KNX universal 4 IN + 4 OUT + binary 4 IN module - DIN rail mounting

GW 90728

Technical manual

Contents

1 Introduction

This manual explains the functions of the **KNX universal 4 IN + 4 OUT + binary 4 IN module - DIN rail** mounting (GW 90728) device and how they are set and configured with the aid of the ETS configuration software.

2 Application

The combined KNX universal 4 IN + 4 OUT + binary 4 IN module - DIN rail is a hybrid device that has 4 universal input channels, 4 PWM digital output channels for LED (3.3 V) and 4 input channels for potential-free contacts.
The universal input channels can be configured like potential free binary inputs (which can be used individually or combined to perform the On/Off command function, dimmer control, roller shutters control, scene management, priority and timed commands, brief / prolonged contact closure management, impulse count, on the KNX BUS), inputs for NTC temperature sensors, analogue inputs, inputs for measurement devices with an impulsive output.
The device is powered by the BUS line and equipped with 8 amber LEDS on the front signalling the status of the inputs.
The module is assembled on the DIN rail, inside the electric boards or junction boxes.

The device is configured with the ETS software, to perform the following functions.

The binary inputs for potential free contacts can be configured with the ETS software to operate as:
- sequence/control edges
- switching sequences
- impulse count
- multiple press management
- 1 push-button dimmer command (cyclical sending or stop command)
- roller shutter command with single push-button
- dimmer command with combined inputs (cyclical sending or stop command)
- roller shutter command with combined inputs
- scenes

The universal inputs/outputs can be configured with the ETS software to operate as follows:
- binary inputs for potential-free contacts (all channels)
    - sequence/control edges
    - switching sequences
    - impulse count
    - multiple press management
    - 1 push-button dimmer command (cyclical sending or stop command)
    - roller shutter command with single push-button
    - dimmer command with combined inputs (cyclical sending or stop command)
    - roller shutter command with combined inputs
    - scenes
- inputs for NTC temperature sensors (all channels)
    - temperature value measurement from external NTC sensors (GW10800 or GW1x900)
    - setting of threshold values with signalling of threshold exceeding and hysteresis management
    - settable thresholds from the BUS
- analogue inputs (all channels)
    - 1/2 inputs -> current measurement 0..20 mA or 4..20 mA
    - 3/4 inputs -> voltage measurement 0..10 V or 0...1 V
    - transmission on the BUS of the value measured with possible conversion scale/value percentage
    - setting of threshold values with signalling of threshold exceeding and hysteresis management
    - settable thresholds from the BUS
- PWM digital outputs for LEDs 3.3V (all channels)

- signalling of the status of the dedicated BUS communication object
- status or inverted status management (night-time signalling)
- brightness level % management via the PWM control

- digital inputs for measurement devices with an S0 interface (all channels)
  - measurement and conversion of the input value originating from the energy meters (KWh or Wh), instantaneous power (KW or W), water (volume in m3) or gas (volume in m3)
  - setting of threshold values with signalling of threshold exceeding
  - settable thresholds from the BUS
- thermostats (max 4) for controlling an equal number of temperature adjustment zones, with inputs for NTC temperature sensors (all channels)
  - temperature value measurement from external NTC sensors (GW10800 or GW1x900)

2.1 Association limits

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

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

3 *“Main”*menu

The **Main**, menu contains the application parameters for all the 8 input channels implemented by the device, as well as the 4 universal channels and the 4 binary channels.

The basic structure of the menu is as follows:

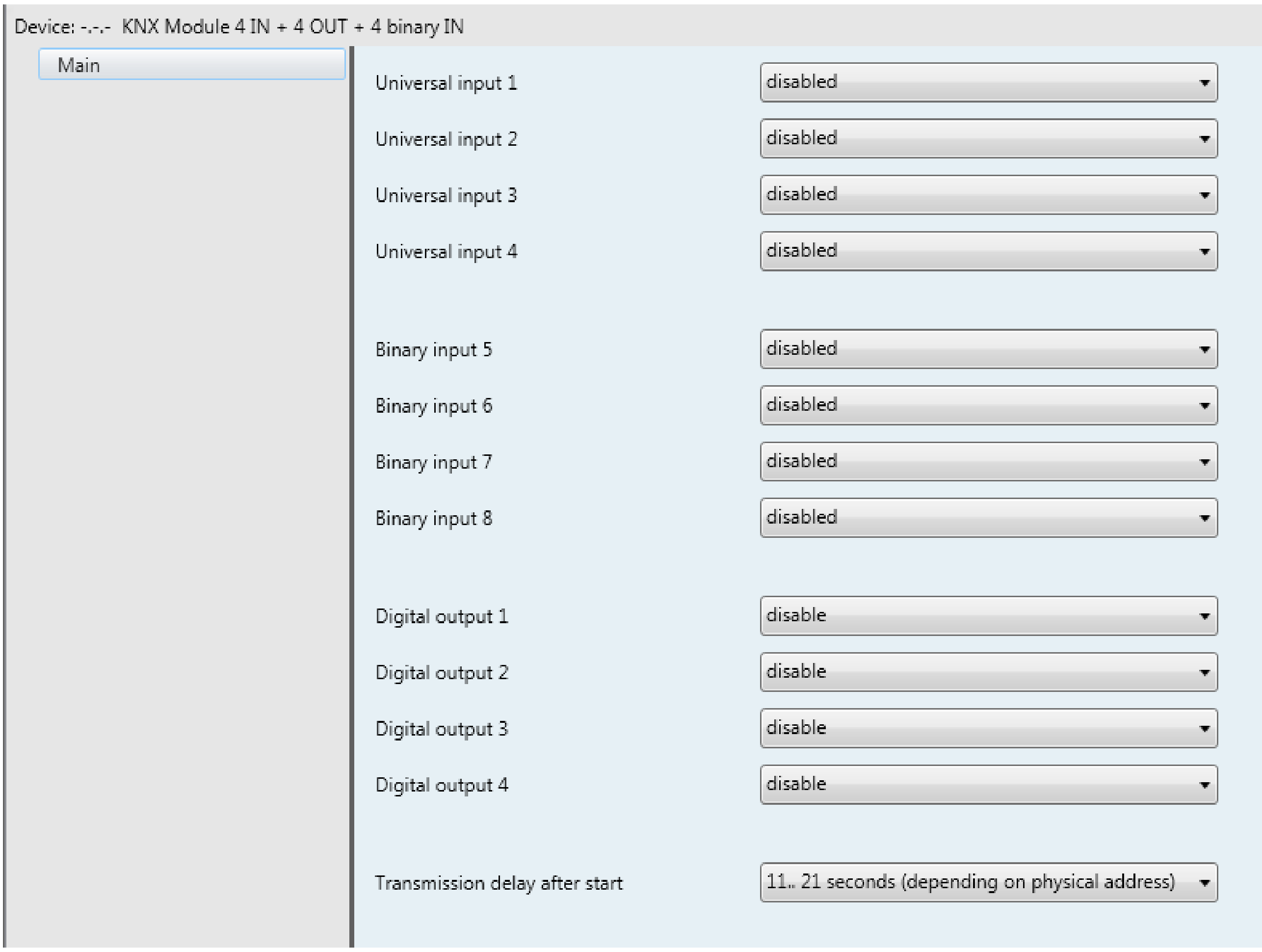


Fig. 3.1

3.1 Parameters

➢ *3.1.1 Universal input 1..4*

Each of the 4 input channels implemented by the module can be managed autonomously by carrying out a function that is autonomous with respect to the others; the parameters **“Universal input 1”**, **“Universal input 2”**, **“Universal input 3”**and **“Universal input 4”** define channel management.
The values that can be set are:

- **disabled** **(default value)**
- Binary input for potential-free contact
- Input for temperature sensor
- Analogue input
- Digital input with S0 interface (for input 2 only)
- Thermostat (for inputs 1 and 3 only)

selecting **binary input for potential free contact**, displays the configuration menu **Binary input 1**, **Binary input 2**, **Binary input 3** or **Binary input 4** (see chap. 4 "Binary input x" menu).
Selecting **input for temperature sensor** displays the configuration menu **Temperature sensor 1**, **Temperature sensor 2**, **Temperature sensor 3** or **Temperature sensor 4** (see Chap 5. "Temperature sensor x" menu).
Selecting **Analog input**, displays the configuration menu **Analog input 1**, **Analog input 2**, **Analog input 3** or **Analog input 4** (See Chap.6 "Analog input x" menu).
Selecting **Digital input with S0 interface** displays the configuration menu **Digital input 1 S0 interface**, **Digital input 2 S0 interface**, **Digital input 3 S0 interface** or **Digital input 4 S0 interface** (see Chap. 7 "Digital input x with S0 interface" menu).
Selecting **thermostat** displays the configuration menu **Thermostat 1**, **Thermostat 2**, **Thermostat 3** or **Thermostat 4** (see Chap. 8 "Thermostat x" menu).

Each of the 4 binary input channels implemented by the module can be managed autonomously by carrying out a function that is autonomous with respect to the others; the parameters **"Binary input 5"**, **"Binary input 6"**, **"Binary input 7"**and **"Binary input 8"** make it possible to enable the configuration of the relative input channels, displaying the configuration menus. The values that can be set are:

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

selecting **enabled**, displays the configuration menu **Binary input 5**, **Binary input 6**, **Binary input 7** or **Binary input 8** (See Chap.4 "Binary input x" menu).

Each of the 4 digital outputs implemented by the module can be managed autonomously by carrying out a function that is autonomous with respect to the others; the parameters **"Digital output 1"**, **"Digital output 2"**, **"Digital output 3"**and **"Digital output 4"** make it possible to enable the configuration of the relative output channels, displaying the configuration menus. The values that can be set are:

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

Selecting **enabled** displays the configuration menu **Digital output 1**, **Digital output 2**, **Digital output 3** or **Digital output 4** (see Chap. 9 "Digital output x" menu).

➢ *3.1.2 Delay time between switching on and first transmission*

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

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

Setting the values **11.. 21 seconds (depending on physical address)** and **5.. 9 seconds**, the device automatically calculates the transmission delay according to an algorithm that examines the physical address of the device itself; the presented values (11/21 or 5/9) indicate the extremes of the value interval that can be calculated.
The switch-on time of the device is estimated as approx. 8 seconds.

4 *“Binary input X”* menu

If the operating mode of the universal input is “binary input for a potential free contact” or if a binary input is enabled, a dedicated menu is displayed for each input called **Binary input X** (x = 1 .. 8, is the input index). The available functions both for a universal input configured as a “binary input for a potential free contact”, as well as for one of the four binary inputs 5,6,7,8 are the same and the menu structure changes depending on the value set for the **“Matched function”** parameter. For the sake of simplicity, the parameters enabled according to the value set for the above parameter are listed in the following paragraphs.

The basic structure of the menu is as follows:

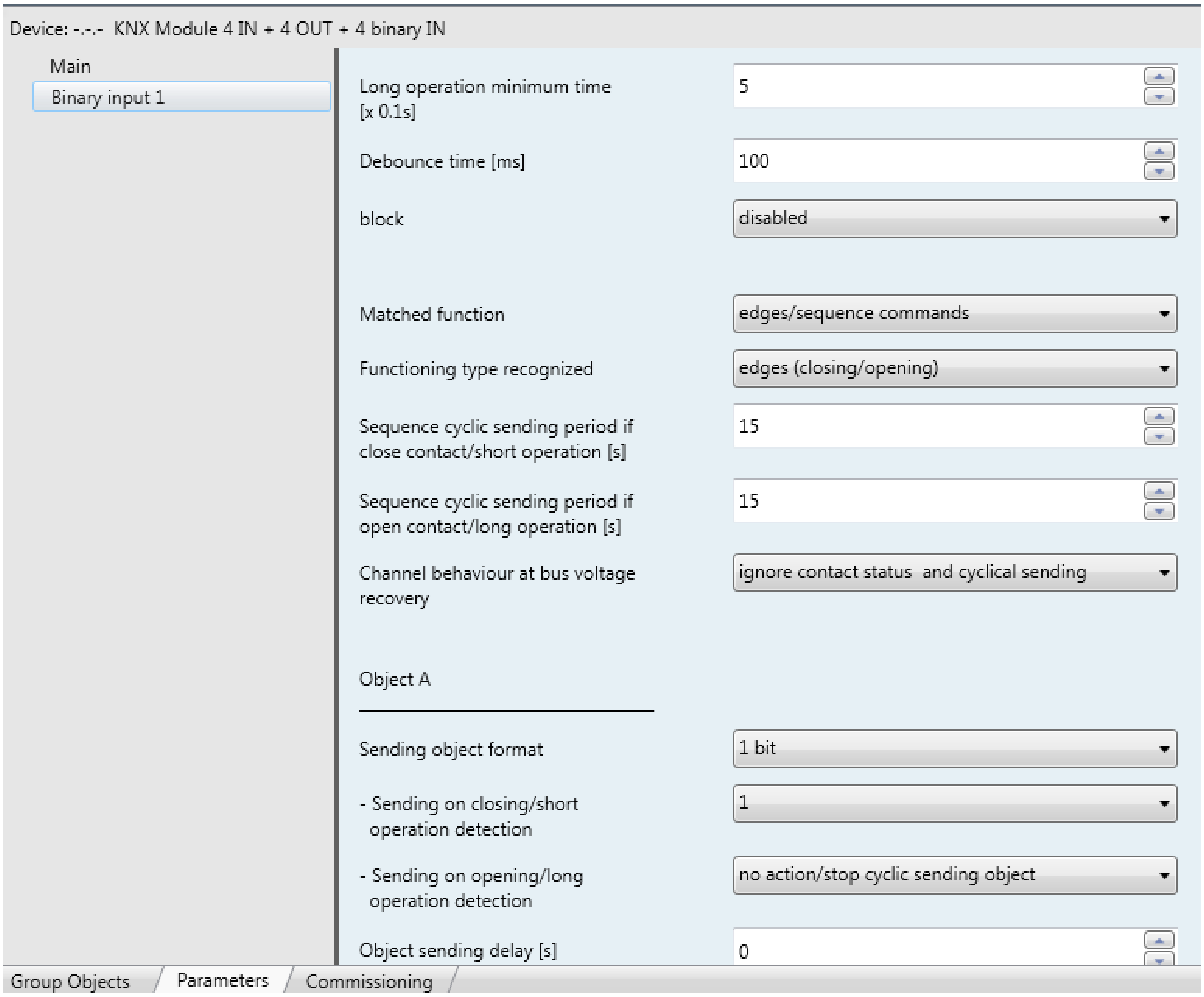


Fig. 4.1: “Binary input X” menu

4.1 Parameters

➢ 4.1.1 Long operation minimum time [x 0.1s]

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

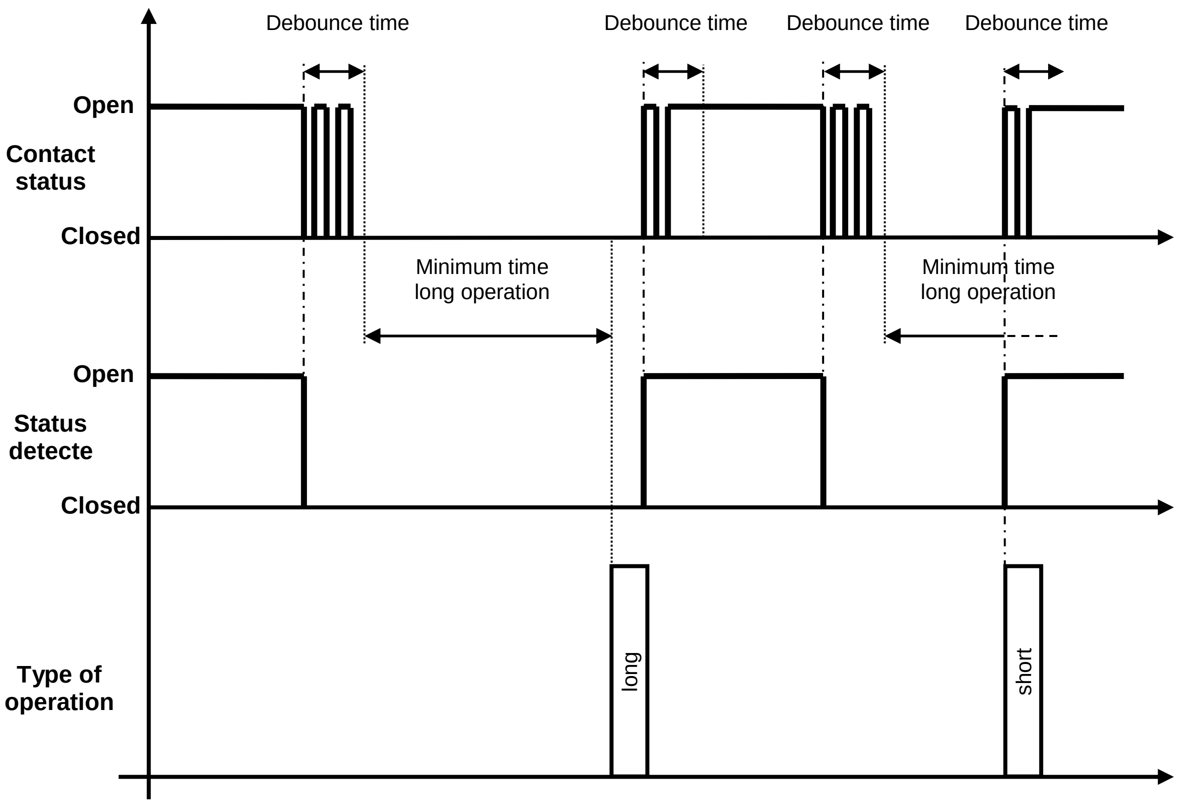
- from 3 to 150 with step 1, **5 (default value)**

➢ 4.1.2 Debounce time [ms]

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

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

The following chart summarises the concepts of **"Long operation minimum time [x 0.1s]"** and **"Debounce time [ms]"** explained above.



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

➢ *4.1.3 Block*

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

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

If **enabled** is selected, this displays the parameters **"Block activation value"**and **"Block function on bus voltage recovery"** and the communication object ***IN.x - Block*** (Data Point Type: 1.003 DPT_Enable), with which you can activate the function via the BUS command.

In particular cases where a front (opening or closure) or operation (short or long) is associated with the cyclical sending of a command/value, the block works in the following way:

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

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

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

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

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

➢ *4.1.4 Matched function*

The parameter used to define the function implemented by the binary input is **“Matched function".**
The values that can be set are:

- **edges/sequence commands** **(default value)**
  (See paragraph 4.1.4.1. “edges/sequence commands” function)
- 1 push-button + stop dimmer
  (See paragraph 4.1.4.2. “1 push-button + stop dimmer” function)
- cyclic sending 1 push-button dimmer
  (See paragraph 4.1.4.3. “Cyclical sending 1 push-button dimmer” function)
- 1 push-button shutter control
  (See paragraph 4.1.4.4. “1 push-button shutter control” function)
- 2 push-button + stop dimmer
  (See paragraph 4.1.4.5. “2 push-button + stop dimmer” function)
- cyclical sending 2 push-button dimmer
  (See paragraph 4.1.4.6. “Cyclical sending 2 push-button dimmer” function)
- 2 push-button roller shutters control
  (See paragraph 4.1.4.7. “2 push-button roller shutters control” function)
- scene management
  (See paragraph 4.1.4.8 “Scene management” function)
- switching sequences
  (See paragraph 4.1.4.9. “Switching sequences” function)
- pulse counter
  (See paragraph 4.1.4.10. “Pulse counter” function)
- multiple press/closing contact
  (See paragraph 4.1.4.11. “Multiple press/closing contact” function)

➢ *4.1.4.1 “Edges/Sequence commands” function*

This function is used to set the type and number of commands to send after a status change has been detected, for up to a total of 4 commands per input; The value of the command can be differentiated according to the event detected (closure/opening, or short/long operation). The sending of commands can also be delayed with a set fixed time, and the cyclical sending of command telegrams can be enabled.

The basic structure of the menu is as follows:

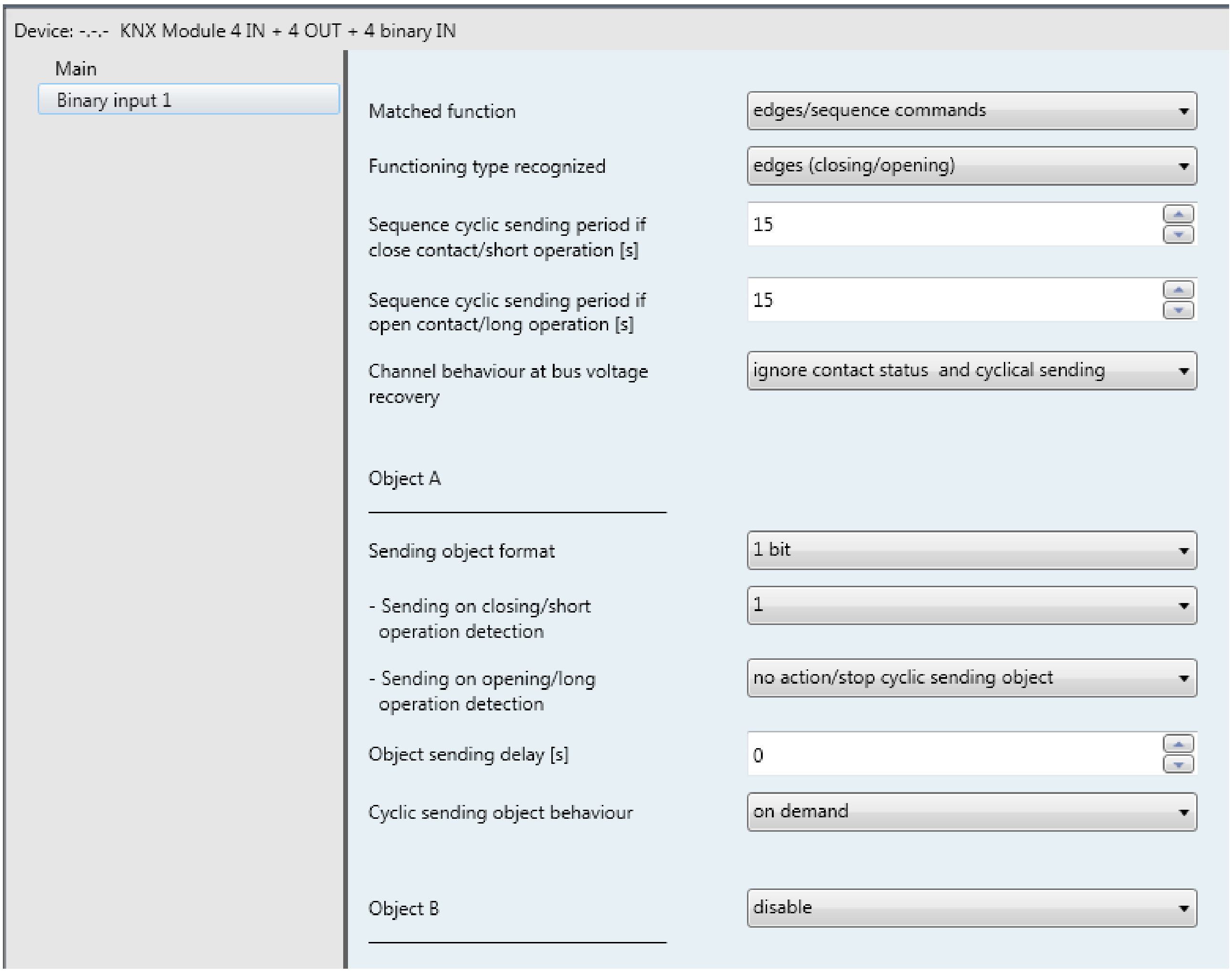


Fig. 4.2: "Edges/Sequence commands" matched function menu

Enabling this function makes the following parameters available for configuration:

➢ *4.1.4.1.1 Type of recognised activation (Functioning type recognized)*

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

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

➢ *4.1.4.1.2 Sequence cyclic sending period if close contact/short operation [s]*

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

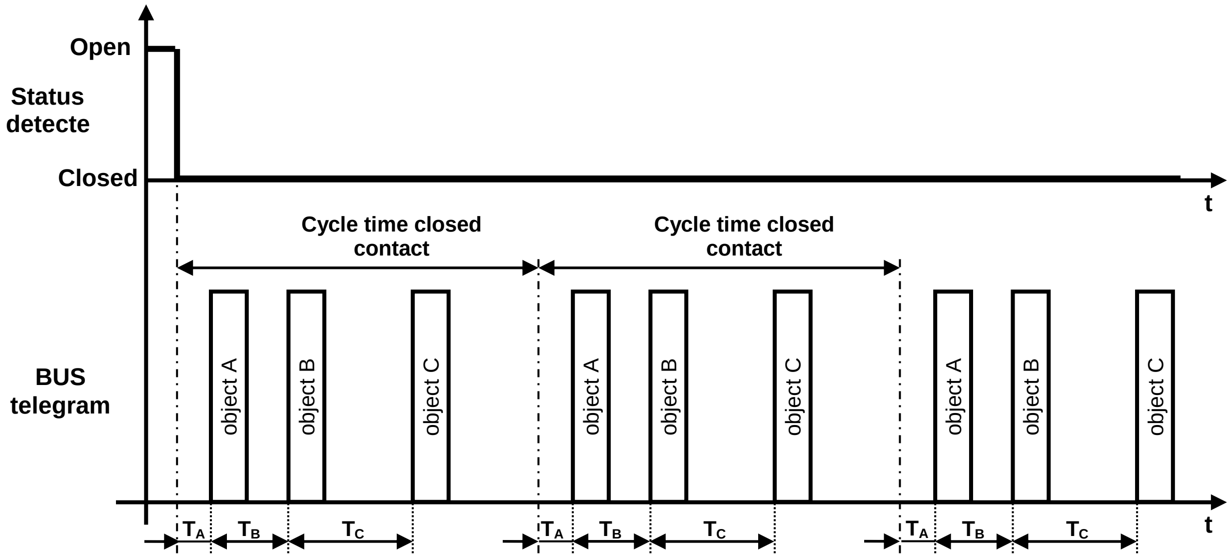
- from 1 to 65535 with steps of 1, **15 (default value)**

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

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

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

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



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

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

4.1.4.1.4 Binary input x behaviour on BUS voltage recovery

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

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

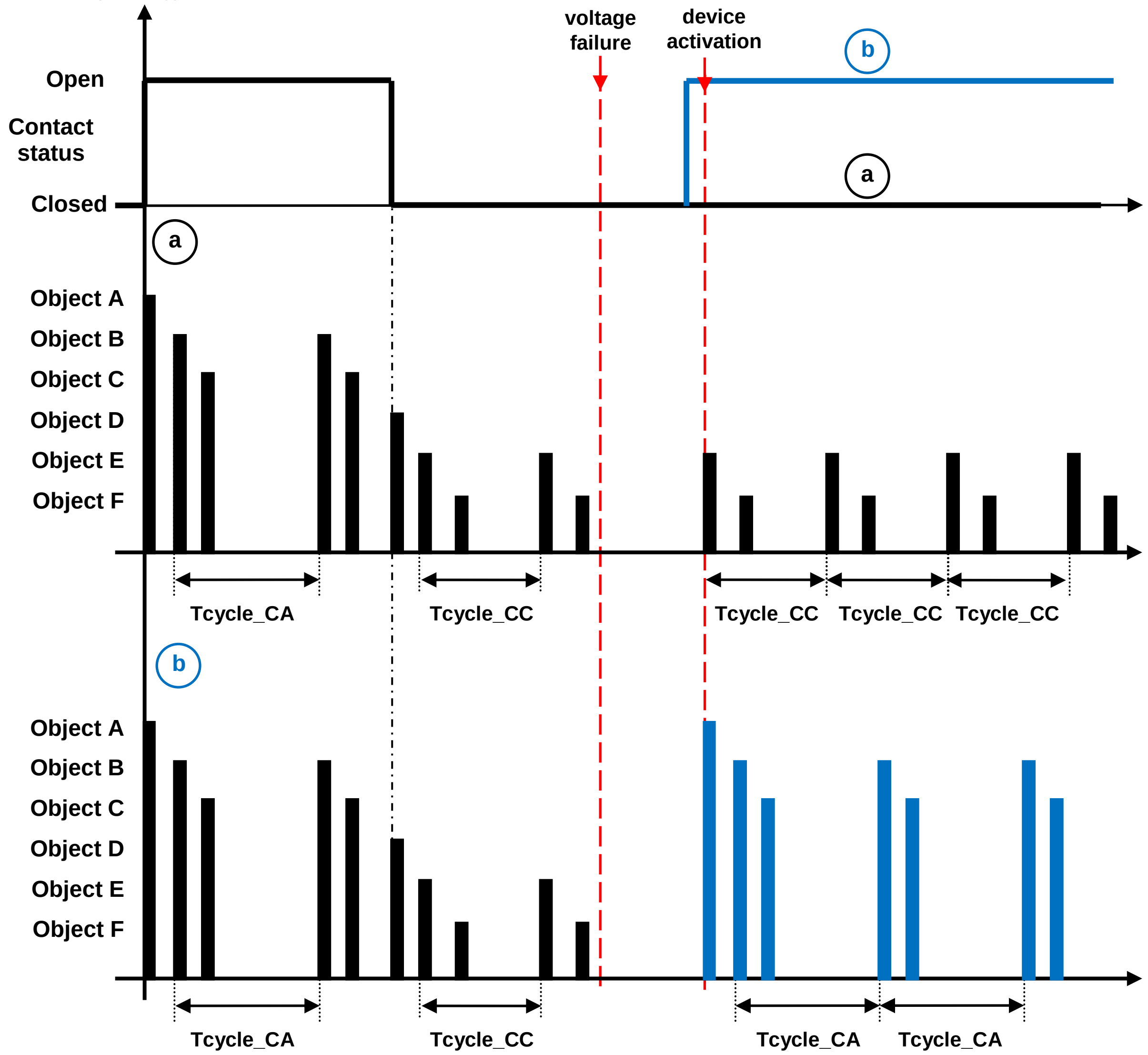
By selecting **evaluate contact status and cyclical sending**, the device behaves in the following way:

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

- if the recognised type of operation is **short operation/long operation**, the device checks the last operation recognised before the voltage failure and, after evaluating the value set for the **"Sending object condition"** items of all the objects of the sequence, it sends only those telegrams for which cyclical sending is enabled (as if the voltage failure had not occurred).

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

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



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

Chart "a"

- On the opening of the contact, the device sends the sequence of telegrams A, B and C on the basis of the set sending delays
- after a period of time equal to the period of cyclical telegram sending with an open contact (Tcycle_CA), the device again sends objects B, C for which cyclical sending is enabled
- on the closure of the contact, the device sends the sequence of telegrams D, E and F on the basis of the set sending delays

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

Chart “b”

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

➢ *4.1.4.1.5 Object A-B-C-D*

For each binary input, up to 4 different objects can be sent (distinguished by the letters A, B, C and D) on the basis of the closure (or short operation) or opening (or long operation) of the contact; object A is always enabled, whereas the parameter **“Object z”** (z is the index of the object associated with the threshold, between **A** and **D)** can be used to enable a new object to be sent. The parameter may assume the following values:

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

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

➢ *4.1.4.1.6 Sending object format*

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

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

- 4 bytes unsigned value
- 4 bytes signed value
- 14 bytes

The value set for this item will alter the values that can be set for the **"Sending on closing/short operation detection"** and **"Sending on opening/long operation detection"** parameters.
The parameter **"Sending on closing/short operation detection"** parameter is used to set the command or value to be sent following the detection of the closure or short operation of the contact (depending on the type of operation selected) associated with the binary input.
The parameter **"Sending on opening/long operation detection"** parameter is used to set the command or value to be sent following the detection of the opening or long operation of the contact (depending on the type of operation selected) associated with the binary input.

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

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

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

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

  Selecting the value **enabled** displays the ***IN.x - z object status feedback*** communication object. In this case, 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.

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

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

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

- If the format of the object to send is **1 byte unsigned**, the ***IN.x - 1 byte value z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount), and the values that can be set for the two parameters listed above are:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- If the format of the object to send is **2 byte unsigned**, the ***IN.x 2 byte value z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the two parameters listed above are:

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

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

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

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

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

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

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

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

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

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

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

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

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

- If the format of the object to send is **4 byte unsigned**, the ***IN.x 4 byte value z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the two parameters listed above are:

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

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

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

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

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

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

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

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

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

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

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

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

4.1.4.1.7 Object sending delay (0..255 seconds)

The parameter **“Object sending delay (0.. 255 seconds)”** parameter sets the delay between the detection of the event associated with the sending of the command, and the effective sending of the command/value on the BUS. With regard to the objects that range from index B to index D, this parameter indicates the delay between sending the command/value associated with the object with the previous index (z-1) and sending the command/value associated with the object to which the parameter refers; the delay in these cases is calculated from the moment when the command/value associated with the object with the previous index (z-1) is sent, not from the moment of detection of the event that generated the sending (closure/opening or short/long operation).
The set delay will only be executed if the event in progress, associated with the object to which the parameter refers, is associated with any value other than **no action**; otherwise, the delay is ignored.
The parameter may assume the following values:

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

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

4.1.4.1.8 *Cyclical sending object condition*

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 **“Cyclic sending object condition”** parameter defines the conditions for the cyclical sending of the command telegrams. The values that can be set are:

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

By selecting **never**, the device will only send the telegram with the set value on the BUS when the contact changes from closed to open or vice versa (or when a short/long operation is detected on the contact).

By selecting **in the case of an open contact/long operation**, the device will only send the telegram with the set value on the BUS when the contact changes from closed to open (or when a long operation is detected on the contact). As long as the contact remains open (or no other operation is recognised), the device will occasionally send the value associated with the event; if a new long operation is recognised, this cyclical sending is interrupted and the sending of the sequence associated with the detected operation restarts.

By selecting **in the case of a closed contact/short operation**,the device will only send the telegram with the set value on the BUS when the contact changes from open to closed (or when a short operation is detected on the contact). As long as the contact remains closed (or no other operation is recognised), the device will occasionally send the value associated with the event; if a new short operation is recognised, the sending of the sequence associated with the detected operation restarts.

By selecting **always**, the device will only send the telegram with the set value on the BUS when the contact changes from closed to open or vice versa (or when a short/long operation is detected on the contact). The command telegram associated with the detected event is repeated at regular intervals. If a short/long operation is recognised, this cyclical sending is interrupted and the sending of the sequence associated with the detected operation restarts.

If the value **no action/stop cyclic sending object** is associated with a specific operation for all the objects enabled, then the cyclical condition will be ignored even if it is enabled. If cyclical sending is active (determined by the setting of the other operation), this is terminated.

➢ 4.1.4.2 "1 push-button + stop dimmer" function

This is used to configure the input to control a dimmer with a single push-button, increasing and decreasing dimmer brightness always using the same input.
For sending on/off telegrams and brightness control telegrams.
As there is only one input 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 enabled by this function are ***IN.x - Switching*** (Data Point Type: 1.001 DPT_Switch) and ***IN.x - Brightness dimming*** (Data Point Type: 3.007 DPT _Control_Dimming).

The structure of the menu is as follows:

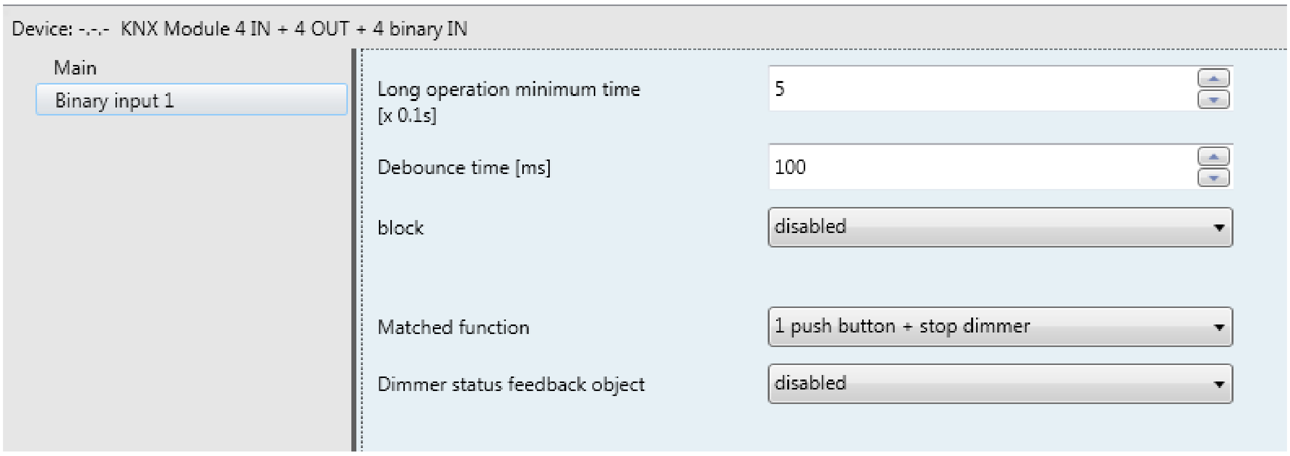


Fig. 4.3: "1 push-button + stop dimmer" matched function menu

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

This behaviour is modified if the user enables the ***IN.x - Dimmer status feedback*** communication object (Data Point Type: 1.001 DPT_Switch), via the parameter **"Dimmer status feedback object"**.

Enabling this function makes the following parameters available for configuration:

➢ *4.1.4.2.1 Dimmer status feedback object*

This parameter may have the following values:

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

If **enabled** is selected, the **"Brightness control commands with dimmer on"** parameter is visualised, along with the communication object ***IN.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 **"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.

➢ *4.1.4.3 "Cyclical sending 1 push-button dimmer" function*

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

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

Unlike the ***1 push-button + stop dimmer*** function, it is possible to define both the brightness variation steps and the time that must elapse between the sending of one command and another when the long operation is drawn out over time. The sending of the "regulation stop" telegram on contact opening is not therefore necessary, because although the regulation does follow the characteristic power/brightness curve, it is the command sent by the device that determines the percentage variation. The communication objects enabled by this function are ***IN.x - Switching*** (Data Point Type: 1.001 DPT_Switch) and ***IN.x - Brightness dimming*** (Data Point Type: 3.007 DPT _Control_Dimming).

The structure of the menu is as follows:

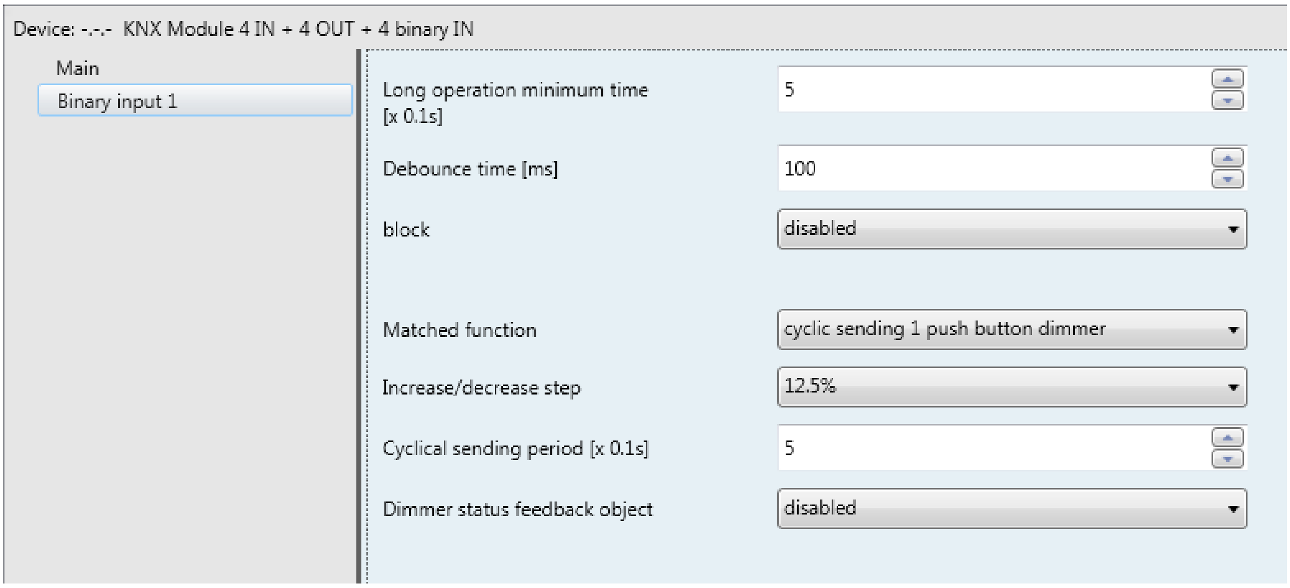


Fig. 4.4: “Cyclic sending 1 push-button dimmer” matched function menu

Enabling this function makes the following parameters available for configuration:

➢ *4.1.4.3.1 Increase/decrease step*

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

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

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

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

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

- from 3 to 50 with steps of 1 **(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 closure is detected:

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

➢ *4.1.4.3.3 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 paragraph 4.1.4.2 “1 push-button + stop dimmer” function.
The parameter used to enable the feedback object is **“Dimmer status feedback object”** which can have the following values:

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

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

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

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.

