential counter value is sent with a change. It is used to define the minimum count variation (in relation to the last value sent) that causes the new measured value to be spontaneously sent. The values that can be set are:

- from 1 to 100 with steps of 1, **10 (default value)**

12.3.7 Primary counter sending period (seconds)

This parameter is visible if the differential counter value is sent periodically. It is used to define the period with which telegrams indicating the current differential counter value are spontaneously sent. The values that can be set are:

- from 1 to 255 with steps of 1, **15 (default value)**

12.3.8 Reset differential counter object

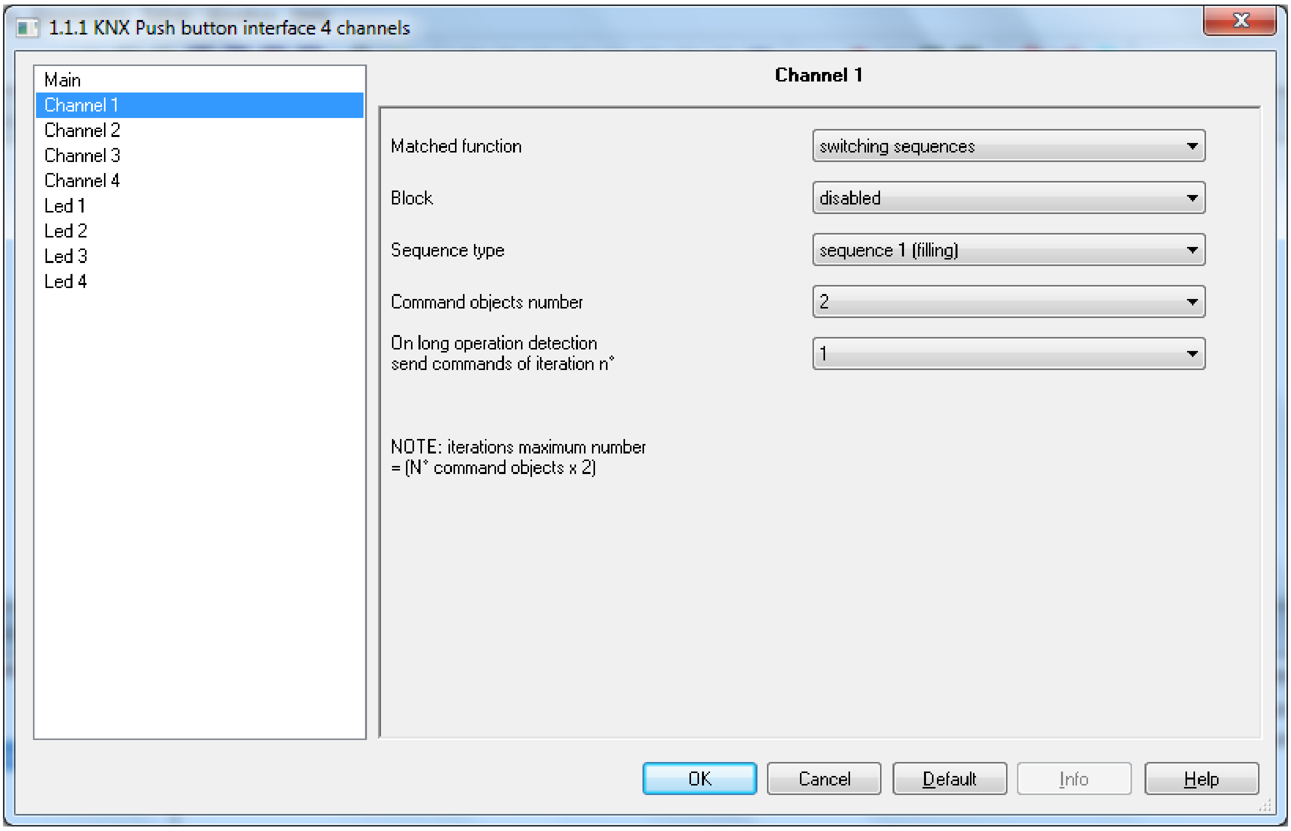
This parameter is used to enable the display - and hence the use - of the ***Ch.x - Reset differential counter*** communication object (Data Point Type: 1.001 DPT_Switch), to receive - via BUS - the differential counter reset command for resetting the value of the differential counter.
The values that can be set are:

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

Selecting **enable**, the ***Ch.x - Differential counter reset*** communication object is made visible, via which the device receives the differential counter reset command. If a value of “1” is received, the differential counter is re-initialised at 0; a value of “0” has no effect.

13 *“Switching sequences”* function

Used to send a sequence of commands following the detection of a specific operation.
The structure of the menu is as follows:



1.1.1 KNX Push button interface 4 channels

Channel 1

Matched function: switching sequences

Block: disabled

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 Parameters

13.1.1 Number of objects to send (Command objects number)

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

- from **2 (default value)** to 8 with steps of 1

13.1.2 Sequence type

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

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

Sequence 1 (filling): each time a closure (edge) is detected, the device sends - on the enabled communication objects - a sequence that follows the filling progress. 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 restart from the beginning

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

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

Edge no.	Value sent on ***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 restart from the beginning

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

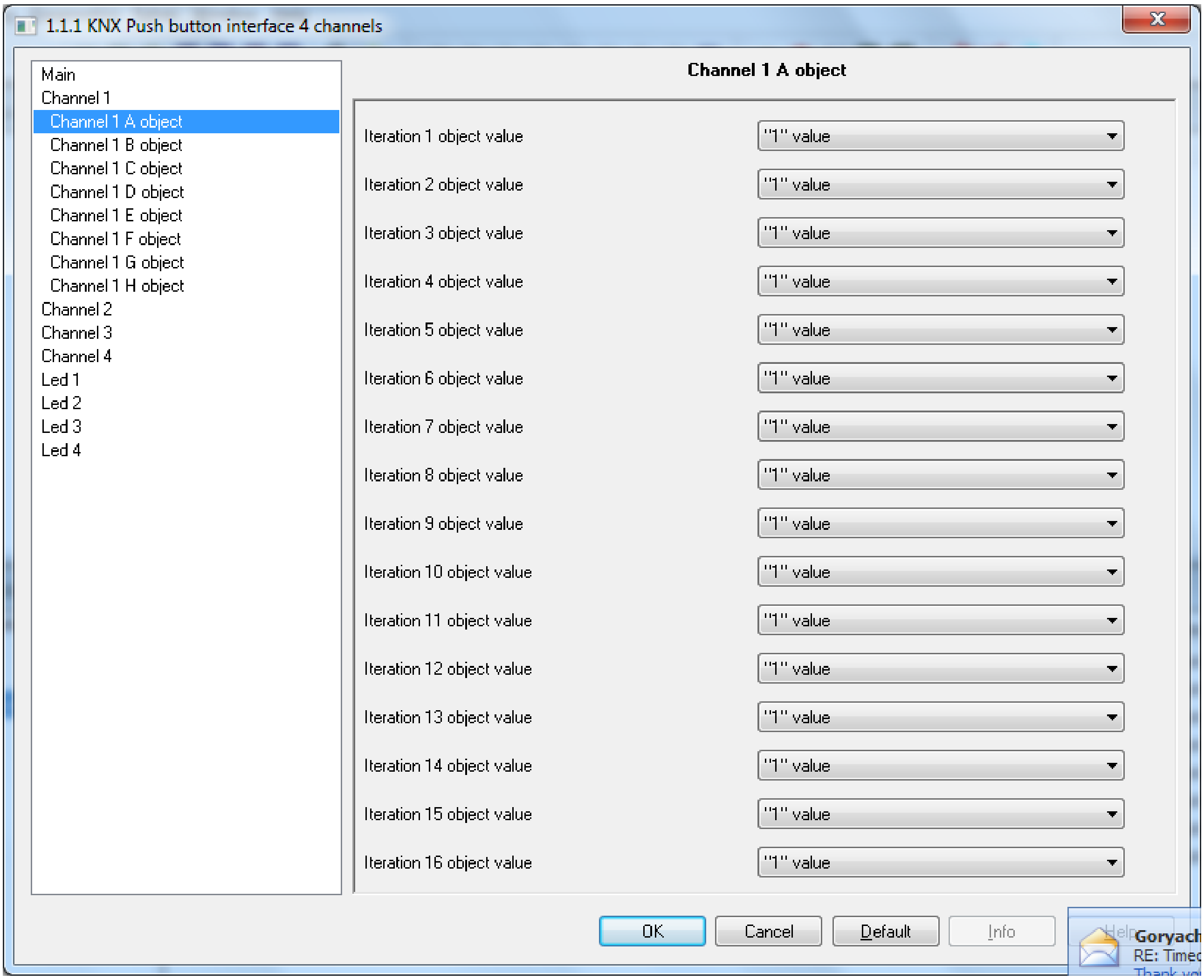
Sequence 3 (free) allows the user to directly set the value for each command for each set edge; this setting enables the parameter **“Number of sequence iterations”** and the configuration menu **z object channel x** (one for each enabled command). The parameter **“Number of sequence iterations”** allows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

