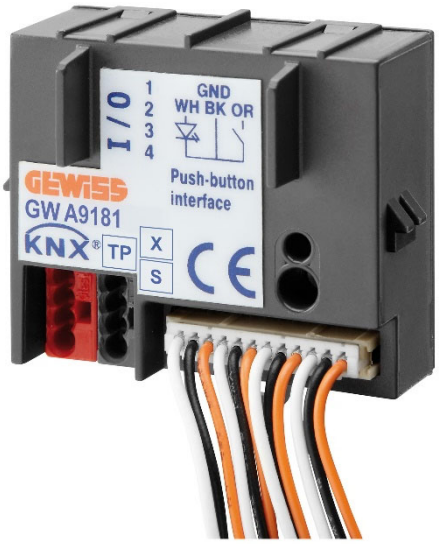
CHORUS MART

KNX Secure 4-channel contacts interface



GWA9181

Technical Manual

Contents

1 Introduction

The KNX contacts interface with 4 channels can be connected with 4 independent, voltage-free input contacts (push-buttons, one-way switches, sensors, etc.) and used for sending the relative commands to actuator devices via the KNX BUS.
The interface is powered via the BUS line. The voltage (SELV) needed to scan the contacts is supplied by the interface itself. The device is equipped with 4 outputs for the possible connection of low-consumption signalling LEDs.

The device supports KNX Data Secure: this technology enhances the security of a KNX installation both during start up and during normal operation, thanks to the exchange of encrypted telegrams.

To make this manual easier to read, all the parameters and communication objects implemented by the device are grouped in different paragraphs, each of which represents the relative configuration menu in the ETS database.

2 Application

The contact interface is an input device that can be inserted:

- in standard flush-mounting boxes, behind the electro-mechanical modules;
- in the supports of the Chorus range, using a GW 10 751, GW 12 751 or GW 14 751 blanking module;
- in junction boxes.

Each input is configured with the ETS software to create one of the functions listed below.

- Input management / Transmission of objects on the BUS
- Scenes
- Priority commands
- Roller shutters/curtain command
- Dimmer command
- Pulse count
- Multiple press/closing contact
- Control of output LED
- Switching sequences
- Temperature sensor
- Setting of threshold values with signalling of threshold exceeding
- Hysteresis management

2.1 Matching limits

The maximum number of logic matchings 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 matched with a maximum of 254 group addresses.

3 “*Main*” menu

The **Main** menu contains the application parameters for all the input channels implemented by the device
The basic structure of the menu is as follows:

Fig 1: Setting ETS parameters – “Main” section

3.1 Parameters

3.1.1 Channel X

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

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

selecting **enabled** displays the **Channel 1**, **Channel 2**, **Channel 3** or **Channel 4** configuration menu (see “Channel X” menu).

3.1.2 Transmission delay after start

To ensure that, with multiple devices in the line, the telegrams sent by the various devices do not collide when the BUS voltage is recovered, it is possible to define the time that must pass after which the device may transmit the telegrams on the BUS following a drop/recovery of the BUS supply voltage. The parameter **Px “Transmission delay after start”** is used to set this delay; The values that can be set are:

- **11.. 21 seconds (depending on physical address)** **(default value)**
- 5.. 9 seconds (depending on physical address)
- 11 seconds
- 13 seconds
- 15 seconds
- 17 seconds
- 19 seconds
- 21 seconds

- no delay

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

4 *"Channel X"* menu

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

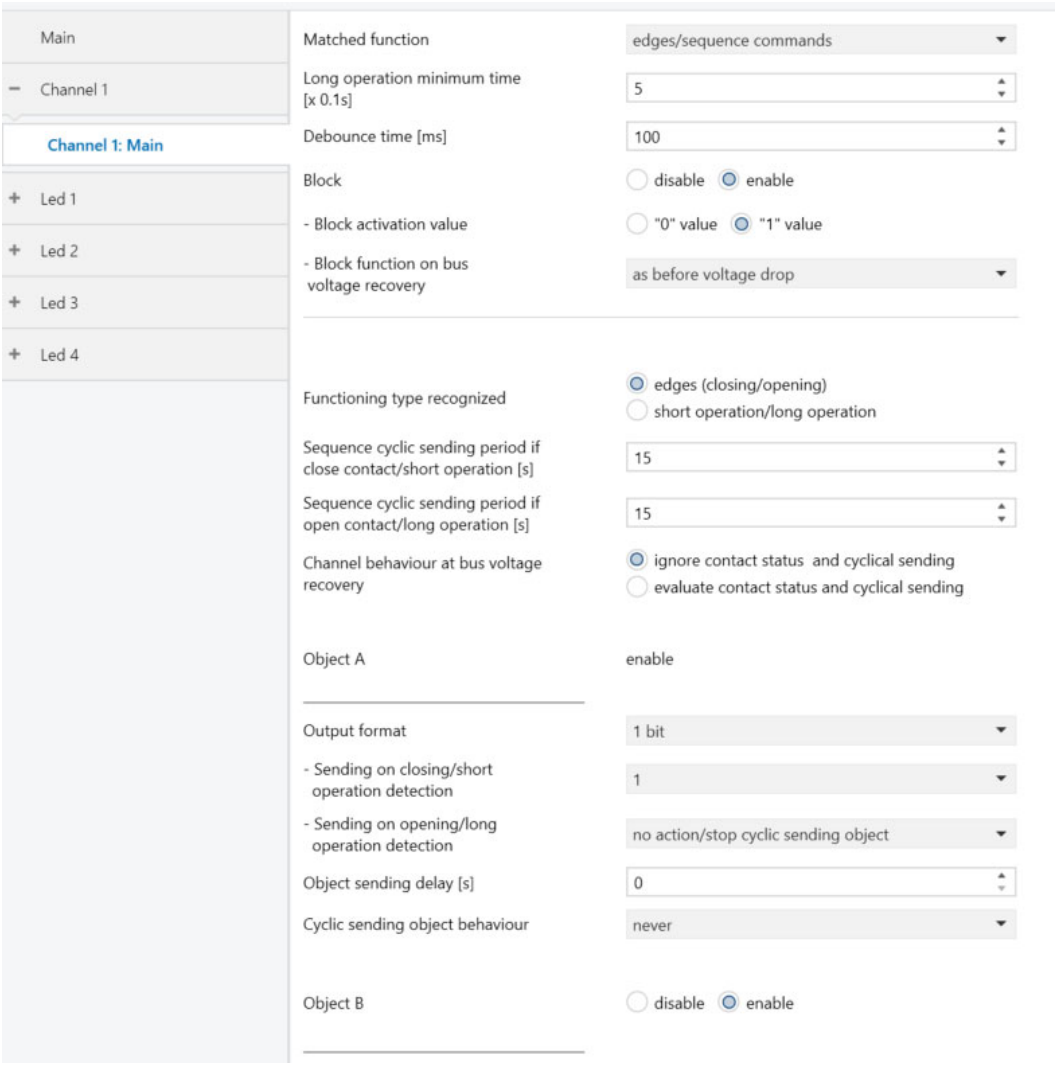


Fig 2: Setting ETS parameters – "Channel x" (independent channels) section

4.1 Parameters

4.1.1 Matched function

The parameter used to define the function implemented by the channel is **Px "Matched function"**; the values that can be set are:

- **edges/sequence commands (default value)**

(See paragraph "Edges/sequence commands" function)
- push-button + stop dimmer

(See paragraph "1 push-button + stop dimmer" function)
- cyclic sending 1 push-button dimmer

(See paragraph "1 push-button dimmer with cyclic sending" function)
- 1 push-button roller shutter control

(See paragraph "1 push-button roller shutter control" function)
- 2 push-button + stop dimmer

(See paragraph "2 push-button + stop dimmer" function)
- cyclic sending 2 push-button dimmer

(See paragraph "2 push-button dimmer with cyclic sending" function)
- 2 push-button roller shutter control

(See paragraph "2 push-button roller shutter control" function)
- scene management

(See paragraph "Scene management" function)
- switching sequences

(See paragraph "Switching sequences" function)
- pulse counter

(See paragraph "Pulse counter" function)
- multiple press/closing contact

(See paragraph "Multiple press/closing contact" function)
- temperature sensor

(See paragraph "Temperature sensor" function)

4.1.2 Long operation minimum time [x 0.1s]

Many of the functions that the channels can perform require differentiation between short and long operation. The parameter **Px "Long operation minimum time** [x 0.1s]" can be used to define the minimum actual time the device must detect the closing contact in order to distinguish a short operation from a long one. The possible values are:

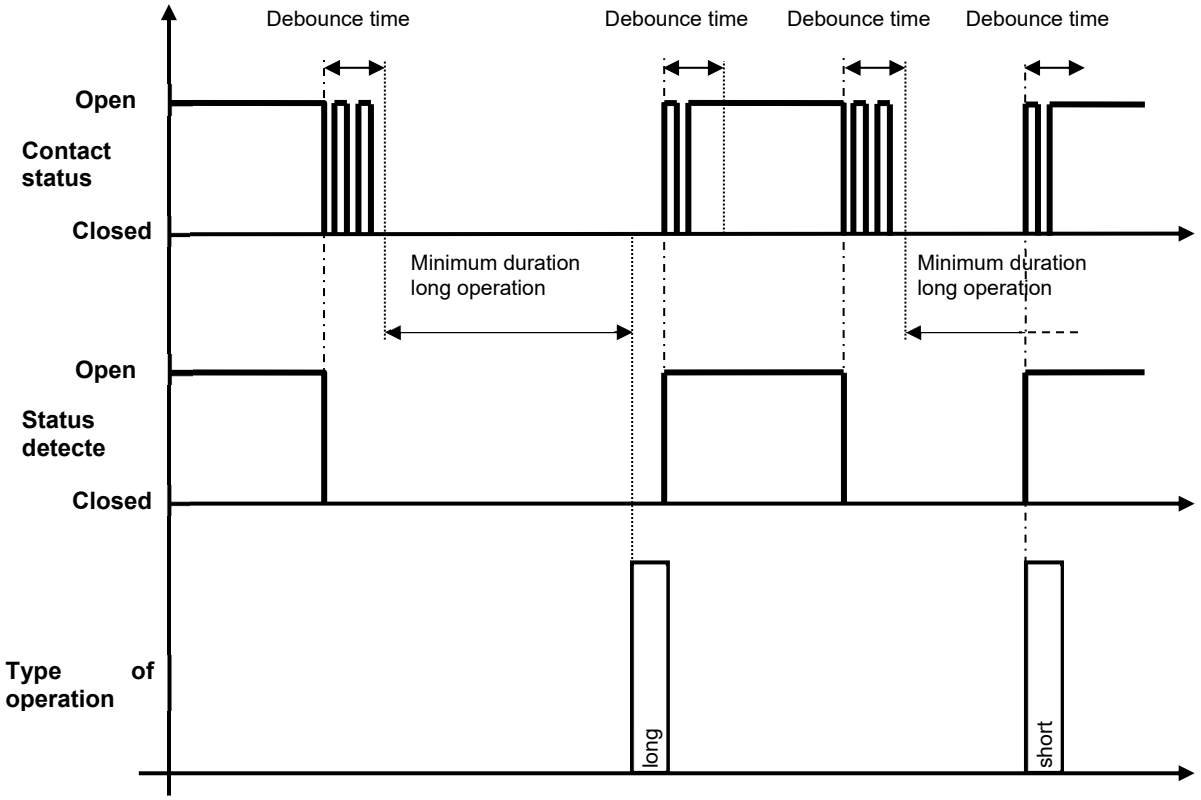
- from 3 to 150 with steps of 1, **5 (default value)**

4.1.3 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 change is detected, a specific time must pass before the device can detect another change. This value can be set in the parameter **Px "Debounce time [ms]"**; the values that can be set are:

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

The following chart summarises the concepts of **Px "Long operation minimum time [x 0.1s]"** and **Px "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 $\mathbf{T_{debounce}}$ starting from the moment when the first change 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 change is from open status to closed status (push-button pressed), it activates a timer whose initial value is the one set in Px "**Long operation minimum time** [x 0.1s]". If the timer expires before the status change 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.

4.1.4 Block

To inhibit the channel when sending commands matched with the closing/opening or long/short operation of the contact, the block function must be activated: this function inhibits the detection of the closing/opening or long/short operation of the contact, thereby preventing the device from sending the telegrams matched with these events on the BUS; If it is activated, any status change that occurs will not be interpreted until a block deactivation command is received. The parameter used to enable the function is **Px "Block"**, visible for all functions matched to the channel with the exception of the "temperature sensor", which may have the following values:

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

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

In particular cases where an edge (opening or closing) or operation (short or long) is matched 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 change 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 change occurred while the block was active).

4.1.5 Block activation value

The parameter **Px "Block activation value"** is used to set the logic value to be assumed by the bit received via BUS telegram to activate the block function; the values that can be set are:

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

4.1.6 Block function on bus voltage recovery

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

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

5 “edges/sequence commands” function

This function is used to set the type and number of commands to send after a status change has been detected, for up to a total of 8 commands per channel; The value of the command can be differentiated according to the event detected (closing/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:

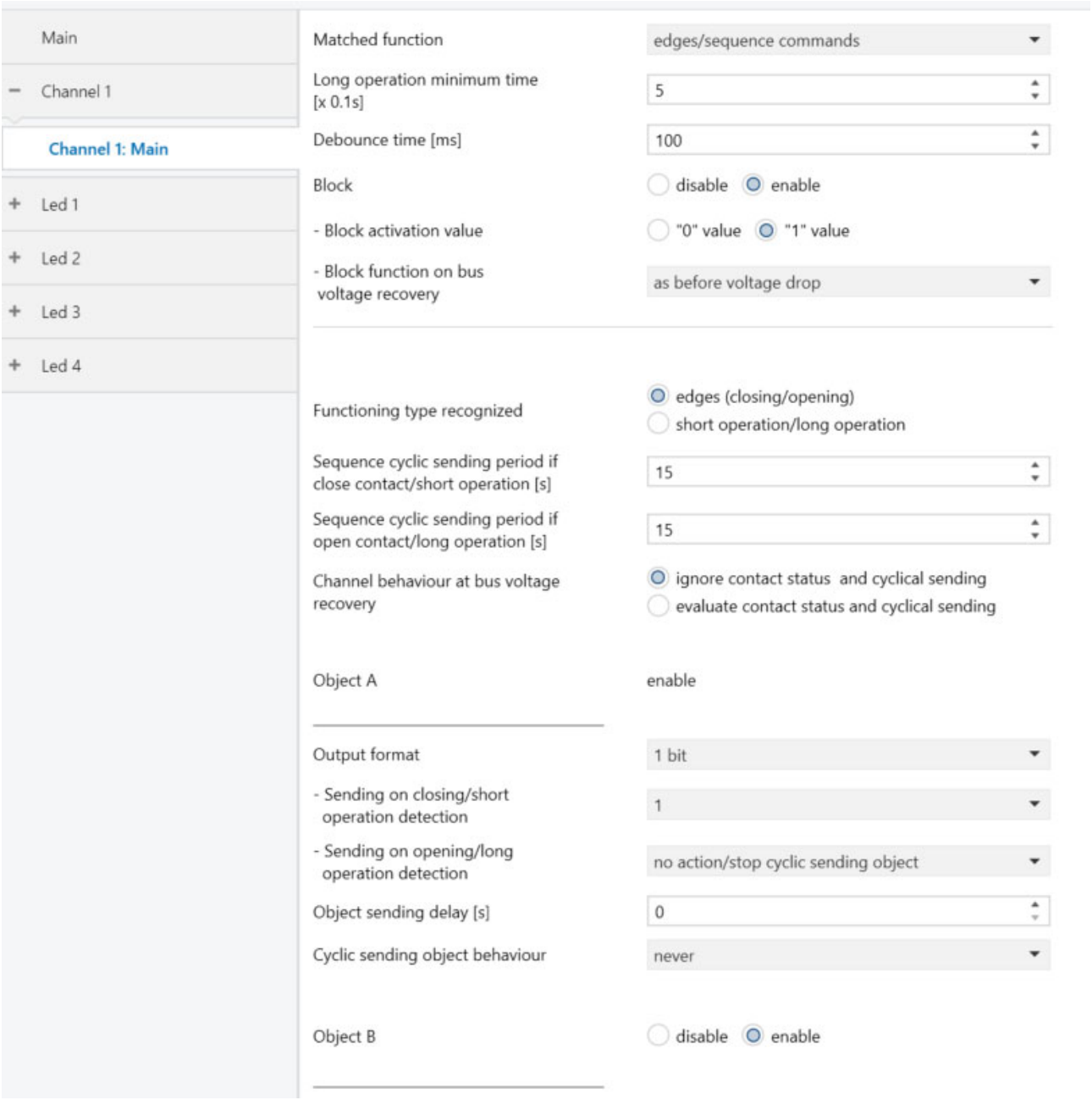


Fig 3: Setting ETS parameters – “Channel x” section - “edges/sequence commands” function

5.1 Parameters

For each channel, 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 closing (or short operation) or opening (or long operation) of the contact; Object A is always enabled, and the parameter **“object z”** (z indicates the object matched with the threshold, included between **B** and **H)** is used to enable a new object to be sent. The parameter can assume the following values:

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

- enable

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

5.1.1 Functioning type recognized

The parameter **Px "Functioning type recognized"** is used to define which type of contact operation generates the sequence sending commands; the values that can be set are:

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

5.1.2 Sending object format

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

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

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

5.1.3 Sending on closing/short operation detection

The parameter **Px "Sending on closing/short operation detection"** is used to set the command or value to be sent following the detection of the closing or short operation of the contact (depending on the type of operation selected) matched with the channel.

5.1.4 Sending on opening/long operation detection

The parameter **Px "Sending on opening/long operation detection"** 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) matched with the channel.

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

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

selecting the value **cyclical switching** makes the parameter **Px "Status feedback object"** visible, and is used to enable and make the communication object ***Ch.x - object z status feedback*** visible, by enabling this object, when the status feedback telegram is received for the object in question, the command that the device will send (via the ***Ch.x - object z 1 bit value*** object) when the event matched 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 - object z status feedback*** object and the last value sent (via the ***Ch.x - object z 1 bit value*** object). The parameter **Px "Status feedback object"** may have the following values:

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

selecting the value **enabled** displays the ***Ch.x - Object z status feedback*** communication object. In this case, every time the BUS voltage is recovered you must send a status read request on this object in order to update the device about the status of the devices connected.

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

  - **no action/stop cyclic sending object** **(default value opening)**
  - **activate forcing ON (down)** **(default closing value)**
  - activate forcing OFF (up)
  - forcing deactivation
  - on forcing/off forcing cyclical switching
  - on forcing/forcing deactivation cyclical switching
  - off forcing/forcing deactivation 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 sending object format is **1 byte unsigned value**, the communication object ***Ch.x - object z 1 byte value*** will be visible and the values that can be set for the two above parameters are:

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

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

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

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

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

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

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

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

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

  by setting **send value**, it is possible to define the value to be sent via the newly displayed parameter **Px "Value (0% ... 100%)"** that can have the following values:

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

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

  - **no action/stop cyclic sending object** **(default value opening)**
  - auto
  - **comfort** **(default value closing)**
  - 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 matched 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 matched 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 sending object format is **2 bytes unsigned value**, the communication object ***Ch.x - object z 2 bytes value*** will be visible and the values that can be set for the two above parameters are:

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

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

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

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

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

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

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

- If the sending object format is **3 bytes RGB colour**, the communication object ***Ch.x - object z 3 bytes value*** will be visible and the values that can be set for the two above parameters are:

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

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

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

- customize

by selecting **customize**, the following parameters are made visible: **Px "Value of RED component (0 ... 255)"**, **Px "Value of GREEN component (0 ... 255)"** and **Px "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, in steps of 1

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

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

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

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

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

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

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

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

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

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

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

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

NOTE: to remedy the problem of coding and the method for entering values with 2/4 bytes floating point format, the user must convert the floating value to a 2/4 byte unsigned or signed value and then enter it in the ETS database selecting the "2/4 bytes signed/unsigned value" format.

5.1.5 Object sending delay (0... 255 seconds)

The parameter **Px"Object sending delay (0... 255 seconds)"** parameter sets the delay between the detection of the event matched 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 matched with the object with the previous index (z-1) and sending the command/value matched with the object to which the parameter refers; the delay in these cases is calculated from the moment when the command/value matched with the object with the previous index (z-1) is sent, not from the moment of detection of the event that generated the sending (closing/opening or short/long operation). The set delay will only be executed if the event in progress, matched with the object to which the parameter refers, is matched with any value other than **no action**; otherwise, the delay is ignored.
The parameter can 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 (closing/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 matched with the detection of the latter event is different from **no action**; otherwise, the command/value sequence will continue to be sent until the last command/value has been sent.

5.1.6 Cyclic sending object behaviour

Given the possibility to interface various devices with the device input contacts, it may be useful to repeat the command telegrams at pre-set intervals (especially if there is a sensor interface); The **Px "Cyclic sending object behaviour"** 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
- if close 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 recognized), the device will occasionally send the value matched with the event; if a new long operation is recognized, this cyclical sending is interrupted and the sending of the sequence matched with the detected operation restarts.
By selecting **if close 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 recognized), the device will occasionally send the value matched with the event; if a new short operation is recognized, the sending of the sequence matched 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 matched with the detected event is repeated at regular intervals. If a short/long operation is recognized, this cyclical sending is interrupted and the sending of the sequence matched with the detected operation restarts.
If the value **no action/stop cyclic sending object** is matched 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.