➢ *4.1.4.4 “1 push-button shutter control” function*

This is used to configure the input 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 input 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 ***IN.x - Shutter movement*** object or of the movement feedback received via the ***IN.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 ***IN.x - Shutter movement*** object or the movement feedback received via the ***IN.x - Movement feedback*** object, depending on which of the two events occurred last; if the last event that occurred is “upward movement feedback reception” or “send upward movement command”, the command will be a "closing louvres adjustment" command, and vice versa. If the shutter is moving, the louvre adjustment command will only stop the shutter up/down movement.

The communication objects enabled by this function are ***IN.x - Shutter movement*** (Data Point Type: 1.008 DPT_UpDown), ***IN.x - Louvre stop/adjustment*** (Data Point Type: 1.007 DPT_Step) and ***IN.x - Movement feedback*** (Data Point Type: 1.008 DPT_UpDown).

The structure of the menu is as follows:

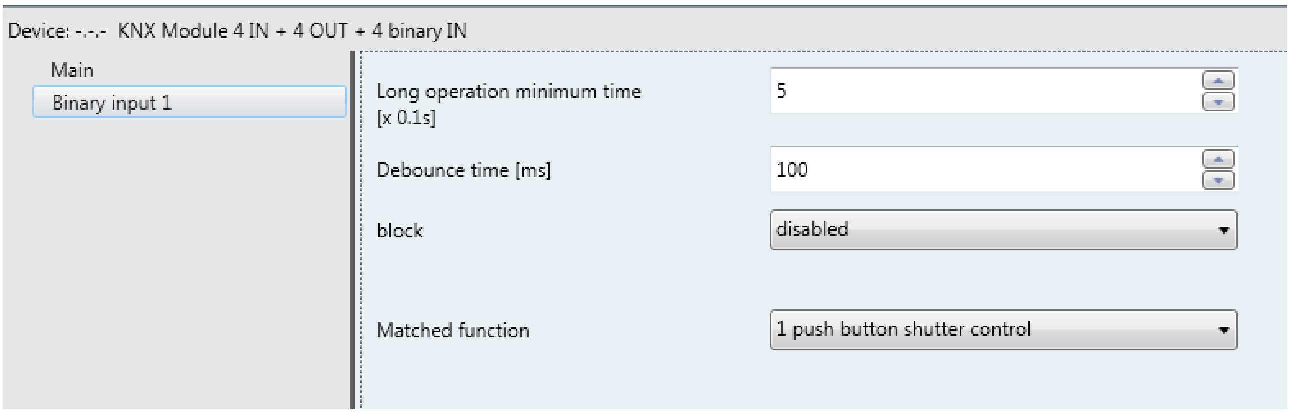


Fig. 4.5: “1 push-button shutter control” matched function menu

No new parameters are enabled with this function.

➢ *4.1.4.5 “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 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 enabled by this function are ***IN.x - Switching*** (Data Point Type: 1.001 DPT_Switch) and ***IN.x - Brightness dimming*** (Data Point Type: 3.007 DPT _Control_Dimming).

The structure of the menu is as follows:

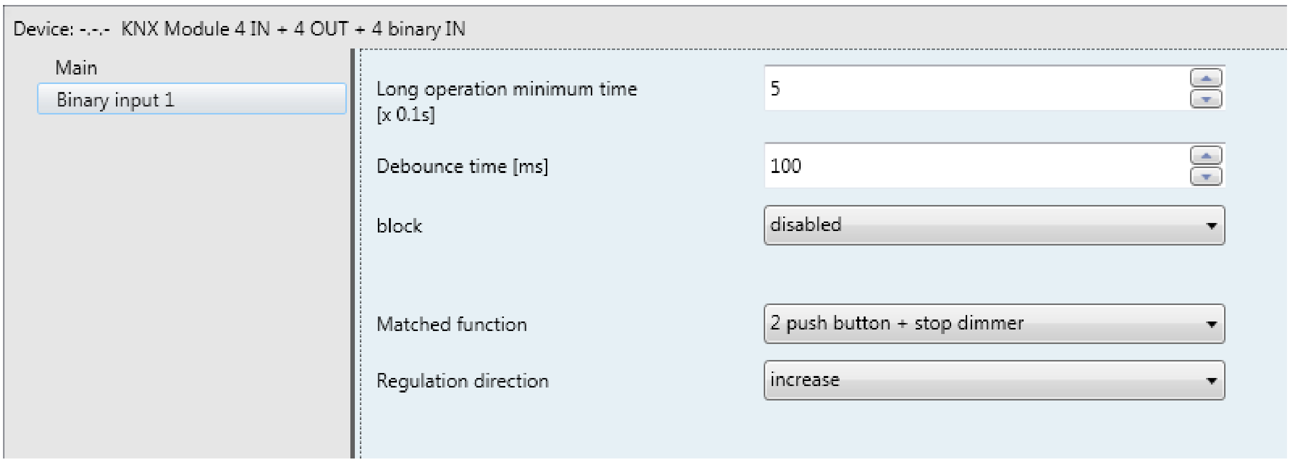


Fig. 4.6: “2 push-button + stop dimmer” matched function menu

Enabling this function makes the following parameters available for configuration:

➢ 4.1.4.5.1 Control direction (=Regulation direction)

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

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

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

➢ 4.1.4.6 "Cyclic sending 2 push-button 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 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 variation steps of the brightness increase/decrease commands as well as the time that must elapse between the sending of one command and another when the push-button remains pressed; The sending of the "regulation stop" telegram on push-button release is not therefore necessary, because although the regulation does follow the characteristic power/brightness curve, it is the command sent by the device that determines the percentage variation. The communication objects enabled by this function are ***IN.x - Switching*** (Data Point Type: 1.001 DPT_Switch) and ***IN.x - Brightness dimming*** (Data Point Type: 3.007 DPT _Control_Dimming).

The structure of the menu is as follows:

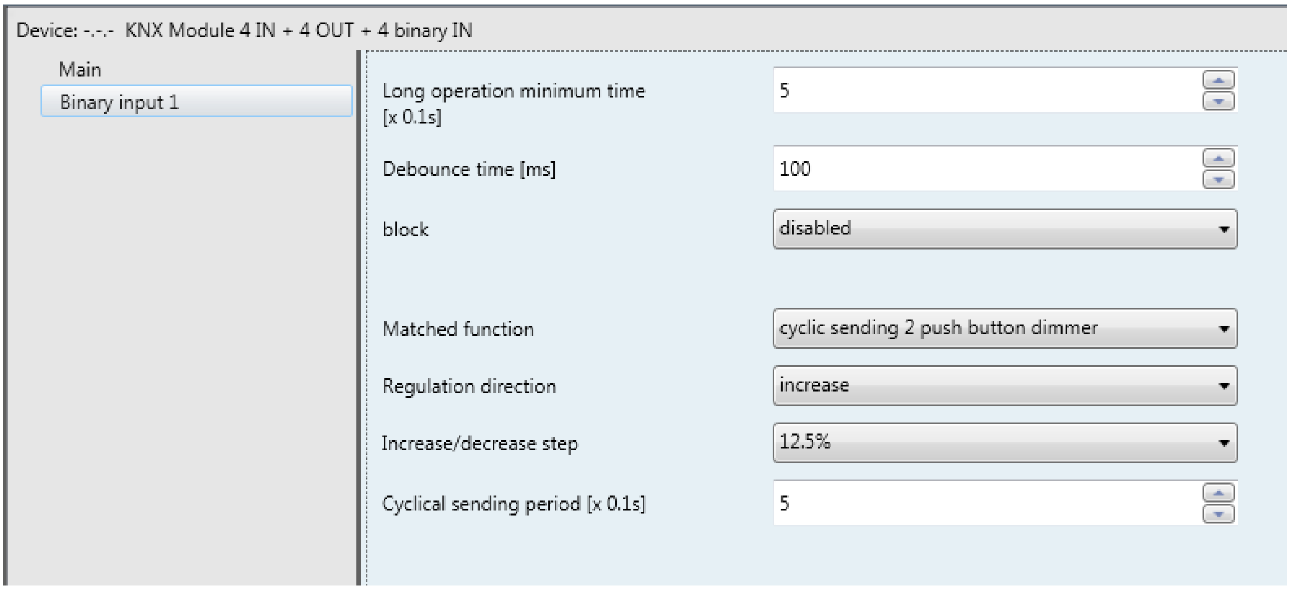


Fig. 4.7: “Cyclic sending 2 push-button dimmer” matched function menu

Enabling this function makes the following parameters available for configuration:

➢ *4.1.4.6.1 Control direction (=Regulation direction)*

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

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

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

➢ *4.1.4.6.2 Increase/decrease step*

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

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

If the contact remains closed after recognising the long operation, the device sends the command cyclically until contact opening is detected.

➢ 4.1.4.6.3 "Cyclical sending period [x 0.1s]"

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

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

To sum up, when a long 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 closure is detected:

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

➢ 4.1.4.7 "2 push-button shutter control" function

This is used to configure the input to control a shutter/venetian blind with two push-buttons, managing in this case only one of the two movement directions (down or up).
Up or down movement telegrams or louvres open or close control telegrams can be sent. Also in this case, there is a difference between short and long 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 louvres opening control command (or stop movement); If the set adjustment direction is "down" the sent command will only be a louvres closing control command (or stop movement).

The communication objects enabled by this function are ***IN.x - Shutter movement*** (Data Point Type: 1.008 DPT_UpDown) and ***IN.x - Louvre stop/adjustment*** (Data Point Type: 1.007 DPT_Step).

The structure of the menu is as follows:

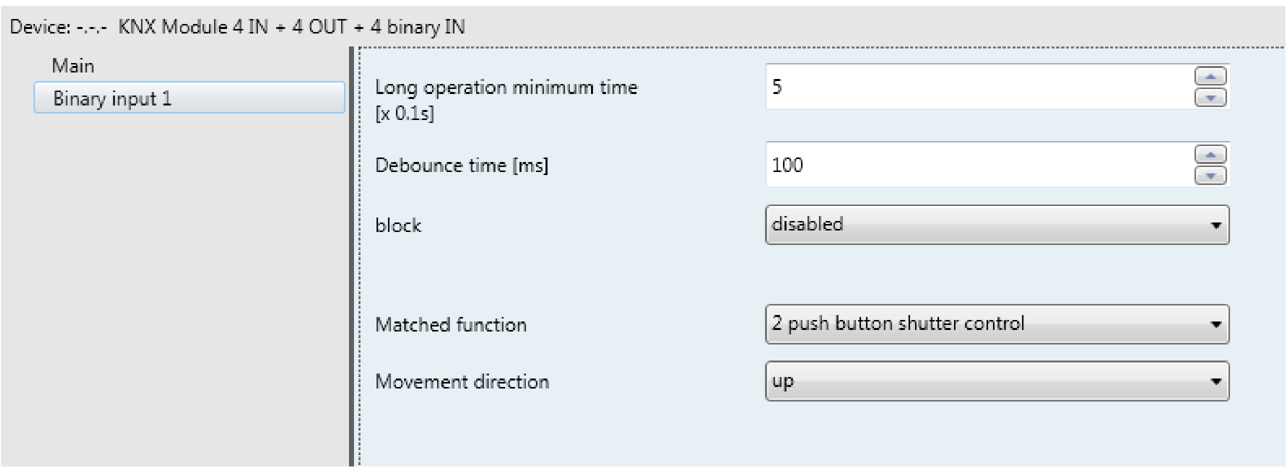


Fig. 4.8: "2 push-button shutter control" matched function menu

Enabling this function makes the following parameters available for configuration:

➢ *4.1.4.7.1 Movement direction*

The parameter **"Movement direction"** is used to configure the direction of movement of shutter the input controls; The values that can be set are:

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

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

➢ *4.1.4.8 "Scene management" function*

This is used to configure the input 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 input.
There is a difference between short and long operations:

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

The communication objects enabled by this function are ***IN.x - Scene*** (Data Point Type: 18.001 DPT_SceneControl) and ***IN.x - Scene storing trigger*** (Data Point Type: 1.017 DPT_Trigger).

The structure of the menu is as follows:

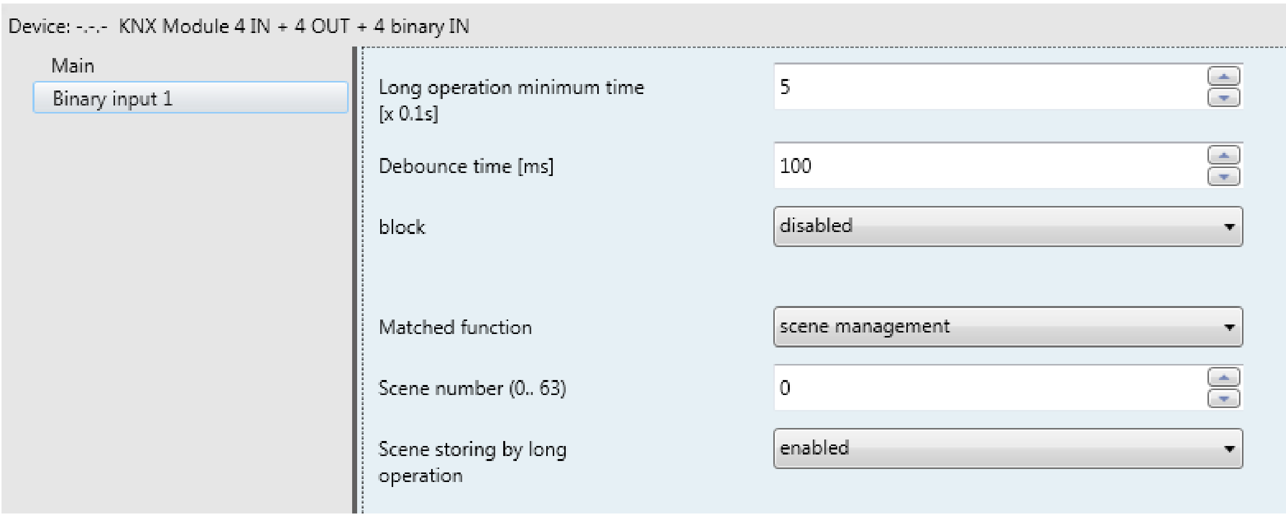


Fig. 4.9: "Scene management" matched function menu

Enabling this function makes the following parameters available for configuration:

4.1.4.8.1 *Scene number (0..63)*

The **“Scene number (0.. 63)”** parameter is used to set the value of the scene to be recalled/stored and as a result the relative values that are sent via the ***IN.x - Scene*** object. The possible values are:

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

4.1.4.8.2 *Scene storing by long operation*

The parameter **“Scene storing by long operation”** enables the sending of a scene memorising command when a long operation is recognised. The values that can be set are:

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

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

4.1.4.9 *“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:

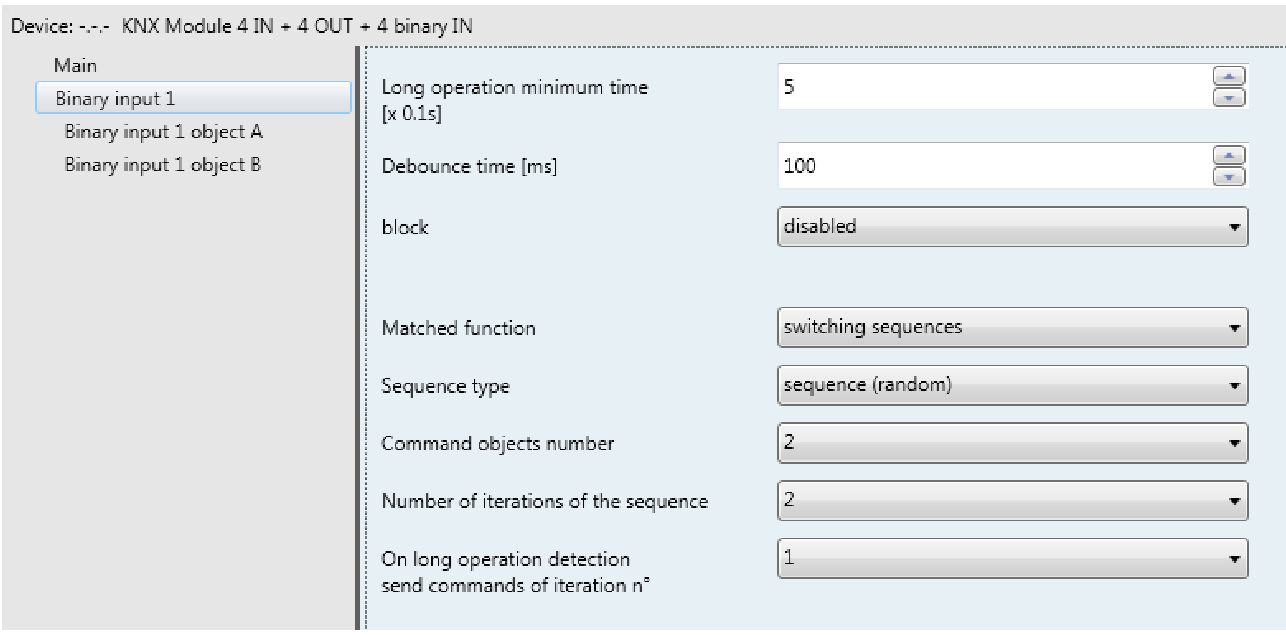


Fig. 4.10: “Switching sequences” matched function menu

Enabling this function makes the following parameters available for configuration:

➢ 4.1.4.9.1 Number of objects to send (Command objects number)

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

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

➢ 4.1.4.9.2 Sequence type

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

Edge no.	Value sent on ***IN.x – Sequence C***	Value sent on ***IN.x – Sequence B***	Value sent on ***IN.x – Sequence A***
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

Note: 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 ***IN.x - C sequence*** whereas the least significant is always the one for the object ***IN.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 ***IN.x – Sequence C***	Value sent on ***IN.x – Sequence B***	Value sent on ***IN.x – Sequence A***
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

Note: 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 ***IN.x - C sequence*** whereas the least significant is always the one for object ***IN.x - A sequence***.

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

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

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

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

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

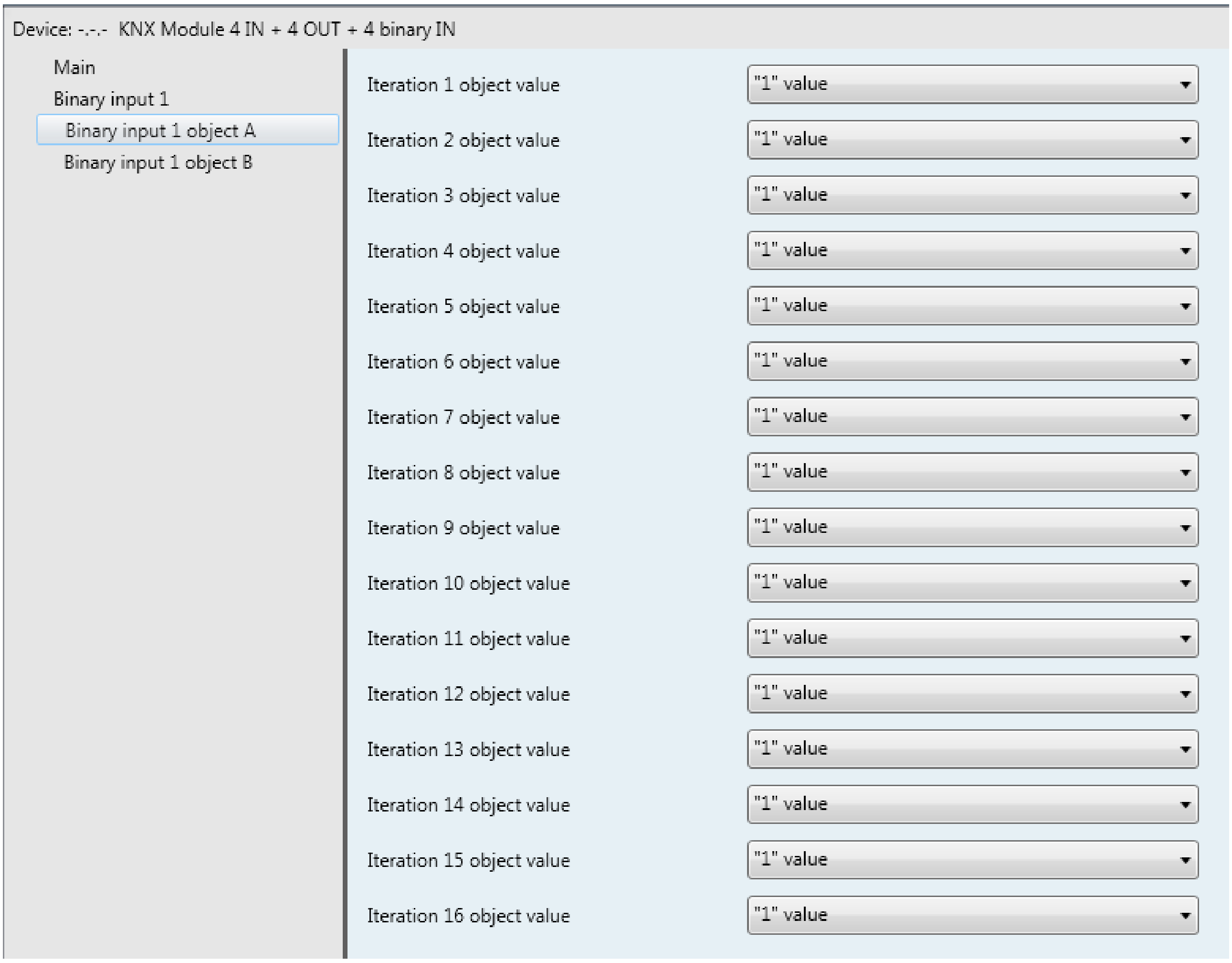


Fig. 4.11: “Binary input x object z” menu

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

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

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

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

"Sum" sequence

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

➢ *4.1.4.10 "Pulse counter" function*

Used to configure the channel for counting the number of contact status variations (edges) by setting the parameters that characterise the count.

The structure of the menu is as follows:

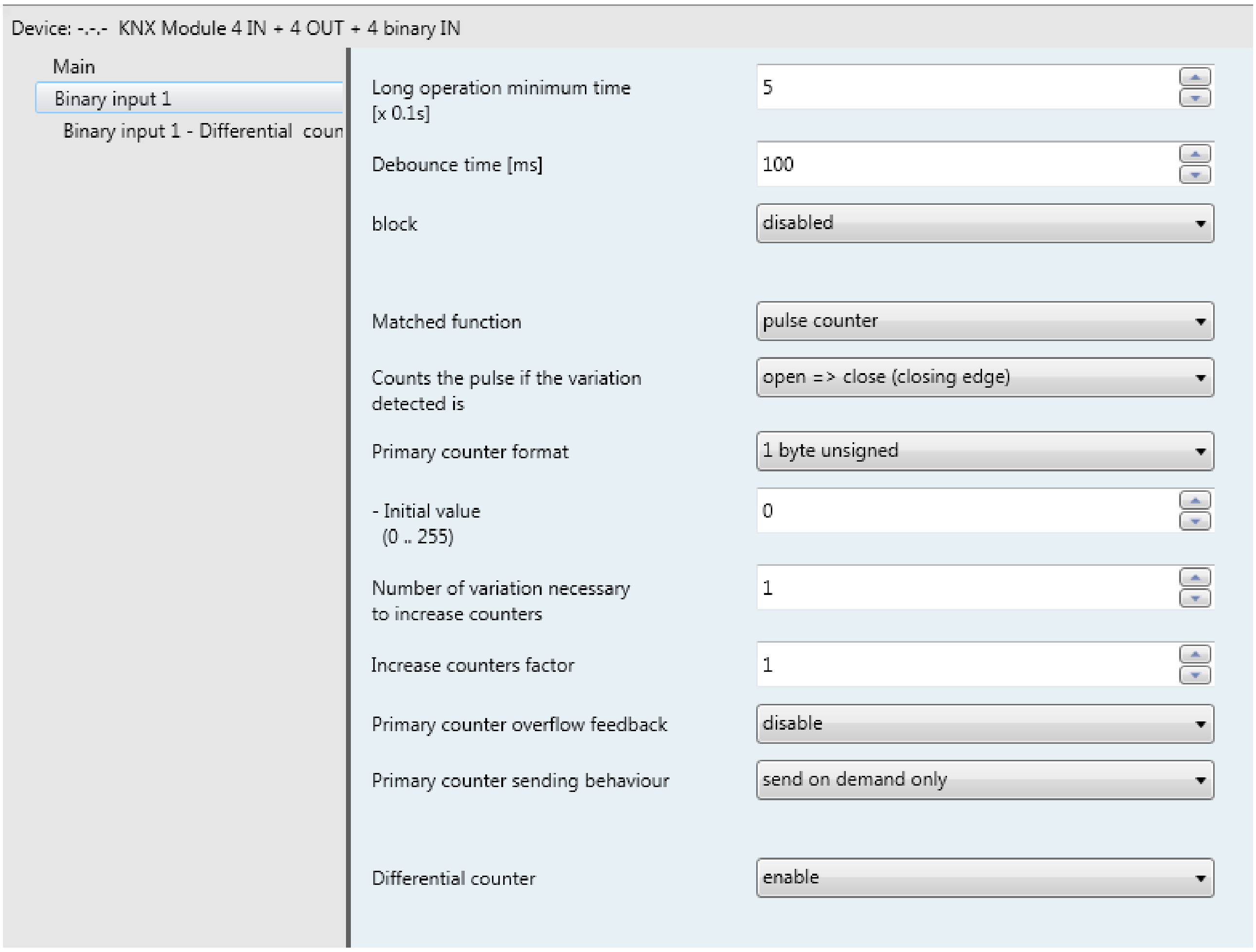


Fig. 4.12: "Pulse counter" matched function menu

Enabling this function makes the following parameters available for configuration:

➢ *4.1.4.10.1 Counts the pulse if the variation detected is*

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

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

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

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

➢ *4.1.4.10.2 Primary counter format*

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

- **1 byte value unsigned (default value)**
- 1 byte signed value
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value

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

- If the format of the primary counter is **1 byte unsigned**, the ***IN.x - Primary counter*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:
    - from **0 (default value)** to 255, with steps of 1
- If the format of the primary counter is **1 byte signed**, the ***IN.x - Primary counter*** communication object will be visible (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:
    - from -128 to 127 with steps of 1 **(default value 0)**
- If the format of the primary counter is **2 byte unsigned**, the ***IN.x - Primary counter*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:
    - from **0 (default value)** to 65535, with steps of 1
- If the format of the primary counter is **2 byte signed**, the ***IN.x - Primary counter*** communication object will be visible (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:
    - from -32768 to +32767 with steps of 1 **(default value 0)**
- If the format of the primary counter is **4 byte unsigned**, the ***IN.x - Primary counter*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

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

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

➢ *4.1.4.10.3 Number of variations necessary to increase counters*

The parameter **"Number of variations 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 with step of 1

➢ *4.1.4.10.4 Increase counters factor*

The parameter **"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 variations needed for a counter increase). This means that, if a value of 2 is set (for example), the counters (both primary and differential) will be increased by two units every time increase conditions occur.

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

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

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

➢ *4.1.4.10.5 Primary counter overflow feedback*

The parameter **"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** **(default value)**
- enable object of 1 bit
- enable objects of 1 bit and 1 byte

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

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

Once the maximum (or minimum) value has been reached, the primary counter restarts from the value set in **“Initial value”**.
If the value set in **“Increase counters factor”** is greater than 1, the number of units needed to trigger the overflow may be less than the increase factor; as the primary counter is circular, it is re-initialised when the overflow value is exceeded and the supplementary units are calculated.

Example: increase counters factor of *7*, the counter is *1 unsigned byte* and the initial value is 50. If the counter value is 253 and the counter increase condition is detected, the overflow telegram is sent and the new counter value is 54 (the initial value is also counted).

➢ *4.1.4.10.6 Primary counter sending behaviour*

This parameter **"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 on change and periodically

Selecting a value other than **send on demand only** displays the communication object ***IN.x - Primary counter sending trigger*** (Data Point Type: 1.017 DPT_Trigger). Selecting **send on change** or **send on change and periodically**, the **“Minimum primary counter variation for sending value”** parameter will be visible, whereas by selecting **send periodically** or **send on change and periodically** the **“Primary counter sending period (seconds)”** parameter will be visible.

Selecting the value **send on demand only**, no new parameter will be enabled because the primary counter value is not sent spontaneously by the device; only in the case of a status read request will it send the user a telegram in response to the command received, giving information about the current value of the primary counter.

If the primary counter sending condition is different from **on demand only**, there is the possibility of indirectly generating the sending of the current counter value following receipt of a BUS telegram on the object ***IN.x - Primary counter sending trigger*** (both with value “1” as well as with value “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.

The parameter **“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 with steps of 1, **10 (default value)**

The parameter **“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 with steps of 1, **15 (default value)**

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

➢ *4.1.4.10.7 Differential counter*

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

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

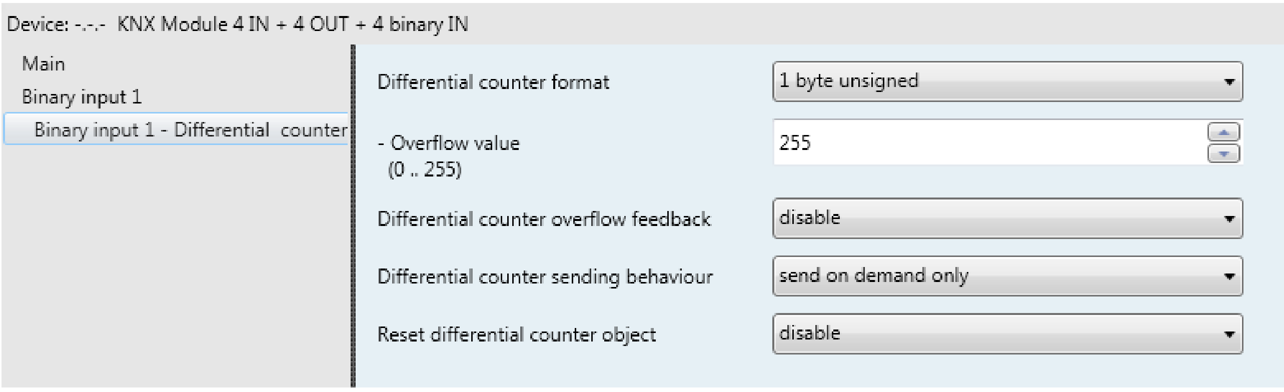


Fig. 4.13: "Binary input x – Differential counter" menu

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

- **1 byte value unsigned** **(default value)**
- 1 byte signed value
- 2 bytes unsigned value
- 2 bytes signed value
- 4 bytes unsigned value
- 4 bytes signed value

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

The **"Overflow value"** parameter is used to set the maximum value of the differential counter; in fact, unlike the primary counter, it is possible to set the maximum count value - i.e. the value beyond which the differential counter is in an overflow condition.

Depending on the value set for the **Differential counter format** parameter, the values that can be set for this item will be different.

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

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

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

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

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

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

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

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

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

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

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

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

The parameter **"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 object of 1 bit
- enable objects of 1 bit and 1 byte

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

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

Once the maximum value has been reached, the differential counter restarts from 0.
If the value set in **"Increase counters factor"** of the **Channel x** menu is greater than 1, the number of units needed to trigger the overflow may be less than the increase factor; as the differential counter is circular, it is re-initialised when the overflow value is exceeded and the supplementary units are calculated. Example: increase counters factor of *7* and the counter is *1 byte unsigned*. If the differential counter value is 253 and the counter increase condition is detected, the overflow telegram is sent and the new counter value is 4 (the initial value is also counted).

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

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

Selecting a value other than **send on demand only** displays the communication object ***IN.x - Differential counter sending trigger*** (Data Point Type: 1.017 DPT_Trigger). Selecting **send on change** or **send on change and periodically**, the **"Minimum differential counter variation for sending value"** parameter will be visible, whereas by selecting **send periodically** or **send on change and periodically** the **"Differential counter sending period (seconds)"** parameter will be visible.
Selecting the value **send on demand only**, no new parameter will be enabled because the differential counter value is not sent spontaneously by the device; only in the case of a status read request will it send the user a telegram in response to the command received, giving information about the current value of the differential counter.
If the differential counter sending condition is different from **on demand only**, there is the possibility of indirectly generating the sending of the current counter value following receipt of a BUS telegram on the object ***IN.x - Differential counter sending trigger*** (both with value "1" as well as with value "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.

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

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

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

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

The parameter **"Differential counter reset object"** is used to enable the display - and hence the use - of the communication object ***IN.x - Differential counter reset*** (Data Point Type: 1.017 DPT_Trigger), to receive - via BUS - the differential counter reset command for resetting the value of the differential counter. The values that can be set are:

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

Selecting **enable**, the ***IN.x - Differential counter reset*** 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 failure, the differential counter value must be saved in a non-volatile memory and restored when the BUS voltage is recovered.

➢ *4.1.4.11 “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 four commands per binary input.

The structure of the menu is as follows:

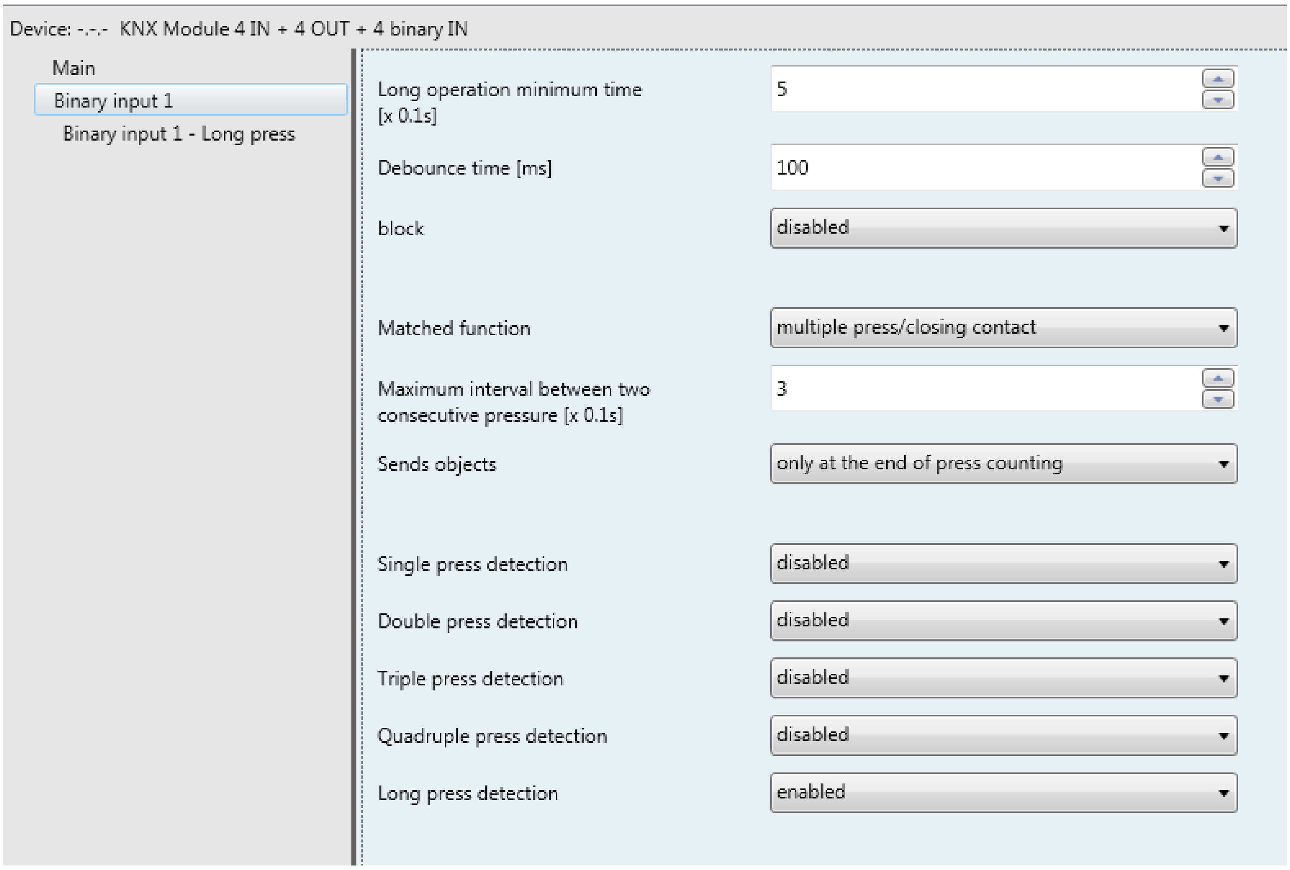


Fig. 4.14: "Multiple press/closing contact" function menu

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

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

Five consecutive presses or more are interpreted as a “quadruple press”.
In order to recognise two consecutive presses, it is necessary to define the maximum gap between the detection of one press and the next; if the time between two presses (not counting the debounce time) is less than the maximum gap, the count of multiple presses is increased. When the time that elapses after the detection of a pressing (not counting the debounce time) exceeds the maximum gap, the device recognises a number of consecutive multiple presses equal to the value counted and, after sending the telegrams associated with this action, it resets their counter.

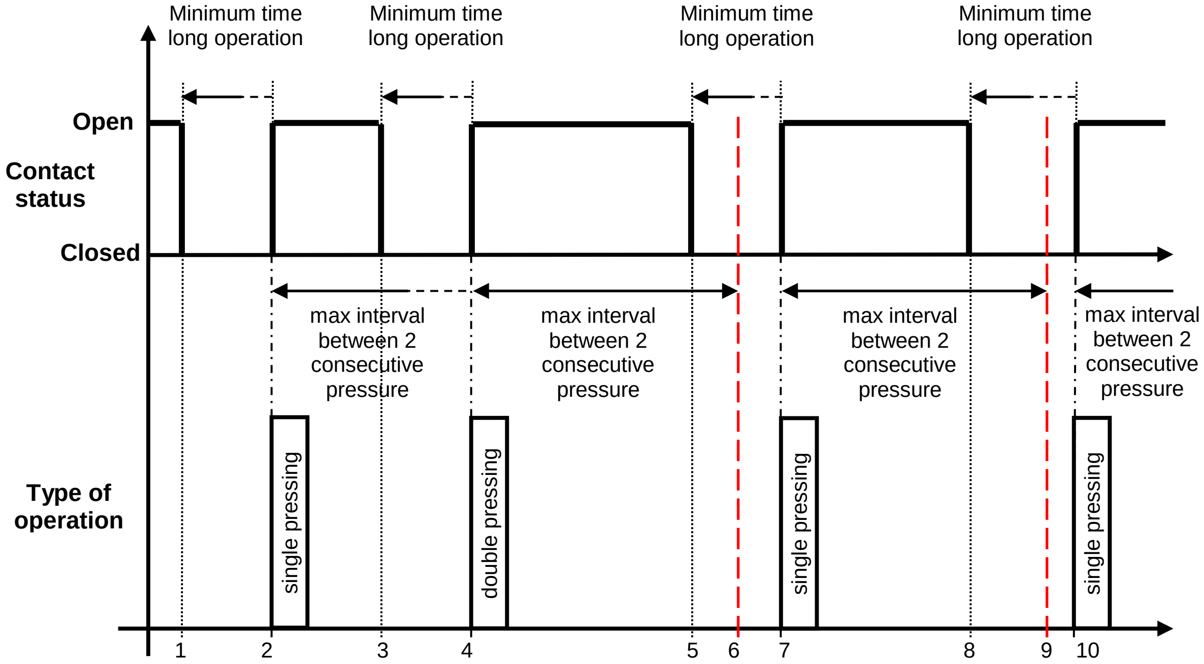
Enabling this function makes the following parameters available for configuration:

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

The parameter **"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 recognised as consecutive presses. The values that can be set are:

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

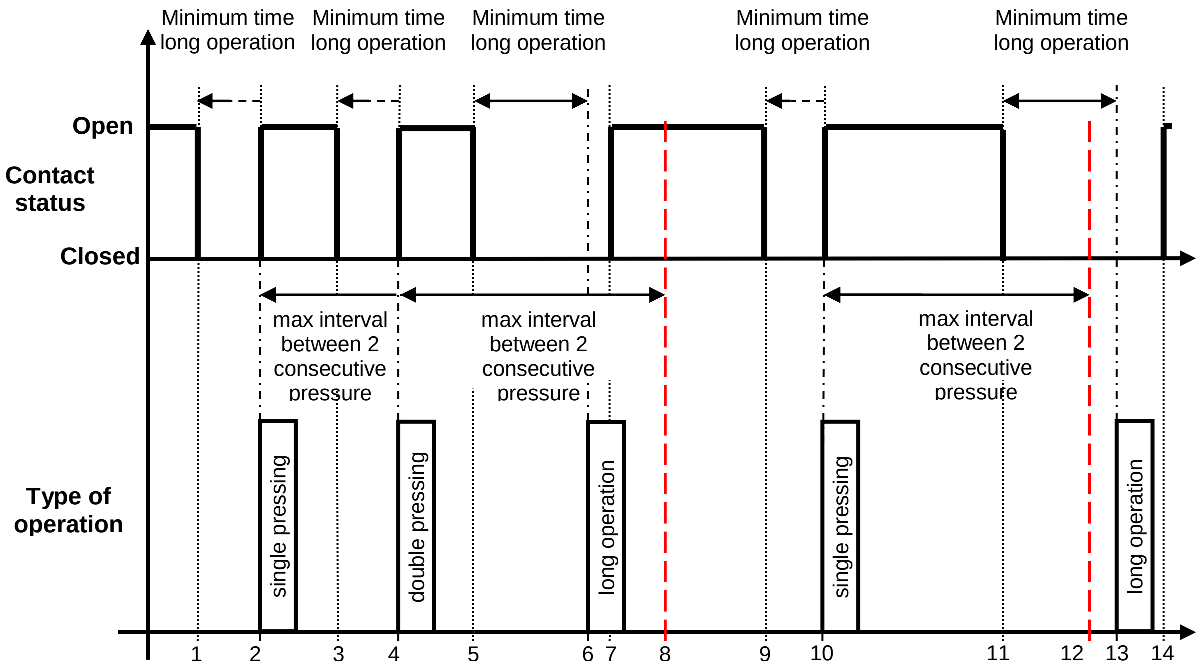
The following chart shows some situations that summarise the concept of multiple presses (the debounce time is not shown).