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

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

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

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



Channel 1 A object

Iteration 1 object value	"1" value
Iteration 2 object value	"1" value
Iteration 3 object value	"1" value
Iteration 4 object value	"1" value
Iteration 5 object value	"1" value
Iteration 6 object value	"1" value
Iteration 7 object value	"1" value
Iteration 8 object value	"1" value
Iteration 9 object value	"1" value
Iteration 10 object value	"1" value
Iteration 11 object value	"1" value
Iteration 12 object value	"1" value
Iteration 13 object value	"1" value
Iteration 14 object value	"1" value
Iteration 15 object value	"1" value
Iteration 16 object value	"1" value

Fig. 13.2

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

- from 1 to 256 with steps of 1, **1 (default value)**

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

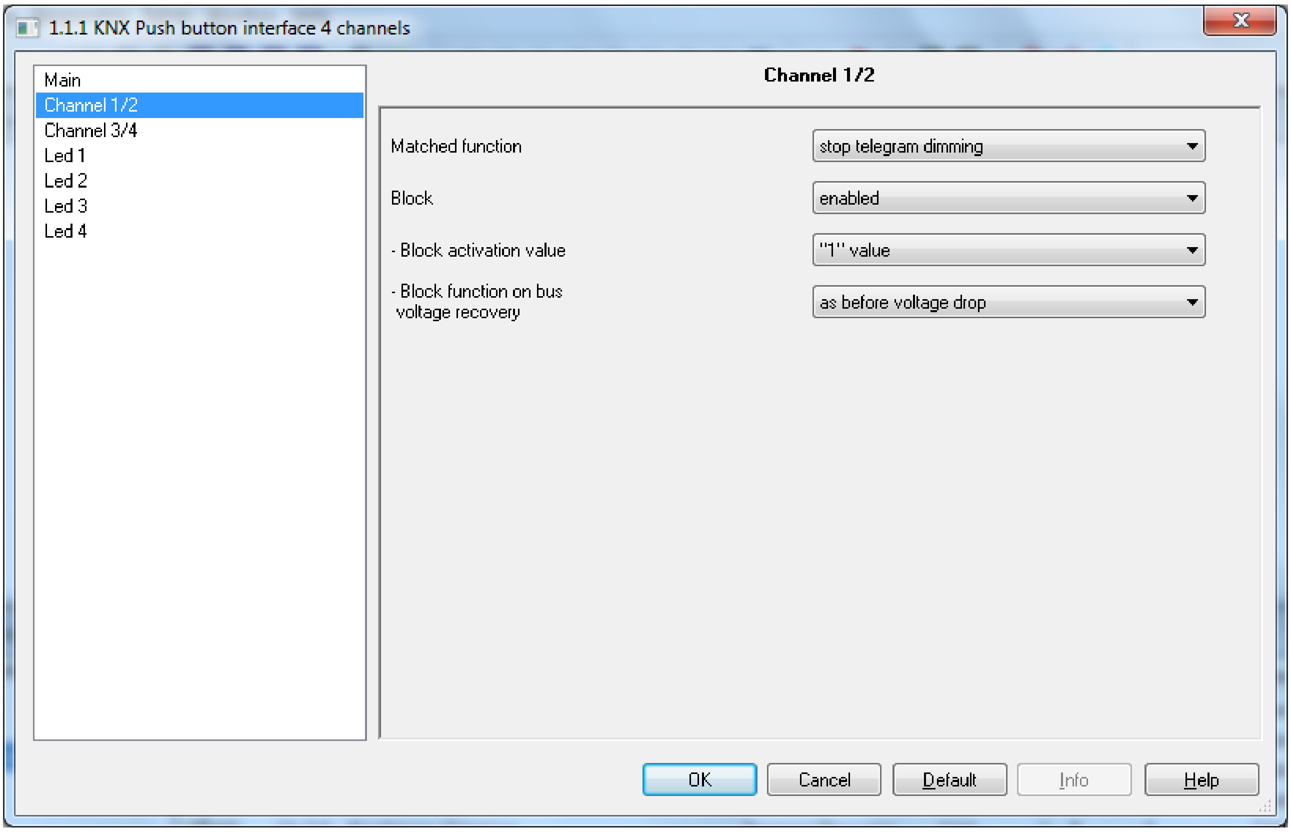
Edge no.	Value sent on ***Ch.x - C sequence***	Value sent on ***Ch.x - B sequence***	Value sent on ***Ch.x - A sequence***
1st edge	0	0	1
2nd edge	0	1	0
3rd edge	0	1	1
4th edge	1	0	0
5th edge	1	0	1
6th edge	1	1	0
7th edge	1	1	1
8th edge	0	0	0

"Sum" sequence

Once a long operation has been detected and the sequence relating to the set iteration has been sent, then when the next short operation is detected, the sequence relating to the iteration immediately after the one associated with the long operation will be sent (in the example given here, the sequence associated with iteration no. 4 will be sent).
To sum up, the value set for the **"On long operation detection, send commands of iteration n°"** parameter defines both the sequence to be sent and the value with which to initialise the iterations counter when a long 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 maximum one.

14 *"Channel x/y"* menu (associated channels)

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



Channel 1/2

Matched function: stop telegram dimming

Block: enabled

- Block activation value: "1" value

- Block function on bus voltage recovery: as before voltage drop

Fig. 14.1

14.1 Parameters

14.1.1 Matched function

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

- **stop telegram dimming (default value)**
  (See the "stop telegram dimming" function paragraph)
- cyclic telegram dimming
  (See the "cyclic telegram dimming" function paragraph)
- shutter control
  (See the "shutter control" function paragraph)

14.1.2 Block

To inhibit the channel for the sending of commands associated with contact closure/opening, the block function must be activated. This function in fact inhibits the detection of contact closure/opening, thereby preventing the device from sending the telegrams associated with these events on the BUS. The parameter for enabling the function is the **"Block"** parameter, that can take the following values:

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

If **enabled** is selected, the following parameters will be visualised: **"Block activation value"** and "**Block function on BUS voltage recovery**, along with the ***Ch.x/y - Block*** communication item (Data Point Type: 1.002 DPT_Enable) with which you can activate the function via the BUS command.
The parameter **"Block activation value"** makes it possible to set which logic value the bit received via BUS telegram should assume to activate the block function; the values that can be set are:

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

14.1.3 Block function on BUS voltage recovery

With the **"Block function on BUS voltage recovery (=Block on BUS tension recovery function)"** parameter, you can set the status of the block function when the BUS voltage is restored. the values that can be set are:

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

15 *“STOP telegram dimming”* function

This is used to configure combined channels to control a dimmer with two push-buttons, using a push-button to control the switching on and increase of dimmer brightness and the other to control the switching-off and the decrease in brightness.

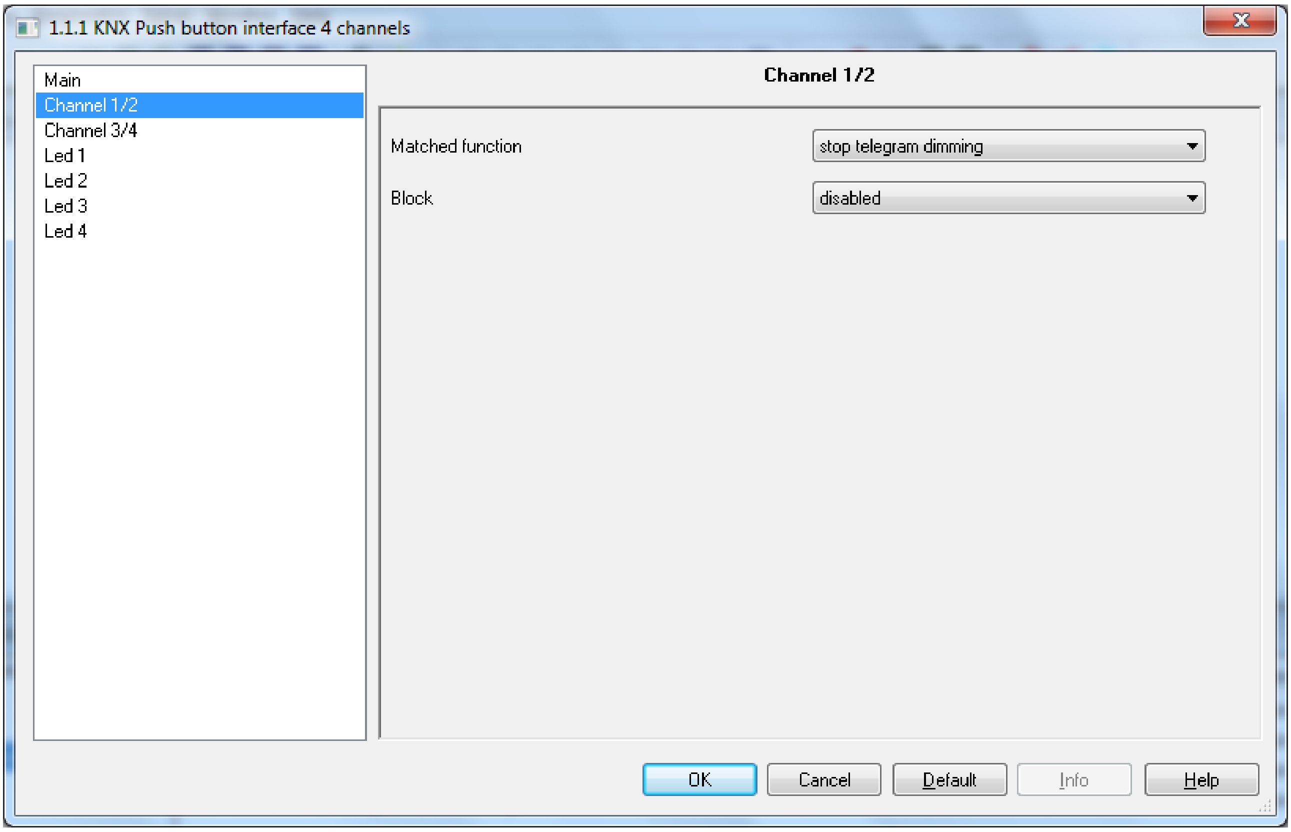
Also in this case there are two channels that manage the function, however a distinction is made between short and long operations:

– a long operation is interpreted as a brightness control command. If this type of operation is recognised on channel x, the device will send a command to increase brightness; on the other hand, if the operation is recognised on channel y, the device will send a command to decrease brightness. In both cases, when the contact is opened, an adjustment stop telegram is sent to stop the brightness increase/decrease operation for the dimmer and to fix the brightness value reached at the moment the stop control command was received.

– a short operation is interpreted as an on/off command. If this type of operation is recognised on channel x, the device will send a switch-on command; on the other hand, if the operation is recognised on channel y, the device will send a switch-off command.

Using this type of function, brightness control depends on the so-called brightness control 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 control*** (Data Point Type: 3.007 DPT _Control_Dimming).

The structure of the menu is as follows:



Matched function: stop telegram dimming

Block: disabled

Fig. 15.1

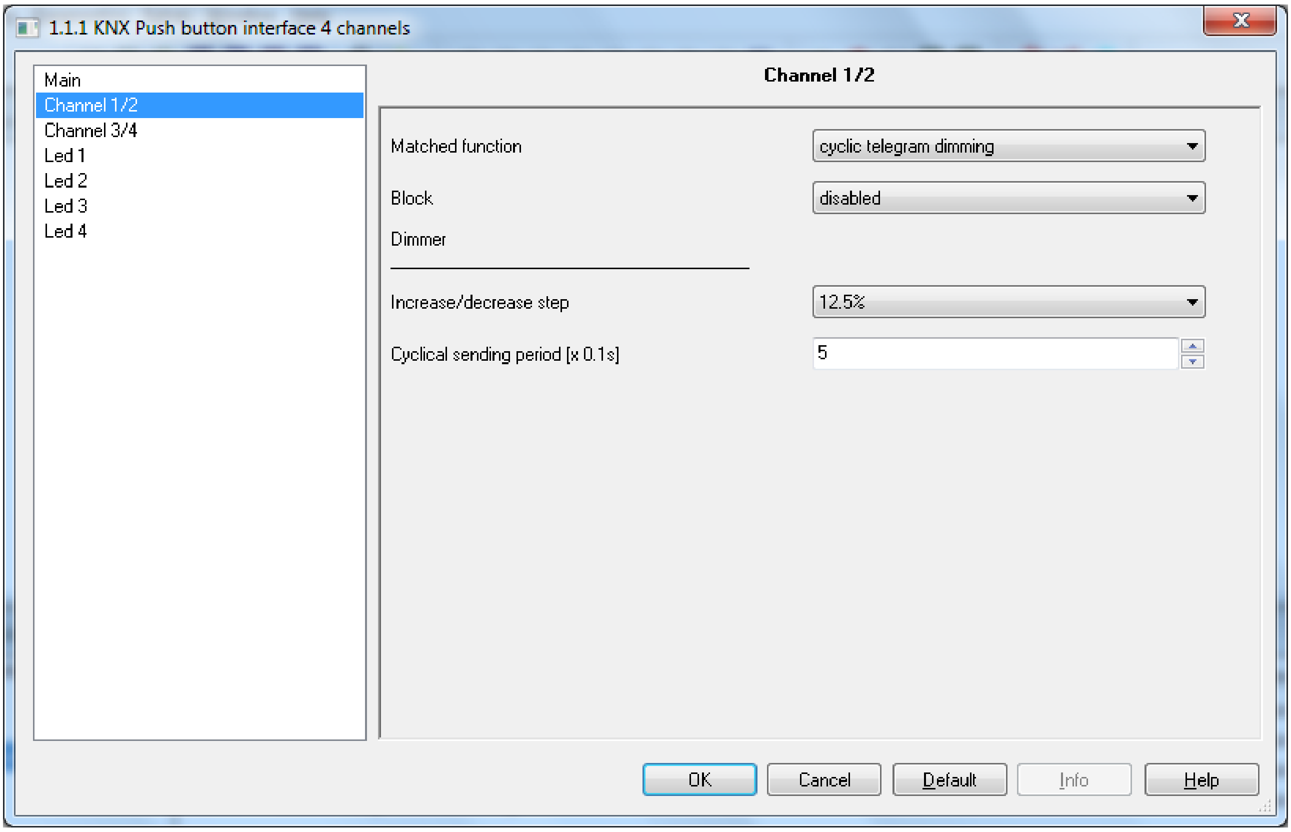
No new parameters are enabled with this function.

16 *”Cyclic telegram dimming”* function

This is used to configure combined channels to control a dimmer with two push-buttons, using a push-button to control the switching on and increase of dimmer brightness and the other to control the switching-off and the decrease in brightness.
Also in this case there are two channels that manage the function, however a distinction is made between short and long operations:

- a long operation is interpreted as a brightness control command. If this type of operation is recognised on channel x, the device will send a command to increase brightness; on the other hand, if the operation is recognised on channel y, the device will send a command to decrease brightness. No telegram is sent when the contact is opened.
- a short operation is interpreted as an on/off command. If this type of operation is recognised on channel x, the device will send a switch-on command; on the other hand, if the operation is recognised on channel y, the device will send a switch-off command.

Unlike the function ***dimming with telegram + STOP***, it is possible to define both the brightness variation steps of the brightness increase/decrease commands as well as the time that must elapse between the sending of one command and another when the push-button remains pressed. 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 interface 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 control*** (Data Point Type: 3.007 DPT _Control_Dimming).
The structure of the menu is as follows:



Channel 1/2

Matched function: cyclic telegram dimming

Block: disabled

Dimmer

Increase/decrease step: 12.5%

Cyclical sending period [x 0.1s]: 5

Fig. 16.1

16.1 Parameters

16.1.1 Increase/decrease step

This 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 the contact remains closed after recognising the long operation, the device sends the command cyclically until contact opening is detected. The **"Cyclical sending period [x 0.1s]"** parameter is used to set the time that must pass between the sending of one increase/decrease command and another, if the contact remains closed after the recognition of a long operation. When the contact is opened, no telegram is sent; the cyclical sending of the brightness control commands is merely stopped.
The values that can be set for the parameter **"Cyclical sending period [x 0.1s]"** are:

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

To sum up, when a long operation is detected, the device sends the first increase/decrease command with the set percentage and, if the contact remains closed, it sends the command cyclically until it detects the opening of the contact.

