

CHORU **S** MART

**GEWISS**

KNX/Easy THERMO ICE - flush mounting



**GW 16974CB**  
**GW 16974CL**  
**GW 16974CN**  
**GW 16974CT**

**Technical manual**

# Contents

1	Introduction .....	3
2	Application.....	3
2.1	Association limits.....	4
3	“Main” menu .....	5
3.1	Parameters.....	5
3.2	Control algorithms .....	19
4	“Heating” menu .....	28
4.1	Parameters.....	28
5	“Air cooling” menu.....	35
5.1	Parameters.....	35
6	“Temperature setpoint” menu .....	42
6.1	Parameters.....	42
7	“Temperature sensors” menu .....	47
7.1	Parameters.....	47
8	“Feedback” menu.....	53
8.1	Parameters.....	53
9	“Scenes” menu.....	58
9.1	Parameters.....	58
10	“Humidity” menu.....	60
10.1	Parameters.....	60
11	“Relative humidity threshold x” menu .....	63
11.1	Parameters.....	63
12	“Thermal comfort” menu .....	71
12.1	Parameters.....	71
13	“Display and touch push-button” menu.....	74
13.1	Parameters.....	76
14	“Input aux 1” and “Input aux 2” menu .....	78
14.1	Parameters.....	79
14.2	“Edges/Sequence commands” function .....	82
14.3	Matched function: “1 push-button + stop dimmer” .....	90
14.4	Matched function: “Cyclic sending 1 push-button dimmer” .....	92
14.5	Matched function: “1 push-button roller shutter control” .....	93
14.6	Matched function: “2 push-button + stop dimmer” .....	95
14.7	Matched function: “cyclical sending 2 push-button dimmer” .....	96
14.8	Matched function: “2 push-button roller shutters control” .....	97
14.9	Matched function: “scene management” .....	98
14.10	Matched function: “window contact” .....	99
15	“Logic” menu .....	101
16	ETS programming error feedback.....	104
17	Communication objects.....	105

# 1 Introduction

This manual describes the functions of the “**KNX/Easy THERMO ICE - flush-mounting**” (GW16974CB - GW16974CN - GW16974CT) device and how they are set and configured with the aid of the ETS configuration software.

## 2 Application

The KNX/Easy THERMO ICE flush-mounting is used to manage a humidification/dehumidification system in parallel with the temperature adjustment system, or to interact with a temperature adjustment system to prevent humidity formation. The temperature and humidity are regulated by commanding - on a KNX BUS - the KNX actuators that control the air heating or cooling elements (including the fancoils) and the humidification/dehumidification elements.

The thermostat can work in "autonomous" control mode, to autonomously manage the temperature adjustment system (or parts of it); whereas when combined with a master device (e.g: a KNX timed thermostat), it can work in "Slave" control mode to create multi-area temperature adjustment systems. The Setpoint values used by the thermostat are the ones configured via ETS, and they can be modified locally and via the BUS (if these options were enabled during the ETS configuration).

The thermostat offers:

- 2 types of operation: heating and cooling with independent control algorithms;
- 4 operating modes: OFF (anti-freeze / high temperature protection), Economy, Pre-comfort and Comfort
- 4 heating adjustment temperatures (Teconomy, Tpre-comfort, Tcomfort, Tantigelo (Tanti-freeze))
- 4 cooling adjustment temperatures (Teconomy, Tprecomfort, Tcomfort, Tprotection\_high\_temperatures);
- 2 control modes: Slave (if combined with a Master device) or autonomous
- 2 types of control: HVAC or Setpoint
- 2 control stages: single stage (with single switchover command) or dual stage (with dual switchover command, for systems with a high degree of thermal inertia)
- 2-way or 4-way system control algorithms (first stage): 2 points (ON/OFF command or 0%/100%), proportional PI (PWM type control or continuous), fan coil (max. 3 speeds);
- control algorithms (second stage): 2 points (ON/OFF control or 0% / 100%);
- 1 input for a potential-free contact (e.g. a window contact, or as a general input with command function on the BUS)
- 1 input that can be configured for a NTC external temperature probe sensor (e.g. protection sensor for underfloor heating) or, alternatively, a potential-free contact.

The thermostat is equipped with a white LED backlit display with sensitive rear-projected areas on a glass plate. The device requires an external 12-24Vac/dc power supply and has an integrated sensor for measuring the ambient temperature (whose value is sent on the BUS with a parameterised frequency or following a temperature variation) and a proximity sensor for the activation of the backlight when the user approaches the device. The thermostat can be configured with ETS or with the Easy Controller.

The device can be configured with the ETS software, to perform the following functions:

#### **Temperature control**

- at 2 points, with ON/OFF commands or 0% / 100% commands
- integral proportional control, with PWM commands or continuous adjustment (0% ÷ 100%).

#### **Fancoil management**

- fancoil speed control with ON/OFF selection commands;
- management of 2-way or 4-way systems, with ON/OFF commands or 0% / 100% commands.

#### **Operating mode setting**

- from the BUS, with distinct 1-bit objects (OFF, ECONOMY, PRE-COMFORT, COMFORT)
- from the BUS, with a 1-byte object.

#### **Operating setpoint setting**

- from the BUS, with a 2-byte object.

#### **Temperature measurement**

- with a built-in sensor
- mixed built-in sensor / external KNX sensor/external NTC sensor with definition of the relative weight;

#### **Relative humidity measurement**

- relative humidity measurement obtained from an external KNX sensor
- setting of up to 5 relative humidity thresholds, with BUS commands sent when the threshold is exceeded and restored:
  - 1-bit, 2-bit, 1-byte commands to act on the humidification/dehumidification system
  - HVAC mode commands to produce a retroactive effect on the heating/cooling system
  - set-point values to produce a retroactive effect on the heating/cooling system
- calculation of specific humidity
- indication of the thermal well-being status

#### **Underfloor probe**

- setting of threshold value for floor temperature alarm.

#### **Temperature control for specific zones**

In "Slave" control mode:

- with the operating mode received by the Master device, and the use of a local Setpoint
- with the Setpoint value received by the Master device, and local residual current device for temperature

In "autonomous" control mode:

- with local selection of operating mode and setpoints
- with local selection of operating Setpoint

#### **Scenes**

- memorisation and activation of 8 scenes (value 0..63)

#### **Other functions**

- setting of the Setpoint (OFF, ECONOMY, PRE-COMFORT, COMFORT) from the BUS
- setting of the operating Setpoint from the BUS
- setting of the type of operation (heating/cooling) from the BUS
- transmission of the status information (mode, type), measured temperature and current Setpoint on the BUS;
- management of the status information arriving from the commanded actuator
- management of the status signalling window for temporarily switching off the thermostat
- auxiliary input for edge management, brief/prolonged operation, dimmer with single or double push-button, roller shutters with single or double push-button, scenes and window contact
- management of display parameters

## **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 parameters used to enable the different functions implemented by the device and to set the main operating parameters. If programming the device with ETS, selecting with the parameter **Programming mode = “ETS mode”**, the structure of the menu that appears is shown in fig. 3.1:

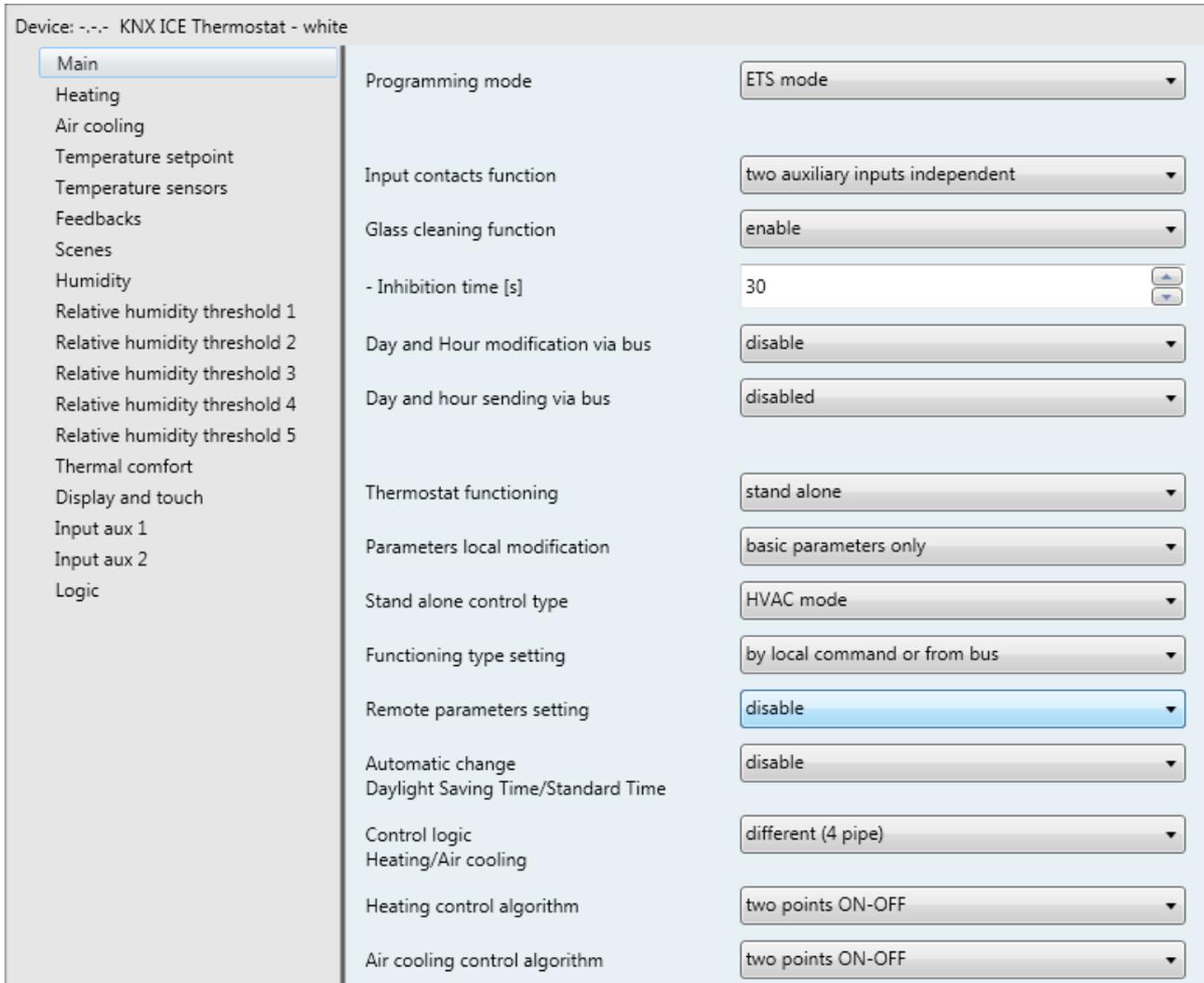


Fig. 3.1

#### 3.1 Parameters

##### ➤ 3.1.1 Input contacts function

The device has two input channels (two terminals plus common) for connecting the potential-free contacts. Each of the 2 input channels activated from the device can be autonomously managed by performing an autonomous function, or they can be combined to perform a common function. Alternatively, the second channel can be used to connect an external NTC sensor that can be used as an external temperature probe or as an underfloor probe. The **“Input contacts function”** parameter defines how the two input channels are managed.

The values that can be set are:

- **two auxiliary inputs independent (default value)**
- one auxiliary input and one temperature sensor

The structure of the input configuration menu will vary according to the values set for the above parameter: an independent setting menu for each channel (**Auxiliary input 1** and **Auxiliary input 2**) if the set value is **two independent auxiliary inputs**, or a single menu (**Auxiliary input 1**) and the new parameter “**External auxiliary sensor**” in the **Temperature sensors** menu if the value is **one auxiliary input and one temperature sensor**.

### ➤ 3.1.2 Glass cleaning function

The “**Glass cleaning function**” parameter enables the inhibition of the capacitive sensors to allow the cleaning of the glass surface. The values that can be set are:

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

Selecting **enable** displays the parameters “**Inhibition time [s]**” and the communication object **Glass cleaning** (Data Point Type: 1.010 DPT\_Start).

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

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

While the cleaning function is active, all the push-buttons and touch sliders are deactivated, and the inhibition time countdown is shown on the display.

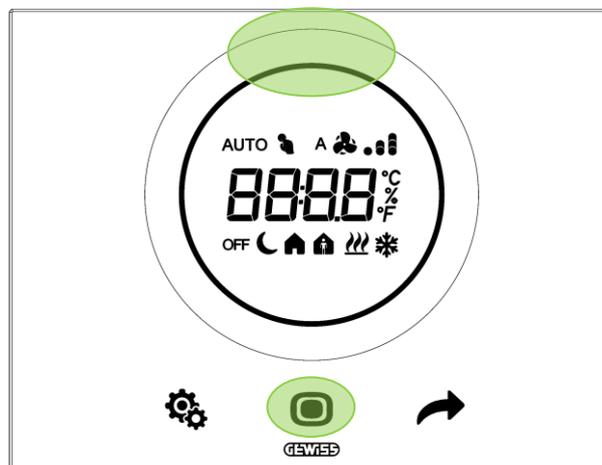
The exact moment of the cleaning function activation and deactivation is signalled to the user via a special sound effect (used only with this function) and with the reproduction of a special light effect (used only with this function) that concerns the backlighting of the touch push-buttons and sliders.

There are 2 ways of activating the glass cleaning function:

- manual activation:

#### **Enabling**

Simultaneous touch of the north sector of the circular slider and the MODE button key > 3 seconds (regardless of the order)



Short beep (if the acoustic signal for glass cleaning is enabled)

Contemporary flashing of the touch push-buttons (if the luminous signal for glass cleaning has been enabled)

Stopping of blinking and display of the countdown on the display

### **Disabling**

Inhibition time timeout

Short beep (if the acoustic signal for glass cleaning is enabled)

Contemporary flashing of the touch push-buttons (if the luminous signal for glass cleaning has been enabled)

Stop of flashing and resetting of previous signals on the display

- BUS commands:

via the **Glass cleaning** communication object (*1.010 DPT\_Start*), it is possible to launch the timed activation of the cleaning function via a BUS command with a logic value of “1”; the function autonomously deactivates when the time period in the “**Inhibition time [s]**” parameter has elapsed or via a BUS command with a logic value of “0”. The arrival of a BUS command with a logic value of “1” and active timing leads to the resetting of the sensor inhibition time.

### ➤ **3.1.3 Day and hour modification via BUS**

This is used to change the hour and day of the week via BUS telegrams. The values that can be set are:

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

Selecting **enable** displays the **Day/hour input** communication object (Data Point Type: 10.001 DPT\_TimeOfDay) which makes it possible for the device to receive the values relative to the day of the week and the hour via the BUS. The information about the day of the week is not used by the device.

The status read request should be sent via the **Day/hour input** object when the BUS voltage is restored, in order to update the time shown.

### ➤ **3.1.4 Day and hour sending via BUS**

The parameter “**Day and hour sending via bus**” is used to enable or disable sending the day and hour of the day currently set on the device to other KNX system devices.

The values that can be set are:

- **disabled** (default value)
- periodic
- after a blackout
- after a blackout or a local modification
- after a blackout, local modification and periodically

selecting any value other than disabled displays the communication object **Day/hour sending** (Data Point Type: 10.001 DPT\_TimeOfDay) which allows the day/hour to be transmitted on the BUS according to the criteria set for the parameter in question.

Setting day/hour sending *after a blackout*, the device sends the BUS telegrams with the current hour and day after the BUS or auxiliary power supply voltage was restored.

Setting *after a modification*, after the day/hour was modified on the device (from the local SET navigation menu, for the various access levels (basic/advanced) consult the programming manual (can be downloaded from [www.gewiss.com](http://www.gewiss.com)).

it sends the new values on the BUS; if the hour is changed after receiving a BUS telegram on the **Day/hour input** object, this update is not signalled by the thermostat to avoid redundancies.

Setting *periodically*, the device sends the values with a periodic frequency, according to the period defined by the value set for the parameter “**Day/hour sending period**”.

The parameter may have the following values:

- 6 hours
- 12 hours
- 1 day
- 2 days
- 3 days
- 4 days
- 5 days
- 6 days
- **1 week** (default value)

### ➤ 3.1.5 Thermostat functioning

The “**Thermostat functioning**” parameter defines the function of the thermostat within the KNX temperature adjustment system. The values that can be set are:

- **stand alone** (default value)  
The device autonomously manages the temperature adjustment system (or part of it) without the aid of the timed thermostats connected to it that control parts of the system. With this configuration, there is a single temperature control centre for the environment.  
Selecting this value displays the “**Stand alone control type**” and “**Remote parameter setting**” parameters.
- slave  
The device is configured so it can manage the temperature adjustment system with the aid of a Master device as a timed thermostat. With this configuration the device does not control the entire system but only a part of it, called a zone, while there is a Master device in the system that controls its mode and functioning type. In this case, the thermostat controls the temperature of the environment where it is located whereas it is the Master device that decides the functioning set by the user. It is not possible to modify the HVAC mode of the device locally.  
Selecting this value displays the “**Master/slave control type**”, “**Temporary setpoint forcing via bus**” and “**Allows local switching off**” parameters, and the **Functioning type input** (Data Point Type: 1.100 DPT\_Heat/Cool) and **Slave function enabling** communication objects (Data Point Type: 1.003 DPT\_Enable).

The **Slave function enabling** communication object is used to modify the thermostat operating mode between stand alone (value 0) and Slave (value 1) via a BUS telegram.

### ➤ 3.1.6 Parameters local modification

This is used to enable or disable changing the thermostat operating parameters via the relative local menu. The values that can be set are:

- disabled
- **basic parameters only** (default value)
- basic and advanced parameters

For the various access levels (basic/advanced) consult the programming manual (can be downloaded from [www.gewiss.com](http://www.gewiss.com)).

### ➤ 3.1.7 Stand alone control type

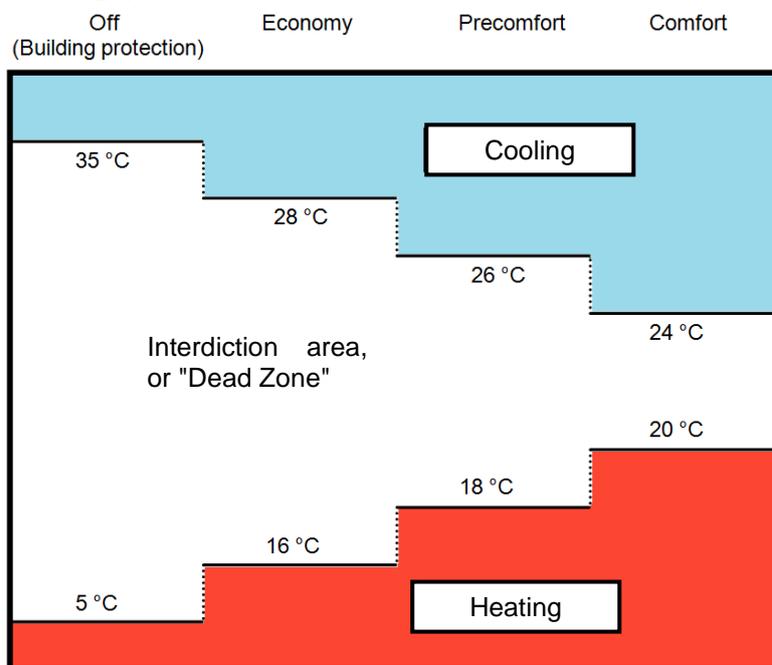
If the device functions as "stand alone", it is possible to define whether only one operating setpoint should be managed or whether the operation should be managed via the HVAC modes. The **"Stand alone control type"** parameter is used to set the type of control you want to use. The values that can be set are:

- HVAC mode (default value)
- Setpoint

Selecting the **setpoint** value displays the **"Allows local switching off"** parameter.

### ➤ 3.1.8 Functioning type setting

The temperature adjustment functioning type (heating or air cooling) can be managed manually, or autonomously by the device itself. The manual method can be managed via the local navigation menu or BUS commands that allow switching between types, changing the dedicated parameter. The automatic method is based on the principle of an interdiction area, or "dead zone", i.e. the temperature gap between the set-points of the HVAC heating and air cooling modes (see figure below), allowing the automatic switchover from one functioning type to the other.



The figure shows that as long as the measured temperature is below the heating set-point, the functioning type is HEATING; in the same manner, if the measured value is higher than the air cooling set-point, then the functioning type is AIR COOLING. If the measured value is within the interdiction area, the functioning type remains as before. The HEATING →AIR COOLING switchover point corresponds to the setpoint of the HVAC mode related to air cooling, and in the same way the AIR COOLING →HEATING switchover corresponds to the heating setpoint. In any case, it is possible to force a functioning type via the local menu or via a BUS command.

Regardless of the functioning type set (stand alone or Slave), the **"Functioning type setting"** parameter defines how the functioning type is modified. The values that can be set are:

- automatically (interdiction area "dead zone") (default value)
- by local command or from BUS

In the case of Slave operation, the **Enable dead zone** communication object will be displayed (Data Point Type: 1.003 DPT\_Enable) which is used to enable or disable the dead zone. When the dead zone is disabled via a BUS command on the **Enable dead zone** object, the active functioning type remains the one

set automatically and can be modified via a BUS command on the **Functioning type input** object (Data Point Type: 1.100 DPT\_Heat/Cool).

### ➤ **3.1.9 Master/slave control type and Stand alone control type**

If the device functions as a Slave, it can be controlled by the Master device via a single setpoint or by changing the active HVAC mode. The “**Master/slave control type**” parameter is used to set the type of control the Master device executes on the thermostat.

The values that can be set are:

- **HVAC mode** (default value)
- setpoint

Selecting **HVAC mode** displays the “**Format of the commands for setting HVAC mode**” and “**Change HVAC mode setpoint via BUS**” parameters. Selecting **setpoint**, the “**Modify functioning setpoint via bus through**” parameter is made visible.

If the device functions as "stand alone", it is possible to define whether only one operating setpoint should be managed or whether the operation should be managed via the HVAC modes. The “**Stand alone control type**” parameter is used to set the type of control you want to use. The values that can be set are:

- **HVAC mode** (default value)
- setpoint

Selecting the setpoint value displays the “**Allows local switching off**” parameter.

### ➤ **3.1.10 Remote parameters setting**

Even if the device functions as stand alone, it can still receive various commands from remote units via the BUS; the parameter “**Remote parameters setting**” displays the various configuration items for the remote control of the thermostat. The values that can be set are:

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

Selecting **enable** displays the “**Format of commands for setting HVAC mode**” and “**Modify setpoint of HVAC modes via bus**” parameters (if the control type is HVAC mode) or the “**Modify functioning setpoint via bus through**” parameter (if the control type is setpoint), and the “**Temporary forcing of setpoint via bus**” and “**Functioning type**” parameters.