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

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

The detection of a long press in no way alters the multiple press count or any calculation of the gap between two consecutive presses, even if the minimum duration of the long operation is less than the maximum gap between two consecutive presses. See below (the debounce time is not shown).



1. Once the closure of the contact has been detected, the contact closure time is calculated in order to distinguish a short press from a long one.
2. When the re-opening of the contact is detected before the long operation time, a short press is recognised and the count of the gap between two consecutive presses is started. The multiple press count is increased.
3. A new contact closure leads to the initialisation of the contact closure time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
4. The re-opening of the contact before the long operation time and before reaching the maximum gap between two consecutive presses means the detection of a new short press that increases the multiple press count and re-initialises the calculation of the gap between two consecutive presses.
5. A new contact closure leads to the initialisation of the contact closure time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
6. If the contact remains closed for a time greater than the minimum duration of a long operation, a long press is recognised and the KNX commands for that action are sent, but neither calculation of the gap between two consecutive presses nor the multiple press count is modified in any way.
7. The re-opening of the contact following the recognition of a long press does not lead to any action.
8. Once the maximum gap between two consecutive presses (dotted red line) has elapsed, the multiple press count is terminated and, after sending the KNX commands relating to this action, the counter is reset.
9. A new contact closure leads to the initialisation of the contact closure time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.
10. The re-opening of the contact before the long operation time means the detection of a new short press that increases the multiple press count and initialises the count of the gap between two consecutive presses.
11. A new contact closure leads to the initialisation of the contact closure time count (to distinguish a short press from a long one), but this does not modify in any way the calculation of the gap between two consecutive presses and the multiple press count.

12. Once the maximum gap between two consecutive presses (dotted red line) has elapsed, the multiple press count is terminated and, after sending the KNX commands relating to this action, the counter is reset.
13. If the contact remains closed for a time greater than the minimum duration of a long operation, a long press is recognised and the KNX commands for that action are sent, but neither calculation of the gap between two consecutive presses nor the multiple press count is modified in any way.
14. The re-opening of the contact following the recognition of a long press does not lead to any action.

➢ *4.1.4.11.2 Send objects*

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

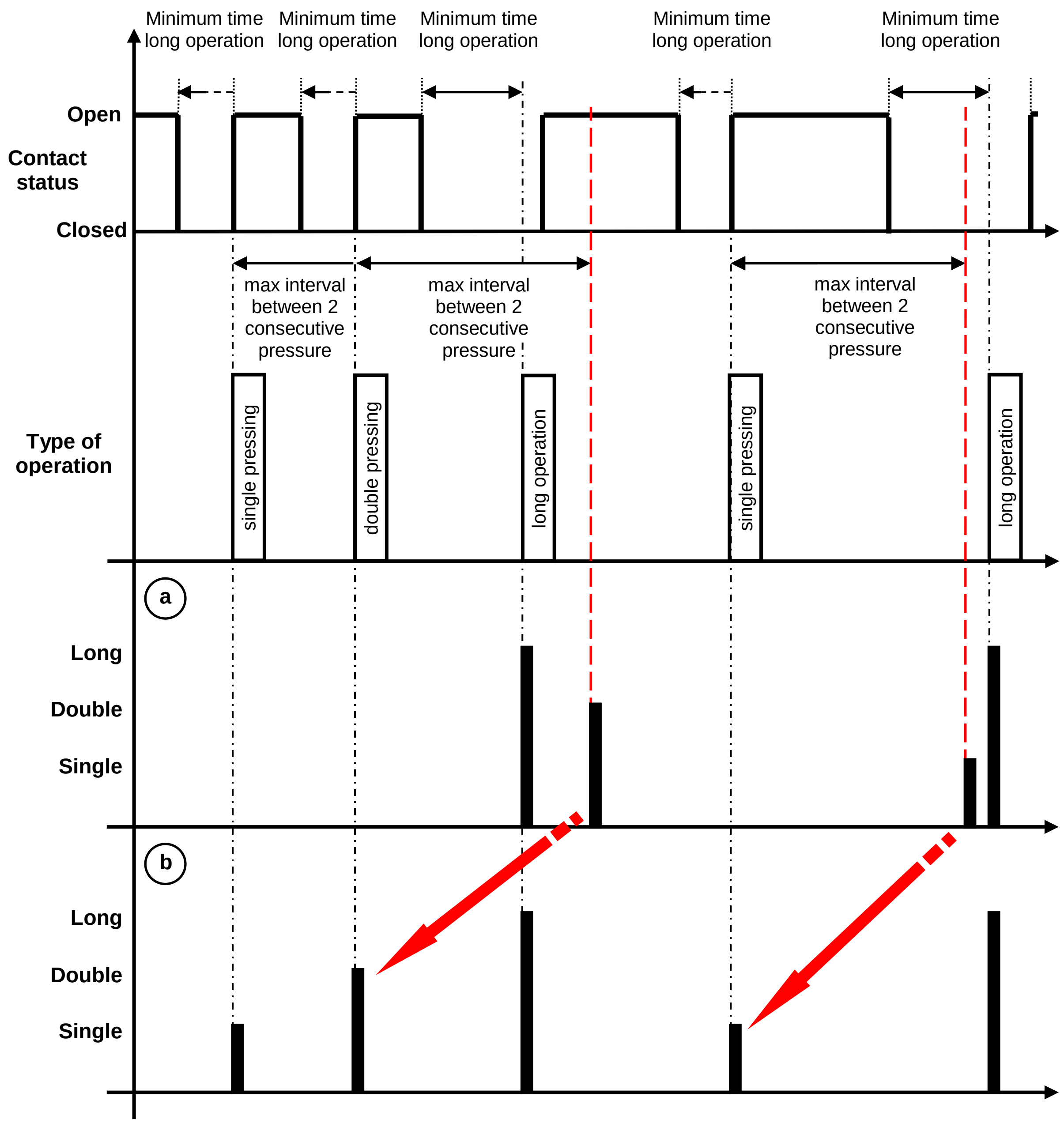
1. the device waits for the gap between two consecutive presses to exceed the maximum value, consequently interrupting the multiple press count and sending the commands associated with the number of presses detected
2. every time the multiple press count is increased, the device sends the telegrams associated with the number of presses detected

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

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

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

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



The chart resumes the situation shown previously, introducing the long press and its effect on counters and timers. The two sections at the bottom show the commands sent on the KNX BUS if the sending is **only at the end of the press count** (case “a”) or **with every press detected** (case “b”). The main difference between the two cases is that in case “b”, every time a multiple press is counted, the associated telegrams are sent, while in case “a” it is necessary to wait until the time between two consecutive presses exceeds the maximum value in order to end the multiple press count, and the telegrams sent are only those associated with the last press detected.
The red arrows highlight the differences between the moments when the telegrams associated with the same long presses are actually sent.

➢ *4.1.4.11.3 Single press detection*

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

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

selecting **enabled**, displays the menu **Binary input x - Single press** as shown in the following figure:

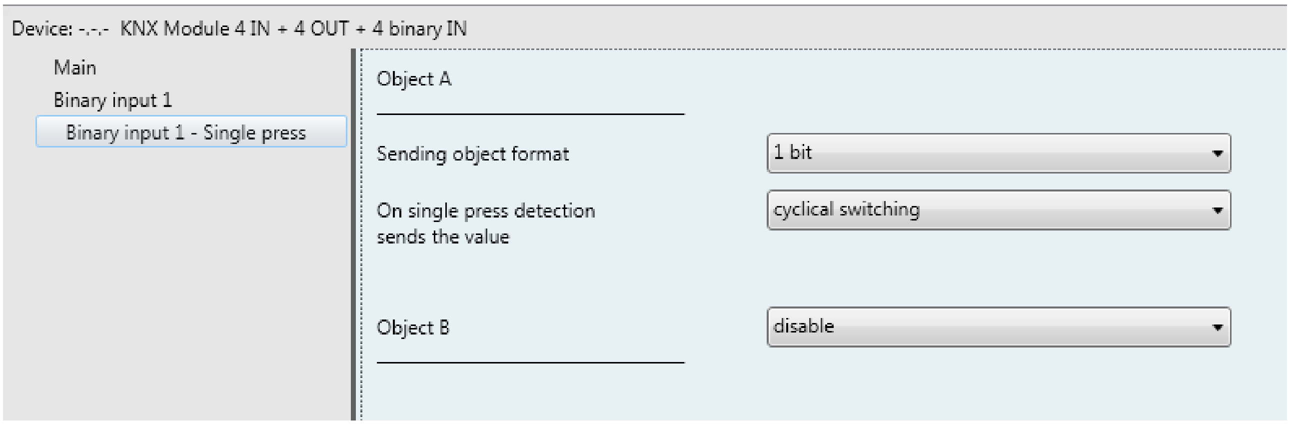


Fig. 4.15: "Binary input x – Single press" menu

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

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

selecting **enabled** displays the parameters **"Sending object format"** and **"On single press detection sends the value"** grouped in the subgroup **Object z** (where z indicates the object associated with the binary input, included between **A** and **D)**.

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

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

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

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

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

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

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

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

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

- If the format of the object to send is **1 byte value unsigned**, the ***IN.x - Single press 1 byte z object*** communication object will be visible (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

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

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

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

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

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

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

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

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

- If the format of the object to send is **2 byte value unsigned**, the ***IN.x - Single press 2 byte z object*** communication object will be visible (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

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

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

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

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

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

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

- If the format of the object to send is **4 byte value unsigned**, the ***IN.x - Single press 4 byte z object*** communication object will be visible (Data Point Type: 12.001 DPT_Value_4_Ucount) and the values that can be set for the above parameter are:

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

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

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

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

➢ *4.1.4.11.4 Double press detection*

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

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

selecting **enabled**, displays the menu **Binary input x - Double press** as shown in the following figure:

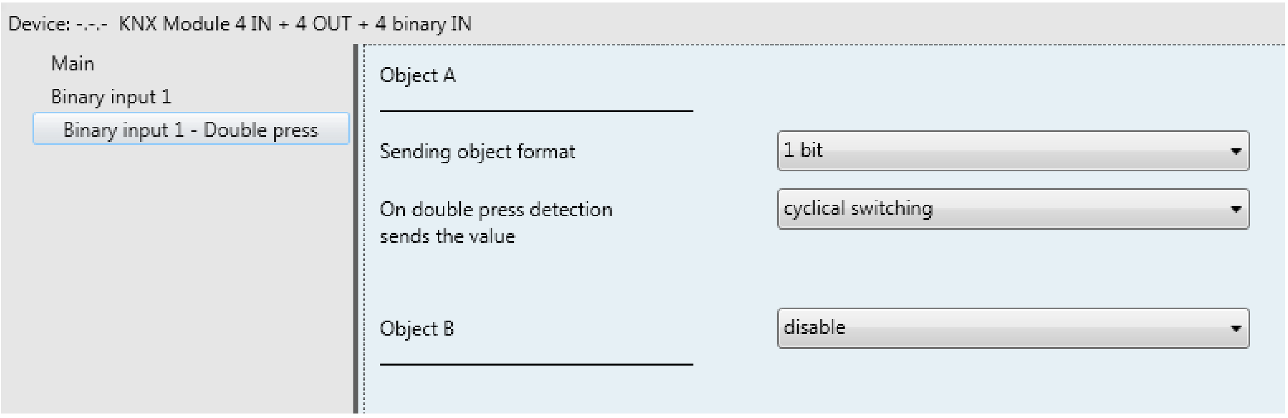


Fig. 4.16: "Binary input x – Double press" menu

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

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

selecting **enabled** displays the parameters **"Sending object format"** and **"On double press detection sends the value"** grouped in the subgroup **Object z** (where z indicates the object associated with the binary input, included between **A** and **D)**.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- cyclical switching (timed thermostat)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

➢ *4.1.4.11.5 Triple press detection*

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

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

selecting **enabled**, displays the menu **Binary input x - Triple press** as shown in the following figure:

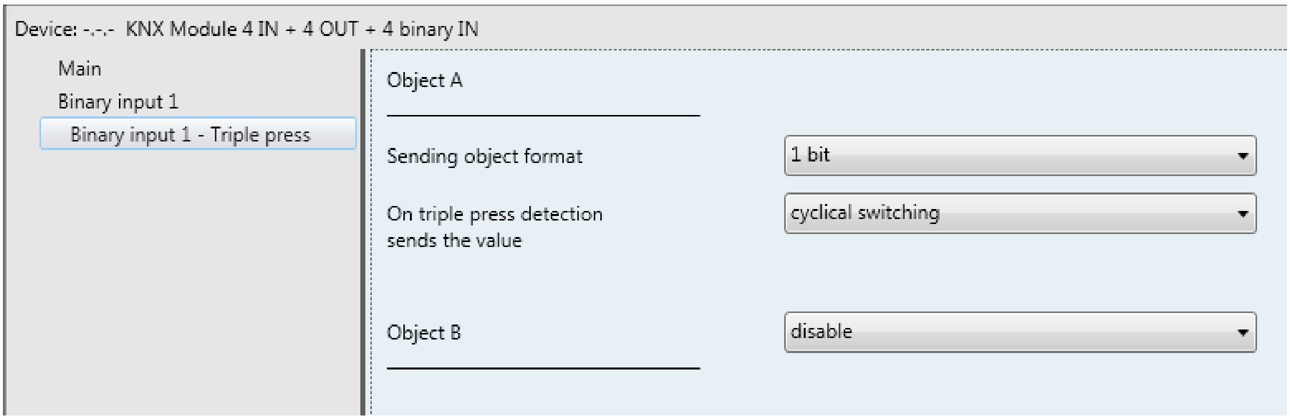


Fig. 4.17: "Binary input x – Triple press" menu

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

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

selecting **enabled** displays the parameters **"Sending object format"** and **"On triple press detection sends the value"** grouped in the subgroup **Object z** (where z indicates the object associated with the binary input, included between **A** and **D)**.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

  - auto mode
  - comfort mode

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

➢ *4.1.4.11.6 Quadruple press detection*

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

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

selecting **enabled**, displays the menu **Binary input x - Quadruple press** as shown in the following figure:

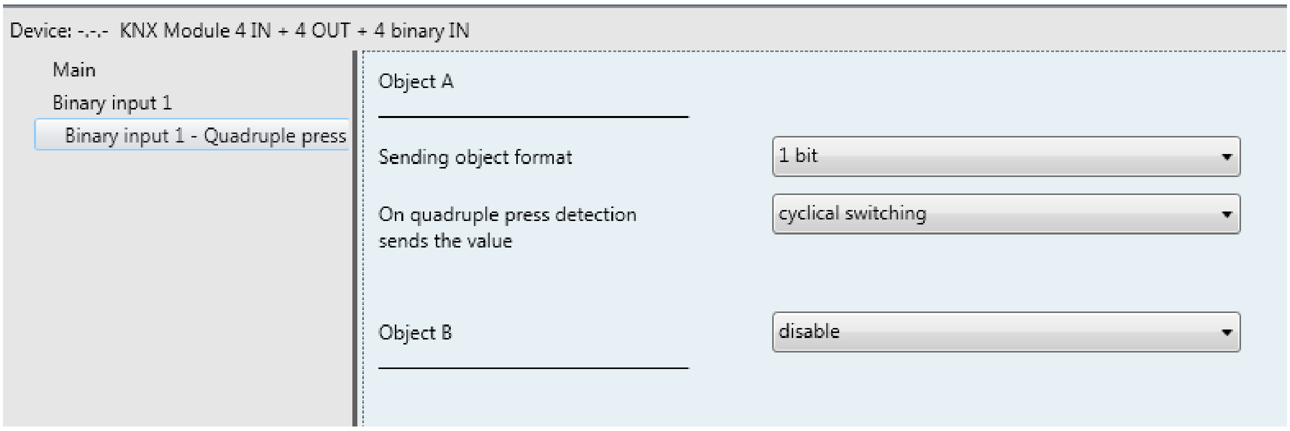


Fig. 4.18: “Binary input x – Quadruple press” menu

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

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

selecting **enabled** displays the parameters **“Sending object format”** and **“On quadruple press detection sends the value”** grouped in the subgroup **Object z** (where z indicates the object associated with the binary input, included between **A** and **D)**.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

➢ *4.1.4.11.7 Long press detection*

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

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

selecting **enabled**, displays the menu **Binary input x - Long press** as shown in the following figure:

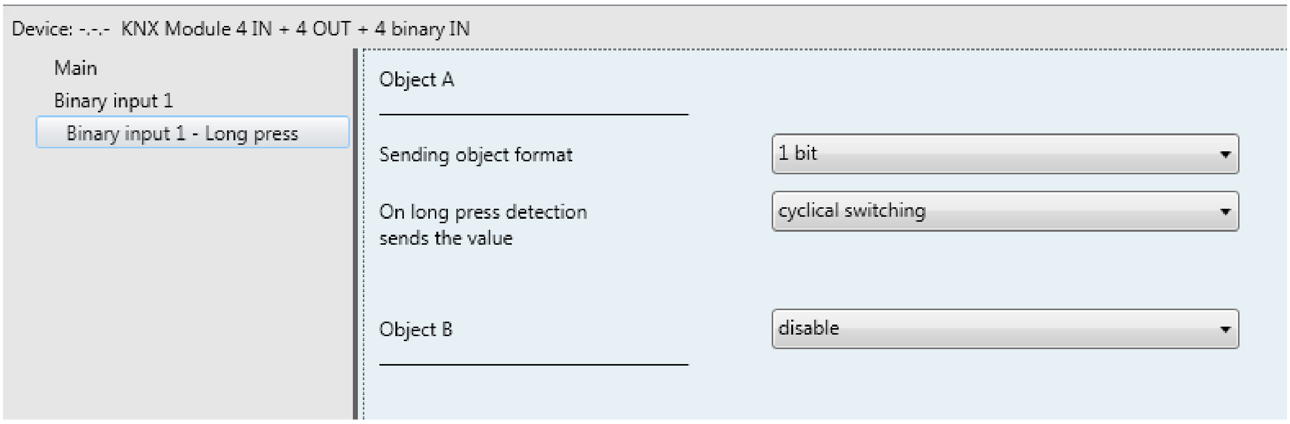


Fig. 4.19: “Binary input x – Long press” menu

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

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

selecting **enabled** displays the parameters **“Sending object format”** and **“On long press detection sends the value”** grouped in the subgroup **Object z** (where z indicates the object associated with the binary input, included between **A** and **D)**.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5 *“Temperature sensor X”* menu

If the operating mode for the universal input is “**Temperature sensor**”, a dedicated menu called **Temperature sensor x** is displayed for each input (x = 1 .. 4, the universal input index). All 4 menus are identical and therefore, for the sake of simplicity, the operation and dedicated parameters are summarised in this paragraph, indicating the reference menu with a generic “X” (1 .. 4).

The basic structure of the menu is as follows:

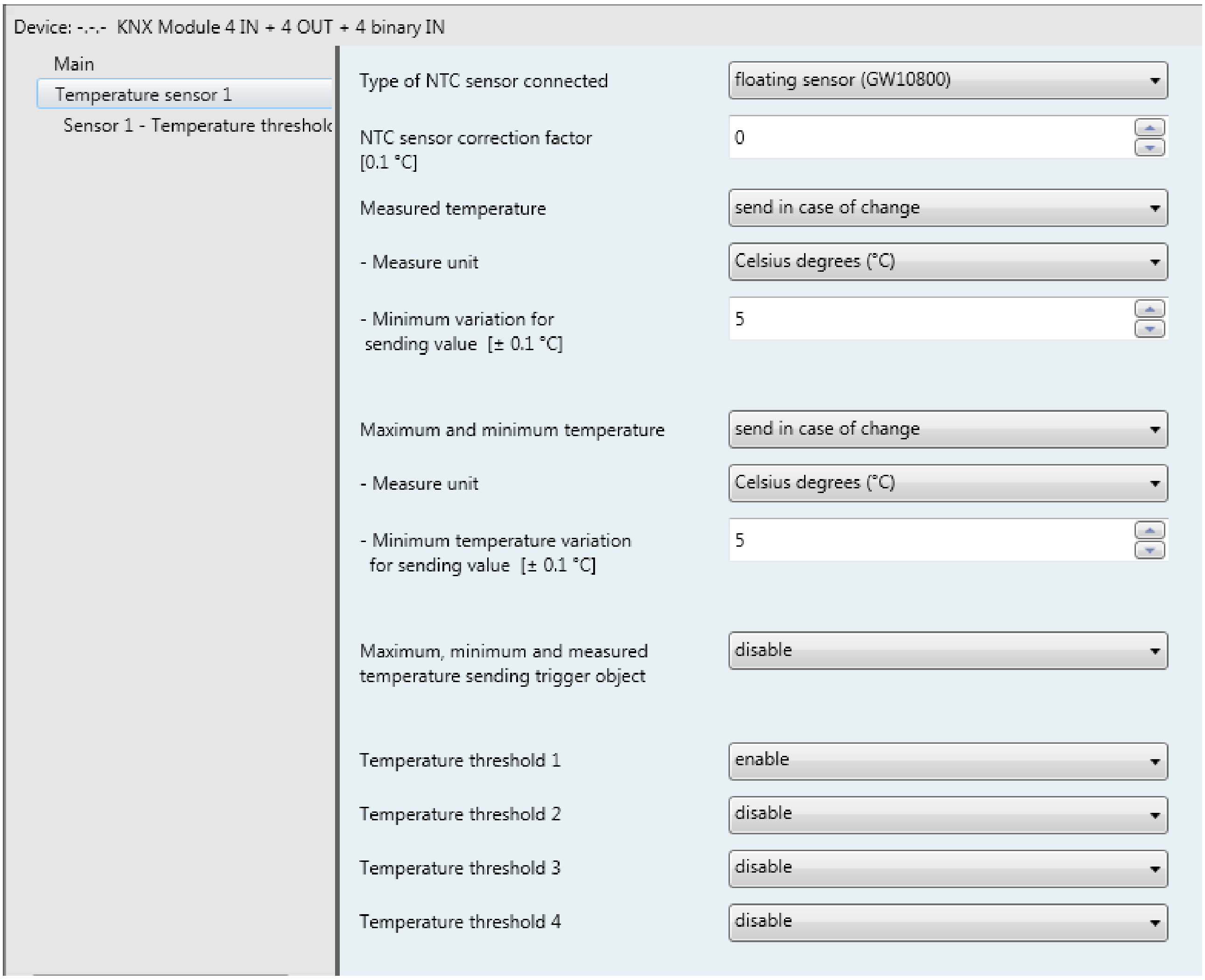


Fig. 5.1: “Temperature sensor X” menu

5.1 Parameters

➢ *5.1.1 Type of NTC sensor connected*

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

The values that can be set are:

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

5.1.2 NTC sensor correction factor [0.1 °C]

The parameter **“NTC sensor correction factor [0.1 °C]”** is used to set the correction factor to be applied to the temperature value of the NTC sensor connected to the input, to eliminate the heat contribution generated by the installation site; The values that can be set are:

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

5.1.3 Measured temperature

The parameter **“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 on change and periodically

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

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

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

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

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

The **“Measure unit”** parameter is used to set the measurement unit for coding and sending the information via the ***IN.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 ***IN.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)**.

➢ *5.1.4 Maximum and minimum temperature*

The parameter **“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
- **send in case of change (default value)**
- send periodically
- send on change and periodically

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

Selecting the value **send on demand only**, no new parameter will be enabled, as the 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 communication object ***IN.x - Maximum and minimum temperature reset*** (Data Point Type: 1.017 DPT_Trigger); when this object receives a BUS telegram with the logical value “0” or “1”, the device automatically resets the maximum and minimum temperature to the current measured temperature, cancelling the previously saved values.

The parameter **“Measure unit”** is used to set the measure unit for coding and sending the information via the communication objects ***IN.x - Maximum measured temperature*** and ***IN.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 ***IN.x - Maximum measured temperature*** and ***IN.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)**.

The parameter **“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 with steps of 1, **(default value 5)**

The parameter **“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 with steps of 1 **(default value 5)**

➢ *5.1.5 Maximum, minimum and measured temperature sending trigger object*

The parameter **“Maximum, minimum and measured temperature sending trigger object”** is used to enable the input object ***IN.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** **(default value)**
- enable

An NTC sensor malfunction or failed connection is signalled by the amber LED associated with the input (blinking at 8 Hz for 3 seconds and switched off for 3 seconds) and all the other input functions are interrupted.

➢ *5.1.6 Temperature threshold x*

For each universal input configured as a temperature sensor, it is possible to configure 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 parameters **“Temperature threshold 1”**, **“Temperature threshold 2”**, **“Temperature threshold 3”** and **“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 **Sensor x - Temperature threshold 1**, **Sensor x - Temperature threshold 2**, **Sensor x - Temperature threshold 3** and **Sensor x - Temperature threshold 4** depending on the reference parameter.

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

The structure of the menu is as follows:

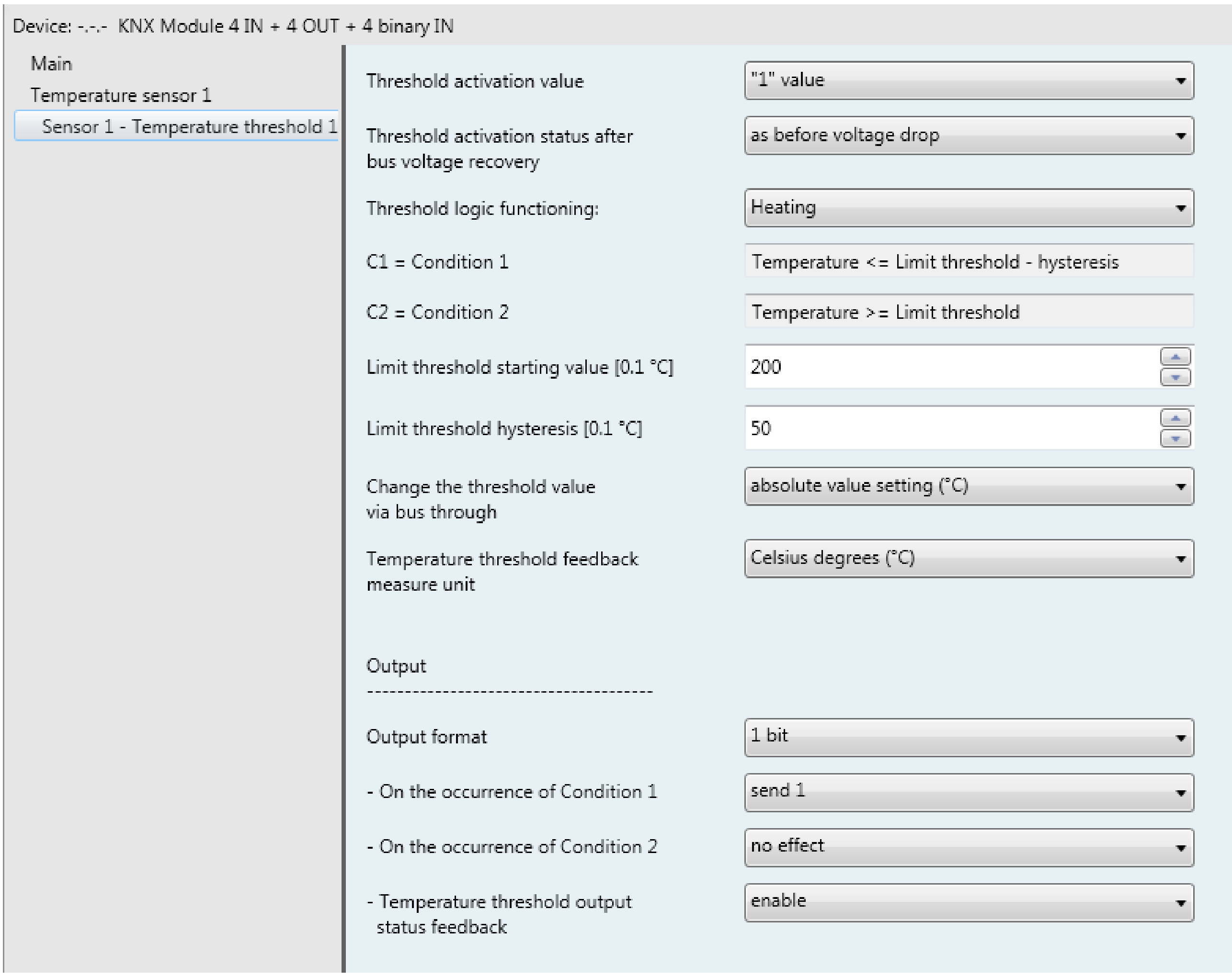


Fig. 5.2: “Sensor X - Temperature threshold y” menu

The meaning of the various parameters in this configuration menu are illustrated below.

➢ *5.1.6.1 Threshold activation value*

The parameter **“Threshold activation value”** determines which logic value received via the communication object ***IN.x - Temperature threshold activation y*** activates the temperature threshold y; Receiving the value opposite to the one set for activation will deactivate the threshold. The possible values are:

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

➢ *5.1.6.2 Threshold activation status after BUS voltage recovery*

The parameter **“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 possible values are:

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

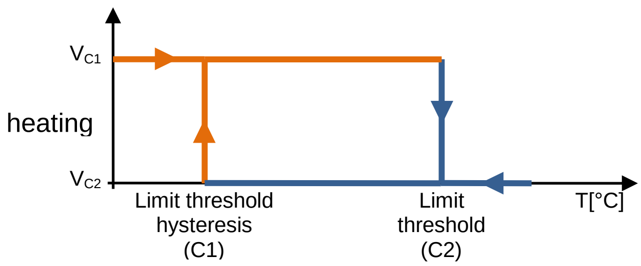
regardless of the value set to this parameter, after an application ETS download the threshold status is always enabled.

➢ *5.1.6.3 Threshold functioning logic*

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

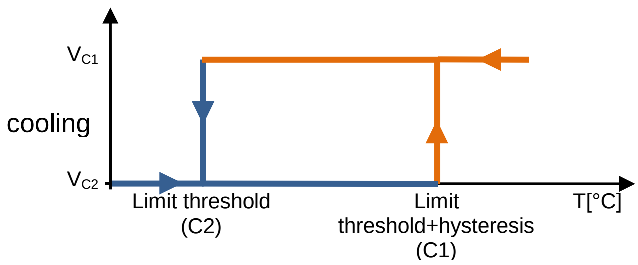
- **heating** **(default value)**
- cooling

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 associated with Condition 1; when the measured temperature reaches value C2 (limit threshold), the device will send the command associated 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 associated with Condition 1; when the measured temperature reaches value C2 (limit threshold), the device will send the command associated with Condition 2.

➢ *5.1.6.4 Limit threshold starting value [0.1 °C]*

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

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

➢ *5.1.6.5 Limit threshold hysteresis [0.1 °C]*

The parameter **“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 may assume the following values:

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

➢ *5.1.6.6 Change the threshold value via BUS through*

The **“Change the threshold value via BUS through”** parameter defines the communication object format needed to set the limit threshold via a BUS telegram. The values that can be set are:

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

selecting the value **absolute value setting** displays the communication object ***IN.x - Temperature threshold value input y*** (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) through which it is possible to set the value of the limit threshold via BUS.
Selecting **increase/decrease step setting** displays the parameter **“Threshold regulation step via bus [0.1 °C]”** and the object ***Temperature threshold regulation y*** (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 **“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 **“Threshold regulation step via bus [0.1 °C]”**.
The limit threshold values received via bus are not limited by the device, as is the case for **“Limit threshold starting value 0.1 °C]”** ETS parameter (range: 0 - 40 ° C).

The parameter **“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 with steps of 1, **5 (default value)**

➢ *5.1.6.7 Temperature threshold feedback measure unit*

The current value of the temperature threshold y is transmitted on the BUS via the object ***IN.x - Temperature threshold feedback y.*** The parameter **“Temperature threshold feedback measure unit”** is used to set the measurement unit for coding and sending the information via the communication object ***IN.x - Temperature threshold feedback y***; 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 ***IN.x - Temperature threshold feedback y***: *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.

➢ *5.1.6.8 Output format*

It is possible to send an object for each threshold depending on the occurrence of conditions 1 and 2; The parameter **“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** **(default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC
- 2 byte unsigned
- 2 byte signed
- 2 byte setpoint in °C
- 2 byte setpoint in °K
- 2 byte setpoint in °F

The value set for this item will cause the values set for the **“On the occurrence of condition 1”** and **“On the occurrence of condition 2”** parameters to change as a result.
The **“On the occurrence of condition 1”** parameter is used to set the command or value to be sent on occurrence of condition 1.
The **“On the occurrence of condition 2”** parameter is used to set the command or value to be sent on occurrence of condition 2.

- If the output format is **1 bit**, this displays the communication object ***IN.x - Temperature threshold output y*** (Data Point Type: 1.001 DPT_Switch) and the parameter **“Temperature threshold output status feedback”** whereas the values that can be set for the two parameters listed above are:

  - **no effect** **(default value on occurrence of cond 2)**
  - send 0
  - **send 1** **(default value on occurrence of cond 1)**

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

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

  selecting the value **enable** displays the communication object ***IN.x - Temperature threshold output status feedback y***; 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 **“Cyclical repetition of the output commands”**.

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

  - **no effect** **(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 ***IN.x - Temperature threshold output y*** (Data Point Type: 5.010 DPT_Value_1_Ucount), and the values that can be set for the two parameters listed above are:

  - **no effect** **(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 new displayed parameter **“Value (0 .. 255)”** which can assume the following values:

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

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

  - **no effect** **(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 **“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 ***IN.x - Temperature threshold output y*** (Data Point Type: 5.001 DPT_Scaling) and the values that can be set for the two parameters listed above are:

  - **no effect** **(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 **“Value (0% .. 100%)”** which can assume the following values:

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

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

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

  Selecting **send HVAC mode of reference ± offset** displays the **“Offset (-3 .. +3)”** and the communication object ***IN.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 ***IN.x - Temperature threshold output reference y*** 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 **“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 ***IN.x - Temperature threshold output y***; The possible values are:

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

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

  - **no effect** **(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 **"Value (0 .. 65535)"** which can assume the following values:

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

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

  - **no effect** **(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 **"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 byte setpoint in °C**, this displays the communication object ***IN.x - Temperature threshold output y*** (Data Point Type: 9.001 DPT_Value_Temp) and the values that can be set for the two parameters listed above are:

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

  selecting **send reference setpoint ± offset**, displays the parameter **"Offset [0.1 °C] (-300 .. +300)"** and the communication object ***IN.x - Temperature threshold output reference y*** (Data Point Type: 9.001 DPT_Value_Temp). in this case, the output value will be the setpoint received via the object ***IN.x - Temperature threshold output reference y*** 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 **"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 ***IN.x - Temperature threshold output y***; The possible values are:

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

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

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

  selecting **send reference setpoint ± offset**, displays the parameter **"Offset [0.1 °C] (-300 .. +300)"** and the communication object ***IN.x - Temperature threshold output reference y*** (Data Point Type: 9.002 DPT_Value_Tempd). in this case, the output value will be the setpoint received via the object ***IN.x - Temperature threshold output reference y*** 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 **"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 ***IN.x - Temperature threshold output y***; The possible values are:

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

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

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

selecting **send reference setpoint ± offset**, displays the parameter **"Offset [0.1 °C] (-300 .. +300)"** and the communication object ***IN.x - Temperature threshold output reference y*** (Data Point Type: 9.027 DPT_Value_Temp_F). in this case, the output value will be the setpoint received via the object ***IN.x - Temperature threshold output reference y*** 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 **"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 ***IN.x - Temperature threshold output y***; The possible values are:

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

The object ***IN.x - Temperature threshold output y*** 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 the telegrams associated with conditions C1 and C2 is inhibited, whereas any change or feedback of the threshold value is still carried out.

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

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

Selecting **enable** displays the **"Command repetition period"** parameter 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)**

6 *“Analog input X”* menu

If the operating mode for the universal input/output is “**Analog input**”, a dedicated menu called **Analog input x** is displayed for each input (x = 1 .. 4, the universal input index). All 4 menus are identical and therefore, for the sake of simplicity, the operation and dedicated parameters are summarised in this paragraph, indicating the reference menu with a generic “X” (1 .. 4).