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

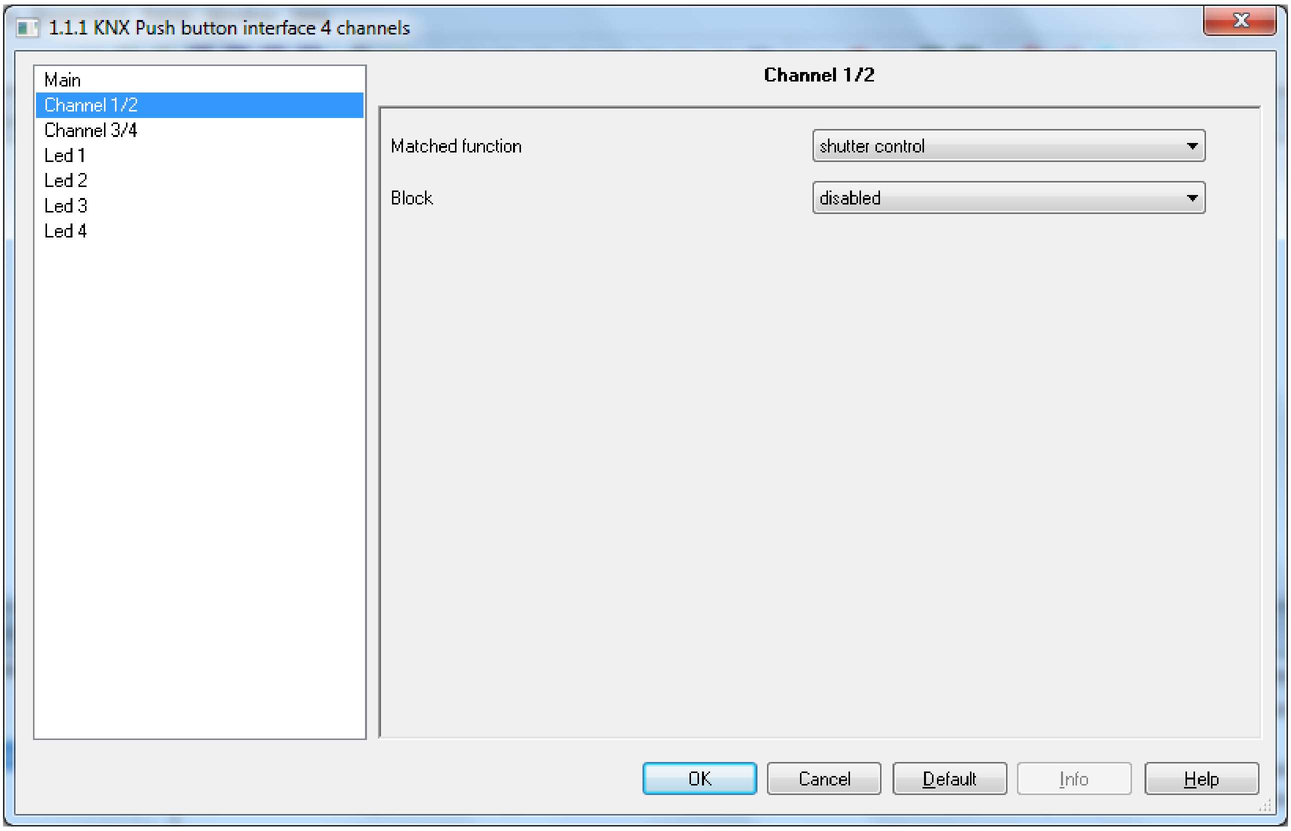
- 0.5 seconds after the detection of the contact closure, a long operation is detected and so the first 12.5% brightness increase/decrease telegram is sent
- from this moment, for every 0.3 seconds that contact remains closed, the device will send the 12.5% brightness increase/decrease command again and again until opening is detected
- when the contact is opened, no telegram is sent; the cyclical sending is merely stopped

17 ”*Shutter control*” function

This is used to configure the channel to control a shutter with two push-buttons, regulating the upward and downward travel of the shutter and, depending on the device version, controlling louvres opening/closing.
For sending up/down telegrams and louvres adjustment telegrams.
Also in this case there are two channels that manage the function, however a distinction is made between short and long operations:

- a long operation is interpreted as an up/down command. If this type of operation is recognised on channel x, the device will send an up command; otherwise, the device will send a down command. When the contact opens, the device does not perform any action.
- a short operation is interpreted as a louvre control command. If this type of operation is recognised on channel x, the device will send an open louvre regulation command; otherwise, the device will send a closing louvres adjustment command. If the shutter is moving, the louvre control command will only stop the shutter up/down movement; the louvres control is carried out when the shutter is stationary.

The communication objects enabled by this function are ***Ch.x - Shutter movement*** (Data Point Type: 1.008 DPT_UpDown) and ***Ch.x - Louvre stop/adjustment*** (Data Point Type: 1.007 DPT_Step).
The structure of the menu is as follows:



1.1.1 KNX Push button interface 4 channels

Channel 1/2

Matched function: shutter control

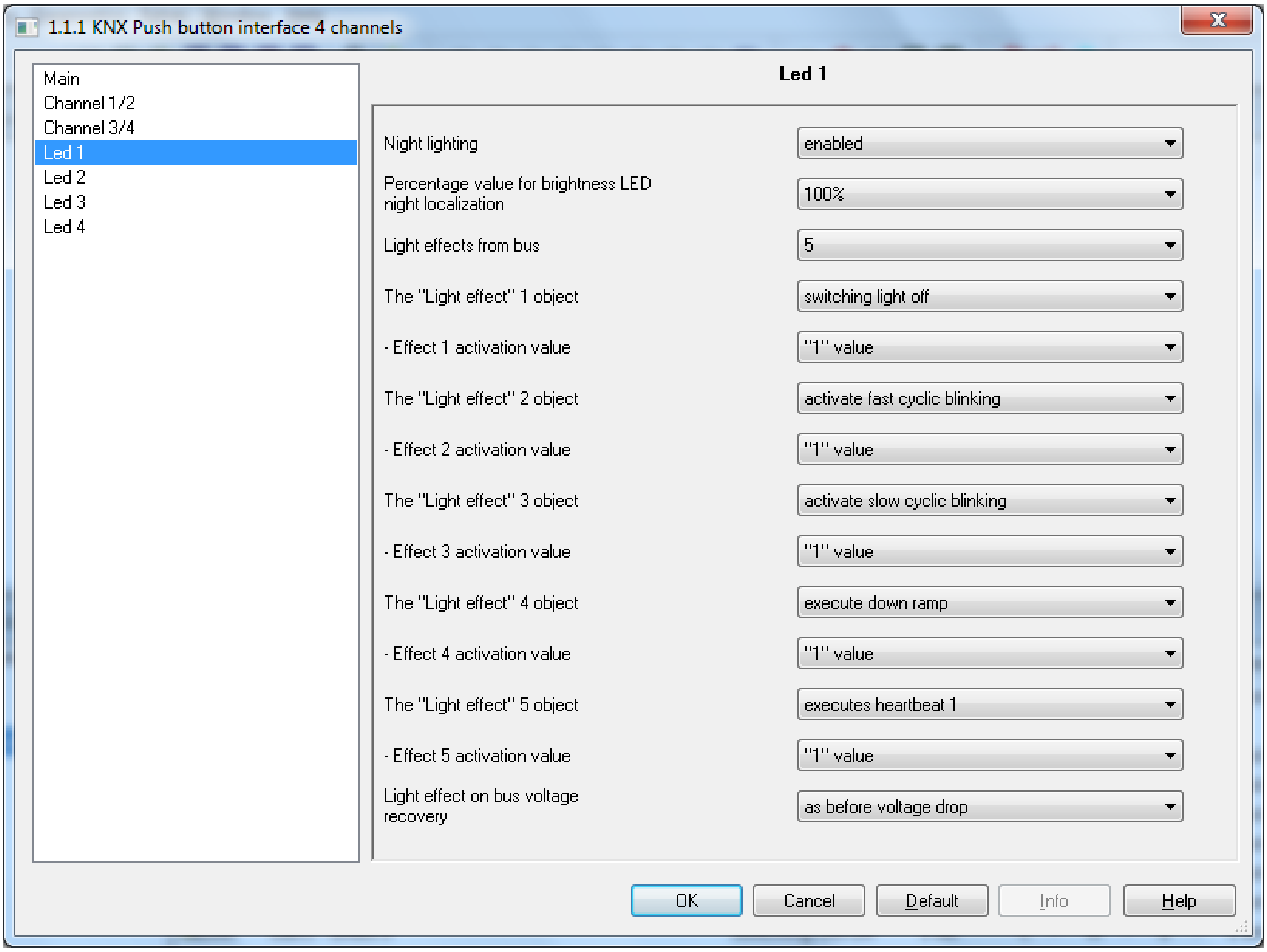
Block: disabled

Fig. 17.1

No new parameters are enabled with this function.

18 “*Led X*” menu

This is used to define and personalise the operation of the signalling LED associated with the channel.
The signalling LED is connected to the device output contacts and can act as night lighting or be autonomously managed by means of the relative communication objects. The communication objects enabled by this function are ***Led x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x - Effect 4*** and ***Led x - Effect 5*** (Data Point Type: 1.001 DPT_Switch).
The basic structure of the menu is as follows:



Led 1

Night lighting	enabled
Percentage value for brightness LED night localization	100%
Light effects from bus	5
The "Light effect" 1 object	switching light off
- Effect 1 activation value	"1" value
The "Light effect" 2 object	activate fast cyclic blinking
- Effect 2 activation value	"1" value
The "Light effect" 3 object	activate slow cyclic blinking
- Effect 3 activation value	"1" value
The "Light effect" 4 object	execute down ramp
- Effect 4 activation value	"1" value
The "Light effect" 5 object	executes heartbeat 1
- Effect 5 activation value	"1" value
Light effect on bus voltage recovery	as before voltage drop

Fig. 18.1

18.1 Parameters

18.1.1 Night lighting

This parameter can be used to enable the night lighting associated with channel x. The values that can be set are:

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

Selecting **enabled**, the **"Percentage value for brightness LED night localisation"** parameter is made visible.