The “**Format of the commands for setting the HVAC mode**” parameter is used to define the format of the communication objects used for the remote control of the thermostat HVAC mode. The values that can be set are:

- 1 bit
- **1 byte** (default value)
- both

Selecting **1 byte** or **both** displays the **HVAC mode input** communication object (Data Point Type: 20.102 DPT\_HVACMode) via which it is possible to modify the HVAC mode via BUS through a single command; selecting **1 bit** or **both** displays the parameter “**Priority among remote HVAC mode setting objects**” and the communication objects **HVAC off mode input**, **HVAC economy mode input**, **HVAC precomfort mode input**, **HVAC comfort mode input**(Data Point Type: 1.003 DPT\_Enable) which are used to activate the relative HVAC mode.

### ➤ 3.1.11 Priority among remote HVAC mode setting objects

The parameter “**Priority among remote HVAC mode setting objects**” is used to define if there is a priority constraint between the communication objects dedicated to remote HVAC mode setting. The values that can be set are:

- disable

the 1 bit communication objects **HVAC off mode input**, **HVAC economy mode input**, **HVAC precomfort mode input**, **HVAC comfort mode input** and the 1 byte communication object **HVAC mode input** all have the same priority, the last command received is the one that determines the HVAC mode to be activated; this means that receiving the value “0” on the 1 bit communication object that had activated a certain HVAC mode is ignored and the mode remains active. With this setting, the priority among the different device functions and the different communication objects that can be used for setting the HVAC mode is as follows:

Function/object	Size	Priority	
Reactivation after manual switch-off (if manual reactivation and remote controls)	-	1	low
Change operating mode locally (Mode button key)	-	2	
Scene	1 byte	2	
HVAC mode input (setpoint input)	1 byte (2 byte)	2	
HVAC comfort mode input	1 bit	2	
HVAC precomfort mode input	1 bit	2	
HVAC economy mode input	1 bit	2	
HVAC off mode input	1 bit	2	
Reactivation after manual switch-off (if only manual reactivation)		3	
Manual switch-off		4	
Window contact function aux input	-	5	

- enable (default value)

the 1 bit communication objects **HVAC off mode input**, **HVAC economy mode input**, **HVAC precomfort mode input**, **HVAC comfort mode input** and the 1 byte communication object **HVAC mode input** all have different priorities, the command with the highest priority will determine the HVAC mode to be activated. This setting displays the parameters “**Reception of 1 bit mode activation with priority > current**”, “**Reception of 1 bit mode activation with priority < current**” and the priority between the different functions of the device and the different communication object that can be used for setting the HVAC mode is as follows:

Function/Object	Size	Priority	
Reactivation after manual switch-off (if manual reactivation and remote controls)	-	1	low
Change operating mode locally (Mode button key)	-	2	
Scene	1 byte	2	
HVAC mode input (setpoint input)	1 byte (2 byte)	2	
HVAC comfort mode input	1 bit	3	
HVAC precomfort mode input	1 bit	4	
HVAC economy mode input	1 bit	5	
HVAC off mode input	1 bit	6	
Reactivation after manual switch-off (if only manual reactivation)		7	
Manual switch-off		8	
Window contact function aux input	-	9	

The 1 bit mode setting objects all have a higher priority than the 1 byte mode setting object; this is due to the fact that, by enabling both the mode setting possibilities, the 1 bit objects can be used to fix the mode if particular events occur.

Naturally there is also a command execution priority between the 1-bit mode setting objects, especially because if only the 1-bit format is set for setting the functioning mode and if multiple objects are enabled, it is necessary to determine which of these have greater priority to determine the active functioning mode on the device. Due to the fact that multiple 1-bit objects can be enabled at the same time, it is possible to define the behaviour of the device when a communication object is received with a higher priority than

the one currently active at that moment using the “**Reception of 1 bit mode activation with priority > current**” parameter. In the same manner, the behaviour of the device can be defined when a communication object is received with a lower priority than the one currently active at that moment using the parameter “**Reception of 1 bit mode activation with priority < current**”.

The values that can be set for the parameter “**Reception of 1-bit mode activation with priority > current**” are:

- **maintain value of objects with lower priority** (default value)
- deactivate objects with lower priority

By selecting **maintain value of objects with lower priority**, when receiving a 1-bit HVAC mode activation command with a priority higher than the one currently active, the mode of the new object is set, but the activation status of the objects with a lower priority is maintained; by selecting **deactivate objects with lower priority**, when receiving a 1-bit HVAC mode activation command with a priority higher than the one currently active, the mode of the new object is set and the activation status of the objects with a lower priority is set = 0 (deactivated).

The values that can be set for the “**Reception of 1 bit mode activation with priority < current**” parameter are:

- **update object value** (default value)
- ignore the command

By selecting **update the object value**, when receiving a 1-bit HVAC mode activation command with a priority lower than the one currently active, the mode of the new object is not set, but the activation status is saved; by selecting **ignore the command**, when receiving a 1-bit HVAC mode activation command with priority lower than the one currently active, the new command is ignored (as if it had not been received).

The parameter “**Modify HVAC mode setpoint via BUS**” 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:

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

selecting **enable setting of absolute value** displays the communication objects *Heating antifreeze setpoint input, Heating economy setpoint input, Heating precomfort setpoint input, Heating comfort setpoint input, Air cooling high temperature protection setpoint input, Air cooling economy setpoint input, Air cooling precomfort setpoint input* and *Air cooling comfort setpoint input* (Data Point Type: 9.001 DPT\_Value\_Temp if °C, 9.002 DPT\_Value\_Tempd if °K and 9.027 DPT\_Value\_Temp\_F if °F) through which it is possible to set the setpoints of every device operating mode via BUS.

Selecting the value **enable increase/decrease step regulation** displays the communication objects *Heating antifreeze setpoint regulation, Heating economy setpoint regulation, Heating precomfort setpoint regulation, Heating comfort setpoint regulation, Air cooling high temperature protection setpoint regulation, Air cooling economy setpoint regulation, Air cooling precomfort setpoint regulation* and *Air cooling comfort setpoint regulation* (Data Point Type: 1.007 DPT\_Step) through which it is possible to regulate the setpoints of every mode. Each time value “1” is received, the associated setpoint is incremented by the value defined by the “**Setpoint regulation step [0.1°C]**” parameter; If the value “0” is received, the setpoint is decreased by the value “**Setpoint regulation step [0.1°C]**”.

The **Setpoint regulation step [0.1°C]** parameter defines the increase/decrease step of the setpoints associated with the HVAC modes after receiving a command on the relative regulation objects. The values that can be set are:

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

Independently of the fact if the mode setpoints are modified via the local menu or via a remote command via BUS, if the stand alone or master/slave control type is HVAC mode, there is a value setting limit between the various setpoints belonging to the same functioning type, which is determined by the following relationship:

- $T_{\text{anti-freeze}} \leq T_{\text{economy}} \leq T_{\text{precomfort}} \leq T_{\text{comfort}}$  in heating mode (“T” indicates the general value of the mode setpoint)
- $T_{\text{comfort}} \leq T_{\text{precomfort}} \leq T_{\text{economy}} \leq T_{\text{high temp. protection}}$  in air cooling mode (“T” indicates the general value of the mode setpoint)
- $T_{\text{comfort heating}} \leq T_{\text{comfort air cooling}} - 1^{\circ}\text{C}$  (“T” indicates the generic value of the mode setpoint)

Furthermore, if the dead zone is active, also the following constraint must be respected:

- $T_{\text{comfort heating}} \leq T_{\text{comfort air cooling}} - 1^{\circ}\text{C}$  (“T” indicates the generic value of the mode setpoint)

These constraints must also be respected when a setpoint value is received via the BUS that lies outside of the interval defined by the above relationship, approximating the setpoint to the permitted limit value

If the stand alone or master/slave control type is setpoint, there is a value setting limit that must be respected between the various setpoints belonging to the same functioning type, and it is determined by the following relationship:

- $T_{\text{building protection}} \leq T_{\text{functioning}}$  in heating mode (“T” indicates the general value of the setpoint)
- $T_{\text{functioning}} \leq T_{\text{building protection}}$  in air cooling mode (“T” indicates the general value of the setpoint)

Furthermore, if the dead zone is active, also the following constraint must be respected:

- $T_{\text{heating operation}} \leq T_{\text{air cooling operation}} - 1^{\circ}\text{C}$  (“T” indicates the general value of the setpoint)

These constraints must also be respected when a setpoint value is received via the BUS that lies outside of the interval defined by the above relationship, approximating the setpoint to the permitted limit value.

The “**Modify functioning setpoint via bus through**” parameter is used to select the format of the communication object via which the functioning setpoint is modified when the control type is setpoint. 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 **setting of the absolute value** displays the **Setpoint input** communication object (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 the operating setpoint of the device can be set from the bus.

Selecting **increase/decrease step regulation** displays the “**Setpoint regulation step [0.1°C]**” parameter and the **Setpoint regulation** communication object (Data Point Type: 1.007 DPT\_Step) via which it is possible to regulate the operating setpoint of the device. Each time value “1” is received, the operating setpoint is incremented by the value defined by the “**Setpoint regulation step [0.1°C]**” parameter. If the value “0” is received, the setpoint is decreased by the value “**Setpoint regulation step [0.1°C]**”.

The “**Setpoint regulation step [0.1°C]**” parameter is used to define the increase/decrease step of the operation setpoint after receiving a command on the relative regulation objects. The values that can be set are:

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

As with the local push-buttons, you can temporarily force the current setpoint from the BUS (regardless if the control type is HVAC mode or setpoint) via a specific communication object; the “**Temporary setpoint forcing via bus**” parameter enables the object via which the device receives the temporary setpoint regulation commands. The values that can be set are:

- **disable (default value)**
- enable increase/decrease step regulation

Selecting **enable increase/decrease step regulation** displays the communication object **Temporary setpoint forcing regulation** (Data Point Type: 1.007 DPT\_Step). If the value "1" is received on this object, the current setpoint is increased temporarily by the value defined by parameter "**Setpoint temporary forcing regulation step [0.1 °C]**"; If the value "0" is received on this object, the current setpoint is decreased temporarily by the value defined by "**Setpoint temporary forcing regulation step [0.1°C]**".

The "**Temporary setpoint forcing regulation step [0.1°C]**" parameter defines the increase/decrease step of the current setpoint after receiving a command on the relative regulation object. The values that can be set are:

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

The parameter "**Remote functioning type setting**" is used to enable a specific communication object for the remote control of the functioning type (same priority as the local modification); The values that can be set are:

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

Select **enable** to view the **Functioning type input** object (Data Point Type: 1.100 DPT\_Heat/Cool) which is used to remotely modify the functioning type and, if the functioning type is modified automatically (dead zone), the **Enable dead zone** object will be visible (Data Point Type: 1.003 DPT\_Enable) which is used to enable or disable the dead zone. When the dead zone is disabled via a BUS command on the **Enable dead zone** object, the active functioning type remains the one set automatically and can be modified from the local menu or via a BUS command on the **Functioning type input** object.

If the device operation is slave, it is not possible to modify the HVAC mode locally (if the slave control type is HVAC mode) or modify the operating setpoint above a certain limit (if the slave control type is setpoint); there are some applications, such as for hotels, where the thermostat is requested to function as a slave and that the client can only slightly change the operating setpoint or turn off the device. Turning off the device involves setting the HVAC OFF mode (if the slave control type is HVAC mode) or setting the BUILDING PROTECTION setpoint (if the slave control type is setpoint); the function is activated by pressing the MODE button key. The parameter "**Local switching-off consents**" is used to enable this function; the values that can be set are:

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

Selecting **enable** displays the "**Reactivate device through**" parameter that defines which condition permits the deactivation of device switch-off from the local menu. The values that can be set are:

- **local command only (default value)**
- local command or remote command

Setting **local command only**, all remote commands that involve changing the HVAC mode or changing the operating setpoint received from the Master device will be suspended and implemented when local switch-off is deactivated. Setting **local command or remote commands**, all the remote commands that involve changing the HVAC mode or changing the operating setpoint are implemented and involve deactivating local switch-off. Switching from Slave to stand alone operation or vice versa via a BUS command or the local menu involves deactivating the function.

If the device is stand alone and the control type is setpoint, it is always possible to "switch off" the device (BUILDING PROTECTION setpoint setting) by pressing the MODE button key. As in the case of Slave operation, if the remote commands are enabled (otherwise deactivation takes place via the local menu), the "**Reactivate device through**" parameter is used to define which condition permits deactivating device switch-off from the local menu.

### ➤ 3.1.12 Modify daylight saving time/standard time via BUS

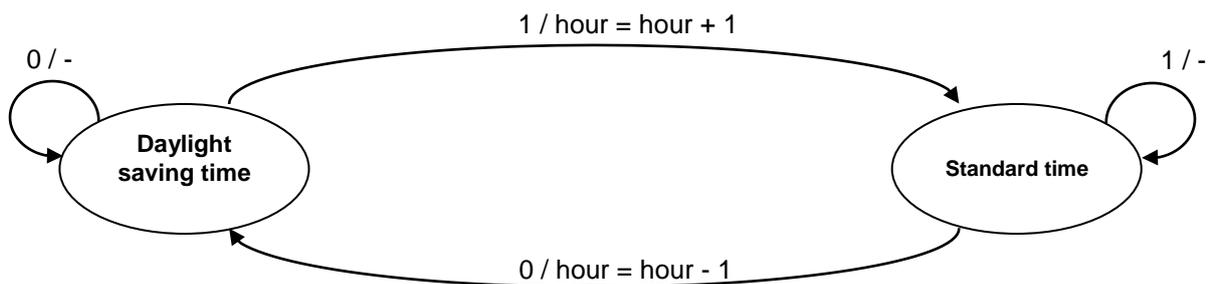
To modify the time automatically during the transit from daylight saving time to standard time, there is a specific communication object that indicates whether which time format is currently being used. The **“Automatic change daylight saving time/standard time”** parameter enables and visualises the **Daylight saving time input** communication object (Data Point Type: 1.001 DPT\_Switch) which the device uses to obtain information about the status of the time being used. The values that can be set are:

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

Selecting **enable** displays the **Daylight saving time input** communication object and the **“Value of Daylight saving time input” object on download** parameter, which defines the time in use when the application is downloaded. The values that can be set are:

- **standard time** (default value)
- daylight saving time

Time is managed as follows: when switching from standard time to daylight saving time, the device automatically adds one hour to the set time; when switching from daylight saving time to standard time, the device automatically subtracts one hour from the set time. The addition and subtraction operations normally take place when switching between one time and another and not every time that the status information is received via the object. Value “1” corresponds to daylight saving time, value “0” to standard time.



*“object value” / “operation to perform”*

If the time on the device is modified locally or via a BUS telegram, the device does not make any change to the new set time.

### ➤ 3.1.13 Control logic Heating/Air cooling

The device 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** to view the **“Heating/air cooling control algorithm”** and **“Heating/air cooling valve status feedback”** parameters, or select **different** to view the **“Heating control algorithm”** and **“Air cooling control algorithm”** parameters.

The “**Heating control algorithm**” parameter defines the control algorithm used for the heating system. The values that can be set are:

- **2 points ON-OFF** (default value)
- 2 points 0%-100%
- PWM proportional-integral
- continuous proportional-integral
- fancoil with ON-OFF speed control

Select **2 points ON-OFF** to view the “**Regulation differential (tenth of °C)**” parameter in the **Heating** menu, and the **Heating valve switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams.

Select **2 points 0%-100%** to view the “**Regulation differential (tenth of °C)**” parameter in the **Heating** menu, and the **Heating valve % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

Select **PWM proportional integral** to view the “**Select heating system**”, “**Proportional band**”, “**Integration time**” and “**Cycle time**” parameters in the **Heating** menu, and the **Heating valve switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams.

Select **continuous proportional-integral** to view the “**Select heating system**”, “**Proportional band**”, “**Integration time**” and “**Min. % variation for continuous command sending**” parameters in the **Heating** menu, and the **Heating valve % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

Selecting **fancoil with ON-OFF speed control** , displays the parameters “**Fancoil valve management**”, “**Valve regulation differential (tenth of °C)**”, “**Number of fancoil speeds**” and “**Fancoil speed status feedback**” in the **Heating** menu.

The “**Air cooling control algorithm**” parameter defines the control algorithm used for the air cooling system. The values that can be set are:

- **2 points ON-OFF** (default value)
- 2 points 0%-100%
- PWM proportional-integral
- continuous proportional-integral
- fancoil with ON-OFF speed control

Select **2 points ON-OFF** to view the “**Regulation differential (tenth of °C)**” parameter in the **Air cooling** menu, and the **Air cooling valve switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams.

Select **2 points 0%-100%** to view the “**Regulation differential (tenth of °C)**” parameter in the **Air cooling** menu, and the **Air cooling valve % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

Select **PWM proportional integral** to view the “**Select air cooling system**”, “**Proportional band**”, “**Integration time**” and “**Cycle time**” parameters in the **Air cooling** menu, and the **Air cooling valve switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams.

Select **continuous proportional-integral** to view the “**Select air cooling system**”, “**Proportional band**”, “**Integration time**” and “**Min. % variation for continuous command sending**” parameters in the **Air cooling** menu, and the **Air cooling valve % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

Selecting **fancoil with ON-OFF speed control**, displays the parameters “**Fancoil valve management**”, “**Valve regulation differential (tenth of °C)**”, “**Number of fancoil speeds**” and “**Fancoil speed status feedback**” in the **Air cooling** menu.

The “**Heating/Air cooling control algorithm**” parameter defines 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)
- 2 points 0%-100%
- PWM proportional-integral
- continuous proportional-integral
- fancoil with ON-OFF speed control

Select **2 points ON-OFF** in the **Heating** and **Air cooling** menus to display the “**Regulation differential (tenth of °C)**” parameter and the **Heating/air cooling valve switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams.

Select **2 points 0%-100%** in the **Heating** and **Air cooling** menus to display the “**Regulation differential (tenth of °C)**” parameter and the **Heating/air cooling valve % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

Select **PWM proportional** in the **Heating** and **Air cooling** menus to display the “**Select heating system**” (“**Select air cooling system**” in the **Air cooling** menu), “**Proportional band**”, “**Integration time**” and “**Cycle time**” parameters, and the **Heating/air cooling valve switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams.

Select **continuous proportional-integral** to display the “**Select heating system**” (“**Select air cooling system**” in the **Air cooling** menu), “**Proportional band**”, “**Integration time**” and “**Min. % variation for continuous command sending**” parameters in the **Heating** and **Air cooling** menus, and the **Heating/air cooling valve % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

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 “**Fancoil valve management**” parameter is used to define the solenoid valve control logic when the selected algorithm is fancoil. The values that can be set are:

- **2 points ON-OFF** (default value)
- 2 points 0%-100%

Select **2 points ON-OFF** to display the **Heating/air cooling valve switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams to the solenoid valve. Select **2 points 0%-100%** to display the **Heating/air cooling valve % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams to the solenoid valve.

The “**Heating/Air cooling valve status feedback**” parameter 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 “**Command repetition period with disabled feedback**” parameter. Select **enable** to view the **Heating/Air cooling valve status feedback** communication object (Data Point Type: 1.001 DPT\_Switch) if the valve control algorithm is **2 points ON-OFF** or **PWM proportional-integral**, or **Heating/air cooling valve % feedback** (Data Point Type: 5.001 DPT\_Scaling) if the valve control algorithm is **2 points 0%-100%** or **continuous proportional-integral**. When the BUS voltage is restored, the device sends the read request command via the **Heating/air cooling valve status feedback** or **Heating/Air**

**cooling valve % feedback** object, in order 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 “**Command repetition period with disabled feedback**” parameter 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)

If the control algorithm selected for Heating, Air cooling or Heating/Air cooling is **fancoil with ON-OFF speed control** or **fancoil with continuous speed control**, it is possible to modify the fancoil speed directly via BUS commands. The “**Modify fancoil speed via BUS**” parameter is used to enable the modification of the fancoil speed via BUS. The values that can be set are:

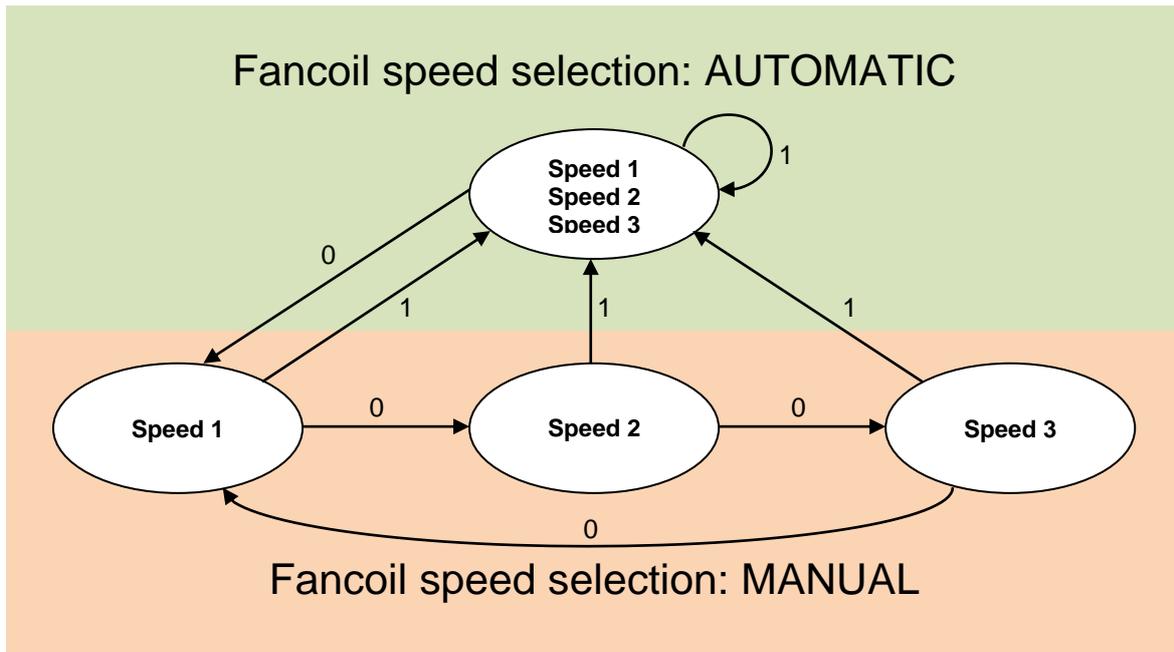
- **disable** (default value)
- enable

Selecting **enable** displays the **Fancoil mode input** communication object (Data Point Type: 1.001 DPT\_Switch) which makes it possible to receive fancoil mode selection commands. When the value “1” is received, the speed is defined autonomously by the device (AUTO fancoil mode) according to the different hystereses defined in the "Control algorithms" paragraph. When the value “0” is received, the fancoil mode changes to MANUAL speed 1; once the mode is MANUAL, each time the “0” value is received, the next speed is selected.

When fancoil mode is MANUAL, to activate the selected speed it is necessary to satisfy the hysteresis of the first regulation differential independently of the selected speed.

Regardless of the status, the value “1” switches the mode to AUTO.

The following block diagram summarises the concept (on the arrows, the value of the KNX telegram received on **Fancoil mode input**):



When the BUS voltage is restored, the fancoil mode is the one that was active before the voltage failure. If a change is made to the active functioning type, if the new functioning type is still fancoil, the fancoil speed (automatic or manual V1/V2/V3) will remain what was set previously, otherwise the AUTOMATIC mode is set again.