The basic structure of the menu is as follows:

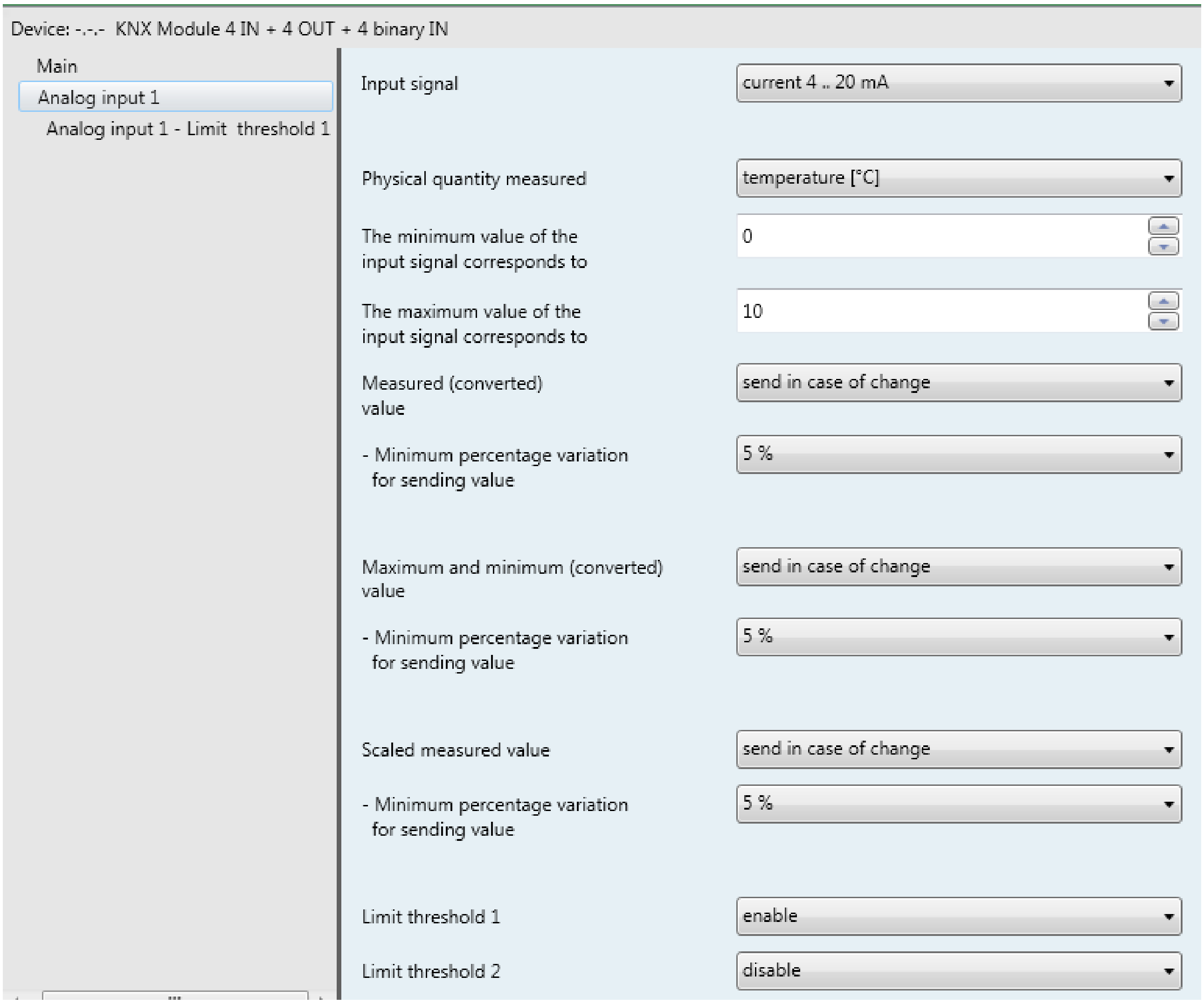


Fig. 6.1: “Analog input X” menu

6.1 Parameters

➢ 6.1.1 Input signal

The parameter **"Input signal"** is used to define the type of signal the analog input must process so the correct data processing circuit is used. The parameter may assume the following values:

- current 0 .. 20 mA (only for inputs 1 and 2)
- **current 4 .. 20 mA (only for inputs 1 and 2, default value)**
- **voltage 0 .. 10 V (only for inputs 3 and 4, default value)**
- voltage 0 .. 1 V (only for inputs 3 and 4)

If the input signal lies outside the selected range, alarm feedback is sent with value "1" via the communication object ***IN.x - Alarm signal out of scale*** (Data Point Type 1.005 DPT_Alarm); when the signal returns within the set range, the end of alarm feedback is sent with value "0". The telegrams are sent following a BUS request, spontaneously upon each alarm status variation and when BUS voltage is restored. The limits beyond which the out of scale alarm is sent are:

- **voltage 0 .. 10 V →** alarm is sent from an input voltage of approximately 10.3 V
- **voltage 0 .. 1 V →** alarm is sent from an input voltage of approximately 1.01 V
- **current 0 .. 20 mA →** alarm is sent with an input current of approximately 21 mA
- **current 4 .. 20 mA →** alarm is sent with an input current of approximately 21 mA or 3 mA

➢ 6.1.2 Physical quantity measured, minimum and maximum value of the signal

The parameter **"Physical quantity measured"** is used to set the physical quantity to be measured and as a result the format and coding of the communication object ***IN.x - Measured (converted) value*** that will be sent by the device to signal the measured (converted) value. The values that can be set are:

- **temperature [°C] (default value)**
  with this setting, the object format is *9.001 DPT_Value_Temp*
- temperature [°K]
  with this setting, the object format is *9.002 DPT_Value_Tempd*
- temperature [°F]
  with this setting, the object format is *9.027 DPT_Value_Temp_F*
- brightness [Lux]
  with this setting, the object format is *9.004 DPT_Value_Lux*
- wind speed [m/s]
  con with this setting, the object format is *9.005 DPT_Value_Wsp*
- pressure [Pa]
  with this setting, the object format is *9.006 DPT_Value_Pres*
- humidity [%]
  with this setting, the object format is *9.007 DPT_Value_Humidity*
- air quality [ppm]
  with this setting, the object format is *9.008 DPT_Value_AirQuality*
- time [s]
  with this setting, the object format is *9.010 DPT_Value_Time1*
- time [ms]
  with this setting, the object format is *9.011 DPT_Value_Time2*
- voltage [mV]
  with this setting, the object format is *9.020 DPT_Value_Volt*
- current [mA]
  with this setting, the object format is *9.021 DPT_Value_Current*
- power density [W/m^2]
  with this setting, the object format is *9.022 DPT_PowerDensity*
- power [kW]
  with this setting, the object format is *9.024 DPT_Power*
- amount of rain [l/m^2]
  with this setting, the object format is *9.026DPT_Rain_Amount*
- wind speed [km/h]

with this setting, the object format is *9.028 DPT_Value_Wsp_kmh*

The input voltage or current values correspond to the measurement of a physical quantity that is transformed into voltage or current via a transducer; to obtain the correct value of the physical quantity to be measured it is necessary to perform a conversion between input current/voltage measurement and the corresponding physical quantity value. To do this, it is necessary to define the range of the physical quantity values, defining the minimum value (that corresponds to the minimum input voltage/current value) and the maximum value (that corresponds to the maximum input voltage/current value).
The parameter **"The minimum value of the input signal corresponds to"** is used to set the minimum measurable value of the physical quantity that corresponds to the minimum input signal value; the unit of measure depends on the value of the **"Physical quantity measured"** parameter.

The values that can be set are:

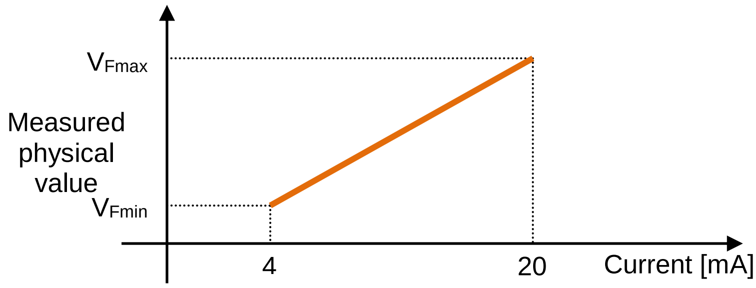
- from -670760 to 670760 with steps of 1 **(default value 0)**

The parameter **"The maximum value of the input signal corresponds to"** is used to set the maximum measurable value of the physical quantity that corresponds to the maximum input signal value; the unit of measure depends on the value of the **"Physical quantity measured"** parameter. The values that can be set are:

- from -670760 to 670760 with steps of 1 **(default value 10)**

If the minimum value of the signal is greater than the maximum value, no value is output.

The following example shows the meaning of the two above parameters:



The input signal is "current 4 .. 20 mA", "**V_{Fmin}**" indicates the value set for the parameter **"The minimum value of the input signal corresponds to"** whereas "**V_{Fmax}**" indicates the value set for the parameter **"The maximum value of the input signal corresponds to"**.
If the input signal is "4 mA", this means that the measurement of the real physical quantity is equal to "**V_{Fmin}**"; similarly, if the input signal is "20 mA", this means that the measurement of the real physical quantity is equal to "**V_{Fmax}**". After defining these two points, the conversion curve is equal to a straight line that joins the two fixed points and as a result all the values included within the range of the input signal will be converted using this curve.

➢ *6.1.3 Measured (converted) value*

The parameter **"Measured (converted) value"**, is used to define the conditions for sending the value measured (converted) by the device via the communication object ***IN.x - Measured (converted) value***; The values that can be set are:

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

Selecting the value **send in case of change** or **send on change and periodically**, displays the parameter **“Minimum percentage variation for sending value”** whereas selecting the value **send periodically** or **send on change and periodically** displays the parameter **“Value sending period [minutes]”**.
Selecting the value **send on demand only**, no new parameter will be enabled because the measured 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 measured value.
The parameter **“Minimum percentage variation for sending value”**, which is visible if the measured (converted) value is sent on change, is used to define the minimum percentage variation of the measured value (calculated within the range from the minimum value to the maximum value of the signal), in relation to the last value sent, that causes the new value to be spontaneously sent; The values that can be set are:

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

The parameter **“Value sending period [minutes]”**, which is visible if the (converted) value is sent periodically, is used to define the period with which the measured value feedback telegrams are sent spontaneously; The values that can be set are:

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

➢ *6.1.4 Maximum and minimum (converted) value*

The parameter **“Maximum and minimum (converted) value”**, is used to define the conditions for sending the maximum and minimum (converted) value measured by the device up to that moment sent via the objects ***IN.x - Maximum measured (converted) value*** and ***IN.x - Minimum measured (converted) value*** (the Data Point Type depends on the value set for the physical quantity measured); The values that can be set are:

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

Selecting the value **send in case of change** or **send on change and periodically**, displays the parameter **“Minimum percentage variation for sending value”** whereas selecting the value **send periodically** or **send on change and periodically** displays the parameter **“Maximum and minimum (converted) value sending period [minutes]”**.
Selecting the value **send on demand only**, no new parameter will be enabled because the maximum and minimum (converted) 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 maximum and minimum measured value.
The maximum and minimum measured value can be reset via the communication object ***IN.x - Maximum and minimum (converted) value reset*** (Data Point Type: 1.017 DPT_Trigger); when this object receives a BUS telegram with the logical value “0” or “1”, the device automatically resets the maximum and minimum measured values to the current measured value, cancelling the previously stored values.

The parameter **“Minimum percentage variation for sending value”**, which is visible if the maximum and minimum measured value is sent on change, is used to define the minimum variation of the value, in relation to the last value sent (calculated within the range from the minimum value to the maximum value of the signal), that causes the new measured value to be spontaneously sent; The values that can be set are:

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

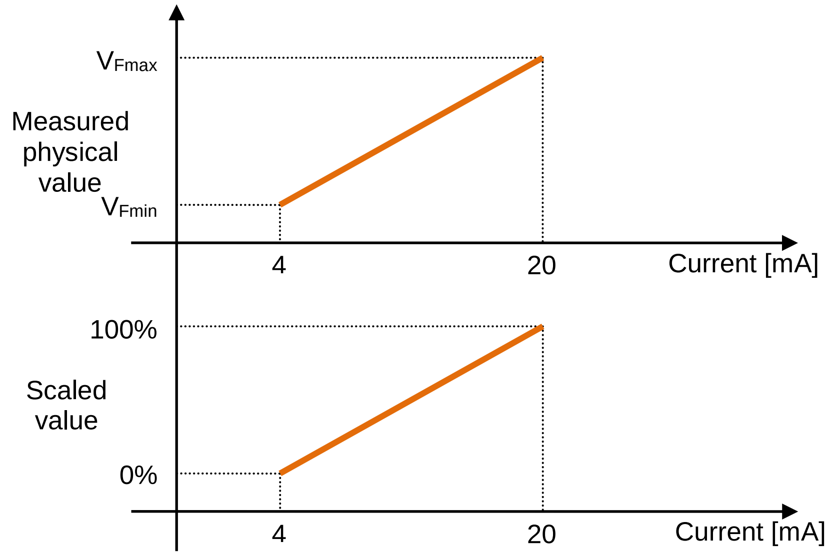
The parameter **“Maximum and minimum (converted) value sending period [minutes]”**, which is visible if the maximum and minimum measured values are send periodically, is used to define the period with which the maximum and minimum measured value feedback telegrams are sent spontaneously; The values that can be set are:

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

➢ *6.1.5 Scaled measured value*

In addition to the measured value, also the communication object ***IN.x - Scaled measured value (%)*** is available (Data Point Type: 5.001 DPT_Scaling) which is used to signal the scaled measured value (%) with respect to the maximum signal value; if the input signal has the minimum value, then the Scaled measured value will be “0%” whereas if the signal has the minimum value, the scaled measured value will be “100%”

Referring to the previous example, you can see the correlation between the measured value and the scaled measured value:



The parameter **“Scaled measured value”**, is used to define the conditions for sending the scaled measured value (%) via the communication object ***IN.x - Scaled measured value***; The values that can be set are:

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

Selecting the value **send in case of change** or **send on change and periodically**, displays the parameter **“Minimum variation for sending value ”** whereas selecting the value **send periodically** or **send on change and periodically** displays the parameter **“Value sending period [minutes]”.**
Selecting the value **send on demand only**, no new parameter will be enabled because the scaled measured 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 scaled measured value.

The parameter **Minimum variation for sending value** , which is visible if the scaled measured value is sent on variation, is used to define the minimum scaled measured value variation (in relation to the last value sent) that causes the new value to be spontaneously sent; The values that can be set are:

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

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

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

6.1.6 Limit threshold 1,2,3,4

The parameters **“Limit threshold 1”**, **“Limit threshold 2”**, **“Limit threshold 3”** and **“Limit threshold 4”** are used to activate the relative analog input thresholds, displaying the various configuration menus. The values that can be set are:

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

Selecting the value **enable** displays the configuration menus **Analog input x - Limit threshold 1**, **Analog input x - Limit threshold 2**, **Analog input x - Limit threshold 3** and **Analog input x - Limit threshold 4** depending on the parameter of reference.

6.1.6.1 Analog input x - Limit threshold y

For each universal input configured as an analogue input, it is possible to configure up to 4 limit 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 **Analog input x** menu.

The structure of the menu is as follows:

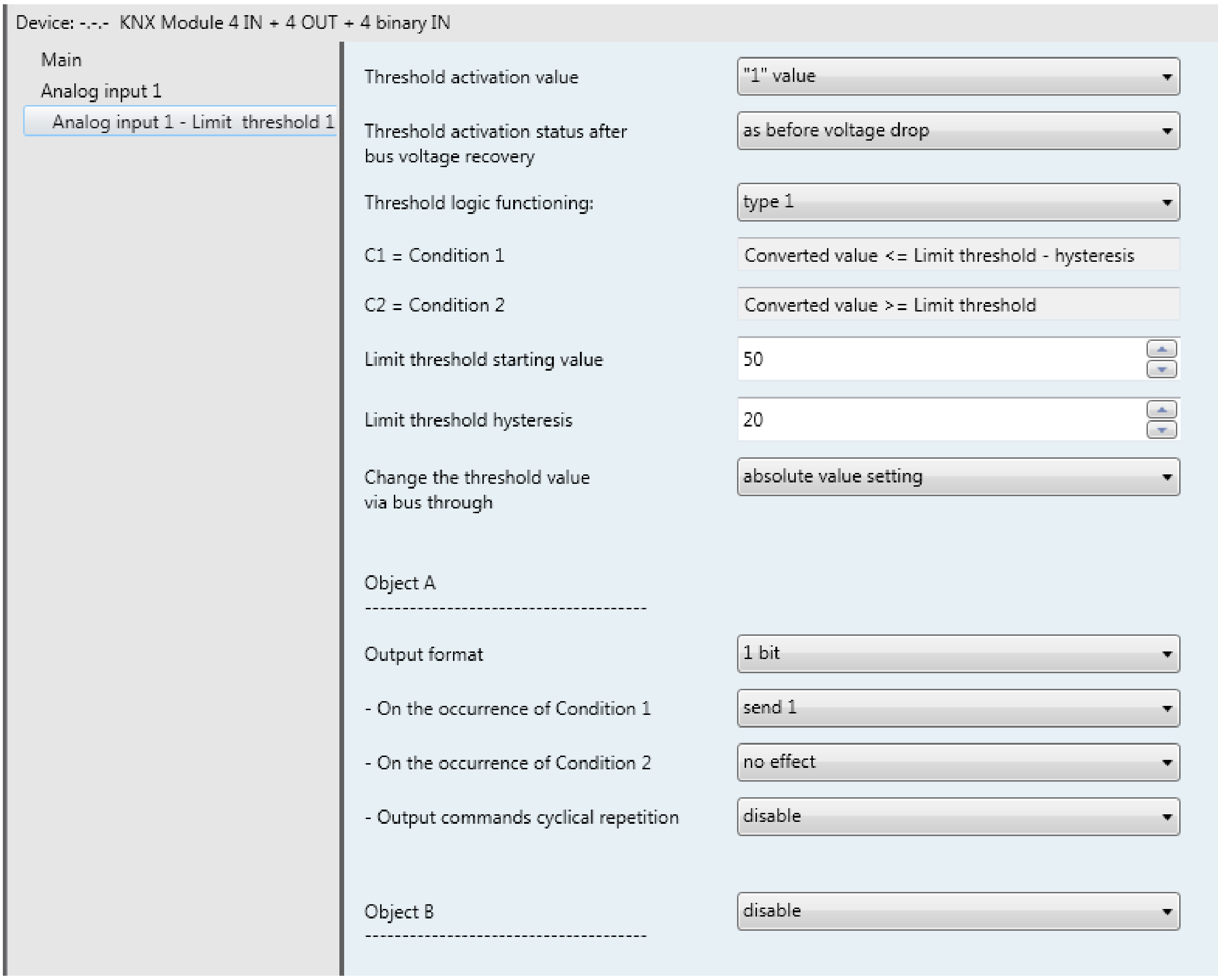


Fig. 6.2: "Analog input x - Limit threshold y" menu.

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

<u>This menu can be used to configure the following parameters:</u>

The parameter **"Threshold activation value"** determines which logic value received via the communication object ***IN.x - Limit threshold activation y*** activates the limit threshold y; Receiving the value opposite to the one set for activation will deactivate the threshold.

the possible values are:

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

The parameter **"Threshold activation status after bus voltage recovery"** is used to set the status of the limit threshold y after the BUS power supply voltage is restored; The possible values are:

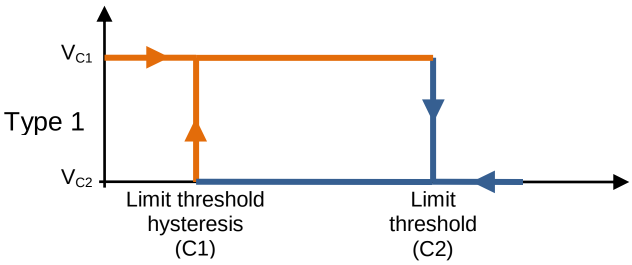
- disabled
- enabled
- **as before voltage drop** **(default value)**

regardless of the value set to this parameter, after an application ETS download the threshold status is always enabled.

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

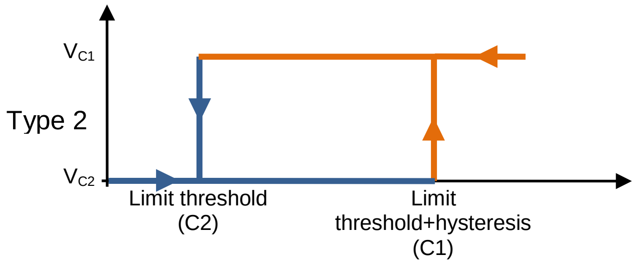
- **type 1** **(default value)**
- Type 2

Selecting the value **type 1**, the two conditions will be defined as follows:
Condition 1 = Converted value ≤ Limit threshold – hysteresis
Condition 2 = Converted value ≥ Limit threshold



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

Selecting the value **type 2**, the two conditions will be defined as follows:
Condition 1 = Converted value ≥ Limit threshold + hysteresis
Condition 2 = Converted value ≤ Limit threshold



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

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

- from -670760 to 670760 with steps of 1 **(default value 50)**

The parameter **"Limit threshold hysteresis"** 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 may assume the following values:

- from 10 to 1000 with steps of 1 **(default value 20)**

The **"Change the threshold value via BUS through"** parameter defines the communication object format needed to set the limit threshold via a BUS telegram. The values that can be set are:

- **absolute value setting** **(default value)**
- increase/decrease step regulation

selecting **absolute value setting** displays the communication object ***IN.x - Limit threshold value input y*** (the Data Point Type depends on the value set for the physical quantity measured) through which it is possible to set the value of the limit threshold via BUS.
Selecting **increase/decrease step setting** displays the parameter **"Threshold regulation step via bus"** and the object ***IN.x - Limit threshold regulation y*** (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 **"Threshold regulation step via bus"**; If the value "0" is received on this object, the limit threshold value will be decreased by the value defined by the parameter **"Threshold regulation step via bus"**.

The parameter **"Threshold regulation step via bus"** 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 2000 with steps of 1 **(default value 5)**

The current value of the limit threshold y is transmitted on the BUS via the object ***IN.x - Limit threshold feedback y*** (the Data Point Type depends on the value set for the physical quantity measured); The feedback sending conditions are, following a BUS request, spontaneously at each threshold change and at BUS voltage recovery.

For each threshold, up to 2 different objects can be sent (which are distinguished by the letters A and B) when the limit threshold is exceeded; object A is always enabled, whereas the **"Object B"** parameter can be used to enable a new object to be sent. The parameter may assume the following values:

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

selecting **enable** displays the object ***IN.x - Output B limit threshold y*** and the parameters **"Output format"**, **"On the occurrence of condition 1"** and **"On the occurrence of condition 2"** grouped in the subgroup **Object z** (z is the index of the object associated with the channel, between **A** and **B)**.

The parameter **"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 (default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC
- 2 byte unsigned
- 2 byte signed

The value set for this item will cause the values set for the parameters **"On the occurrence of condition 1"** and **"On the occurrence of condition 2"** to change as a result.
The parameter **"On the occurrence of condition 1"** is used to set the command or value to be sent on occurrence of condition 1.
The parameter **"On the occurrence of condition 2"** is used to set the command or value to be sent on occurrence of condition 2.

- If the output format is **1 bit**, this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 1.001 DPT_Switch) and the parameter **"Limit threshold output status feedback"** whereas the values that can be set for the two parameters listed above are:

    - **no effect (default value on occurrence of cond 2)**
    - send 0
    - **send 1 (default value on occurrence of cond 1)**

- If the output format is **2 bit**, this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 2.001 DPT_Switch_Control) and the values that can be set for the two parameters listed above are:

    - **no effect (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 ***IN.x - Output z limit threshold y*** (Data Point Type: 5.010 DPT_Value_1_Ucount), and the values that can be set for the two parameters listed above are:

    - **no effect (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 **"Value (0 .. 255)"** which can assume the following values:

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

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

    - **no effect (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 **“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 ***IN.x - Output z limit threshold y*** (Data Point Type: 5.001 DPT_Scaling) and the values that can be set for the two parameters listed above are:

  - **no effect (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 **“Value (0% .. 100%)”** which can assume the following values:

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

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

  - **no effect (default value cond 2)**
  - send auto
  - **send comfort (default value cond 1)**
  - send precomfort
  - send economy
  - send off (building protection)

- If the output format is **2 byte unsigned** , this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the two parameters listed above are:

  - **no effect (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 **“Value (0 .. 65535)”** which can assume the following values:

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

- If the output format is **2 byte signed** , this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the two parameters listed above are:

  - **no effect (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 **“Value (-32768 .. +32767)”** which can assume the following values:

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

The object ***IN.x - Output z limit threshold y*** 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 the telegrams associated with conditions C1 and C2 is inhibited, whereas any change or feedback of the threshold value is still carried out.

The **“Cyclical repetition of the output commands”** 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 **"Command repetition period"** 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)**

7 *“Digital input x with S0 interface”* menu

If the operating mode for the universal input is “Digital input with S0 interface”, a dedicated menu called **Digital input x with S0 interface** is displayed for each input (x = 1 .. 4, the universal input index). All 4 menus are identical and therefore, for the sake of simplicity, the operation and dedicated parameters are summarised in this paragraph, indicating the reference menu with a generic “X” (1 .. 4).

The basic structure of the menu is as follows:

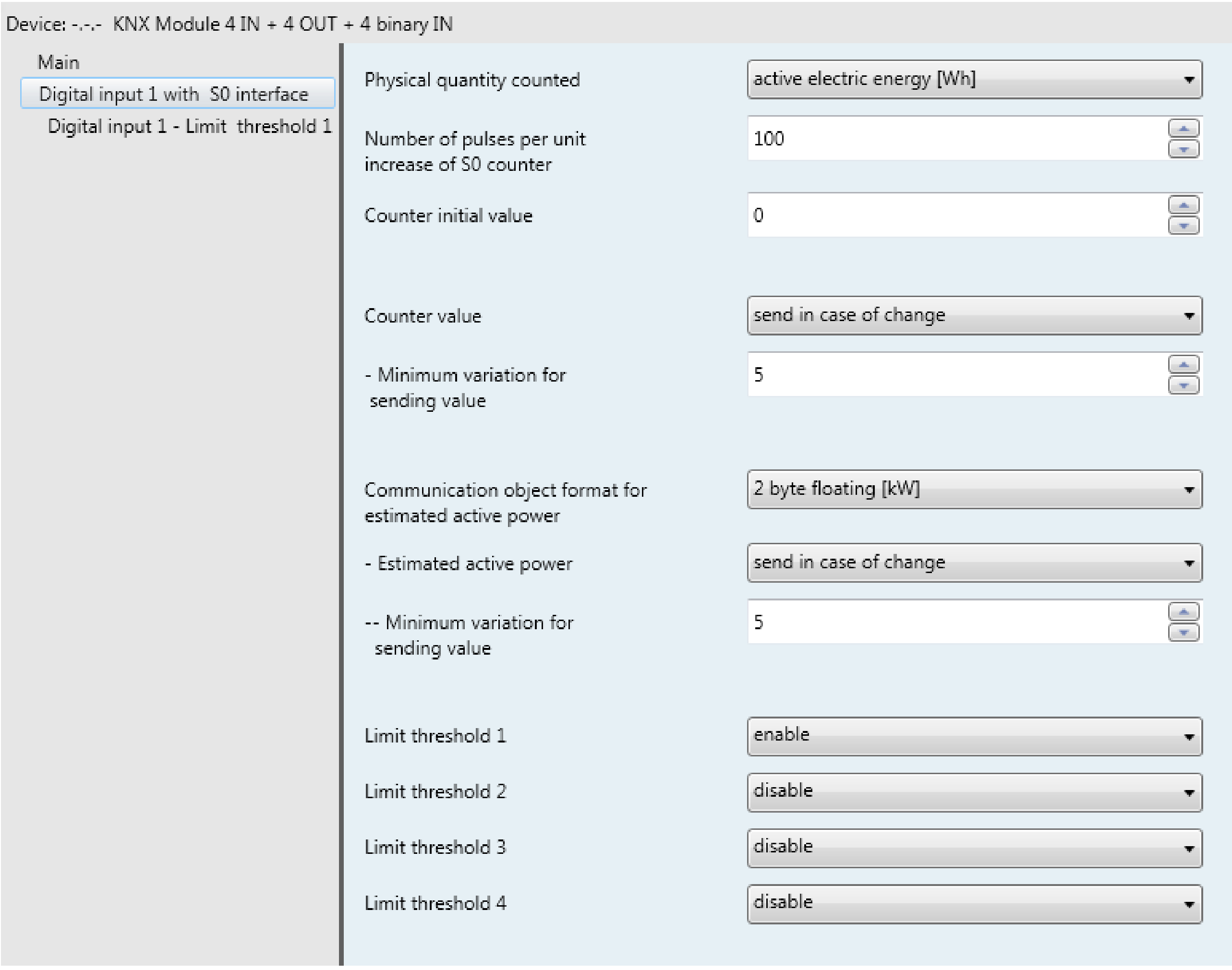


Fig. 7.1: “Digital input x with S0 interface” menu

7.1 Parameters

➢ *7.1.1 Physical quantity counted*

The parameter **“Physical quantity counted”** is used to set the physical quantity to be counted and as a result the format and coding of the communication object that will be sent by the device to signal the counted consumption. The values that can be set are:

- **active electric energy [Wh]** **(default value)**
  this setting displays the object ***IN.x- Active energy counter [Wh]*** (Data Point Type: 13.010 DPT_ActiveEnergy)

- active electric energy [kWh]
  this setting displays the object ***IN.x- Active energy counter [kWh]*** (Data Point Type: 13.013 DPT_ActiveEnergy_kWh)
- water [m^3]
  this setting displays the object ***IN.x- Water counter [m^3]*** (Data Point Type: 14.076 DPT_Value_Volume)
- gas [m^3]
  this setting displays the object ***IN.x- Gas counter [m^3]*** (Data Point Type: 14.076 DPT_Value_Volume)

➢ *7.1.2 Number of pulses per unit increase of S0 counter*

Each consumption meter with S0 output is used to set the number of output pulses that correspond to the unit increase of the quantity counted. For this reason, the parameter **"Number of pulses per unit increase of S0 counter"** is used to set the number of S0 pulses that correspond to a unit increase in the quantity to be counted; The values that can be set are:

- from 1 to 65535 with steps of 1 **(default value 100)**

➢ *7.1.3 Counter initial value*

In order to align the value of the KNX counter to the effective value of the interfaced physical counter, it is possible to define the initial value of the counter via the parameter **"Counter initial value"**. The values to be entered are substantially the values read on the physical counter at the moment the device is started up; from this moment, the displayed values and those actually recorded should coincide. The possible values for these parameters are:

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

The initial value can always be corrected via the communication object ***IN.x - Counter initial value*** (Data Point Type: 12.001 DPT_Value_4_Ucount); when the new initial value is received, the value of the KNX counter is re-initialised to the new received value.

The counter value can be reset via the communication object ***IN.x - Counter reset*** (Data Point Type: 1.017 DPT_Trigger); when this object receives a BUS telegram with the logical value "0" or "1", the device automatically resets the counter to its initial value, cancelling the previously stored value.

➢ *7.1.4 Counter value*

The parameter **"Counter value"** is used to define the conditions for sending the consumption counted by the device via the communication object ***IN.x - Active energy counter [Wh]***, ***IN.x - Active energy counter [kWh]***, ***IN.x - Water counter [m^3]*** or ***IN.x - Gas counter [m^3]*** (depending on the selected quantity to be counted); The values that can be set are:

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

Selecting the value **send in case of change** or **send on change and periodically**, displays the parameter **"Minimum variation for sending value "** whereas selecting the value **send periodically** or **send on change and periodically** displays the parameter **"Value sending period [minutes]"**.

Selecting the value **send on demand only**, no new parameter will be enabled because the counter 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 counted value.

The parameter **"Minimum variation for sending value"**, which is visible if the counted value is sent on change, is used to define the minimum variation of the counted value (in relation to the last value sent) that causes the new value to be spontaneously sent; The values that can be set are:

- from 1 to 1000 with steps of 1 **(default value 5)**

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

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

➢ *7.1.5 Communication object format for estimated active power*

If the physical quantity counted is **active electric energy [Wh]** or **active electric energy [kWh]**, it is possible to estimate the active power from the count of the active energy (the power is equal to the derived energy over time); the parameter **"Communication object format for estimated active power"** is used to define the format and coding of the BUS telegrams sent by the device via the communication object ***IN.x - Estimated active power***. The possible values are:

- 2 byte floating
  with this setting, the format of the above cited object is *9.024 DPT_Power [kW]*
- **4 byte floating (default value)**
  with this setting, the format of the above cited object is *14.056 DPT_Value_Power [W]*

The parameter **"Estimated active power"** is used to define the conditions for sending the estimated active power by the device via the communication object ***IN.2 - Estimated active power***; The values that can be set are:

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

Selecting the value **send in case of change** or **send on change and periodically**, displays the parameter **"Minimum variation for sending value "** whereas selecting the value **send periodically** or **send on change and periodically** displays the parameter **"Value sending period [minutes] "**.
Selecting the value **send on demand only**, no new parameter will be enabled, as the estimated active power 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 estimated value.

The parameter **Minimum variation for sending value** , which is visible if the estimated power is sent on change, is used to define the minimum variation of the estimated value (in relation to the last value sent) that causes the new value to be spontaneously sent; The values that can be set are:

- from 1 to 1000 with steps of 1 **(default value 5)**

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

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

7.1.6 Limit threshold 1,2,3,4

The parameters **"Limit threshold 1"**, **"Limit threshold 2"**, **"Limit threshold 3"** and **"Limit threshold 4"** are used to activate the thresholds of the relative digital inputs with S0 interface, displaying the various configuration menus. The values that can be set are:

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

Selecting the value **enable** displays the configuration menus **Digital input 2 - Limit threshold 1**, **Digital input 2 - Limit threshold 2**, **Digital input 2 - Limit threshold 3** and **Digital input 2 - Limit threshold 4** depending on the parameter of reference.

7.1.6.1 Digital input x – Limit threshold y

For each universal input configured as a digital input with S0 interface, it is possible to configure up to 4 limit 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 **Digital input x with S0 interface** menu.

The structure of the menu is as follows:

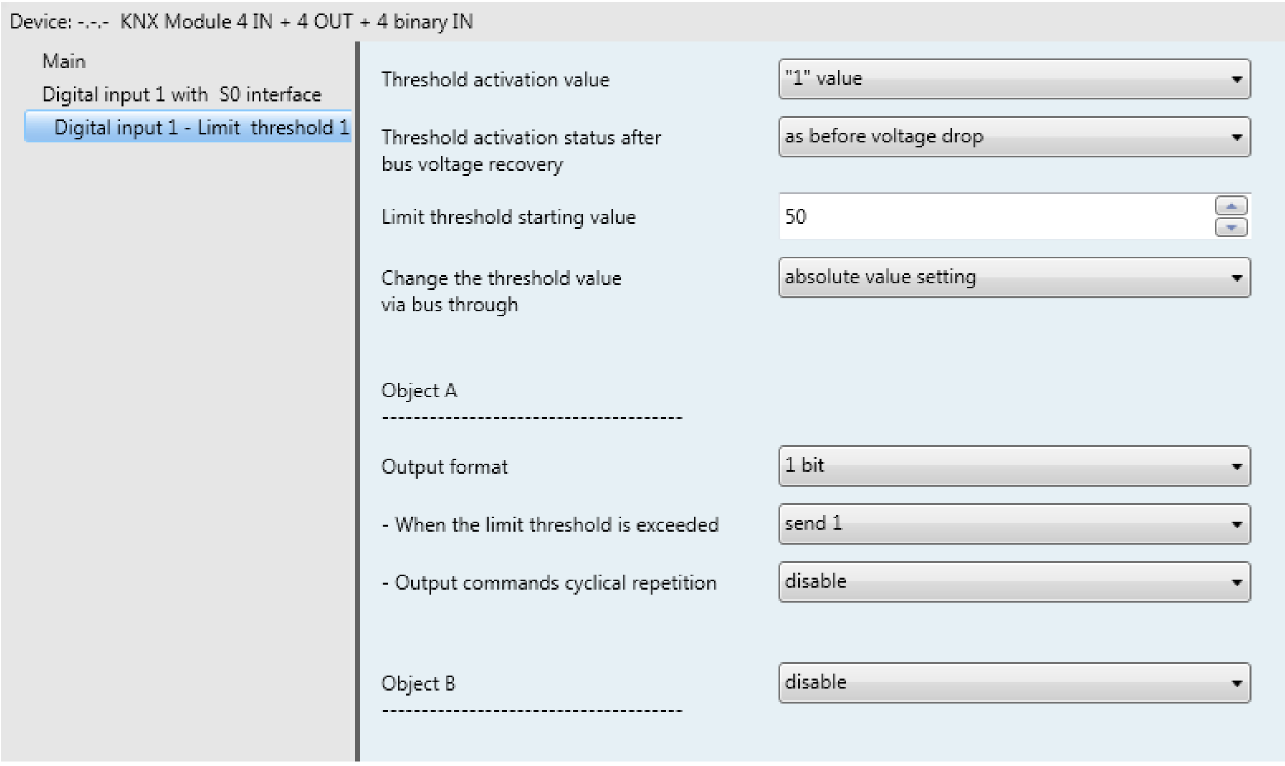


Fig. 7.2: "Digital input 2 – Limit threshold y" menu

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

This menu can be used to configure the following parameters:

The parameter **"Threshold activation value"** determines which logic value received via the communication object ***IN.x - Limit threshold activation y*** activates the limit threshold y; Receiving the value opposite to the one set for activation will deactivate the threshold. The possible values are:

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

The parameter **"Threshold activation status after bus voltage recovery"** is used to set the status of the limit threshold y after the BUS power supply voltage is restored; The possible values are:

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

regardless of the value set to this parameter, after an application ETS download the threshold status is always enabled.

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

- from 0 to 4294967295 with step of 1, **(default value 5000)**

The **"Change the threshold value via BUS through"** parameter defines the communication object format needed to set the limit threshold via a BUS telegram. The values that can be set are:

- **absolute value setting** **(default value)**
- increase/decrease step regulation

Selecting the value **absolute value setting** displays the communication object ***IN.x - Limit threshold value input y*** (Data Point Type: 14.005 DPT_Value_Amplitude) through which it is possible to set the value of the limit threshold via BUS.
Selecting **increase/decrease step setting** displays the parameter **"Threshold regulation step via bus"** and the object ***IN.x - Limit threshold regulation y*** (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 **"Threshold regulation step via bus"**; If the value "0" is received on this object, the limit threshold value will be decreased by the value defined by the parameter **"Threshold regulation step via bus"**.

The parameter **"Threshold regulation step via bus"** 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 2000 with steps of 1 **(default value 5)**

The current value of the limit threshold y is sent on BUS via the object ***IN.x - Limit threshold feedback y*** (Data Point Type: 14.005 DPT_Value_Amplitude); The feedback sending conditions are, following a BUS request, spontaneously at each threshold change and at BUS voltage recovery.

For each threshold, up to 2 different objects can be sent (which are distinguished by the letters A and B) when the limit threshold is exceeded; object A is always enabled, whereas the **"Object B"** parameter can be used to enable a new object to be sent. The parameter may assume the following values:

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

selecting **enable** displays the object ***IN.x - Output B Limit threshold y*** and the parameters **"Output format"** and **"When the limit threshold is exceeded"** grouped in the subgroup **Object z** (z is the index of the object associated with the channel, between **A** and **B)**.

The parameter **"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** **(default value)**
- 2 bit
- 1 byte unsigned
- 1 byte signed
- 1 byte percentage
- 1 byte HVAC
- 2 byte unsigned
- 2 byte signed

Depending on the value set for this item, the values that can be set for the **"When the limit threshold is exceeded"** parameter will be different.
The parameter **"When the limit threshold is exceeded"** is used to set the command or value to be sent after the limit value is exceeded.

- If the output format is **1 bit**, this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 1.001 DPT_Switch) and the values that can be set for the above parameter are:

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

- If the output format is **2 bit**, this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 2.001 DPT_Switch_Control) and the values that can be set for the above parameter are:

  - **send forcing activation on (down)** **(default value)**
  - send forcing activation off (up)
  - send deactivate forcing

- If the output format is **1 byte unsigned** , this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 5.010 DPT_Value_1_Ucount) and the values that can be set for the above parameter are:

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

- If the output format is **1 byte signed** , this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 6.010 DPT_Value_1_Count) and the values that can be set for the above parameter are:

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

- If the output format is **1 byte percentage value**, this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 5.001 DPT_Scaling) and the values that can be set for the above parameter are:

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

- If the output format is **1 byte HVAC mode**, this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 20.102 DPT_HVACMode) and the values that can be set for the above parameter are:

  - send auto

- **send comfort** **(default value)**
- send precomfort
- send economy
- send off (building protection)

- If the output format is **2 byte unsigned** , this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 7.001 DPT_Value_2_Ucount) and the values that can be set for the above parameter are:

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

- If the output format is **2 byte signed** , this displays the communication object ***IN.x - Output z limit threshold y*** (Data Point Type: 8.001 DPT_Value_2_Count) and the values that can be set for the above parameter are:

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

The object ***IN.x - Output z limit threshold y*** is sent upon request, spontaneously when the limit threshold is exceeded, periodically (if cyclical repetition is enabled) and when voltage is restored. When the threshold is disabled, the sending of the telegrams associated with exceeding the limit threshold is inhibited, whereas any change or feedback of the threshold value is still carried out.

The **"Cyclical repetition of the output commands"** 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 **"Command repetition period"** 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)**

8 *“Thermostat x”* menu

If the operating mode for the universal input is “Thermostat”, a dedicated menu called **Thermostat x** is displayed for each input (x = 1 .. 4, is the input index). All 4 menus are identical and therefore, for the sake of simplicity, the operation and dedicated parameters are summarised in this paragraph, indicating the reference menu with a generic “X” (1 .. 4).