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

The parameter **Px "Sequence cyclic sending period if close contact/short operation [s]"** is used to set the repeat period for the sequence commands matched with the closed contact (or short operation) event; the values that can be set are:

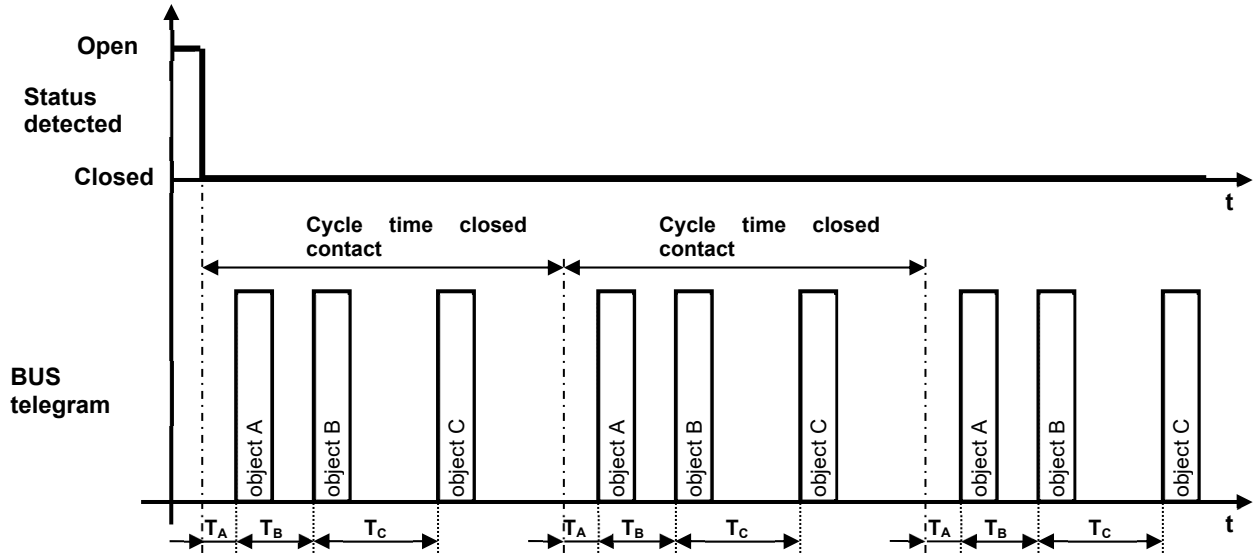
- from 1 to 65535 in steps of 1, **(default value 15)**

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

The parameter **Px "Sequence cyclic sending period if open contact/long operation [s]"** is used to set the repeat period for the sequence commands matched with the open contact (or long operation) event; the values that can be set are:

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

The cyclical sending time starts to be counted when the operation matched 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= Object sending delayA T_B= Object sending delay B T_C= Object sending delay C

The chart shows that, once the contact closing 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 (in relation to the sequence sending and the cyclical sending of telegrams) can be defined via the parameter **Px "Channel x behaviour at BUS voltage recovery"**. The values that can be set are:

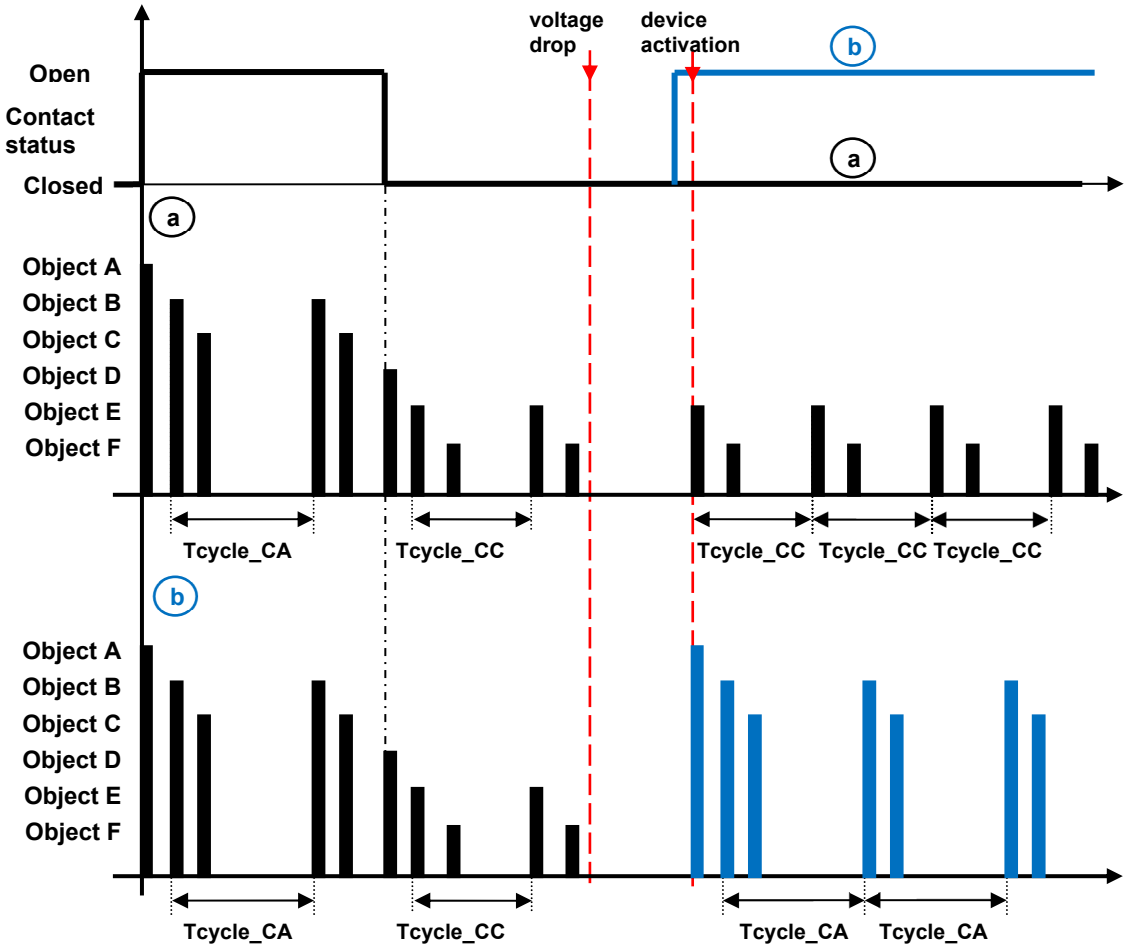
- **ignore contact status and cyclical sending (default value)**
- evaluate contact status and cyclical sending

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

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

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
- when the closed contact is detected, 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 drop, 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
- when the closed contact is detected, 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 drop, 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 closing is detected

6 “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.
It is possible to send 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 value does not depend on the commands sent from the device, but the latter regulates the brightness itself by stopping its increase/decrease based on the desired value. The communication objects that this function enables are ***Ch.x - Switching*** and ***Ch.x - Brightness control***.

The structure of the menu is as follows:

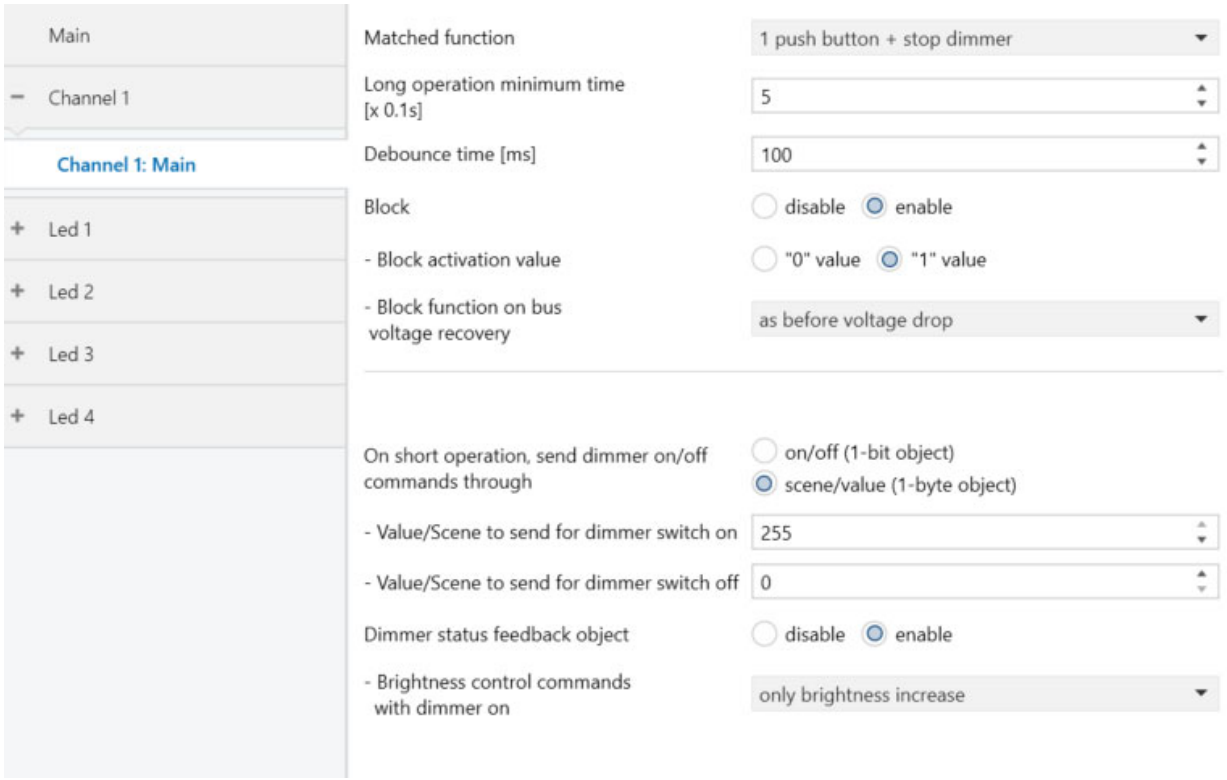


Fig 4: Setting ETS parameters – “Channel x” section - “1 push-button + stop dimmer” function

6.1 Parameters

The normal behaviour of the device foresees that if the command to be sent is the opposite of the last command sent, this is transformed into:

- long operation: if the last sent command was an off command or a decrease brightness command, the new command will be an increase brightness command; vice versa, if the last command was an on command or an increase brightness command, the new one 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 control increase/decrease commands in this case do not determine the value of the last command sent to distinguish the value of the new command to be sent.

6.1.1 Dimmer status feedback object

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

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

6.1.2 Brightness control commands with dimmer on

if **enable** is selected, the parameter **Px "Brightness control commands with dimmer on"** is made visible, 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 device sends depend on the parameter **Px "Brightness control commands with dimmer on"**, which can assume the following values:

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

  By setting **brightness increase and decrease**, if the value of the last two events "last sent command" and "dimmer status feedback" is ON, the new brightness 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.

If the feedback object is enabled, every time the BUS voltage is restored you must send a status read request on this object in order to update the device about the status of the devices connected.

The dimmer on and off commands are sent to the BUS through the object ***Ch.x - Switching*** and coded as:

- “1” = On, to switch on
- “0” = Off, to switch off

Alternatively, it is possible to select an alternative dimmer on/off format using the object ***Ch.x - Command value 1 byte*** after the contact short closing; in this way, it will be possible, for example, to send a percentage (appropriately selecting the value to send) or a scene (appropriately selecting the value/index of the scene to recall) to switch the dimmer on/off.

6.1.3 On short operation, send dimmer on/off commands through

The parameter **Px “On short operation, send dimmer on/off commands through”** is used to select the sending of the On/Off command or send two 1 byte unsigned values as dimmer on/off commands following a contact short closing; the values that can be set are:

- **On/Off (1-bit object)** **(default value)**
- Scene/Value (1-byte object)

selecting **On/Off (1-bit object)**, the object ***Ch.x - Switching*** is made visible and the on/off commands will be respectively “1”=“On” and “0”=“Off” of 1 bit; selecting **Scene/Value (1-byte object)**, the object ***Ch.x - Command value 1 byte*** is made visible and the on and off commands depend on the values configured in the new parameters **Px “Value/scene to send for dimmer switch on”** and **Px “Value/scene to send for dimmer switch off”**.

6.1.4 Value/scene to send for dimmer switch on

The parameter **Px “Value/scene to send for dimmer switch on”** is used to select the value to send to switch on the remote dimmer after the contact short closing; the values that can be set are:

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

6.1.5 Value/scene to send for dimmer switch off

The parameter **Px “Value/scene to send for dimmer switch off”** is used to select the value to send to switch off the remote dimmer after the contact short closing; the values that can be set are:

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

7 “Cyclic sending 1 push-button dimmer” function

This is used to configure the channel to control a dimmer with a single push-button, increasing and decreasing dimmer brightness always using the same push-button, with defined and settable regulation steps.
As there is only one 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 change 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 device that determines the percentage change. The communication objects that this function enables are ***Ch.x - Switching*** and ***Ch.x - Brightness control***.

The structure of the menu is as follows:

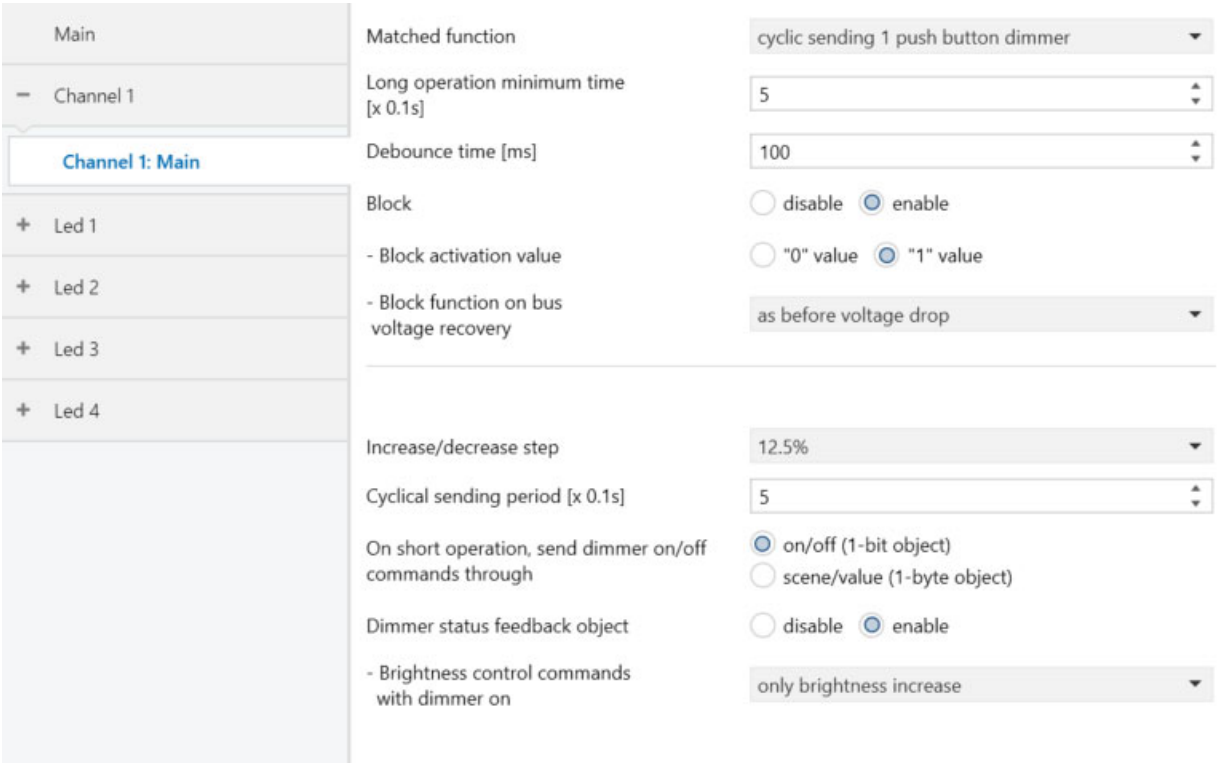


Fig 5: Setting ETS parameters – “Channel x” section - “1 push-button dimmer with cyclic sending” function

7.1 Parameters

7.1.1 Increase/decrease step

The parameter **Px "Increase/decrease step"** is used to set the percentage value of the brightness change matched 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%

7.1.2 Cyclical sending period [x 0.1s]

If the contact remains closed after recognizing the long operation, the device sends the command cyclically until contact opening is detected; The parameter **Px "Cyclical sending period [x 0.1s]"** 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 **Px "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:

if long operation minimum time is set to ***0.5 sec***, and the **Increase/decrease step** parameter is set to **12.5%** and the **Cyclical sending period [x 0.1s]** parameter is set to **3** (0.3 sec) and contact closing is detected:

- 0.5 seconds after the contact closing is detected, 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

7.1.3 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 **Px "Dimmer status feedback object"**, which can have the following values:

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

If **enable** is selected, the parameter **Px "Brightness control commands with dimmer on"** is made visible, along with the communication object ***Ch.x - Dimmer status feedback***, which makes it possible to receive status feedback from the controlled dimmer actuator.

The parameter **Px "Brightness control commands with dimmer on"** can have the following values:

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

If the feedback object is enabled, every time the BUS voltage is restored you must send a status read request on this object in order to update the device about the status of the devices connected.

The dimmer on and off commands are sent to the BUS through the object ***Ch.x - Switching*** and coded as:

- "1" = On, to switch on
- "0" = Off, to switch off

Alternatively, it is possible to select an alternative dimmer on/off format using the object ***Ch.x - Command value 1 byte*** after the contact short closing, in this way it will be possible to send for example a percentage (appropriately selecting the value to send) or a scene (appropriately selecting the value/index of the scene to recall) to switch the dimmer on/off.
The parameter **Px "On short operation, send dimmer on/off commands through"** is used to select the sending of the On/Off command or send two 1 byte unsigned values as dimmer on/off commands following a contact short closing; the values that can be set are:

- **On/Off (1-bit object) (default value)**
- Scene/Value (1-byte object)

selecting **On/Off (1-bit object)**, the object ***Ch.x - Switching*** is made visible and the on/off commands will be respectively "1"="On" and "0"="Off" of 1 bit; selecting **Scene/Value (1-byte object)**, the object ***Ch.x - Command value 1 byte*** is made visible and the on and off commands depend on the values configured in the new parameters **Px "Value/scene to send for dimmer switch on"** and **Px "Value/scene to send for dimmer switch off"**.

The parameter **Px "Value/scene to send for dimmer switch on"** is used to select the value to send to switch on the remote dimmer after the contact short closing; the values that can be set are:

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

The parameter **Px "Value/scene to send for dimmer switch off"** is used to select the value to send to switch off the remote dimmer after the contact short closing; the values that can be set are:

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

8 "1 push-button roller shutter control" function

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

As only one channel manages the louvre up/down and control functions, operation is managed so that with each activation, a command is sent that is the opposite to the last movement signal received by the actuator that manages the shutter. 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 ***Ch.x - Shutter movement*** object or the movement feedback received via the ***Ch.x - Movement feedback*** object, depending on which of the two events occurred last; if the last event that occurred is “upward movement feedback reception” or “send upward movement command”, the command will be a "closing louvres adjustment" command, and vice versa. If the shutter is moving, the louvre adjustment command will only stop the shutter up/down movement.

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

The structure of the menu is as follows:

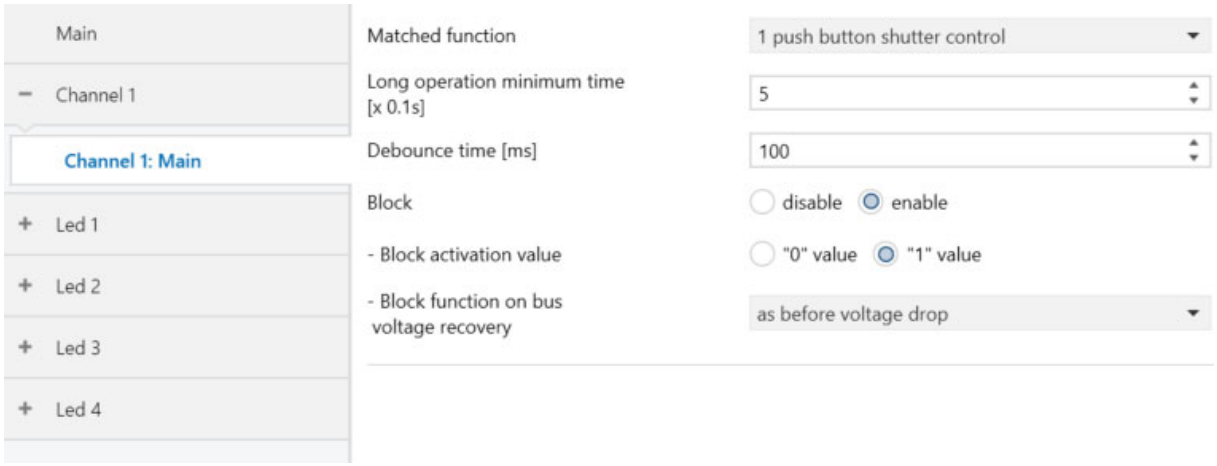


Fig 6: Setting ETS parameters – “Channel x” section - “1 push-button roller shutter control” function

No new parameters are enabled with this function.

9 “2 push-button + stop dimmer” function

This is used to configure the channel to control a dimmer with two push-buttons, managing in this case only one of the two control directions (brightness increase/decrease).
On or off telegrams and brightness increase or decrease dimming telegrams can be sent, based on the configured control direction. Also in this case, there is a difference between short and long operations:

- a long operation is interpreted as a brightness control command. If the set control direction is "increase", the control will only be increasing, otherwise if the set control direction is "decrease" the control will be decreasing. In both cases, when reopening the contact, an adjustment stop telegram is sent to stop the brightness increase or decrease operation for the dimmer and to fix the brightness value reached at the moment the stop control command was received.

- a short operation is transformed in to an on or off command depending on the set control direction. If the set control direction is "increase" the sent command will only be an ON command. If the set control direction is "decrease" the sent command will only be an 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 that this function enables are ***Ch.x - Switching*** and ***Ch.x - Brightness control***.

The structure of the menu is as follows:

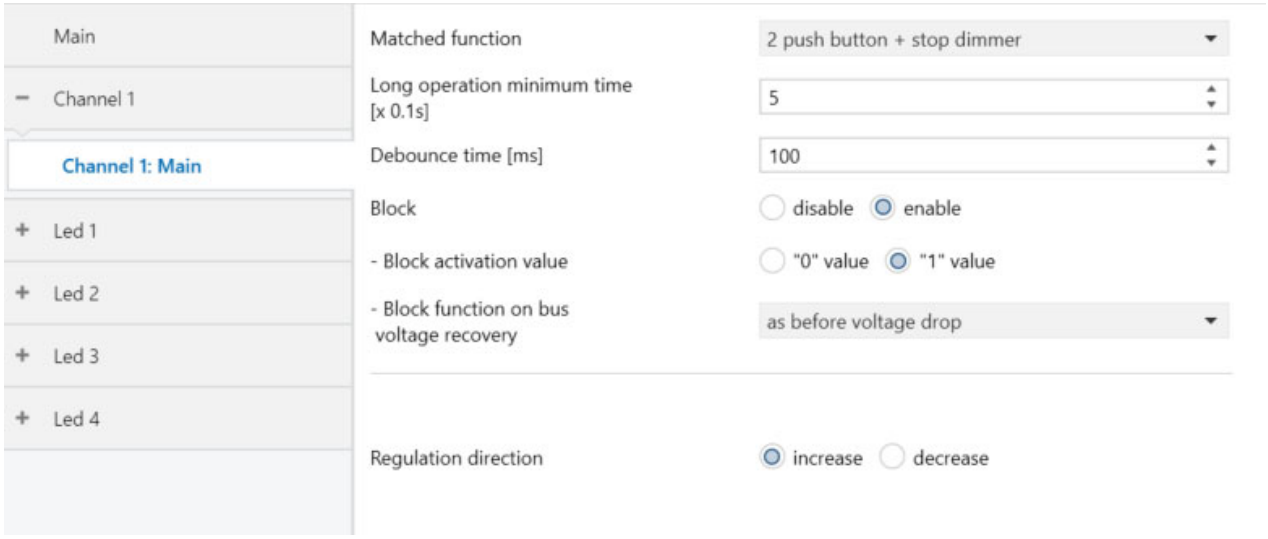


Fig 7: Setting ETS parameters – “Channel x” section - “2 push-button + stop dimmer” function

9.1 Parameters

9.1.1 Regulation direction

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

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

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

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

This is used to configure the channel to control a dimmer with two push-buttons, managing in this case only one of the two control directions (brightness increase/decrease).
On or off telegrams and brightness increase or decrease dimming telegrams can be sent, based on the configured control direction. Also in this case, there is a difference between short and long operations:

- a long operation is interpreted as a brightness control command. If the set control direction is "increase", the control will only be increasing, otherwise if the set control direction is "decrease" the control will be decreasing. In both cases, no telegram is sent when the contact is opened again.

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

Unlike the ***2 push-button + stop dimmer*** function, it is possible to define both the brightness change 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 adjustment stop telegram on push-button release is not therefore necessary, because although the control does follow the characteristic power/brightness curve, it is the command sent by the device that determines the percentage change. The communication objects that this function enables are ***Ch.x - Switching*** and ***Ch.x - Brightness control***.

The structure of the menu is as follows:

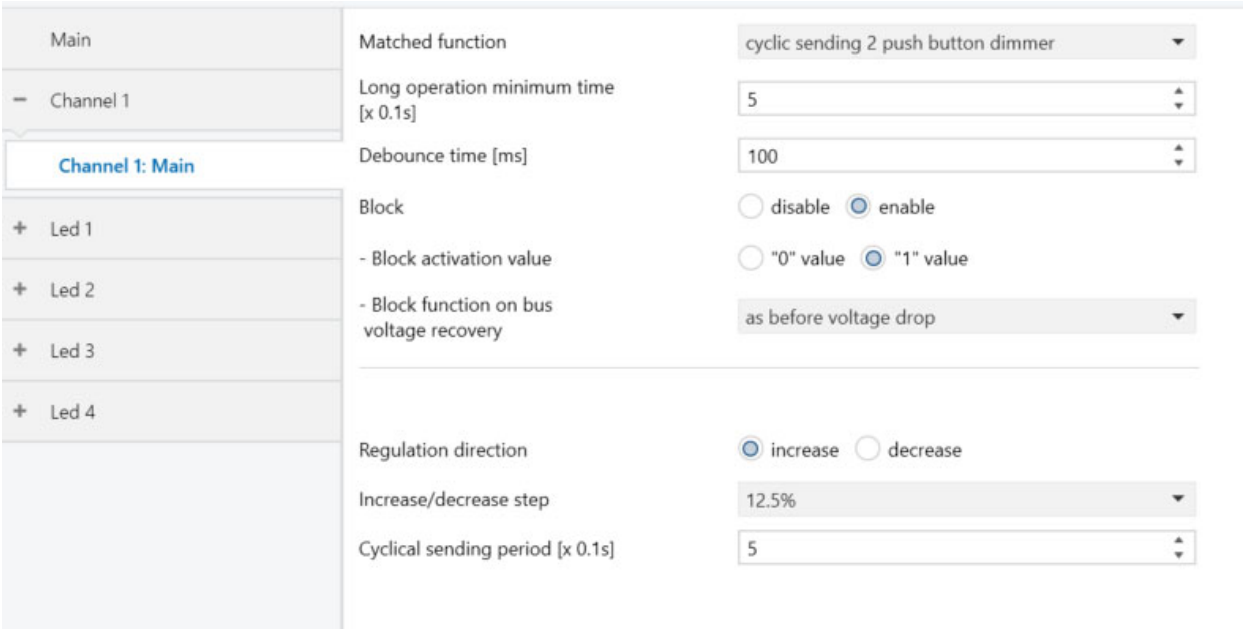


Fig 8: Setting ETS parameters – “Channel x” section - “2 push-button dimmer with cyclic sending” function

10.1 Parameters

10.1.1 Regulation direction

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

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

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

10.1.2 Increase/decrease step

The parameter **Px "Increase/decrease step"** is used to set the percentage value of the brightness change matched 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%

10.1.3 Cyclical sending period [x 0.1s]

If the contact remains closed after recognizing the long operation, the device sends the command cyclically until contact opening is detected; The parameter **Px "Cyclical sending period [x 0.1s]"** 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 **Px "Cyclical sending period [x 0.1s]"** are:

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

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:

if long operation minimum time is set to ***0.5 sec***, and the **Increase/decrease step** parameter is set to **12.5%** and the **Cyclical sending period [x 0.1s]** parameter is set to **3** (0.3 sec) and contact closing is detected:

- 0.5 seconds after the contact closing is detected, 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

11 “2 push-button shutter control” function

This is used to configure the channel to control a shutter/Venetian blind with two buttons, managing in this case only one of the two movement directions (down or up).
Up or down movement telegrams or louvres open or close control telegrams can be sent.
Also in this case, there is a difference between short and long operations:

- a long operation is transformed into a movement command. If the set movement direction is "up", the movement will only be up; vice versa if the set direction is "down" the movement will be down. When the contact reopens, the device does not perform any action.