If the control algorithm selected for Heating, Air cooling or Heating/Air cooling is **fancoil with ON-OFF speed control** or **fancoil with continuous speed control**, it is possible to indicate the fancoil speed control mode. The “**Fancoil mode feedback (automatic/manual)**” parameter is used to enable the **Fancoil mode feedback** communication object via which the device indicates - via a BUS telegram - the fancoil speed control mode (manual/automatic). The values that can be set are:

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

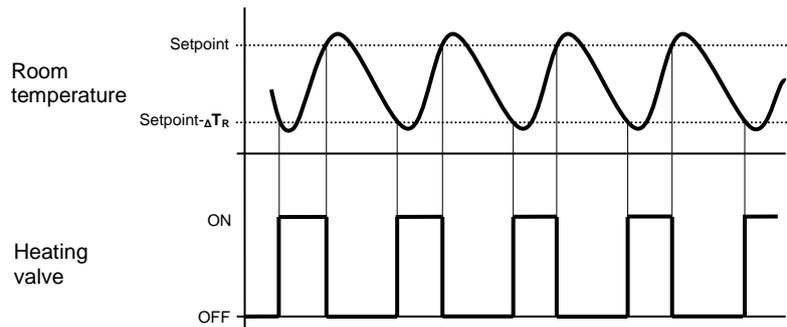
Selecting **enable** displays the **Fancoil mode feedback** communication object used to transmit the feedback. When the fancoil speed control mode changes from MANUAL to AUTOMATIC, the device sends a telegram via the BUS with the logical value “1”; when the fancoil speed control mode changes from AUTOMATIC to MANUAL, the device sends a “0” on the BUS.

### 3.2 Control algorithms

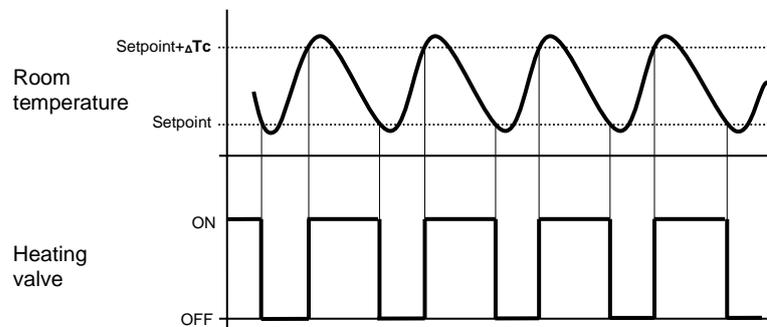
Independently of the fact if the control logic is common or different between the two operating types, the logic is as follows depending on the selected algorithm:

- **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.



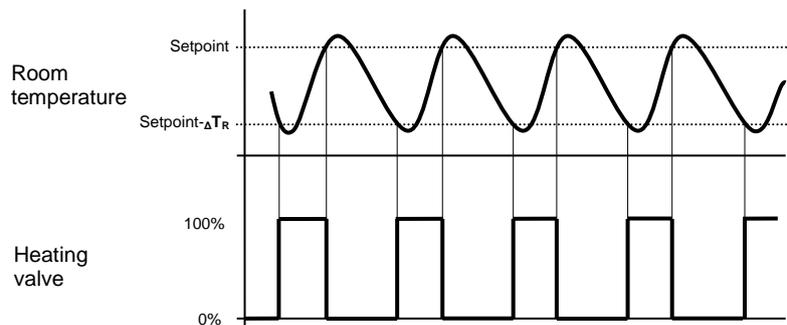
When the measured temperature is lower than the value “setpoint- $\Delta T_R$ ” (where  $\Delta T_R$  identifies the value of the heating regulation differential) the device turns on the heating system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches 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-  $\Delta T_R$ ” 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+  $\Delta T_c$ ” (where  $\Delta T_c$  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+ $\Delta T_c$ ” 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.

• **2 points 0%-100%**

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.

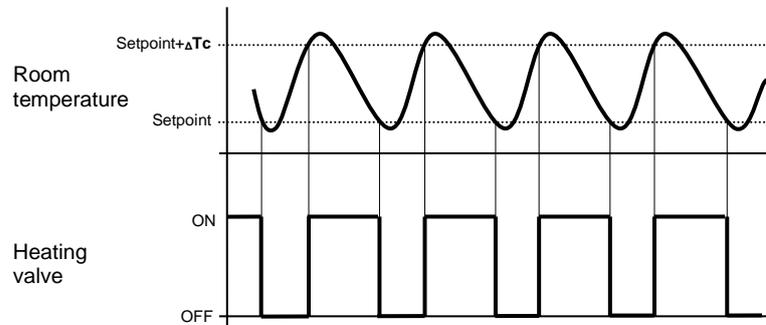


When the measured temperature is lower than the value setpoint- $\Delta T_R$ ” (where  $\Delta T_R$  identifies the value of the heating regulation differential) the device turns on the heating system, sending the relative BUS percentage

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 percentage 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-  $\Delta T_R$ ” 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.

To avoid the continuous switching of the solenoid valves, after a 0%-100%-0% sequence, the next 100% command can only be sent after at least 2 minutes have elapsed.



When the measured temperature is higher than the value “setpoint+  $\Delta T_c$ ” (where  $\Delta T_c$  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+ $\Delta T_c$ ” above which the device turns on the system.

- **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**

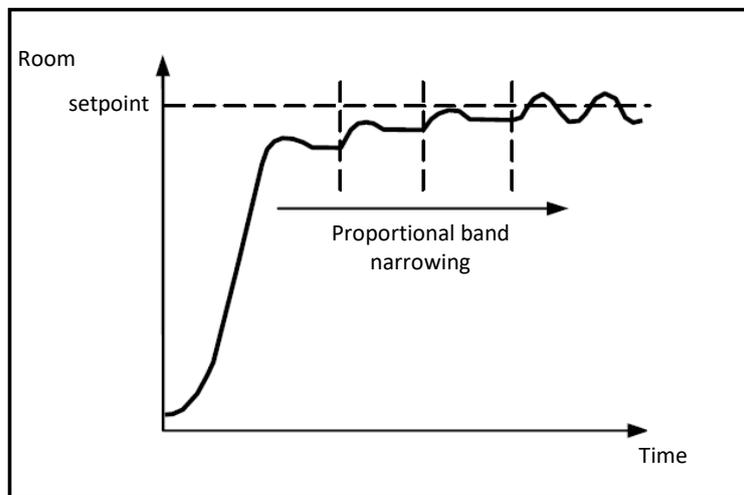
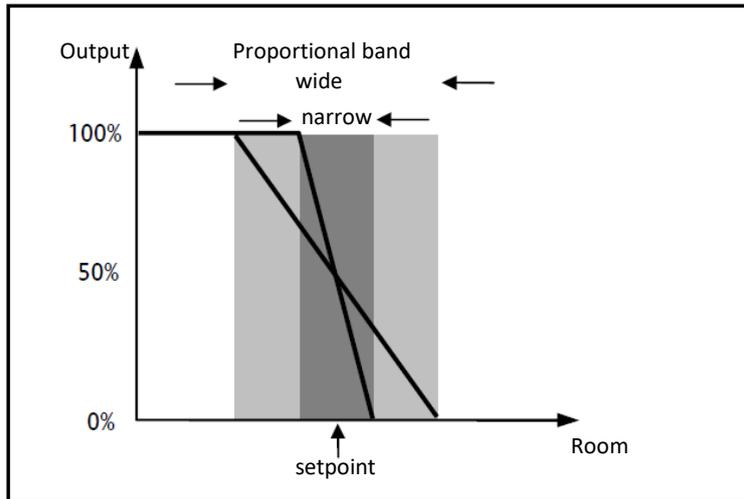
In the proportional component, the output function is proportional to the error (difference between setpoint and measured temperature).

$$P_{out} = K_p e(t)$$

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.



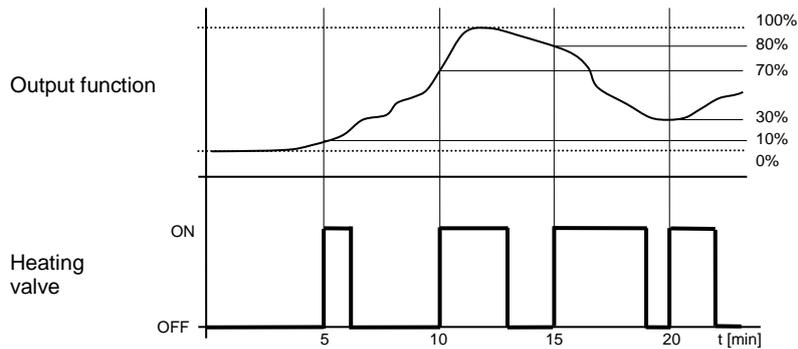
### Integral component

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.

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

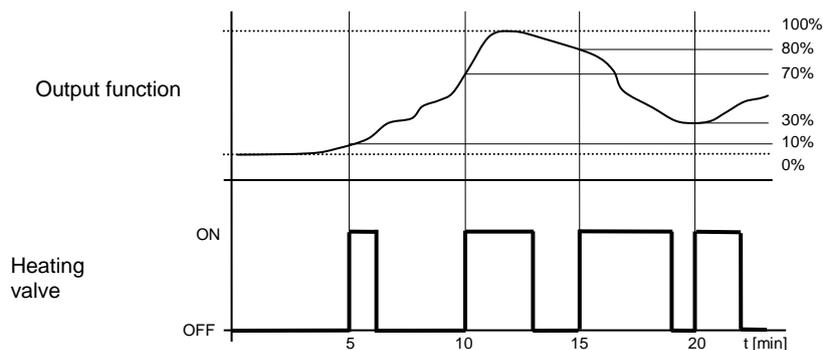
The integral period accelerates the dynamics of the process towards the setpoint and eliminates the residuals of the stationary error status that takes place with a pure proportional controller.

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.

- **continuous 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 continuous control. This type of control involves the continuous control of the difference between the measured temperature and the fixed set-point and as a result sends the commands for the modulation of the power of the temperature adjustment system. 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**

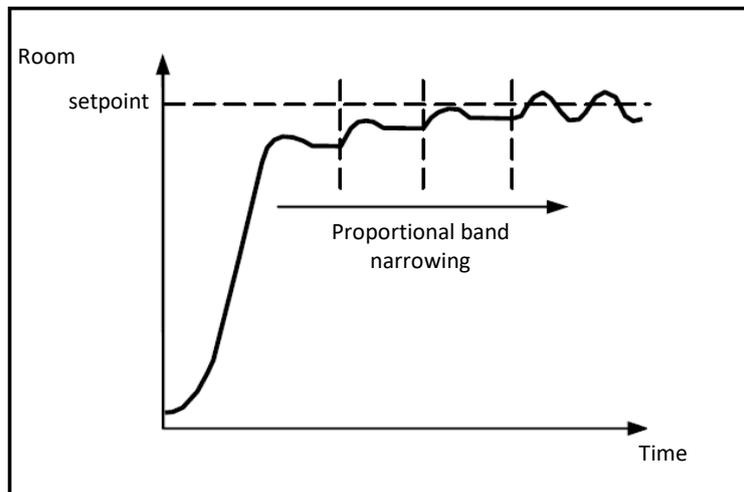
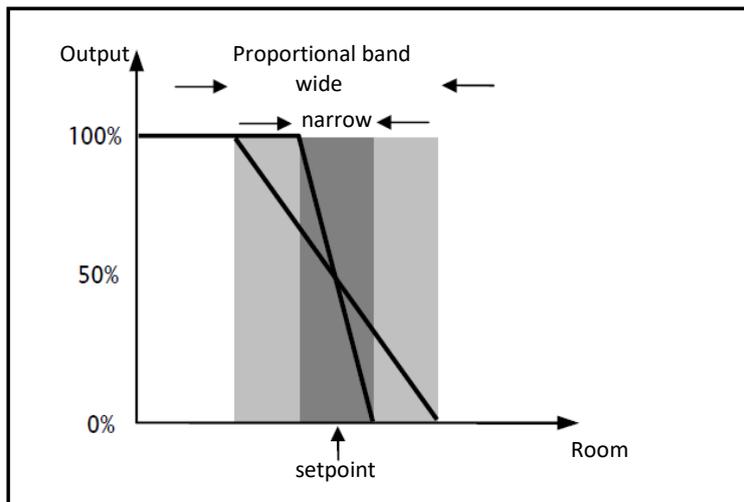
In the proportional component, the output function is proportional to the error (difference between setpoint and measured temperature).

$$P_{out} = K_p e(t)$$

Once the proportional band is defined, the output within the band varies between 0% and 100%; outside the band, the output will be the maximum power or the minimum power depending on the reference limit.

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**

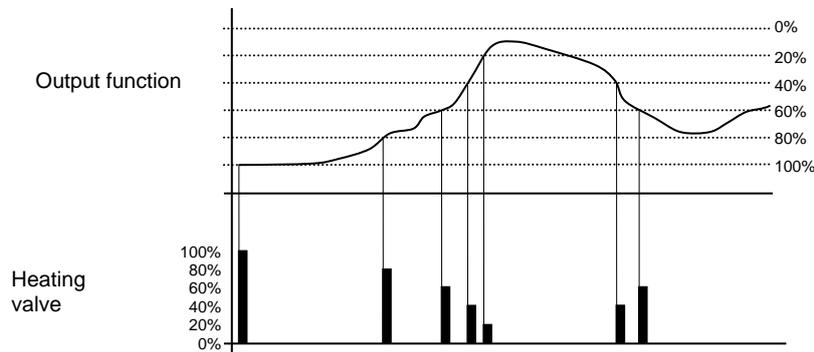
The contribute of the integral period is proportional to the error (difference between the setpoint and the measured temperature) and its duration. The integral is the sum of the instantaneous error for every moment of time and provides the accumulated offset that should have been previously corrected.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

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

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.

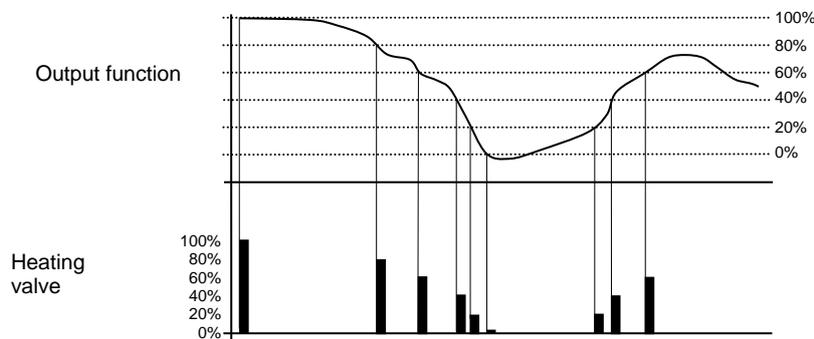
The integration time is the parameter that determines the action of the integral component. The longer the integration time, the slower the modification of the output and hence the slower the system response. If the

time is too short, the threshold value will be exceeded (overshoot), and the function will swing around the set-point.



As can be seen in the figure, the device sends the commands to the actuator that manages the heating system based on the output function of the proportional-integral control, along the vertical axis, the 0% - 100% interval of the output function of the proportional-integral control is divided into different levels with a distance equal to the value defined by the parameter “**Min. % variation for continuous command sending**” (in the figure, the value is **20%**) and the device continuously adjusts the heating system by sending percentage activation values of the solenoid valve (shown along the vertical axis) that depend on the intersection of the output function value calculated with a determined level. In this way, the KNX BUS will not be saturated with continuous telegrams.

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 can be seen in the figure, the device sends the commands to the actuator that manages the air cooling system based on the output function of the proportional-integral control; along the vertical axis, the 0% - 100% interval of the output function of the proportional-integral control is divided into different levels with a distance equal to the value defined by the parameter “**Min. % variation for continuous command sending**” (in the figure, the value is **20%**) and the device continuously adjusts the air cooling system by sending percentage activation values of the solenoid valve (shown along the vertical axis) that depend on the intersection of the output function value calculated with a determined level. In this way, the KNX BUS will not be saturated with continuous telegrams.

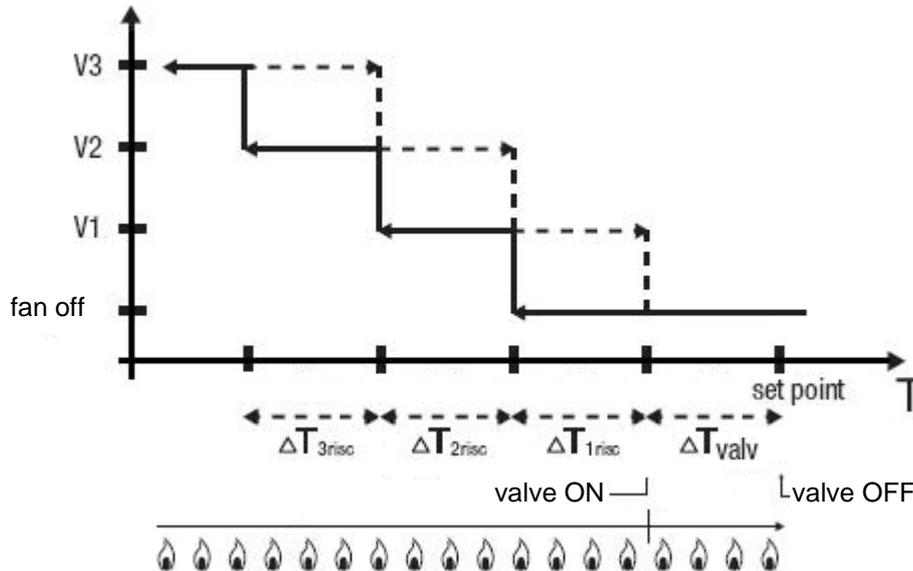
With this type of algorithm, there is no longer a hysteresis cycle on the cooling device, so the inertia times (system air cooling and heating times) introduced by the 2-point control are eliminated. This produces energy savings because the system does not remain switched on when it is not needed and, 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.

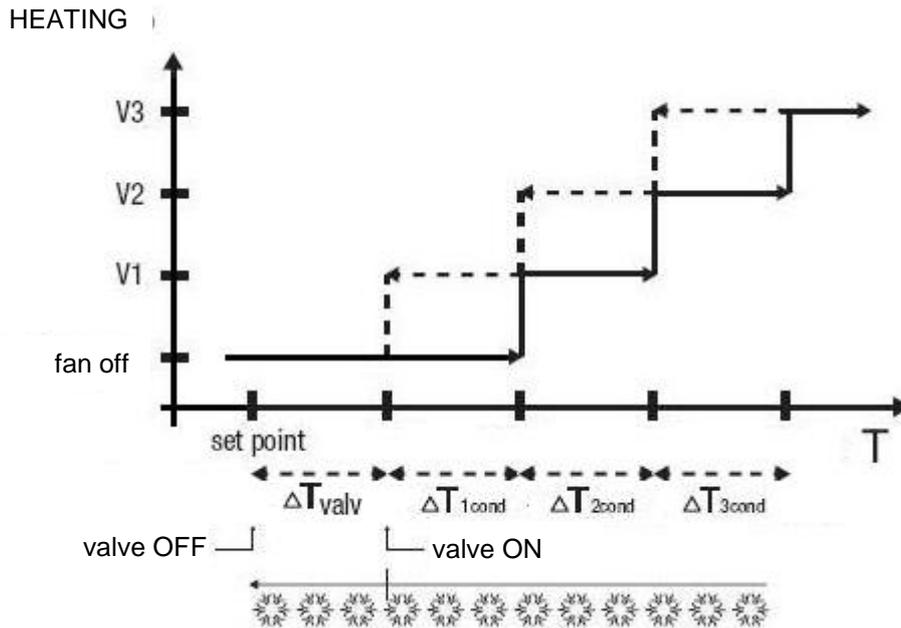
HEATING



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- $\Delta T_{valv}$ - $\Delta T_{1\ heat}$ ” and turned off when the temperature value reaches the value “setpoint- $\Delta T_{valv}$ ” (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- $\Delta T_{valv}$ - $\Delta T_{1\ heat}$ - $\Delta T_{2\ heat}$ ” and turned off when the temperature value reaches the value “setpoint- $\Delta T_{valv}$ - $\Delta T_{1\ heat}$ ”. 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- $\Delta T_{valv}$ - $\Delta T_{1\ heat}$ - $\Delta T_{2\ heat}$ - $\Delta T_{3\ heat}$ ” and turned off when the temperature value reaches the value “setpoint- $\Delta T_{valv}$ - $\Delta T_{1\ heat}$ - $\Delta T_{2\ heat}$ ”

With regards the heating solenoid valve, once the measured temperature is lower than the value “setpoint- $\Delta T_{valv}$ ”, the thermostat 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 set-point value. In this way, the heating of the fan coil 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+ $\Delta T_{valv}+\Delta T_{1cool}$ ” and turned off when the temperature value reaches the value “setpoint+ $\Delta T_{valv}$ ” (or the “setpoint” value if  $\Delta T_{1cool}=0$ ). 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+ $\Delta T_{valv}+\Delta T_{1cool}+\Delta T_{2cool}$ ” and turned off when the temperature value reaches the value “setpoint+ $\Delta T_{valv}+\Delta T_{1cool}$ ”. 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+ $\Delta T_{valv}+\Delta T_{1cool}+\Delta T_{2cool}+\Delta T_{3cool}$ ” and turned off when the temperature value reaches the value “setpoint+ $\Delta T_{valv}+\Delta T_{1cool}+\Delta T_{2cool}$ ”

With regards the air cooling solenoid valve, once the measured temperature is higher than the value “setpoint+ $\Delta T_{valv}$ ”, the thermostat 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 set-point value. In this way, the air cooling of the fan coil can also be exploited for irradiation, without any speed being activated.

To avoid continuous switchovers, the thermostat 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 fancoil speeds.

Both figures refer to the three-stage control of the fancoil, as the descriptions are complete. For two-stage or single-stage control, the logic is the same, but not all the speeds are controlled.