The basic structure of the menu is as follows:

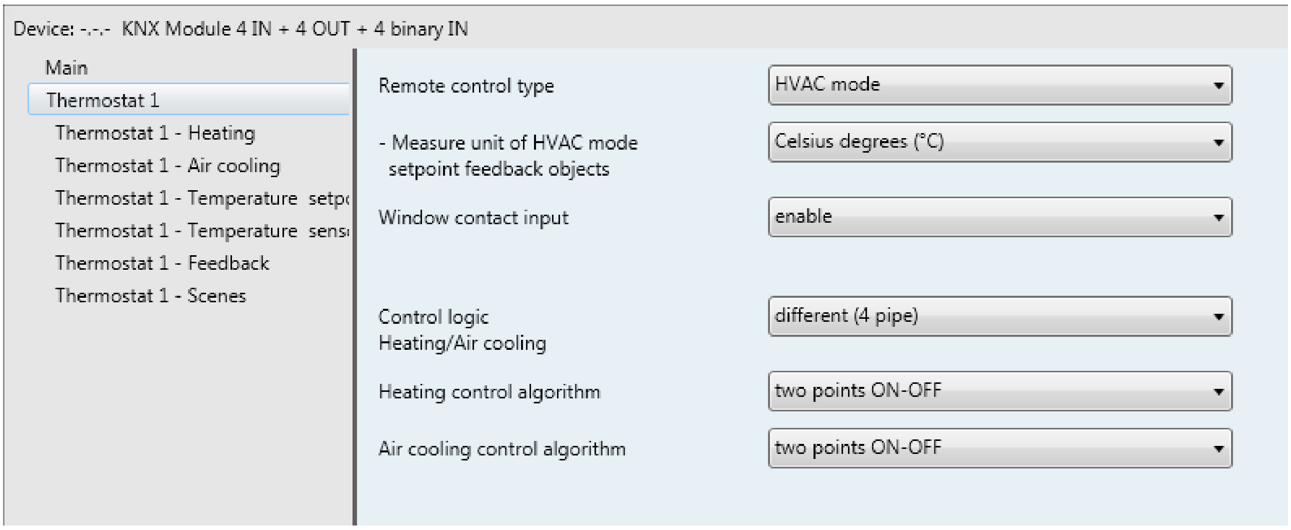


Fig. 8.1: "Thermostat" menu

The device is configured so it can manage the temperature adjustment system with the aid of a remote device (such as the timed thermostat/programmer or KNX thermostat), by connecting a NTC sensor to the universal input terminals for temperature measurement. With this configuration the device does not control the entire system but only a part of it, called a zone. Whereas there is a remote device in the system that controls its mode and functioning type; in this case, the device thermostat function controls the temperature of the environment where it is located whereas it is the remote device that decides the functioning set by the user. The operating parameters of the device cannot be changed locally.\

The device management of the temperature adjustment functioning type (heating/cooling) is managed remotely via the communication object ***IN.x - Functioning type*** (Data Point Type: 1.100 DPT_Heat/Cool) which makes it possible to receive the remote functioning type setting commands.
When the application is downloaded, the set functioning type is HEATING.
With the same communication object ***IN.x - Functioning type***, the device sends the feedback of the functioning type (heating/cooling) set on the device via BUS telegram. The object sending conditions are upon demand (read request), spontaneously on change of the type of active functioning and on BUS voltage recovery.
The object ***IN.x - Functioning type*** is an input/output object; this is because with KNX technology, a communication object is sent to a single destination group address, if this object is associated with more than one group address, the device will send the BUS telegram to the group address where the object has the “S” flag (sending).

8.1 Parameters

➢ *8.1.1 Remote control type*

The device can be controlled remotely through a single setpoint or by changing the active HVAC mode; the parameter **"Remote control type"** is used to set the type of control the remote device executes on the thermostat function; The values that can be set are:

- **HVAC mode** **(default value)**
- setpoint (°C)
- setpoint (°K)
- setpoint (°F)

Selecting **HVAC mode** displays the parameter **"Measure unit of HVAC mode setpoint feedback objects"** and the communication object ***IN.x - HVAC mode input*** (Data Point Type: 20.102 DPT_HVACMode) via which the remote device changes the HVAC mode; selecting **setpoint** displays the communication object ***IN.x - Setpoint input*** (Data Point Type: *9.001 DPT_Value_Temp* if the value is **setpoint (°C)**, *9.002 DPT_Value_Tempd* if the value is **setpoint (°K)** and *9.027 DPT_Value_Temp_F* if the value is **setpoint (°F)**) via which the remote device changes the operating setpoint.

The parameter **"Measure unit of HVAC mode setpoint feedback objects"** is used to enable the communication objects necessary for setting the setpoints for every device mode via BUS telegram; The values that can be set are:

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

Depending on the value set for the parameter, this changes the coding of the communication objects ***IN.x - Heating anti-freeze setpoint*** , ***IN.x - Heating economy setpoint***, ***IN.x - Heating precomfort setpoint***, ***IN.x - Heating comfort setpoint***, ***IN.x - Air cooling high temperature protection setpoint***, ***IN.x - Air cooling economy setpoint***, ***IN.x - Air cooling precomfort setpoint*** and ***IN.x - Air cooling comfort setpoint*** (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 setpoints of every device operating mode via BUS.

Via the same communication objects, the device sends the current value feedback for the setpoint of each HVAC operating mode via BUS telegram. The sending conditions of the objects are on demand (read request), spontaneously on setpoint change and at BUS voltage recovery.
The objects ***IN.x - Heating anti-freeze setpoint***, ***IN.x - Heating economy setpoint***, ***IN.x - Heating precomfort setpoint***, ***IN.x - Heating comfort setpoint***, ***IN.x - Air cooling high temperature protection setpoint***, ***IN.x - Air cooling economy setpoint***, ***IN.x - Air cooling precomfort setpoint*** and ***IN.x - Air cooling comfort setpoint*** are input/output objects; this is because with KNX technology, a communication object is sent to a single destination group address, if these objects are associated with more than one group address, the device will send the BUS telegram to the group address where the objects have the "S" flag (sending).

There is a priority constraint between the different functions of the device and the different communication objects that can be used for the remote commands, as summarised in the following table:

Priority	Object	Size
Maximum	Window contact function via the BUS	1 bit
Minimum	HVAC mode input/Setpoint input/Scene	1 byte/2 byte/1 byte

If the remote control type is HVAC mode, between the various setpoints belonging to the same functioning type, there is a value setting limit that must be respected, which is determined by the following relationship:

- $T_{anti\text{-}freeze} \leq T_{economy} \leq T_{precomfort} \leq T_{comfort}$ in heating mode ("T" indicates the general value of the mode setpoint)
- $T_{comfort} \leq T_{precomfort} \leq T_{economy} \leq T_{high\ temperature\ protection}$ in air cooling mode ("T" indicates the generic value of the mode setpoint)

If the remote control type is setpoint, between the various setpoints belonging to the same functioning type, there is a value setting limit that must be respected, which is determined by the following relationship:

- 5°C ≤ $T_{operation}$ in heating mode ("T" indicates the generic setpoint value)
- $T_{operation}$ ≤ 40°C in air cooling mode ("T" indicates the generic setpoint value)

➢ *8.1.2 Window contact input*

The device implements the window contact function which makes it possible, when a window open condition is detected by a remote device, to force the device to the HVAC OFF/Building Protection mode if the remote control type is **HVAC mode** or to set the BUILDING PROTECTION setpoint if the remote control type is **setpoint.**
When the window closed condition is restored, the device resumes the condition it was in beforehand. The parameter **"Window contact input"** is used to enable the "window contact" function of the device.

The values that can be set are:

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

Select **enable** to view the ***Window status input*** object (Data Point Type: 1.019 DPT_Window_Door) which makes it possible for the device to be aware of the window status.
To enable the status read request via this object at bus voltage recovery, you have to enable the Transmission (T) and Update (U) ETS flags. The update at bus voltage recovery is not successful if the ***Window status input*** object is connected to an object belonging to the same device.

➢ *8.1.3 Control logic Heating/Air cooling*

The thermostat function implements a stand alone control logic by using various control algorithms. Given the different types of temperature adjustment systems, it is possible to dedicate a common solenoid valve control object to the heating and air cooling system or dedicate one to each of the two types of operation. The **"Control logic Heating/Air cooling"** parameter defines whether the system control logic (and therefore the control communication object) is common for the heating and air cooling or whether it is different. The values that can be set are:

- common (2 pipe)
- **different (4 pipe)** **(default value)**

select **common (2 pipe)** to view the parameters **"Heating/air cooling control algorithm"** and **"Heating/air cooling valve status feedback"**, whereas select **different (4 pipe)** to view the parameters **"Heating control algorithm"** and **"Air cooling control algorithm"**.

➢ *8.1.4 Heating control algorithm*

The parameter **"Heating control algorithm"** is used to define the control algorithm used for the heating system; The values that can be set are:

- **2 points ON-OFF** **(default value)**
- PWM proportional-integral
- fancoil with ON-OFF speed control

Selecting **2 points ON-OFF** displays the parameter **"Regulation differential (tenth of °C)"** in the **Thermostat x - Heating** menu and the communication object ***IN.x -Heating valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.

Selecting **PWM proportional integral** displays the parameters **"Select heating system"**, **"Proportional band"**, **"Integration time"** and **"Cycle time"** in the **Thermostat x - Heating** menu and the communication object ***IN.x - Heating valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.

Select **fancoil with ON-OFF speed control** to view the parameters **"Valve regulation differential (tenth of °C)"**, **"Speed 1 regulation differential (tenth of °C)"**, **"Speed 2 regulation differential (tenth of °C)"**, **"Speed 3 regulation differential (tenth of °C)"**, **"Speed 1 inertia time (seconds)"**, **"Speed 2 inertia time (seconds)"**, **"Speed 3 inertia time (seconds)"** and **"Fancoil speed status feedback"** in the **Thermostat x - Heating** menu and the communication objects ***IN.x - Heating fan V1 switching***, ***IN.x - Heating fan V2 switching*** and ***IN.x Heating fan V3 switching***.

➢ *8.1.5 Air cooling control algorithm*

The parameter **"Air cooling control algorithm"** is used to define the control algorithm used for the air cooling system; The values that can be set are:

- **2 points ON-OFF** **(default value)**
- PWM proportional-integral
- fancoil with ON-OFF speed control

Selecting **2 points ON-OFF** displays the parameter **"Regulation differential (tenth of °C)"** in the **Thermostat x - Air cooling** menu and the communication object ***IN.x -Air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.

Selecting **PWM proportional integral** displays the parameters **"Select air cooling system"**, **"Proportional band"**, **"Integration time"** and **"Cycle time"** in the **Thermostat x - Air cooling** menu and the communication object ***IN.x - Air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.

Select **fancoil with ON-OFF speed control** to view the parameters **"Valve regulation differential (tenth of °C)"**, **"Speed 1 regulation differential (tenth of °C)"**, **"Speed 2 regulation differential (tenth of °C)"**, **"Speed 3 regulation differential (tenth of °C)"**, **"Speed 1 inertia time (seconds)"**, **"Speed 2 inertia time (seconds)"**, **"Speed 3 inertia time (seconds)"** and **"Fancoil speed status feedback"** in the **Thermostat x - Air cooling** menu and the communication objects ***IN.x - Air cooling fan V1 switching***, ***IN.x - Air cooling fan V2 switching*** and ***IN.x Air cooling fan V3 switching***.

➢ *8.1.6 Heating/air cooling control algorithm*

The parameter **"Heating/air cooling control algorithm"** is used to define the control algorithm used both for the heating system as well as for the air cooling system, as the control logic is common; The values that can be set are:

- **2 points ON-OFF** **(default value)**
- PWM proportional-integral
- fancoil with ON-OFF speed control

Select **2 points ON-OFF** in the **Thermostat x - Heating** and **Thermostat x - Air cooling** menus to display the parameters **"Regulation differential (tenth of °C)"** and the communication object ***IN.x. - Heating/air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.

Select **PWM proportional integral** in the **Thermostat x - Heating** and **Thermostat x - Air cooling** menus to display the parameters **"Select heating system**(**air cooling** in the **Thermostat x - Air cooling menu)"**, **"Proportional band"**, **"Integration time"** and **"Cycle time"** and the communication object ***IN.x - Heating/air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.

Selecting **fancoil with ON-OFF speed control** in the **Thermostat x - Heating** and **Thermostat x - Air cooling** menus displays the parameters **"Speed 1 regulation differential (tenth of °C)"**, **"Speed 2 regulation differential (tenth of °C)"**, **"Speed 3 regulation differential (tenth of °C)"**, **"Speed 1 inertia time (seconds)"**, **"Speed 2 inertia time (seconds)"**, **"Speed 3 inertia time (seconds)"**, **"Fancoil speed status feedback"** and **"Valve regulation differential (tenth of °C)"** and the communication objects ***IN.x - Heating fan V1 switching***, ***IN.x - Heating fan V2 switching***, ***IN.x - Heating fan V3 switching***, ***IN.x - Air cooling fan V1 switching*** , ***IN.x - Air cooling fan V2 switching*** and ***IN.x - Air cooling fan V3 switching***.
If the control algorithm is fancoil, the format of the heating/air cooling solenoid valve commands (2-way system) is independent of that of the fancoil speed control; the solenoid valve control logic when the selected algorithm is fancoil is **2 points ON-OFF**. Via the communication object ***IN.x - Heating/air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) the device sends the command telegrams to the solenoid valve.

➢ *8.1.7 Heating/air cooling valve status feedback*

The parameter **"Heating/air cooling valve status feedback"** is used to enable the device to receive feedback from the actuator that commands the heating/air cooling solenoid valve; in this way, the device is able to receive the telegram after the solenoid valve switched and to repeat the command if the switching did not take place. The values that can be set are:

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

Select **disable** to view the parameter **"Command repetition period with disabled feedback"**; selecting **enable** displays the communication object ***IN.x - Heating/air cooling valve status feedback*** (Data Point Type: 1.001 DPT_Switch).
Upon BUS voltage recovery, the device sends the read request via the object ***IN.x - Heating/air cooling valve status feedback*** to be updated about the status of the heating/air cooling solenoid valve.

With feedback enabled, after the device sends the switching command to the solenoid valve, it waits for one minute of its clock for the actuator to send the feedback that switching took place; if this does not take place, it sends the command again to the solenoid valve every minute until it receives the feedback of correct switching. It can happen that, during normal operation of the temperature adjustment, the actuator status can be changed by an entity external of the thermostat, that forces its status, modifying it. In this case, the device repeats the valve switching command to realign the status of the actuator with the one determined by the control logic of the thermostat, triggering the process for waiting for confirmation and repeating the command until the confirmation is received.

With the solenoid valve status feedback disabled, it may be useful to cyclically repeat the command to the actuator that manages the solenoid valve so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The parameter **"Command repetition period with disabled feedback"** is used to define the frequency of the cyclical sending; The values that can be set are:

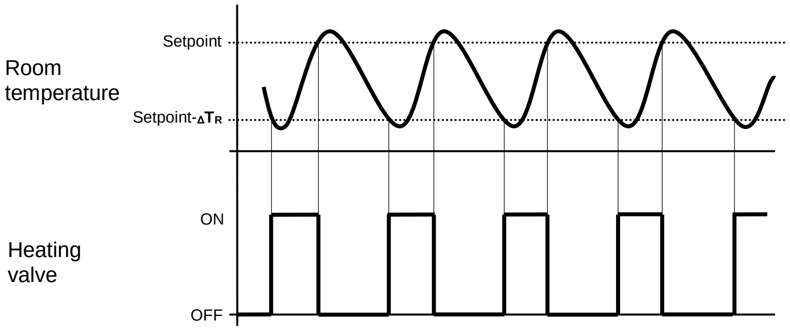
- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** **(default value)**

8.2 Control algorithms

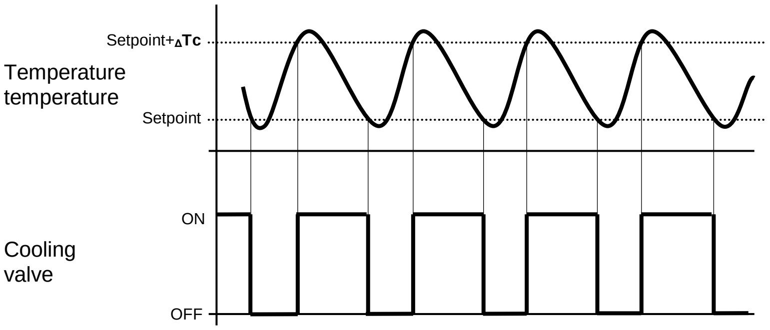
Independently of the fact if the control logic is common or different between the two operating types, the logic is as follows depending on the selected algorithm:

- ***2 points ON-OFF***

The algorithm used for controlling the temperature adjustment system is the classic type, called 2-point control. This type of control involves the turning on and off of the temperature adjustment system following a hysteresis cycle. This means there isn't a single threshold that discriminates between the turning on and off of the system, but two.



When the measured temperature is lower than the value setpoint-**ΔTR**“ (where **ΔTR** identifies the value of the heating regulation differential) the device turns on the heating system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches the fixed setpoint value, the device turns off the heating system, sending the relative BUS command to the actuator that manages it.
This makes it clear that there are two decision thresholds for turning the heating system on and off, the first consists of the value “setpoint- **ΔTR**“ below which the device turns on the system, the second consists of the setpoint value that was set, above which the device turns off the system.



When the measured temperature is higher than the value “setpoint+ **ΔTc**“ (where **ΔTc** identifies the air cooling regulation differential) the device turns on the air cooling system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches the fixed setpoint value, the device turns off the air cooling system, sending the relative BUS command to the actuator that manages it.
This makes it clear that there are two decision thresholds for turning the air cooling system on and off, the first is the setpoint value that was set, below which the device turns off the system, the second is the value “setpoint+**ΔTc**“ above which the device turns on the system.
To avoid the continuous switching of the solenoid valves, after an OFF-ON-OFF sequence, the next ON command can only be sent after at least 2 minutes have elapsed.

- ***PWM proportional-integral***

The algorithm used to control the temperature adjustment system allows you to drastically reduce the times subject to thermal inertia and introduced by the 2-point control, called PWM control. This type of control involves the modulation of the impulse duty-cycle, represented by the temperature adjustment system activation time, on the basis of the difference between the fixed set-point and the temperature effectively detected. Two components are needed to calculate the output function: the proportional component and the integral component.

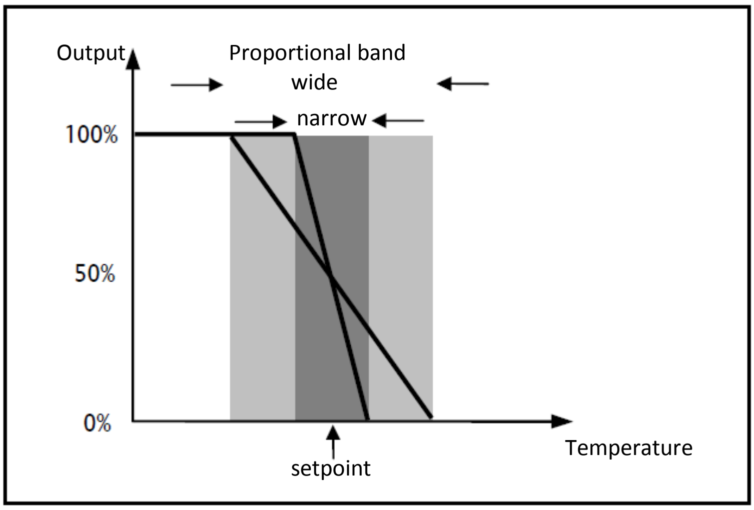
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

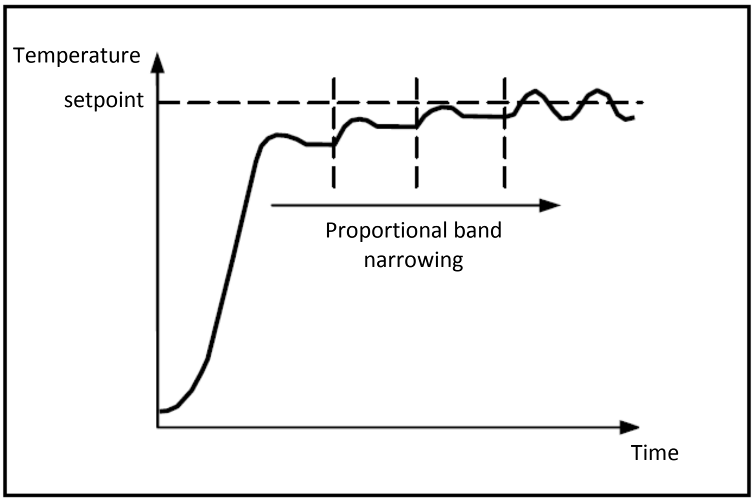
Proportional component
In the proportional component, the output function is proportional to the error (difference between setpoint and measured temperature).

$$P_{\text{out}} = K_p\, e(t)$$

Once the proportional band is defined, the duty-cycle within the band varies between 0% and 100%; outside the band, the duty-cycle will be maximum or minimum depending on the reference limits.

The width of the proportional band determines the extent of the response to the error. If the band is too “narrow”, the system will oscillate as it becomes more reactive; if the band is too "wide" the control system is slow. The ideal situation is when the proportional band is as narrow as possible without causing oscillations.
The diagram below shows the effect of narrowing the proportional band until the oscillation point of the output function. A "wide" proportional band results as a straight line in the control, but with an initial error between the setpoint and the actually perceptible temperature. As the band becomes narrower, the temperature approaches the reference value (setpoint) until it becomes unstable and starts to oscillate around it.



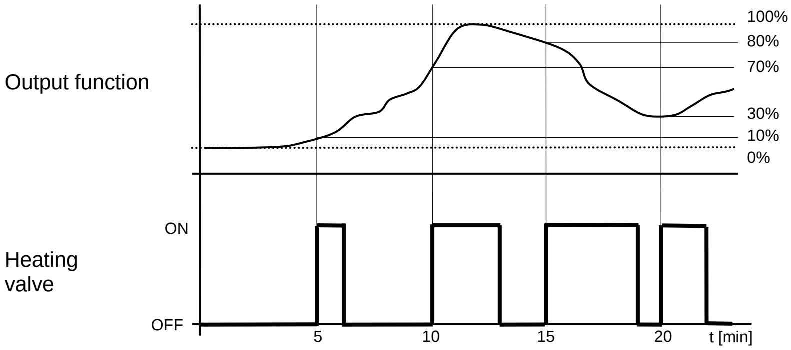


Integral component
The contribute of the integral period is proportional to the error (difference between the setpoint and the measured temperature) and its duration. The integral is the sum of the instantaneous error for every moment of time and provides the accumulated offset that should have been previously corrected. The accumulated error is then added to the regulator output.

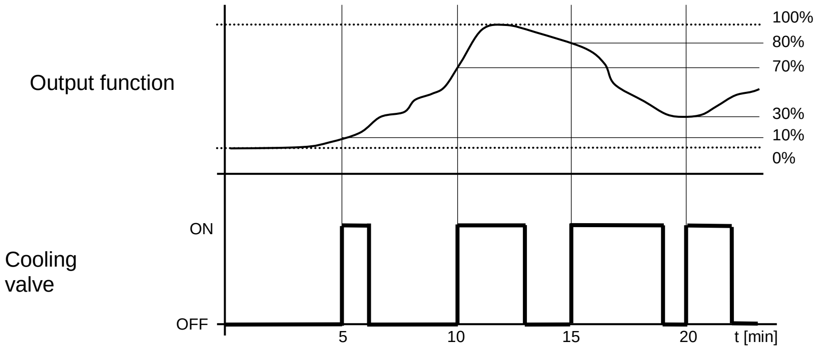
$$I_{\mathrm{out}} = K_i \int_0^t e(\tau)\, d\tau$$

The integral period accelerates the dynamics of the process towards the setpoint and eliminates the residuals of the stationary error status that takes place with a pure proportional controller.

The integration time is the parameter that determines the action of the integral component. The longer the integration time, the slower the modification of the output and hence the slower the system response. If the time is too short, the threshold value will be exceeded (overshoot), and the function will swing around the set-point.



The device keeps the heating system switched on for a cycle time percentage that depends on the output function of the proportional-integral control; the device continuously regulates the heating system, modulating the system turning on-off times with a duty-cycle (shown to the right along the vertical axis) that depends on the output function value calculated at every time interval equal to the cycle time. The cycle time is reinitialised every time the reference set-point is modified.
With this type of algorithm, there is no longer a hysteresis cycle on the heating device, so the inertia times (system heating and air cooling times) introduced by the 2-point control are eliminated. This produces energy savings because the system does not remain switched on when it is not needed and, once the required temperature has been reached, it continues to provide a heat limited contribution to compensate for the environmental heat dispersion.

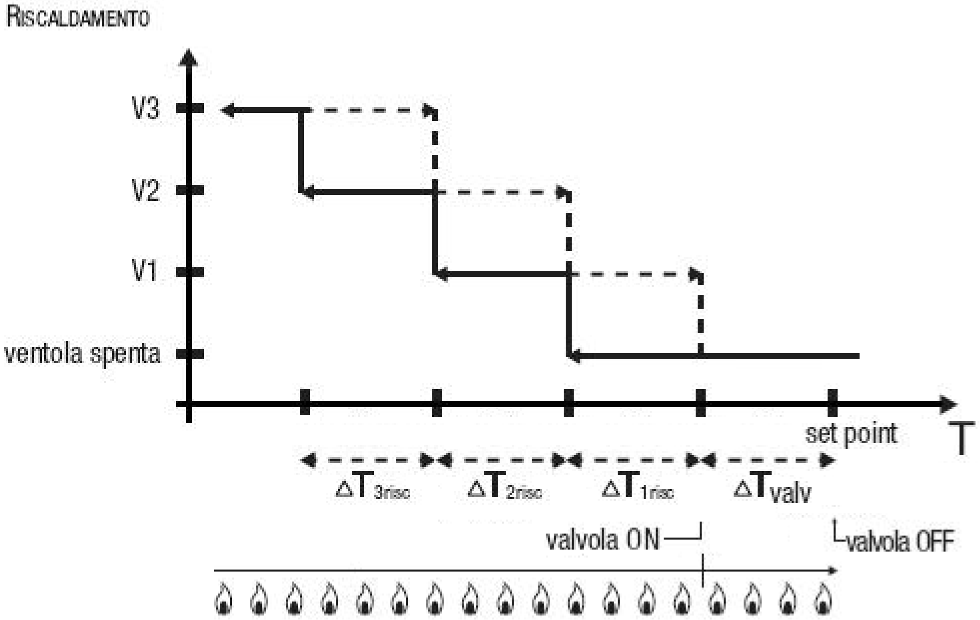


As seen in the figure, the device keeps the air cooling system switched on for a cycle time percentage that depends on the output function of the proportional-integral control; the device continuously regulates the air cooling system, modulating the system turning on-off times with a duty-cycle (shown to the right along the vertical axis) that depends on the output function value calculated at every time interval equal to the cycle time. The cycle time is reinitialised every time the reference set-point is modified.

With this type of algorithm, there is no longer a hysteresis cycle on the air cooling device, so the inertia times (system air cooling and heating times) introduced by the 2-point control are eliminated. This produces energy savings because the system does not remain switched on when it is not needed and, once the required temperature has been reached, it continues to provide a limited contribution of cold air to compensate for the contribution of environmental heat.

- ***fancoil with ON-OFF speed control***

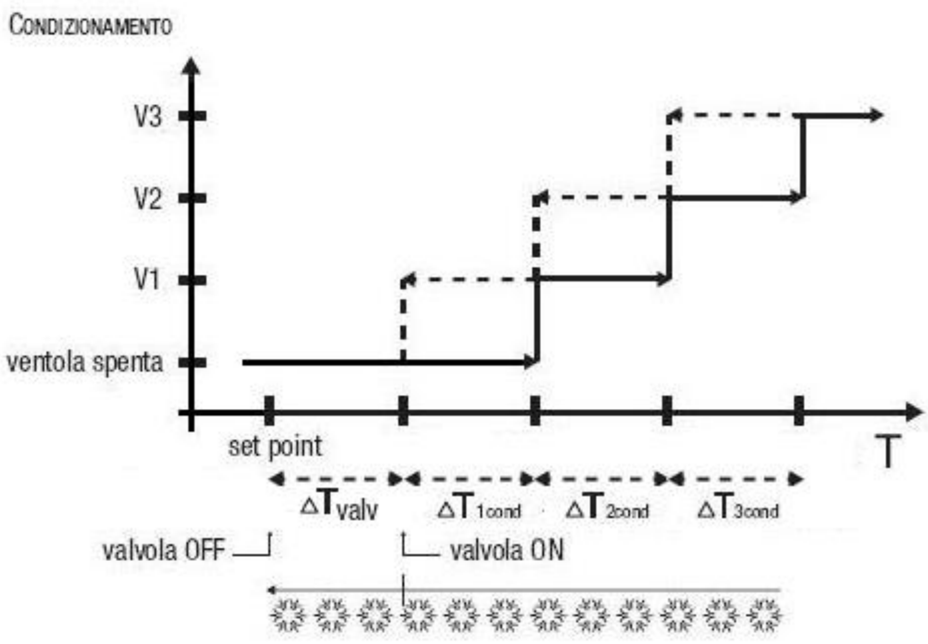
The type of control that is applied when the fancoil control is enabled is similar to the 2-point control analysed in previous sections, which is to turn the fancoil speed on /off based on the difference between the setpoint that was set and the measured temperature.
The substantial difference with the 2-point algorithm is that, in this case, there is only one stage on which the hysteresis cycle is carried out, fixing the speed on and off thresholds, but there can be three (depending on the number of fancoil speeds); substantially, this means that each stage corresponds to a speed and when the difference between the measured temperature and the setpoint that was set causes a certain speed to be turned on, this means that before turning on the new speed, the other two must absolutely be turned off.



The figure refers to the control of the fancoil speeds with three operating stages for heating. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine its activation and deactivation. The thresholds are determined by values set for the various regulation differentials, and can be summarised as follows:

- Speed V1 (1st stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$" and turned off when the temperature value reaches the value "setpoint-ΔT_{valv}" (or the "setpoint" value if $\Delta T_{1\ heat}$=0). The first speed is also turned off when a higher speed needs to be activated
- Speed V2 (2nd stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$" and turned off when the temperature value reaches the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$". The second speed is also turned off when the V3 speed needs to be activated
- Speed V3 (3rd stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$-$\Delta T_{3\ heat}$" and turned off when the temperature value reaches the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$"

With regard to the heating solenoid valve, once the measured temperature is lower than the value "setpoint-ΔT_{valv}", the device sends the activation command to the solenoid valve that manages the heating system; the solenoid valve is deactivated when the detected temperature reaches the fixed setpoint value. In this way, the heating of the fancoil can also be exploited for irradiation, without any speed being activated.



The figure refers to the control of the speeds of a fancoil with three operating stages for air cooling. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine its activation and deactivation. The thresholds are determined by values set for the various regulation differentials, and can be summarised as follows:

- Speed V1 (1st stage): the speed is turned on when the temperature value is higher than the value "setpoint+ΔT_{valv}+ΔT_{1cool}" and turned off when the temperature value reaches the value "setpoint+ΔT_{valv}" (or the "setpoint" value if $\Delta T_{1\ cool}$=0). The first speed is also turned off when a higher speed needs to be activated
- Speed V2 (2nd stage): the speed is turned on when the temperature value is higher than the value "setpoint+ΔT_{valv}+ΔT_{1cool}+ΔT_{2cool}" and turned off when the temperature value reaches the value "setpoint+ΔT_{valv}+$\Delta T_{1\ cool}$". The second speed is also turned off when the V3 speed needs to be activated
- Speed V3 (3rd stage): the speed is turned on when the temperature value is higher than the value "setpoint+ΔT_{valv}+ΔT_{1cool}+ΔT_{2cool}+ΔT_{3cool}" and turned off when the temperature value reaches the value "setpoint+ΔT_{valv}+ΔT_{1cool}+ΔT_{2cool}"

With regard to the air cooling solenoid valve, once the measured temperature is higher than the value "setpoint+ΔT_{valv}", the device sends the activation command to the solenoid valve that manages the air cooling system; the solenoid valve is deactivated when the detected temperature reaches the fixed setpoint value. In this way, the air cooling of the fancoil can also be exploited for irradiation, without any speed being activated.

To avoid continuous switchovers, the device can wait up to 2 minutes before sending the activation command to the actuator that controls the temperature adjustment system, or to the actuator channels that command the fan coil speeds.

8.3 "Thermostat x - Heating" menu

The **Thermostat x - Heating** menu contains the characteristic parameters of the load control algorithms for the heating system.

The structure of the menu is as follows:

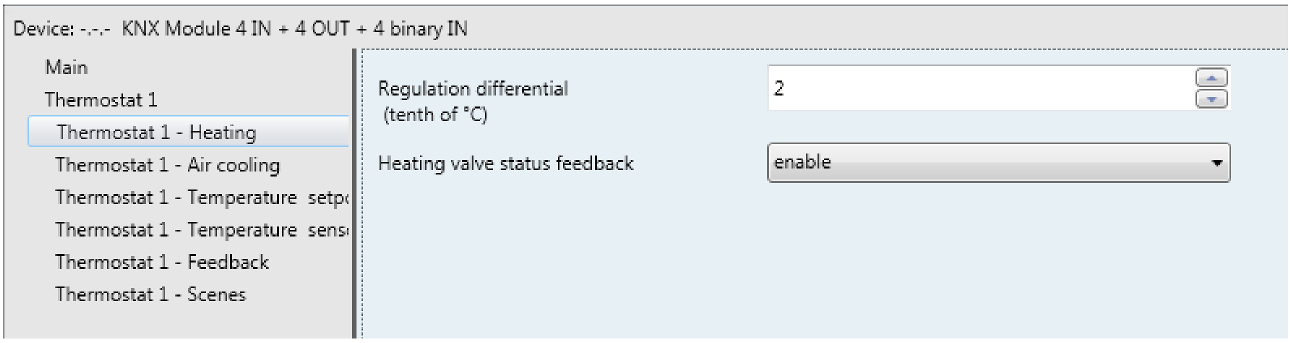


Fig. 8.2: “Thermostat x - Heating” menu

The Thermostat x - Heating menu varies based on what is set in the parameters “**Control logic Heating/Air cooling”** and “**Heating control algorithm**” in the “**Thermostat 1**” page.

All the possible parameters that can be displayed in this menu based on the system type are summarised below.

Parameters

➢ *8.3.1 Regulation differential (tenth of °C)*

The parameter **“Regulation differential (tenth of °C)”** is used to set the regulation differential value of the heating **two points ON-OFF** control algorithm, already mentioned in paragraph 10.2. Control algorithms, which, when subtracted from the value of the setpoint that was set, determines the threshold value below which the heating system is turned on in the 2 points control. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

➢ *8.3.2 Select heating system*

The parameter **“Select heating system”** is used to automatically measure the operating parameters (Proportional band and Integration time) of the proportional integral algorithm based on the selected heating system. The values that can be set are:

- hot water heating
- **underfloor heating (default value)**
- fancoil unit
- electric heating
- customised

Selecting **hot water heating**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **5.0 °C** and **150** are displayed.

Selecting **underfloor heating**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **5.0 °C** and **240** are displayed.

Selecting **fancoil unit**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **4.0 °C** and **90** are displayed.

Selecting **electric heating**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **4.0 °C** and **100** are displayed.

Selecting **customised**, the **"Proportional band"** and **"Integration time (minutes)"** parameters will be displayed but cannot be modified.

The parameter **"Proportional band"** is used to set the width of the PWM proportional band of the heating **PWM proportional integral**, already cited in paragraph 10.2. Control algorithms, which, when subtracted from the value of the setpoint that was set, determines the lower limit of the proportional band used for the proportional integral control. The values that can be set are:

- 1.0 °C
- 1.5 °C
- **2.0 °C (default value)**
- 2.5 °C
- 3.0 °C
- 3.5 °C
- 4.0 °C
- 4.5 °C
- 5.0 °C
- 5.5 °C
- 6.0 °C
- 6.5 °C
- 7.0 °C
- 7.5 °C
- 8.0 °C
- 8.5 °C
- 9.0 °C
- 9.5 °C
- 10.0°C

The parameter **"Integration time"** is used to set the contribution of the integral action in the proportional integral control (See paragraph 10.2 Control algorithms). The values that can be set are:

- from 1 minute to 250 minutes with steps of 1, plus the value "no integral" (255) **(default value 60)**

Selecting **no integral**, the integral component is zero and the pure effect of proportional control is obtained.

The parameter **"Cycle time"** is used to set the value of the period within which the device carries out PWM modulation, modifying the duty-cycle. The values that can be set are:

- 5 minutes
- 10 minutes
- 15 minutes
- **20 minutes (default value)**
- 30 minutes
- 40 minutes
- 50 minutes
- 60 minutes

If the control algorithm is fancoil, the format of the heating solenoid valve commands (4-way system) is independent of that of the fancoil speed control. the solenoid valve control logic when the selected algorithm is fancoil is **2 points ON-OFF**. Via the communication object ***IN.x - Heating valve switching*** (Data Point Type: 1.001 DPT_Switch) the device sends the command telegrams to the heating solenoid valve.

The parameter **"Valve regulation differential (tenth of °C)"** is used to set the regulation differential value of the 2 points control of the fancoil operating solenoid valve, as mentioned in paragraph 10.2 Control algorithms. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

The parameter **"Speed 1 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the first speed of the heating control algorithm **fancoil with ON-OFF speed control**, already mentioned in paragraph 10.2 Control algorithms; This value, subtracted from the "setpoint-ΔT_{valv}" value, determines the threshold value below which speed 1 of the fancoil is turned on. The values that can be set are:

- from 0 to 20 with steps of 1 **(default value 2)**

Setting the value **0** obtains the condition "$\Delta T_{1\ heat}=\Delta T_{valv}$" for which the value of the speed 1 activation threshold is "setpoint-ΔT_{valv}" and the off value is "setpoint".

The parameter **"Speed 2 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the second speed of the heating control algorithm **fancoil with ON-OFF speed control** as mentioned in paragraph 10.2 Control algorithms; This value, subtracted from the "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$" value, determines the threshold value below which speed 2 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

The parameter **"Speed 3 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the third speed of the heating control algorithm **fancoil with ON-OFF speed control**, as mentioned in paragraph 10.2 Control algorithms; This value, subtracted from the "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$" value, determines the threshold value below which speed 3 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

When, according to the "fancoil with speed control" algorithm, the device must activate any speed and speed 1 is active, a delay can be inserted between the moment in which the speed 1 deactivation notice is received (or the moment when the speed 1 deactivation command is sent, if fancoil speed info is disabled) and the moment in which the activation command for the new speed is sent; the **"Speed 1 inertia time (seconds)"** parameter is used to define the extent of the delay between the deactivation of speed 1 and the activation of the new speed. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

When, according to the "fancoil with speed control" algorithm, the device must activate any speed and speed 2 is active, a delay can be inserted between the moment in which the speed 2 deactivation notice is received (or the moment when the speed 2 deactivation command is sent, if fancoil speed info is disabled) and the moment in which the activation command for the new speed is sent; the **"Speed 2 inertia time (seconds)"** parameter is used to define the extent of the delay between the deactivation of speed 1 and the activation of the new speed. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

When, according to the "fancoil with speed control" algorithm, the device must activate any speed and speed 3 is active, a delay can be inserted between the moment in which the speed 3 deactivation notice is received (or the moment when the speed 3 deactivation command is sent, if fancoil speed info is disabled) and the moment in which the activation command for the new speed is sent; the **"Speed 3 inertia time (seconds)"** parameter is used to define the extent of the delay between the deactivation of speed 3 and the activation of the new speed. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

Defining the inertia times is useful for preserving the integrity of the fancoil, because the fact of turning off the power supply to the motor (turning off the actuator) of a fancoil speed does not guarantee that current is no longer circulating in the winding and the instantaneous supply of power to another winding could damage the fancoil (simultaneous powering of multiple windings).

➢ *8.3.3 Heating valve status feedback*

The parameter **"Heating valve status feedback"** is used to enable the device to receive feedback from the actuator that commands the heating solenoid valve; in this way, the device is able to receive the telegram after the solenoid valve switched and to repeat the command if the switching did not take place. The values that can be set are:

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

select **disable** to view the parameter **"Command repetition period with disabled feedback"**; select **enable** to view the communication object ***IN.x - Heating valve status feedback*** (Data Point Type: 1.001 DPT_Switch).
When BUS voltage is restored, the device sends the read request command via the ***IN.x Heating valve status feedback*** object to be updated about the status of the heating solenoid valve.