- a short operation is transformed into a louvres control command (stop movement if the shutter is moving), depending on the set movement direction. If the set movement direction is "up", the sent command will only be a slat opening control command (or stop movement); If the set regulation direction is "down", the sent command will only be a slat closing control command (or stop movement).

The communication objects that this function enables are ***Ch.x - Shutter movement*** and ***Ch.x - Shutter stop/adjustment***.

The structure of the menu is as follows:

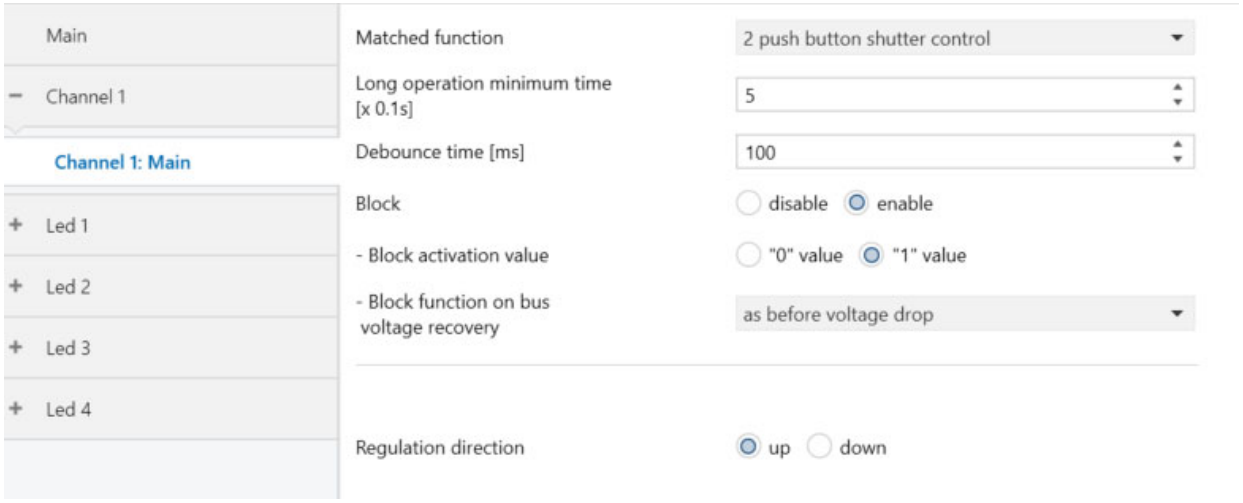


Fig 9: Setting ETS parameters – “Channel x” section - “2 push-button roller shutter control” function

11.1 Parameters

11.1.1 Regulation direction

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

- **up** **0 (default value odd channel)**
- **down** **1 (default value even channel)**

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

12 “Scene management” function

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

- a long operation is interpreted as a scene storing command.
- a short operation is interpreted as a scene execution command.

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

The structure of the menu is as follows:

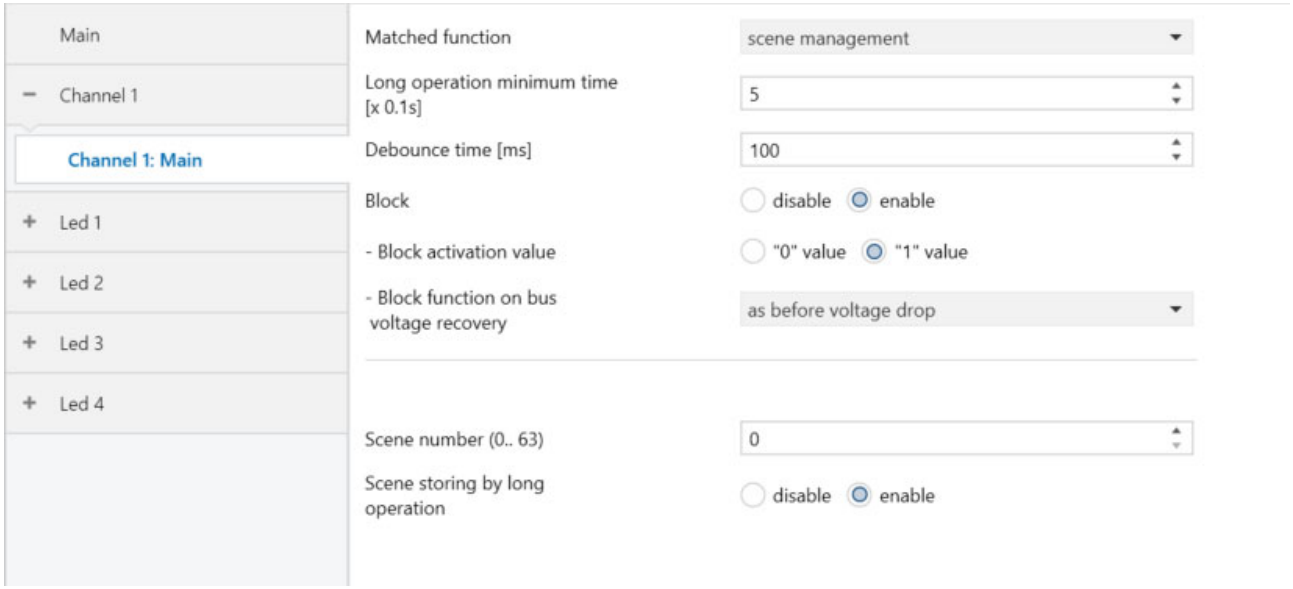


Fig 10: Setting ETS parameters – "Channel x" section - "scene management" function

12.1 Parameters

12.1.1 Scene number (0.. 63)

The parameter **Px "Scene number (0... 63)"** 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, in steps of 1

12.1.2 Scene storing by long operation

The parameter **Px "Scene storing by long operation"** enables the sending of a scene memorising command when a long press is recognized. 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 recognized 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.

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:

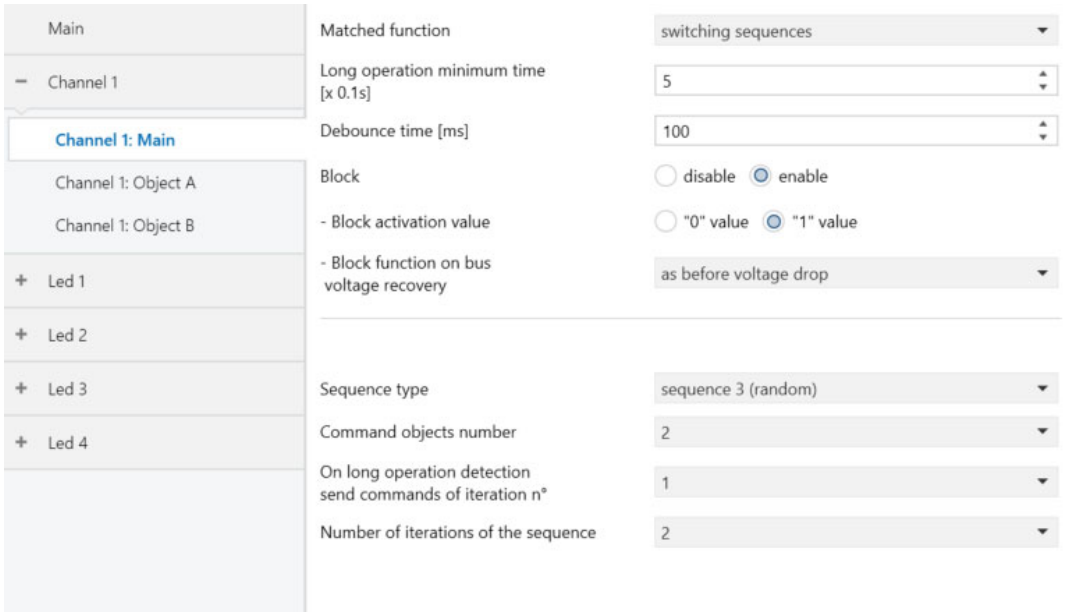


Fig 11: Setting ETS parameters – “Channel x” section - “switching sequences” function

13.1 Parameters

13.1.1 Number of objects to send

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

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

13.1.2 Sequence type

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

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

Once the 8th edge is detected, the sequence will 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 **Px "Number of iterations of the sequence"** and the configuration menu **object z channel j** (one for each enabled command). The parameter **Px "Number of iterations of the sequence"** 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, in steps of 1

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

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

14 “Channel x object z” menu

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

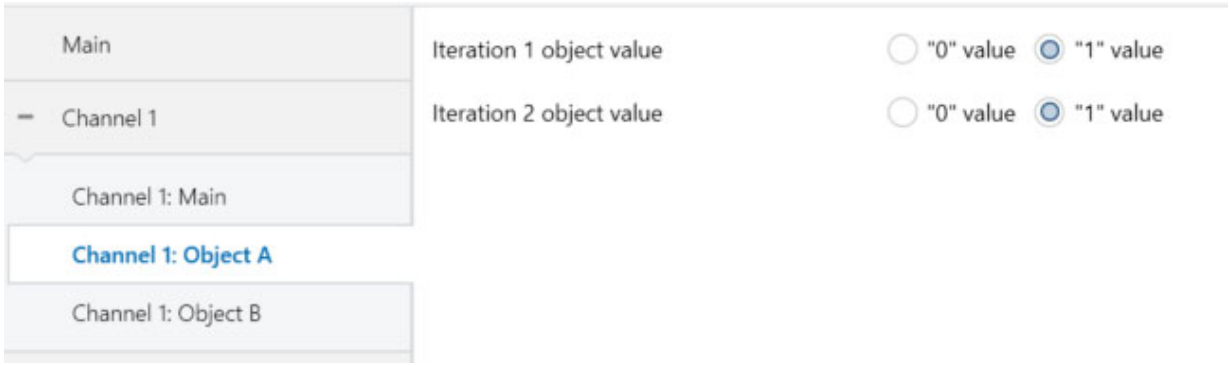


Fig 12: Setting ETS parameters – “Channel x object z” section

14.1 Parameters

14.1.1 On long operation detection send commands of iteration n°

Regardless of the type of sequence selected, the parameter **Px** “**On long operation detection send commands of iteration n°”** 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 16 in steps of 1, **(default value 1)**

EXAMPLE:

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

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

"Filling" sequence

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

"Sum" sequence

Once 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 matched with the long operation will be sent (in the example given here, the sequence matched with iteration no. 4 will be sent).

To sum up, the value set for the parameter **Px "On long operation detection send commands of iteration n°"** 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 matched with the sequence to be sent with a long press is less than - or equal to - the maximum number of iterations matched with the sequence; otherwise, the iteration to be taken into consideration is the limit one.

15 “Pulse counter” function

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

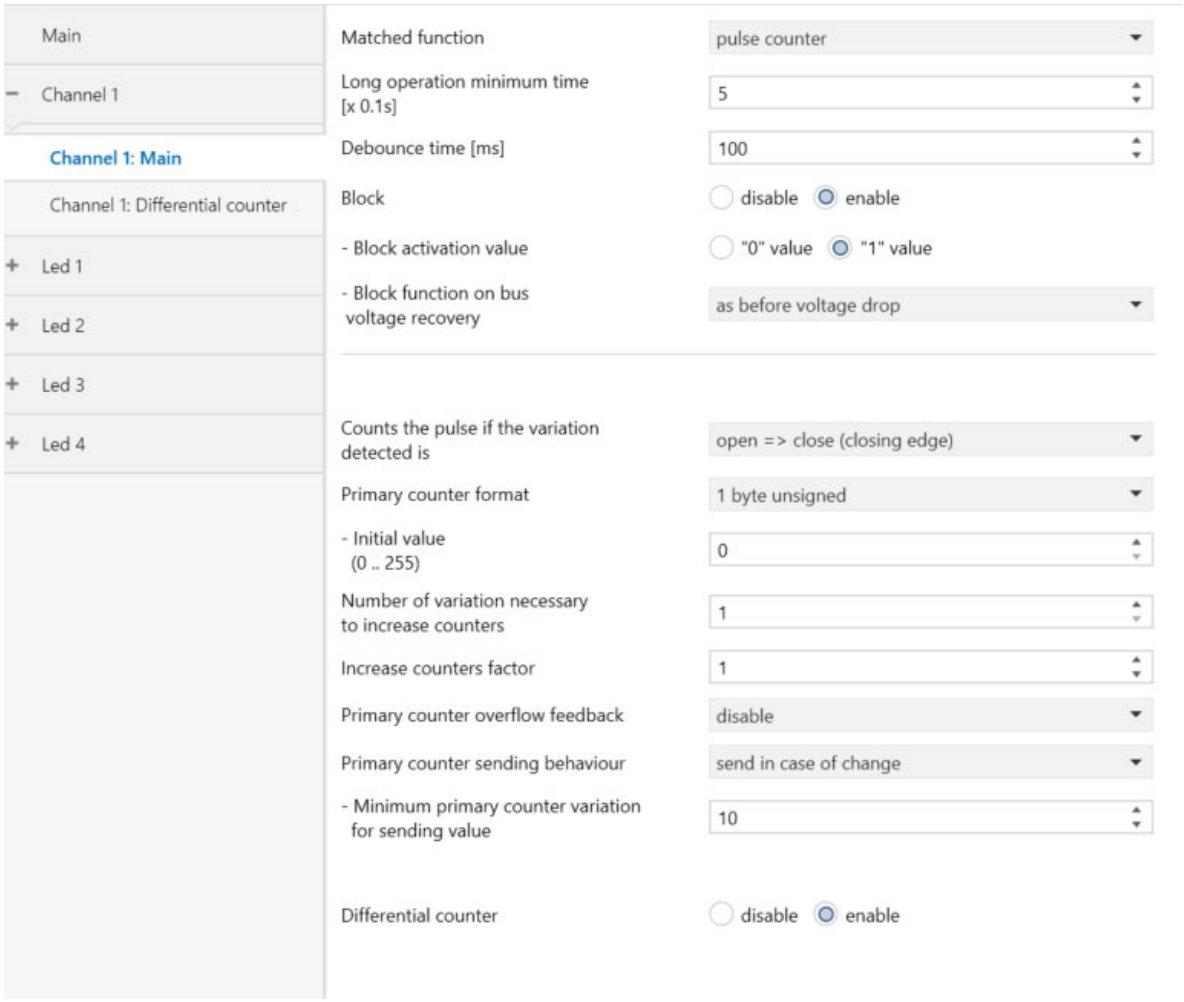


Fig 13: Setting ETS parameters – “Channel x” section - “pulse counter” function

15.1 Parameters

15.1.1 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 closing and opening; The parameter **Px “Counts the pulse if the variation detected is”** is used to set the type of contact status change 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 change from open contact to closed contact (closing edge) will be considered by the device as a pulse, so it is this that produces an increase in the count value; the opposite status change will have no effect.

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

15.1.2 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 parameter **Px "Primary counter format"**, 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 unsigned value** **(default value)**
- 1 byte signed value
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value

15.1.3 Initial value

The parameter **Px "Initial value"** 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 parameter **Px "Primary counter format"**, the values that can be set for this item will be different:

- If the primary counter format is **1 byte unsigned value**, the communication object ***Ch.x - Primary counter*** will be visible and the values that can be set for the above parameter are:
  - from **0 (default value)** to 255, in steps of 1
- If the primary counter format is **1 byte signed value**, the communication object ***Ch.x - Primary counter*** will be visible and the values that can be set for the above parameter are:
  - from -128 to 127 with steps of 1 **(default value 0)**
- If the primary counter format is **2 bytes unsigned value**, the communication object ***Ch.x - Primary counter*** will be visible and the values that can be set for the above parameter are:
  - from **0 (default value)** to 65535, in steps of 1
- If the primary counter format is **2 bytes signed value**, the communication object ***Ch.x - Primary counter*** will be visible and the values that can be set for the above parameter are:
  - from -32768 to +32767 with steps of 1 **(default value 0)**
- If the primary counter format is **4 bytes unsigned value**, the communication object ***Ch.x - Primary counter*** will be visible and the values that can be set for the above parameter are:
  - from **0 (default value)** to 4294967295, in steps of 1
- If the primary counter format is **4 bytes signed value**, the communication object ***Ch.x - Primary counter*** will be visible and the values that can be set for the above parameter are:
  - from -2147483648 to 2147483647 with steps of 1 **(default value 0)**

15.1.4 Number of variation necessary to increase counters

The parameter **Px "Number of variation necessary to increase counters"** 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, in steps of 1

15.1.5 Increase counters factor

The parameter **Px "Increase counters factor"** 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 variation necessary to increase counters). 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 parameters **"Number of variation necessary to increase counters"** and **Px "Increase counters factor"**, let's consider the case where the increase factor is 2 and the number of variation necessary to increase counters 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.

15.1.6 Primary counter overflow feedback

The parameter **Px "Primary counter overflow feedback"** 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:

- **disable** **0 (default value)**
- enable 1-bit object 1
- enable 1-bit and 1-byte objects 2

selecting a value other than **disabled**, the communication object ***Ch.x – Primary counter overflow bit*** is made visible, through which the device signals the primary counter overflow; when the overflow occurs, a value of "1" is sent; a value of "0" is never sent.
Selecting the value **enable 1-bit and 1-byte objects**, the communication object ***Ch.x – Primary counter overflow byte*** is made visible, through which the device signals the primary counter overflow; When the overflow occurs, the value defined by the new parameter **Px "Send the value with primary counter overflow"** is sent. This parameter may assume the following values:

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

Once the maximum (or minimum) value has been reached, the primary counter restarts from the value set in **Px "Initial value"**.
If the value set in **Px "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 byte unsigned* 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).

15.1.7 Primary counter sending behaviour

This parameter **Px "Primary counter sending behaviour"** 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 in case of change and periodically

selecting a value other than **send on demand only**, the ***Ch.x – Primary counter sending trigger*** communication object is made visible.

Selecting the value **send in case of change** or **send in case of change and periodically**, displays the parameter **Px "Minimum primary counter variation for sending value"** whereas selecting the value **send periodically** or **send in case of change and periodically** displays the parameter **Px "Primary counter sending period (seconds)"**.
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.

15.1.8 Minimum primary counter variation for sending value

The parameter **Px "Minimum primary counter variation for sending value"**, is visible if the primary counter value is sent on 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 in steps of 1, **(default value 10)**

15.1.9 Primary counter sending period (seconds)

The parameter **Px "Primary counter sending period (seconds)"** 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 in steps of 1, **(default value 15)**

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

15.1.10 Differential counter

The parameter **Px "Differential counter"** is used to enable the display - and hence the use - of the ***Ch.x - Differential counter*** communication object, and makes the ***Channel x - Differential counter*** configuration menu visible (see paragraph "Channel x - Differential counter menu").

The differential counter is a counter that, in contrast to the primary 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

16 “Channel x – differential counter” menu

Used to configure a second counter for counting the number of contact status changes (edges) by setting the parameters that characterise the differential count.

The structure of the menu is as follows:

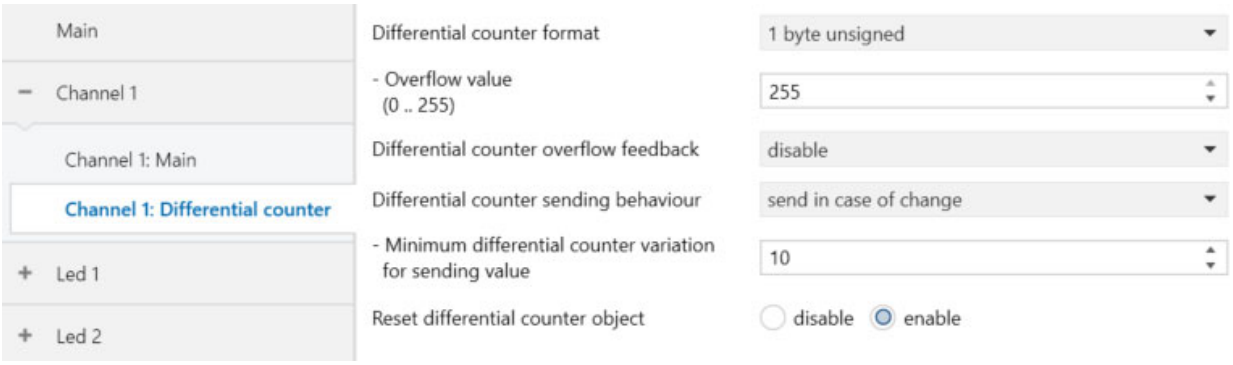


Fig 14: Setting ETS parameters – “Channel x” section - “Channel x - Differential counter” menu

16.1 Parameters

16.1.1 Differential counter format

The differential counter used for the pulse count must be of a sufficient capacity to count the maximum required number of pulses. With the parameter **Px “Differential counter format”**, 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 unsigned value** **(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.

16.1.2 Overflow value

The parameter **Px “Overflow value”** 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 parameter **Px “Differential counter format”**, the values that can be set for this item will be different:

- If the differential counter format is **1 byte unsigned value**, the communication object ***Ch.x - Differential counter*** will be visible and the values that can be set for the above parameter are:
  - from 0 to **255 (default value)** with steps of 1
- If the differential counter format is **1 byte signed value**, the communication object ***Ch.x - Differential counter*** will be visible and the values that can be set for the above parameter are:
  - from -128 to **127 (default value)** with steps of 1

- If the differential counter format is **2 bytes unsigned value**, the communication object ***Ch.x - Differential counter*** will be visible and the values that can be set for the above parameter are:
  - from 0 to **65535 (default value)** with steps of 1
- If the differential counter format is **2 bytes signed value**, the communication object ***Ch.x - Differential counter*** will be visible and the values that can be set for the above parameter are:
  - from -32768 to **+32767 (default value)** with steps of 1
- If the differential counter format is **4 bytes unsigned value**, the communication object ***Ch.x - Differential counter*** will be visible and the values that can be set for the above parameter are:
  - from 0 to **4294967295 (default value)** with steps of 1
- If the differential counter format is **4 bytes signed value**, the communication object ***Ch.x - Differential counter*** will be visible and the values that can be set for the above parameter are:
  - from -2147483648 to **2147483647 (default value)** with steps of 1

16.1.3 Differential counter overflow feedback

The parameter **Px "Differential counter overflow feedback"** 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:

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

selecting a value other than **disabled**, the communication object ***Ch.x – Differential counter overflow bit*** is made visible, through which the device signals the differential counter overflow; when the overflow occurs, a value of "1" is sent; a value of "0" is never sent.

Selecting the value **enable 1-bit and 1-byte objects**, the communication object ***Ch.x – Differential counter overflow byte*** is made visible, through which the device signals the differential counter overflow; When the overflow occurs, the value defined by the new parameter **Px "Send the value with differential counter overflow"** is sent. This parameter may assume the following values:

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

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

16.1.4 Increase counters factor

If the value set in **Px "Increase counters factor"** of the **Channel j** 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).

16.1.5 Differential counter sending behaviour

The parameter **Px "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:

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

selecting a value other than **send on demand only**, the ***Ch.x – Differential counter sending trigger*** communication object is made visible.

Selecting the value **send in case of change** or **send in case of change and periodically**, displays the parameter **Px "Minimum differential counter variation for sending value"** whereas selecting the value **send periodically** or **send in case of change and periodically** displays the parameter **Px "Differential counter sending period"**.

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 to indirectly generate 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.