## 4 “Heating” menu

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

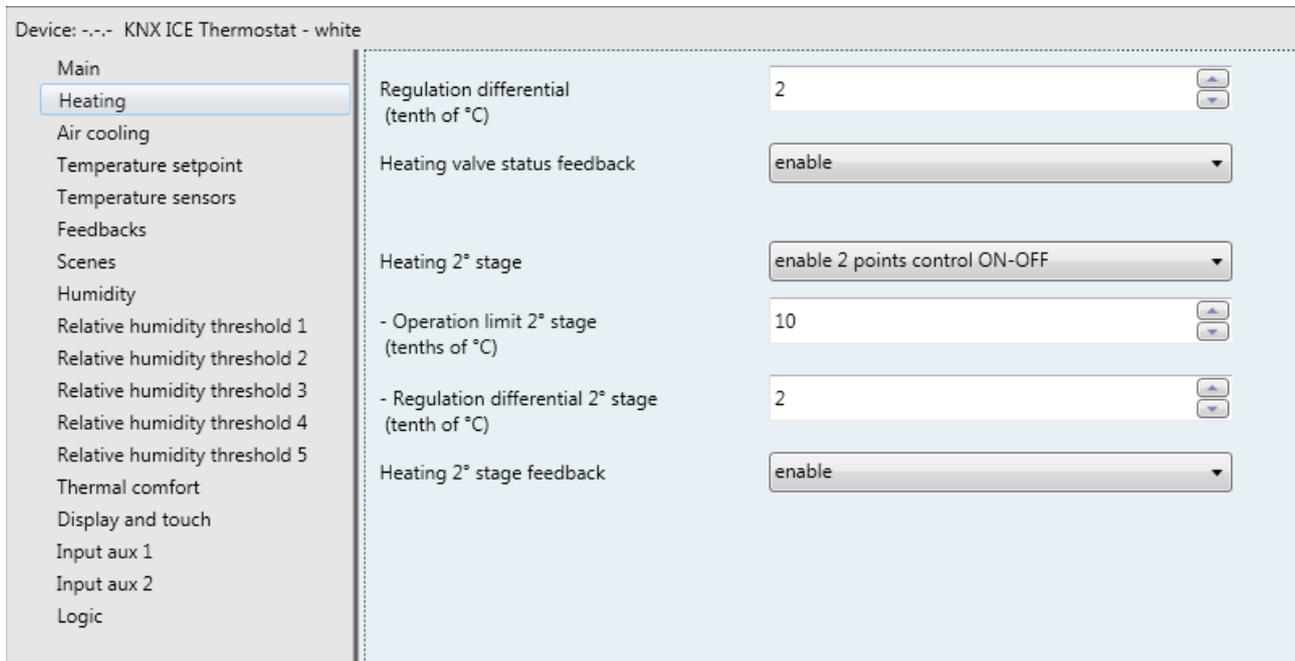


Fig. 4.1

### 4.1 Parameters

#### ➤ 4.1.1 Regulation differential (tenth of °C)

This is used to set the regulation differential value of the **2 points ON-OFF** or **2 points 0%-100%** control algorithm of the heating system (already mentioned in the "Control algorithms" section) which, when subtracted from the established setpoint value, determines the threshold value below which the heating system is turned on in the 2-point control. The values that can be set are:

- from 1 to 20 with steps of 1, (**default value 2**)

#### ➤ 4.1.2 Select heating system

The “**Select heating system**” parameter is used to automatically size the operating parameters (Proportional band and Integration time) of the proportional integral algorithm PWM or Continuous, 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 “**Proportional band**” and “**Integration time (minutes)**” parameters will be displayed but cannot be modified, and the values **5.0°C** and **150** are displayed.

Selecting **floor heating**, the “**Proportional band**” and “**Integration time (minutes)**” parameters will be displayed but cannot be modified, and the values **5.0°C** and **240** are displayed.

Selecting **fancoil unit**, the “**Proportional band**” and “**Integration time (minutes)**” parameters will be displayed but cannot be modified, and the values **4.0°C** and **90** are displayed.

Selecting **electric heating**, the “**Proportional band**” and “**Integration time (minutes)**” parameters 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.

It is not necessary to store the “**Select heating system**” parameter in the memory.

The “**Proportional band**” parameter is used to set the width of the PWM proportional band of the **PWM proportional integral** or **continuous proportional integral** control algorithm of the heating system (already mentioned in the Control algorithms section) which, when subtracted from the setpoint value, 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 “**Integration time**” parameter is used to set the contribution of the integral action in the proportional integral control (see the Control algorithms section). 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

The “**Min. % variation for continuous command sending**” parameter is used to set the minimum variation of the percentage command value (in comparison to the last sent command) to generate the sending of the command itself. The values that can be set are:

- 1%
- 2%

- 3%
- 4%
- **5% (default value)**
- 10%
- 20%

Intrinsically, this value also determines the number of proportional sub-bands within which the device determines the value of the power to send to the system (see the Control algorithms section); there is no fixed number of proportional sub-bands, as it depends on the value set for this item.

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 “**Fancoil valve management**” parameter is used to define the solenoid valve control logic when the selected algorithm is fancoil. The values that can be set are:

- **2 points ON-OFF (default value)**
- 2 points 0%-100%

Selecting **2 points ON-OFF** displays the *Heating valve switching* communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams. Selecting **2 points 0%-100%** displays the *Heating valve % command* communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

The “**Valve regulation differential (tenth of °C)**” parameter is used to set the regulation differential value of the 2-point control of the fancoil operating solenoid valve, as mentioned in the Control algorithms section. The values that can be set are:

- from 1 to 20 with steps of 1, **(default value 2)**

The “**Number of fancoil speeds**” parameter is used to set the number of stages for controlling the fancoil speed, based on the type of fancoil used. The values that can be set are:

- 1

Setting this value, the number of stages for controlling the fancoil speeds is 1; this setting displays the parameters “**Speed 1 regulation differential (tenth of °C)**” and “**Speed 1 inertia time (seconds)**” and the communication object *Heating fan V1 switching* (Data Point Type: 1.001 DPT\_Switch) for controlling the first and only fancoil speed.

- 2

Setting this value, the number of stages for controlling the fancoil speeds is 2. This setting displays the parameters “**Speed 1 regulation differential (tenth of °C)**”, “**Speed 2 regulation differential (tenth of °C)**”, “**Speed 1 inertia time (seconds)**” and “**Speed 2 inertia time (seconds)**” and the communication objects *Heating fan V1 switching* and *Heating fan V2 switching* (Data Point Type: 1.001 DPT\_Switch) for controlling the first and second fancoil speeds respectively.

- **3 (default value)**

Setting this value, the number of stages for controlling the fancoil speeds is 3. This setting 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)**” and “**Speed 3 inertia time (seconds)**” and the communication objects *Heating fan V1 switching*, *Heating fan V2 switching* and *Heating fan V3 switching* (Data Point Type: 1.001 DPT\_Switch) for controlling the first, second and third fancoil speeds respectively.

The “**Speed 1 regulation differential (tenth of °C)**” parameter 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** or **fancoil with continuous speed control**, already mentioned in the Control algorithms section. This value, subtracted from the “setpoint- $\Delta T_{valv}$ ” determines the value of the threshold 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 " $\Delta T_{1 \text{ heat}} = \Delta T_{\text{valv}}$ " for which the value of the speed 1 activation threshold is " $\text{setpoint} - \Delta T_{\text{valv}}$ " and the off value is "setpoint".

The "**Speed 2 regulation differential (tenth of °C)**" parameter 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** or **fancoil with continuous speed control**, already mentioned in the Control algorithms section. This value, subtracted from the value " $\text{setpoint} - \Delta T_{\text{valv}} - \Delta T_{1 \text{ heat}}$ " determines the value of the threshold, 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, **(default value 2)**

The "**Speed 3 regulation differential (tenth of °C)**" parameter 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** or **fancoil with continuous speed control**, already mentioned in the Control algorithms section. This value, subtracted from the value " $\text{setpoint} - \Delta T_{\text{valv}} - \Delta T_{1 \text{ heat}} - \Delta T_{2 \text{ heat}}$ " determines the value of the threshold, 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, **(default value 2)**

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).

The "**Heating valve status feedback**" parameter 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 "**Command repetition period with disabled feedback**" parameter. Select **enable** to view the *Heating valve status feedback* communication object (Data Point Type: 1.001

DPT\_Switch) if the valve control algorithm is **2 points ON-OFF** or **PWM proportional-integral**, or **Heating valve % feedback** (Data Point Type: 5.001 DPT\_Scaling) if the valve control algorithm is **2 points 0%-100%** or **continuous proportional-integral**. When the BUS voltage is restored, the device sends the read request command via the **Heating valve status feedback** or **Heating valve % feedback** object, in order 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, 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. 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. In the particular case in which the control algorithm is continuous proportional-integral, the feedback received can be different than the sent command by approx.  $\pm 1.8\%$  (3 units out of 255). In this way, if for reasons of approximation, the actuator that controls the valve sends a value that differs slightly from the value requested by the thermostat, the operation is guaranteed and the process of periodically sending the command is not triggered.

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)

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, 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 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 “**Fancoil speed command repetition period**” parameter. Select **enable** to view the following communication objects

**Heating fan V1 status feedback, Heating fan V2 status feedback and Heating fan V3 status feedback** (Data Point Type: 1.001 DPT\_Switch), according to the number of fancoil speeds.

When the BUS voltage is restored, the device sends the read request command via the **Heating fan V1 status feedback, Heating fan V2 status feedback, Heating fan V3 status feedback** or **Heating valve % feedback** objects in order to be updated about the activation status of the fancoil speeds.

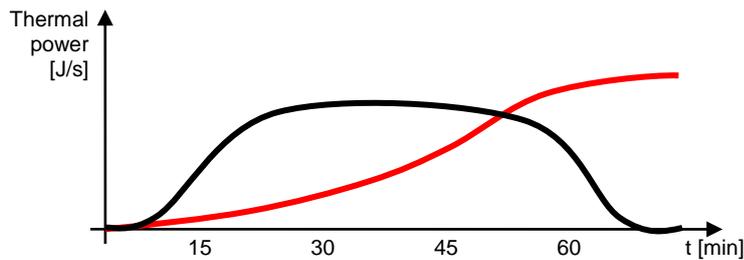
If fancoil feedback is disabled, deactivation commands of the inactive speeds must be sent for every speed activation command; in the same manner, every speed deactivation command must be sent together with deactivation commands for the other speeds.

The “**Fancoil speed command repetition period**” parameter is used to define the time range of the cycling sending to the fancoil speeds. The values that can be set are:

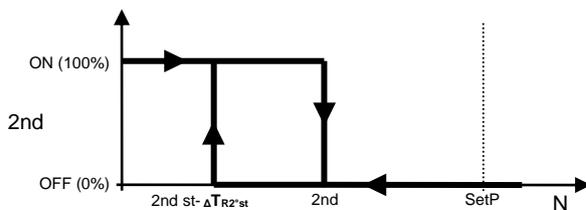
- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** (default value)

In the particular case that the control is **fancoil with ON-OFF speed control**, then the commands are repeated on all speed communication objects.

Some heating systems (for example, underfloor heating) have very high thermal inertia levels and require a considerable amount of time to bring the room temperature into line with the required setpoint. In order to reduce this inertia, another heating system with less inertia is often installed to help the main system to heat the room when the difference between the setpoint and the measured temperature is particularly large. This system, known as 2nd stage, helps to heat the room during the initial phase, then it stops working when the difference between the setpoint and the temperature can be managed faster.



The control algorithm for the second stage can usually be a 2-point control, either ON-OFF or 0%-100%, and the intervention thresholds of the second stage are as follows:



When the measured temperature is lower than the value “2nd st-  $\Delta T_{R2nd\ st}$ ” (where  $\Delta T_{R2nd\ st}$  identifies the value of the regulation differential of 2nd stage heating) the device activates 2nd stage heating by sending the relative BUS command to the actuator that manages it; when the measured temperature reaches the value “2nd st” (defined by the 2nd stage intervention limit-setpoint), the device deactivates 2nd stage heating by sending the relative BUS command to the actuator that manages it.

This makes it clear that there are two decision thresholds for activating and deactivating 2nd stage heating. The first consists of the value “2nd st-  $\Delta T_{R2nd\ st}$ ” below which the device turns on the system, and the second consists of the value “2nd st” above which the device switches off the system.

### ➤ 4.1.3 2nd stage heating

This is used to enable and define the control algorithm for the second heating stage. The values that can be set are:

- **disabled** (default value)
- enable 2-point control ON-OFF

- enable 2-point control 0%-100%

Selecting any value other than **disabled** displays the “**Operating limit 2nd stage (tenths of °C)**”, “**Regulation differential 2nd stage (tenth of °C)**” and “**Heating 2nd stage feedback**” parameters. Selecting **enable 2-point control ON-OFF** displays the *Heating 2nd stage switching* communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams. Selecting **enable 2-point control 0%-100%** displays the *Heating 2nd stage % command* communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

The “**Operation limit 2nd stage (tenths of °C)**” parameter is used to define the operation threshold of 2nd stage heating. The value set for this parameter, when subtracted from the setpoint currently in use, determines the upper 2nd stage operation limit (**2nd St** in the chart above) beyond which the latter is deactivated. The values that can be set are:

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

The “**Regulation differential 2° stage (tenths of °C)**” parameter is used to set the value of the regulation differential of the 2nd stage heating control algorithm that, when subtracted from the "operation limit-setpoint" determines the value of the threshold (**2nd st- $\Delta T_{R2nd\ st}$**  in the chart above) below which the 2nd stage heating system in the 2-point control is activated. The values that can be set are:

- from 1 to 20 with steps of 1, (default value 2)

As for the basic heating algorithm, the “**Heating 2nd stage feedback**” parameter is used to enable the device to receive feedback from the actuator that controls 2nd stage heating. In this way, the device is able to receive the telegram after the actuator 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 “**2nd stage command repetition period without feedback**” parameter. Select **enable** to view the *Heating 2nd stage feedback* communication object (Data Point Type: 1.001 DPT\_Switch) if the 2nd stage control algorithm is **2 points ON-OFF**, or *Heating 2nd stage valve % feedback* (Data Point Type: 5.001 DPT\_Scaling) if the control algorithm is **2 points 0%-100%**. When the BUS voltage is restored, the device sends the read request command via the *Heating 2nd stage feedback* or *Heating 2nd stage valve % feedback* object, in order to be updated about the status of the second stage heating.

With feedback enabled, after the device sends the switching command, 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 every minute until it receives the feedback of correct switching. It can happen that, during normal operation of the temperature adjustment, the status of the actuator that manages the 2nd stage can be changed by an entity external of the thermostat, that forces its status, modifying it. In this case, the device repeats the switching command to realign the status of the actuator with the one determined by the control logic, triggering the process for waiting for confirmation and repeating the command until the confirmation is received.

With the 2nd stage heating status feedback disabled, it may be useful to cyclically repeat the command to the actuator so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The “**2nd stage command repetition period without 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)**

## 5 "Air cooling" menu

The **Air cooling** menu contains the characteristic parameters of the load control algorithms for the air cooling system.

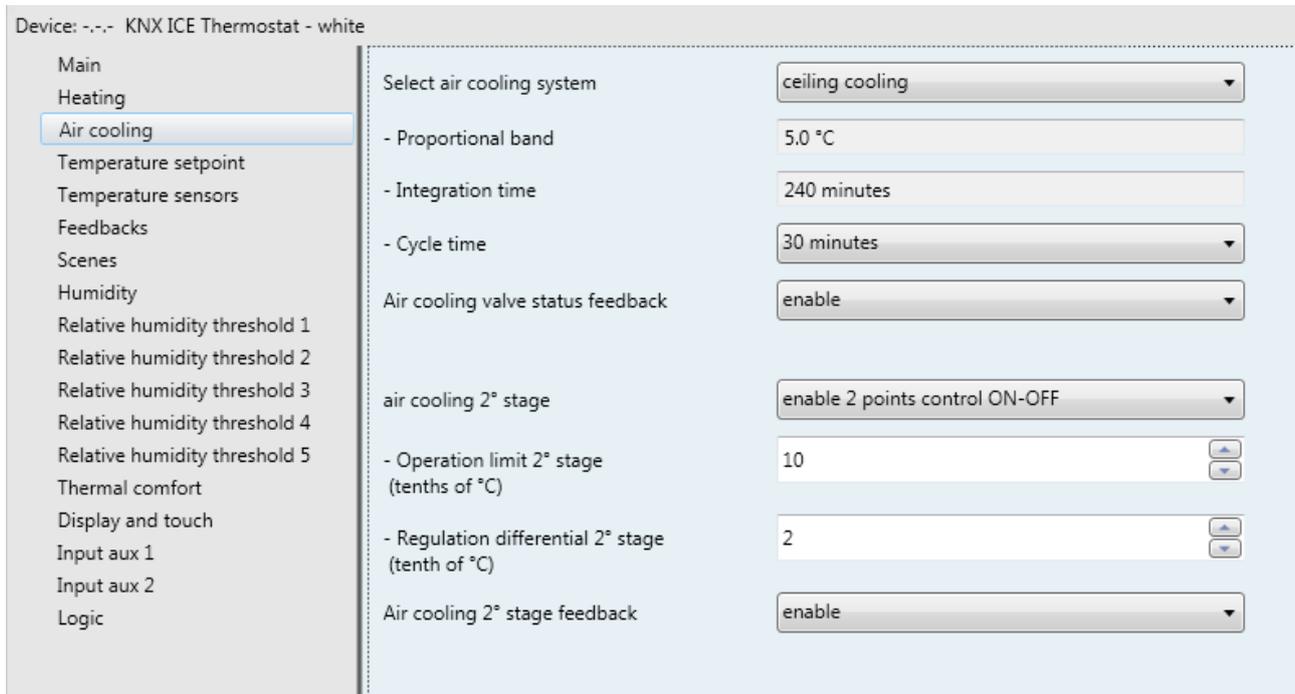


Fig. 5.1

### 5.1 Parameters

#### ➤ 5.1.1 Regulation differential (tenth of °C)

This is used to set the regulation differential value of the **2 points ON-OFF** or **2 points 0%-100%** control algorithm of the heating system (already mentioned in the "Control algorithms" section) which, when subtracted from the established setpoint value, determines the threshold value below which the heating system is turned on in the 2-point control. The values that can be set are:

- from 1 to 20 with steps of 1, (**default value 2**)

#### ➤ 5.1.2 Select air cooling system

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 "**Proportional band**" and "**Integration time (minutes)**" parameters will be displayed but cannot be modified, and the values **5.0°C** and **240** are displayed.

Selecting **fancoil unit**, the "**Proportional band**" and "**Integration time (minutes)**" parameters will be displayed but cannot be modified, and the values **4.0°C** and **90** are displayed.

Selecting **customised**, the "**Proportional band**" and "**Integration time (minutes)**" parameters will be displayed but cannot be modified.

It is not necessary to store the “**Select air cooling system**” parameter in the memory.

The “**Proportional band**” parameter is used to set the width of the PWM proportional band of the **PWM proportional integral** or **continuous proportional integral** control algorithm of the cooling system (already mentioned in the Control algorithms section) which, when added to the setpoint value, determines the upper limit of the proportional band used for 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 “**Integration time (minutes)**” parameter is used to set the contribution of the integral action in the proportional integral control (see the Control algorithms section). 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

The “**Min. % variation for continuous command sending**” parameter is used to set the minimum variation of the percentage command value (in comparison to the last sent command) to generate the sending of the command itself. The values that can be set are:

- 1%
- 2%
- 3%
- 4%
- **5%** (default value)
- 10%
- 20%

Intrinsically, this value also determines the number of proportional sub-bands within which the device determines the value of the power to send to the system (see the Control algorithms section); there is no fixed number of proportional sub-bands, as it depends on the value set for this item.

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 “**Fancoil valve management**” parameter is used to define the solenoid valve control logic when the selected algorithm is fancoil. The values that can be set are:

- **2 points ON-OFF** (default value)
- 2 points 0%-100%

Selecting **2 points ON-OFF** displays the *Air cooling valve switching* communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams. Selecting **2 points 0%-100%** displays the *Air cooling valve % command* communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

The “**Valve regulation differential (tenth of °C)**” parameter is used to set the regulation differential value of the 2-point control of the fancoil operating solenoid valve, as mentioned in the Control algorithms section. The values that can be set are:

- from 1 to 20 with steps of 1, (default value 2)

The “**Number of fancoil speeds**” parameter is used to set the number of stages for controlling the fancoil speed, based on the type of fancoil used. The values that can be set are:

- 1

Setting this value, the number of stages for controlling the fancoil speeds is 1; this setting displays the “**Speed 1 regulation differential (tenth of °C)**” and “**Speed 1 inertia time (seconds)**” parameters.

In this case, based on the value set for “**Air cooling control algorithm**” in the **Main** menu, this enables the communication object *Air cooling fan V1 switching* (Data Point Type: 1.001 DPT\_Switch) for controlling the first and only fancoil speed.

- 2

Setting this value, the number of stages for controlling the fancoil speeds is 2. This setting displays the “**Speed 1 regulation differential (tenth of °C)**”, “**Speed 2 regulation differential (tenth of °C)**”, “**Speed 1 inertia value (seconds)**” and “**Speed 2 inertia value (seconds)**” parameters.

In this case, based on the value set for “**Air cooling control algorithm**” in the **Main** menu, the following communication objects are enabled *Air cooling fan V1 switching* and *Air cooling fan V2 switching* (Data Point Type: 1.001 DPT\_Switch) for controlling the first and second fancoil speeds respectively.

- **3 (default value)**

Setting this value, the number of stages for controlling the fancoil speeds is 3. This setting displays the “**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)**” and “**Speed 3 inertia time (seconds)**” parameters.

In this case, based on the value set for “**Air cooling control algorithm**” in the **Main** menu, the following communication objects are enabled *Air cooling fan V1 switching*, *Air cooling fan V2 switching* and *Air cooling fan V3 switching* (Data Point Type: 1.001 DPT\_Switch) for controlling the first, second and third fancoil speeds respectively.

The “**Speed 1 regulation differential (tenth of °C)**” parameter 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**, already mentioned in the Control algorithms section. This value, added to the value “setpoint+ $\Delta T_{valv}$ ” determines the value of the threshold 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 “ $\Delta T_1 \text{ cool} = \Delta T_{valv}$ ” for which the value of the speed 1 activation threshold is “setpoint+ $\Delta T_{valv}$ ” and the off value is “setpoint”.

The “**Speed 2 regulation differential (tenth of °C)**” parameter 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**, already mentioned in the Control algorithms section. This value,

subtracted from the value “setpoint+ $\Delta T_{valv}$ + $\Delta T_{1\ cool}$ ” determines the value of the threshold 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, **(default value 2)**

The “**Speed 3 regulation differential (tenth of °C)**” parameter 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**, already mentioned in the Control algorithms section. This value, subtracted from the value “setpoint+ $\Delta T_{valv}$ + $\Delta T_{1\ cool}$ + $\Delta T_{2\ cool}$ ” determines the value of the threshold 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, **(default value 2)**

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

The “**Air cooling valve status feedback**” parameter 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 “**Command repetition period with disabled feedback**” parameter. Select **enable** to view the **Air cooling valve status feedback** communication object (Data Point Type: 1.001 DPT\_Switch) if the valve control algorithm is **2 points ON-OFF** or **PWM proportional-integral**, or **Air cooling valve % feedback** (Data Point Type: 5.001 DPT\_Scaling) if the valve control algorithm is **2 points 0%-100%** or **continuous proportional-integral**. When the BUS voltage is restored, the device sends the read request command via the **Air cooling valve status feedback** or **Air cooling valve % feedback** object, in order 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, 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. 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. In the particular case in which the control algorithm is continuous proportional-integral, the feedback received can be different than the sent command by approx.  $\pm 1.8\%$  (3 units out of 255). In this way, if for reasons of approximation, the actuator that controls the valve sends a value that differs slightly from the value requested by the thermostat, the operation is guaranteed and the process of periodically sending the command is not triggered.

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 “**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)

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, 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 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 display the parameter “**Fancoil speed command repetition period**” and the following communication objects: ***Air cooling fan V1 status feedback, Air cooling fan V2 status feedback and Air cooling fan V3 status feedback*** (Data Point Type: 1.001 DPT\_Switch), according to the number of fancoil speeds.