With feedback enabled, after the device sends the switching command to the solenoid valve, it waits for one minute of its clock for the actuator to send the feedback that switching took place; if this does not take place, it sends the command again to the solenoid valve every minute until it receives the feedback of correct switching. It can happen that, during normal operation of the temperature adjustment, the actuator status can be changed by an entity external of the thermostat function, that forces its status, modifying it. In this case, the device repeats the valve switching command to realign the status of the actuator with the one determined by the control logic of the thermostat function, triggering the process for waiting for confirmation and repeating the command until the confirmation is received. In the same manner, if the control algorithm is operating in heating mode and feedback is received that the air cooling valve is activated, the algorithm is suspended immediately while the command for deactivating the air cooling solenoid valve is sent (triggering the process for waiting for confirmation and repeating the command until the confirmation is received) until the problem is resolved.

With the heating solenoid valve status feedback disabled, it may be useful to cyclically repeat the command to the actuator that manages the solenoid valve so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The **"Command repetition period with disabled feedback"** parameter is used to define the time range of the cyclical sending. The values that can be set are:

- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** **(default value)**

➢ *8.3.4 Fancoil speed status feedback*

If the control algorithm is fancoil, the possibility to receive feedback about the fancoil speed ON status is more important than the valve feedback. By enabling feedback, the device is always aware of the status of the speeds it commands; in fact, if within one minute from sending the command to the actuator that manages a certain speed it does not send confirmation that the command was performed to the thermostat function, it will send the command again every minute until correct confirmation is received from the actuator. As the system does not always have actuators dedicated to the fancoil with mechanically interlocked outputs, the logical interlock function must be implemented on a firmware level which makes it possible to turn on a fancoil speed that is different than what is on only if the correct feedback is received from the latter that it was turned off (providing speed feedback is enabled); as long as the thermostat function does not receive feedback that the active speed was turned off, it will not send the command to turn on the new speed to prevent multiple fancoil windings from being supplied with power at the same time, which would break the fancoil. The **"Fancoil speed status feedback"** parameter enables the device to receive feedback from the actuator that commands the fancoil speeds. The values that can be set are:

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

Select **disable** to view the parameter **"Fancoil speed command repetition period"**; Selecting the value **enable** displays the communication objects ***IN.x - Heating fan V1 status feedback***, ***IN.x - Heating fan V2 status feedback*** *and* ***IN.x - Heating fan V3 status feedback*** (Data Point Type: 1.001 DPT_Switch).
When BUS voltage is restored, the device sends the read request command via the ***IN.x - Heating fan V1 status feedback***, ***IN.x - Heating fan V2 status feedback***, ***IN.x - Heating fan V3 status feedback*** objects to be updated about the activation status of the fancoil speeds.
If fancoil feedback is disabled, deactivation commands of the inactive speeds must be sent for every speed activation command; in the same manner, every speed deactivation command must be sent together with deactivation commands for the other speeds.

The parameter **"Fancoil speed command repetition period "** is used to define the time range of the cycling sending to the fancoil speeds; The values that can be set are:

- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** **(default value)**

By selecting any value other than **no repetition**, the commands are repeated on all speed communication objects.

8.4 "Thermostat x – Air cooling" menu

The **"Thermostat x - Air cooling"** menu contains the characteristic parameters of the load control algorithms for the air cooling system.

The structure of the menu is as follows:

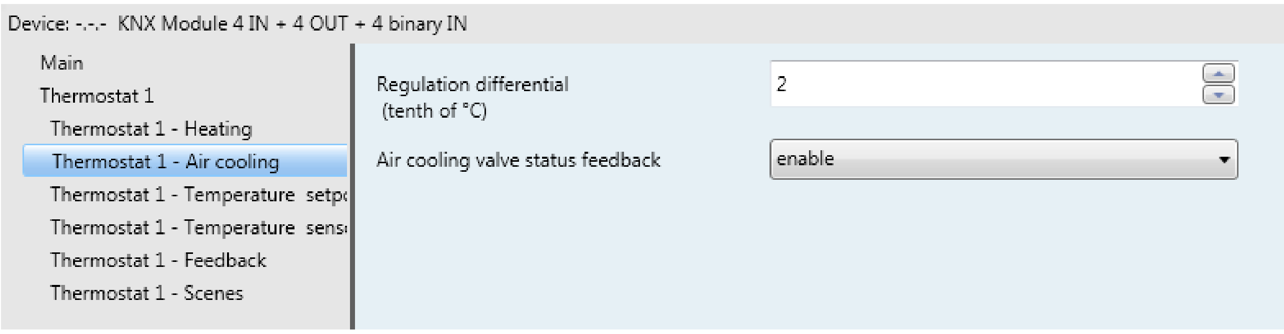


Fig. 8.3: "Thermostat x – Air cooling" menu

The Thermostat x - Air cooling menu varies based on what is set in the parameters "**Control logic Heating/Air cooling"** and "**Air cooling control algorithm**" in the "**Thermostat 1**" page.

All the possible parameters that can be displayed in this menu based on the system type are summarised below.

Parameters

➢ 8.4.1 Regulation differential (tenth of °C)

The parameter **“Regulation differential (tenth of °C)”** is used to set the value of the regulation differential value of the air cooling **two points ON-OFF** control algorithm, already mentioned in paragraph 10.2. Control algorithms, which, when summed with the value of the setpoint that was set, determines the threshold value above which the air cooling system is turned on in the 2 points control. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

➢ 8.4.2 Select air cooling system

The parameter **“Select air cooling system”** is used to automatically measure the operating parameters (Proportional band and Integration time) of the proportional integral algorithm based on the selected air cooling system. The values that can be set are:

- **ceiling air cooling (default value)**
- fancoil unit
- customised

Selecting **ceiling cooling**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **5.0 °C** and **240** are displayed.

Selecting **fan coil unit**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **4.0 °C** and **90** are displayed.

Selecting **customised**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified.

The parameter **“Proportional band”** is used to set the width of the PWM proportional band of the air cooling **PWM proportional integral**, already cited in paragraph 10.2. Control algorithms, which, when summed with the value of the setpoint that was set, determines the upper limit of the proportional band used for the proportional integral control. The values that can be set are:

- 1.0 °C
- 1.5 °C
- **2.0 °C (default value)**
- 2.5 °C
- 3.0 °C
- 3.5 °C
- 4.0 °C
- 4.5 °C
- 5.0 °C
- 5.5 °C
- 6.0 °C
- 6.5 °C
- 7.0 °C
- 7.5 °C
- 8.0 °C
- 8.5 °C
- 9.0 °C
- 9.5 °C
- 10.0°C

The parameter **“Integration time (minutes)”** is used to set the contribution of the integral action in the proportional integral control (See paragraph 10.2 Control algorithms). The values that can be set are:

- from 1 minute to 250 minutes with steps of 1, plus the value "no integral" (255) **(default value 60)**

Selecting **no integral**, the integral component is zero and the pure effect of proportional control is obtained.

The **"Cycle time"** parameter is used to set the value of the period within which the device carries out PWM modulation, modifying the duty-cycle. The values that can be set are:

- 5 minutes
- 10 minutes
- 15 minutes
- **20 minutes** **(default value)**
- 30 minutes
- 40 minutes
- 50 minutes
- 60 minutes

If the control algorithm is fancoil, the format of the air cooling solenoid valve commands (4-way system) is independent of that of the fancoil speed control; the solenoid valve control logic when the selected algorithm is fancoil is **2 points ON-OFF**. Via the communication object ***IN.x - Air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) the device sends the command telegrams to the air cooling solenoid valve.

The parameter **"Valve regulation differential (tenth of °C)"** is used to set the regulation differential value of the 2 points control of the fancoil operating solenoid valve, as mentioned in paragraph 10.2 Control algorithms. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

The parameter **"Speed 1 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the first speed of the air cooling control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control**, as mentioned in paragraph 10.2 Control algorithms; This value, added to the "setpoint+ΔT_{valv}" value, determines the threshold value below which speed 1 of the fancoil is turned on. The values that can be set are:

- from 0 to 20 with steps of 1 **(default value 2)**

setting the value **0** obtains the condition "$\Delta T_{1\ cool}=\Delta T_{valv}$" for which the value of the speed 1 activation threshold is "setpoint+ΔT_{valv}" and the off value is "setpoint".

The parameter **"Speed 2 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the second speed of the air cooling control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control**, as mentioned in paragraph 10.2 Control algorithms; This value, subtracted from the "setpoint+$\Delta T_{valv}+\Delta T_{1\ cool}$" value, determines the threshold value below which speed 2 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

The parameter **"Speed 3 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the third speed of the air cooling control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control**, as mentioned in paragraph 10.2 Control algorithms; This value, subtracted from the "setpoint+$\Delta T_{valv}+\Delta T_{1\ cool}+\Delta T_{2\ cool}$" value, determines the threshold value below which speed 3 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 with steps of 1, **2 (default value)**

When, according to the "fancoil with speed control" algorithm, the device must activate any speed and speed 1 is active, a delay can be inserted between the moment in which the speed 1 deactivation notice is received (or the moment when the speed 1 deactivation command is sent, if fancoil speed info is disabled) and the moment in which the activation command for the new speed is sent; the **"Speed 1 inertia time (seconds)"** parameter is used to define the extent of the delay between the deactivation of speed 1 and the activation of the new speed. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

When, according to the "fancoil with speed control" algorithm, the device must activate any speed and speed 2 is active, a delay can be inserted between the moment in which the speed 2 deactivation notice is received (or the moment when the speed 2 deactivation command is sent, if fancoil speed info is disabled) and the moment in which the activation command for the new speed is sent; the **"Speed 2 inertia time (seconds)"** parameter is used to define the extent of the delay between the deactivation of speed 1 and the activation of the new speed. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

When, according to the "fancoil with speed control" algorithm, the device must activate any speed and speed 3 is active, a delay can be inserted between the moment in which the speed 3 deactivation notice is received (or the moment when the speed 3 deactivation command is sent, if fancoil speed info is disabled) and the moment in which the activation command for the new speed is sent; the **"Speed 3 inertia time (seconds)"** parameter is used to define the extent of the delay between the deactivation of speed 3 and the activation of the new speed. The values that can be set are:

- from **0 (default value)** to 10, with steps of 1

➢ *8.4.3 Air cooling valve status feedback*

The parameter **"Air cooling valve status feedback"** is used to enable the device to receive feedback from the actuator that commands the air cooling solenoid valve; in this way, the device is able to receive the telegram after the solenoid valve switched and to repeat the command if the switching did not take place. The values that can be set are:

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

Select **disable** to view the parameter **"Command repetition period with disabled feedback"**; selecting **enable** displays the communication object ***IN.x - Air cooling valve status feedback*** (Data Point Type: 1.001 DPT_Switch).
When BUS voltage is restored, the device sends the read request command via the ***IN.x Air cooling valve status feedback*** object to be updated about the status of the air cooling solenoid valve.

With feedback enabled, after the device sends the switching command to the solenoid valve, it waits for one minute of its clock for the actuator to send the feedback that switching took place; if this does not take place, it sends the command again to the solenoid valve every minute until it receives the feedback of correct switching. It can happen that, during normal operation of the temperature adjustment, the actuator status can be changed by an entity external of the thermostat function, that forces its status, modifying it. In this case, the device repeats the valve switching command to realign the status of the actuator with the one determined by the control logic of the thermostat function, triggering the process for waiting for confirmation and repeating the command until the confirmation is received. In the same manner, if the control algorithm is operating in air cooling mode and feedback is received that the heating valve is activated, the algorithm is suspended immediately while the command for deactivating the heating solenoid valve is sent (triggering the process for waiting for confirmation and repeating the command until the confirmation is received) until the problem is resolved.

With the air cooling solenoid valve status feedback disabled, it may be useful to cyclically repeat the command to the actuator that manages the solenoid valve so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The parameter **"Command repetition period with disabled feedback"** is used to define the time range of the cyclical sending.

The values that can be set are:

- no repetition

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

➢ *8.4.4 Fancoil speed status feedback*

If the control algorithm is fancoil, the possibility to receive feedback about the fancoil speed ON status is more important than the valve feedback. By enabling feedback, the device is always aware of the status of the speeds it commands; in fact, if within one minute from sending the command to the actuator that manages a certain speed it does not send confirmation that the command was performed to the thermostat function, it will send the command again every minute until correct confirmation is received from the actuator. As the system does not always have actuators dedicated to the fancoil with mechanically interlocked outputs, the logical interlock function must be implemented on a firmware level which makes it possible to turn on a fancoil speed that is different than what is on only if the correct feedback is received from the latter that it was turned off (providing speed feedback is enabled); as long as the thermostat function does not receive feedback that the active speed was turned off, it will not send the command to turn on the new speed to prevent multiple fancoil windings from being supplied with power at the same time, which would break the fancoil. The **"Fancoil speed status feedback"** parameter enables the device to receive feedback from the actuator that commands the fancoil speeds.

The values that can be set are:

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

Selecting **disable** displays the parameter **"Fancoil speed command repetition period"** and the communication objects ***IN.x - Air cooling fan V1 status feedback***, ***IN.x - Air cooling fan V2 status feedback*** and ***IN.x - Air cooling fan V3 status feedback*** (Data Point Type: 1.001 DPT_Switch).
When BUS voltage is restored, the device sends the read request command via the ***IN.x - Air cooling fan V1 status feedback***, ***IN.x - Air cooling fan V2 status feedback***, ***IN.x - Air cooling fan V3 status feedback*** objects to be updated about the activation status of the fancoil speeds.

The parameter **"Fancoil speed command repetition period "** is used to define the time range of the cycling sending to the fancoil speeds; The values that can be set are:

- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** **(default value)**

By selecting any value other than **no repetition**, the commands are repeated on all speed communication objects.

8.5 “Thermostat x - Temperature setpoint” menu

The **Thermostat x - Temperature setpoint** menu contains the parameters used to configure the setpoint values of the various temperature adjustment modes of the two functioning types.

The structure of the menu is as follows:

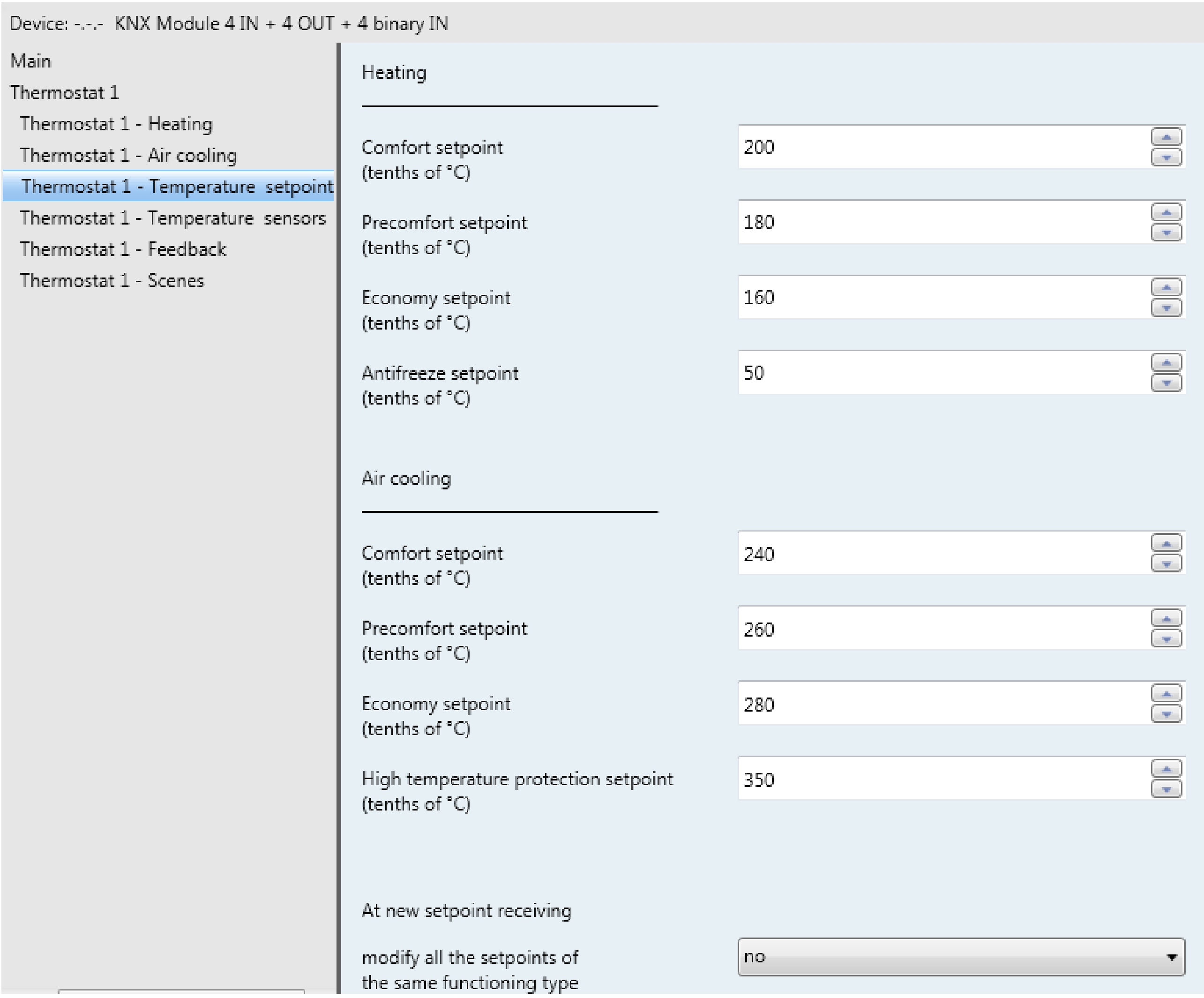


Fig. 8.4: “Thermostat x - Temperature setpoint” menu

Parameters - Heating

➢ 8.5.1 Comfort setpoint (tenths of °C)

The parameter **“Comfort setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the COMFORT mode for HEATING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 200)**

Remember that when setting this value, there is a constraint that it must be higher than the value set for the **“Precomfort setpoint (tenth of °C)”** for heating operation.

Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ *8.5.2 Precomfort setpoint (tenths of °C)*

The parameter **“Precomfort setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the PRECOMFORT mode for HEATING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 180)**

Remember that when setting this value, there is a constraint that it must lie between the value set for the **“Comfort setpoint (tenth of °C)”** and the value set for the **“Economy setpoint (tenth of °C)”**for heating operation.
Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ *8.5.3 Economy setpoint (tenths of °C)*

The parameter **“Economy setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the ECONOMY mode for HEATING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 160)**

Remember that when setting this value, there is a constraint that it must lie between the value set for the **“Precomfort setpoint (tenth of °C)”** and the value set for the **“Anti-freeze setpoint (tenth of °C)”**for heating operation.
Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ *8.5.4 Operating setpoint (tenths of °C)*

The parameter **“Operating setpoint (tenth of °C)”**, which is visible if the remote control type is setpoint, is used to set the operating setpoint value for HEATING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 200)**

Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ *8.5.5 Anti-freeze setpoint (tenths of °C)*

The parameter **“Antifreeze setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the OFF mode for HEATING operation; the values that can be set are:

- from 20 to 70 with steps of 1 **(default value 50)**

Remember that when setting this value, there is a constraint that it must be lower than the value set for the **“Economy setpoint (tenth of °C)”** for heating operation.
Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ 8.5.6 Anti-freeze setpoint for window contact (tenths of °C)

The parameter **“Antifreeze setpoint for window contact (tenths of °C)”**, which is visible if the remote control type is setpoint, is used to set the operating setpoint value for HEATING operation when the device is switched off from the window contact function; The values that can be set are:

- from 20 to 70 with steps of 1 **(default value 50)**

Parameters - Air cooling

➢ 8.5.7 Comfort setpoint (tenths of °C)

The parameter **“Comfort setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the COMFORT mode for AIR COOLING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 240)**

Remember that when setting this value, there is a constraint that it must be lower than the value set for the **“Precomfort setpoint (tenth of °C)”** for air cooling operation.
Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ 8.5.8 Precomfort setpoint (tenths of °C)

The parameter **“Precomfort setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the PRECOMFORT mode for AIR COOLING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 260)**

Remember that when setting this value, there is a constraint that it must lie between the value set for the **“Comfort setpoint (tenth of °C)”** and the value set for the **“Economy setpoint (tenth of °C)”**for air cooling operation.
Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ 8.5.9 Economy setpoint (tenths of °C)

The parameter **“Economy setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the ECONOMY mode for AIR COOLING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 280)**

Remember that when setting this value, there is a constraint that it must lie between the value set for the **“Precomfort setpoint (tenth of °C)”** and the value set for the **“High temperature protection setpoint (tenth of °C)”**for air cooling operation.
Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ *8.5.10 Operating setpoint (tenths of °C)*

The parameter **“Operating setpoint (tenth of °C)”**, which is visible if the remote control type is setpoint, is used to set the operating setpoint value for AIR COOLING operation; The values that can be set are:

- from 50 to 400 with steps of 1 **(default value 240)**

Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ *8.5.11 High temperature protection setpoint (tenths of °C)*

The parameter **“High temperature protection setpoint (tenth of °C)”**, which is visible if the remote control type is HVAC mode, is used to set the setpoint value of the OFF mode for AIR COOLING operation; The values that can be set are:

- from 300 to 400 with steps of 1 **(default value 350)**

Remember that when setting this value, there is a constraint that it must be higher than the value set for the **“Economy setpoint (tenth of °C)”** for air cooling operation.
Remember that this value can be modified via the BUS telegram on the communication object assigned to it.

➢ *8.5.12 High temperature protection setpoint for window contact (tenths of °C)*

The parameter **“High temperature protection setpoint for window contact (tenths of °C)”**, which is visible if the remote control type is setpoint, is used to set the operating setpoint value for AIR COOLING operation when the device is switched off from the window contact function; The values that can be set are:

- from 300 to 400 with steps of 1 **(default value 350)**

Parameters - At new setpoint receiving

➢ *8.5.13 At new setpoint receiving modify contemporary the setpoint of the other functioning type*

By modifying the setpoint of a particular HVAC mode of a functioning type (if the remote control type is HVAC mode) or operating setpoint (if the remote control type is setpoint), it may be useful to modify in the same manner the setpoint of the same mode for the opposite functioning type .

EXAMPLE: Heating Comfort setpoint = 20 °C and Air cooling Comfort setpoint = 24 °C; if a Heating Comfort setpoint value equal to 21.5 °C is received, also the Air cooling Comfort setpoint is automatically modified and set equal to 25.5 °C.

The parameter that makes it possible to enable the contemporary modification of the same mode for the two different functioning types is **“At new setpoint receiving, modify all setpoints of the other functioning type”.** The possible values are:

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

➢ *8.5.14 At new setpoint receiving, modify the setpoints of the same functioning type*

In the same manner, if the remote control type is HVAC mode, may be useful to modify manner the setpoints of the same functioning type (excluding the OFF mode) after only one of them has been modified. Changing the OFF setpoint does not update the setpoints of the other modes.
EXAMPLE: Heating Comfort setpoint = 20 °C, Heating Precomfort setpoint = 18 °C and Heating Economy setpoint = 16 °C; if a Heating Comfort setpoint equal to 21.5 °C is received, then automatically the Heating Precomfort setpoint becomes 19.5 °C and the Heating Economy setpoint becomes 17.5 °C.

The parameter that makes it possible to enable the contemporary modification of the setpoints for the same functioning type is **"At new setpoint receiving, modify all setpoints of the same functioning type"**. The possible values are:

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

If both modifications are enabled, when a setpoint is modified, as a result also all the other setpoints of the other modes will be modified, both for heating and for air cooling.

8.6 *“Thermostat x – Temperature sensors” menu*

The **Thermostat x - Temperature sensors** menu contains the parameters used to configure the operation of the external NTC sensor connected to the universal input.

The structure of the menu is as follows:

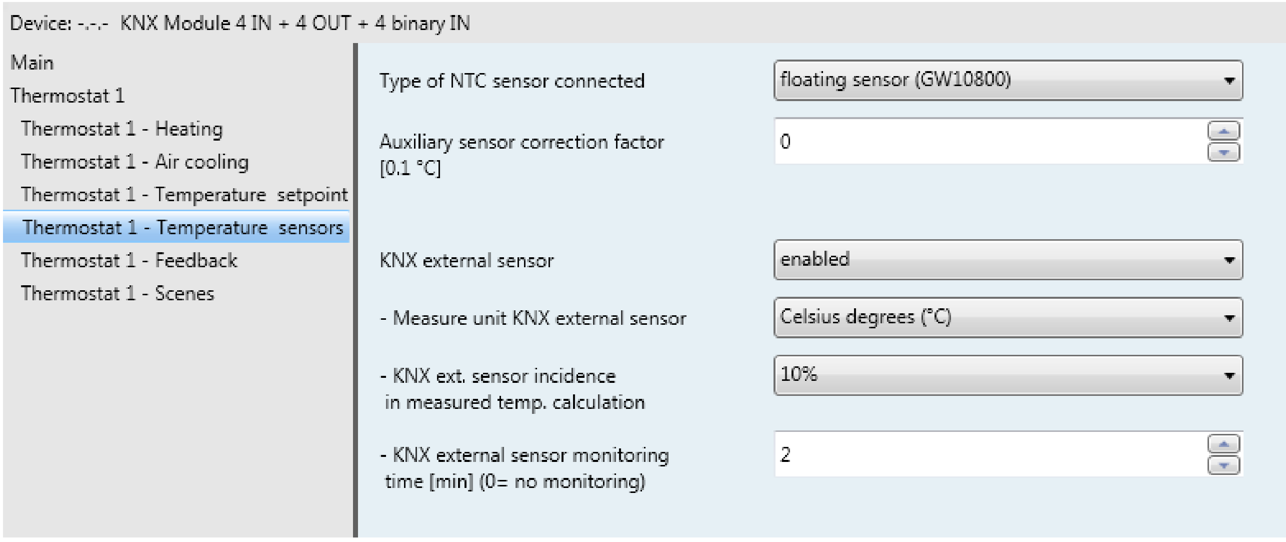


Fig. 8.5: “Thermostat x – Temperature sensors” menu

Parameters

➢ *8.6.1 Type of NTC sensor connected*

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

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

➢ *8.6.2 Auxiliary sensor correction factor (tenths of °C]*

The parameter **“Auxiliary sensor correction factor [°C]”** is used to set the correction factor to be applied to the temperature value of the NTC sensor connected to the input, to eliminate the heat contribution generated by the installation site; The values that can be set are:

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

An NTC sensor malfunction or failed connection is signalled by the amber LED associated with the input (blinking at 8 Hz for 3 seconds and switched off for 3 seconds) and all the other input functions are interrupted.

8.6.3 KNX external sensor

The parameter **"KNX external sensor"** is used to enable a communication object for measuring the room temperature and as a result the configuration items; The values that can be set are:

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

Selecting **enabled** displays the parameters **"Measure unit KNX external sensor"**, **"KNX external sensor incidence in the measured temperature calculation"**, **"KNX external sensor monitoring time [min] (0=no monitoring)"** and **"Behaviour at KNX external sensor signal absence"** and the communication object ***IN.x - KNX external sensor input*** which is used to receive the temperature measured by the external sensor.
At BUS voltage recovery, the device must immediately update the value received from the KNX temperature sensor, sending the read request via the object ***IN.x - KNX external sensor input*** and storing the received value.

The parameter **"Measure unit KNX external sensor"** is used to set the measurement unit used for decoding the information received via the communication object ***IN.x - KNX external sensor input***; 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 ***IN.x - KNX external sensor input***: *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)**.

Once the KNX external sensor is enabled, the measured temperature will not only be determined by the sensor connected to the universal input of the device, but it will be determined by the weighted average between the value measured by the sensor connected to the device and the value measured by the KNX external sensor. The parameter **"KNX external sensor incidence in measured temperature calculation"** is used to determine the incidence of the value measured by the KNX external sensor when calculating the measured temperature, which ranges from a minimum of 10% to a maximum of 100% (external sensor measured value = measured temperature).

The complete formula for the temperature calculation is:

$T_{measured}= T_{external\ sensor} x\ Incidence_{external\ sensor}+T_{connected\ sensor}\ x\ (100\%-Incidence_{external\ sensor})$

The parameter may assume the following values:

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

The **"KNX external sensor monitoring time [min] (0=no monitoring)"** parameter is used to define the monitoring time of the KNX external sensor and can assume the following values:

- from 0 to 10 with steps of 1 **(default value 2)**

By selecting the value **0**, the object enabled for the input will not be monitored. The meaning of the monitoring time is: if, within the set monitoring time, the telegram with the measured value is not received periodically, the contribution of the KNX sensor in the calculation of the measured temperature will be excluded until a new value is received. If the incidence of the external sensor KNX was 100%, at the end of the monitoring time the control algorithm operation; this malfunction is signalled by the amber LED associated with the input (blinking at 8 Hz for 3 seconds and switched off for 3 seconds).

8.7 “Thermostat x - Feedback” menu

The **Thermostat x - Feedback** menu contains the parameters used to set the conditions for sending the feedback that the device sends via BUS telegrams.

The structure of the menu is as follows:

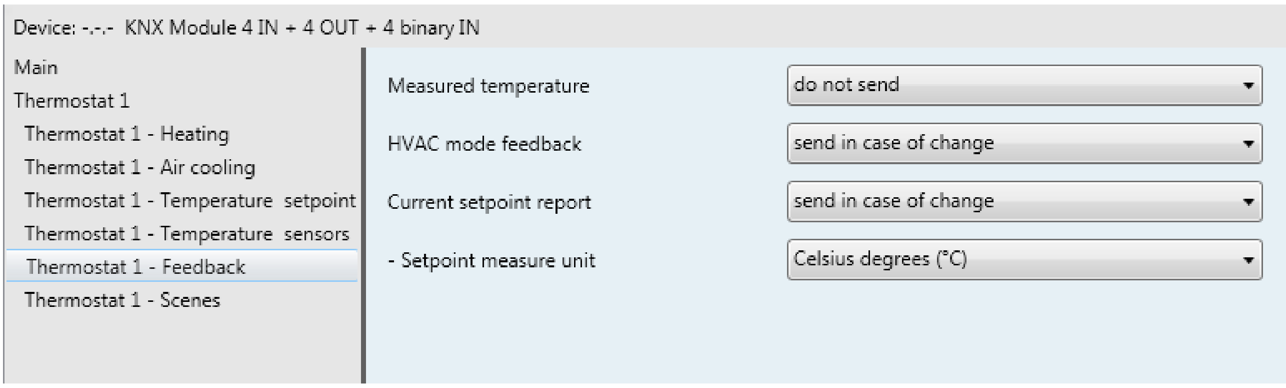


Fig. 8.6: “Thermostat x - Feedback” menu

Parameters

➢ 8.7.1 Measured temperature

The parameter **“Measured temperature”**, is used to define the conditions for sending the value of the temperature measured by the NTC sensor connected to the universal input; The values that can be set are:

- **do not send** **(default value)**
- send on demand only
- send in case of change
- send periodically
- send on change and periodically

Selecting any value other than **do not send** displays the communication object ***IN.x - Measured temperature*** (Data Point Type: 9.001 DPT_Temp) and the parameter **“Measure unit”**.

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

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

The **“Measure unit”** parameter is used to set the measurement unit for coding and sending the information via the ***IN.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 ***IN.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)**.

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

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

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

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

If the measured temperature depends only by the external KNX sensor (100% incidence), the device does not send any feedback until it is received at least one telegram from the external sensor, regardless of the sending conditions setting.

➢ *8.7.2 HVAC mode feedback*

The parameter **"HVAC mode feedback"**, which is visible if the remote control type is HVAC mode, is used to enable and set the conditions for sending HVAC mode feedback via the communication object ***IN.x - HVAC mode feedback*** (Data Point Type: 20.102 DPT_HVACMode).
The values that can be set are:

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

selecting **send on demand only**, the HVAC mode feedback will not be sent spontaneously by the device via the communication object ***IN.x - HVAC mode feedback***; 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 HVAC mode set on the device. Selecting **send in case of change**, the HVAC mode feedback will be sent spontaneously by the device via the communication object ***IN.x - HVAC mode feedback***, each time the mode is changed. After a BUS voltage recovery, feedback about the active mode should be sent in order to update any connected devices.

➢ *8.7.3 Current setpoint feedback*

The parameter **"Current setpoint report"** is used to enable and set the conditions for sending the feedback regarding the current setpoint value set on the device via a BUS telegram on the communication object ***IN.x - Current setpoint feedback***. The values that can be set are:

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

selecting a value other than **disabled** displays the communication object ***IN.x - Current setpoint feedback*** (Data Point Type: 9.001 DPT_Temp) and the parameter **"Setpoint measure unit"**. Selecting **send on demand only**, the feedback regarding the active setpoint on the device will not be sent spontaneously by the device via the communication object ***IN.x - Current setpoint feedback***; 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 setpoint set on the device. Selecting **send in case of change**, feedback regarding the active setpoint on the device is sent spontaneously by the device via the communication object ***IN.x - Current setpoint feedback***, each time there is a change in the setpoint (also following temporary forcing). After a BUS voltage recovery, feedback about the active setpoint should be sent in order to update any connected devices.

The **“Setpoint measure unit”** parameter is used to set the measurement unit for coding and sending the information via the ***IN.x – Current setpoint feedback*** 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 ***IN.x - Current setpoint 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)**.

8.8 “Thermostat x - Scenes” menu

The scene function is used to replicate a certain previously memorised condition upon receipt of the scene execution command.

The structure of the menu is as follows:

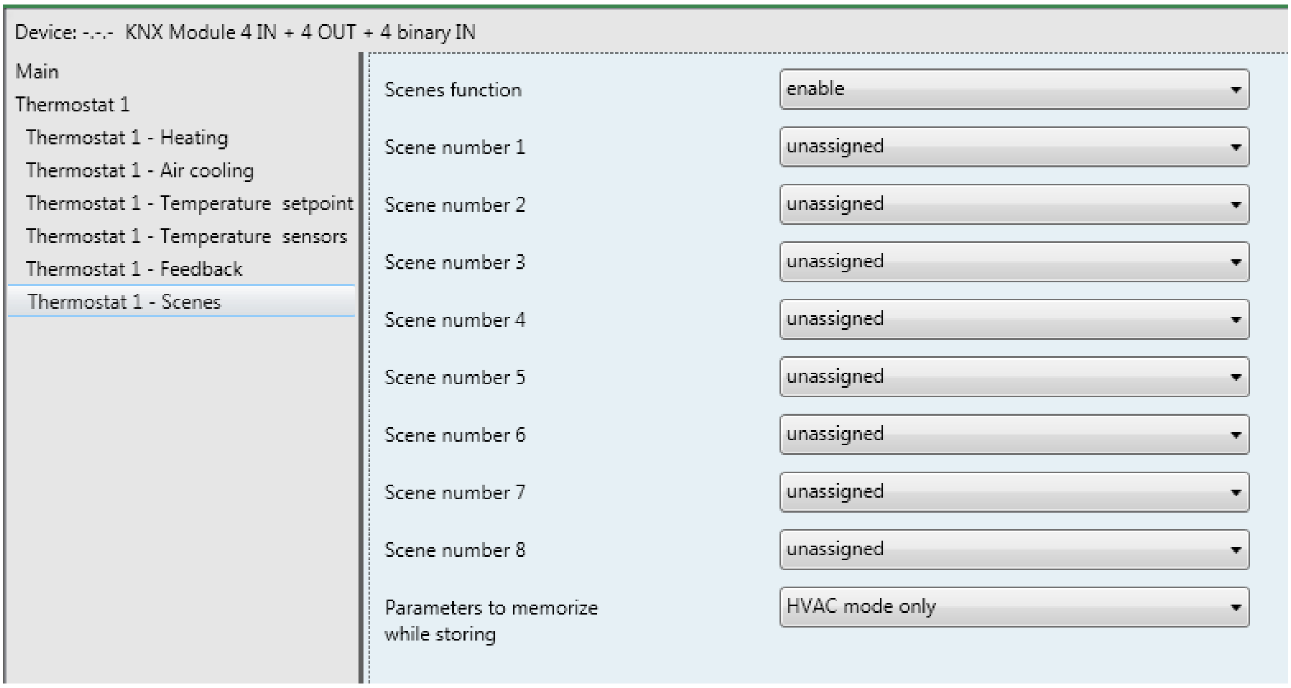


Fig. 8.7: “Thermostat x - Scenes” menu

Parameters

➢ 8.8.1 Scene function

The parameter **“Scenes function”** is used to activate and configure the function, displaying the various function configuration parameters and the relative communication object ***IN.x - Thermostat scene function*** (Data Point Type: 18.001 DPT_SceneControl).
The scene function is used to send two possible commands to the device:

– execute scene, which is a command to assume a determined condition
– store scene, which is a command to memorise the current status (the moment the command is received) of the different functional parameters of the device defined in the configuration phase.

This function provides 8 scenes, for which the device can store/reproduce 8 different conditions of these functional parameters. The values that can be set are:

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

selecting **enable** displays the parameters **“Scene number 1”**, **“Scene number 2”**, **“Scene number 3”**, **“Scene number 4”**, **“Scene number 5”**, **“Scene number 6”**, **“Scene number 7”**, **“Scene number 8”** and **“Parameters to memorize while storing”** and the communication object ***Thermostat scene***, through which the scene execution/memorise telegrams are received.

➢ *8.8.2 Scene number i*

With the **"Scene number i"** (1 ≤ i ≤ 8) parameters you can set the numerical value for identifying and therefore executing/memorising the i-th scene. The possible values are:

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

➢ *8.8.3 Parameters to memorize while storing*

As the thermostat has various operating parameters that can change during its operation, the parameter **"Parameters to memorize while storing"** can be used to configure which of them should be memorised while storing the scene, to then be replicated following an execution command. The values that can be set, if the remote control type is "HVAC mode", are:

- **HVAC mode only** **(default value)**
- HVAC mode and functioning type

The values that can be set, if the remote control type is "setpoint", are:

- **setpoint only** **(default value)**
- setpoint and functioning type

9 *“Digital output x”* menu

A dedicated menu is displayed for each enabled digital output that is called **Digital output x** (x = 1 .. 4, is the output index). All 4 menus are identical and therefore, for the sake of simplicity, the operation and dedicated parameters are summarised in this paragraph, indicating the reference menu with a generic “X” (1 .. 4).

The basic structure of the menu is as follows:

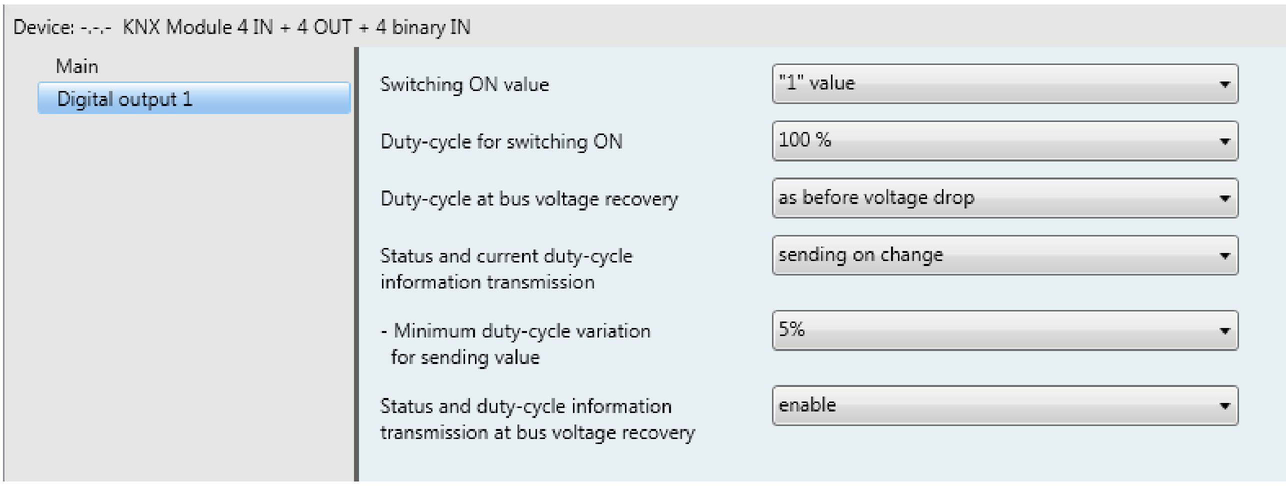


Fig. 9.1: “Digital output x” menu

The function is used to configure the output for controlling a PWM digital output.
The digital output can be regulated via the communication object ***CH.x - Digital output % command*** (Data Point Type: 5.001 DPT_Scaling) which permits percentage regulation by modifying the duty-cycle of the output signal.