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

- from 1% to **100% (default value)** with steps of 5

18.1.2 Light effects from BUS

This is used to enable various communication objects to activate light signalling via BUS telegram. The values that can be set are:

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

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

18.1.3 Light object x

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

- **status feedback** **(default value)**, only visible if localisation is disabled
- switches off the signalling **(default value)**, only visible if localisation is enabled
- activate fast cyclic blinking
- activate slow cyclic blinking
- execute down ramp
- execute heartbeat 1
- execute heartbeat 2
- execute medusa
- execute blink
- execute heartbeat 3
- execute heartbeat 4
- execute fast blinking
- execute slow blinking
- execute very slow blinking
- execute 3 blinks
- execute personalised effect

If **personalised effect** is selected, the new **Personalise effect y** configuration menu will be visualised, with 1≤Y≤5 (see the "Personalise effect y menu" paragraph)

18.1.4 Effect x activation values

The parameters **“Effect 1 activation value”**, **“Effect 2 activation value”**, **“Effect 3 activation value”**, **“Effect 4 activation value”** and **“Effect 5 activation value”** are used to define which logic value received via the objects ***Led x - Effect 1***, ***Led x - Effect 2***, ***Led x - Effect 3***, ***Led x - Effect 4*** and ***Led x - Effect 5*** activates the associated lighting effect. The values that can be set for this parameter are:

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

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

18.1.5 Light effect on BUS voltage recovery

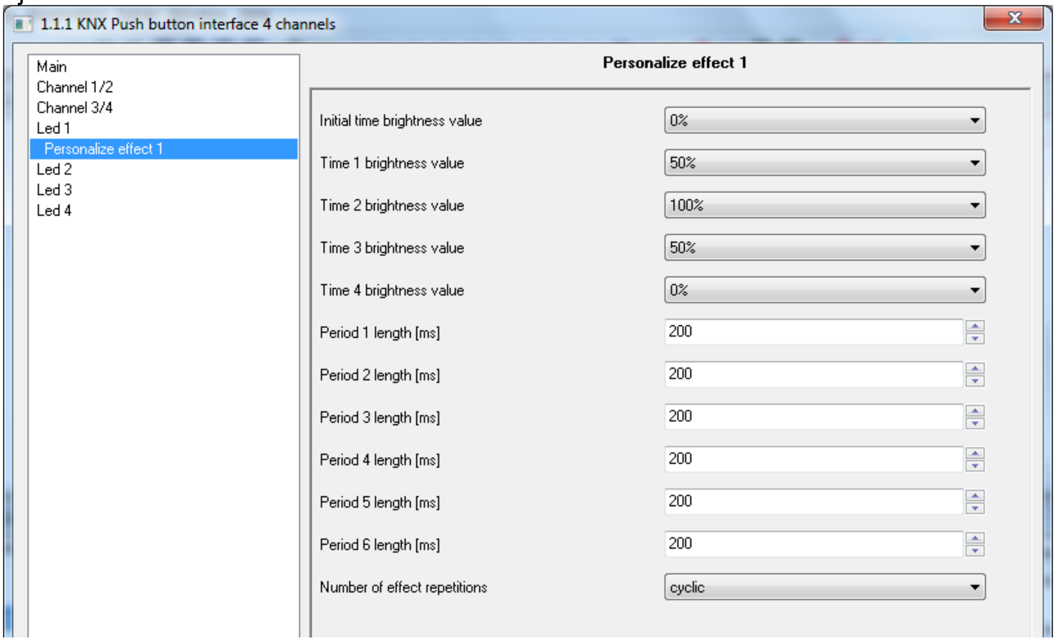
This parameter is used to set the light signalling effect that is activated when the BUS voltage is recovered. The values that can be set are:

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

Selecting the value **no effect**, if night lighting is activate, when BUS voltage is recovered the night lighting will be activated.

18.2 “Personalise effect y” menu

This menu is displayed if the value for the **"Light effect” y object”** parameter in the **LED x** menu is set to **execute personalised effect**.
In this menu, all the parameters used for creating the light effects are visible and configurable; in this way, the user can create a personalized light effect. These light effects are activated/deactivated by the ***LED x - Effect y*** objects. The basic structure of the menu is as follows:



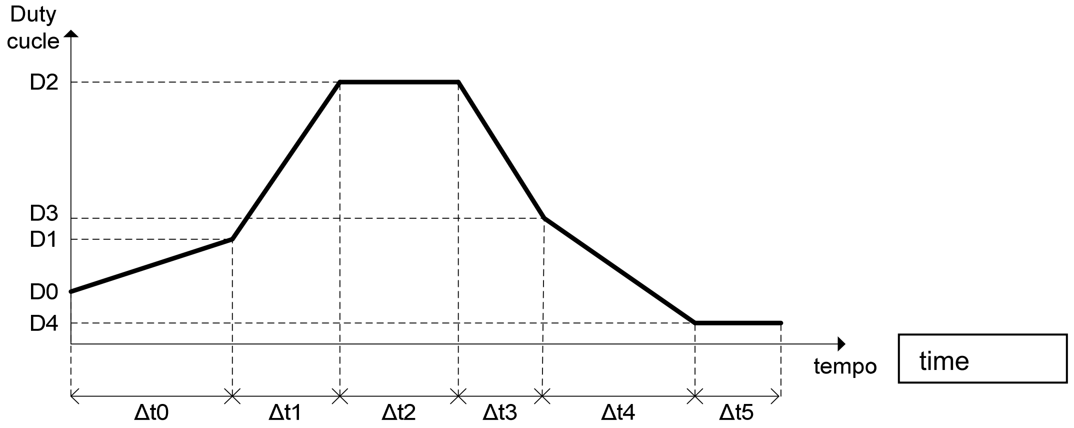
1.1.1 KNX Push button interface 4 channels

Personalize effect 1

Initial time brightness value	0%
Time 1 brightness value	50%
Time 2 brightness value	100%
Time 3 brightness value	50%
Time 4 brightness value	0%
Period 1 length [ms]	200
Period 2 length [ms]	200
Period 3 length [ms]	200
Period 4 length [ms]	200
Period 5 length [ms]	200
Period 6 length [ms]	200
Number of effect repetitions	cyclic

Fig. 18.2

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



The available variables are:

- 6 parameters Δ_t0, Δ_t1, Δ_t2, Δ_t3, Δ_t4, Δ_t5 expressed in ms (0-65535) that define the duration of the brightness control ramp between the value $\Delta_t(n)$ and $\Delta_t(n+1)$ or, in this case Δ_t2, the interval for which the brightness D2 will be maintained
- 5 parameters D0, D1, D2, D3, D4 (0-255) that define LED brightness values (duty-cycle). The values that can be set for these parameters will be displayed to the user as a percentage value between 0% and 100%, according to the proportion D(n)= Parameter*255/100
- 1 parameter that defines the number of cycles for **repeating** the effect (1 .. 254);

18.2.1 Parameters

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

- from 0% to 100% with steps of 5, **0% (default value D0-D4)**, **50% (default value D1-D3)** and **100% (default value D2)**

The parameters used to define the duration of the control ramp between a brightness value and the next are **“Period 1 length [ms]”** (Δ_t0), **“Period 2 length [ms]”** (Δ_t1), **“Period 3 length [ms]”** (Δ_t2), **“Period 4 length [ms]”** (Δ_t3), **“Period 5 length [ms]”** (Δ_t4) and **“Period 6 length [ms]”** (Δ_t5), which can assume the following values:

- from 0 to 65535 with steps of 1, **200 (default value)**

The **“Number of effect repetitions”** parameter defines how many times the set light effect must be repeated when an activation command is received from the BUS. The values that can be set are:

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

19 Communication objects

The communication objects are listed in the tables in the following paragraphs, divided according to functions.

19.1 Communication objects with output functions

The variations of the objects shown in light blue in the table below are not shown for objects B (objects 2/51/100/149), C (objects 3/52/101/150), D (object 4/53/102/151), E (objects 5/54/103/152), F (objects 6/55/104/153), G (objects 7/56/105/154) and H (objects 8/57/106/155) due to space limitations, but they are present.
The variations of the objects shown in pink in the table below are not shown for objects B (objects 10/59/108/157), C (objects 11/60/109/158), D (object 12/61/110/159), E (objects 13/62/111/160), F (objects 14/63/112/161), G (objects 15/64/113/162) and H (objects 16/65/114/163) due to space limitations, but they are present.
The variations of the objects shown in orange in the table below are not shown for objects B (objects 18/67/116/165), C (objects 19/68/117/166), D (object 20/69/118/167), E (objects 21/70/119/168), F (objects 22/71/120/169), G (objects 23/72/121/170) and H (objects 24/73/122/171) due to space limitations, but they are present.
The variations of the objects shown in green in the table below are not shown for objects B (objects 26/75/124/173), C (objects 27/75/125/174), D (object 28/76/126/175), E (objects 29/77/127/176), F (objects 30/78/128/177), G (objects 31/79/129/178) and H (objects 32/81/130/179) due to space limitations, but they are present.
The variations of the objects shown in grey in the table below are not shown for objects B (objects 34/83/132/181), C (objects 35/84/133/182), D (object 36/85/134/183), E (objects 37/86/135/184), F (objects 38/92/136/185), G (objects 39/93/137/186) and H (objects 40/89/138/187) due to space limitations, but they are present.