16.1.6 Minimum differential counter variation for sending value

This parameter **Px "Minimum differential counter variation for sending value"** 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 in steps of 1, **(default value 10)**

16.1.7 Differential counter sending period (seconds)

The parameter **Px "Differential counter sending period (seconds)"** 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 in steps of 1, **(default value 15)**

16.1.8 Reset differential counter object

The parameter **Px "Reset differential counter object"** is used to enable the viewing and hence the use of the communication object ***Ch.x - Reset differential counter***, to receive the differential counter reset command from the BUS to reset the differential counter value.
The values that can be set are:

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

selecting **enable**, the ***Ch.x – Reset differential counter*** communication object is made visible, via which the

device receives the differential counter reset command; If a value of “1” or "0" is received, the differential counter is re-initialised at 0.
In the event of a BUS voltage drop, the differential counter value must be saved in a non-volatile memory and restored when the BUS voltage is recovered.

17 "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.
The structure of the menu is as follows:

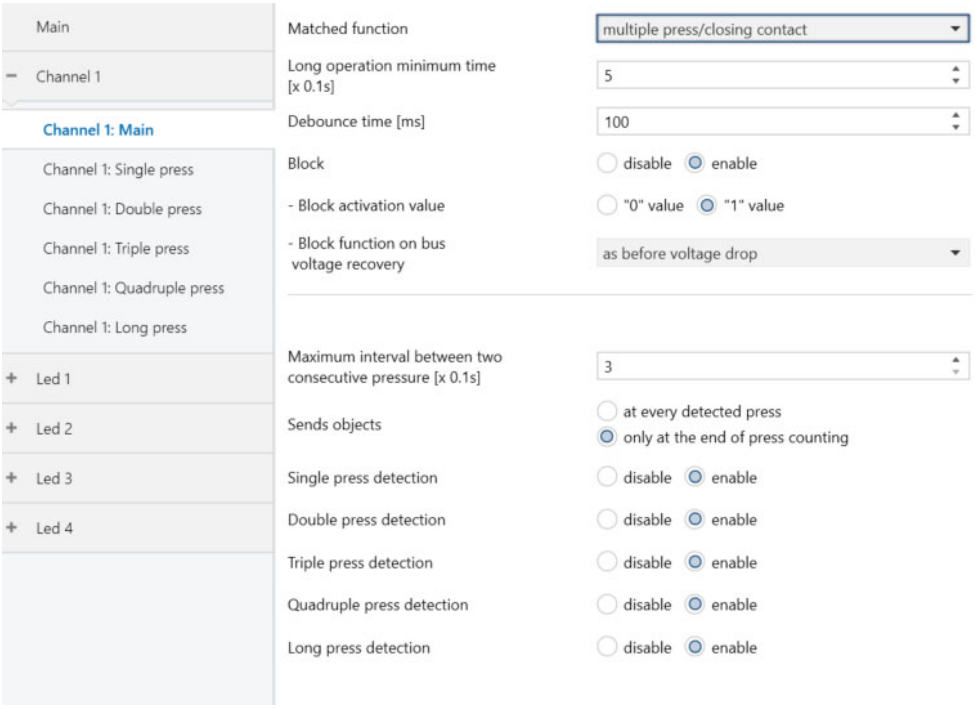


Fig 15: Setting ETS parameters – "Channel x" section - "multiple press/closing contact" function

17.1 Parameters

In this mode, every channel can send a series of KNX telegrams following the detection of several consecutive contact pressing operations; a pressing is recognized when the contact re-opens after closing (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 closing

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

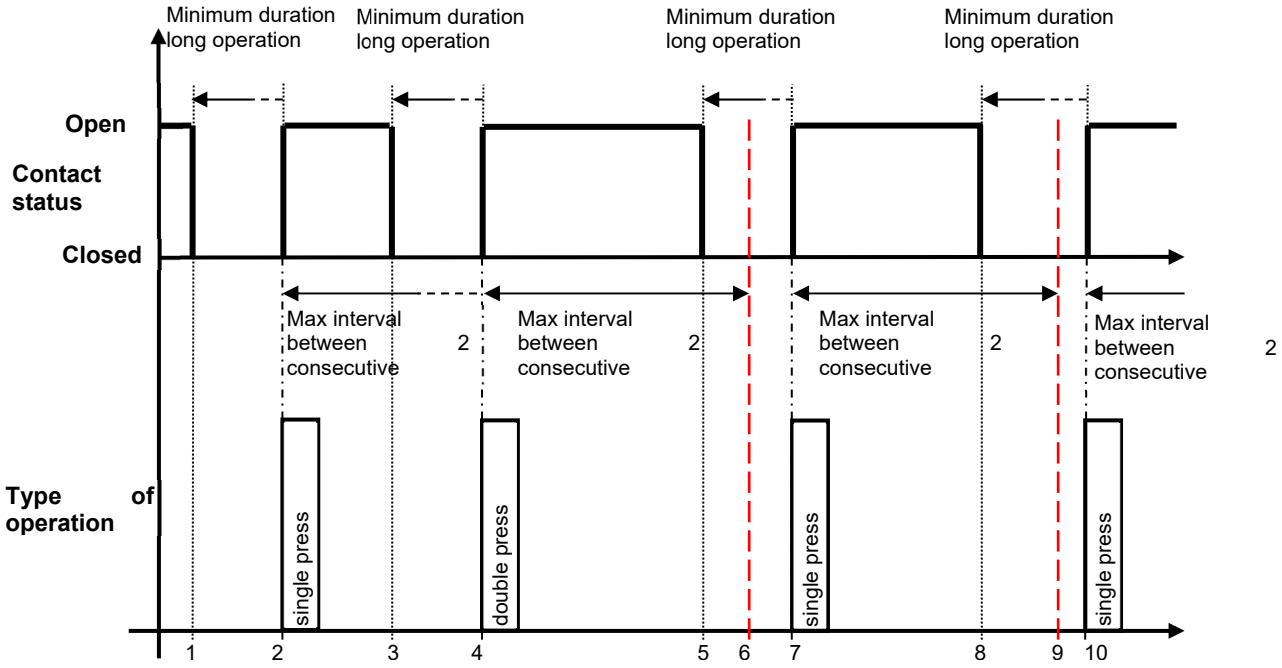
In order to recognize two consecutive operations, it's necessary to define the maximum gap between the detection of one press and the next; if the time between two presses (not counting the debounce time) is less than the maximum gap, the count of multiple presses is increased. When the time that elapses after the detection of a pressing (not counting the debounce time) exceeds the maximum gap, the device recognizes a number of consecutive multiple presses equal to the value counted and, after sending the telegrams matched with this action, it resets their counter.

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

The parameter **Px "Maximum interval between two consecutive pressure [x 0.1s]**" is used to define the maximum gap between the detection of one press and the next, so that they are recognized 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).



1. Once the contact closing has been detected, the contact closing 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 recognized and the count of the interval 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 interval 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 interval between two consecutive pressure 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 interval between two consecutive presses and the multiple press count;

6. Once the maximum interval between two consecutive pressure (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;

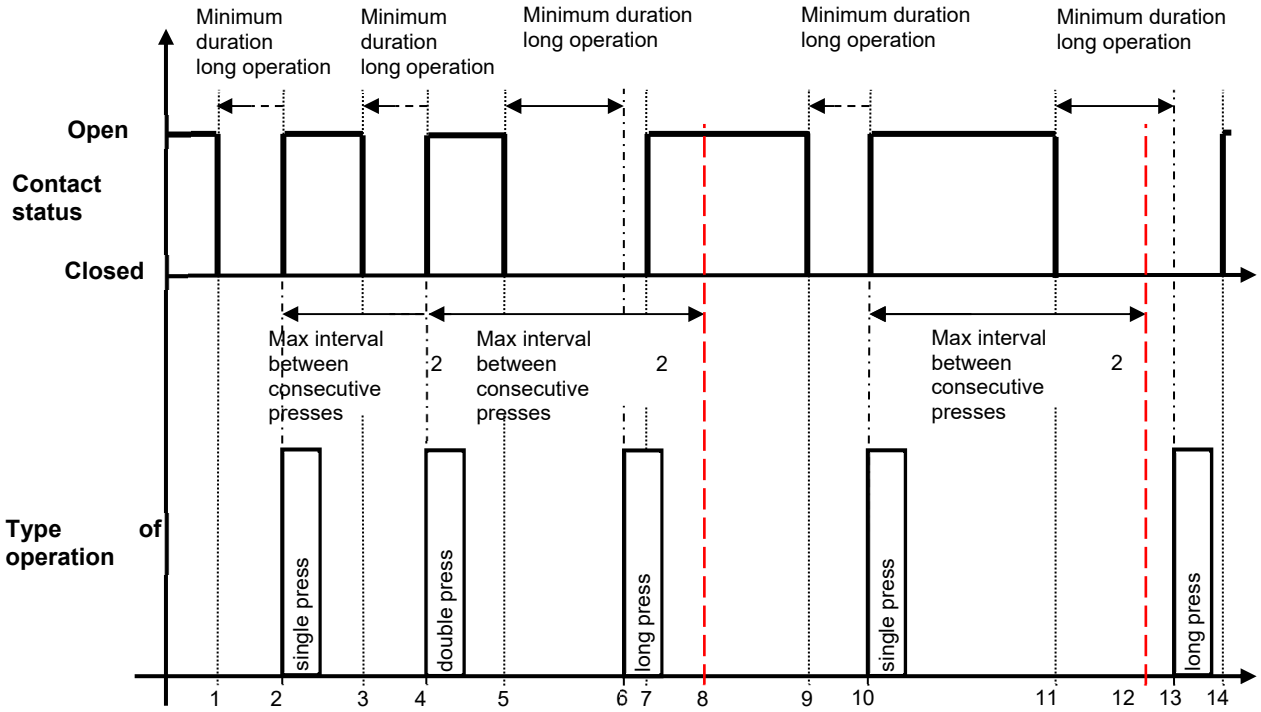
7. The re-opening of the contact before the long operation time means the detection of a new short press that increases the multiple press count and initialises the count of the interval between two consecutive presses;

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 interval between two consecutive presses and the multiple press count;

9. Once the maximum interval between two consecutive pressure (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 interval 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 interval between two consecutive pressure. See below (the debounce time is not shown).



1. Once the contact closing has been detected, the contact closing 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 recognized and the count of the interval 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 interval 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 interval between two consecutive pressure 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 interval 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 recognized and the KNX commands for that action are sent, but neither calculation of the interval

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 interval between two consecutive pressure (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 interval 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 interval 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 interval between two consecutive presses and the multiple press count;

12. Once the maximum interval between two consecutive pressure (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 recognized and the KNX commands for that action are sent, but neither calculation of the interval 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.

17.1.2 Single press detection

The parameter **Px "Single press detection"** is used to enable the detection of a single press, and to make the **Channel x - Single press** menu visible for enabling and configuring the commands that will be sent following the detection 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).

17.1.3 Double press detection

The parameter **Px "Double press detection"** is used to enable the detection of a double press, and to make the **Channel x - Double press** menu visible for enabling and configuring the commands that will be sent following the detection 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).

17.1.4 Triple press detection

The parameter **Px "Triple press detection"** is used to enable the detection of a triple press, and to make the **Channel x - Triple press** menu visible for enabling and configuring the commands that will be sent following the detection 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).

17.1.5 Quadruple press detection

The parameter **Px “Quadruple press detection”** is used to enable the detection of a quadruple press, and to make the **Channel x - Quadruple press** menu visible for enabling and configuring the commands that will be sent following the detection 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).

17.1.6 Long press detection

The parameter **“Long press detection”** is used to enable the detection of a long press, and to make the **Channel x - Long press** menu visible for enabling and configuring the commands that will be sent following the detection 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).

The commands matched 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 matched with the number of presses detected;

b) every time the multiple press count is increased, the device sends the telegrams matched with the number of presses detected.

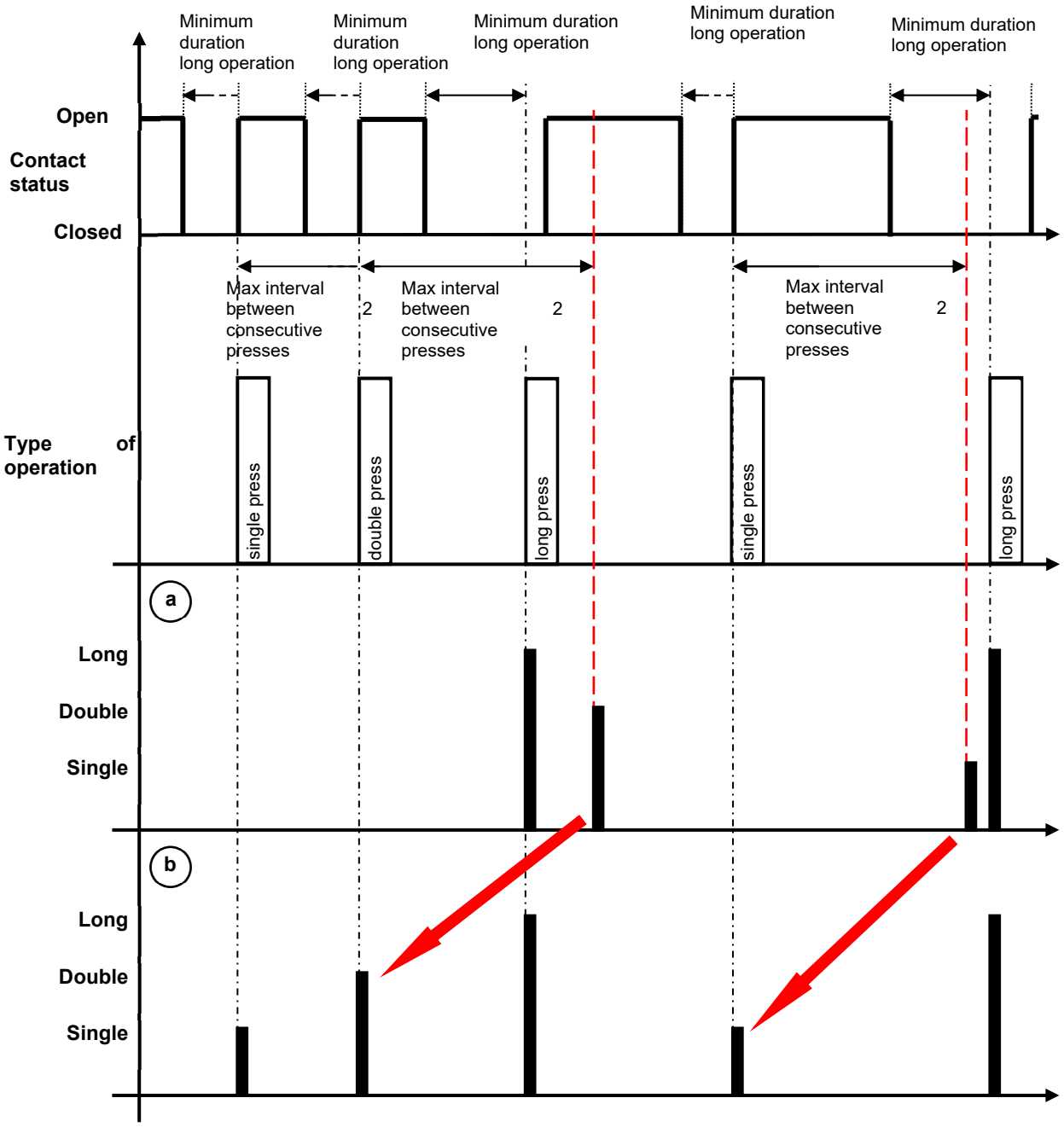
17.1.7 Sends objects

The commands matched with a “long press” are always sent as soon as the long press is detected.
The parameter **Px “Sends objects”** is used to define the sending conditions of the objects matched with multiple presses; the values that can be set are:

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

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



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 counting** (case “a”) or **at every detected press** (case “b”). The main difference between the two cases is that in case “b”, every time a multiple press is counted, the matched 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 matched with the last press detected.
The red arrows highlight the differences between the moments when the telegrams matched with the same long presses are actually sent.

18 “Channel x – Single press” menu

This menu is visible if the value of the parameter **Px “Single press detection”** 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 - matched with the “single press” event.
The structure of the menu is as follows:

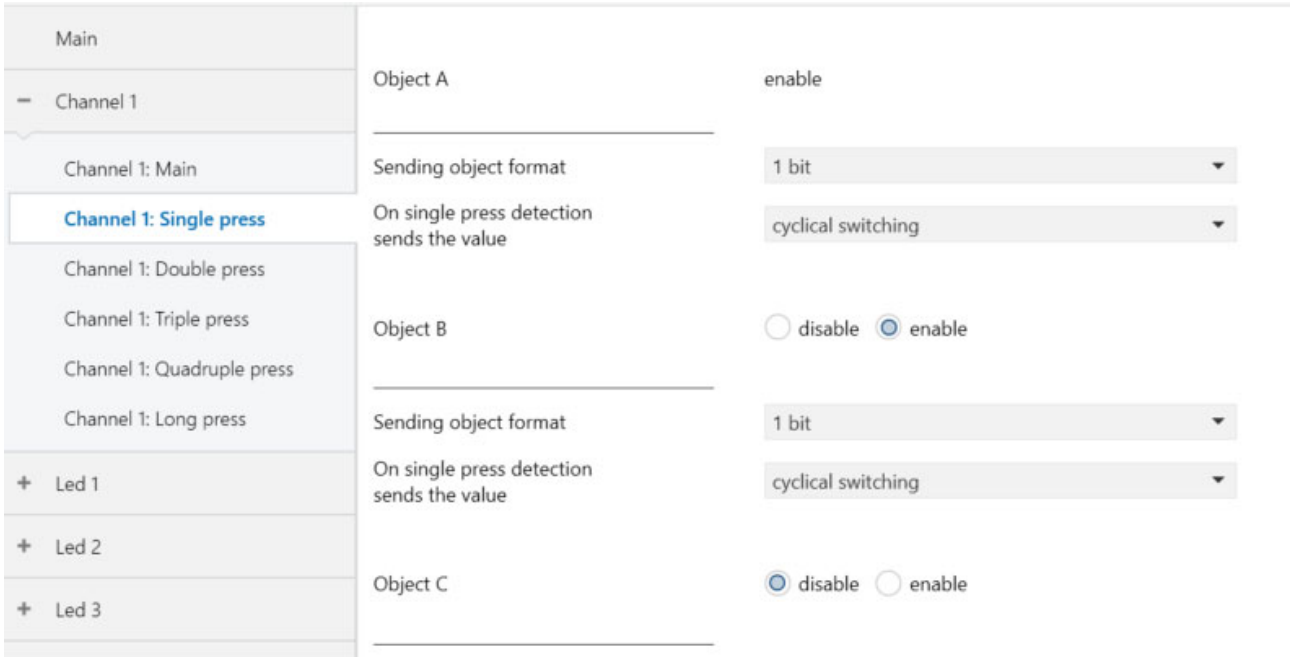


Fig 16: Setting ETS parameters – “Channel x” section - “Channel x - Single press” menu

18.1 Parameters

18.1.1 Object z

Upon detection of a single press, it is possible to send up to 8 different objects (which are distinguished by the letters A, B, C, D, E, F, G and H); Object A is always enabled, and the parameter **“object z”** (z indicates the object matched with the threshold, included between **B** and **H)** is used to enable a new object to be sent. The parameter can assume the following values:

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

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

18.1.2 Sending object format

The parameter **Px “Sending object format”** 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 bits
- 1 byte unsigned value
- 1 byte signed value
- 1 byte percentage value

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

Depending on the value set for this item, the values that can be set for the parameter **Px "On single press detection sends the value"** will be different.

18.1.3 On single press detection sends the value

The parameter **Px "On single press detection sends the value"** 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) matched with the channel. The values that can be set are:

- If the sending object format is **1 bit**, the communication object ***Ch.x - single press 1 bit object z*** will be visible and the values that can be set for the above parameter are:

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

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

- If the sending object format is **2 bits**, the communication object ***Ch.x - single press 2 bits object z*** will be visible and the values that can be set for the above parameter are:

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

  selecting **cyclical switching**, the command that the device will send (via the object ***Ch.x - Single press 2-bits object z***) when the event matched with the cyclical switching detected will be the opposite of the last value sent (via the object ***Ch.x - Single press 2 bits object z***).

- If the sending object format is **1 byte unsigned value**, the communication object ***Ch.x - Single press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte signed value**, the communication object ***Ch.x - Single press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte percentage value**, the communication object ***Ch.x - Single press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte HVAC mode**, the communication object ***Ch.x - Single press 1 byte object z*** will be visible 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 matched 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 matched 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 sending object format is **2 bytes unsigned value**, the communication object ***Ch.x - Single press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **2 bytes signed value**, the communication object ***Ch.x - Single press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **3 bytes RGB colour**, the parameter **Px "On single press detection sends the value"** 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). Furthermore, the communication object ***Ch.x - Single press 3 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

  by selecting **customize**, the following parameters are made visible: **Px "Value of RED component (0 ... 255)"**, **Px "Value of GREEN component (0 ... 255)"** and **Px "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, in steps of 1

- If the sending object format is **4 bytes unsigned value**, the communication object ***Ch.x - Single press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **4 byte signed value**, the communication object ***Ch.x - Single press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

NOTE: to remedy the problem of coding and the method for entering values with 2/4 bytes floating point format, the user must convert the floating value to a 2/4 byte unsigned or signed value and then enter it in the ETS database selecting the “2/4 bytes signed/unsigned value” format.

19 “Channel x – Double press” menu

This menu is visible if the value of the parameter **Px “Double press detection”** 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 - matched with the “double press” event.
The structure of the menu is as follows:

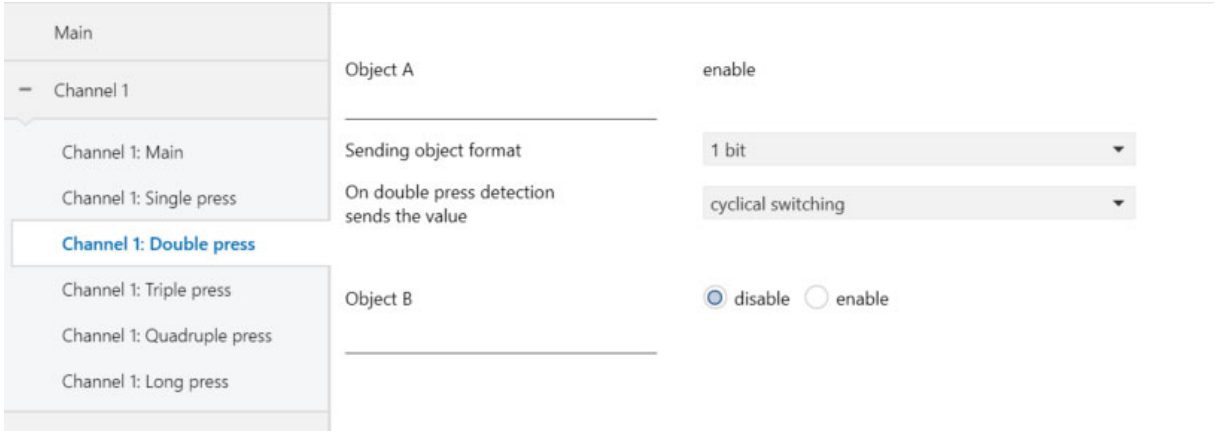


Fig 17: Setting ETS parameters – “Channel x” section - “Channel x - Double press” menu

19.1 parameters

19.1.1 Object z

Upon detection of a double press, it is possible to send up to 8 different objects (which are distinguished by the letters A, B, C, D, E, F, G and H); Object A is always enabled, and the parameter **“object z”** (z indicates the object matched with the threshold, included between **B** and **H)** is used to enable a new object to be sent. The parameter can assume the following values:

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

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

19.1.2 Sending object format

The parameter **Px “Sending object format”** 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 bits
- 1 byte unsigned value
- 1 byte signed value
- 1 byte percentage value
- 1 byte HVAC mode
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value
- 14 bytes
- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the parameter **Px "On double press detection sends the value"** will be different.

19.1.3 On double press detection sends the value

The parameter **Px "On double press detection sends the value"** 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) matched with the channel. The values that can be set are:

- If the sending object format is **1 bit**, the communication object ***Ch.x - Double press 1 bit object z*** will be visible and the values that can be set for the above parameter are:

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

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

- If the sending object format is **2 bits**, the communication object ***Ch.x - Double press 2 bits object z*** will be visible and the values that can be set for the above parameter are:

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

  selecting **cyclical switching**, the command that the device will send (via the object ***Ch.x - Double press 2 bits object z***) when the event matched with the cyclical switching detected will be the opposite of the last value sent (via the object ***Ch.x - Double press 2 bits object z***).