9.1 Parameters

➢ *9.1.1 Switching ON value*

The on/off switching of the digital output can be implemented via the communication object ***CH.x - Digital output switching*** (Data Point Type 1.001 DPT_Switch). The parameter **“Switching ON value”** determines which logic value received on the communication object ***CH.x - Digital output switching*** switches the digital output to the ON status; The possible values are:

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

➢ *9.1.2 Duty-cycle for switching ON*

When the ON value is received, the device switches the duty-cycle of the digital output to the value set via the parameter **“Duty-cycle for switching ON**” which can have the following values:

- 5% 5
- 10% 10
- 15% 15
- …
- **100%** **100 (default value)**

➢ 9.1.3 Duty-cycle at BUS voltage recovery

The parameter **“Duty-cycle at bus voltage recovery”** is used to set the duty-cycle of the digital output at BUS power supply voltage recovery. The values that can be set are:

- 0% (off)
- 5%
- 10%
- 15%
- …
- 100%
- **as before voltage drop (default value)**

➢ 9.1.4 Status and current duty-cycle information transmission

The on/off status of the digital output could be transmitted on the BUS via the communication object ***CH.x - Digital output status*** (Data Point Type 1.001 DPT_Switch); the feedback value is 1 = ON when the duty-cycle is >0%, and the value 0 = OFF when the duty-cycle is 0%.
The current value of the digital output duty-cycle can be transmitted on the BUS via the communication object ***CH. x - Digital output % value*** (Data Point Type 5.001 DPT_Scaling).

The parameter used to enable transmitting this information is **“Status and current duty-cycle information transmission”**, which can have the following values:

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

Selecting any value other than **disabled** displays the communication object ***CH.x - Digital output status*** and the object ***CH. x - Digital output % value***.

If the set value is **on demand only**, the status and value of the duty-cycle are never sent spontaneously by the device and the device only sends the response telegram with the requested value upon receipt of a BUS status read request.

If the status is signalled **on change**, the status object is sent spontaneously when the status changes from ON to OFF or vice versa whereas the current duty-cycle value is sent if the new value, with respect to the last value sent, has had a variation greater than the minimum set for the parameter **“Minimum duty-cycle variation for sending value”**, which can have the following values:

- 1%
- 2%
- **5% (default value)**
- 10%
- 15%
- 25%

If the status information is sent **on change**, the **“Status and current duty-cycle transmission on bus voltage recovery“** parameter is visualised; this enables the transmission of the current status and duty-cycle information of the digital output when the BUS voltage is restored. This parameter may have the following values:

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

10 Communication objects

The communication objects are divided according to the function implemented by the device.

10.1 Communication objects for "Binary input x"

Communication objects with **output** functions:

#								Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4	IN 5	IN 6	IN 7	IN 8				
1	33	65	97	129	154	179	204	IN.j - Switch	On/Off	Dimmer switching on/off commands	1.001 DPT_Switch
1	33	65	97	129	154	179	204	IN.j - Shutter movement	Up/Down	Moves up/down the shutter	1.008 DPT_UpDown
1	33	65	97	129	154	179	204	IN.j - Scene	Execute/Store	Send learn/execute scene commands	18.001 DPT_SceneControl
1	33	65	97	129	154	179	204	IN.j - A Sequence	On/Off	Send On/Off commands associated to object A of the sequence	1.001 DPT_Switch
1	33	65	97	129	154	179	204	IN.j - Object A 1 bit value	1/0 value	Sends values 1/0 associated with object A	1.002 DPT_Bool
1	33	65	97	129	154	179	204	IN.j - Primary counter	1 byte unsigned value	Send unsigned value (0..255) of the primary counter	5.010 DPT_Value_1_Ucount
1	33	65	97	129	154	179	204	IN.j - Primary counter	1 byte signed value	Send signed value (-128..127) of the primary counter	6.010 DPT_Value_1_Count
1	33	65	97	129	154	179	204	IN.j - Primary counter	2 byte unsigned value	Send unsigned value (0..65535) of the primary counter	7.001 DPT_Value_2_Ucount
1	33	65	97	129	154	179	204	IN.j - Primary counter	2 byte signed value	Send signed value (-32768..32767) of the primary counter	8.001 DPT_Value_2_Count
1	33	65	97	129	154	179	204	IN.j - Primary counter	4 byte unsigned value	Send unsigned value (0.. 4294967295) of the primary counter	12.001 DPT_Value_4_Ucount
1	33	65	97	129	154	179	204	IN.j - Primary counter	4 byte signed value	Send signed value (-2147483648.. 2147483647) of the primary counter	13.001 DPT_Value_4_Count
1	33	65	97	129	154	179	204	IN.j - A object 2 bits value	On/Off forced positioning	Send priority commands associated to object A of the sequence	1.002 DPT_Switch_Control
1	33	65	97	129	154	179	204	IN.j - A object 1 byte value	Unsigned value	Send unsigned values (0..255) associated to object A	5.010 DPT_Value_1_Ucount
1	33	65	97	129	154	179	204	IN.j - A object 1 byte value	Signed value	Send signed values (-128..127) associated to object A	6.010 DPT_Value_1_Count
1	33	65	97	129	154	179	204	IN.j - A object 1 byte value	% value	Send percent values (0%..100%) associated to object A	5.001 DPT_Scaling
1	33	65	97	129	154	179	204	IN.j - A object 1 byte value	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated to object A	20.102 DPT_HVACMode
1	33	65	97	129	154	179	204	IN.j - A object 2 bytes value	Unsigned value	Send unsigned values (0..65535) associated to object A	7.001 DPT_Value_2_Ucount
1	33	65	97	129	154	179	204	IN.j - A object 2 bytes	Signed value	Send signed values (-	8.001

								value		32768..32767) associated to object A	DPT_Value_2_Count
1	33	65	97	129	154	179	204	IN.j - A object 3 byte value	RGB color	Send RGB color components value associated to object A	232.600 DPT_Colour_RGB
1	33	65	97	129	154	179	204	IN.j - A object 4 bytes value	Unsigned value	Sends unsigned values (0.. 4294967295) associated to object A	12.001 DPT_Value_4_Ucount
1	33	65	97	129	154	179	204	IN.j - A object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated to object A	13.001 DPT_Value_4_Count
1	33	65	97	129	154	179	204	IN.j - A object 14 bytes value	Characters ISO 8859-1	Send characters codified with ISO 8859-1 standard associated to object A	16.001 DPT_String_8859_1
1	33	65	97	129	154	179	204	IN.j - Single press 1 bit A object	1/0 value	Send 1/0 values associated to object A of the single press	1.002 DPT_Bool
1	33	65	97	129	154	179	204	IN.j - Single press 2 bit A object	On/Off forced positioning	Send priority commands associated to object A of the single press	1.002 DPT_Switch_Control
1	33	65	97	129	154	179	204	IN.j - Single press 1 byte A object	Unsigned value	Send unsigned values (0..255) associated to object A of the single press	5.010 DPT_Value_1_Ucount
1	33	65	97	129	154	179	204	IN.j - Single press 1 byte A object	Signed value	Send signed values (-128..127) associated to object A of the single press	6.010 DPT_Value_1_Count
1	33	65	97	129	154	179	204	IN.j - Single press 1 byte A object	% value	Send percent values (0%..100%) associated to object A of the single press	5.001 DPT_Scaling
1	33	65	97	129	154	179	204	IN.j - Single press 1 byte A object	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated to object A of single press	20.102 DPT_HVAC Mode
1	33	65	97	129	154	179	204	IN.j - Single press 2 byte A object	Unsigned value	Send unsigned values (0..65535) associated to object A of the single press	7.001 DPT_Value_2_Ucount
1	33	65	97	129	154	179	204	IN.j - Single press 2 byte A object	Signed value	Send signed values (-32768..32767) associated to object A of the single press	8.001 DPT_Value_2_Count
1	33	65	97	129	154	179	204	IN.j - Single press 3 byte A object	RGB color	Send RGB color components value associated to object A of the single press	232.600 DPT_Colour_RGB
1	33	65	97	129	154	179	204	IN.j - Single press 4 byte A object	Unsigned value	Sends unsigned values (0.. 4294967295) associated to object A of the single press	12.001 DPT_Value_4_Ucount
1	33	65	97	129	154	179	204	IN.j - Single press 4 byte A object	Signed value	Sends signed values (-2147483648.. 2147483647) associated to object A of the single press	13.001 DPT_Value_4_Count
2	34	66	98	130	155	180	205	IN.j - Brightness dimming	Increase/Decrease	Dimmer brightness increasing/decreasing commands	3.007 DPT_Control_Dimming
2	34	66	98	130	155	180	205	IN.j - Shutter stop/Louvres control	Stop/Step	Send stop movement/slat regulation commands	1.007 DPT_Step
2	34	66	98	130	155	180	205	IN.j - B sequence	On/Off	Send On/Off commands associated to object B	1.001 DPT_Switch

										of the sequence	
2	34	66	98	130	155	180	205	IN.j - Primary counter bit overflow	Overflow status	Sends the primary counter overflow feedback	1.002 DPT_Bool
2	34	66	98	130	155	180	205	IN.j - Primary counter bit overflow	Overflow status	Send the primary counter overflow feedback	1.002 DPT_Bool
2	34	66	98	130	155	180	205	IN.j - Single press 1 bit B object	1/0 value	Send 1/0 values associated to object B of the single press	1.002 DPT_Bool
3	35	67	99	131	156	181	206	IN.j - C sequence	On/Off	Send On/Off commands associated to object C of the sequence	1.001 DPT_Switch
3	35	67	99	131	156	181	206	IN.j - Primary counter byte overflow	Overflow status	Send the value associated to the primary counter overflow feedback	5.010 DPT_Value_1_Ucount
3	35	67	99	131	156	181	206	IN.j - C object 1 bit value	1/0 value	Send 1/0 values associated to object C	1.002 DPT_Bool
3	35	67	99	131	156	181	206	IN.j - Single press 1 bit C object	1/0 value	Send 1/0 values associated to object C of the single press	1.002 DPT_Bool
4	36	68	100	132	157	182	207	IN.j - D sequence	On/Off	Send On/Off commands associated to object D of the sequence	1.001 DPT_Switch
4	36	68	100	132	157	182	207	IN.j - Differential counter	1 byte unsigned value	Send unsigned value (0..255) of the differential counter	5.010 DPT_Value_1_Ucount
4	36	68	100	132	157	182	207	IN.j - Differential counter	1 byte signed value	Send signed value (-128..127) of the differential counter	6.010 DPT_Value_1_Count
4	36	68	100	132	157	182	207	IN.j - Differential counter	2 byte unsigned value	Send unsigned value (0..65535) of the differential counter	7.001 DPT_Value_2_Ucount
4	36	68	100	132	157	182	207	IN.j - Differential counter	2 byte signed value	Send signed value (-32768..32767) of the differential counter	8.001 DPT_Value_2_Count
4	36	68	100	132	157	182	207	IN.j - Differential counter	4 byte unsigned value	Send unsigned value (0.. 4294967295) of the differential counter	12.001 DPT_Value_4_Ucount
4	36	68	100	132	157	182	207	IN.j - Differential counter	4 byte signed value	Send signed value (-2147483648.. 2147483647) of the differential counter	13.001 DPT_Value_4_Count
4	36	68	100	132	157	182	207	IN.j - D object 1 bit value	1/0 value	Send 1/0 values associated to object D	1.002 DPT_Bool
4	36	68	100	132	157	182	207	IN.j - Single press 1 bit D object	1/0 value	Send 1/0 values associated to object D of the single press	1.002 DPT_Bool
5	37	69	101	133	158	183	208	IN.j - Differential counter bit overflow	Overflow status	Send the differential counter overflow feedback	1.002 DPT_Bool
5	37	69	101	133	158	183	208	IN.j - Double press 1 bit A object	1/0 value	Send 1/0 values associated to object A of the double press	1.002 DPT_Bool
5	37	69	101	133	158	183	208	IN.j - Double press 2 bit A object	On/Off forced positioning	Send priority commands associated to object A of the double press	1.002 DPT_Switch_Control
5	37	69	101	133	158	183	208	IN.j - Double press 1 byte A object	Unsigned value	Send unsigned values (0..255) associated to object A of the double press	5.010 DPT_Value_1_Ucount
5	37	69	101	133	158	183	208	IN.j - Double press 1 byte A object	Signed value	Send signed values (-128..127) associated to object A of the double press	6.010 DPT_Value_1_Count
5	37	69	101	133	158	183	208	IN.j - Double press 1 byte	% value	Send percent values	5.001

								A object		(0%..100%) associated to object A of the double press	DPT_Scaling
5	37	69	101	133	158	183	208	IN.j - Double press 1 byte A object	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated to object A of double press	20.102 DPT_HVAC Mode
5	37	69	101	133	158	183	208	IN.j - Double press 2 byte A object	Unsigned value	Send unsigned values (0..65535) associated to object A of the double press	7.001 DPT_Value_2_Ucount
5	37	69	101	133	158	183	208	IN.j - Double press 2 byte A object	Signed value	Send signed values (-32768..32767) associated to object A of the double press	8.001 DPT_Value_2_Count
5	37	69	101	133	158	183	208	IN.j - Double press 3 byte A object	RGB color	Send RGB color components value associated to object A of the double press	232.600 DPT_Colour_RGB
5	37	69	101	133	158	183	208	IN.j - Double press 4 byte A object	Unsigned value	Sends unsigned values (0.. 4294967295) associated to object A of the double press	12.001 DPT_Value_4_Ucount
5	37	69	101	133	158	183	208	IN.j - Double press 4 byte A object	Signed value	Sends signed values (-2147483648.. 2147483647) associated to object A of the double press	13.001 DPT_Value_4_Count
6	38	70	102	134	159	184	209	IN.j - Differential counter byte overflow	Overflow status	Send the value associated to the differential counter overflow feedback	5.010 DPT_Value_1_Ucount
6	38	70	102	134	159	184	209	IN.j - Double press 1 bit B object	1/0 value	Send 1/0 values associated to object B of the double press	1.002 DPT_Bool
7	39	71	103	135	160	185	210	IN.j - Double press 1 bit C object	1/0 value	Send 1/0 values associated to object C of the double press	1.002 DPT_Bool
8	40	72	104	136	161	186	211	IN.j - Double press 1 bit D object	1/0 value	Send 1/0 values associated to object D of the double press	1.002 DPT_Bool
9	41	73	105	137	162	187	212	IN.j - Triple press 1 bit A object	1/0 value	Send 1/0 values associated to object A of the triple press	1.002 DPT_Bool
9	41	73	105	137	162	187	212	IN.j - Triple press 2 bit A object	On/Off forced positioning	Send priority commands associated to object A of the triple press	1.002 DPT_Switch_Control
9	41	73	105	137	162	187	212	IN.j - Triple press 1 byte A object	Unsigned value	Send unsigned values (0..255) associated to object A of the triple press	5.010 DPT_Value_1_Ucount
9	41	73	105	137	162	187	212	IN.j - Triple press 1 byte A object	Signed value	Send signed values (-128..127) associated to object A of the triple press	6.010 DPT_Value_1_Count
9	41	73	105	137	162	187	212	IN.j - Triple press 1 byte A object	% value	Send percent values (0%..100%) associated to object A of the triple press	5.001 DPT_Scaling
9	41	73	105	137	162	187	212	IN.j - Triple press 1 byte A object	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated to object A of triple press	20.102 DPT_HVAC Mode
9	41	73	105	137	162	187	212	IN.j - Triple press 2 byte A object	Unsigned value	Send unsigned values (0..65535) associated to object A of the triple	7.001 DPT_Value_2_Ucount

										press	
9	41	73	105	137	162	187	212	IN.j - Triple press 2 byte A object	Signed value	Send signed values (-32768..32767) associated to object A of the triple press	8.001 DPT_Value_2_Count
9	41	73	105	137	162	187	212	IN.j - Triple press 3 byte A object	RGB color	Send RGB color components value associated to object A of the triple press	232.600 DPT_Colour_RGB
9	41	73	105	137	162	187	212	IN.j - Triple press 4 byte A object	Unsigned value	Sends unsigned values (0.. 4294967295) associated to object A of the triple press	12.001 DPT_Value_4_Ucount
9	41	73	105	137	162	187	212	IN.j - Triple press 4 byte A object	Signed value	Sends signed values (-2147483648.. 2147483647) associated to object A of the triple press	13.001 DPT_Value_4_Count
10	42	74	106	138	163	188	213	IN.j - Triple press 1 bit B object	1/0 value	Send 1/0 values associated to object B of the triple press	1.002 DPT_Bool
11	43	75	107	139	164	189	214	IN.j - Triple press 1 bit C object	1/0 value	Send 1/0 values associated to object C of the triple press	1.002 DPT_Bool
12	44	76	108	140	165	190	215	IN.j - Triple press 1 bit D object	1/0 value	Send 1/0 values associated to object D of the triple press	1.002 DPT_Bool
13	45	77	109	141	166	191	216	IN.j - Quadruple press 1 bit A object	1/0 value	Send 1/0 values associated to object A of the quadruple press	1.002 DPT_Bool
13	45	77	109	141	166	191	216	IN.j - Quadruple press 2 bit A object	On/Off forced positioning	Send priority commands associated to object A of the quadruple press	1.002 DPT_Switch_Control
13	45	77	109	141	166	191	216	IN.j - Quadruple press 1 byte A object	Unsigned value	Send unsigned values (0..255) associated to object A of the quadruple press	5.010 DPT_Value_1_Ucount
13	45	77	109	141	166	191	216	IN.j - Quadruple press 1 byte A object	Signed value	Send signed values (-128..127) associated to object A of the quadruple press	6.010 DPT_Value_1_Count
13	45	77	109	141	166	191	216	IN.j - Quadruple press 1 byte A object	% value	Send percent values (0%..100%) associated to object A of the quadruple press	5.001 DPT_Scaling
13	45	77	109	141	166	191	216	IN.j - Quadruple press 1 byte A object	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated to object A of quadruple press	20.102 DPT_HVACMode
13	45	77	109	141	166	191	216	IN.j - Quadruple press 2 byte A object	Unsigned value	Send unsigned values (0..65535) associated to object A of the quadruple press	7.001 DPT_Value_2_Ucount
13	45	77	109	141	166	191	216	IN.j - Quadruple press 2 byte A object	Signed value	Send signed values (-32768..32767) associated to object A of the quadruple press	8.001 DPT_Value_2_Count
13	45	77	109	141	166	191	216	IN.j - Quadruple press 3 byte A object	RGB color	Send RGB color components value associated to object A of the quadruple press	232.600 DPT_Colour_RGB
13	45	77	109	141	166	191	216	IN.j - Quadruple press 4 byte A object	Unsigned value	Sends unsigned values (0.. 4294967295) associated to object A of the quadruple press	12.001 DPT_Value_4_Ucount
13	45	77	109	141	166	191	216	IN.j - Quadruple press 4 byte A object	Signed value	Sends signed values (-2147483648..	13.001 DPT_Value_

										2147483647) associated to object A of the quadruple press	4_Count
14	46	78	110	142	167	192	217	IN.j - Quadruple press 1 bit B object	1/0 value	Send 1/0 values associated to object B of the quadruple press	1.002 DPT_Bool
15	47	79	111	143	168	193	218	IN.j - Quadruple press 1 bit C object	1/0 value	Send 1/0 values associated to object C of the quadruple press	1.002 DPT_Bool
16	48	80	112	144	169	194	219	IN.j - Quadruple press 1 bit D object	1/0 value	Send 1/0 values associated to object D of the quadruple press	1.002 DPT_Bool
17	49	81	113	145	170	195	220	IN.j - Long press 1 bit A object	1/0 value	Send 1/0 values associated to object A of the long press	1.002 DPT_Bool
17	49	81	113	145	170	195	220	IN.j - Long press 2 bit A object	On/Off forced positioning	Send priority commands associated to object A of the long press	1.002 DPT_Switch_Control
17	49	81	113	145	170	195	220	IN.j - Long press 1 byte A object	Unsigned value	Send unsigned values (0..255) associated to object A of the long press	5.010 DPT_Value_1_Ucount
17	49	81	113	145	170	195	220	IN.j - Long press 1 byte A object	Signed value	Send signed values (-128..127) associated to object A of the long press	6.010 DPT_Value_1_Count
17	49	81	113	145	170	195	220	IN.j - Long press 1 byte A object	% value	Send percent values (0%..100%) associated to object A of the long press	5.001 DPT_Scaling
17	49	81	113	145	170	195	220	IN.j - Long press 1 byte A object	HVAC mode	Send HVAC mode (auto/comfort/precomfort/economy/off) associated to object A of long press	20.102 DPT_HVAC Mode
17	49	81	113	145	170	195	220	IN.j - Long press 2 byte A object	Unsigned value	Send unsigned values (0..65535) associated to object A of the long press	7.001 DPT_Value_2_Ucount
17	49	81	113	145	170	195	220	IN.j - Long press 2 byte A object	Signed value	Send signed values (-32768..32767) associated to object A of the long press	8.001 DPT_Value_2_Count
17	49	81	113	145	170	195	220	IN.j - Long press 3 byte A object	RGB color	Send RGB color components value associated to object A of the long press	232.600 DPT_Colour_RGB
17	49	81	113	145	170	195	220	IN.j - Long press 4 byte A object	Unsigned value	Sends unsigned values (0.. 4294967295) associated to object A of the long press	12.001 DPT_Value_4_Ucount
17	49	81	113	145	170	195	220	IN.j - Long press 4 byte A object	Signed value	Sends signed values (-2147483648.. 2147483647) associated to object A of the long press	13.001 DPT_Value_4_Count
18	50	82	114	146	171	196	221	IN.j - Long press 1 bit B object	1/0 value	Send 1/0 values associated to object B of the long press	1.002 DPT_Bool
19	51	83	115	147	172	197	222	IN.j - Long press 1 bit C object	1/0 value	Send 1/0 values associated to object C of the long press	1.002 DPT_Bool
20	52	84	116	148	173	198	223	IN.j - Long press 1 bit D object	1/0 value	Send 1/0 values associated to object D of the long press	1.002 DPT_Bool

The object variants shown in blue in the table above are not shown for objects B (objects 2/34/66/98/130/155/180/205), C (objects 3/35/67/99/131/156/181/206) and D (objects 4/36/68/100/132/157/182/207) due to space problems, but are still present.
The object variants shown in red in the table above are not shown for objects B (objects 6/38/70/102/134/159/184/209), C (objects 7/39/71/103/135/160/185/210) and D (objects 8/40/72/104/136/161/186/211) due to space problems, but are still present.
The object variants shown in orange in the table above are not shown for objects B (objects 10/42/74/106/138/163/188/213), C (objects 11/43/75/107/139/164/189/214) and D (objects 12/44/76/108/140/165/190/215) due to space problems, but are still present.
The object variants shown in green in the table above are not shown for objects B (objects 14/46/78/110/142/167/192/217), C (objects 15/47/79/111/143/168/193/218) and D (objects 16/48/80/112/144/169/194/219) due to space problems, but are still present.
The object variants shown in grey in the table above are not shown for objects B (objects 18/50/82/114/146/171/196/221), C (objects 19/51/83/115/147/171/197/222) and D (objects 20/52/84/116/148/172/198/223) due to space problems, but are still present.

Communication objects with **input** functions:

#								Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4	IN 5	IN 6	IN 7	IN 8				
0	32	64	96	128	153	178	203	IN.j - Block	Switching On/Off	Allows to activate/deactivate the block function	1.003 DPT_Enable
2	34	66	98	130	155	180	205	IN.j - Scene storing trigger	Store	Receive the sending learn scene message request (trigger)	1.017 DPT_Trigger
21	53	85	117	149	174	199	224	IN.j - Dimmer status feedback	On/Off status	Dimmer activation status feedbacks	1.001 DPT_Switch
21	53	85	117	149	174	199	224	IN.j - A object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object A	1.001 DPT_Switch
21	53	85	117	149	174	199	224	IN.j - Movement feedback	Increase/Decrease	Receive movement direction feedback from shutter actuator	1.008 DPT_UpDown
21	53	85	117	149	174	199	224	IN.j - Primary counter sending trigger	Counter value transmission	Receive the sending primary counter actual value request (trigger)	1.017 DPT_Trigger
22	54	86	118	150	175	200	225	IN.j - B object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object B	1.001 DPT_Switch
22	54	86	118	150	175	200	225	IN.j - Differential counter sending trigger	Counter value transmission	Receive the sending differential counter actual value request (trigger)	1.017 DPT_Trigger
23	55	87	119	151	176	201	226	IN.j - C object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object C	1.001 DPT_Switch
23	55	87	119	151	176	201	226	IN.j - Differential counter reset	Reset value	Receives the differential counter value reset	1.017 DPT_Trigger
24	56	88	120	152	177	202	227	IN.j - D object status feedback	On/Off status	Receive status feedback from actuator for cyclical commutation function of object D	1.001 DPT_Switch

10.2 Communication objects for "Temperature sensor x"

Communication objects with **output** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
0	32	64	96	IN.x - Measured temperature	Value °C	Send temperature values in Celsius degrees	9.001 DPT_Temp
0	32	64	96	IN.x - Measured temperature	°K value	Send temperature values in Kelvin degrees	9.002 DPT_Tempd
0	32	64	96	IN.x - Measured temperature	°F value	Send temperature values in Fahrenheit degrees	9.027 DPT_Temp_F
1	33	65	97	IN.x - Maximum temperature measured	Value °C	Send measured temperature maximum values in Celsius degrees	9.001 DPT_Temp
1	33	65	97	IN.x - Maximum temperature measured	°K value	Send measured temperature maximum values in Kelvin degrees	9.002 DPT_Tempd
1	33	65	97	IN.x - Maximum temperature measured	°F value	Send measured temperature maximum values in Fahrenheit degrees	9.027 DPT_Temp_F
2	34	66	98	IN.x - Minimum temperature measured	Value °C	Send measured temperature minimum values in Celsius degrees	9.001 DPT_Temp
2	34	66	98	IN.x - Minimum temperature measured	°K value	Send measured temperature minimum values in Kelvin degrees	9.002 DPT_Tempd
2	34	66	98	IN.x - Minimum temperature measured	°F value	Send measured temperature minimum values in Fahrenheit degrees	9.027 DPT_Temp_F
6	38	70	102	IN.x - Temperature threshold 1 enabling status	Enabled/Disabled	Sends temperature threshold 1 enabling status	1.003 DPT_Enable
8	40	72	104	IN.x - Temperature threshold 1 feedback	Value °C	Sends the temperature threshold 1 actual value in Celsius degrees	9.001 DPT_Temp
8	40	72	104	IN.x - Temperature threshold 1 feedback	°K value	Sends the temperature threshold 1 actual value in Kelvin degrees	9.002 DPT_Tempd
8	40	72	104	IN.x - Temperature threshold 1 feedback	°F value	Sends the temperature threshold 1 actual value in Fahrenheit degrees	9.027 DPT_Temp_F
9	41	73	105	IN.x - Temperature threshold 1 output	1/0 value	Sends the values 1/0 associated to the output of temperature threshold 1	1.001 DPT_Switch
9	41	73	105	IN.x - Temperature threshold 1 output	On/Off forced positioning	Sends the 2 bit values associated to the output of temperature threshold 1	2.001 DPT_Switch_Control
9	41	73	105	IN.x - Temperature threshold 1 output	Value 0..255	Sends the unsigned values (0..255) associated to the output of temperature threshold 1	5.010 DPT_Value_1_Ucount
9	41	73	105	IN.x - Temperature threshold 1 output	Value -128 .. +127	Sends the signed values (-128..127) associated to the output of temperature threshold 1	6.010 DPT_Value_1_Count
9	41	73	105	IN.x - Temperature threshold 1 output	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output of temperature threshold 1	5.001 DPT_Percentage
9	41	73	105	IN.x - Temperature threshold 1 output	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output	20.102 DPT_HVAC_Mode

						of temperature threshold 1	
9	41	73	105	IN.x - Temperature threshold 1 output	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output of temperature threshold 1	7.001 DPT_Value_2_Ucount
9	41	73	105	IN.x - Temperature threshold 1 output	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of temperature threshold 1	8.001 DPT_Value_2_Count
9	41	73	105	IN.x - Temperature threshold 1 output	Setpoint value [°C]	Sends the setpoint values (°C) associated to the output of temperature threshold 1	9.001 DPT_Value_Temp
9	41	73	105	IN.x - Temperature threshold 1 output	Setpoint value [°K]	Sends the setpoint values (°K) associated to the output of temperature threshold 1	9.002 DPT_Value_Tempd
9	41	73	105	IN.x - Temperature threshold 1 output	Setpoint value [°F]	Sends the setpoint values (°F) associated to the output of temperature threshold 1	9.027 DPT_Value_Temp_F
12	44	76	108	IN.x - Temperature threshold 2 enabling status	Enabled/Disabled	Sends temperature threshold 2 enabling status	1.003 DPT_Enable
14	46	78	110	IN.x - Temperature threshold 2 feedback	Value °C	Sends the temperature threshold 2 actual value in Celsius degrees	9.001 DPT_Temp
14	46	78	110	IN.x - Temperature threshold 2 feedback	°K value	Sends the temperature threshold 2 actual value in Kelvin degrees	9.002 DPT_Tempd
14	46	78	110	IN.x - Temperature threshold 2 feedback	°F value	Sends the temperature threshold 2 actual value in Fahrenheit degrees	9.027 DPT_Temp_F
15	47	79	111	IN.x - Temperature threshold 2 output	1/0 value	Sends the values 1/0 associated to the output of temperature threshold 2	1.001 DPT_Switch
15	47	79	111	IN.x - Temperature threshold 2 output	On/Off forced positioning	Sends the 2 bit values associated to the output of temperature threshold 2	2.001 DPT_Switch_Control
15	47	79	111	IN.x - Temperature threshold 2 output	Value 0..255	Sends the unsigned values (0..255) associated to the output of temperature threshold 2	5.010 DPT_Value_1_Ucount
15	47	79	111	IN.x - Temperature threshold 2 output	Value -128 .. +127	Sends the signed values (-128..127) associated to the output of temperature threshold 2	6.010 DPT_Value_1_Count
15	47	79	111	IN.x - Temperature threshold 2 output	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output of temperature threshold 2	5.001 DPT_Percentage
15	47	79	111	IN.x - Temperature threshold 2 output	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output of temperature threshold 2	20.102 DPT_HVAC_Mode
15	47	79	111	IN.x - Temperature threshold 2 output	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output of temperature threshold 2	7.001 DPT_Value_2_Ucount
15	47	79	111	IN.x - Temperature threshold 2 output	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of temperature threshold 2	8.001 DPT_Value_2_Count
15	47	79	111	IN.x - Temperature threshold 2 output	Setpoint value [°C]	Sends the setpoint values (°C) associated to the output of temperature threshold 2	9.001 DPT_Value_Temp
15	47	79	111	IN.x - Temperature threshold 2 output	Setpoint value [°K]	Sends the setpoint values (°K) associated to the output of temperature threshold 2	9.002 DPT_Value_Tempd

15	47	79	111	IN.x - Temperature threshold 2 output	Setpoint value [°F]	Sends the setpoint values (°F) associated to the output of temperature threshold 2	9.027 DPT_Value_Temp_F
18	50	82	114	IN.x - Temperature threshold 3 enabling status	Enabled/Disabled	Sends temperature threshold 3 enabling status	1.003 DPT_Enable
20	52	84	116	IN.x - Temperature threshold 3 feedback	Value °C	Sends the temperature threshold 3 actual value in Celsius degrees	9.001 DPT_Temp
20	52	84	116	IN.x - Temperature threshold 3 feedback	°K value	Sends the temperature threshold 3 actual value in Kelvin degrees	9.002 DPT_Tempd
20	52	84	116	IN.x - Temperature threshold 3 feedback	°F value	Sends the temperature threshold 3 actual value in Fahrenheit degrees	9.027 DPT_Temp_F
21	53	85	117	IN.x - Temperature threshold 3 output	1/0 value	Sends the values 1/0 associated to the output of temperature threshold 3	1.001 DPT_Switch
21	53	85	117	IN.x - Temperature threshold 3 output	On/Off forced positioning	Sends the 2 bit values associated to the output of temperature threshold 3	2.001 DPT_Switch_Control
21	53	85	117	IN.x - Temperature threshold 3 output	Value 0..255	Sends the unsigned values (0..255) associated to the output of temperature threshold 3	5.010 DPT_Value_1_Ucount
21	53	85	117	IN.x - Temperature threshold 3 output	Value -128 .. +127	Sends the signed values (-128..127) associated to the output of temperature threshold 3	6.010 DPT_Value_1_Count
21	53	85	117	IN.x - Temperature threshold 3 output	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output of temperature threshold 3	5.001 DPT_Percentage
21	53	85	117	IN.x - Temperature threshold 3 output	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output of temperature threshold 3	20.102 DPT_HVAC_Mode
21	53	85	117	IN.x - Temperature threshold 3 output	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output of temperature threshold 3	7.001 DPT_Value_2_Ucount
21	53	85	117	IN.x - Temperature threshold 3 output	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of temperature threshold 3	8.001 DPT_Value_2_Count
21	53	85	117	IN.x - Temperature threshold 3 output	Setpoint value [°C]	Sends the setpoint values (°C) associated to the output of temperature threshold 3	9.001 DPT_Value_Temp
21	53	85	117	IN.x - Temperature threshold 3 output	Setpoint value [°K]	Sends the setpoint values (°K) associated to the output of temperature threshold 3	9.002 DPT_Value_Tempd
21	53	85	117	IN.x - Temperature threshold 3 output	Setpoint value [°F]	Sends the setpoint values (°F) associated to the output of temperature threshold 3	9.027 DPT_Value_Temp_F
24	56	88	120	IN.x - Temperature threshold 4 enabling status	Enabled/Disabled	Sends temperature threshold 4 enabling status	1.003 DPT_Enable
26	58	90	122	IN.x - Temperature threshold 4 feedback	Value °C	Sends the temperature threshold 4 actual value in Celsius degrees	9.001 DPT_Temp
26	58	90	122	IN.x - Temperature threshold 4 feedback	°K value	Sends the temperature threshold 4 actual value in Kelvin degrees	9.002 DPT_Tempd
26	58	90	122	IN.x - Temperature threshold 4 feedback	°F value	Sends the temperature threshold 4 actual value in Fahrenheit degrees	9.027 DPT_Temp_F

27	59	91	123	IN.x - Temperature threshold 4 output	1/0 value	Sends the values 1/0 associated to the output of temperature threshold 4	1.001 DPT_Switch
27	59	91	123	IN.x - Temperature threshold 4 output	On/Off forced positioning	Sends the 2 bit values associated to the output of temperature threshold 4	2.001 DPT_Switch_Control
27	59	91	123	IN.x - Temperature threshold 4 output	Value 0..255	Sends the unsigned values (0..255) associated to the output of temperature threshold 4	5.010 DPT_Value_1_Ucount
27	59	91	123	IN.x - Temperature threshold 4 output	Value -128 .. +127	Sends the signed values (-128..127) associated to the output of temperature threshold 4	6.010 DPT_Value_1_Count
27	59	91	123	IN.x - Temperature threshold 4 output	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output of temperature threshold 4	5.001 DPT_Percentage
27	59	91	123	IN.x - Temperature threshold 4 output	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output of temperature threshold 4	20.102 DPT_HVAC_Mode
27	59	91	123	IN.x - Temperature threshold 4 output	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output of temperature threshold 4	7.001 DPT_Value_2_Ucount
27	59	91	123	IN.x - Temperature threshold 4 output	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of temperature threshold 4	8.001 DPT_Value_2_Count
27	59	91	123	IN.x - Temperature threshold 4 output	Setpoint value [°C]	Sends the setpoint values (°C) associated to the output of temperature threshold 4	9.001 DPT_Value_Temp
27	59	91	123	IN.x - Temperature threshold 4 output	Setpoint value [°K]	Sends the setpoint values (°K) associated to the output of temperature threshold 4	9.002 DPT_Value_Tempd
27	59	91	123	IN.x - Temperature threshold 4 output	Setpoint value [°F]	Sends the setpoint values (°F) associated to the output of temperature threshold 4	9.027 DPT_Value_Temp_F

Communication objects with **input** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
3	35	67	99	IN.x - Maximum and minimum temperature value reset	Values reset	Receives trigger commands of maximum and minimum temperature values reset	1.017 DPT_Trigger
4	36	68	100	IN.x - Feedbacks sending trigger	Send feedback	Receives trigger commands of feedback sending request	1.017 DPT_Trigger
5	37	69	101	IN.x - Temperature threshold 1 enabling	0=enable / 1=disable	Receives temperature threshold 1 enabling commands	1.002 DPT_Bool
5	37	69	101	IN.x - Temperature threshold 1 enabling	1=enable / 0=disable	Receives temperature threshold 1 enabling commands	1.002 DPT_Bool
7	39	71	103	IN.x - Temperature threshold 1 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the temperature threshold 1 value	1.007 DPT_Step
7	39	71	103	IN.x - Temperature threshold 1 value input	Value °C	Receives the temperature threshold 1 values in Celsius degrees	9.001 DPT_Temp
7	39	71	103	IN.x - Temperature threshold 1 value input	°K value	Receives the temperature threshold 1 values in Kelvin degrees	9.002 DPT_Tempd
7	39	71	103	IN.x - Temperature threshold 1 value input	°F value	Receives the temperature threshold 1 values in Fahrenheit degrees	9.027 DPT_Temp_F
10	42	74	106	IN.x - Temperature threshold 1 output reference	HVAC Mode (com/precom/eco/off)	Receives the reference HVAC mode values (comfort/precomfort/economy /off) for the calculation of the output associated with the temperature threshold 1	20.102 DPT_HVAC_Mode
10	42	74	106	IN.x - Temperature threshold 1 output reference	Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output associated with the temperature threshold 1	9.001 DPT_Temp
10	42	74	106	IN.x - Temperature threshold 1 output reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output associated with the temperature threshold 1	9.002 DPT_Tempd
10	42	74	106	IN.x - Temperature threshold 1 output reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of the output associated with the temperature threshold 1	9.027 DPT_Temp_F
10	42	74	106	IN.x - Temperature threshold 1 output status	On/Off status	Receives feedback about the state of activation of the output associated with the temperature threshold 1	1.001 DPT_Switch
11	43	75	107	IN.x - Temperature threshold 2 enabling	0=enable / 1=disable	Receives temperature threshold 2 enabling commands	1.002 DPT_Bool
11	43	75	107	IN.x - Temperature threshold 2 enabling	1=enable / 0=disable	Receives temperature threshold 2 enabling commands	1.002 DPT_Bool
13	45	77	109	IN.x - Temperature threshold 2 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the temperature threshold 2 value	1.007 DPT_Step
13	45	77	109	IN.x - Temperature threshold 2 value input	Value °C	Receives the temperature threshold 2 values in Celsius degrees	9.001 DPT_Temp