***NB:** for device GW90727 - 2-channel KNX contact interface, the objects relating to channels 3 and 4 are not present.*

#				Object name	Object function	Description	Datapoint type
Ch 1	Ch 2	Ch 3	Ch 4				
1	50	99	148	Ch.x - Switch	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
1		99		Ch.x/y - Switch	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
1	50	99	148	Ch.x – Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
1		99		Ch.x/y – Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
1	50	99	148	Ch.x - Scene	Execute/Store	Sends scene memorising/execution commands	18.001 DPT_SceneControl
1	50	99	148	Ch.x – Sequence A	On/Off	Sends On/Off commands associated with object A of the sequence	1.001 DPT_Switch
1	50	99	148	Ch.x – Object A 1-bit value	1/0 value	Sends values 1/0 associated with object A	1.002 DPT_Bool
1	50	99	148	Ch.x – Primary counter	Value 1 byte unsigned	Sends the unsigned value (0..255) of the primary counter	5.010 DPT_Value_1_Ucount
1	50	99	148	Ch.x – Primary counter	Value 1 byte signed	Sends the signed value (-128..127) of the primary counter	6.010 DPT_Value_1_Count
1	50	99	148	Ch.x – Primary counter	Value 2 byte unsigned	Sends the unsigned value (0..65535) of the primary counter	7.001 DPT_Value_2_Ucount
1	50	99	148	Ch.x – Primary counter	Value 2 byte signed	Sends the signed value (-32768..32767) of the primary counter	8.001 DPT_Value_2_Count

1	50	99	148	Ch.x - Primary counter	Value 4 byte unsigned	Sends the unsigned value (0.. 4294967295) of the primary counter	12.001 DPT_Value_4_Ucount
1	50	99	148	Ch.x - Primary counter	Value 4 byte signed	Sends the signed value (-2147483648.. 2147483647) of the primary counter	13.001 DPT_Value_4_Count
1	50	99	148	Ch.x – Temperature sensor	Measured value [°C]	Sends the temperature value in °C	9.001 DPT_Temp
1	50	99	148	Ch.x – Temperature sensor	Measured value [°K]	Sends the temperature value in °K	9.002 DPT_Tempd
1	50	99	148	Ch.x – Temperature sensor	Measured value [°F]	Sends the temperature value in °F	9.027 DPT_Temp_F
1	50	99	148	Ch.x – Object A 2-bit value	On/Off forced positioning	Sends values 1/0 associated with object A	1.002 DPT_Switch_Control
1	50	99	148	Ch.x - Object A 1-byte value	Unsigned value	Sends unsigned values (0..255) associated with object A	5.010 DPT_Value_1_Ucount
1	50	99	148	Ch.x - Object A 1-byte value	Signed value	Sends signed values (-128..127) associated with object A	6.010 DPT_Value_1_Count
1	50	99	148	Ch.x - Object A 1-byte value	% Value	Sends the percentage values (0%..100%) associated with object A	5.001 DPT_Scaling
1	50	99	148	Ch.x - Object A 1-byte value	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off)	20.102 DPT_HVACMode
1	50	99	148	Ch.x - Object A 2-byte value	Unsigned value	Sends unsigned values (0..65535) associated with object A	7.001 DPT_Value_2_Ucount
1	50	99	148	Ch.x - Object A 2-byte value	Signed value	Sends signed values (-32768..32767) associated with object A	8.001 DPT_Value_2_Count
1	50	99	148	Ch.x - Object A 3-byte value	RGB colour	Sends the values of the three RGB colour components associated with object A	232.600 DPT_Colour_RGB
1	50	99	148	Ch.x - Object A 4-byte value	Unsigned value	Sends unsigned values (0.. 4294967295) associated with object A	12.001 DPT_Value_4_Ucount
1	50	99	148	Ch.x - Object A 4-byte value	Signed value	Sends signed values (-2147483648.. 2147483647) associated with object A	13.001 DPT_Value_4_Count
1	50	99	148	Ch.x - Object A 14-byte value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
1	50	99	148	Ch.x – Single press 1-bit object A	1/0 value	Sends values 1/0 associated with single press on object A	1.002 DPT_Bool
1	50	99	148	Ch.x – Single press 2-bit object A	On/Off forced positioning	Sends values 1/0 associated with single press on object A	1.002 DPT_Switch_Control
1	50	99	148	Ch.x - Single press 1-byte object A	Unsigned value	Sends unsigned values (0..255) associated with single press on object A	5.010 DPT_Value_1_Ucount
1	50	99	148	Ch.x - Single press 1-byte object A	Signed value	Sends signed values (-128..127) associated with single press on object A	6.010 DPT_Value_1_Count
1	50	99	148	Ch.x - Single press 1-byte object A	% Value	Sends the percentage values (0%..100%) associated with single press on object A	5.001 DPT_Scaling
1	50	99	148	Ch.x - Single press 1-byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) associated with the single press on object A	20.102 DPT_HVACMode
1	50	99	148	Ch.x - Single press 2-byte object A	Unsigned value	Sends unsigned values (0..65535) associated with single press on object A	7.001 DPT_Value_2_Ucount
1	50	99	148	Ch.x - Single press 2-byte object A	Signed value	Sends signed values (-32768..32767) associated with single press on object A	8.001 DPT_Value_2_Count

1	50	99	148	Ch.x - Single press 3-byte object A	RGB colour	Sends the values of the three RGB colour components associated with the single press on object A	232.600 DPT_Colour_RGB
1	50	99	148	Ch.x - Single press 4-byte object A	Unsigned value	Sends unsigned values (0.. 4294967295) associated with single press on object A	12.001 DPT_Value_4_Ucount
1	50	99	148	Ch.x - Single press 4-byte object A	Signed value	Sends signed values (-2147483648.. 2147483647) associated with single press on object A	13.001 DPT_Value_4_Count
2	51	100	149	Ch.x – Brightness dimming	Increase/decrease	Sends brightness dimming commands	3.007 DPT_Control_Dimming
2		100		Ch.x/y - Brightness dimming	Increase/decrease	Sends brightness dimming commands	3.007 DPT_Control_Dimming
2	51	100	149	Ch.x – Shutter stop/Louvres control	Stop/Step	Send louvres stop/adjustment commands	1.007 DPT_Step
2		100		Ch.x/y – Shutter stop/Louvres control	Stop/Step	Send louvres stop/adjustment commands	1.007 DPT_Step
2	51	100	149	Ch.x – Sequence B	On/Off	Sends On/Off commands associated with object B of the sequence	1.001 DPT_Switch
2	51	100	149	Ch.x – Primary counter bit overflow	Overflow status	Sends the primary counter overflow feedback	1.002 DPT_Bool
2	51	100	149	Ch.x – Object B 1-bit value	1/0 value	Sends values 1/0 associated with object B	1.002 DPT_Bool
2	51	100	149	Ch.x – Single press 1-bit object B	1/0 value	Sends values 1/0 associated with single press on object B	1.002 DPT_Bool
3	52	101	150	Ch.x – Sequence C	On/Off	Sends On/Off commands associated with object C of the sequence	1.001 DPT_Switch
3	52	101	150	Ch.x – Primary counter byte overflow	Overflow status	Sends the value associated with the primary counter overflow feedback	5.010 DPT_Value_1_Ucount
3	52	101	150	Ch.x – Object C 1-bit value	1/0 value	Sends values 1/0 associated with object C	1.002 DPT_Bool
3	52	101	150	Ch.x – Single press 1-bit object C	1/0 value	Sends values 1/0 associated with single press on object C	1.002 DPT_Bool
4	53	102	151	Ch.x – Sequence D	On/Off	Sends On/Off commands associated with object D of the sequence	1.001 DPT_Switch
4	53	102	151	Ch.x – Differential counter	Value 1 byte unsigned	Sends the unsigned value (0..255) of the differential counter	5.010 DPT_Value_1_Ucount
4	53	102	151	Ch.x – Differential counter	Value 1 byte signed	Sends the signed value (-128..127) of the differential counter	6.010 DPT_Value_1_Count
4	53	102	151	Ch.x - Differential counter	Value 2 byte unsigned	Sends the unsigned value (0..65535) of the differential counter	7.001 DPT_Value_2_Ucount
4	53	102	151	Ch.x - Differential counter	Value 2 byte signed	Sends the signed value (-32768..32767) of the differential counter	8.001 DPT_Value_2_Count
4	53	102	151	Ch.x - Differential counter	Value 4 byte unsigned	Sends the unsigned value (0.. 4294967295) of the differential counter	12.001 DPT_Value_4_Ucount
4	53	102	151	Ch.x - Differential counter	Value 4 byte signed	Sends the signed value (-2147483648.. 2147483647) of the differential counter	13.001 DPT_Value_4_Count
4	53	102	151	Ch.x – Object D 1-bit value	1/0 value	Sends values 1/0 associated with object D	1.002 DPT_Bool
4	53	102	151	Ch.x – Single press 1-bit object D	1/0 value	Sends values 1/0 associated with single press on object D	1.002 DPT_Bool
5	54	103	152	Ch.x – Sequence E	On/Off	Sends On/Off commands associated with object E of	1.001 DPT_Switch