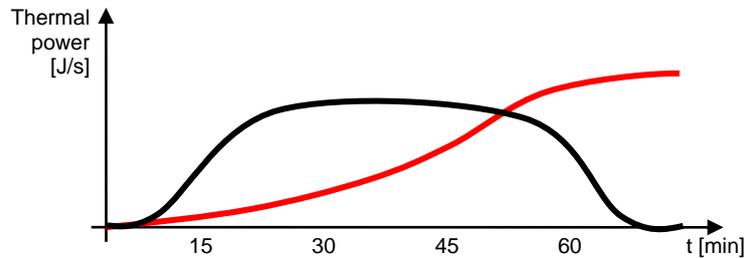
When the BUS voltage is restored, the device sends the read request command via the ***Air cooling fan V1 status feedback, Air cooling fan V2 status feedback, Air cooling fan V3 status feedback or Air cooling valve % feedback*** object in order to be updated about the activation status of the fancoil speeds.

The “**Fancoil speed command repetition period**” parameter is used to define the time range of the cycling sending to the fancoil speeds. The values that can be set are:

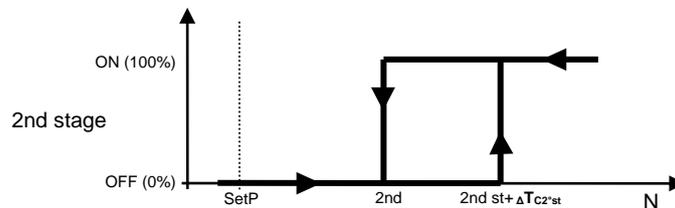
- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** (default value)

In the particular case that the control is **fancoil with ON-OFF speed control**, then the commands are repeated on all speed communication objects.

Some air-cooling systems have very high thermal inertia levels and require a considerable amount of time to bring the room temperature into line with the required setpoint. In order to reduce this inertia, another heating system with less inertia is often installed to help the main system to heat the room when the difference between the setpoint and the measured temperature is particularly large. This system, known as 2nd stage, helps to heat the room during the initial phase, then it stops working when the difference between the setpoint and the temperature can be managed faster.



The control algorithm for the second stage can usually be a 2-point control, either ON-OFF or 0%-100%, and the intervention thresholds of the second stage are as follows:



When the measured temperature is higher than the value “2nd st+  $\Delta T_{C2^{*st}}$ ” (where  $\Delta T_{C2^{*st}}$  identifies the value of the regulation differential of 2nd stage air-cooling), the device activates the 2nd stage air-cooling by sending the relative BUS command to the actuator that manages it. When the measured temperature reaches the value “2nd st” (defined by the 2nd stage intervention limit-setpoint), the device deactivates 2nd stage air-cooling by sending the relative BUS command to the actuator that manages it.

This makes it clear that there are two decision thresholds for activating and deactivating 2nd stage air-cooling. The first consists of the value “2nd st+  $\Delta T_{C2^{*st}}$ ” above which the device turns on the system, and the second consists of the value “2nd st” below which the device switches off the system.

### ➤ 5.1.3 2nd stage air cooling

This is used to enable and define the control algorithm for second stage air cooling. The values that can be set are:

- **disabled** (default value)
- enable 2-point control ON-OFF
- enable 2-point control 0%-100%

Selecting any value other than **disabled** displays the “**Operating limit 2nd stage, “Regulation differential 2nd stage (tenth of °C)”** and “**Air cooling 2nd stage feedback**” parameters. Selecting **enable 2-point control ON-OFF** displays the **2nd stage air cooling switching** communication object (Data Point Type: 1.001 DPT\_Switch) via which the device sends the command telegrams. Selecting **enable 2-point control 0%-100%** displays the **2nd stage air cooling % command** communication object (Data Point Type: 5.001 DPT\_Scaling) via which the device sends the command telegrams.

The “**Operation limit 2nd stage (tenths of °C)**” parameter is used to define the operation threshold of 2nd stage air cooling. The value set for this parameter, when subtracted from the setpoint currently in use, determines the lower operating limit of the 2nd stage below which the latter is deactivated. The values that can be set are:

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

The “**Regulation differential 2nd stage (tenths of °C)**” parameter is used to set the value of the regulation differential of the 2nd stage air cooling control algorithm that, when added to the "operation limit-setpoint" determines the value of the threshold (**2nd st+  $\Delta T_{C2nd\ st}$**  in the chart above) below which the 2nd stage air cooling system in the 2-point control is activated. The values that can be set are:

- from 1 to 20 with steps of 1, **(default value 2)**

As for the basic air cooling algorithm, the “**2nd stage air cooling feedback**” parameter is used to enable the device to receive feedback from the actuator that controls 2nd stage air cooling. In this way, the device is able to receive the telegram after the actuator 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 “**2nd stage command repetition period without feedback**” parameter. Select **enable** to view the **2nd stage air cooling feedback** communication object (Data Point Type: 1.001 DPT\_Switch) if the 2nd stage control algorithm is **2 points ON-OFF**, or **Air cooling 2° stage % feedback** (Data Point Type: 5.001 DPT\_Scaling) if the control algorithm is **2 points 0%-100%**. When the BUS voltage is restored, the device sends the read request command via the **2nd stage air cooling feedback** or **2nd stage air cooling % feedback** object in order to be updated about the status of the second stage air cooling.

With feedback enabled, after the device sends the switching command, 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 every minute until it receives the feedback of correct switching. It can happen that, during normal operation of the temperature adjustment, the status of the actuator that manages the 2nd stage can be changed by an entity external of the thermostat, that forces its status, modifying it. In this case, the device repeats the switching command to realign the status of the actuator with the one determined by the control logic, triggering the process for waiting for confirmation and repeating the command until the confirmation is received.

With 2nd stage air cooling status feedback disabled, it may be useful to cyclically repeat the command to the actuator so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The “**2nd stage command repetition period without 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)**

## 6 "Temperature setpoint" menu

The **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:

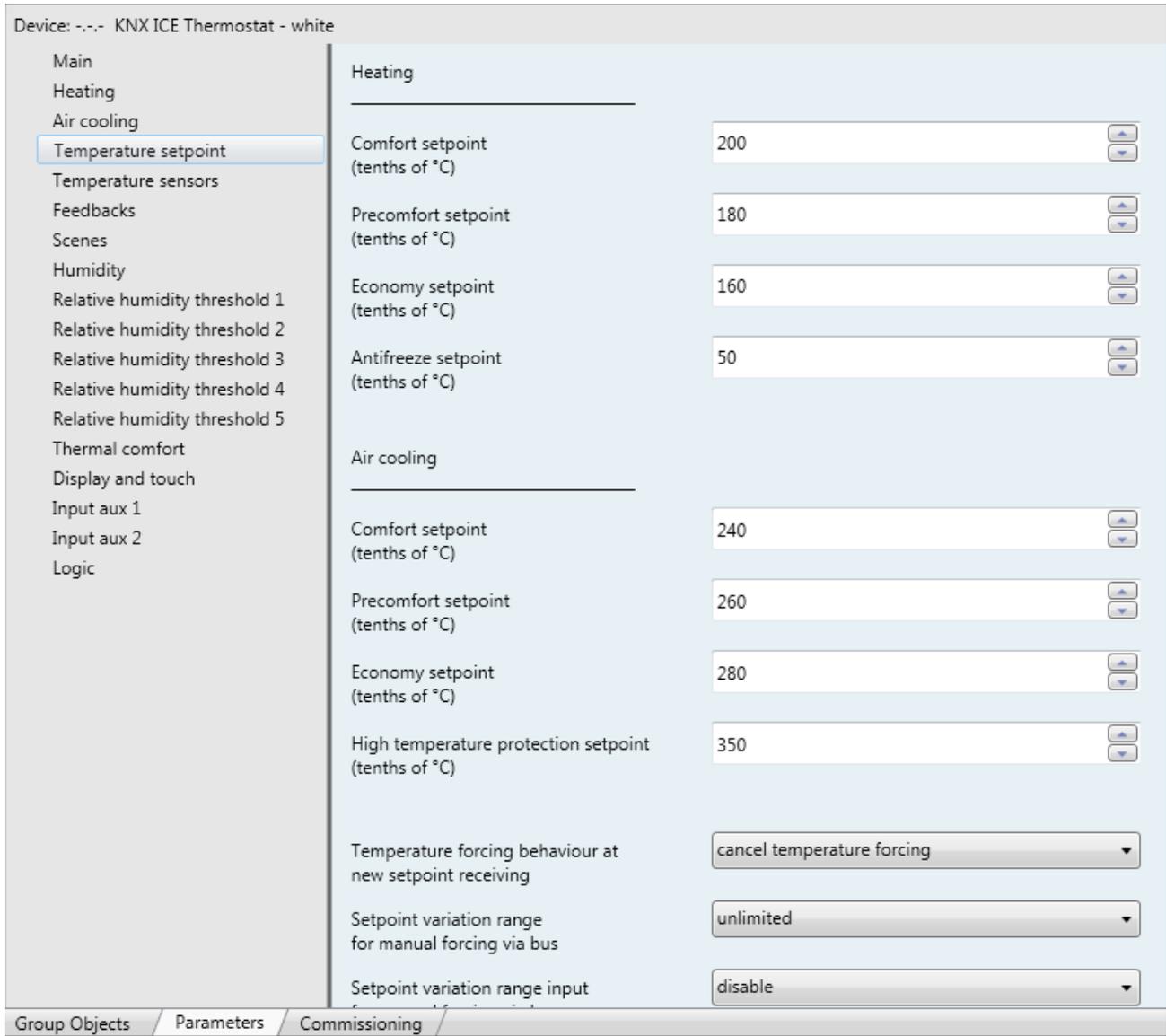


Fig. 6.1

### 6.1 Parameters

#### ➤ 6.1.1 Heating section

The “**Comfort setpoint (tenths of °C)**” parameter 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 (tenths of °C)**” for heating operation.

Remember that this value can always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Precomfort setpoint (tenth of °C)**” parameter 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Economy setpoint (tenth of °C)**” parameter 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 “**Antifreeze setpoint (tenth of °C)**” for heating operation.

Remember that this value can always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Operating setpoint (tenth of °C)**” parameter, visible if the stand alone or master/slave 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Antifreeze setpoint (tenth of °C)**” parameter 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Antifreeze setpoint after manual switch-off (tenth of °C)**” parameter, visible if the stand alone or master/slave control type is setpoint, is used to set the setpoint value for HEATING operation when the device (in slave operation) is switched-off manually by the user. The values that can be set are:

- from 20 to 70 with steps of 1 (**default value 50**)

Remember that this value can always be changed by the user using the relative parameter in the device's local navigation menu.

## ➤ 6.1.2 Air cooling section

The “**Comfort setpoint (tenth of °C)**” parameter 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Precomfort setpoint (tenth of °C)**” parameter 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Economy setpoint (tenth of °C)**” parameter 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**Operating setpoint (tenth of °C)**” parameter, visible if the stand alone or master/slave 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**High temperature protection setpoint (tenth of °C)**” parameter 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 always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The “**High temperature protection setpoint after manual switch-off (tenth of °C)**” parameter, visible if the stand alone or master/slave control type is setpoint, is used to set the setpoint value for AIR COOLING

operation when the device (in Slave operation) is switched-off manually by the user. The values that can be set are:

- from 300 to 400 with steps of 1 (**default value 350**)

Remember that this value can always be changed by the user using the relative parameter in the device's local navigation menu.

### ➤ **6.1.3 Temperature forcing behaviour at new setpoint receiving**

As the setpoint values can potentially be modified also via the local menu or remotely (regardless of whether the control type is master/slave or stand alone), if momentary forcing is applied to a setpoint and the value of this setpoint is received via BUS or locally, the device can behave in two different ways: apply the forcing to the new value, or ignore the forcing and maintain only the new value.

EXAMPLE: if the setpoint of the Comfort/Heating mode is 21 and a momentary forcing of +0.5°C is active (while active the setpoint is 21.5°C):

- if the device receives a new setpoint value equal to 22°C and forcing is maintained, then the active setpoint will be equal to 22.5°C (the mode setpoint becomes 22°C)
- if the device receives a new setpoint value equal to 22°C and forcing is cancelled, then the active setpoint will be equal to 22°C (the mode setpoint becomes 22°C)

The “**Temperature forcing behaviour at new setpoint receiving**” parameter makes it possible to set the behaviour if the above condition occurs. The values that can be set are:

- **cancel temperature forced positioning** (default value)
- maintain temperature forcing

The user can use the UP and DOWN button keys to temporarily force the active setpoint in order to personalise the room temperature. However, the adjustment range around the active setpoint can be limited and set when configuring the device. The “**Setpoint variation range for manual forcing**” parameter is used to set the maximum variation of the active setpoint value on the device for its temporary forcing via local commands. The values that can be set are:

- ± 0.0°C (no variation)
- ± 0.1°C
- ...
- **± 3.0°C** (default value if Slave)
- ...
- ± 5.0°C
- **unlimited** (default value if stand alone)

This value is expressed in °C; if the measurement unit of the display is °F, the variation range is converted into degrees Fahrenheit (e.g. ± 3.0°C → ± 37.4°F). In both cases, the maximum possible variation is what is defined by this parameter (or by the object via the BUS). If "unlimited" is selected, there are no setpoint adjustment limits (except for those between the setpoints of the same functioning type).

Via the BUS, it is possible to modify the setpoint variation range using the relative communication object. The “**Setpoint variation range input for manual forcing via BUS**” parameter is used to enable the **Setpoint adjustment range setting** communication object (Data Point Type: 9.001 DPT\_Value\_Temp) which is used to set - via a BUS telegram - the value of the setpoint adjustment interval for temporary forcing. The values that can be set are:

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

Selecting **enable** displays the **Setpoint adjustment range setting** communication object. If this communication object receives a telegram for setting the setpoint variation range with a value less than 0°C, the value is limited to 0 for reasons of safety; values above 5°C will be interpreted as an unlimited range.

The manual forcing of the setpoint is reset each time a change is made to the active HVAC mode or functioning type (regardless of Slave or stand-alone device operation).

➤ **6.1.4 At new setpoint receiving, modify the setpoints of the other functioning type**

By modifying the setpoint of a particular HVAC mode or functioning type, it may be useful to modify in the same manner the setpoint of the same mode for the opposite functioning type (especially when the device functioning type is modified autonomously via the "dead" zone). 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, the Air cooling Comfort setpoint is also automatically modified and set at 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 contemporary the setpoint of the other functioning type**". The possible values are:

- **no** (default value)
- yes

➤ **6.1.5 At new setpoint receiving, modify the setpoints of the same functioning type**

In the same manner, it may be useful to modify in the same manner the setpoints of the same functioning type (excluding the OFF mode) after only one of them has been modified. 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 the 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.

## 7 “Temperature sensors” menu

The **Temperature sensors** menu contains the parameters used to configure the operation of the sensor inside the device and the two potential external sensors: a KNX external sensor and an NTC external probe sensor. The structure of the menu is as follows:

Device: --- KNX ICE Thermostat - white

Main	Internal sensor correction factor (tenths of °C)	0
Heating	KNX external sensor	external sensor
Air cooling	- Measure unit KNX external sensor	Celsius degrees (°C)
Temperature setpoint	- KNX ext. sensor incidence in measured temp. calculation	10%
Temperature sensors	- KNX external sensor monitoring time [min] (0= no monitoring)	2
Feedbacks	- Behaviour at KNX external sensor signal absence	use last value received
Scenes	Auxiliary external sensor	external sensor
Humidity	- Type of NTC sensor connected	floating sensor (GW10800)
Relative humidity threshold 1	- Correction factor auxiliary sensor (tenths of °C)	0
Relative humidity threshold 2	- Auxiliary ext. sensor incidence in measured temp. calculation	10%
Relative humidity threshold 3	- Behaviour at auxiliary external sensor signal absence	use last value received
Relative humidity threshold 4	Auxiliary sensor measured temperature	do not send
Relative humidity threshold 5		
Thermal comfort		
Display and touch		
Input aux 1		
Logic		

Fig. 7.1

### 7.1 Parameters

#### ➤ 7.1.1 Internal sensor correction factor (tenths of °C)

This is used to set the correction factor to be applied to the temperature value measured by the sensor on the device, to eliminate the contribution of heat generated by the device or the site of installation. The values that can be set are:

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

#### ➤ 7.1.2 KNX temperature sensor function

The **KNX temperature sensor function** parameter is used to enable a communication object for measuring the room temperature or the floor temperature and, as a result, the configuration items. The values that can be set are:

- **Disabled** (default value)
- external sensor
- floor sensor

Selecting **external sensor** displays the “**KNX external sensor measurement unit**”, “**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**” parameters, and the **KNX external sensor input** communication object which is used to receive the temperature measured by the external sensor.

Selecting **floor sensor** displays the “**KNX floor sensor measurement unit**”, “**Threshold temperature alarm (tenths of °C)**”, “**Temperature alarm hysteresis (tenth of °C)**”, “**KNX floor sensor monitoring time [min] (0=no monitoring)**” and “**Behaviour at KNX floor sensor signal absence**” parameters, and the **KNX floor sensor input** communication object which is used to receive the temperature measured by the external sensor.

In any case, when the BUS voltage is restored, the device must immediately update the value received from the KNX temperature or floor sensor, sending the read request command via the **KNX external sensor input** object or the **KNX floor sensor input** object and storing the received value.

On an implementation level, even if the function is enabled in the ETS parameters, check that the relative communication object **KNX external sensor input** or **KNX floor sensor input** is connected to a valid address group before considering it enabled.

### ➤ **7.1.3 Measurement unit KNX external sensor (or KNX floor sensor)**

The “**Measure unit KNX external sensor**” (or “**Measure unit KNX floor sensor**”) parameter is used to set the unit of measure used to decode the information received via the **KNX external sensor input** (or **KNX floor sensor input**) 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 code of the **KNX external sensor input** (or **KNX floor sensor input**) communication object: *9.001 DPT\_Value\_Temp* if the value is **degrees Celsius (°C)**, *9.002 DPT\_Value\_Tempd* if the value is **degrees Kelvin (°K)** and *9.027 DPT\_Value\_Temp\_F* if the value is **degrees Fahrenheit (°F)**.

### ➤ **7.1.4 KNX external sensor incidence in measured temperature calculation**

Once the KNX external sensor is enabled, the measured temperature will not only be determined by the device sensor, but it will be determined by the weighted average between the value measured by the device sensor and the value measured by the KNX external sensor. The “**KNX external sensor incidence in measured temp. calculation**” parameter 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 probe measured value = measured temperature). The complete formula for the temperature calculation is:

$$T_{\text{measured}} = T_{\text{external sensor}} \times \text{Incidence}_{\text{external sensor}} + T_{\text{device sensor}} \times (100\% - \text{Incidence}_{\text{external sensor}})$$

The parameter may assume the following values:

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

### ➤ 7.1.5 KNX external sensor monitoring time [min] (0=no monitoring)

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 external sensor 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 device will behave differently according to the setting of the “**Behaviour at KNX external sensor signal absence**” parameter. This parameter may have the following values:

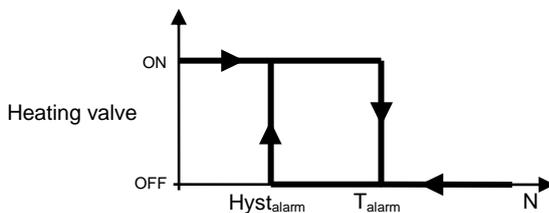
- **use last value received** (**default value**)
- exclude KNX sensor contribution

Selecting **exclude KNX sensor contribution**, the contribution of the KNX sensor will be excluded when calculating the measured temperature.

### ➤ 7.1.6 Threshold temperature alarm (tenths of °C)

This is used to define the limit floor temperature above which the device stops heating as the temperature of the conduits is too high and could cause damage (temperature alarm); the floor temperature value is received via the communication object **KNX floor sensor input**. The values that can be set are:

- from 150 to 1000 in steps of 1 (**default value 500**)



### ➤ 7.1.7 Temperature alarm hysteresis (tenth of °C)

The “**Temperature alarm hysteresis (tenth of °C)**” parameter is used to set the hysteresis threshold of the floor temperature alarm that, subtracted from the temperature alarm threshold, determines the value below which the heating system is reactivated. The values that can be set are:

- from 10 to 100 in steps of 1 (**default value 30**)

### ➤ 7.1.8 KNX floor sensor monitoring time [min] (0=no monitoring)

The “**KNX floor sensor monitoring time [min] (0=no monitoring)**” parameter is used to define the monitoring time of the KNX external floor probe. It 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 external sensor 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 device will behave differently according to the setting of the “**Behaviour at KNX floor sensor signal absence**” parameter. This parameter may have the following values:

- use last value received
- **activate temperature alarm** (default value)

### ➤ 7.1.9 Auxiliary temperature sensor function

The “**Auxiliary temperature sensor function**” parameter, displayed if the “**Input contacts function**” parameter in the **Main** menu is set at **one auxiliary input and one temperature sensor**, is used to configure auxiliary input 2 to connect an NTC temperature sensor for measuring the room temperature or the floor temperature. The terminals of auxiliary input 2 are used for this function. The values that can be set are:

- **external sensor** (default value)
- floor sensor

Selecting **external sensor** displays the “**Type of NTC sensor connected**”, “**Correction factor auxiliary sensor [tenths of °C]**”, “**Auxiliary external sensor incidence in the measured temperature calculation**”, “**Behaviour at auxiliary external sensor signal absence**” and “**Temperature measured by the auxiliary sensor**” parameters.

Selecting **floor sensor** displays the “**Type of NTC sensor connected**”, “**Correction factor auxiliary sensor [tenths of °C]**”, “**Temperature alarm threshold (tenths of °C)**”, “**Temperature alarm hysteresis (tenths of °C)**”, “**Behaviour at auxiliary floor sensor signal absence**” and “**Auxiliary sensor measured temperature**” parameters.

Different temperature sensors can be connected to the device. Given the different characteristics of each transducer, the “**Type of NTC sensor connected**” parameter defines 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)

Once the auxiliary sensor input is enabled for the external temperature sensor, the measured temperature will not only be determined by the sensor on the device, but it will be determined by the weighted average between the value measured by the sensor on the device, the possible contribution of the KNX external sensor and the value measured by the NTC auxiliary external sensor.

The “**Auxiliary ext. sensor incidence in measured temp. calculation**” parameter is used to determine the incidence of the value measured by the auxiliary 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:

$$T_{\text{measured}} = T_{\text{KNX external device}} \times \text{Incidence}_{\text{KNX external sensor}} + T_{\text{auxiliary external sensor}} \times \text{Incidence}_{\text{auxiliary external sensor}} + T_{\text{device sensor}} \times (100\% - \text{Incidence}_{\text{auxiliary external sensor}} - \text{Incidence}_{\text{external sensor}}).$$

If both the external sensors (KNX and auxiliary) are enabled, the sum of the incidences must obviously not exceed 100%; this means that if the incidence of the KNX sensor is 30%, the maximum incidence of the auxiliary sensor is 70%.