13	45	77	109	IN.x - Temperature threshold 2 value input	°K value	Receives the temperature threshold 2 values in Kelvin degrees	9.002 DPT_Tempd
13	45	77	109	IN.x - Temperature threshold 2 value input	°F value	Receives the temperature threshold 2 values in Fahrenheit degrees	9.027 DPT_Temp_F
16	48	80	112	IN.x - Temperature threshold 2 output reference	HVAC Mode (com/precom/eco/off)	Receives the reference HVAC mode values (comfort/precomfort/economy /off) for the calculation of the output associated with the temperature threshold 2	20.102 DPT_HVAC_Mode
16	48	80	112	IN.x - Temperature threshold 2 output reference	Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output associated with the temperature threshold 2	9.001 DPT_Temp
16	48	80	112	IN.x - Temperature threshold 2 output reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output associated with the temperature threshold 2	9.002 DPT_Tempd
16	48	80	112	IN.x - Temperature threshold 2 output reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of the output associated with the temperature threshold 2	9.027 DPT_Temp_F
16	48	80	112	IN.x - Temperature threshold 2 output status	On/Off status	Receives feedback about the state of activation of the output associated with the temperature threshold 2	1.001 DPT_Switch
17	49	81	113	IN.x - Temperature threshold 3 enabling	0=enable / 1=disable	Receives temperature threshold 3 enabling commands	1.002 DPT_Bool
17	49	81	113	IN.x - Temperature threshold 3 enabling	1=enable / 0=disable	Receives temperature threshold 3 enabling commands	1.002 DPT_Bool
19	51	83	115	IN.x - Temperature threshold 3 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the temperature threshold 3 value	1.007 DPT_Step
19	51	83	115	IN.x - Temperature threshold 3 value input	Value °C	Receives the temperature threshold 3 values in Celsius degrees	9.001 DPT_Temp
19	51	83	115	IN.x - Temperature threshold 3 value input	°K value	Receives the temperature threshold 3 values in Kelvin degrees	9.002 DPT_Tempd
19	51	83	115	IN.x - Temperature threshold 3 value input	°F value	Receives the temperature threshold 3 values in Fahrenheit degrees	9.027 DPT_Temp_F
22	54	86	118	IN.x - Temperature threshold 3 output reference	HVAC Mode (com/precom/eco/off)	Receives the reference HVAC mode values (comfort/precomfort/economy /off) for the calculation of the output associated with the temperature threshold 3	20.102 DPT_HVAC_Mode
22	54	86	118	IN.x - Temperature threshold 3 output reference	Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output associated with the temperature threshold 3	9.001 DPT_Temp
22	54	86	118	IN.x - Temperature threshold 3 output reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output associated with the temperature threshold 3	9.002 DPT_Tempd
22	54	86	118	IN.x - Temperature threshold 3 output	Setpoint [°F]	Receives the reference setpoint values (°F) for the	9.027 DPT_Temp_F

				reference		calculation of the output associated with the temperature threshold 3	
22	54	86	118	IN.x - Temperature threshold 3 output status	On/Off status	Receives feedback about the state of activation of the output associated with the temperature threshold 3	1.001 DPT_Switch
23	55	87	119	IN.x - Temperature threshold 4 enabling	0=enable / 1=disable	Receives temperature threshold 4 enabling commands	1.002 DPT_Bool
23	55	87	119	IN.x - Temperature threshold 4 enabling	1=enable / 0=disable	Receives temperature threshold 4 enabling commands	1.002 DPT_Bool
25	57	89	121	IN.x - Temperature threshold 4 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the temperature threshold 4 value	1.007 DPT_Step
25	57	89	121	IN.x - Temperature threshold 4 value input	Value °C	Receives the temperature threshold 4 values in Celsius degrees	9.001 DPT_Temp
25	57	89	121	IN.x - Temperature threshold 4 value input	°K value	Receives the temperature threshold 4 values in Kelvin degrees	9.002 DPT_Tempd
25	57	89	121	IN.x - Temperature threshold 4 value input	°F value	Receives the temperature threshold 4 values in Fahrenheit degrees	9.027 DPT_Temp_F
28	60	92	124	IN.x - Temperature threshold 4 output reference	HVAC Mode (com/precom/eco/off)	Receives the reference HVAC mode values (comfort/precomfort/economy /off) for the calculation of the output associated with the temperature threshold 4	20.102 DPT_HVAC_Mode
28	60	92	124	IN.x - Temperature threshold 4 output reference	Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output associated with the temperature threshold 4	9.001 DPT_Temp
28	60	92	124	IN.x - Temperature threshold 4 output reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output associated with the temperature threshold 4	9.002 DPT_Tempd
28	60	92	124	IN.x - Temperature threshold 4 output reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of the output associated with the temperature threshold 4	9.027 DPT_Temp_F
28	60	92	124	IN.x - Temperature threshold 4 output status	On/Off status	Receives feedback about the state of activation of the output associated with the temperature threshold 4	1.001 DPT_Switch

10.3 Communication objects for “Analog input x”

Communication objects with **output** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
0	32	64	96	IN.x - Measured (converted) value	2 byte floating value	Send measured (converted) values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
1	33	65	97	IN.x - Maximum measured (converted) value	2 byte floating value	Send maximum measured (converted) values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
2	34	66	98	IN.x - Minimum measured (converted) value	2 byte floating value	Send minimum measured (converted) values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
4	36	68	100	IN.x - Measured scaled value (%)	Value [0% .. 100%]	Send scaled percentage measured values (0%..100%)	5.001 DPT_Percentage
5	37	69	101	IN.x - Signal out of range alarm	Alarm/No alarm	Sends the input signal out of range alarm feedbacks	1.005. DPT_Alarm
7	39	71	103	IN.x - Limit threshold 1 enabling status	Enabled/Disabled	Sends limit threshold 1 enabling status	1.003 DPT_Enable
9	41	73	105	IN.x - Limit threshold 1 feedback	2 byte floating value	Sends the limit threshold 1 actual values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
10	42	74	106	IN.x - Limit threshold 1 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 1	1.001 DPT_Switch
10	42	74	106	IN.x - Limit threshold 1 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 1	2.001 DPT_Switch_Control
10	42	74	106	IN.x - Limit threshold 1 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 1	5.010 DPT_Value_1_Ucount
10	42	74	106	IN.x - Limit threshold 1 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 1	6.010 DPT_Value_1_Count
10	42	74	106	IN.x - Limit threshold 1 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 1	5.001 DPT_Percentage
10	42	74	106	IN.x - Limit threshold 1 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output A of limit threshold 1	20.102 DPT_HVAC_Mode
10	42	74	106	IN.x - Limit threshold 1 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 1	7.001 DPT_Value_2_Ucount
10	42	74	106	IN.x - Limit threshold 1 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of limit threshold 1	8.001 DPT_Value_2_Count
11	43	75	107	IN.x - Limit threshold 1 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 1	1.001 DPT_Switch
11	43	75	107	IN.x - Limit threshold 1 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 1	2.001 DPT_Switch_Control
11	43	75	107	IN.x - Limit threshold 1 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 1	5.010 DPT_Value_1_Ucount
11	43	75	107	IN.x - Limit threshold 1 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the	6.010 DPT_Value_1_Coun

						output B of limit threshold 1	t
11	43	75	107	IN.x - Limit threshold 1 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 1	5.001 DPT_Percentage
11	43	75	107	IN.x - Limit threshold 1 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output B of limit threshold 1	20.102 DPT_HVAC_Mode
11	43	75	107	IN.x - Limit threshold 1 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 1	7.001 DPT_Value_2_Ucount
11	43	75	107	IN.x - Limit threshold 1 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 1	8.001 DPT_Value_2_Count
13	45	77	109	IN.x - Limit threshold 2 enabling status	Enabled/Disabled	Sends limit threshold 2 enabling status	1.003 DPT_Enable
15	47	79	111	IN.x - Limit threshold 2 feedback	2 byte floating value	Sends the limit threshold 2 actual values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
16	48	80	112	IN.x - Limit threshold 2 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 2	1.001 DPT_Switch
16	48	80	112	IN.x - Limit threshold 2 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 2	2.001 DPT_Switch_Control
16	48	80	112	IN.x - Limit threshold 2 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 2	5.010 DPT_Value_1_Ucount
16	48	80	112	IN.x - Limit threshold 2 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 2	6.010 DPT_Value_1_Count
16	48	80	112	IN.x - Limit threshold 2 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 2	5.001 DPT_Percentage
16	48	80	112	IN.x - Limit threshold 2 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output A of limit threshold 2	20.102 DPT_HVAC_Mode
16	48	80	112	IN.x - Limit threshold 2 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 2	7.001 DPT_Value_2_Ucount
16	48	80	112	IN.x - Limit threshold 2 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of limit threshold 2	8.001 DPT_Value_2_Count
17	49	81	113	IN.x - Limit threshold 2 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 2	1.001 DPT_Switch
17	49	81	113	IN.x - Limit threshold 2 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 2	2.001 DPT_Switch_Control
17	49	81	113	IN.x - Limit threshold 2 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 2	5.010 DPT_Value_1_Ucount
17	49	81	113	IN.x - Limit threshold 2 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the output B of limit threshold 2	6.010 DPT_Value_1_Count
17	49	81	113	IN.x - Limit threshold 2 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 2	5.001 DPT_Percentage
17	49	81	113	IN.x - Limit threshold 2 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output	20.102 DPT_HVAC_Mode

						B of limit threshold 2	
17	49	81	113	IN.x - Limit threshold 2 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 2	7.001 DPT_Value_2_Ucount
17	49	81	113	IN.x - Limit threshold 2 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 2	8.001 DPT_Value_2_Count
19	51	83	115	IN.x - Limit threshold 3 enabling status	Enabled/Disabled	Sends limit threshold 3 enabling status	1.003 DPT_Enable
21	53	85	117	IN.x - Limit threshold 3 feedback	2 byte floating value	Sends the limit threshold 3 actual values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
22	54	86	118	IN.x - Limit threshold 3 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 3	1.001 DPT_Switch
22	54	86	118	IN.x - Limit threshold 3 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 3	2.001 DPT_Switch_Control
22	54	86	118	IN.x - Limit threshold 3 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 3	5.010 DPT_Value_1_Ucount
22	54	86	118	IN.x - Limit threshold 3 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 3	6.010 DPT_Value_1_Count
22	54	86	118	IN.x - Limit threshold 3 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 3	5.001 DPT_Percentage
22	54	86	118	IN.x - Limit threshold 3 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output A of limit threshold 3	20.102 DPT_HVAC_Mode
22	54	86	118	IN.x - Limit threshold 3 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 3	7.001 DPT_Value_2_Ucount
22	54	86	118	IN.x - Limit threshold 3 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of limit threshold 3	8.001 DPT_Value_2_Count
23	55	87	119	IN.x - Limit threshold 3 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 2	1.001 DPT_Switch
23	55	87	119	IN.x - Limit threshold 3 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 3	2.001 DPT_Switch_Control
23	55	87	119	IN.x - Limit threshold 3 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 3	5.010 DPT_Value_1_Ucount
23	55	87	119	IN.x - Limit threshold 3 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the output B of limit threshold 3	6.010 DPT_Value_1_Count
23	55	87	119	IN.x - Limit threshold 3 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 3	5.001 DPT_Percentage
23	55	87	119	IN.x - Limit threshold 3 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output B of limit threshold 3	20.102 DPT_HVAC_Mode
23	55	87	119	IN.x - Limit threshold 3 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 3	7.001 DPT_Value_2_Ucount
23	55	87	119	IN.x - Limit threshold 3 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 3	8.001 DPT_Value_2_Count

25	57	89	121	IN.x - Limit threshold 4 enabling status	Enabled/Disabled	Sends limit threshold 4 enabling status	1.003 DPT_Enable
27	59	91	123	IN.x - Limit threshold 4 feedback	2 byte floating value	Sends the limit threshold 4 actual values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
28	60	92	124	IN.x - Limit threshold 4 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 4	1.001 DPT_Switch
28	60	92	124	IN.x - Limit threshold 4 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 4	2.001 DPT_Switch_Control
28	60	92	124	IN.x - Limit threshold 4 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 4	5.010 DPT_Value_1_Ucount
28	60	92	124	IN.x - Limit threshold 4 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 4	6.010 DPT_Value_1_Count
28	60	92	124	IN.x - Limit threshold 4 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 4	5.001 DPT_Percentage
28	60	92	124	IN.x - Limit threshold 4 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output A of limit threshold 4	20.102 DPT_HVAC_Mode
28	60	92	124	IN.x - Limit threshold 4 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 4	7.001 DPT_Value_2_Ucount
28	60	92	124	IN.x - Limit threshold 4 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 4	8.001 DPT_Value_2_Count
29	61	93	125	IN.x - Limit threshold 3 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 3	1.001 DPT_Switch
29	61	93	125	IN.x - Limit threshold 3 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 3	2.001 DPT_Switch_Control
29	61	93	125	IN.x - Limit threshold 3 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 3	5.010 DPT_Value_1_Ucount
29	61	93	125	IN.x - Limit threshold 3 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the output B of limit threshold 3	6.010 DPT_Value_1_Count
29	61	93	125	IN.x - Limit threshold 3 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 3	5.001 DPT_Percentage
29	61	93	125	IN.x - Limit threshold 3 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output B of limit threshold 3	20.102 DPT_HVAC_Mode
29	61	93	125	IN.x - Limit threshold 3 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 3	7.001 DPT_Value_2_Ucount
29	61	93	125	IN.x - Limit threshold 3 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 3	8.001 DPT_Value_2_Count

Communication objects with **input** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
3	35	67	99	IN.X - Maximum and minimum (converted) value reset	Values reset	Receives trigger commands of maximum and minimum measured (converted) values reset	1.017 DPT_Trigger
6	38	70	102	IN.X - Limit threshold 1 enabling	0=enable / 1=disable	Receives limit threshold 1 enabling commands	1.002 DPT_Bool
6	38	70	102	IN.X - Limit threshold 1 enabling	1=enable / 0=disable	Receives limit threshold 1 enabling commands	1.002 DPT_Bool
8	40	72	104	IN.X - Limit threshold 1 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the limit threshold 1 value	1.007 DPT_Step
8	40	72	104	IN.X - Limit threshold 1 value input	2 byte floating value	Receives limit threshold 1 values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
12	44	76	108	IN.X - Limit threshold 2 enabling	0=enable / 1=disable	Receives limit threshold 2 enabling commands	1.002 DPT_Bool
12	44	76	108	IN.X - Limit threshold 2 enabling	1=enable / 0=disable	Receives limit threshold 2 enabling commands	1.002 DPT_Bool
14	46	78	110	IN.X - Limit threshold 2 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the limit threshold 2 value	1.007 DPT_Step
14	46	78	110	IN.X - Limit threshold 2 value input	2 byte floating value	Receives limit threshold 2 values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
18	50	82	114	IN.X - Limit threshold 3 enabling	0=enable / 1=disable	Receives limit threshold 3 enabling commands	1.002 DPT_Bool
18	50	82	114	IN.X - Limit threshold 3 enabling	1=enable / 0=disable	Receives limit threshold 3 enabling commands	1.002 DPT_Bool
20	52	84	116	IN.X - Limit threshold 3 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the limit threshold 3 value	1.007 DPT_Step
20	52	84	116	IN.X - Limit threshold 3 value input	2 byte floating value	Receives limit threshold 3 values with 2 byte floating format	9.xxx DPT_2-Octet Float Value
24	56	88	120	IN.X - Limit threshold 4 enabling	0=enable / 1=disable	Receives limit threshold 4 enabling commands	1.002 DPT_Bool
24	56	88	120	IN.X - Limit threshold 4 enabling	1=enable / 0=disable	Receives limit threshold 4 enabling commands	1.002 DPT_Bool
26	58	90	122	IN.X - Limit threshold 4 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the limit threshold 4 value	1.007 DPT_Step
26	58	90	122	IN.X - Limit threshold 4 value input	2 byte floating value	Receives limit threshold 4 values with 2 byte floating format	9.xxx DPT_2-Octet Float Value

10.4 Communication objects for “Digital input x with S0 interf. ”

Communication objects with **output** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
0	32	64	96	IN.x - Active energy counter [kWh]	Counter value [kWh]	Sends the energy counter values expressed in kilowatt-hour	13.010 DPT_ActiveEnergy
0	32	64	96	IN.x - Water counter [m^3]	Counter value [m^3]	Sends the water counter values expressed in cubic meters	14.076 DPT_Value_Volume
0	32	64	96	IN.x - Gas counter [m^3]	Counter value [m^3]	Sends the gas counter values expressed in cubic meters	14.076 DPT_Value_Volume
1	33	65	97	IN.x - Estimated active power [kW]	Estimated value [kW]	Sends the estimated active power expressed in kilowatt	9.024 DPT_Power
1	33	65	97	IN.x - Estimated active power [W]	Estimated value [W]	Sends the estimated active power expressed in watt	14.056 DPT_Value_Power
5	37	69	101	IN.x - Limit threshold 1 enabling status	Enabled/Disabled	Sends limit threshold 1 enabling status	1.003 DPT_Enable
7	39	71	103	IN.x - Limit threshold 1 feedback	4 byte floating value	Sends the limit threshold 1 actual values with 4 byte floating format	14.005 DPT_Value_Amplitude
8	40	72	104	IN.x - Limit threshold 1 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 1	1.001 DPT_Switch
8	40	72	104	IN.x - Limit threshold 1 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 1	2.001 DPT_Switch_Control
8	40	72	104	IN.x - Limit threshold 1 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 1	5.010 DPT_Value_1_Ucount
8	40	72	104	IN.x - Limit threshold 1 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 1	6.010 DPT_Value_1_Count
8	40	72	104	IN.x - Limit threshold 1 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 1	5.001 DPT_Percentage
8	40	72	104	IN.x - Limit threshold 1 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output A of limit threshold 1	20.102 DPT_HVAC_Mode
8	40	72	104	IN.x - Limit threshold 1 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 1	7.001 DPT_Value_2_Ucount
8	40	72	104	IN.x - Limit threshold 1 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of limit threshold 1	8.001 DPT_Value_2_Count
9	41	73	105	IN.x - Limit threshold 1 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 1	1.001 DPT_Switch
9	41	73	105	IN.x - Limit threshold 1 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 1	2.001 DPT_Switch_Control
9	41	73	105	IN.x - Limit threshold 1 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 1	5.010 DPT_Value_1_Ucount
9	41	73	105	IN.x - Limit threshold 1 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the output B of limit threshold 1	6.010 DPT_Value_1_Count

9	41	73	105	IN.x - Limit threshold 1 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 1	5.001 DPT_Percentage
9	41	73	105	IN.x - Limit threshold 1 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output B of limit threshold 1	20.102 DPT_HVAC_Mode
9	41	73	105	IN.x - Limit threshold 1 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 1	7.001 DPT_Value_2_Ucount
9	41	73	105	IN.x - Limit threshold 1 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 1	8.001 DPT_Value_2_Count
11	43	75	107	IN.x - Limit threshold 2 enabling status	Enabled/Disabled	Sends limit threshold 2 enabling status	1.003 DPT_Enable
13	45	77	109	IN.x - Limit threshold 2 feedback	4 byte floating value	Sends the limit threshold 2 actual values with 4 byte floating format	14.005 DPT_Value_Amplitude
14	46	78	110	IN.x - Limit threshold 2 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 2	1.001 DPT_Switch
14	46	78	110	IN.x - Limit threshold 2 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 2	2.001 DPT_Switch_Control
14	46	78	110	IN.x - Limit threshold 2 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 2	5.010 DPT_Value_1_Ucount
14	46	78	110	IN.x - Limit threshold 2 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 2	6.010 DPT_Value_1_Count
14	46	78	110	IN.x - Limit threshold 2 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 2	5.001 DPT_Percentage
14	46	78	110	IN.x - Limit threshold 2 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output A of limit threshold 2	20.102 DPT_HVAC_Mode
14	46	78	110	IN.x - Limit threshold 2 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 2	7.001 DPT_Value_2_Ucount
14	46	78	110	IN.x - Limit threshold 2 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of limit threshold 2	8.001 DPT_Value_2_Count
15	47	79	111	IN.x - Limit threshold 2 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 2	1.001 DPT_Switch
15	47	79	111	IN.x - Limit threshold 2 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 2	2.001 DPT_Switch_Control
15	47	79	111	IN.x - Limit threshold 2 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 2	5.010 DPT_Value_1_Ucount
15	47	79	111	IN.x - Limit threshold 2 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the output B of limit threshold 2	6.010 DPT_Value_1_Count
15	47	79	111	IN.x - Limit threshold 2 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 2	5.001 DPT_Percentage
15	47	79	111	IN.x - Limit threshold 2 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy /off) associated to the output B of limit threshold 2	20.102 DPT_HVAC_Mode

15	47	79	111	IN.x - Limit threshold 2 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 2	7.001 DPT_Value_2_Ucount
15	47	79	111	IN.x - Limit threshold 2 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 2	8.001 DPT_Value_2_Count
17	49	81	113	IN.x - Limit threshold 3 enabling status	Enabled/Disabled	Sends limit threshold 3 enabling status	1.003 DPT_Enable
19	51	83	115	IN.x - Limit threshold 3 feedback	4 byte floating value	Sends the limit threshold 3 actual values with 4 byte floating format	14.005 DPT_Value_Amplitude
20	52	84	116	IN.x - Limit threshold 3 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 3	1.001 DPT_Switch
20	52	84	116	IN.x - Limit threshold 3 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 3	2.001 DPT_Switch_Control
20	52	84	116	IN.x - Limit threshold 3 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 3	5.010 DPT_Value_1_Ucount
20	52	84	116	IN.x - Limit threshold 3 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 3	6.010 DPT_Value_1_Count
20	52	84	116	IN.x - Limit threshold 3 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 3	5.001 DPT_Percentage
20	52	84	116	IN.x - Limit threshold 3 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output A of limit threshold 3	20.102 DPT_HVAC_Mode
20	52	84	116	IN.x - Limit threshold 3 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 3	7.001 DPT_Value_2_Ucount
20	52	84	116	IN.x - Limit threshold 3 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of limit threshold 3	8.001 DPT_Value_2_Count
21	53	85	117	IN.x - Limit threshold 3 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 3	1.001 DPT_Switch
21	53	85	117	IN.x - Limit threshold 3 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 3	2.001 DPT_Switch_Control
21	53	85	117	IN.x - Limit threshold 3 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 3	5.010 DPT_Value_1_Ucount
21	53	85	117	IN.x - Limit threshold 3 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the output B of limit threshold 3	6.010 DPT_Value_1_Count
21	53	85	117	IN.x - Limit threshold 3 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 3	5.001 DPT_Percentage
21	53	85	117	IN.x - Limit threshold 3 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output B of limit threshold 3	20.102 DPT_HVAC_Mode
21	53	85	117	IN.x - Limit threshold 3 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 3	7.001 DPT_Value_2_Ucount
21	53	85	117	IN.x - Limit threshold 3 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 3	8.001 DPT_Value_2_Count

23	55	87	119	IN.x - Limit threshold 4 enabling status	Enabled/Disabled	Sends limit threshold 4 enabling status	1.003 DPT_Enable
25	57	89	121	IN.x - Limit threshold 4 feedback	4 byte floating value	Sends the limit threshold 4 actual values with 4 byte floating format	14.005 DPT_Value_Amplitude
26	58	90	122	IN.x - Limit threshold 4 output A	1/0 value	Sends the values 1/0 associated to the output A of limit threshold 4	1.001 DPT_Switch
26	58	90	122	IN.x - Limit threshold 4 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of limit threshold 4	2.001 DPT_Switch_Control
26	58	90	122	IN.x - Limit threshold 4 output A	Value 0..255	Sends the unsigned values (0..255) associated to the output A of limit threshold 4	5.010 DPT_Value_1_Ucount
26	58	90	122	IN.x - Limit threshold 4 output A	Value -128 .. +127	Sends the signed values (-128..127) associated to the output A of limit threshold 4	6.010 DPT_Value_1_Count
26	58	90	122	IN.x - Limit threshold 4 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of limit threshold 4	5.001 DPT_Percentage
26	58	90	122	IN.x - Limit threshold 4 output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output A of limit threshold 4	20.102 DPT_HVAC_Mode
26	58	90	122	IN.x - Limit threshold 4 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of limit threshold 4	7.001 DPT_Value_2_Ucount
26	58	90	122	IN.x - Limit threshold 4 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of limit threshold 4	8.001 DPT_Value_2_Count
27	59	91	123	IN.x - Limit threshold 4 output B	1/0 value	Sends the values 1/0 associated to the output B of limit threshold 4	1.001 DPT_Switch
27	59	91	123	IN.x - Limit threshold 4 output B	On/Off forced positioning	Sends the 2 bit values associated to the output B of limit threshold 4	2.001 DPT_Switch_Control
27	59	91	123	IN.x - Limit threshold 4 output B	Value 0..255	Sends the unsigned values (0..255) associated to the output B of limit threshold 4	5.010 DPT_Value_1_Ucount
27	59	91	123	IN.x - Limit threshold 4 output B	Value -128 .. +127	Sends the signed values (-128..127) associated to the output B of limit threshold 4	6.010 DPT_Value_1_Count
27	59	91	123	IN.x - Limit threshold 4 output B	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output B of limit threshold 4	5.001 DPT_Percentage
27	59	91	123	IN.x - Limit threshold 4 output B	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated to the output B of limit threshold 4	20.102 DPT_HVAC_Mode
27	59	91	123	IN.x - Limit threshold 4 output B	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output B of limit threshold 4	7.001 DPT_Value_2_Ucount
27	59	91	123	IN.x - Limit threshold 4 output B	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output B of limit threshold 4	8.001 DPT_Value_2_Count

Communication objects with **input** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
2	34	66	98	IN.x - Counter initial value	Value (0 .. 4294967295)	Receives the counter initial value	12.001 DPT_Value_4_Ucount
3	35	67	99	IN.x - Counter reset	Reset the initial value	Receives trigger commands of counter value and subsequent resetting of the initial value	1.017 DPT_Trigger
4	36	68	100	IN.X - Limit threshold 1 enabling	0=enable / 1=disable	Receives limit threshold 1 enabling commands	1.002 DPT_Bool
4	36	68	100	IN.x - Limit threshold 1 enabling	1=enable / 0=disable	Receives limit threshold 1 enabling commands	1.002 DPT_Bool
6	38	70	102	IN.X - Limit threshold 1 enabling	1=enable / 0=disable	Receives limit threshold 1 enabling commands	1.007 DPT_Step
6	38	70	102	IN.X - Limit threshold 1 value input	4 byte floating value	Receives the limit threshold 1 value	14.005 DPT_Value_Amplitude
10	42	74	106	IN.X - Limit threshold 2 enabling	0=enable / 1=disable	Receives limit threshold 2 enabling commands	1.002 DPT_Bool
10	42	74	106	IN.X - Limit threshold 2 enabling	1=enable / 0=disable	Receives limit threshold 2 enabling commands	1.002 DPT_Bool
12	44	76	108	IN.X - Limit threshold 2 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the limit threshold 2 value	1.007 DPT_Step
12	44	76	108	IN.X - Limit threshold 2 value input	4 byte floating value	Receives the limit threshold 3 value	14.005 DPT_Value_Amplitude
16	48	80	112	IN.X - Limit threshold 3 enabling	0=enable / 1=disable	Receives limit threshold 3 enabling commands	1.002 DPT_Bool
16	48	80	112	IN.X - Limit threshold 3 enabling	1=enable / 0=disable	Receives limit threshold 3 enabling commands	1.002 DPT_Bool
18	50	82	114	IN.X - Limit threshold 3 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the limit threshold 3 value	1.007 DPT_Step
18	50	82	114	IN.X - Limit threshold 3 value input	4 byte floating value	Receives the limit threshold 3 value	14.005 DPT_Value_Amplitude
22	54	86	118	IN.X - Limit threshold 4 enabling	0=enable / 1=disable	Receives limit threshold 4 enabling commands	1.002 DPT_Bool
22	54	86	118	IN.X - Limit threshold 4 enabling	1=enable / 0=disable	Receives limit threshold 4 enabling commands	1.002 DPT_Bool
24	56	88	120	IN.X - Limit threshold 4 value regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the limit threshold 4 value	1.007 DPT_Step
24	56	88	120	IN.X - Limit threshold 4 value input	4 byte floating value	Receives the limit threshold 4 value	14.005 DPT_Value_Amplitude

10.5 Communication objects for “Thermostat x”

Communication objects with **output** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
1	33	65	97	IN.x - Functioning type	Heating/Air cooling	Receive commands of functioning type setting and send feedback about functioning type set	1.100 DPT_Heat/Cool
2	34	66	98	IN.x - Heating anti-freeze setpoint feedback	Value °C	Receives the value and sends the feedback of heating building protection HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
2	34	66	98	IN.x - Heating anti-freeze setpoint feedback	°K value	Receives the value and sends the feedback of heating building protection HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
2	34	66	98	IN.x - Heating anti-freeze setpoint feedback	°F value	Receives the value and sends the feedback of heating building protection HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
3	35	67	99	IN.x - Heating economy setpoint feedback	Value °C	Receives the value and sends the feedback of heating economy HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
3	35	67	99	IN.x - Heating economy setpoint feedback	°K value	Receives the value and sends the feedback of heating economy HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
3	35	67	99	IN.x - Heating economy setpoint feedback	°F value	Receives the value and sends the feedback of heating economy HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
4	36	68	100	IN.x - Heating precomfort setpoint feedback	Value °C	Receives the value and sends the feedback of heating precomfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
4	36	68	100	IN.x - Heating precomfort setpoint feedback	°K value	Receives the value and sends the feedback of heating precomfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
4	36	68	100	IN.x - Heating precomfort setpoint feedback	°F value	Receives the value and sends the feedback of heating precomfort HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
5	37	69	101	IN.x - Heating comfort setpoint feedback	Value °C	Receives the value and sends the feedback of heating comfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
5	37	69	101	IN.x - Heating comfort setpoint feedback	°K value	Receives the value and sends the feedback of heating comfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
5	37	69	101	IN.x - Heating comfort	°F value	Receives the value and	9.027 DPT_Temp_F

				setpoint feedback		sends the feedback of heating comfort HVAC mode setpoint values in Fahrenheit degrees	
6	38	70	102	IN.x - Air cond. high tem. prot. setpoint feedback	Value °C	Receives the value and sends the feedback of air cooling building protection HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
6	38	70	102	IN.x - Air cond. high tem. prot. setpoint feedback	°K value	Receives the value and sends the feedback of air cooling building protection HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
6	38	70	102	IN.x - Air cond. high tem. prot. setpoint feedback	°F value	Receives the value and sends the feedback of air cooling building protection HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
7	39	71	103	IN.x - Air cooling economy setpoint feedback	Value °C	Receives the value and sends the feedback of air cooling economy HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
7	39	71	103	IN.x - Air cooling economy setpoint feedback	°K value	Receives the value and sends the feedback of air cooling economy HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
7	39	71	103	IN.x - Air cooling economy setpoint feedback	°F value	Receives the value and sends the feedback of air cooling economy HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
8	40	72	104	IN.x - Air cooling precomfort setpoint feedback	Value °C	Receives the value and sends the feedback of air cooling precomfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
8	40	72	104	IN.x - Air cooling precomfort setpoint feedback	°K value	Receives the value and sends the feedback of air cooling precomfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
8	40	72	104	IN.x - Air cooling precomfort setpoint feedback	°F value	Receives the value and sends the feedback of air cooling precomfort HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
9	41	73	105	IN.x - Air cooling comfort setpoint feedback	Value °C	Receives the value and sends the feedback of air cooling comfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
9	41	73	105	IN.x - Air cooling comfort setpoint feedback	°K value	Receives the value and sends the feedback of air cooling comfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
9	41	73	105	IN.x - Air cooling comfort setpoint feedback	°F value	Receives the value and sends the feedback of air cooling comfort HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
10	42	74	106	IN.x - HVAC mode feedback	Eco/Precom/Comf/Off	Sends feedback about HVAC mode set	20.102 DPT_HVACMode
11	43	75	107	IN.x - Current setpoint feedback	Value °C	Sends active setpoint values in degrees Celsius	9.001 DPT_Temp

11	43	75	107	IN.x - Current setpoint feedback	°K value	Sends active setpoint values in degrees Kelvin	9.002 DPT_Tempd
11	43	75	107	IN.x - Current setpoint feedback	°F value	Sends active setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F
12	44	76	108	IN.x - Measured temperature	Value °C	Send temperature values in Celsius degrees	9.001 DPT_Temp
12	44	76	108	IN.x - Measured temperature	°K value	Send temperature values in Kelvin degrees	9.002 DPT_Tempd
12	44	76	108	IN.x - Measured temperature	°F value	Send temperature values in Fahrenheit degrees	9.027 DPT_Temp_F
16	48	80	112	IN.x - Heating valve switch	On/Off	Send heating electrovalve activation/deactivation commands	1.001 DPT_Switch
16	48	80	112	IN.x - Heating/Air cooling valve switch	On/Off	Sends heating/air cooling electrovalve activation/deactivation commands	1.001 DPT_Switch
18	50	82	114	IN.x - V1 fan switching heating	On/Off	Sends heating fancoil speed 1 electrovalve activation/deactivation commands	1.001 DPT_Switch
20	52	84	116	IN.x - V2 fan switching heating	On/Off	Sends commands of activation/deactivation fancoil speed 2 heating	1.001 DPT_Switch
22	54	86	118	IN.x - V3 fan switching heating	On/Off	Sends commands of activation/deactivation fancoil speed 3 heating	1.001 DPT_Switch
24	56	88	120	IN.x - Air cooling valve switch	On/Off	Send air cooling electrovalve activation/deactivation commands	1.001 DPT_Switch
26	58	90	122	IN.x - V1 fan switching air cooling	On/Off	Sends air cooling fancoil speed 1 electrovalve activation/deactivation commands	1.001 DPT_Switch
28	60	92	124	IN.x - V2 fan switching air cooling	On/Off	Sends commands of activation/deactivation fancoil speed 2 air cooling	1.001 DPT_Switch
30	62	94	126	IN.x - V3 fan switching air cooling	On/Off	Sends commands of activation/deactivation fancoil speed 3 air cooling	1.001 DPT_Switch

Communication objects with **input** functions:

#				Object name	Object function	Description	Datapoint type
IN 1	IN 2	IN 3	IN 4				
0	32	64	96	IN.x - HVAC mode input	Eco/Precom/Comf/Off	Receives HVAC mode setting commands	20.102 DPT_HVACMode
0	32	64	96	IN.x - Setpoint input	Value °C	Receives functioning setpoint values in degrees Celsius	9.001 DPT_Temp
0	32	64	96	IN.x - Setpoint input	°K value	Receives functioning setpoint values in degrees Kelvin	9.002 DPT_Tempd
0	32	64	96	IN.x - Setpoint input	°F value	Receives functioning setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F
1	33	65	97	IN.x - Functioning type	Heating/Air cooling	Receive commands of functioning type setting and send feedback about functioning type set	1.100 DPT_Heat/Cool
2	34	66	98	IN.1 - Heating anti-freeze setpoint	Value °C	Receives the value of heating building protection HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
2	34	66	98	IN.1 - Heating anti-freeze setpoint	°K value	Receives the value of heating building protection HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
2	34	66	98	IN.1 - Heating anti-freeze setpoint	°F value	Receives the value of heating building protection HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
3	35	67	99	IN.x - Heating economy setpoint	Value °C	Receives the value of heating economy HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
3	35	67	99	IN.x - Heating economy setpoint	°K value	Receives the value of heating economy HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
3	35	67	99	IN.x - Heating economy setpoint	°F value	Receives the value of heating economy HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
4	36	68	100	IN.x - Heating precomfort setpoint feedback	Value °C	Receives the value and sends the feedback of heating precomfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
4	36	68	100	IN.x - Heating precomfort setpoint	°K value	Receives the value of heating precomfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
4	36	68	100	IN.x - Heating precomfort setpoint	°F value	Receives the value of heating precomfort HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
5	37	69	101	IN.x - Heating comfort setpoint	Value °C	Receives the value of heating comfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
5	37	69	101	IN.x - Heating comfort setpoint	°K value	Receives the value of heating comfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
5	37	69	101	IN.x - Heating comfort setpoint	°F value	Receives the value of heating comfort HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
6	38	70	102	IN.x - Air cond. high tem. prot. setpoint	Value °C	Receives the value of air cooling building protection HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
6	38	70	102	IN.x - Air cond. high tem.	°K value	Receives the value of air	9.002 DPT_Tempd

				prot. setpoint		cooling building protection HVAC mode setpoint values in Kelvin degrees	
6	38	70	102	IN.x - Air cond. high tem. prot. setpoint	°F value	Receives the value of air cooling building protection HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
7	39	71	103	IN.x - Air cooling economy setpoint	Value °C	Receives the value of air cooling economy HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
7	39	71	103	IN.x - Air cooling economy setpoint	°K value	Receives the value of air cooling economy HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
7	39	71	103	IN.x - Air cooling economy setpoint	°F value	Receives the value of air cooling economy HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
8	40	72	104	IN.x - Air cooling precomfort setpoint	Value °C	Receives the value of air cooling precomfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
8	40	72	104	IN.x - Air cooling precomfort setpoint	°K value	Receives the value of air cooling precomfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
8	40	72	104	IN.x - Air cooling precomfort setpoint	°F value	Receives the value of air cooling precomfort HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
9	41	73	105	IN.x - Air cooling comfort setpoint	Value °C	Receives the value of air cooling comfort HVAC mode setpoint values in Celsius degrees	9.001 DPT_Temp
9	41	73	105	IN.x - Air cooling comfort setpoint	°K value	Receives the value of air cooling comfort HVAC mode setpoint values in Kelvin degrees	9.002 DPT_Tempd
9	41	73	105	IN.x - Air cooling comfort setpoint	°F value	Receives the value of air cooling comfort HVAC mode setpoint values in Fahrenheit degrees	9.027 DPT_Temp_F
13	45	77	109	IN.x - Thermostat function scene	Execute/Store	Receives thermostat function scene execute/learn commands	18.001 DPT_SceneControl
14	46	78	110	IN.x - Window status input	1 = open/0 = closed	Receive window contact status	1.019 DPT_Window_Door
15	47	79	111	IN.x - Heating valve status feedback	On/Off status	Receive feedbacks on electrovalve activation status heating	1.001 DPT_Switch
15	47	79	111	IN.x - Heating/air cooling valve status feedback	On/Off status	Receives feedbacks on electrovalve activation status heating/air cooling	1.001 DPT_Switch
17	49	81	113	IN.x - Heating fan V1 status feedback	On/Off status	Receives feedbacks on fancoil speed 1 activation status heating	1.001 DPT_Switch
19	51	83	115	IN.x - Heating fan V2 status feedback	On/Off status	Receives feedbacks on heating fancoil speed 2 activation status	1.001 DPT_Switch
21	53	85	117	IN.x - Heating fan V3 status feedback	On/Off status	Receives feedbacks on heating fancoil speed 3 activation status	1.001 DPT_Switch
23	55	87	119	IN.x - Air cooling valve status feedback	On/Off status	Receive feedbacks on electrovalve activation status air cooling	1.001 DPT_Switch
25	57	89	121	IN.x - Air cooling fan V1 status feedback	On/Off status	Receives feedbacks on fancoil speed 1 activation	1.001 DPT_Switch

						status air cooling	
27	59	91	123	IN.x - Air cooling fan V2 status feedback	On/Off status	Receives feedbacks on air cooling fancoil speed 2 activation status	1.001 DPT_Switch
29	61	93	125	IN.x - Air cooling fan V3 status feedback	On/Off status	Receives feedbacks on air cooling fancoil speed 3 activation status	1.001 DPT_Switch
31	63	95	127	IN.x - KNX external sensor input	Value °C	Receives external KNX sensor values in degrees Celsius	9.001 DPT_Temp
31	63	95	127	IN.x - KNX external sensor input	°K value	Receives external KNX sensor values in degrees Kelvin	9.002 DPT_Tempd
31	63	95	127	IN.x - KNX external sensor input	°F value	Receives external KNX sensor values in degrees Fahrenheit	9.027 DPT_Temp_F

10.6 Communication objects for "Digital output x"

Communication objects with **output** functions:

#				Object name	Object function	Description	Datapoint type
CH 1	CH 2	CH 3	CH 4				
230	234	238	242	CH.x - Digital output status	On/Off status	Sends the digital output status	1.001 DPT_Switch
231	235	239	243	CH.x - Digital output % value	% value	Sends the digital output % actual value	5.001 DPT_Percentage

Communication objects with **input** functions:

#				Object name	Object function	Description	Datapoint type
CH 1	CH 2	CH 3	CH 4				
228	232	236	240	CH.x - Digital output switch	On/Off	Receives digital output activation/deactivation commands	1.001 DPT_Switch
229	233	237	241	CH.x - Digital output % command	% value	Receives the percentage values of digital output regulation	5.001 DPT_Percentage

11 ETS programming error feedback

The device is able to detect and therefore signal various programming errors following the download of the application from the ETS, via the blinking (8 Hz) of the amber LED associated with the input:

Possible errors
The constraints between the setpoints of the various HVAC modes belonging to the same functioning type are not respected: - $T_{anti-freeze} \leq T_{economy} \leq T_{precomfort} \leq T_{comfort}$ in heating mode - $T_{comfort} \leq T_{precomfort} \leq T_{economy} \leq T_{high\ temp.\ protection}$ in air cooling mode or if the control type is setpoint, the constraints are - $T_{anti-freeze} \leq T_{operation}$ in heating mode - $T_{operation} \leq T_{high\ temp.\ protection}$ in air cooling mode Connection error of the communication objects dedicated to sending commands to the actuation devices (valves and fancoils): - if different control logic is enabled but command objects for the heating and air cooling solenoid valves are connected in the same group address - Coherency between the connection of the speeds from the fancoil. With regard to the command objects for 1 bit fancoil speeds, they can be linked to the same address or to different addresses, however it is important to maintain coherency between the pairs: if the two objects for controlling heating and air cooling speed 1 are linked to two addresses that are the same, also the pairs for speeds 2 and 3 must be the linked to two addresses that are the same. In this manner, if linked to two different addresses, also the pairs for speeds 2 and 3 must be linked to two different addresses. If a solenoid valve is connected (heating or air cooling), also the respective speeds must be connected, and vice versa. In common logic mode, and if connecting the heating fan speeds, they must also be connected for air cooling, and vice versa.

If multiple errors are detected, the error signalling (amber LED blinking) will continue until the ETS application is downloaded again with the necessary corrections.

code 7.01.7.750.6

LAST REVISION 05/2015