						the sequence	
5	54	103	152	Ch.x – Differential counter bit overflow	Overflow status	Sends the differential counter overflow feedback	1.002 DPT_Bool
5	54	103	152	Ch.x – Object E 1-bit value	1/0 value	Sends values 1/0 associated with object E	1.002 DPT_Bool
5	54	103	152	Ch.x – Single press 1-bit object E	1/0 value	Sends values 1/0 associated with single press on object E	1.002 DPT_Bool
6	55	104	153	Ch.x – Sequence F	On/Off	Sends On/Off commands associated with object F of the sequence	1.001 DPT_Switch
6	55	104	153	Ch.x – Differential counter byte overflow	Overflow status	Sends the value associated with the differential counter overflow feedback	5.010 DPT_Value_1_Ucount
6	55	104	153	Ch.x – Object F 1-bit value	1/0 value	Sends values 1/0 associated with object F	1.002 DPT_Bool
6	55	104	153	Ch.x – Single press 1-bit object F	1/0 value	Sends values 1/0 associated with single press on object F	1.002 DPT_Bool
7	56	105	154	Ch.x – Sequence G	On/Off	Sends On/Off commands associated with object G of the sequence	1.001 DPT_Switch
7	56	105	154	Ch.x – Object G 1-bit value	1/0 value	Sends values 1/0 associated with object G	1.002 DPT_Bool
7	56	105	154	Ch.x – Single press 1-bit object G	1/0 value	Sends values 1/0 associated with single press on object G	1.002 DPT_Bool
8	57	106	155	Ch.x – Sequence H	On/Off	Sends On/Off commands associated with H Object of the sequence	1.001 DPT_Switch
8	57	106	155	Ch.x – Object H 1-bit value	1/0 value	Sends values 1/0 associated with object H	1.002 DPT_Bool
8	57	106	155	Ch.x – Single press 1-bit object H	1/0 value	Sends values 1/0 associated with single press on object H	1.002 DPT_Bool
9	58	107	156	Ch.x – Double press 1-bit object A	1/0 value	Sends values 1/0 associated with double press on object A	1.002 DPT_Bool
9	58	107	156	Ch.x – Double press 2-bit object A	On/Off forced positioning	Sends values 1/0 associated with double press on object A	1.002 DPT_Switch_Control
9	58	107	156	Ch.x – Double press 1-byte object A	Unsigned value	Sends unsigned values (0..255) associated with double press on object A	5.010 DPT_Value_1_Ucount
9	58	107	156	Ch.x – Double press 1-byte object A	Signed value	Sends signed values (-128..127) associated with double press on object A	6.010 DPT_Value_1_Count
9	58	107	156	Ch.x – Double press 1-byte object A	% Value	Sends the percentage values (0%..100%) associated with double press on object A	5.001 DPT_Scaling
9	58	107	156	Ch.x – Double press 1-byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) associated with the double press on object A	20.102 DPT_HVACMode
9	58	107	156	Ch.x – Double press 2-byte object A	Unsigned value	Sends unsigned values (0..65535) associated with double press on object A	7.001 DPT_Value_2_Ucount
9	58	107	156	Ch.x – Double press 2-byte object A	Signed value	Sends signed values (-32768..32767) associated with double press on object A	8.001 DPT_Value_2_Count
9	58	107	156	Ch.x – Double press 3-byte object A	RGB colour	Sends the values of the three RGB colour components associated with the double press on object A	232.600 DPT_Colour_RGB
9	58	107	156	Ch.x – Double press 4-byte object A	Unsigned value	Sends unsigned values (0.. 4294967295) associated with double press on object A	12.001 DPT_Value_4_Ucount
9	58	107	156	Ch.x – Double press 4-byte object A	Signed value	Sends signed values (-2147483648.. 2147483647) associated with double press on object A	13.001 DPT_Value_4_Count
10	59	108	157	Ch.x – Double press 1-bit	1/0 value	Sends values 1/0 associated	1.002 DPT_Bool

				object B		with double press on object B	
11	60	109	158	Ch.x – Double press 1-bit object C	1/0 value	Sends values 1/0 associated with double press on object C	1.002 DPT_Bool
12	61	110	159	Ch.x – Double press 1-bit object D	1/0 value	Sends values 1/0 associated with double press on object D	1.002 DPT_Bool
13	62	111	160	Ch.x – Double press 1-bit object E	1/0 value	Sends values 1/0 associated with double press on object E	1.002 DPT_Bool
14	63	112	161	Ch.x – Double press 1-bit object F	1/0 value	Sends values 1/0 associated with double press on object F	1.002 DPT_Bool
15	64	113	162	Ch.x – Double press 1-bit object G	1/0 value	Sends values 1/0 associated with double press on object G	1.002 DPT_Bool
16	65	114	163	Ch.x – Double press 1-bit object H	1/0 value	Sends values 1/0 associated with double press on object H	1.002 DPT_Bool
17	66	115	164	Ch.x – Triple press 1-bit object A	1/0 value	Sends values 1/0 associated with triple press on object A	1.002 DPT_Bool
17	66	115	164	Ch.x – Triple press 2-bit object A	On/Off forced positioning	Sends values 1/0 associated with triple press on object A	1.002 DPT_Switch_Control
17	66	115	164	Ch.x – Triple press 1-byte object A	Unsigned value	Sends unsigned values (0..255) associated with triple press on object A	5.010 DPT_Value_1_Ucount
17	66	115	164	Ch.x – Triple press 1-byte object A	Signed value	Sends signed values (-128..127) associated with triple press on object A	6.010 DPT_Value_1_Count
17	66	115	164	Ch.x – Triple press 1-byte object A	% Value	Sends the percentage values (0%..100%) associated with triple press on object A	5.001 DPT_Scaling
17	66	115	164	Ch.x – Triple press 1-byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) associated with the triple press on object A	20.102 DPT_HVACMode
17	66	115	164	Ch.x – Triple press 2-byte object A	Unsigned value	Sends unsigned values (0..65535) associated with triple press on object A	7.001 DPT_Value_2_Ucount
17	66	115	164	Ch.x – Triple press 2-byte object A	Signed value	Sends signed values (-32768..32767) associated with triple press on object A	8.001 DPT_Value_2_Count
17	66	115	164	Ch.x – Triple press 3-byte object A	RGB colour	Sends the values of the three RGB colour components associated with the triple press on object A	232.600 DPT_Colour_RGB
17	66	115	164	Ch.x – Triple press 4-byte object A	Unsigned value	Sends unsigned values (0.. 4294967295) associated with triple press on object A	12.001 DPT_Value_4_Ucount
17	66	115	164	Ch.x – Triple press 4-byte object A	Signed value	Sends signed values (-2147483648.. 2147483647) associated with triple press on object A	13.001 DPT_Value_4_Count
18	67	116	165	Ch.x – Triple press 1-bit object B	1/0 value	Sends values 1/0 associated with triple press on object B	1.002 DPT_Bool
19	68	117	166	Ch.x – Triple press 1-bit object C	1/0 value	Sends values 1/0 associated with triple press on object C	1.002 DPT_Bool
20	69	118	167	Ch.x – Triple press 1-bit object D	1/0 value	Sends values 1/0 associated with triple press on object D	1.002 DPT_Bool
21	70	119	168	Ch.x – Triple press 1-bit object E	1/0 value	Sends values 1/0 associated with triple press on object E	1.002 DPT_Bool
22	71	120	169	Ch.x – Triple press 1-bit object F	1/0 value	Sends values 1/0 associated with triple press on object F	1.002 DPT_Bool
23	72	121	170	Ch.x – Triple press 1-bit object G	1/0 value	Sends values 1/0 associated with triple press on object G	1.002 DPT_Bool
24	73	122	171	Ch.x – Triple press 1-bit object H	1/0 value	Sends values 1/0 associated with triple press on object H	1.002 DPT_Bool
25	74	123	172	Ch.x – Quadruple press 1-bit object A	1/0 value	Sends values 1/0 associated with quadruple press on object A	1.002 DPT_Bool
25	74	123	172	Ch.x – Quadruple press 2-bit object A	On/Off forced positioning	Sends values 1/0 associated with quadruple press on	1.002 DPT_Switch_Control