- If the sending object format is **1 byte unsigned value**, the communication object ***Ch.x - Double press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte signed value**, the communication object ***Ch.x - Double press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte percentage value**, the communication object ***Ch.x - Double press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte HVAC mode**, the communication object ***Ch.x - Double press 1 byte object z*** will be visible 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 matched 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 matched 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 sending object format is **2 bytes unsigned value**, the communication object ***Ch.x - Double press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **2 bytes signed value**, the communication object ***Ch.x - Double press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **3 bytes RGB colour**, the parameter **Px "On double press detection sends the value"** 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). Furthermore, the communication object ***Ch.x - Double press 3 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

  by selecting **customize**, the following parameters are made visible: **Px "Value of RED component (0 ... 255)"**, **Px "Value of GREEN component (0 ... 255)"** and **Px "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, in steps of 1

- If the sending object format to send is **4 bytes unsigned value**, the communication object ***Ch.x - Double press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **4 bytes signed value**, the communication object ***Ch.x - Double press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

NOTE: to remedy the problem of coding and the method for entering values with 2/4 bytes floating point format, the user must convert the floating value to a 2/4 byte unsigned or signed value and then enter it in the ETS database selecting the “2/4 bytes signed/unsigned value” format.

20 “Channel x – Triple press” menu

This menu is visible if the value of the parameter **Px “Triple press detection”** 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 - matched with the “triple press” event.
The structure of the menu is as follows:

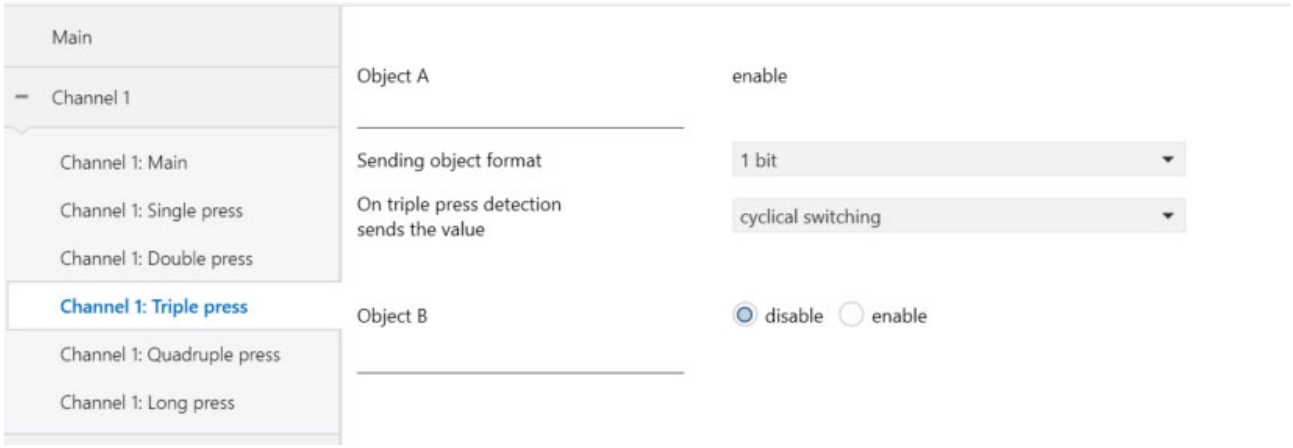


Fig 18: Setting ETS parameters – “Channel x” section - “Channel x - Triple press” menu

20.1 Parameters

20.1.1 Object z

Upon detection of a triple press, it is possible to send up to 8 different objects (which are distinguished by the letters A, B, C, D, E, F, G and H); Object A is always enabled, and the parameter **“object z”** (z indicates the object matched with the threshold, included between **B** and **H)** is used to enable a new object to be sent. The parameter can assume the following values:

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

selecting **enable** displays the parameters **Px “Sending object format”** and **Px “On triple press detection sends the value”** found in the sub-group **Object z** (where z is the index of the object matched with the channel, included between **A** and **H**).

20.1.2 Sending object format

The parameter **Px “Sending object format”** 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 bits
- 1 byte unsigned value
- 1 byte signed value
- 1 byte percentage value
- 1 byte HVAC mode
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value

- 14 bytes
- 3 bytes RGB colour

20.1.3 On triple press detection sends the value

Depending on the value set for this item, the values that can be set for the parameter **Px "On triple press detection sends the value"** will be different.
The parameter **Px "On triple press detection sends the value"** 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) matched with the channel.
The values that can be set are:

- If the sending object format is **1 bit**, the communication object ***Ch.x - Triple press 1 bit object z*** will be visible and the values that can be set for the above parameter are:

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

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

- If the sending object format is **2 bits**, the communication object ***Ch.x - Triple press 2 bits object z*** will be visible and the values that can be set for the above parameter are:

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

  selecting **cyclical switching**, the command that the device will send (via the object ***Ch.x - Triple press 2 bits object z***) when the event matched with the cyclical switching detected will be the opposite of the last value sent (via the object ***Ch.x - Triple press 2 bits object z***).

- If the sending object format is **1 byte unsigned value**, the communication object ***Ch.x - Triple press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte signed value**, the communication object ***Ch.x - Triple press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte percentage value**, the communication object ***Ch.x - Triple press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte HVAC mode**, the communication object ***Ch.x - Triple press 1 byte object z*** will be visible 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 matched 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 matched 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 sending object format is **2 bytes unsigned value**, the communication object ***Ch.x - Triple press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **2 bytes signed value**, the communication object ***Ch.x - Triple press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **3 bytes RGB colour**, the parameter **Px "On triple press detection sends the value"** 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). Furthermore, the communication object ***Ch.x - Triple press 3 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

  by selecting **customize**, the following parameters are made visible: **Px "Value of RED component (0 ... 255)"**, **Px "Value of GREEN component (0 ... 255)"** and **Px "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, in steps of 1

- If the sending object format is **4 bytes unsigned value**, the communication object ***Ch.x - Triple press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **4 bytes signed value**, the communication object ***Ch.x - Triple press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

NOTE: to remedy the problem of coding and the method for inserting values with 2/4 bytes floating point format (DPT 9.0xx and 14.0xx), the user must convert the floating value to a 2/4 bytes unsigned or signed value and then enter it in the ETS database selecting the “2/4 bytes signed/unsigned value” format.

21 “Channel x – Quadruple press” menu

This menu is visible if the value of the parameter **Px “Quadruple press detection”** 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 - matched with the “quadruple press” event.
The structure of the menu is as follows:

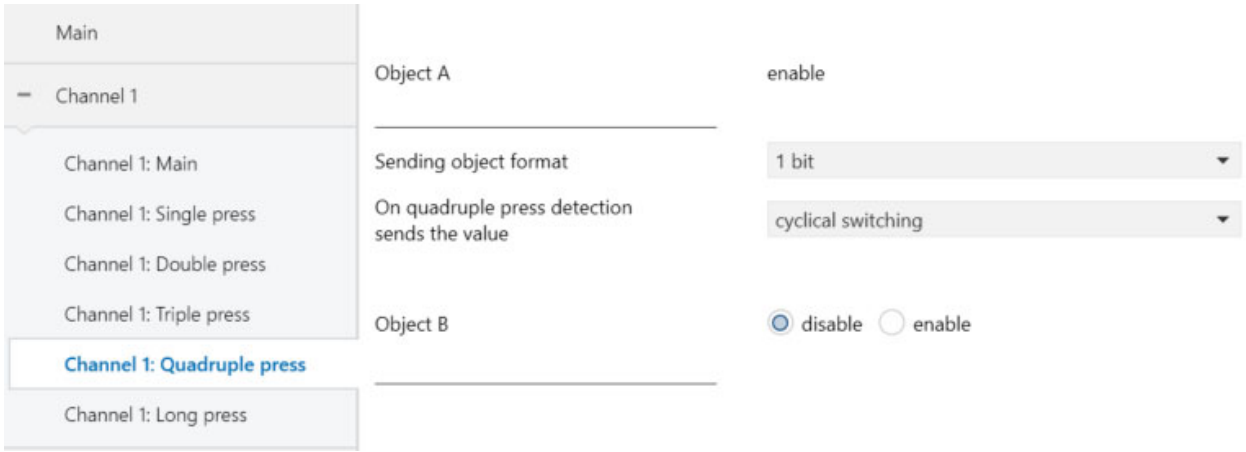


Fig 19: Setting ETS parameters – “Channel x” section - “Channel x - Quadruple press” menu

21.1 Parameters

21.1.1 Object z

Upon detection of a quadruple press, it is possible to send up to 8 different objects (which are distinguished by the letters A, B, C, D, E, F, G and H); Object A is always enabled, and the parameter **“object z”** (z indicates the object matched with the threshold, included between **B** and **H)** is used to enable a new object to be sent. The parameter can assume the following values:

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

selecting **enable** displays the parameters **Px “Sending object format”** and **Px “On quadruple press detection sends the value”** found in the sub-group **Object z** (where z is the index of the object matched with the channel, included between **A** and **H**).

21.1.2 Sending object format

The parameter **Px “Sending object format”** 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 bits
- 1 byte unsigned value
- 1 byte signed value
- 1 byte percentage value
- 1 byte HVAC mode
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value
- 14 bytes

- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the parameter **Px "On quadruple press detection sends the value"** will be different.

21.1.3 On quadruple press detection sends the value

The parameter **Px "On quadruple press detection sends the value"** 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) matched with the channel. The values that can be set are:

- If the sending object format is **1 bit**, the communication object ***Ch.x - Quadruple press 1 bit object z*** will be visible and the values that can be set for the above parameter are:

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

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

- If the sending object format is **2 bits**, the communication object ***Ch.x - Quadruple press 2 bits object z*** will be visible and the values that can be set for the above parameter are:

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

  selecting **cyclical switching**, the command that the device will send (via the object ***Ch.x - Quadruple press 2 bits object z***) when the event matched with the cyclical switching detected will be the opposite of the last value sent (via the object ***Ch.x - Quadruple press 2 bits object z***).

- If the sending object format is **1 byte unsigned value**, the communication object ***Ch.x - Quadruple press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte signed value**, the communication object ***Ch.x - Quadruple press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte percentage value**, the communication object ***Ch.x - Quadruple press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte HVAC mode**, the communication object ***Ch.x - Quadruple press 1 byte object z*** will be visible 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 matched 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 matched 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 sending object format is **2 bytes unsigned value**, the communication object ***Ch.x - Quadruple press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **2 bytes signed value**, the communication object ***Ch.x - Quadruple press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **3 bytes RGB colour**, the parameter **Px "On quadruple press detection sends the value"** 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). Furthermore, the communication object ***Ch.x - Quadruple press 3 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

  by selecting **customize**, the following parameters are made visible: **Px "Value of RED component (0 ... 255)"**, **Px "Value of GREEN component (0 .. 255)"** and **Px "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, in steps of 1

- If the sending object format is **4 bytes unsigned value**, the communication object ***Ch.x - Quadruple press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **4 bytes signed value**, the communication object ***Ch.x - Quadruple press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

NOTE: to remedy the problem of coding and the method for inserting values with 2/4 bytes floating point format (DPT 9.0xx and 14.0xx), the user must convert the floating value to a 2/4 bytes unsigned or signed value and then enter it in the ETS database selecting the “2/4 bytes signed/unsigned value” format.

22 “Channel X – Long press” menu

This menu is visible if the value of the parameter **Px “Long press detection”** 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 - matched with the “long press” event.
The structure of the menu is as follows:

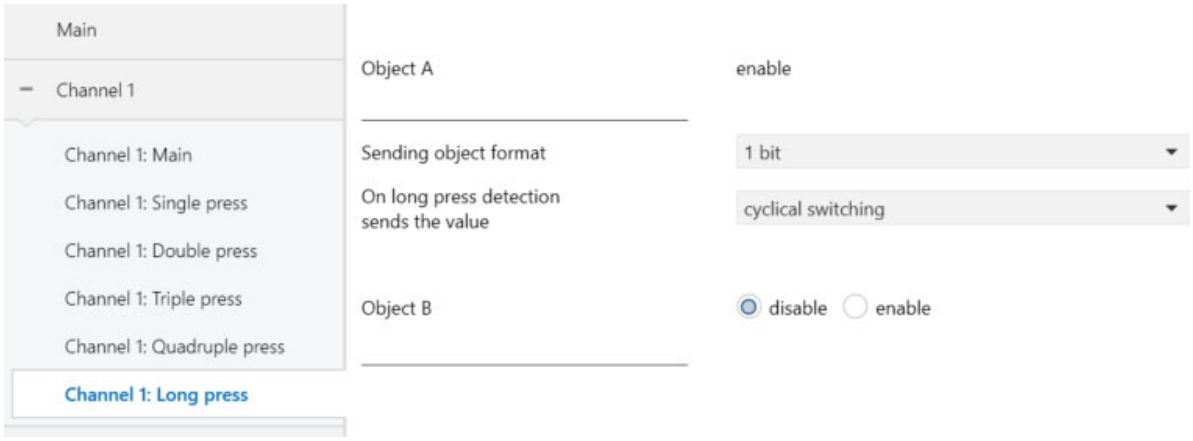


Fig 20: Setting ETS parameters – “Channel x” section - “Channel x - Long press” menu

22.1 Parameters

22.1.1 Object z

Upon detection of the long press, it is possible to send up to 8 different objects (which are distinguished by the letters A, B, C, D, E, F, G and H); Object A is always enabled, and the parameter **“object z”** (z indicates the object matched with the threshold, included between **B** and **H)** is used to enable a new object to be sent. The parameter can assume the following values:

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

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

22.1.2 Sending object format

The parameter **Px “Sending object format”** 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 bits
- 1 byte unsigned value
- 1 byte signed value
- 1 byte percentage value
- 1 byte HVAC mode
- 2 bytes unsigned value
- 2 bytes signed value

- 4 bytes unsigned value
- 4 bytes signed value
- 14 bytes
- 3 bytes RGB colour

Depending on the value set for this item, the values that can be set for the parameter **Px "On long press detection sends the value"** will be different.

22.1.3 On long press detection sends the value

The parameter **Px "On long press detection sends the value"** 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) matched with the channel.

The values that can be set are:

- If the sending object format is **1 bit**, the communication object ***Ch.x - Long press 1 bit object z*** will be visible and the values that can be set for the above parameter are:

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

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

- If the sending object format is **2 bits**, the communication object ***Ch.x - Long press 2 bits object z*** will be visible and the values that can be set for the above parameter are:

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

  selecting **cyclical switching**, the command that the device will send (via the object ***Ch.x - Long press 2 bits object z***) when the event matched with the cyclical switching detected will be the opposite of the last value sent (via the object ***Ch.x - Long press 2 bits object z***).

- If the sending object format is **1 byte unsigned value**, the communication object ***Ch.x - Long press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte signed value**, the communication object ***Ch.x - Long press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte percentage value**, the communication object ***Ch.x - Long press 1 byte object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **1 byte HVAC mode**, the communication object ***Ch.x - Long press 1 byte object z*** will be visible 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 matched 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 matched 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 sending object format is **2 bytes unsigned value**, the communication object ***Ch.x - Long press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **2 bytes signed value**, the communication object ***Ch.x - Long press 2 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **3 bytes RGB colour**, the parameter **Px "On long press detection sends the value"** 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). Furthermore, the communication object ***Ch.x - Long press 3 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

  by selecting **customize**, the following parameters are made visible: **Px "Value of RED component (0 ... 255)"**, **Px "Value of GREEN component (0 ... 255)"** and **Px "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, in steps of 1

- If the sending object format is **4 bytes unsigned value**, the communication object ***Ch.x - Long press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

- If the sending object format is **4 bytes signed value**, the communication object ***Ch.x - Long press 4 bytes object z*** will be visible and the values that can be set for the above parameter are:

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

NOTE: to remedy the problem of coding and the method for inserting values with 2/4 bytes floating point format (DPT 9.0xx and 14.0xx), the user must convert the floating value to a 2/4 bytes unsigned or signed value and then enter it in the ETS database selecting the “2/4 bytes signed/unsigned value” format.

23 “Temperature sensor” function

Used to configure the channel to send the temperature measured by the Gewiss wired NTC sensors.
The structure of the menu is as follows:

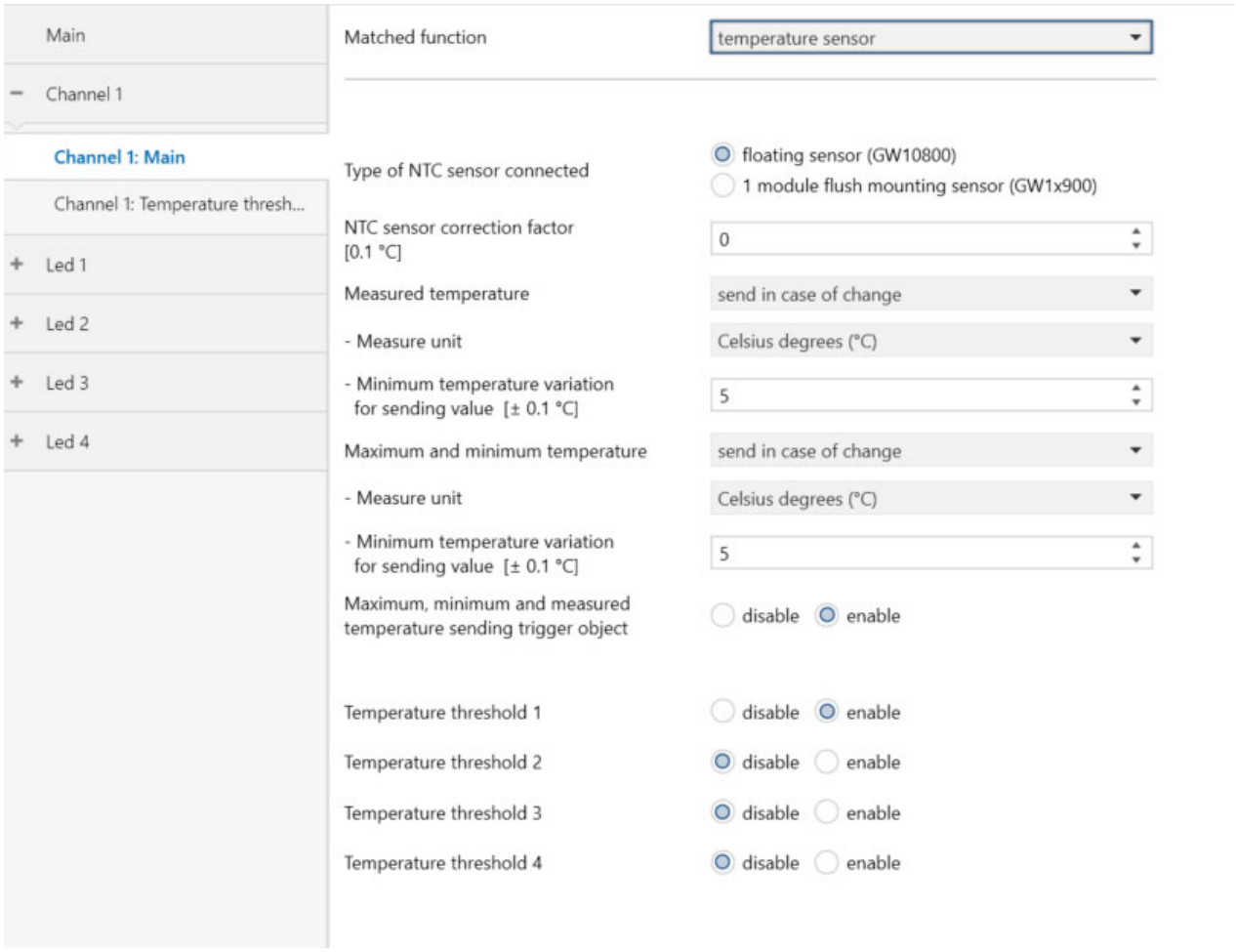


Fig 21: Setting ETS parameters – “Channel x” section - “temperature sensor” function

23.1 Parameters

23.1.1 Type of NTC sensor connected

Various Gewiss temperature sensors can be connected to the input contacts; Given the different characteristics of each transducer, the parameter **Px “Type of NTC sensor connected”** is used to define which of the possible sensors will be connected to the device contacts, in order to interface correctly with the sensor; the values that can be set are:

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

23.1.2 NTC sensor correction factor [0.1°C]

The parameter **Px "NTC sensor correction factor [0.1 °C]"** is used to set the correction factor to be applied to the measured temperature value of the NTC sensor connected to the channel, 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 **(default value 0)**

23.1.3 Measured temperature

The parameter **Px "Measured temperature"** 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 in case of change and periodically

Selecting the value **send in case of change** or **send in case of change and periodically**, the parameter **Px "Minimum temperature variation for sending value [± 0.1°C]"** will also be visible, whereas by selecting **send periodically** or **send in case of change and periodically** the parameter **Px "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.

23.1.4 Measure unit

The parameter **"Measure unit"** is used to set the measurement unit for coding and sending the information via the ***Ch.x – Measured temperature*** 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 changes the coding of the communication object ***Ch.x - Measured temperature***: *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)**.

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

The parameter **Px "Minimum temperature variation for sending value [± 0.1°C]"** (visible if the temperature is sent due to a variation) is used to define the minimum temperature variation - in relation to the last sent value - for generating the spontaneous sending of the new measured value; the values that can be set are:

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

23.1.6 Temperature sending period [minutes]

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

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

23.1.7 Maximum and minimum temperature

The parameter **Px "Maximum and minimum temperature"** is used to define the conditions for sending the value of the maximum and minimum temperature measured by the device; the values that can be set are:

- send on demand only 1
- **send in case of change** **2 (default value)**
- send periodically 3
- send in case of change and periodically 4

selecting the value **send in case of change** or **send in case of change and periodically**, the parameter **Px "Minimum temperature variation for sending value [± 0.1°C]"** will also be visible, whereas by selecting **send periodically** or **send in case of change and periodically** the parameter **Px "Temperature sending period [minutes]"** will be visible.

Selecting the value **send on demand only**, no new parameter will be enabled, as the maximum and minimum temperature value is not sent spontaneously by the device; in the case of a status read request, it sends the requester a telegram in response to the received command, which includes information about the measured maximum and minimum temperature value.

The maximum and minimum temperature value can be reset using the ***Ch.x - Maximum and minimum temperature reset*** communication object (Data Point Type: 1.017 DPT_Trigger); when this object receives a BUS telegram with the logic value "0" or "1", the device automatically resets the maximum and minimum temperature to the current measured temperature, cancelling the previously saved values.

23.1.8 Measure unit

The parameter **Px "Measure unit"** is used to set the measurement unit for coding and sending the information via the communication objects ***Ch.x - Maximum measured temperature*** and ***Ch.x - Minimum measured temperature***; the values that can be set are:

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

the value set for this parameter changes the coding of the communication objects ***Ch.x - Maximum measured temperature*** and ***Ch.x - Minimum measured temperature***: *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)**.

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

The parameter **Px "Minimum temperature variation for sending value [± 0.1 °C]"**, which is visible if the maximum and minimum temperature are sent due to a variation, is used to define the minimum temperature variation in comparison to the last sent value, which generates the spontaneous sending of the new measured value; the values that can be set are:

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

23.1.10 Maximum and minimum temperature sending period [minutes]

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

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

23.1.11 Maximum, minimum and measured temperature sending trigger object

The parameter **Px "Maximum, minimum and measured temperature sending trigger object"** is used to enable the input object ***Ch.x - Feedback sending trigger*** (Data Point Type: 1.017 DPT_Trigger); when this object receives a BUS telegram with the logical value "0" or "1", the device automatically sends all the temperature feedbacks (measured, maximum and minimum) that are sent "on change" or "on change and periodically". The values that can be set are:

- **disable 0 (default value)**
- enable 1

The malfunction or failed connection of the NTC sensor is signalled by the value " 7FFFh" as indicated in the KNX specification:

3.10 Datapoint Types "2-Octet Float Value"

Format:	2 octets: F_{16}
octet nr	2 MSB 1 LSB
field names	FloatValue
encoding	M E E E E M M M M M M M M M M M
Encoding:	FloatValue = $(0,01*M)*2^{(E)}$ E = [0 … 15] M = [-2 048 … 2 047], two's complement notation For all Datapoint Types 9.xxx, the encoded value 7FFFh shall always be used to denote invalid data.
Range:	[-671 088,64 … 670 433,28]
PDT:	PDT_KNX_FLOAT

23.1.12 Temperature threshold X

The parameters **Px "Temperature threshold 1"**, **Px "Temperature threshold 2"**, **Px "Temperature threshold 3"** and **Px "Temperature threshold 4"** are used to activate the relative temperature thresholds and display the different configuration menus. The values that can be set are:

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

selecting **enable** displays the configuration menus **Channel x - Temperature threshold 1**, **Channel x - Temperature threshold 2**, **Channel x - Temperature threshold 3** and **Channel x - Temperature threshold 4** depending on the reference parameter.