The parameter may assume the following values:

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

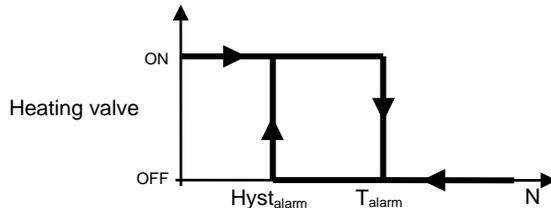
The auxiliary external sensor is always connected to the device, therefore there is no discussion in the case about sensor monitoring time; however it is possible to detect potential NTC sensor malfunction and define how the device must react via the “**Behaviour at auxiliary external sensor signal absence**” parameter. This parameter may have the following values:

- **use last value received** (default value)
- exclude auxiliary sensor contribution

Selecting **exclude auxiliary sensor contribution**, the contribution of the NTC sensor will be excluded when calculating the measured temperature.

The “**Temperature alarm threshold temperature (tenth of °C)**” parameter is used to define the floor limit temperature above which the device turns off the heating, as the temperature of the conduits is too high and could cause damage (temperature alarm). The floor temperature value is measured using the NTC sensor connected to the device on the contacts of the auxiliary sensor input. The values that can be set are:

- from 150 to 1000 in steps of 1 (**default value 500**)



The “**Temperature alarm hysteresis (tenth of °C)**” parameter is used to set the hysteresis threshold of the floor temperature alarm that, subtracted from the temperature alarm threshold, determines the value below which the heating system is reactivated. The values that can be set are:

- from 10 to 100 in steps of 1 (**default value 30**)

The auxiliary external sensor is always connected to the device, so we cannot speak - in this case - of sensor monitoring time; it is possible, however, to detect potential NTC sensor malfunction and define how the device must react via the “**Behaviour at auxiliary external floor signal absence**” parameter. This parameter may have the following values:

- use last value received
- **activate temperature alarm** (**default value**)

The “**Temperature measured by auxiliary sensor**” parameter is used to define the conditions for sending the temperature value measured by the NTC sensor connected to the device. The values that can be set are:

- **do not send** (**default value**)
- send on demand only
- send on change
- send periodically
- send on change and periodically

Selecting any value other than **do not send** displays the *Temperature measured by auxiliary sensor* communication object and the “**Measure unit auxiliary sensor**” parameter.

Selecting the value **send on change** or **send on change and periodically**, the “**Minimum auxiliary sensor temperature variation for sending value [ $\pm 0.1^\circ\text{C}$ ]**” parameter will also be visible, whereas by selecting **send periodically** or **send on change and periodically** the “**Auxiliary sensor temperature sending period [minutes]**” parameter 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 auxiliary sensor**” parameter is used to set the measure unit with which the information will be coded and sent via the *Temperature measured by auxiliary sensor* communication object. The values that can be set are:

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

The value set for this parameter changes the coding of the *Temperature measured by auxiliary sensor* communication object: *9.001 DPT\_Value\_Temp* if the value is **degrees Celsius (°C)**, *9.002 DPT\_Value\_Tempd* if the value is **degrees Kelvin (°K)** and *9.027 DPT\_Value\_Temp\_F* if the value is **degrees Fahrenheit (°F)**.

The “**Minimum auxiliary sensor temperature variation for sending value [± 0.1 °C]**” parameter, which is visible if the temperature of the auxiliary sensor 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 value measured by the NTC sensor. The values that can be set are:

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

The “**Auxiliary sensor temperature sending period [minutes]**” parameter, which is visible if the auxiliary sensor 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)**

In the case of a floor temperature alarm (regardless of whether it was generated by the KNX or the auxiliary external sensor), the green LED will flash.

## 8 "Feedback" menu

The **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:

Parameter	Value
Measured temperature	on variation and periodically
- Measure unit	Celsius degrees (°C)
- Minimum variation for sending value [± 0.1 °C]	5
- Temperature sending period [minutes]	5
HVAC mode feedback	disabled
Functioning type feedback	disabled
Current setpoint report	disabled
Thermostat functioning feedback	disabled
HVAC mode setpoint feedback	disabled
Feedbacks sending trigger object	disable

Fig. 8.1

### 8.1 Parameters

#### ➤ 8.1.1 Measured temperature

This is used to define the conditions for sending the value of the temperature measured by the device (which may or may not be influenced by the external sensor). The values that can be set are:

- **do not send** (default value)
- send on demand only
- send on change
- send periodically
- send on change and periodically

Selecting any value other than **do not send** displays the **Measured temperature** communication object and the **“Measure unit”** parameter. Selecting the value **send in case of change** or **send on change and periodically**, the **“Minimum temperature variation for sending value [± 0.1°C]”** parameter will also be visible, whereas by selecting **send periodically** or **send on change and periodically** the **“Temperature sending period [minutes]”** parameter 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.

### ➤ 8.1.2 Measure unit

This is used to set the measure unit with which the information will be coded and sent via the **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 **Temperature sensor** communication object: 9.001 DPT\_Value\_Temp if the value is **degrees Celsius (°C)**, 9.002 DPT\_Value\_Tempd if the value is **degrees Kelvin (°K)** and 9.027 DPT\_Value\_Temp\_F if the value is **degrees Fahrenheit (°F)**.

### ➤ 8.1.3 Minimum temperature variation for sending value [± 0.1°C]

This is visible if the temperature is sent on variation, and is used to define the minimum temperature variation (in relation to the last temperature value sent) which 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)

### ➤ 8.1.4 Temperature sending period [minutes]

This is visible if the temperature is sent periodically, and is used to define the period after which the measured temperature indication telegrams are sent spontaneously. The values that can be set are:

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

### ➤ 8.1.5 HVAC mode feedback

The “**HVAC mode feedback**” parameter, visible if the master/slave or stand alone control type is in HVAC mode, is used to enable and set the format of the BUS telegrams used by the device to indicate the active HVAC mode. The values that can be set are:

- **disabled** (default value)
- 1 bit
- 1 byte
- both

Selecting **1 bit** or **both** displays the “**1 bit HVAC mode**” parameter and the **HVAC off mode feedback**, **HVAC economy mode feedback**, **HVAC precomfort mode feedback**, **HVAC comfort mode feedback** and **HVAC auto mode feedback** communication objects (Data Point Type: 1.003 DPT\_Enable) which are used to signal the active HVAC mode. When a mode is actually active, this status is signalled via the BUS telegram on the object associated with the new mode and at the same time, mode deactivation feedback is sent on the object associated with the mode that was previously active. There is no case in which multiple activated temperature adjustment modes are signalled. Selecting the value **1 byte** or **both** displays the “**1 byte HVAC mode**” parameter and the **HVAC mode feedback** communication object (Data Point Type: 20.102 DPT\_HVACMode) which are used to signal the active HVAC mode.

The “**1 bit HVAC mode**” parameter is used to set the conditions for sending the operating mode feedback via the **HVAC off mode feedback**, **HVAC economy mode feedback**, **HVAC precomfort mode feedback**, **HVAC comfort mode feedback** and **HVAC auto mode feedback** communication objects with a 1-bit size. The values that can be set are:

- on demand only
- **on variation** (default value)

By selecting **on demand only**, the operating mode feedback is not sent spontaneously by the device via the **HVAC off mode feedback**, **HVAC economy mode feedback**, **HVAC precomfort mode feedback**, **HVAC comfort mode feedback** and **HVAC auto mode feedback** communication objects with a 1 bit size; 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 status of the operating mode relative to the object on which the request was made. This means that, when there is a status reading request on one of the objects cited above, the device responds with the status of that mode (active/deactivated) and not with the status of the mode set on the device, which is instead the case of a 1 byte object.

By selecting **on variation**, the operating mode feedback is sent spontaneously by the device via the **HVAC off mode feedback**, **HVAC economy mode feedback**, **HVAC precomfort mode feedback**, **HVAC comfort mode feedback** and **HVAC auto mode feedback** communication objects with a 1 bit size, each time the mode is varied. This means that, every time the device HVAC mode is modified, it signals the activation of the new mode via the communication object associated with it and signals the deactivation of the previously active mode via the communication object associated with the latter. After a BUS voltage recovery, the value of all the 1-bit objects should be sent in order to update any connected devices.

The “**1 byte HVAC mode**” parameter is used to set the conditions for sending the HVAC mode feedback via **HVAC mode feedback** communication objects with a 1-byte size. The values that can be set are:

- on demand only
- **on variation** (default value)

Selecting **on demand only**, the HVAC mode feedback will not be sent spontaneously by the device via the **HVAC mode feedback** communication object with a 1 byte size. 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 **on variation**, the HVAC mode feedback will be sent spontaneously by the device via the **HVAC mode feedback** communication object with a 1 byte size, 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.

### ➤ **8.1.6 Functioning type feedback**

This is used to enable and set the conditions for sending feedback about the set functioning type (Heating/Air cooling) by the device via the BUS telegram on the **Functioning type feedback** communication object (Data Point Type: 1.100 DPT\_Heat/Cool). The values that can be set are:

- **Disabled** (default value)
- send on demand only
- send on variation

Selecting **send on demand only** the feedback of the functioning type set on the device is not sent spontaneously by the device via the **Functioning type feedback** communication object. 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 type of functioning set on the device.

Selecting **send on variation**, the feedback of the functioning type set on the device is sent spontaneously by the device via the **Functioning type feedback** communication object each time the functioning type changes. After a BUS voltage recovery, feedback about the type of active operation should be sent in order to update any connected devices.

### ➤ **8.1.7 Functioning setpoint feedback**

The “**Functioning setpoint feedback**” parameter, visible if the Master/Slave or stand-alone control type is setpoint, is used to enable and set the conditions for sending feedback about the functioning setpoint value (the value stored - any temporary forcing is NOT taken into account) set on the device, via a BUS telegram on the **Functioning setpoint feedback** communication object (Data Point Type: 9.001 DPT\_Temp if object in °C, 9.002 DPT\_Tempd if object in °K and 9.027 DPT\_Value\_Temp\_F if object in °F). The values that can be set are:

- **disabled** (default value)
- send object (°C) on demand only
- send object (°K) on demand only
- send object (°F) on demand only
- send object (°C) on variation
- send object (°K) on variation
- send object (°F) on variation

By selecting **send object in (°C) on demand only**, **send object in (°K) on demand only** or **send object in (°F) on demand only**, the feedback regarding the active setpoint on the device is not sent spontaneously by the device via the **Current setpoint feedback** communication object. 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. By selecting **send object in (°C) on variation**, **send object in (°K) on variation** or **send object in (°F) on variation**, the feedback regarding the active setpoint on the device is sent spontaneously by the device via the **Current setpoint feedback** communication object each time there is a variation in the setpoint itself (also following a temporary forcing). After a BUS voltage recovery, feedback about the active setpoint should be sent in order to update any connected devices. If the device is in building protection mode (following manual switch-off or the opening of the window contact), the functioning setpoint is not updated with the building protection setpoint value (only the current setpoint is updated).

### ➤ **8.1.8 Current setpoint feedback**

The “**Current setpoint feedback**” parameter is used to enable and set the conditions for sending feedback regarding the current setpoint value (which takes into account any temporary forcing) set on the device, via a BUS telegram on the **Current setpoint feedback** communication object (Data Point Type: 9.001 DPT\_Temp if object in °C, 9.002 DPT\_Tempd if object in °K and 9.027 DPT\_Value\_Temp\_F if object in °F). The values that can be set are:

- **disabled** (default value)
- send object (°C) on demand only
- send object (°K) on demand only
- send object (°F) on demand only
- send object (°C) on variation
- send object (°K) on variation
- send object (°F) on variation

By selecting **send object in (°C) on demand only**, **send object in (°K) on demand only** or **send object in (°F) on demand only**, the feedback regarding the active setpoint on the device is not sent spontaneously by the device via the **Current setpoint feedback** communication object. 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.

By selecting **send object in (°C) on variation**, **send object in (°K) on variation** or **send object in (°F) on variation**, the feedback regarding the active setpoint on the device is sent spontaneously by the device via the **Current setpoint feedback** communication object each time there is a variation in the setpoint itself (also following a temporary forcing). After a BUS voltage recovery, feedback about the active setpoint should be sent in order to update any connected devices.

### ➤ **8.1.9 Thermostat functioning feedback**

This is used to enable and set the conditions for sending feedback about the thermostat functioning type (Stand alone/Slave) set on the device, via a BUS telegram on the **Thermostat functioning feedback** communication object (Data Point Type: 1.001 DPT\_Switch). Logical value “1” corresponds to SLAVE operation; logical value “0” corresponds to STAND ALONE operation. The values that can be set are:

- **Disabled** (default value)
- send on demand only

- send on variation

Selecting **send on demand only**, the feedback of the thermostat functioning type is not sent spontaneously by the device via the **Thermostat functioning feedback** communication object. 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 type of timed thermostat functioning. Selecting **send on variation**, the feedback of the functioning type set on the device is sent spontaneously by the device via the **Thermostat functioning feedback** communication object, each time the functioning type changes. After a BUS voltage recovery, feedback about the thermostat operation should be sent in order to update any connected devices.

### ➤ 8.1.10 HVAC mode setpoint feedback

The “**HVAC mode setpoint feedback**” parameter, visible if the Master/Slave or stand-alone control type is HVAC mode, is used to enable the sending of the setpoint value of the HVAC mode via the **Heating antifreeze setpoint feedback**, **Heating economy setpoint feedback**, **Heating precomfort setpoint feedback**, **Heating comfort setpoint feedback**, **Air cooling high temperature protection setpoint feedback**, **Air cooling economy setpoint feedback**, **Air cooling precomfort setpoint feedback** and **Air cooling comfort setpoint feedback** objects (Data Point Type: 9.001 DPT\_Temp if object in °C, 9.002 DPT\_Temp if object in °K and 9.027 DPT\_Value\_Temp\_F if object in °F).

The values that can be set are:

- **disabled** (default value)
- send object (°C) on demand only
- send object (°K) on demand only
- send object (°F) on demand only
- send object (°C) on variation
- send object (°K) on variation
- send object (°F) on variation

By selecting **send object in (°C) on demand only**, **send object in (°K) on demand only** or **send object in (°F) on demand only**, the feedback regarding the HVAC mode setpoints is not sent spontaneously by the device via the **Heating antifreeze setpoint feedback**, **Heating economy setpoint feedback**, **Heating precomfort setpoint feedback**, **Heating comfort setpoint feedback**, **Air cooling high temperature protection setpoint feedback**, **Air cooling economy setpoint feedback**, **Air cooling precomfort setpoint feedback** and **Air cooling comfort setpoint feedback** communication objects. 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 setpoint associated with the device.

By selecting **send object in (°C) on variation**, **send object in (°K) on variation** or **send object in (°F) on variation**, the feedback regarding the HVAC mode setpoints is sent spontaneously by the device via the **Heating antifreeze setpoint feedback**, **Heating economy setpoint feedback**, **Heating precomfort setpoint feedback**, **Heating comfort setpoint feedback**, **Air cooling high temperature protection setpoint feedback**, **Air cooling economy setpoint feedback**, **Air cooling precomfort setpoint feedback** and **Air cooling comfort setpoint feedback** communication objects each time the functioning type changes. After a BUS voltage recovery, feedback about the mode setpoints should be sent in order to update any connected devices.

### ➤ 8.1.11 Feedbacks sending trigger object

The “**Feedbacks sending trigger object**” parameter is used to enable the **Feedback sending trigger** input object (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 feedback present in the **Feedback** menu, for which sending is “enabled on variation” (including the option “periodically” for the measured temperature value).

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

## 9 "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. 9.1

### 9.1 Parameters

#### ➤ 9.1.1 Scene function

This is used to activate and configure the function, displaying the various function configuration parameters and the relative **Thermostat scene** communication object (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 “**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 memorise while storing**” parameters, and the **Thermostat scene** communication object via which the scene execution/memorise telegrams are received.

### ➤ 9.1.2 Scene number *i*

With the “**Scene number *i***” ( $1 \leq i \leq 8$ ) parameter, it is possible to 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

### ➤ 9.1.3 Parameters to memorise while storing

As the thermostat has various operating parameters that can change during its operation, the “**Parameters to memorise while storing**” parameter can be used to establish 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 Master/Slave or stand-alone control type is “HVAC mode”, are:

- **HVAC mode only** (default value)
- HVAC mode and functioning type
- HVAC mode, functioning type and forcing

The values that can be set, if the Master/Slave or stand-alone control type is “setpoint”, are:

- **setpoint only** (default value)
- setpoint and functioning type
- setpoint, functioning type and forcing

If the scene storing command is received when the building protection setpoint is active, it will be ignored.

If the storing of the functioning type is enabled, when the scene storing command is received, if the functioning type is set automatically via the dead zone, this setting will be stored. Upon the request to execute the scene, the dead zone will be reactivated even if it was deactivated before the command was received.

## 10 “Humidity” menu

The **Humidity** menu contains parameters used to configure the operation of the internal humidity sensor and a potential external sensor.

The structure of the menu is as follows:

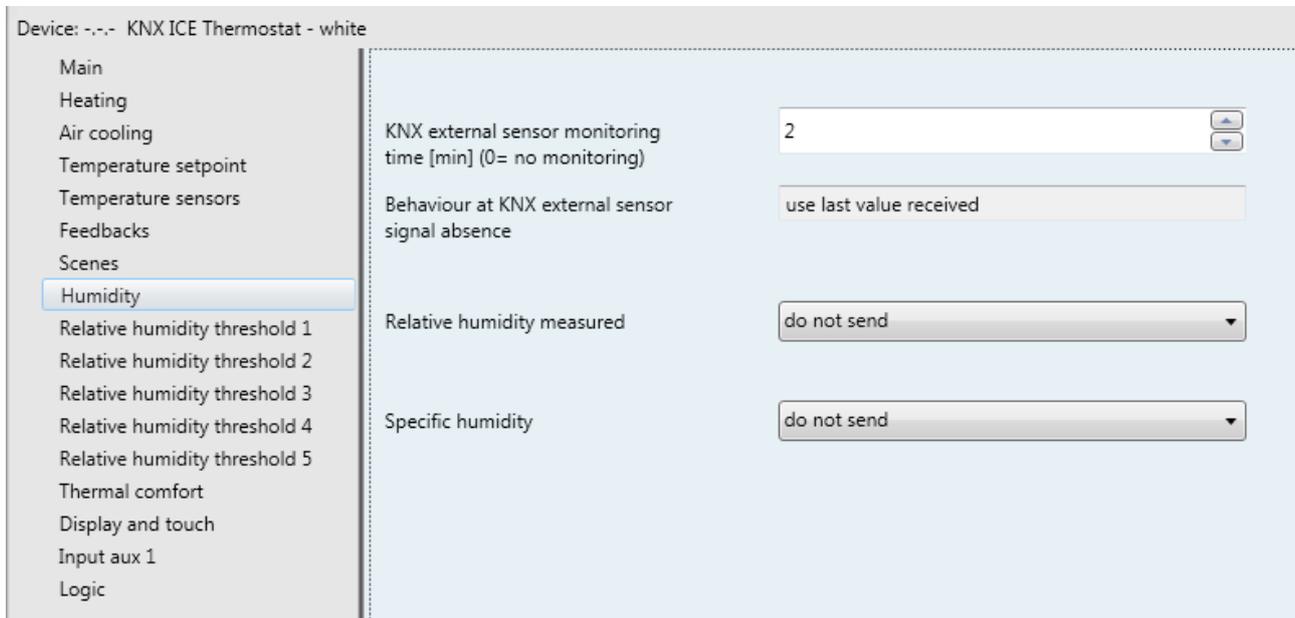


Fig. 10.1

As there is no internal humidity sensor, the relative humidity measured by the device is determined solely by the external KNX humidity sensor. The relative humidity values are received by the device via the **KNX external humidity sensor input** communication object (Data Point Type: 9.007 DPT\_Value\_Humidity).

### 10.1 Parameters

#### ➤ 10.1.1 KNX external sensor monitoring time [min] (0=no monitoring)

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 device will behave differently according to the setting of the “**Behaviour at KNX external sensor signal absence**” parameter.

#### ➤ 10.1.2 Behaviour at KNX external sensor signal absence

This parameter may have the following values: **use last value received** (default value)

### ➤ 10.1.3 Relative humidity measured

The “**Relative humidity measured**” parameter defines which event should trigger the sending of the **Relative humidity measured** object on the BUS (Data Point Type: 9.007 DPT\_Value\_Humidity). The parameter may assume the following values:

- **do not send** (default value)
- send on demand only
- send on change
- send periodically
- send on change and periodically

Selecting any value other than **do not send** displays the **Relative humidity measured** communication object.

Selecting **send on variation** or **send on variation and periodically** displays the “**Minimum variation for sending value [% RH]**” parameter, whereas selecting **send periodically** or **send on variation and periodically** displays the “**Relative humidity sending period [minutes]**” parameter.

Selecting the value **send on demand only**, no new parameter will be enabled, as the humidity 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 humidity value.

The “**Minimum variation for sending value**” parameter, displayed if the humidity is sent on variation, is used to define the minimum humidity variation with respect to the last humidity value that was sent, which generates the spontaneous sending of the new measured value. The values that can be set are:

- ± 1%
- ± 2%
- **± 5%** (default)
- ± 10%

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

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

### ➤ 10.1.4 Specific humidity

The device can calculate the specific humidity of the place where it is located. Follow these indications:

We refer to  $X$  specific humidity in kg / kg a.s. and, knowing that the vapour pressure equation is the following:

$$p_v = \frac{p \cdot X}{0.622 + X}$$

where  $p = 101325$  Pa

we can obtain:

$$X = \frac{p_v * 0.622}{p - p_v}$$

the “**Specific humidity**” parameter that defines any conditions for sending the specific humidity value measured. The parameter may assume the following values:

- **do not send** (default value)
- send on demand only
- send on change
- send periodically
- send on change and periodically

Selecting any value other than **do not send** displays the **Specific humidity** communication object (Data Point Type: 14.005 DPT\_Value\_Amplitude). Selecting **send on variation** or **send on variation and periodically** displays the “**Minimum variation for sending value**” parameter, whereas selecting **send periodically** or **send on variation and periodically** displays the “**Specific humidity sending period [minutes]**” parameter.

Selecting the value **send on demand only**, no new parameter will be enabled, as the humidity 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 specific humidity value.

The “**Minimum variation for sending value**” parameter, displayed if the specific humidity is sent on variation, is used to define the minimum specific humidity variation with respect to the last specific humidity value that was sent, which generates the spontaneous sending of the new measured value. The values that can be set are:

- $\pm 1$  g/Kg
- $\pm 2$  g/Kg
- **$\pm 5$  g/Kg** (default)
- $\pm 10$  g/Kg

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

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

ATTENTION: If the internal humidity sensor is not present, then the humidity is taken directly from KNX via the dedicated communication object; the incidence of the external sensor will be equal to 100% and if the signal from the external sensor is absent the last received value will be maintained.

## 11 “Relative humidity threshold x” menu

The device makes it possible to configure 5 relative humidity thresholds to which the sending of different BUS commands is associated when the fixed threshold value is exceeded. All 5 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 general “x” (1 .. 5).