						object A	
25	74	123	172	Ch.x – Quadruple press 1-byte object A	Unsigned value	Sends unsigned values (0..255) associated with quadruple press on object A	5.010 DPT_Value_1_Ucount
25	74	123	172	Ch.x – Quadruple press 1-byte object A	Signed value	Sends signed values (-128..127) associated with quadruple press on object A	6.010 DPT_Value_1_Count
25	74	123	172	Ch.x – Quadruple press 1-byte object A	% Value	Sends the percentage values (0%..100%) associated with quadruple press on object A	5.001 DPT_Scaling
25	74	123	172	Ch.x – Quadruple press 1-byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) associated with the quadruple press on object A	20.102 DPT_HVACMode
25	74	123	172	Ch.x – Quadruple press 2-byte object A	Unsigned value	Sends unsigned values (0..65535) associated with quadruple press on object A	7.001 DPT_Value_2_Ucount
25	74	123	172	Ch.x – Quadruple press 2-byte object A	Signed value	Sends signed values (-32768..32767) associated with quadruple press on object A	8.001 DPT_Value_2_Count
25	74	123	172	Ch.x – Quadruple press 3-byte object A	RGB colour	Sends the values of the three RGB colour components associated with the quadruple press on object A	232.600 DPT_Colour_RGB
25	74	123	172	Ch.x – Quadruple press 4-byte object A	Unsigned value	Sends unsigned values (0.. 4294967295) associated with quadruple press on object A	12.001 DPT_Value_4_Ucount
25	74	123	172	Ch.x – Quadruple press 4-byte object A	Signed value	Sends signed values (-2147483648.. 2147483647) associated with quadruple press on object A	13.001 DPT_Value_4_Count
26	75	124	173	Ch.x – Quadruple press 1-bit object B	1/0 value	Sends values 1/0 associated with quadruple press on object B	1.002 DPT_Bool
27	76	125	174	Ch.x – Quadruple press 1-bit object C	1/0 value	Sends values 1/0 associated with quadruple press on object C	1.002 DPT_Bool
28	77	126	175	Ch.x – Quadruple press 1-bit object D	1/0 value	Sends values 1/0 associated with quadruple press on object D	1.002 DPT_Bool
29	78	127	176	Ch.x – Quadruple press 1-bit object E	1/0 value	Sends values 1/0 associated with quadruple press on object E	1.002 DPT_Bool
30	79	128	177	Ch.x – Quadruple press 1-bit object F	1/0 value	Sends values 1/0 associated with quadruple press on object F	1.002 DPT_Bool
31	80	129	178	Ch.x – Quadruple press 1-bit object G	1/0 value	Sends values 1/0 associated with quadruple press on object G	1.002 DPT_Bool
32	81	130	179	Ch.x – Quadruple press 1-bit object H	1/0 value	Sends values 1/0 associated with quadruple press on object H	1.002 DPT_Bool
33	82	131	180	Ch.x – Long press 1-bit object A	1/0 value	Sends values 1/0 associated with long press on object A	1.002 DPT_Bool
33	82	131	180	Ch.x - Long press 2-bit object A	On/Off forced positioning	Sends values 1/0 associated with long press on object A	1.002 DPT_Switch_Control
33	82	131	180	Ch.x – Long press 1-byte object A	Unsigned value	Sends unsigned values (0..255) associated with long press on object A	5.010 DPT_Value_1_Ucount
33	82	131	180	Ch.x – Long press 1-byte object A	Signed value	Sends signed values (-128..127) associated with long press on object A	6.010 DPT_Value_1_Count
33	82	131	180	Ch.x – Long press 1-byte object A	% Value	Sends the percentage values (0%..100%) associated with	5.001 DPT_Scaling

						long press on object A	
33	82	131	180	Ch.x – Long press 1-byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) associated with the long press on object A	20.102 DPT_HVACMode
33	82	131	180	Ch.x - Long press 2-byte object A	Unsigned value	Sends unsigned values (0..65535) associated with long press on object A	7.001 DPT_Value_2_Ucount
33	82	131	180	Ch.x - Long press 2-byte object A	Signed value	Sends signed values (-32768..32767) associated with long press on object A	8.001 DPT_Value_2_Count
33	82	131	180	Ch.x - Long press 3-byte object A	RGB colour	Sends the values of the three RGB colour components associated with the long press on object A	232.600 DPT_Colour_RGB
33	82	131	180	Ch.x - Long press 4-byte object A	Unsigned value	Sends unsigned values (0.. 4294967295) associated with long press on object A	12.001 DPT_Value_4_Ucount
33	82	131	180	Ch.x - Long press 4-byte object A	Signed value	Sends signed values (-2147483648.. 2147483647) associated with long press on object A	13.001 DPT_Value_4_Count
34	83	132	181	Ch.x – Long press 1-bit object B	1/0 value	Sends values 1/0 associated with long press on object B	1.002 DPT_Bool
35	84	133	182	Ch.x – Long press 1-bit object C	1/0 value	Sends values 1/0 associated with long press on object C	1.002 DPT_Bool
36	85	134	183	Ch.x – Long press 1-bit object D	1/0 value	Sends values 1/0 associated with long press on object D	1.002 DPT_Bool
37	86	135	184	Ch.x – Long press 1-bit object E	1/0 value	Sends values 1/0 associated with long press on object E	1.002 DPT_Bool
38	87	136	185	Ch.x – Long press 1-bit object F	1/0 value	Sends values 1/0 associated with long press on object F	1.002 DPT_Bool
39	88	137	186	Ch.x – Long press 1-bit object G	1/0 value	Sends values 1/0 associated with long press on object G	1.002 DPT_Bool
40	89	138	187	Ch.x – Long press 1-bit object H	1/0 value	Sends values 1/0 associated with long press on object H	1.002 DPT_Bool

19.2 Communication objects with input functions

***NB:** for device GW90727 - 2-channel KNX contact interface, the objects relating to channels 3 and 4 are not present and the objects relating to light signalling have a scaled index:*

- *Led 1 - Effect 1 196 → 98*
- *Led 1 - Effect 2 197 → 99*
- *Led 1 - Effect 3 198 → 100*
- *Led 1 - Effect 4 199 → 101*
- *Led 1 - Effect 5 200 → 102*
- *Led 2 - Effect 1 201 → 103*
- *Led 2 - Effect 2 202 → 104*
- *Led 2 - Effect 3 203 → 105*
- *Led 2 - Effect 4 204 → 106*
- *Led 2 - Effect 5 205 → 107*

#				**Object name**	**Object function**	**Description**	**Datapoint type**
Ch 1	**Ch 2**	**Ch 3**	**Ch 4**				
0	49	98	147	Ch.x - Block	Activate/Deactivate	Used to activate/deactivate the block function	1.003 DPT_Enable
0		98		Ch.x/y - Block	Activate/Deactivate	Used to activate/deactivate the block function	1.003 DPT_Enable
2	51	100	149	Ch.x – Scene storing trigger	Store	Receives the request (trigger) to send a scene storing message	1.017 DPT_Trigger
41	90	139	188	Ch.x – Dimmer status feedback	On/Off status	Receives the dimmer status feedback	1.001 DPT_Switch
41	90	139	188	Ch.x – Status feedback	On/Off status	Receives the actuator status feedback for cyclical switching	1.001 DPT_Switch
41	90	139	188	Ch.x – object A status feedback	On/Off status	Receives the actuator status feedback for object A cyclical switching	1.001 DPT_Switch
41	90	139	188	Ch.x – Movement feedback	Up/down	Receives the feedback about the current movement direction of the motor command actuator	1.008 DPT_UpDown
41	90	139	188	Ch.x – Primary counter sending trigger	Counter value transmission	Receives the request (trigger) to send the current value of the primary counter	1.017 DPT_Trigger
42	91	140	189	Ch.x – Object B status feedback	On/Off status	Receives the actuator status feedback for object B cyclical switching	1.001 DPT_Switch
42	91	140	189	Ch.x - Differential counter sending trigger	Counter value transmission	Receives the request (trigger) to send the current value of the differential counter	1.017 DPT_Trigger
43	92	141	190	Ch.x – Object C status feedback	On/Off status	Receives the actuator status feedback for object C cyclical switching	1.001 DPT_Switch
43	92	141	190	Ch.x – Differential counter reset	Resets the value	Receives the value reset command for the differential counter	1.001 DPT_Switch
44	93	142	191	Ch.x – Object D status feedback	On/Off status	Receives the actuator status feedback for object D cyclical switching	1.001 DPT_Switch
45	94	143	192	Ch.x – Object E status feedback	On/Off status	Receives the actuator status feedback for object E cyclical switching	1.001 DPT_Switch
46	95	144	193	Ch.x – Object F status feedback	On/Off status	Receives the actuator status feedback for object F cyclical switching	1.001 DPT_Switch
47	96	145	194	Ch.x – Object G status feedback	On/Off status	Receives the actuator status feedback for object G cyclical	1.001 DPT_Switch

						switching	
48	97	146	195	Ch.x – Object H status feedback	On/Off status	Receives the actuator status feedback for object H cyclical switching	1.001 DPT_Switch
196	201	206	211	Led x - Effect 1	Activate/Deactivate	Switching On /Off light effect 1	1.001 DPT_Switch
197	202	207	212	Led x - Effect 2	Activate/Deactivate	Switching On /Off light effect 2	1.001 DPT_Switch
198	203	208	213	Led x - Effect 3	Activate/Deactivate	Switching On /Off light effect 3	1.001 DPT_Switch
199	204	209	214	Led x - Effect 4	Activate/Deactivate	Switching On /Off light effect 4	1.001 DPT_Switch
200	205	210	215	Led x - Effect 5	Activate/Deactivate	Switching On /Off light effect 5	1.001 DPT_Switch

cod. 7.01.6.764.8

Ai sensi dell'articolo 9 comma 2 della Direttiva Europea 2004/108/CE si informa che responsabile dell'immissione del prodotto sul mercato Comunitario è:
According to article 9 paragraph 2 of the European Directive 2004/108/EC, the responsible for placing the apparatus on the Community market is:
GEWISS S.p.A Via A. Volta, 1 - 24069 Cenate Sotto (BG) Italy Tel: +39 035 946 111 Fax: +39 035 945 270 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

LAST REVISION 04/2014