24 “Channel X - Temperature threshold Y” menu

For each channel configured as a temperature sensor, it is possible to activate up to 4 temperature thresholds to associate with the sending of different BUS commands when the fixed threshold is exceeded. All 4 thresholds are identical and therefore, for the sake of simplicity, the operation and dedicated parameters are summarised in this paragraph, indicating the reference threshold with a generic “Y” (1 ... 4). The configuration menus are enabled by the relative parameter in the **Temperature sensor x** menu.
The structure of the menu is as follows:

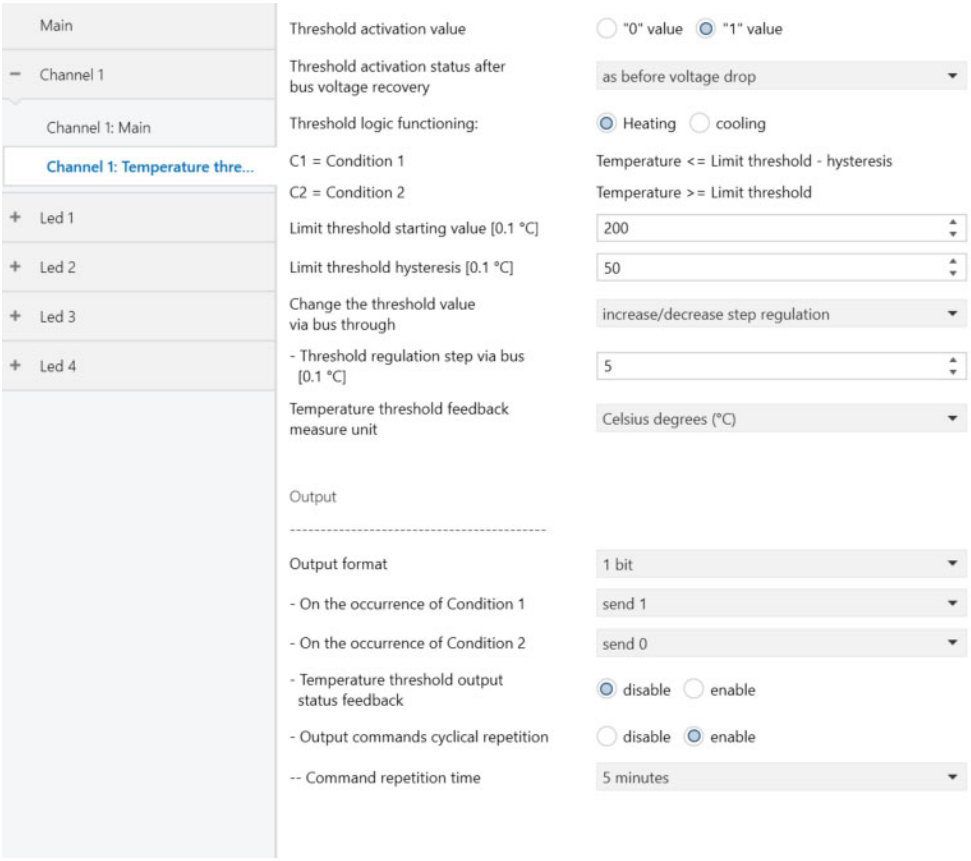


Fig 22: Setting ETS parameters – “Channel x” section - “Channel x - Temperature threshold y” menu

24.1 Parameters

Via the communication objects ***Ch.x - Temperature threshold y enabling*** (Data Point Type:1.002 DPT_Boolean) and ***Temperature threshold y enabling status*** (Data Point Type:1.003 DPT_Enable), it is possible to receive threshold activation commands and send threshold activation feedback respectively; the telegrams are sent via the object ***Ch.x - Temperature threshold y enabling status*** following a BUS request, spontaneously after each change to the threshold activation status and upon BUS voltage recovery.

24.1.1 Threshold activation value

The parameter **Px “Threshold activation value”** determines which logic value received via the communication object ***Ch.x - Temperature threshold y enabling*** activates the temperature threshold y; the arrival of the opposite value will deactivate the threshold. The values that can be set are:

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

24.1.2 Threshold activation status after bus voltage recovery

The parameter **Px "Threshold activation status after bus voltage recovery"** is used to set the status of the temperature threshold y after the BUS power supply voltage is restored; the values that can be set are:

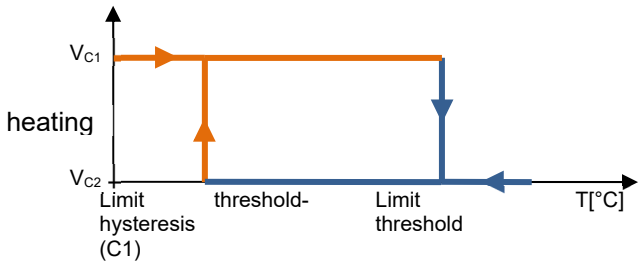
- disabled
- enabled
- **as before voltage drop** **(default value)**

24.1.3 Threshold operating logic

The **Px "Threshold operating logic"** parameter is used to define the type of hysteresis to be adopted and, as a result, the hysteresis limit values. The parameter can assume the following values:

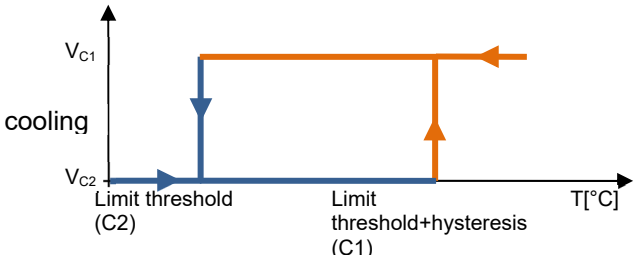
- **heating** **(default value)**
- cooling

By selecting heating, the two conditions will be defined as follows:
Condition 1 = Measured temperature ≤ Limit threshold – Hysteresis
Condition 2 = Measured temperature ≥ Limit threshold



When the measured temperature is lower than value C1 (limit threshold-hysteresis), the device will send the command matched with Condition 1; when the measured temperature reaches value C2 (limit threshold), the device will send the command matched with Condition 2.

By selecting cooling, the two conditions will be defined as follows:
Condition 1 = Measured temperature ≥ Limit threshold + Hysteresis
Condition 2 = Measured temperature ≤ Limit threshold



When the measured temperature is higher than value C1 (limit threshold+hysteresis), the device will send the command matched with Condition 1; when the measured temperature reaches value C2 (limit threshold), the device will send the command matched with Condition 2.

24.1.4 Limit threshold starting value [0.1°C]

Via the parameter **Px "Limit threshold starting value 0.1 °C]"** it is possible to set the initial value of the limit threshold matched with the temperature threshold y (which can be changed via BUS via the specific communication object). The parameter can assume the following values:

- from 0 to 400 in steps of 1 **(default value 200)**

24.1.5 Limit threshold hysteresis [0.1°C]

The parameter **Px "Limit threshold hysteresis [0.1 °C]"** can be used to set the hysteresis value that, when added to or subtracted from the limit threshold, contributes towards defining the second limit value for sending the commands. This parameter can assume the following values:

- from 1 to 100 in steps of 1 **(default value 50)**

24.1.6 Change the threshold value via bus through

The parameter **Px "Change the threshold value via bus through"** is used to define the format of the communication object necessary for setting the limit threshold via BUS telegram; the values that can be set are:

- **absolute value setting (°C)** **(default value)**
- absolute value setting (°K)
- absolute value setting (°F)
- increase/decrease step regulation

selecting the value **absolute value setting** displays the communication object ***Ch.x - Temperature threshold y value input*** (Data Point Type: *9.001 DPT_Value_Temp* if °C, *9.002 DPT_Value_Tempd* if °K and *9.027 DPT_Value_Temp_F* if °F), via which it is possible to set the value of the limit threshold via BUS.

Selecting **increase/decrease step setting** displays the parameter **Px "Threshold regulation step via BUS [0.1 °C]"** and the object ***Temperature threshold y regulation*** (Data Point Type: 1.007 DPT_Step). If the value "1" is received on this object, the limit threshold value will be increased by the value defined by the parameter **Px "Threshold regulation step via BUS [0.1 °C]"**; if the value "0" is received on this object, the limit threshold value will be decreased by the value defined by the parameter **Px "Threshold regulation step via BUS [0.1 °C]"**.

24.1.7 Threshold regulation step via bus [0.1°C]

The parameter **Px "Threshold regulation step via bus [0.1 °C]"** is used to define the increase/decrease step of the limit threshold value after receiving a command on the relative regulation object. The values that can be set are

- from 1 to 20 in steps of 1, **(default value 5)**

The current value of the temperature threshold y is transmitted on the BUS via the object ***Ch.x - Temperature threshold y feedback***; the parameter **Px "Temperature threshold feedback measure unit"** is used to set the measure unit for coding and sending the information via the communication object ***Ch.x - Temperature threshold y feedback***; the values that can be set are:

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

the value set for this parameter changes the coding of the communication object ***Ch.x - Temperature threshold y feedback***: *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)**. The feedback sending conditions are, following a BUS request, spontaneously at each threshold change and at BUS voltage recovery.

It is possible to send an object for each threshold depending on the occurrence of conditions 1 and 2; The parameter **Px "Output format"** is used to set the format and coding of the BUS telegram that will be sent by the device. The values that can be set are:

- 1 bit
- 2 bits
- 1 byte unsigned values
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC
- 2 bytes unsigned values
- 2 bytes signed
- 2 bytes setpoint in °C
- 2 bytes setpoint in °K
- 2 bytes setpoint in °F

The value set for this item will cause the values set for the parameters **"On the occurrence of condition 1"** and **Px "On the occurrence of condition 2"** to change as a result.

24.1.8 On the occurrence of condition X

The parameter **Px "On the occurrence of condition 1"** is used to set the command or value to be sent when condition 1 arises.
The parameter **Px "On the occurrence of condition 2"** is used to set the command or value to be sent when condition 2 arises.

- If the output format is **1 bit**, this makes the communication object ***Ch.x - Temperature threshold y output*** visible (Data Point Type: 1.001 DPT_Switch) and the parameter **Px "Temperature threshold output status feedback"**. The values that can be set for the two parameters listed above are:

  - **no action** **(default value when cond 2 occurs)**
  - send 0
  - **send 1** **(default value when cond 1 occurs)**

24.1.9 Temperature threshold output status feedback

The parameter **Px "Temperature threshold output status feedback"** is used to enable and display the communication object ***Ch.x - Temperature threshold y output status feedback*** (Data Point Type: 1.001 DPT_Switch). enabling this object, the command on the object ***Ch.x - Temperature threshold y output*** is repeated until the status received on the object ***Ch.x - Temperature threshold y output status feedback*** coincides with the command. Possible repetition of the command will take place once a minute. The parameter can assume the following values:

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

selecting the value **enable** displays the communication object ***Ch.x - Temperature threshold y output status feedback***; in this case, each time the BUS voltage is reset, the device sends a status read command on this object to update the device about the status of the connected devices only if condition C1 or C2 has occurred (otherwise, no request is sent).

Selecting **disable** displays the parameter **Px "Cyclical repetition of the output commands"**.

- If the output format is **2 bits**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 2.001 DPT_Switch_Control). The values that can be set for the two parameters above are:

  - **no action** **(default value cond 2)**
  - **send forcing activation on (down)** **(default value cond 1)**
  - send forcing activation off (up)
  - send deactivate forcing

- If the output format is **1 byte unsigned**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 5.010 DPT_Value_1_Ucount). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - **send value** **(default value cond 1)**

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

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

- If the output format is **1 byte signed**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 6.010 DPT_Value_1_Count). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - **send value** **(default value cond 1)**

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

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

- If the output format is **1 byte percentage value**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 5.001 DPT_Scaling). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - **send value** **(default value cond 1)**

  by setting **send value**, it is possible to define the value to be sent via the newly displayed parameter **Px "Value (0% .. 100%)"** that can have the following values:

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

- If the output format is **1 byte HVAC mode**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 20.102 DPT_HVACMode). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - send auto
  - **send comfort** **(default value cond 1)**
  - send precomfort
  - send economy
  - send off (building protection)
  - send reference HVAC mode ± offset

  selecting **send reference HVAC mode ± offset** displays the parameter **Px "Offset (-3 .. +3)"** and the communication object ***Ch.x - Temperature threshold output reference y*** (Data Point Type: 20.102 DPT_HVACMode). in this case, the output value will be the HVAC mode received via the object ***Ch.x - Temperature threshold y output reference*** to which the offset will be added (the order of the modes is:

auto, comfort, precomfort, economy, off). If nothing was ever received on the reference object, the initial value is equal to “auto”.

EXAMPLE: to switch from “comfort” mode to “economy” mode, the offset must be “+2”; The set is not circular, so once the limit values are reached (“auto” or “off”) the calculation is ended even if the offset that is set is greater than what is actually applied to reach the limit value.

The parameter **Px “Offset (-3 ... +3)”** is used to set the offset to be applied to the current or reference HVAC mode to obtain the value to be sent via the object ***Ch.x - Temperature threshold y output***; the values that can be set are:

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

- If the output format is **2 bytes unsigned**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 7.001 DPT_Value_2_Ucount). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - **send value** **(default value cond 1)**

  by setting **send value**, it is possible to define the value to be sent via the newly displayed parameter **Px “Value (0 .. 65535)”**, which can assume the following values:

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

- If the output format is **2 bytes signed**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 8.001 DPT_Value_2_Count). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - **send value** **(default value cond 1)**

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

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

- If the output format is **2 bytes setpoint in °C**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 9.001 DPT_Value_Temp). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - **send reference setpoint ± offset** **(default value cond 1)**

  selecting **send reference setpoint ± offset** displays the parameter **Px “Offset [0.1 °C] (-300 ... +300)”** and the communication object ***Ch.x - Temperature threshold y output reference*** (Data Point Type: 9.001 DPT_Value_Temp). in this case, the output value will be the setpoint received via the object ***Ch.x - Temperature threshold y output reference*** to which the offset will be added. If nothing was ever received on the reference object, the initial value is equal to “20°C”.

  The parameter **Px “Offset [0.1°C] (-300 ... +300)”** is used to set the offset to be applied to the current or reference setpoint to obtain the value to be sent via the object ***Ch.x - Temperature threshold y output***; the values that can be set are:

  - from -300 to +300 in steps of 1 **(default value +10)**

- If the output format is **2 bytes setpoint in °K**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 9.002 DPT_Value_Tempd). The values that can be set for the two parameters listed above are:

- **no action** **(default value cond 2)**
- **send reference setpoint + offset** **(default value cond 1)**

selecting **send reference setpoint ± offset** displays the parameter **Px "Offset [0.1 °C] (-300 ... +300)"** and the communication object ***Ch.x - Temperature threshold y output reference*** (Data Point Type: 9.002 DPT_Value_Tempd). in this case, the output value will be the setpoint received via the object ***Ch.x - Temperature threshold y output reference*** to which the offset will be added. If nothing was ever received on the reference object, the initial value is equal to "293°K".

The parameter **Px "Offset [0.1°C] (-300 ... +300)"** is used to set the offset to be applied to the current or reference setpoint to obtain the value to be sent via the object ***Ch.x - Temperature threshold y output***; the values that can be set are:

- from -300 to +300 in steps of 1 **(default value +10)**

- If the output format is **2 bytes setpoint in °F**, this displays the communication object ***Ch.x - Temperature threshold y output*** (Data Point Type: 9.027 DPT_Value_Temp_F). The values that can be set for the two parameters listed above are:

  - **no action** **(default value cond 2)**
  - **send reference setpoint + offset** **(default value cond 1)**

  selecting **send reference setpoint ± offset** displays the parameter **Px "Offset [0.1 °C] (-300 .. +300)"** and the communication object ***Ch.x - Temperature threshold y output reference*** (Data Point Type: 9.027 DPT_Value_Temp_F). in this case, the output value will be the setpoint received via the object ***Ch.x - Temperature threshold y output reference*** to which the offset will be added. If nothing was ever received on the reference object, the initial value is equal to "68°F".

  The parameter **Px "Offset [0.1°C] (-300 ... +300)"** is used to set the offset to be applied to the current or reference setpoint to obtain the value to be sent via the object ***Ch.x - Temperature threshold y output***; the values that can be set are:

  - from -300 to +300 in steps of 1 **(default value +10)**

The object ***Ch.x - Temperature threshold y output*** is sent upon demand, spontaneously on variation of the current condition (C1 or C2), periodically (if cyclical repetition is enabled) and when BUS voltage is restored only if condition C1 or C2 is verified (otherwise, no value is sent). When the threshold is disabled, the sending of telegrams matched with conditions C1 and C2 is inhibited, but any possible change or feedback of the threshold value is still executed.

The **Px "Output commands cyclical repetition"** parameter is used to enable the periodic sending of the output value; The possible values are:

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

Selecting **enable** displays the parameter **Px "Command repetition time"** which is used to set the repetition period of the commands. The values that can be set are:

- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** **(default)**

25 *“LED X”* menu

This is used to define and personalise the operation of the signalling LED matched 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:

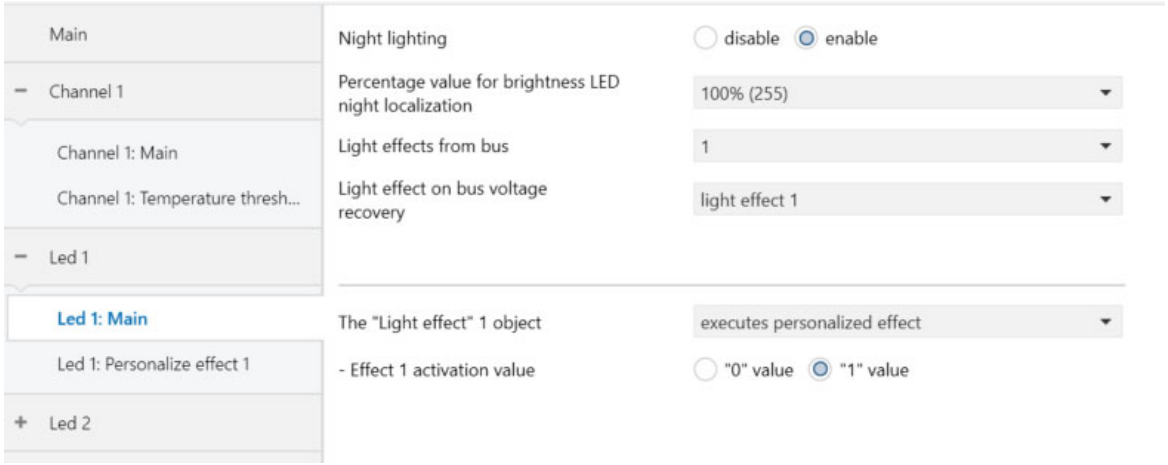


Fig 23: Setting ETS parameters – “LED x” section

25.1 Parameters

25.1.1 Night lighting

This parameter can be used to enable the night lighting matched with channel x; the values that can be set are:

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

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

25.1.2 Percentage value for brightness LED night localization

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

25.1.3 Light effects from bus

The parameter **“Light effects from bus”** 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”**.

25.1.4 Light object x

The 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 brightness signal from the BUS. The values that can be set for this parameter are:

- **status feedback** **(default value)**, only visible if night lighting is disabled
- switches off the night lighting **(default value)**, only visible if night lighting 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 personalized effect

If **personalized effect** is selected, the new **Personalize effect y** configuration menu will be made visible, with 1≤Y≤5 (see paragraph "Personalize effect y menu")

25.1.5 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 matched 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.

25.1.6 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 active, when bus voltage is recovered the night lighting will be activated.

25.2 “Personalize 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 personalized effect**.
In this menu, all the parameters used for creating the light effects are visible and configurable; in this way, the user can create a personalized light effect. These light effects are activated/deactivated by the ***LED x - Effect y*** objects; The basic structure of the menu is as follows:

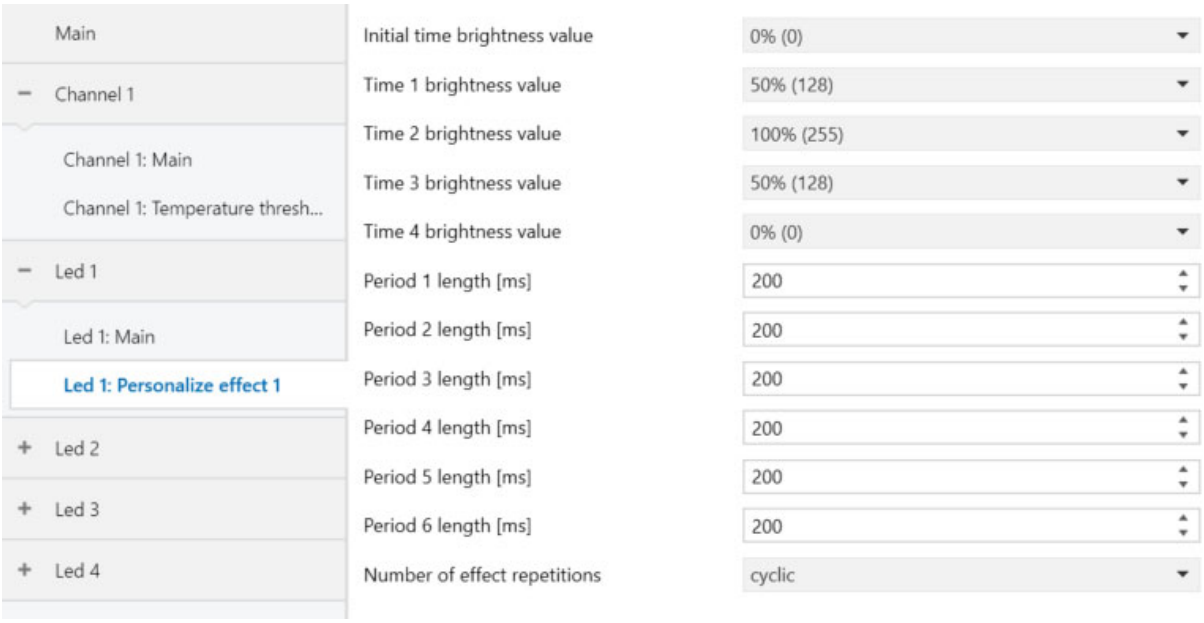
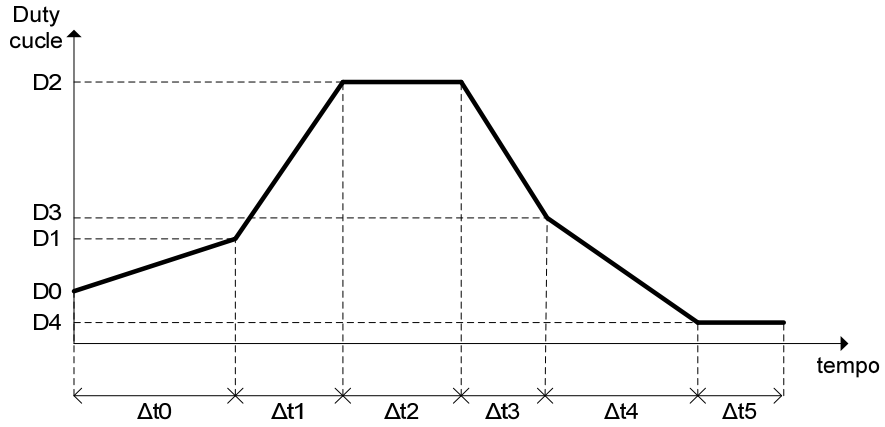


Fig 24: Setting ETS parameters – “Personalize Effect y”

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);

25.2.1 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]”** ($\Delta_t 0$), **“Period 2 length [ms]”** ($\Delta_t 1$), **“Period 3 length [ms]”** ($\Delta_t 2$), **“Period 4 length [ms]”** ($\Delta_t 3$), **“Period 5 length [ms]”** ($\Delta_t 4$) and **“Period 6 length [ms]”** ($\Delta_t 5$), which can assume the following values:

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

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)**

26 Factory reset / Feedback of ETS download in progress / Application deleted

The device can be used to perform a factory reset via the following procedure:
1. Disconnect the power (KNX terminal)
2. Press the KNX address programming key
3. Reconnect the power supply (KNX terminal) with the programming key still pressed
4. After approx. 5 seconds, 5 fast flashes of the programming LED signal that the factory reset was performed

After a factory reset, all the factory parameters are reset along with the physical factory address, and the FDSK (Factory Default Setup Key - do master reset) is reactivated.

Attention! If an application is downloaded from ETS with KNX Secure enabled, it will not be possible to download another from a different ETS project before performing the factory reset from the local menu. The procedure for resetting the FDSK is essential and therefore it is not enough to delete the application on the device via ETS.