The structure of the menu is as follows:

The screenshot shows the configuration interface for a KNX ICE Thermostat. The left sidebar lists various menu items, with 'Relative humidity threshold 1' selected. The main area displays the following configuration options:

- Relative humidity threshold 1: enable
- Threshold logic functioning: humidification
- C1 = Condition 1: Relative humidity <= Limit threshold - hysteresis
- C2 = Condition 2: Relative humidity >= Limit threshold
- Limit threshold starting value [% RH]: 50
- Limit threshold hysteresis [% RH]: 5
- Change the threshold value via bus through: absolute value
- Enable/disable the humidity threshold (via bus and device menu): no
- Object A: -----
- Output format: 1 bit
- On the occurrence of Condition 1: sends 1
- On the occurrence of Condition 2: no effect
- Relative humidity threshold output status feedback: enable
- Object B: -----

Fig. 11.1

### 11.1 Parameters

#### ➤ 11.1.1 Relative humidity threshold x

The “Relative humidity threshold x” parameter allows you to activate and configure the function, making the various function configuration parameters and communication objects visible. The values that can be set are:

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

Selecting **enable** displays the configuration parameters and the **Relative humidity threshold x feedback** communication object (Data Point Type: 9.007 DPT\_Value\_Humidity); the telegrams are sent via this object following a BUS request, spontaneously upon each threshold variation and when BUS voltage is restored.

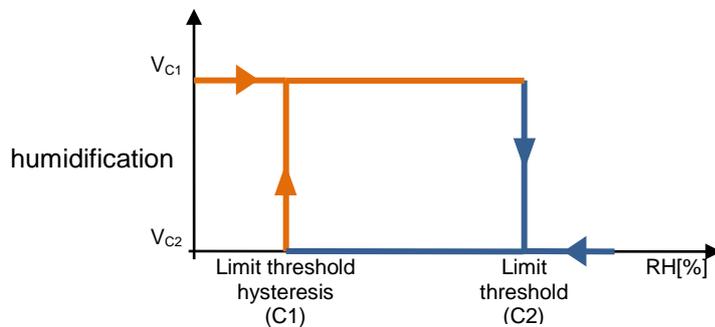
### ➤ 11.1.2 Threshold logic functioning

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

- **humidification (default value)**
- dehumidification

By selecting humidification, the two conditions will be defined as follows:

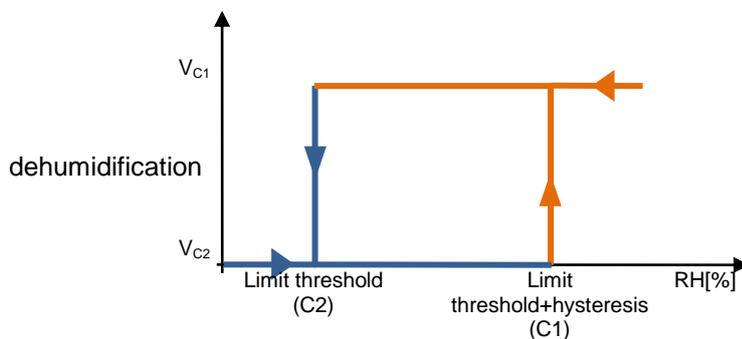
Condition 1 = Relative humidity  $\leq$  Limit threshold – Hysteresis  
 Condition 2 = Relative humidity  $\leq$  Limit threshold



When the relative humidity of reference is lower than value C1 (limit threshold hysteresis”), the device will sent the command associated with Condition 1. When the relative humidity of reference reaches value C2 (limit threshold), the device will sent the command associated with Condition 2.

By selecting dehumidification, the two conditions will be defined as follows:

Condition 1 = Relative humidity  $\geq$  Limit threshold + Hysteresis  
 Condition 2 = Relative humidity  $\leq$  Limit threshold



When the relative humidity of reference is higher than value C1 (limit threshold+hysteresis”), the device will sent the command associated with Condition 1. When the relative humidity of reference reaches value C2 (limit threshold), the device will sent the command associated with Condition 2.

### ➤ 11.1.3 Limit threshold starting value [%RH]

Via the “**Limit threshold starting value [%RH]**” parameter it is possible to set the initial value of the limit threshold associated with the relative humidity threshold x (which can be changed via BUS via the specific communication object). The parameter may assume the following values:

- from 0 to 100 in steps of 1 (**default value 50**)

### ➤ 11.1.4 Limit threshold hysteresis [%RH]

The “**Limit threshold hysteresis [%RH]**” parameter is 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 20 with steps of 1, (**default value 5**)

### ➤ 11.1.5 Change the threshold value via BUS through

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

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

Selecting **absolute value** displays the *Relative humidity threshold x input* object (Data Point Type: 9.007 DPT\_Value\_Humidity) which is used to directly change the limit threshold.

Selecting **increase/decrease step** displays the “**Threshold regulation step via BUS [%]**” parameter and the *Relative humidity threshold x regulation* object (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 “**Threshold regulation step via BUS [%]**” parameter. If the value “0” is received on this object, the limit threshold value will be decreased temporarily by the value defined by the “**Threshold regulation step via BUS [%]**” parameter.

The **Threshold regulation step via BUS [%]** parameter 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, (**default value 5**)

### ➤ 11.1.6 Enable/disable humidity threshold (via BUS and local menu)

The “**Enable/disable humidity threshold (via bus and local menu)**” parameter is used to enable the possibility to activate/deactivate the relative humidity threshold x via the parameter in the local thermostat menu and/or via a specific communication object. The parameter may assume the following values:

- **no** (default value)
- yes

Selecting **yes** displays the *Relative humidity threshold x enabling* (Data Point Type:1.002 DPT\_Boolean) and *Relative humidity threshold x enabling status* (Data Point Type:1.003 DPT\_Enable) communication objects to receive the threshold activation commands and threshold activation status feedback and the “**Threshold enabling value**” and “**Threshold enabling status when BUS voltage is restored**”

parameters. The humidity threshold can therefore be enabled/disabled via the BUS or via the relative parameter in the local device menu.

The “**Threshold enabling value**” parameter determines which logical value received via the **Relative humidity threshold x enabling** communication object activates the relative humidity threshold x. If the value opposite the one set for activation is received, this will deactivate the threshold. The possible values are:

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

The “**Threshold enabling status when BUS voltage is restored**” parameter is used to set the status of the relative humidity threshold x after the BUS power supply voltage is restored. The possible values are:

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

### ➤ 11.1.7 Object A/B/C/D

For each threshold, up to 4 different objects can be sent (which are distinguished by the letters A, B, C and D) depending on the occurrence of conditions 1 and 2; object A is always enabled, whereas the “**Object z**” parameter (where z is the index of the object associated with the threshold, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

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

Selecting **enable** displays the **Relative humidity threshold x output z** object (where z is the index of the object associated with the threshold, between **A** and **D**) and the “**Output format**”, “**On the occurrence of condition 1**” and “**On the occurrence of condition 2**” parameters grouped in the **Channel x Object z** sub-group (z is the index of the object associated with the channel, between **A** and **D**).

The “**Output format**” parameter is used to set the format and coding of the BUS telegram that will be sent by the device. The values that can be set are:

- 1 bit
- 2 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 **Relative humidity threshold x output z** communication object (Data Point Type: 1.001 DPT\_Switch) and the “**Relative humidity threshold output status feedback**” parameter. 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 “**Relative humidity threshold output status feedback**” parameter is used to enable and display the **Relative humidity threshold x output status feedback z** communication object (Data Point Type: 1.001 DPT\_Switch). Enabling this object, the command on the **Relative humidity threshold x output z** object will be repeated until the status received on the **Relative humidity threshold x output status feedback z** object coincides with the command. Any possible repetition of the command will take place periodically, once a minute. The parameter may assume the following values:

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

Selecting **enable** displays the **Relative humidity threshold x output status z 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 interface about the status of the devices connected if condition C1 or C2 has occurred (otherwise, no request is sent).

Selecting **disable** displays the “**Cyclical repetition of the output commands**” parameter.

- If the output format is **2 bit**, this displays the **Relative humidity threshold x output z** communication object (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 value**, this displays the **Relative humidity threshold x output z** communication object (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 value**, this displays the **Relative humidity threshold x output z** communication object (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 **Relative humidity threshold x output z** communication object (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 **Relative humidity threshold x output z** communication object (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 current HVAC mode ± offset
    - send HVAC mode of reference ± offset

Selecting **send current HVAC mode ± offset** displays the “**Offset (-3 .. +3)**” parameter, whereas the output value will be the current HVAC mode of the device to which the offset will be added (the order of the modes is: auto, comfort, precomfort, economy, off); if device operation is at setpoint, no output value will be sent.

Selecting **send HVAC mode of reference ± offset** displays the “**Offset (-3 .. +3)**” parameter and the **Reference for relative humidity threshold x for output z** communication object (Data Point Type: 20.102 DPT\_HVACMode). In this case, the output value will be the HVAC mode received via the **Reference for relative humidity threshold x for output z** object 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”.

Each time the BUS voltage or auxiliary voltage is reset, send the read request command on the object **Reference for relative humidity threshold x for output z** to update the thermostat about the status of the connected devices.

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)**” parameter 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 **Relative humidity threshold x output z** object. The possible values are:

- from -3 to +3 with steps of 1 (default value +1)
- If the output format is **2 byte unsigned value**, this displays the **Relative humidity threshold x output z** communication object (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 value**, this displays the **Relative humidity threshold x output z** communication object (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 **Relative humidity threshold x output z** communication object (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 current setpoint ± offset** (default value cond 1)
- send reference setpoint ± offset

Selecting **send current setpoint ± offset** displays the “**Offset [0.1°C] (-300 .. +300)**” parameter, whereas the output value will be the current device setpoint to which the offset is added).

Selecting **send reference setpoint ± offset** displays the “**Offset [0.1°C] (-300 .. +300)**” parameter and the **Reference for relative humidity threshold x for output z** communication object (Data Point Type: 9.001 DPT\_Value\_Temp). In this case, the output value will be the setpoint received via the **Reference for relative humidity threshold x for output z** object to which the offset will be added. If nothing was ever received on the reference object, the initial value is equal to “20°C”.

Each time the bus voltage or auxiliary voltage is reset, send the read request command on the object **Reference for relative humidity threshold x for output z** to update the thermostat about the status of the connected devices.

The “**Offset [0.1°C] (-300 .. +300)**” parameter is used to set the offset to be applied to the current or reference setpoint to obtain the value to be sent via the **Relative humidity threshold x output z** object. 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 **Relative humidity threshold x output z** communication object (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 current setpoint + offset** (default value cond 1)
- send reference setpoint + offset

Selecting **send current setpoint ± offset** displays the “**Offset [0.1°C] (-300 .. +300)**” parameter, whereas the output value will be the current device setpoint to which the offset is added).

Selecting **send reference setpoint ± offset** displays the “**Offset [0.1°C] (-300 .. +300)**” parameter and the **Reference for relative humidity threshold x for output z** communication object (Data Point Type: 9.002 DPT\_Value\_Tempd). In this case, the output value will be the setpoint received via the **Reference for relative humidity threshold x for output z** object to which the offset will be added. If nothing was ever received on the reference object, the initial value is equal to “293°K”.

Each time the BUS voltage or auxiliary voltage is reset, send the read request command on the object **Reference for relative humidity threshold x for output z** to update the thermostat about the status of the connected devices

The “**Offset [0.1°C] (-300 .. +300)**” parameter is used to set the offset to be applied to the current or reference setpoint to obtain the value to be sent via the **Relative humidity threshold x output z** object. 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 **Relative humidity threshold x output z** communication object (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 current setpoint + offset** (default value cond 1)
  - send reference setpoint + offset

Selecting **send current setpoint ± offset** displays the “**Offset [0.1°C] (-300 .. +300)**” parameter, whereas the output value will be the current device setpoint to which the offset is added).

Selecting **send reference setpoint ± offset** displays the “**Offset [0.1°C] (-300 .. +300)**” parameter and the **Reference for relative humidity threshold x for output z** communication object (Data Point Type: 9.027 DPT\_Value\_Temp\_F). In this case, the output value will be the setpoint received via the **Reference for relative humidity threshold x for output z** object to which the offset will be added. If nothing was ever received on the reference object, the initial value is equal to “68°F”.

Each time the bus voltage or auxiliary voltage is reset, send the read request command on the object **Reference for relative humidity threshold x for output z** to update the thermostat about the status of the connected devices.

The “**Offset [0.1°C] (-300 .. +300)**” parameter is used to set the offset to be applied to the current or reference setpoint to obtain the value to be sent via the **Relative humidity threshold x output z** object. The possible values are:

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

The **Relative humidity threshold x output z** object 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 if condition C1 or C2 has occurred (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 “**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)

## 12 “Thermal comfort” menu

The **Thermal comfort** menu contains parameters used to enable and configure the thermal comfort feedback of the environment derived from the measurement of the relative humidity and the measured temperature.

The structure of the menu is as follows:

Device: --.- KNX ICE Thermostat - white

Main	Enable thermal comfort status feedback	enable
Heating	The season summer/winter	follows active functioning type
Air cooling	Limit conditions for comfort environment in the summer:	
Temperature setpoint	Maximum temperature [0.1 °C]	260
Temperature sensors	Minimum temperature [0.1 °C]	240
Feedbacks	Maximum relative humidity [% RH]	60
Scenes	Minimum relative humidity [% RH]	40
Humidity	Maximum specific humidity [% RH]	115
Relative humidity threshold 1	Limit conditions for comfort environment in the winter:	
Relative humidity threshold 2	Maximum temperature [0.1 °C]	220
Relative humidity threshold 3	Minimum temperature [0.1 °C]	180
Relative humidity threshold 4	Maximum relative humidity [% RH]	60
Relative humidity threshold 5	Minimum relative humidity [% RH]	40
Thermal comfort	Maximum specific humidity [% RH]	115
Display and touch		
Input aux 1		
Logic		

Fig. 12.1

### 12.1 Parameters

#### ➤ 12.1.1 Enable thermal comfort status feedback

The “**Enable thermal comfort status**” parameter is used to enable the function, making the configuration parameters visible. The possible values are:

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

Selecting **enable** displays all the objects and parameters relative to the thermal comfort feedback function.

### ➤ 12.1.2 The season (summer/winter)

To determine the thermal comfort status, it is necessary to know the current season (summer or winter); the “**The season summer/winter**” parameter establishes how the season is set. The possible values are:

- **follows active functioning type** (default)
- depends on BUS object input

By selecting **follows active functioning type**, the summer or winter season will be inferred from the type of operation active on the device: summer if the active type of operation is air cooling, winter if heating is active. Selecting **depends on BUS object input** displays the **Season selection input** object (Data Point Type 1.002 DPT\_Bool) and the “**Logical value associated with the seasons**” and “**Active season following an ETS download**” parameters. With this setting, the summer or winter season depends on the value of the communication object.

The “**Logical value associated with the seasons**” parameter defines which logical value (received via the **Season selection input** communication object) is associated with summer and with winter. The possible values are:

- 1 = winter / 0 = summer
- **1 = summer / 0 winter** (default value)

The “**Active season following an ETS download**” parameter is used to preselect the season after download from ETS. The possible values are:

- **summer** (default value)
- winter

the setting is valid until a value is received on the **Season selection input** object.

### ➤ 12.1.3 Limit conditions for comfortable environment in the summer/winter

The environment comfort condition depends on the current relative humidity and temperature value; if both the current values for these two variables lie within the limit values, then the environment is defined as “comfortable”.

For summer and for winter, the limit temperature and humidity values must be defined in order to define the environment as “comfortable”.

The maximum temperature in summer or winter, depending on the subgroup to which the parameter belongs, is defined via the “**Maximum temperature [0.1°C]**” parameter, which can assume the following values:

- from 200 to 400 in steps of 1 (**default value for summer 260, default value for winter 220**)

The minimum temperature in summer or winter, depending on the subgroup to which the parameter belongs, is defined via the “**Minimum temperature [0.1°C]**” parameter, which can assume the following values:

- from 200 to 400 in steps of 1 (**default value for summer 240, default value for winter 180**)

The maximum relative humidity in summer or winter, depending on the subgroup to which the parameter belongs, is defined via the “**Maximum relative humidity [°RH]**” parameter, which can assume the following values:

- from 50 to 95 in steps of 1 (**default value 60**)

The minimum relative humidity in summer or winter, depending on the subgroup to which the parameter belongs, is defined via the “**Minimum relative humidity [°RH]**” parameter, which can assume the following values:

- from 10 to 45 in steps of 1 (**default value 40**)

The specific maximum humidity in summer or winter, depending on the subgroup to which the parameter belongs, is defined via the “**maximum specific humidity [0.1 g/kg]**” parameter, which can assume the following values:

- from 50 to 220 in steps of 1 (**default value 115**)

Once the limit values are defined, the environment is comfortable if all the three conditions indicated below are observed:

- Minimum temperature  $\leq$  Current temperature  $\leq$  Maximum temperature
- Minimum relative humidity  $\leq$  Current relative humidity  $\leq$  Maximum relative humidity
- Current specific humidity  $\leq$  Maximum specific humidity

The ***Thermal comfort status feedback*** object (Data Point Type 1.002 DPT\_Bool) is used to send the comfort status of the environment in which the device is installed on the BUS. If the three conditions have all been observed (comfortable environment), the object takes the logical value “1”; otherwise, the value is “0”. The value of the object is sent on demand and spontaneously on variation and when the BUS is restored.

## 13 “Display and touch push-button” menu

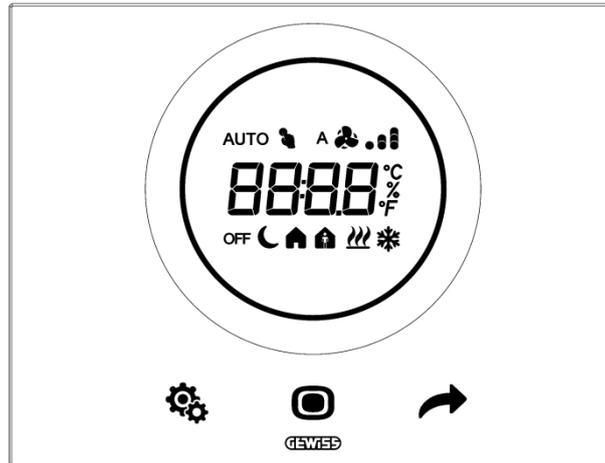
All the icons used for the thermostat display are listed below.



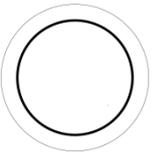
The following table summarises the meaning of the display icons:

Icon	Function	On fixed	Flashing
	Type of operation: heating	Type of operation, active heating	Floor temperature alarm in progress.
	Type of operation: valve	Type of operation, active air cooling	
	Fancoil management: active speed and automatic-manual fancoil mode	Fan always on. Speed indication (3 segments to the right that turn on up to the active speed); if no segment is active, then the fancoil is off.(A) on for the automatic fancoil speed operating mode	No/incorrect reception of fancoil speed feedback (fan). If the set speed (manually or via an algorithm) is waiting to be activated, the relative segment will flash.
	Measured temperature/Measured relative humidity/Indicates time/Indicates name of the parameter/Indicates value of the parameter/Cleaning function countdown	Temperature value/Measured relative humidity value/Time of day/Name of the parameter (P 1 ...P N)/Parameter value /Displays the inhibition time countdown for glass cleaning	Flashes during manual forcing of the setpoint. Flashes when humidity is displayed at the end of the external humidity sensor monitoring time
	Unit of measure of the temperature value/Relative humidity	Temperature in degrees Celsius (°C) or degrees Fahrenheit (°F)/Relative humidity (%)	
	Active HVAC mode (OFF = building protection)/building protection operation	Active HVAC mode /Setpoint building protection active	The icon of the selected mode flashes during the HVAC mode selection phase. OFF flashing for manual switch-off (slave function) or window open.
	Indication of manual setpoint forcing active	The segment is on if the setpoint is forced temporarily	

Also the control sliders and push-buttons are backlit and the colour changes depending on the context; if a specific function is not active, the colour of the push-buttons is TBD - idle, the same colour of the display segments.



The following table summarises the behaviour of the push-buttons back light depending on the active function:

Push-button	Function	Fixed ON	Flashing
	Activation of fancoil mode selection/Selected parameter change confirmation/Parameter change confirmation request	Normally on fixed, colour TBD, turns to TBD (specific function activation colour) indicating that the selection of fancoil mode is in progress	The icon flashes colour TBD (specific function activation colour) to signal the request to save the value (by pressing the MODE push-button) after changing the value of a fancoil parameter or mode
	No dedicated function	Normally on fixed, colour TBD	
	SET menu activation	The icon turns colour TBD (specific function activation colour) to indicate the activation phase of the SET menu.	
	Activation of the heating/air cooling system	The slider turns colour TBD (heat) to signal that the control algorithm is activated in the heating system; the slider also remains coloured if in that particular moment the heating valve is deactivated (e.g. PWM). The slider turns colour TBD (cool) to signal that the control algorithm is activated in the air cooling system; the slider also remains coloured if in that particular moment the air cooling valve is deactivated (e.g. PWM).	It flashes in the case of no/incorrect reception of feedback from the heating/air cooling system, depending on which of the two systems is active. This signal does not distinguish between no/incorrect first or second stage feedback

The structure of the ETS menu is as follows:

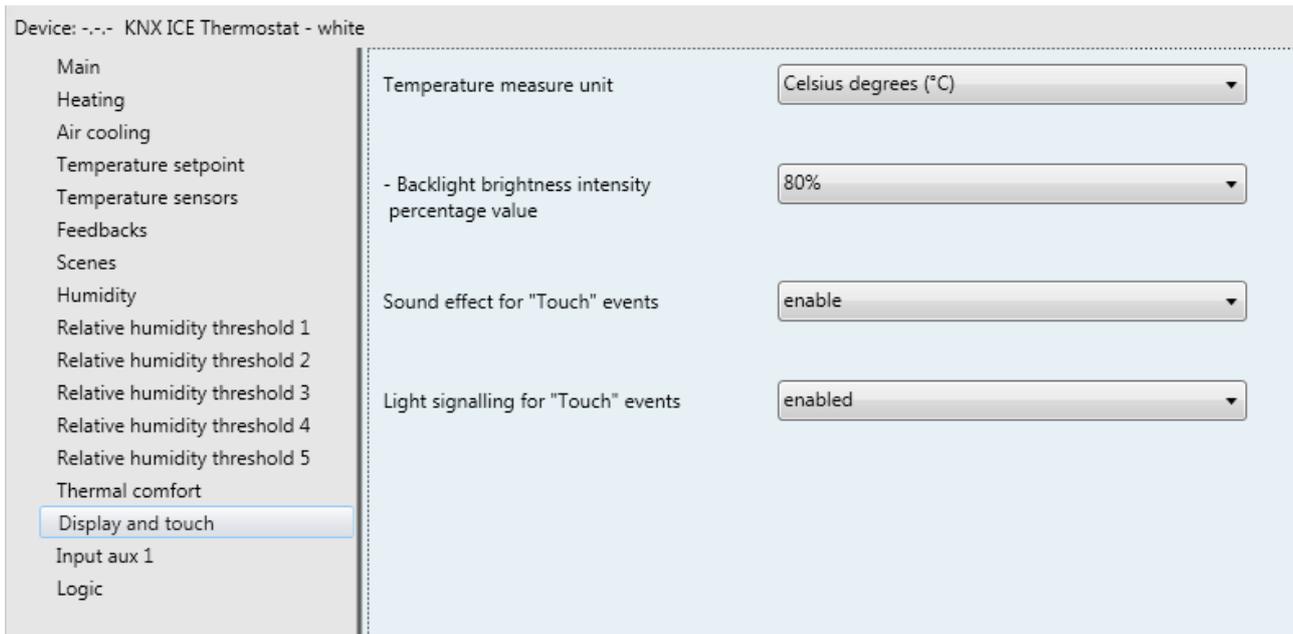


Fig. 13.1

## 13.1 Parameters

### ➤ 13.1.1 Temperature measure unit

This is used to define the temperature measure unit shown on the display. The values that can be set are:

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

### ➤ 13.1.2 Backlight brightness intensity percentage value

In stand-by, the thermostat only displays the measured temperature/current time/relative humidity (depending on which information was displayed before switching to stand-by, whereas the touch slider and push-button backlight is deactivated. The backlight turns on when the user approaches the glass surface, and then turns off automatically after a certain period of inactivity (energy saving); The device is equipped with a proximity sensor that is used to activate the backlight of the touch push-buttons when the user comes within a distance of approx. 10 cm from the glass plate.