During the download of the ETS application, the red physical address programming LED blinks cyclically approx. every 1.5 seconds. The LED is deactivated at the end of the download.

Following the “delete application” command performed via ETS, the device performs a factory reset (see above), maintaining the physical address (unless the command “delete application and physical address” is performed).

27 Communication objects

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

27.1 Communication objects with output functions

#				Object name	Function object	Description	Datapoint type
Ch 1	Ch 2	Ch 3	Ch 4				
1	50	99	148	Ch.x - Measured temperature	Value °C	Sends the temperature values in degrees Celsius	9.001 DPT_Temp
1	50	99	148	Ch.x - Measured temperature	Value °K	Sends the temperature values expressed in degrees Kelvin	9.002 DPT_Tempd
1	50	99	148	Ch.x - Measured temperature	Value °F	Sends the temperature values in degrees Fahrenheit	9.027 DPT_Temp_F
2	51	100	149	Ch.x - Maximum measured temperature	Value °C	Sends the maximum measured temperature values expressed in degrees Celsius	9.001 DPT_Temp
2	51	100	149	Ch.x - Maximum measured temperature	Value °K	Sends the maximum measured temperature values expressed in degrees Kelvin	9.002 DPT_Tempd
2	51	100	149	Ch.x - Maximum measured temperature	Value °F	Sends the maximum measured temperature values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
2	51	100	149	Ch.x - Switching	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
2	51	100	149	Ch.x - Command value 1 byte	Value 0 - 255	Sends values 0-255 matched to the dimmer on/off	5.010 DPT_Value_1_Ucount
2	51	100	149	Ch.x - Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
2	51	100	149	Ch.x - Scene	Execute/Store	Sends scene memorising/execution commands	18.001 DPT_SceneControl
2	51	100	149	Ch.x - A Sequence	On/Off	Sends On/Off commands matched with object A of the sequence	1.001 DPT_Switch
2	51	100	149	Ch.x - 1 bit value object A	Value 1/0	Sends values 1/0 matched with object A	1.002 DPT_Bool
2	51	100	149	Ch.x - Primary counter	Value 1 byte unsigned	Sends the unsigned value (0..255) of the primary counter	5.010 DPT_Value_1_Ucount
2	51	100	149	Ch.x - Primary counter	Value 1 byte signed	Sends the signed value (-128..127) of the primary counter	6.010 DPT_Value_1_Count
2	51	100	149	Ch.x - Primary counter	Value 2 bytes unsigned	Sends the unsigned value (0..65535) of the primary counter	7.001 DPT_Value_2_Ucount
2	51	100	149	Ch.x - Primary counter	Value 2 bytes signed	Sends the signed value (-32768..32767) of the primary counter	8.001 DPT_Value_2_Count
2	51	100	149	Ch.x - Primary counter	Value 4 bytes unsigned	Sends the unsigned value (0.. 4294967295) of the primary counter	12.001 DPT_Value_4_Ucount
2	51	100	149	Ch.x - Primary counter	Value 4 bytes signed	Sends the signed value (-2147483648.. 2147483647) of the primary counter	13.001 DPT_Value_4_Count
2	51	100	149	Ch.x - 2 bits value object A	On/Off forced positioning	Sends values 1/0 matched with object A	1.002 DPT_Switch_Control
2	51	100	149	Ch.x - 1 byte value object A	Unsigned value	Sends unsigned values (0..255) matched with object A	5.010 DPT_Value_1_Ucount
2	51	100	149	Ch.x - 1 byte value object A	Signed value	Sends signed values (-128..127) matched with object A	6.010 DPT_Value_1_Count
2	51	100	149	Ch.x - 1 byte value object A	Value %	Sends the percentage values (0%..100%) matched with object A	5.001 DPT_Scaling
2	51	100	149	Ch.x - 1 byte value object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off)	20.102 DPT_HVACMode
2	51	100	149	Ch.x - 2 bytes value object A	Unsigned value	Sends unsigned values (0..65535) matched with object A	7.001 DPT_Value_2_Ucount

2	51	100	149	Ch.x - 2 bytes value object A	Signed value	Sends signed values (-32768..32767) matched with object A	8.001 DPT_Value_2_Count
2	51	100	149	Ch.x - 3 bytes value object A	RGB colour	Sends the values of the three RGB colour components matched with object A	232.600 DPT_Colour_RGB
2	51	100	149	Ch.x - 4 bytes value object A	Unsigned value	Sends unsigned values (0.. 4294967295) matched with object A	12.001 DPT_Value_4_Ucount
2	51	100	149	Ch.x - 4 bytes value object A	Signed value	Sends signed values (-2147483648.. 2147483647) matched with object A	13.001 DPT_Value_4_Count
2	51	100	149	Ch.x - 14 bytes value object A	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
2	51	100	149	Ch.x - Single press 1 bit object A	Value 1/0	Sends values 1/0 matched with single press on object A	1.002 DPT_Bool
2	51	100	149	Ch.x - Single press 2 bits object A	On/Off forced positioning	Sends values 1/0 matched with single press on object A	1.002 DPT_Switch_Control
2	51	100	149	Ch.x - Single press 1 byte object A	Unsigned value	Sends unsigned values (0..255) matched with single press on object A	5.010 DPT_Value_1_Ucount
2	51	100	149	Ch.x - Single press 1 byte object A	Signed value	Sends signed values (-128..127) matched with single press on object A	6.010 DPT_Value_1_Count
2	51	100	149	Ch.x - Single press 1 byte object A	Value %	Sends the percentage values (0%..100%) matched with single press on object A	5.001 DPT_Scaling
2	51	100	149	Ch.x - Single press 1 byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) matched with the single press on object A	20.102 DPT_HVACMode
2	51	100	149	Ch.x - Single press 2 bytes object A	Unsigned value	Sends unsigned values (0..65535) matched with single press on object A	7.001 DPT_Value_2_Ucount
2	51	100	149	Ch.x - Single press 2 bytes object A	Signed value	Sends signed values (-32768..32767) matched with single press on object A	8.001 DPT_Value_2_Count
2	51	100	149	Ch.x - Single press 3 bytes object A	RGB colour	Sends the values of the three RGB colour components matched with the single press on object A	232.600 DPT_Colour_RGB
2	51	100	149	Ch.x - Single press 4 bytes object A	Unsigned value	Sends unsigned values (0.. 4294967295) matched with single press on object A	12.001 DPT_Value_4_Ucount
2	51	100	149	Ch.x - Single press 4 bytes object A	Signed value	Sends signed values (-2147483648.. 2147483647) matched with single press on object A	13.001 DPT_Value_4_Count
3	52	101	150	Ch.x - Minimum measured temperature	Value °C	Sends the minimum measured temperature values expressed in degrees Celsius	9.001 DPT_Temp
3	52	101	150	Ch.x - Minimum measured temperature	Value °K	Sends the minimum measured temperature values expressed in degrees Kelvin	9.002 DPT_Tempd
3	52	101	150	Ch.x - Minimum measured temperature	Value °F	Sends the minimum measured temperature values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
3	52	101	150	Ch.x - Brightness control	Increase/decrease	Sends brightness control commands	3.007 DPT_Control_Dimming
3	52	101	150	Ch.x - Louvres stop/adjustment	Stop/Step	Send louvres stop/adjustment commands	1.007 DPT_Step
3	52	101	150	Ch.x - Sequence B	On/Off	Sends On/Off commands matched with object B of the sequence	1.001 DPT_Switch
3	52	101	150	Ch.x - Primary counter overflow bit	Overflow status	Sends the primary counter overflow feedback	1.002 DPT_Bool
3	52	101	150	Ch.x - 1 bit value object B	Value 1/0	Sends values 1/0 matched with object B	1.002 DPT_Bool

3	52	101	150	Ch.x - Single press 1 bit object B	Value 1/0	Sends values 1/0 matched with single press on object B	1.002 DPT_Bool
4	53	102	151	Ch.x - Sequence C	On/Off	Sends On/Off commands matched with object C of the sequence	1.001 DPT_Switch
4	53	102	151	Ch.x - Primary counter overflow byte	Overflow status	Sends the value matched with the primary counter overflow feedback	5.010 DPT_Value_1_Ucount
4	53	102	151	Ch.x - 1 bit value object C	Value 1/0	Sends values 1/0 matched with object C	1.002 DPT_Bool
4	53	102	151	Ch.x - Single press 1 bit object C	Value 1/0	Sends values 1/0 matched with single press on object C	1.002 DPT_Bool
5	54	103	152	Ch.x - Sequence D	On/Off	Sends On/Off commands matched with object D of the sequence	1.001 DPT_Switch
5	54	103	152	Ch.x - Differential counter	Value 1 byte unsigned	Sends the unsigned value (0..255) of the differential counter	5.010 DPT_Value_1_Ucount
5	54	103	152	Ch.x - Differential counter	Value 1 byte signed	Sends the signed value (-128..127) of the differential counter	6.010 DPT_Value_1_Count
5	54	103	152	Ch.x - Differential counter	Value 2 bytes unsigned	Sends the unsigned value (0..65535) of the differential counter	7.001 DPT_Value_2_Ucount
5	54	103	152	Ch.x - Differential counter	Value 2 bytes signed	Sends the signed value (-32768..32767) of the differential counter	8.001 DPT_Value_2_Count
5	54	103	152	Ch.x - Differential counter	Value 4 bytes unsigned	Sends the unsigned value (0.. 4294967295) of the differential counter	12.001 DPT_Value_4_Ucount
5	54	103	152	Ch.x - Differential counter	Value 4 bytes signed	Sends the signed value (-2147483648.. 2147483647) of the differential counter	13.001 DPT_Value_4_Count
5	54	103	152	Ch.x - 1 bit value object D	Value 1/0	Sends values 1/0 matched with object D	1.002 DPT_Bool
5	54	103	152	Ch.x - Single press 1 bit object D	Value 1/0	Sends values 1/0 matched with single press on object D	1.002 DPT_Bool
6	55	104	153	Ch.x - Sequence E	On/Off	Sends On/Off commands matched with object E of the sequence	1.001 DPT_Switch
6	55	104	153	Ch.x - Differential counter overflow bit	Overflow status	Sends the differential counter overflow feedback	1.002 DPT_Bool
6	55	104	153	Ch.x - 1 bit value object E	Value 1/0	Sends values 1/0 matched with object E	1.002 DPT_Bool
6	55	104	153	Ch.x - Single press 1 bit object E	Value 1/0	Sends values 1/0 matched with single press on object E	1.002 DPT_Bool
7	56	105	154	Ch.x - Temperature threshold 1 enabling status	Enabled/disabled	Sends the feedback regarding the temperature threshold 1 enabling/disabling status	1.003 DPT_Enable communication objects
7	56	105	154	Ch.x - Sequence F	On/Off	Sends On/Off commands matched with object F of the sequence	1.001 DPT_Switch
7	56	105	154	Ch.x - Differential counter overflow byte	Overflow status	Sends the value matched with the differential counter overflow feedback	5.010 DPT_Value_1_Ucount
7	56	105	154	Ch.x - 1 bit value object F	Value 1/0	Sends values 1/0 matched with object F	1.002 DPT_Bool
7	56	105	154	Ch.x - Single press 1 bit object F	Value 1/0	Sends values 1/0 matched with single press on object F	1.002 DPT_Bool
8	57	106	155	Ch.x - Sequence G	On/Off	Sends On/Off commands matched with object G of the sequence	1.001 DPT_Switch
8	57	106	155	Ch.x - 1 bit value object G	Value 1/0	Sends values 1/0 matched with object G	1.002 DPT_Bool
8	57	106	155	Ch.x - Single press 1 bit object G	Value 1/0	Sends values 1/0 matched with single press on object G	1.002 DPT_Bool
9	58	107	156	Ch.x - Temperature threshold 1 feedback	Value °C	Sends the current value of the temperature threshold 1 expressed in degrees Celsius	9.001 DPT_Temp
9	58	107	156	Ch.x - Temperature threshold 1 feedback	Value °K	Sends the current value of the temperature threshold 1 expressed in degrees Kelvin	9.002 DPT_Tempd

9	58	107	156	Ch.x - Temperature threshold 1 feedback	Value °F	Sends the current value of the temperature threshold 1 expressed in degrees Fahrenheit	9.027 DPT_Temp_F
9	58	107	156	Ch.x - Sequence H	On/Off	Sends On/Off commands matched with H Object of the sequence	1.001 DPT_Switch
9	58	107	156	Ch.x - 1 bit value object H	Value 1/0	Sends values 1/0 matched with object H	1.002 DPT_Bool
9	58	107	156	Ch.x - Single press 1 bit object H	Value 1/0	Sends values 1/0 matched with single press on object H	1.002 DPT_Bool
10	59	108	157	Ch.x - Temperature threshold 1 output	Value 1/0	Sends the 1/0 values matched to the temperature threshold 1 output	1.001 DPT_Switch
10	59	108	157	Ch.x - Temperature threshold 1 output	On/Off forced positioning	Sends the 2 bits values matched to the temperature threshold 1 output	2.001 DPT_Switch_Control
10	59	108	157	Ch.x - Temperature threshold 1 output	Value 0..255	Sends the unsigned values (0..255) matched to the temperature threshold 1 output	5.010 DPT_Value_1_Ucount
10	59	108	157	Ch.x - Temperature threshold 1 output	Value -128.. +127	Sends the signed values (-128..127) matched to the temperature threshold 1 output	6.010 DPT_Value_1_Count
10	59	108	157	Ch.x - Temperature threshold 1 output	Value 0% .. 100%	Sends the percentage values (0%..100%) matched to the temperature threshold 1 output	5.001 DPT_Percentage
10	59	108	157	Ch.x - Temperature threshold 1 output	HVAC mode (com/precom/eco/off)	Sends the HVAC modes (comfort/precomfort/economy/off) matched to the temperature threshold 1 output	20.102 DPT_HVAC_Mode
10	59	108	157	Ch.x - Temperature threshold 1 output	Value 0..65535	Sends the unsigned values (0..65535) matched to the temperature threshold 1 output	7.001 DPT_Value_2_Ucount
10	59	108	157	Ch.x - Temperature threshold 1 output	Value -32768..32767	Sends the signed values (-32768..32767) matched to the temperature threshold 1 output	8.001 DPT_Value_2_Count
10	59	108	157	Ch.x - Temperature threshold 1 output	Setpoint value in °C	Sends the values of the setpoint expressed in degrees Celsius matched to the temperature threshold 1 output	9.001 DPT_Value_Temp
10	59	108	157	Ch.x - Temperature threshold 1 output	Setpoint value in °K	Sends the values of the setpoint expressed in degrees Kelvin matched to the temperature threshold 1 output	9.002 DPT_Value_Tempd
10	59	108	157	Ch.x - Temperature threshold 1 output	Setpoint value in °F	Sends the values of the setpoint expressed in degrees Fahrenheit matched to the temperature threshold 1 output	9.027 DPT_Value_Temp_F
10	59	108	157	Ch.x - Double press 1 bit object A	Value 1/0	Sends values 1/0 matched with double press on object A	1.002 DPT_Bool
10	59	108	157	Ch.x - Double press 2 bits object A	On/Off forced positioning	Sends values 1/0 matched with double press on object A	1.002 DPT_Switch_Control
10	59	108	157	Ch.x - Double press 1 byte object A	Unsigned value	Sends unsigned values (0..255) matched with double press on object A	5.010 DPT_Value_1_Ucount
10	59	108	157	Ch.x - Double press 1 byte object A	Signed value	Sends signed values (-128..127) matched with double press on object A	6.010 DPT_Value_1_Count
10	59	108	157	Ch.x - Double press 1 byte object A	Value %	Sends the percentage values (0%..100%) matched with double press on object A	5.001 DPT_Scaling
10	59	108	157	Ch.x - Double press 1 byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) matched with the double press on object A	20.102 DPT_HVACMode
10	59	108	157	Ch.x - Double press 2 bytes object A	Unsigned value	Sends unsigned values (0..65535) matched with double press on object A	7.001 DPT_Value_2_Ucount
10	59	108	157	Ch.x - Double press 2 bytes object A	Signed value	Sends signed values (-32768..32767) matched with double press on object A	8.001 DPT_Value_2_Count

10	59	108	157	Ch.x - Double press 3 bytes object A	RGB colour	Sends the values of the three RGB colour components matched with the double press on object A	232.600 DPT_Colour_RGB
10	59	108	157	Ch.x - Double press 4 bytes object A	Unsigned value	Sends unsigned values (0.. 4294967295) matched with double press on object A	12.001 DPT_Value_4_Ucount
10	59	108	157	Ch.x - Double press 4 bytes object A	Signed value	Sends signed values (-2147483648.. 2147483647) matched with double press on object A	13.001 DPT_Value_4_Count
11	60	109	158	Ch.x - Double press 1 bit object B	Value 1/0	Sends values 1/0 matched with double press on object B	1.002 DPT_Bool
12	61	110	159	Ch.x - Double press 1 bit object C	Value 1/0	Sends values 1/0 matched with double press on object C	1.002 DPT_Bool
13	62	111	160	Ch.x - Temperature threshold 2 enabling status	Enabled/disabled	Sends the feedback regarding the temperature threshold 2 enabling/disabling status	1.003 DPT_Enable communication objects
13	62	111	160	Ch.x - Double press 1 bit object D	Value 1/0	Sends values 1/0 matched with double press on object D	1.002 DPT_Bool
14	63	112	161	Ch.x - Double press 1 bit object E	Value 1/0	Sends values 1/0 matched with double press on object E	1.002 DPT_Bool
15	64	113	162	Ch.x - Temperature threshold 2 feedback	Value °C	Sends the current value of the temperature threshold 2 expressed in degrees Celsius	9.001 DPT_Temp
15	64	113	162	Ch.x - Temperature threshold 2 feedback	Value °K	Sends the current value of the temperature threshold 2 expressed in degrees Kelvin	9.002 DPT_Tempd
15	64	113	162	Ch.x - Temperature threshold 2 feedback	Value °F	Sends the current value of the temperature threshold 2 expressed in degrees Fahrenheit	9.027 DPT_Temp_F
15	64	113	162	Ch.x - Double press 1 bit object F	Value 1/0	Sends values 1/0 matched with double press on object F	1.002 DPT_Bool
16	65	114	163	Ch.x - Temperature threshold 2 output	Value 1/0	Sends the 1/0 values matched to the temperature threshold 2 output	1.001 DPT_Switch
16	65	114	163	Ch.x - Temperature threshold 2 output	On/Off forced positioning	Sends the 2 bits values matched to the temperature threshold 2 output	2.001 DPT_Switch_Control
16	65	114	163	Ch.x - Temperature threshold 2 output	Value 0..255	Sends the unsigned values (0..255) matched to the temperature threshold 2 output	5.010 DPT_Value_1_Ucount
16	65	114	163	Ch.x - Temperature threshold 2 output	Value -128.. +127	Sends the signed values (-128..127) matched to the temperature threshold 2 output	6.010 DPT_Value_1_Count
16	65	114	163	Ch.x - Temperature threshold 2 output	Value 0% .. 100%	Sends the percentage values (0%..100%) matched to the temperature threshold 2 output	5.001 DPT_Percentage
16	65	114	163	Ch.x - Temperature threshold 2 output	HVAC mode (com/precom/eco/off)	Sends the HVAC modes (comfort/precomfort/economy/off) matched to the temperature threshold 2 output	20.102 DPT_HVAC_Mode
16	65	114	163	Ch.x - Temperature threshold 2 output	Value 0..65535	Sends the unsigned values (0..65535) matched to the temperature threshold 2 output	7.001 DPT_Value_2_Ucount
16	65	114	163	Ch.x - Temperature threshold 2 output	Value -32768..32767	Sends the signed values (-32768..32767) matched to the temperature threshold 2 output	8.001 DPT_Value_2_Count
16	65	114	163	Ch.x - Temperature threshold 2 output	Setpoint value in °C	Sends the values of the setpoint expressed in degrees Celsius matched to the temperature threshold 2 output	9.001 DPT_Value_Temp
16	65	114	163	Ch.x - Temperature threshold 2 output	Setpoint value in °K	Sends the values of the setpoint expressed in degrees Kelvin matched to the temperature threshold 2 output	9.002 DPT_Value_Tempd
16	65	114	163	Ch.x - Temperature threshold 2 output	Setpoint value in °F	Sends the values of the setpoint expressed in degrees Fahrenheit matched to the temperature threshold 2 output	9.027 DPT_Value_Temp_F

16	65	114	163	Ch.x - Double press 1 bit object G	Value 1/0	Sends values 1/0 matched with double press on object G	1.002 DPT_Bool
17	66	115	164	Ch.x - Double press 1 bit object H	Value 1/0	Sends values 1/0 matched with double press on object H	1.002 DPT_Bool
18	67	116	165	Ch.x - Triple press 1 bit object A	Value 1/0	Sends values 1/0 matched with triple press on object A	1.002 DPT_Bool
18	67	116	165	Ch.x - Triple press 2 bits object A	On/Off forced positioning	Sends values 1/0 matched with triple press on object A	1.002 DPT_Switch_Control
18	67	116	165	Ch.x - Triple press 1 byte object A	Unsigned value	Sends unsigned values (0..255) matched with triple press on object A	5.010 DPT_Value_1_Ucount
18	67	116	165	Ch.x - Triple press 1 byte object A	Signed value	Sends signed values (-128..127) matched with triple press on object A	6.010 DPT_Value_1_Count
18	67	116	165	Ch.x - Triple press 1 byte object A	Value %	Sends the percentage values (0%..100%) matched with triple press on object A	5.001 DPT_Scaling
18	67	116	165	Ch.x - Triple press 1 byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) matched with the triple press on object A	20.102 DPT_HVACMode
18	67	116	165	Ch.x - Triple press 2 bytes object A	Unsigned value	Sends unsigned values (0..65535) matched with triple press on object A	7.001 DPT_Value_2_Ucount
18	67	116	165	Ch.x - Triple press 2 bytes object A	Signed value	Sends signed values (-32768..32767) matched with triple press on object A	8.001 DPT_Value_2_Count
18	67	116	165	Ch.x - Triple press 3 bytes object A	RGB colour	Sends the values of the three RGB colour components matched with the triple press on object A	232.600 DPT_Colour_RGB
18	67	116	165	Ch.x - Triple press 4 bytes object A	Unsigned value	Sends unsigned values (0.. 4294967295) matched with triple press on object A	12.001 DPT_Value_4_Ucount
18	67	116	165	Ch.x - Triple press 4 bytes object A	Signed value	Sends signed values (-2147483648.. 2147483647) matched with triple press on object A	13.001 DPT_Value_4_Count
19	68	117	166	Ch.x - Temperature threshold 3 enabling status	Enabled/disabled	Sends the feedback regarding the temperature threshold 3 enabling/disabling status	1.003 DPT_Enable communication objects
19	68	117	166	Ch.x - Triple press 1 bit object B	Value 1/0	Sends values 1/0 matched with triple press on object B	1.002 DPT_Bool
20	69	118	167	Ch.x - Triple press 1 bit object C	Value 1/0	Sends values 1/0 matched with triple press on object C	1.002 DPT_Bool
21	70	119	168	Ch.x - Temperature threshold 3 feedback	Value °C	Sends the current value of the temperature threshold 3 expressed in degrees Celsius	9.001 DPT_Temp
22	71	120	169	Ch.x - Temperature threshold 3 feedback	Value °K	Sends the current value of the temperature threshold 3 expressed in degrees Kelvin	9.002 DPT_Tempd
21	70	119	168	Ch.x - Temperature threshold 3 feedback	Value °F	Sends the current value of the temperature threshold 3 expressed in degrees Fahrenheit	9.027 DPT_Temp_F
21	70	119	168	Ch.x - Triple press 1 bit object D	Value 1/0	Sends values 1/0 matched with triple press on object D	1.002 DPT_Bool
22	71	120	169	Ch.x - Temperature threshold 3 output	Value 1/0	Sends the 1/0 values matched to the temperature threshold 3 output	1.001 DPT_Switch
22	71	120	169	Ch.x - Temperature threshold 3 output	On/Off forced positioning	Sends the 2 bits values matched to the temperature threshold 3 output	2.001 DPT_Switch_Control
22	71	120	169	Ch.x - Temperature threshold 3 output	Value 0..255	Sends the unsigned values (0..255) matched to the temperature threshold 3 output	5.010 DPT_Value_1_Ucount
22	71	120	169	Ch.x - Temperature threshold 3 output	Value -128.. +127	Sends the signed values (-128..127) matched to the temperature threshold 3 output	6.010 DPT_Value_1_Count

22	71	120	169	Ch.x - Temperature threshold 3 output	Value 0% .. 100%	Sends the percentage values (0%..100%) matched to the temperature threshold 3 output	5.001 DPT_Percentage
22	71	120	169	Ch.x - Temperature threshold 3 output	HVAC mode (com/precom/eco/off)	Sends the HVAC modes (comfort/precomfort/economy/off) matched to the temperature threshold 3 output	20.102 DPT_HVAC_Mode
22	71	120	169	Ch.x - Temperature threshold 3 output	Value 0..65535	Sends the unsigned values (0..65535) matched to the temperature threshold 3 output	7.001 DPT_Value_2_Ucount
22	71	120	169	Ch.x - Temperature threshold 3 output	Value -32768..32767	Sends the signed values (-32768..32767) matched to the temperature threshold 3 output	8.001 DPT_Value_2_Count
22	71	120	169	Ch.x - Temperature threshold 3 output	Setpoint value in °C	Sends the values of the setpoint expressed in degrees Celsius matched to the temperature threshold 3 output	9.001 DPT_Value_Temp
22	71	120	169	Ch.x - Temperature threshold 3 output	Setpoint value in °K	Sends the values of the setpoint expressed in degrees Kelvin matched to the temperature threshold 3 output	9.002 DPT_Value_Tempd
22	71	120	169	Ch.x - Temperature threshold 3 output	Setpoint value in °F	Sends the values of the setpoint expressed in degrees Fahrenheit matched to the temperature threshold 3 output	9.027 DPT_Value_Temp_F
22	71	120	169	Ch.x - Triple press 1 bit object E	Value 1/0	Sends values 1/0 matched with triple press on object E	1.002 DPT_Bool
23	72	121	170	Ch.x - Triple press 1 bit object F	Value 1/0	Sends values 1/0 matched with triple press on object F	1.002 DPT_Bool
24	73	122	171	Ch.x - Triple press 1 bit object G	Value 1/0	Sends values 1/0 matched with triple press on object G	1.002 DPT_Bool
25	74	123	172	Ch.x - Temperature threshold 4 enabling status	Enabled/disabled	Sends the feedback regarding the temperature threshold 4 enabling/disabling status	1.003 DPT_Enable communication objects
25	74	123	172	Ch.x - Triple press 1 bit object H	Value 1/0	Sends values 1/0 matched with triple press on object H	1.002 DPT_Bool
26	75	124	173	Ch.x - Quadruple press 1 bit object A	Value 1/0	Sends values 1/0 matched with quadruple press on object A	1.002 DPT_Bool
26	75	124	173	Ch.x - Quadruple press 2 bits object A	On/Off forced positioning	Sends values 1/0 matched with quadruple press on object A	1.002 DPT_Switch_Control
26	75	124	173	Ch.x - Quadruple press 1 byte object A	Unsigned value	Sends unsigned values (0..255) matched with quadruple press on object A	5.010 DPT_Value_1_Ucount
26	75	124	173	Ch.x - Quadruple press 1 byte object A	Signed value	Sends signed values (-128..127) matched with quadruple press on object A	6.010 DPT_Value_1_Count
26	75	124	173	Ch.x - Quadruple press 1 byte object A	Value %	Sends the percentage values (0%..100%) matched with quadruple press on object A	5.001 DPT_Scaling
26	75	124	173	Ch.x - Quadruple press 1 byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) matched with the quadruple press on object A	20.102 DPT_HVACMode
26	75	124	173	Ch.x - Quadruple press 2 bytes object A	Unsigned value	Sends unsigned values (0..65535) matched with quadruple press on object A	7.001 DPT_Value_2_Ucount
26	75	124	173	Ch.x - Quadruple press 2 bytes object A	Signed value	Sends signed values (-32768..32767) matched with quadruple press on object A	8.001 DPT_Value_2_Count
26	75	124	173	Ch.x - Quadruple press 3 bytes object A	RGB colour	Sends the values of the three RGB colour components matched with the quadruple press on object A	232.600 DPT_Colour_RGB
26	75	124	173	Ch.x - Quadruple press 4 bytes object A	Unsigned value	Sends unsigned values (0.. 4294967295) matched with quadruple press on object A	12.001 DPT_Value_4_Ucount
26	75	124	173	Ch.x - Quadruple press 4 bytes object A	Signed value	Sends signed values	13.001 DPT_Value_4_Count

						(-2147483648.. 2147483647) matched with quadruple press on object A	
27	76	125	174	Ch.x - Temperature threshold 4 feedback	Value °C	Sends the current value of the temperature threshold 4 expressed in degrees Celsius	9.001 DPT_Temp
27	76	125	174	Ch.x - Temperature threshold 4 feedback	Value °K	Sends the current value of the temperature threshold 4 expressed in degrees Kelvin	9.002 DPT_Tempd
27	76	125	174	Ch.x - Temperature threshold 4 feedback	Value °F	Sends the current value of the temperature threshold 4 expressed in degrees Fahrenheit	9.027 DPT_Temp_F
27	76	125	174	Ch.x - Quadruple press 1 bit object B	Value 1/0	Sends values 1/0 matched with quadruple press on object B	1.002 DPT_Bool
28	77	126	175	Ch.x - Temperature threshold 4 output	Value 1/0	Sends the 1/0 values matched to the temperature threshold 4 output	1.001 DPT_Switch
28	77	126	175	Ch.x - Temperature threshold 4 output	On/Off forced positioning	Sends the 2 bits values matched to the temperature threshold 4 output	2.001 DPT_Switch_Control
28	77	126	175	Ch.x - Temperature threshold 4 output	Value 0..255	Sends the unsigned values (0..255) matched to the temperature threshold 4 output	5.010 DPT_Value_1_Ucount
28	77	126	175	Ch.x - Temperature threshold 4 output	Value -128.. +127	Sends the signed values (-128..127) matched to the temperature threshold 4 output	6.010 DPT_Value_1_Count
28	77	126	175	Ch.x - Temperature threshold 4 output	Value 0% .. 100%	Sends the percentage values (0%..100%) matched to the temperature threshold 4 output	5.001 DPT_Percentage
28	77	126	175	Ch.x - Temperature threshold 4 output	HVAC mode (com/precom/eco/off)	Sends the HVAC modes (comfort/precomfort/economy/off) matched to the temperature threshold 4 output	20.102 DPT_HVAC_Mode
28	77	126	175	Ch.x - Temperature threshold 4 output	Value 0..65535	Sends the unsigned values (0..65535) matched to the temperature threshold 4 output	7.001 DPT_Value_2_Ucount
28	77	126	175	Ch.x - Temperature threshold 4 output	Value -32768..32767	Sends the signed values (-32768..32767) matched to the temperature threshold 4 output	8.001 DPT_Value_2_Count
28	77	126	175	Ch.x - Temperature threshold 4 output	Setpoint value in °C	Sends the values of the setpoint expressed in degrees Celsius matched to the temperature threshold 4 output	9.001 DPT_Value_Temp
28	77	126	175	Ch.x - Temperature threshold 4 output	Setpoint value in °K	Sends the values of the setpoint expressed in degrees Kelvin matched to the temperature threshold 4 output	9.002 DPT_Value_Tempd
28	77	126	175	Ch.x - Temperature threshold 4 output	Setpoint value in °F	Sends the values of the setpoint expressed in degrees Fahrenheit matched to the temperature threshold 4 output	9.027 DPT_Value_Temp_F
28	77	126	175	Ch.x - Quadruple press 1 bit object C	Value 1/0	Sends values 1/0 matched with quadruple press on object C	1.002 DPT_Bool
29	78	127	176	Ch.x - Quadruple press 1 bit object D	Value 1/0	Sends values 1/0 matched with quadruple press on object D	1.002 DPT_Bool
30	79	128	177	Ch.x - Quadruple press 1 bit object E	Value 1/0	Sends values 1/0 matched with quadruple press on object E	1.002 DPT_Bool
31	80	129	178	Ch.x - Quadruple press 1 bit object F	Value 1/0	Sends values 1/0 matched with quadruple press on object F	1.002 DPT_Bool
32	81	130	179	Ch.x - Quadruple press 1 bit object G	Value 1/0	Sends values 1/0 matched with quadruple press on object G	1.002 DPT_Bool
33	82	131	180	Ch.x - Quadruple press 1 bit object H	Value 1/0	Sends values 1/0 matched with quadruple press on object H	1.002 DPT_Bool
34	83	132	181	Ch.x - Long press 1 bit object A	Value 1/0	Sends values 1/0 matched with long press on object A	1.002 DPT_Bool
34	83	132	181	Ch.x - Long press 2 bits object A	On/Off forced positioning	Sends values 1/0 matched with long press on object A	1.002 DPT_Switch_Control

34	83	132	181	Ch.x - Long press 1 byte object A	Unsigned value	Sends unsigned values (0..255) matched with long press on object A	5.010 DPT_Value_1_Ucount
34	83	132	181	Ch.x - Long press 1 byte object A	Signed value	Sends signed values (-128..127) matched with long press on object A	6.010 DPT_Value_1_Count
34	83	132	181	Ch.x - Long press 1 byte object A	Value %	Sends the percentage values (0%..100%) matched with long press on object A	5.001 DPT_Scaling
34	83	132	181	Ch.x - Long press 1 byte object A	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off) matched with the long press on object A	20.102 DPT_HVACMode
34	83	132	181	Ch.x - Long press 2 bytes object A	Unsigned value	Sends unsigned values (0..65535) matched with long press on object A	7.001 DPT_Value_2_Ucount
34	83	132	181	Ch.x - Long press 2 bytes object A	Signed value	Sends signed values (-32768..32767) matched with long press on object A	8.001 DPT_Value_2_Count
34	83	132	181	Ch.x - Long press 3 bytes object A	RGB colour	Sends the values of the three RGB colour components matched with the long press on object A	232.600 DPT_Colour_RGB
34	83	132	181	Ch.x - Long press 4 bytes object A	Unsigned value	Sends unsigned values (0.. 4294967295) matched with long press on object A	12.001 DPT_Value_4_Ucount
34	83	132	181	Ch.x - Long press 4 bytes object A	Signed value	Sends signed values (-2147483648.. 2147483647) matched with long press on object A	13.001 DPT_Value_4_Count
35	84	133	182	Ch.x - Long press 1 bit object B	Value 1/0	Sends values 1/0 matched with long press on object B	1.002 DPT_Bool
36	85	134	183	Ch.x - Long press 1 bit object C	Value 1/0	Sends values 1/0 matched with long press on object C	1.002 DPT_Bool
37	86	135	184	Ch.x - Long press 1 bit object D	Value 1/0	Sends values 1/0 matched with long press on object D	1.002 DPT_Bool
38	87	136	185	Ch.x - Long press 1 bit object E	Value 1/0	Sends values 1/0 matched with long press on object E	1.002 DPT_Bool
39	88	137	186	Ch.x - Long press 1 bit object F	Value 1/0	Sends values 1/0 matched with long press on object F	1.002 DPT_Bool
40	89	138	187	Ch.x - Long press 1 bit object G	Value 1/0	Sends values 1/0 matched with long press on object G	1.002 DPT_Bool
41	90	139	188	Ch.x - Long press 1 bit object H	Value 1/0	Sends values 1/0 matched with long press on object H	1.002 DPT_Bool

The variations of the objects shown in blue in the table above below are not shown for objects B (objects 3/52/101/150), C (objects 4/53/102/151), D (object 5/54/103/152), E (objects 6/55/104/153), F (objects 7/56/105/154), G (objects 8/57/106/155) and H (objects 9/58/107/156) due to space limitations, but they are present.

The variations of the objects shown in red in the table above below are not shown for objects B (objects 11/60/109/158), C (objects 12/61/110/159), D (object 13/62/111/160), E (objects 14/63/112/161), F (objects 15/64/113/162), G (objects 16/65/114/163) and H (objects 17/66/115/164) due to space limitations, but they are present.

The variations of the objects shown in orange in the table above are not shown for objects B (objects 19/68/117/166), C (objects 20/69/118/167), D (object 21/70/119/168), E (objects 22/71/120/169), F (objects 23/72/121/170), G (objects 24/73/122/171) and H (objects 25/74/123/172) due to space limitations, but they are present.

The variations of the objects shown in green in the table above are not shown for objects B (objects 27/75/125/174), C (objects 28/76/126/175), D (object 29/77/127/176), E (objects 30/78/128/177), F (objects 31/79/129/178), G (objects 32/81/130/179) and H (objects 33/82/131/180) due to space limitations, but they are present.

The variations of the objects shown in grey in the table above below are not shown for objects B (objects 35/84/133/182), C (objects 36/85/134/183), D (object 37/86/135/184), E (objects

38/92/136/185), F (objects 39/93/137/186), G (objects 40/89/138/187) and H (objects 41/90/139/188) due to space limitations, but they are present.

27.2 Communication objects with input functions

#				Object name	Function object	Description	Datapoint type
Ch 1	Ch 2	Ch 3	Ch 4				
1	50	99	148	Ch.x - Block	Activate/Deactivate	Used to activate/deactivate the block function	1.003 DPT_Enable communication objects
3	52	101	150	Ch.x - Scene storing trigger	Store	Receives the request (trigger) to send a scene storing message	1.017 DPT_Trigger
4	36	68	100	Ch.x - Reset maximum and minimum temperature	Reset values	Receives the trigger commands for resetting the maximum and minimum temperature values	1.017 DPT_Trigger
5	37	69	101	Ch.x - Feedback sending trigger	Send feedback	Receives the feedback sending request trigger commands	1.017 DPT_Trigger
6	38	70	102	Ch.x - Temperature threshold 1 enabling	0=enable / 1=disable	Receives the temperature threshold 1 enabling/disabling commands	1.002 DPT_Bool
6	38	70	102	Ch.x - Temperature threshold 1 enabling	1=enable / 0=disable	Receives the temperature threshold 1 enabling/disabling commands	1.002 DPT_Bool
8	40	72	104	Ch.x - Temperature threshold 1 regulation	1 = increase/0 = decrease	Receives the increase/decrease step commands for the temperature threshold 1 value	1.007 DPT_Step
8	40	72	104	Ch.x - Temperature threshold 1 value input	Value °C	Receives the temperature threshold 1 values expressed in degrees Celsius	9.001 DPT_Temp
8	40	72	104	Ch.x - Temperature threshold 1 value input	Value °K	Receives the temperature threshold 1 values expressed in degrees Kelvin	9.002 DPT_Tempd
8	40	72	104	Ch.x - Temperature threshold 1 value input	Value °F	Receives the temperature threshold 1 values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
11	43	75	107	Ch.x - Temperature threshold 1 output reference	HVAC mode (com/precom/eco/off)	Receives the reference HVAC modes (comfort/precomfort/economy/off) to calculate the output matched to temperature threshold 1	20.102 DPT_HVAC_Mode
11	43	75	107	Ch.x - Temperature threshold 1 output reference	Setpoint in °C	Receives the values in degrees Celsius of the reference setpoint to calculate the output matched to temperature threshold 1	9.001 DPT_Temp
11	43	75	107	Ch.x - Temperature threshold 1 output reference	Setpoint in °K	Receives the values in degrees Kelvin of the reference setpoint to calculate the output matched to temperature threshold 1	9.002 DPT_Tempd
11	43	75	107	Ch.x - Temperature threshold 1 output reference	Setpoint in °F	Receives the values in degrees Fahrenheit of the reference setpoint to calculate the output matched to temperature threshold 1	9.027 DPT_Temp_F
11	43	75	107	Ch.x - Temperature threshold 1 output status feedback	On/Off status	Receives feedback about the activation status of the output matched to temperature threshold 1	1.001 DPT_Switch
12	44	76	108	Ch.x - Temperature threshold 2 enabling	0=enable / 1=disable	Receives the temperature threshold 2 enabling/disabling commands	1.002 DPT_Bool
12	44	76	108	Ch.x - Temperature threshold 2 enabling	1=enable / 0=disable	Receives the temperature threshold 2 enabling/disabling commands	1.002 DPT_Bool

14	46	78	110	Ch.x - Temperature threshold 2 regulation	1 = increase/0 = decrease	Receives the increase/decrease step commands for the temperature threshold 2 value	1.007 DPT_Step
14	46	78	110	Ch.x - Temperature threshold 2 value input	Value °C	Receives the temperature threshold 2 values expressed in degrees Celsius	9.001 DPT_Temp
14	46	78	110	Ch.x - Temperature threshold 2 value input	Value °K	Receives the temperature threshold 2 values expressed in degrees Kelvin	9.002 DPT_Tempd
14	46	78	110	Ch.x - Temperature threshold 2 value input	Value °F	Receives the temperature threshold 2 values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
17	49	81	113	Ch.x - Temperature threshold 2 output reference	HVAC mode (com/precom/eco/off)	Receives the reference HVAC modes (comfort/precomfort/economy/off) to calculate the output matched to temperature threshold 2	20.102 DPT_HVAC_Mode
17	49	81	113	Ch.x - Temperature threshold 2 output reference	Setpoint in °C	Receives the values in degrees Celsius of the reference setpoint to calculate the output matched to temperature threshold 2	9.001 DPT_Temp
17	49	81	113	Ch.x - Temperature threshold 2 output reference	Setpoint in °K	Receives the values in degrees Kelvin of the reference setpoint to calculate the output matched to temperature threshold 2	9.002 DPT_Tempd
17	49	81	113	Ch.x - Temperature threshold 2 output reference	Setpoint in °F	Receives the values in degrees Fahrenheit of the reference setpoint to calculate the output matched to temperature threshold 2	9.027 DPT_Temp_F
17	49	81	113	Ch.x - Temperature threshold 2 output status feedback	On/Off status	Receives feedback about the activation status of the output matched to temperature threshold 2	1.001 DPT_Switch
18	50	82	114	Ch.x - Temperature threshold 3 enabling	0=enable / 1=disable	Receives the temperature threshold 3 enabling/disabling commands	1.002 DPT_Bool
18	50	82	114	Ch.x - Temperature threshold 3 enabling	1=enable / 0=disable	Receives the temperature threshold 3 enabling/disabling commands	1.002 DPT_Bool
20	52	84	116	Ch.x - Temperature threshold 3 regulation	1 = increase/0 = decrease	Receives the increase/decrease step commands for the temperature threshold 3 value	1.007 DPT_Step
20	52	84	116	Ch.x - Temperature threshold 3 value input	Value °C	Receives the temperature threshold 3 values expressed in degrees Celsius	9.001 DPT_Temp
20	52	84	116	Ch.x - Temperature threshold 3 value input	Value °K	Receives the temperature threshold 3 values expressed in degrees Kelvin	9.002 DPT_Tempd
20	52	84	116	Ch.x - Temperature threshold 3 value input	Value °F	Receives the temperature threshold 3 values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
23	55	87	119	Ch.x - Temperature threshold 3 output reference	HVAC mode (com/precom/eco/off)	Receives the reference HVAC modes (comfort/precomfort/economy/off) to calculate the output matched to temperature threshold 3	20.102 DPT_HVAC_Mode
23	55	87	119	Ch.x - Temperature threshold 3 output reference	Setpoint in °C	Receives the values in degrees Celsius of the reference setpoint to calculate the output matched to temperature threshold 3	9.001 DPT_Temp
23	55	87	119	Ch.x - Temperature threshold 3 output reference	Setpoint in °K	Receives the values in degrees Kelvin of the reference setpoint to calculate the output matched to temperature threshold 3	9.002 DPT_Tempd
23	55	87	119	Ch.x - Temperature threshold 3 output reference	Setpoint in °F	Receives the values in degrees Fahrenheit of the reference setpoint to calculate the output matched to temperature threshold 3	9.027 DPT_Temp_F

23	55	87	119	Ch.x - Temperature threshold 3 output status feedback	On/Off status	Receives feedback about the activation status of the output matched to temperature threshold 3	1.001 DPT_Switch
24	56	88	120	Ch.x - Temperature threshold 4 enabling	0=enable / 1=disable	Receives the temperature threshold 4 enabling/disabling commands	1.002 DPT_Bool
24	56	88	120	Ch.x - Temperature threshold 4 enabling	1=enable / 0=disable	Receives the temperature threshold 4 enabling/disabling commands	1.002 DPT_Bool
26	58	90	122	Ch.x - Temperature threshold 4 regulation	1 = increase/0 = decrease	Receives the increase/decrease step commands for the temperature threshold 4 value	1.007 DPT_Step
26	58	90	122	Ch.x - Temperature threshold 4 value input	Value °C	Receives the temperature threshold 4 values expressed in degrees Celsius	9.001 DPT_Temp
26	58	90	122	Ch.x - Temperature threshold 4 value input	Value °K	Receives the temperature threshold 4 values expressed in degrees Kelvin	9.002 DPT_Tempd
26	58	90	122	Ch.x - Temperature threshold 4 value input	Value °F	Receives the temperature threshold 4 values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
29	61	93	125	Ch.x - Temperature threshold 4 output reference	HVAC mode (com/precom/eco/off)	Receives the reference HVAC modes (comfort/precomfort/economy/off) to calculate the output matched to temperature threshold 4	20.102 DPT_HVAC_Mode
29	61	93	125	Ch.x - Temperature threshold 4 output reference	Setpoint in °C	Receives the values in degrees Celsius of the reference setpoint to calculate the output matched to temperature threshold 4	9.001 DPT_Temp
29	61	93	125	Ch.x - Temperature threshold 4 output reference	Setpoint in °K	Receives the values in degrees Kelvin of the reference setpoint to calculate the output matched to temperature threshold 4	9.002 DPT_Tempd
29	61	93	125	Ch.x - Temperature threshold 4 output reference	Setpoint in °F	Receives the values in degrees Fahrenheit of the reference setpoint to calculate the output matched to temperature threshold 4	9.027 DPT_Temp_F
29	61	93	125	Ch.x - Temperature threshold 4 output status feedback	On/Off status	Receives feedback about the activation status of the output matched to temperature threshold 4	1.001 DPT_Switch
42	91	140	189	Ch.x - Dimmer status feedback	On/Off status	Receives the dimmer status feedback	1.001 DPT_Switch
42	91	140	189	Ch.x - Status feedback	On/Off status	Receives the actuator status feedback for cyclical switching	1.001 DPT_Switch
42	91	140	189	Ch.x - Status feedback A	On/Off status	Receives the actuator status feedback for object A cyclical switching	1.001 DPT_Switch
42	91	140	189	Ch.x - Movement feedback	Up/down	Receives the feedback about the current movement direction of the motor command actuator	1.008 DPT_UpDown
42	91	140	189	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
43	92	141	190	Ch.x - Status feedback B	On/Off status	Receives the actuator status feedback for object B cyclical switching	1.001 DPT_Switch
43	92	141	190	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
44	93	142	191	Ch.x - Status feedback C	On/Off status	Receives the actuator status feedback for object C cyclical switching	1.001 DPT_Switch
44	93	142	191	Ch.x - Differential counter reset	Resets the value	Receives the value reset command for the differential counter	1.001 DPT_Switch

45	94	143	192	Ch.x - Status feedback D	On/Off status	Receives the actuator status feedback for object D cyclical switching	1.001 DPT_Switch
46	95	144	193	Ch.x - Status feedback E	On/Off status	Receives the actuator status feedback for object E cyclical switching	1.001 DPT_Switch
47	96	145	194	Ch.x - Status feedback F	On/Off status	Receives the actuator status feedback for object F cyclical switching	1.001 DPT_Switch
48	97	146	195	Ch.x - Status feedback G	On/Off status	Receives the actuator status feedback for object G cyclical switching	1.001 DPT_Switch
49	98	147	196	Ch.x - Status feedback H	On/Off status	Receives the actuator status feedback for object H cyclical switching	1.001 DPT_Switch
197	202	207	212	LED x -- Effect 1	Activate/Deactivate	Activate/deactivate- Light effect 1	1.001 DPT_Switch
198	203	208	213	LED x -- Effect 2	Activate/Deactivate	Activate/deactivate- Light effect 2	1.001 DPT_Switch
199	204	209	214	LED x -- Effect 3	Activate/Deactivate	Activate/deactivate- Light effect 3	1.001 DPT_Switch
200	205	210	215	LED x -- Effect 4	Activate/Deactivate	Activate/deactivate- Light effect 4	1.001 DPT_Switch
201	206	211	216	LED x -- Effect 5	Activate/Deactivate	Activate/deactivate- Light effect 5	1.001 DPT_Switch

code 70207188

LAST REVISION 09/2025

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

According to applicable UK regulations, the company responsible for placing the goods in UK market is:
GEWISS UK LTD - Unity House, Compass Point Business Park, 9 Stocks Bridge Way, ST IVES Cambridgeshire, PE27 5JL, United Kingdom tel: +44 1954 712757 E-mail: gewiss-uk@gewiss.com

+39 035 946 111
8:30 - 12:30 / 14:00 - 18:00
lunedì - venerdì / monday - friday

www.gewiss.com