The backlight turns off automatically after 30 seconds of user inactivity on the device (pressing the touch push-buttons/sliders); when the backlight turns off, only the display remains on with the display of the measured temperature, current time/relative humidity (depending on what information was displayed before switching to stand-by).

The brightness intensity of the backlight for the display and the touch push-buttons can be defined by the user via the parameter **“Backlight brightness intensity percentage value”** which makes it possible to select the percentage value of brightness intensity of the backlight for the display and touch push-buttons when on. The values that can be set are:

- from 30% to 100% in steps of 10% **(default value 80%)**

### ➤ 13.1.3 Sound effects for “Touch” events

The parameter “**Sound effects for “Touch” events**” is used to enable a sound effect for each touch event detected by the capacitive sensor. The values that can be set are:

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

by selecting **enable**, each time the touch push-buttons are touched, the TBD effect is reproduced, whereas if a long operation is detected (“SET” and “MODE” button keys) the TBD effect will be produced.

### ➤ 13.1.4 Light signalling for “Touch” events

It is possible to enable the reproduction of special light effects to signal different events detected by the touch push-buttons: *touch* and *long operation*. All makes it possible to personalize the backlight so that the various phases listed above can be easily identified directly by the user of the touch push-buttons. These effects do not concern the signalling colour change, which remains the same as before the touch detection. The parameter “**Light signalling for “Touch” events**” makes it possible to enable the light signals associated with the touch events. The values that can be set are:

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

by selecting **enabled**, each time the touch push-buttons are touched, the TBD effect is reproduced, whereas if a long operation is detected (“SET” and “MODE” button keys) the TBD effect will be produced

The sound and light signalling associated with the glass cleaning function input and output is always active regardless of the setting of the parameters “**Sound effects for “Touch” events**” and “**Light signalling for “Touch” events**”.

## 14 “Input aux 1” and “Input aux 2” menu

The device has two terminals for the connection of a potential-free contact that can be used as a general input (auxiliary input 1), and two terminals for the connection of a potential-free contact that can be used as a general input (auxiliary input 2) or as an auxiliary external sensor.

The **Input aux 1** menu is displayed if the “**Input contacts function**” parameter in the **Main** menu has the value **two auxiliary inputs independent** or **one auxiliary input and one temperature sensor**.

The **Input aux 2** menu is displayed if the “**Input contacts function**” parameter in the **Main** menu has the value **two auxiliary inputs independent**.

The **Input aux 1** menu contains the parameters used to set the operation of auxiliary input 1, and the **Input aux 2** menu contains the parameters used to set the operation of auxiliary input 2.

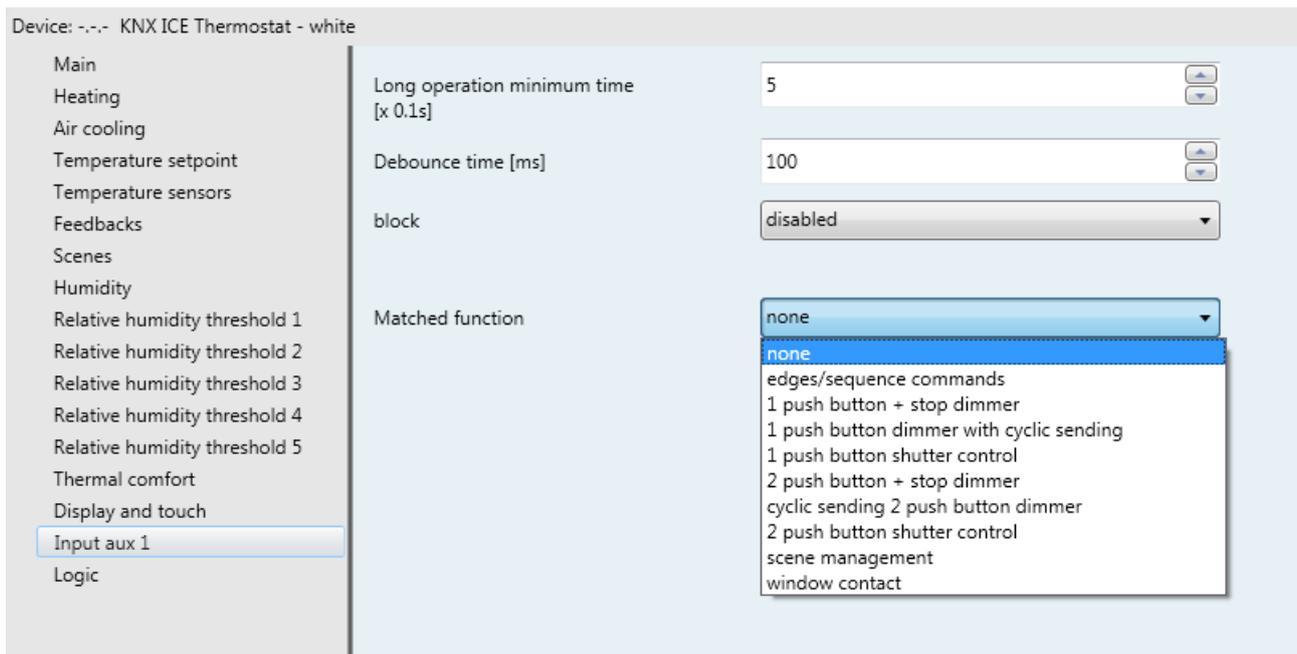


Fig. 14.1

The parameter used to define the function implemented by the auxiliary input 1 or 2 is “**Matched function**”. For the sake of simplicity, the parameters enabled according to the value set for that item are listed in the following paragraphs; the communication objects will refer to input X, where X assumes the values 1 and 2. The values that can be set are:

- **none** (default value)
- edges/sequence commands  
(See paragraph 14.1 “edges/sequence commands” function)
- 1 push-button + stop dimmer  
(See paragraph 14.2 “1 push-button + stop dimmer” function)
- cyclic sending 1 push-button dimmer  
(See paragraph 14.3 “Push-button dimmer with cyclic sending” function)
- 1 push-button shutter control  
(See paragraph 14.4 “1 push-button shutter control” function)
- 2 push-button + stop dimmer  
(See paragraph 14.5. “2 push-button + stop dimmer” function)
- cyclical sending 2 push-button dimmer  
(See paragraph 14.6. “Cyclical sending 2 push-button dimmer” function)
- 2 push-button roller shutters control  
(See paragraph 14.7. “2 push-button roller shutters control” function)
- scene management  
(see paragraph 14.8 “Scene management” function)
- window contact (only available for **Auxiliary input 1**).  
(see paragraph 14.9 “Window contact” function)

## 14.1 Parameters

### ➤ 14.1.1 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)**

### ➤ 14.1.2 Debounce time [ms]

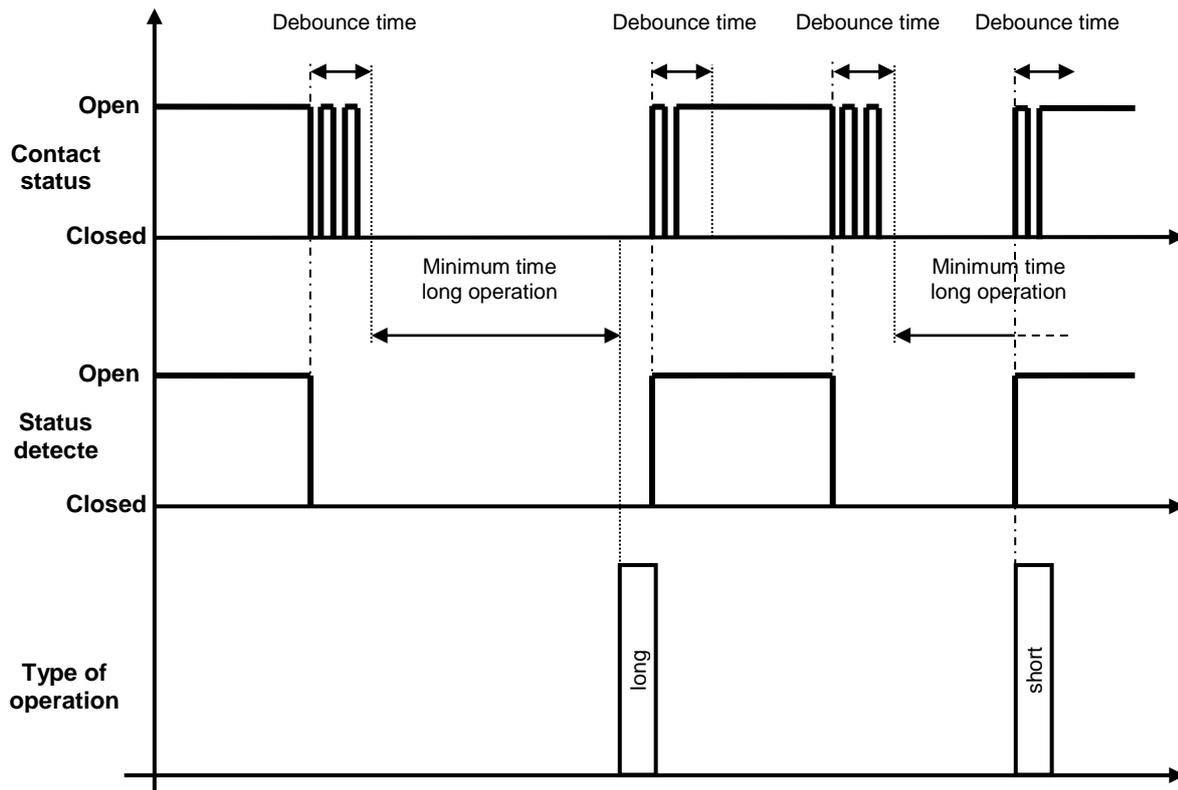
When an electro-mechanical device such as a push-button is pressed, there is a series of brief bounces (quick closing and opening of the contact) before the contact shifts definitively to the open or closed status; if suitable precautions are not taken, these bounces may be detected by the application 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  $T_{\text{debounce}}$  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.

### ➤ 14.1.3 Block

To inhibit the auxiliary 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                      0 (default value)
- enabled                        1

Setting **enabled** 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**” is used to set the logical value to be assumed by the bit received via BUS telegram to activate the block function. The values that can be set are:

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

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

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

## 14.2 “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:

Device: -.- KNX ICE Thermostat - white

Main  
Heating  
Air cooling  
Temperature setpoint  
Temperature sensors  
Feedbacks  
Scenes  
Humidity  
Relative humidity threshold 1  
Relative humidity threshold 2  
Relative humidity threshold 3  
Relative humidity threshold 4  
Relative humidity threshold 5  
Thermal comfort  
Display and touch  
**Input aux 1**  
Logic

Debounce time (ms): 200

block: disabled

Matched function: edges/sequence commands

Functioning type recognized: edges (opening/closing)

Sequence cyclic sending period if close contact/short operation [s]: 15

Sequence cyclic sending period if open contact/long operation [s]: 15

Channel behaviour at bus voltage recovery: ignore contact status and cyclical sending

Object A

-----

Sending object format: 1 bit

- Sending on closing/short operation detection: 1

- Sending on opening/long operation detection: no effect

Object sending delay [s]: 0

Cyclic sending object behaviour: on demand

Object B: disable

-----

Fig. 14.2

### ➤ 14.2.1 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

➤ **14.2.2 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:

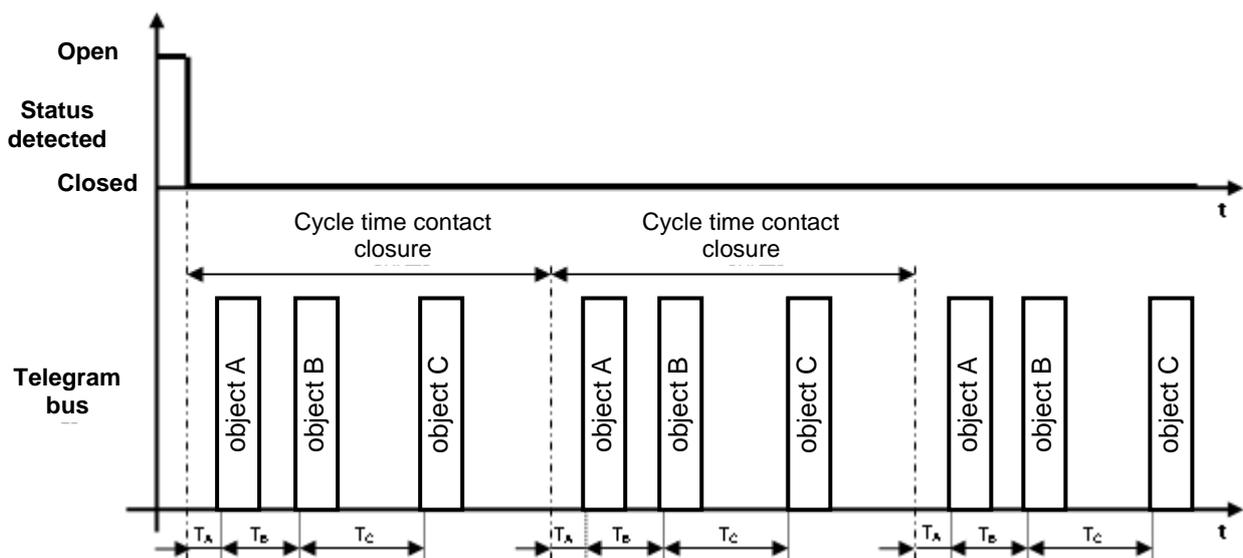
- from 1 to 65535 with steps of 1, **15 (default value)**

➤ **14.2.3 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.

➤ **14.2.4 Channel behaviour at BUS voltage recovery**

When the BUS or auxiliary voltage is restored, the behaviour of the auxiliary input x (with regard to the sending of the sequence and the cyclical sending of telegrams) can be defined via the “**Channel behaviour at bus voltage recovery**” parameter. The values that can be set are:

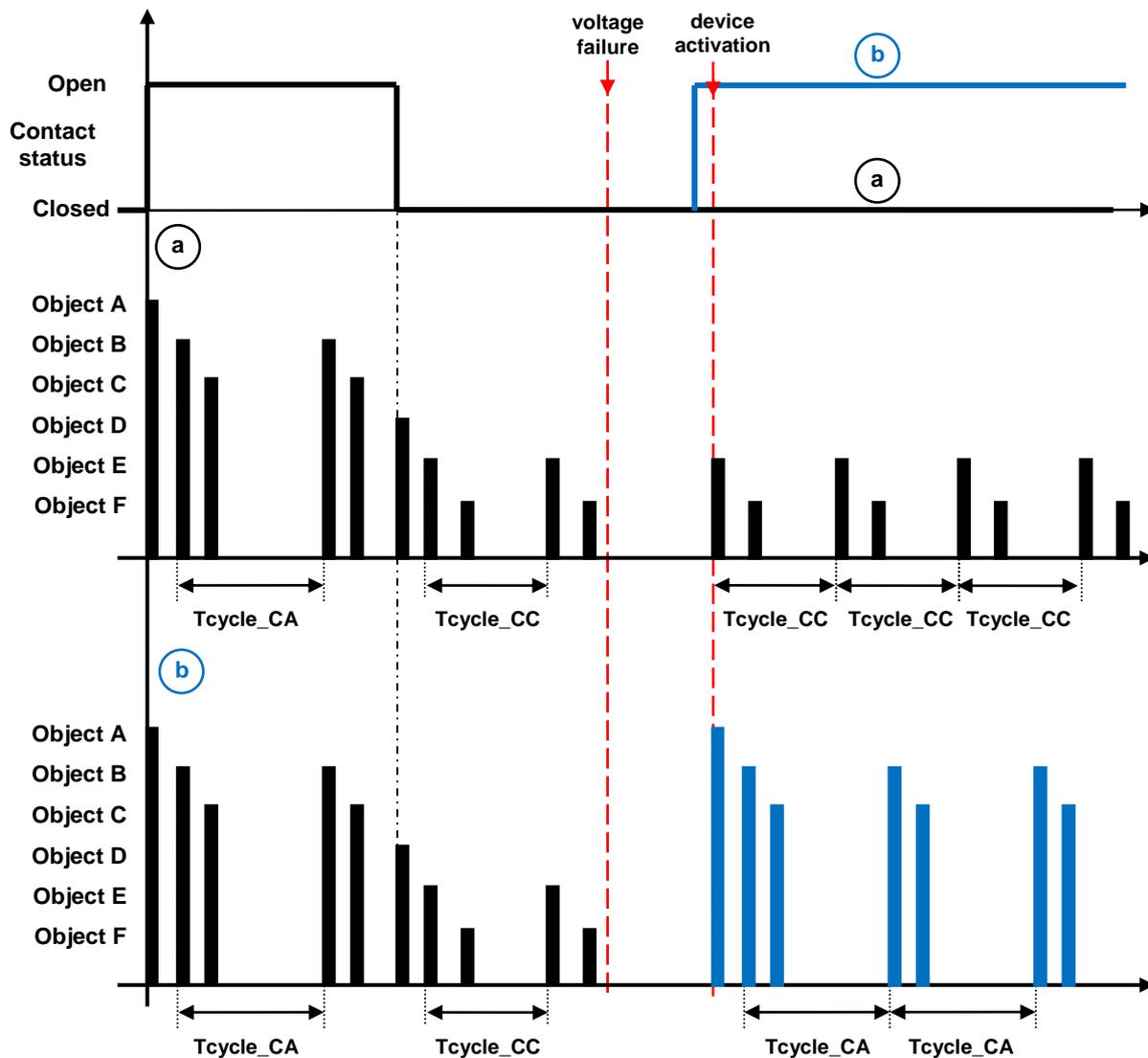
- **ignore contact status and cyclical sending (default value)**
- evaluate contact status and cyclical sending

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.

### ➤ 14.2.5 A/B/C/D object menu

For each auxiliary 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 “**Object z**” parameter (where z is the index of the object associated with the threshold, between **A** and **D**) can be used to enable a new object to be sent. The parameter may assume the following values:

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

If **enable** is selected, the 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 auxiliary input, included between **A** and **D**).

The “**Sending object format**” parameter is used to set the format and code of the BUS telegram that will be sent by the device following a variation in the status of the auxiliary input contact.

The values that can be set are:

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

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

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 auxiliary 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 auxiliary 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 or auxiliary voltage is restored, 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** (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 (-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**" 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

The parameter “**Object sending delay [s]**” is used to set the delay between the detection of the event associated with the sending of the command, and the actual sending of the command/value on the BUS. With regard to the objects that range from index B to index D, this parameter indicates the delay between sending the command/value associated with the object with the previous index (z-1) and sending the command/value associated with the object to which the parameter refers; the delay 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.

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.

### **14.3 Matched function: “1 push-button + stop dimmer”**

This is used to configure the auxiliary 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:

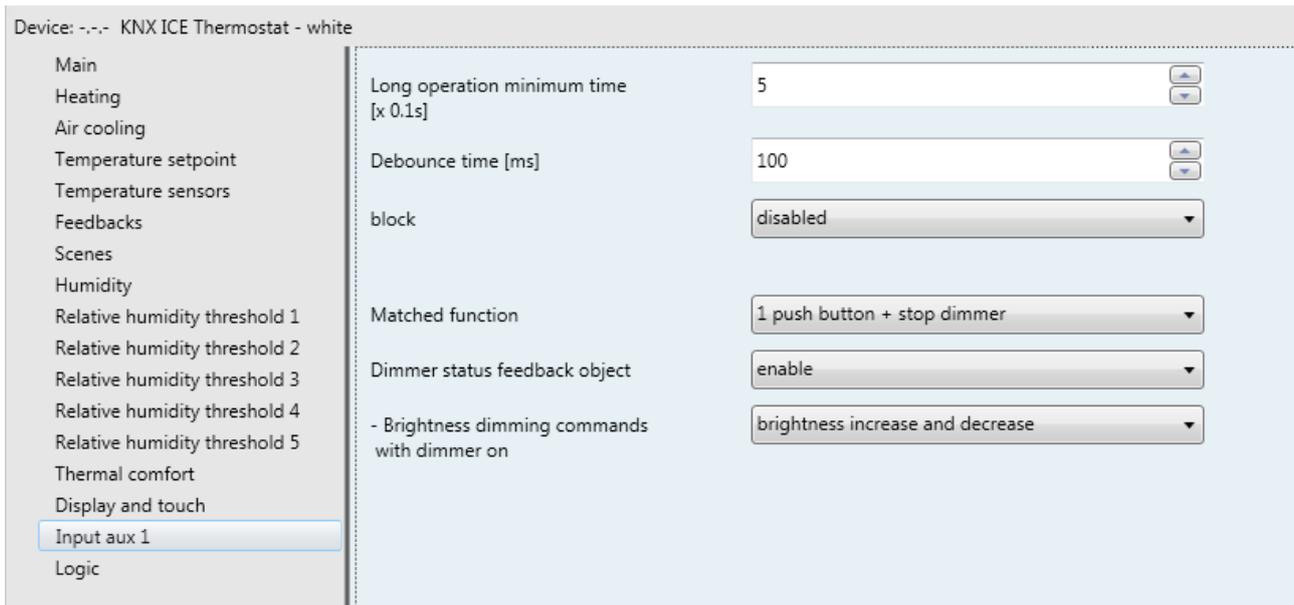


Fig. 14.3

The normal behaviour 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**”.

### ➤ 14.3.1 Parameters

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

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

If **enable** 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 push-button panel about the status of the devices connected.

## 14.4 Matched function: “Cyclic sending 1 push-button dimmer”

This is used to configure the auxiliary input to control a dimmer with a single push-button, increasing and decreasing dimmer brightness always using the same 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:

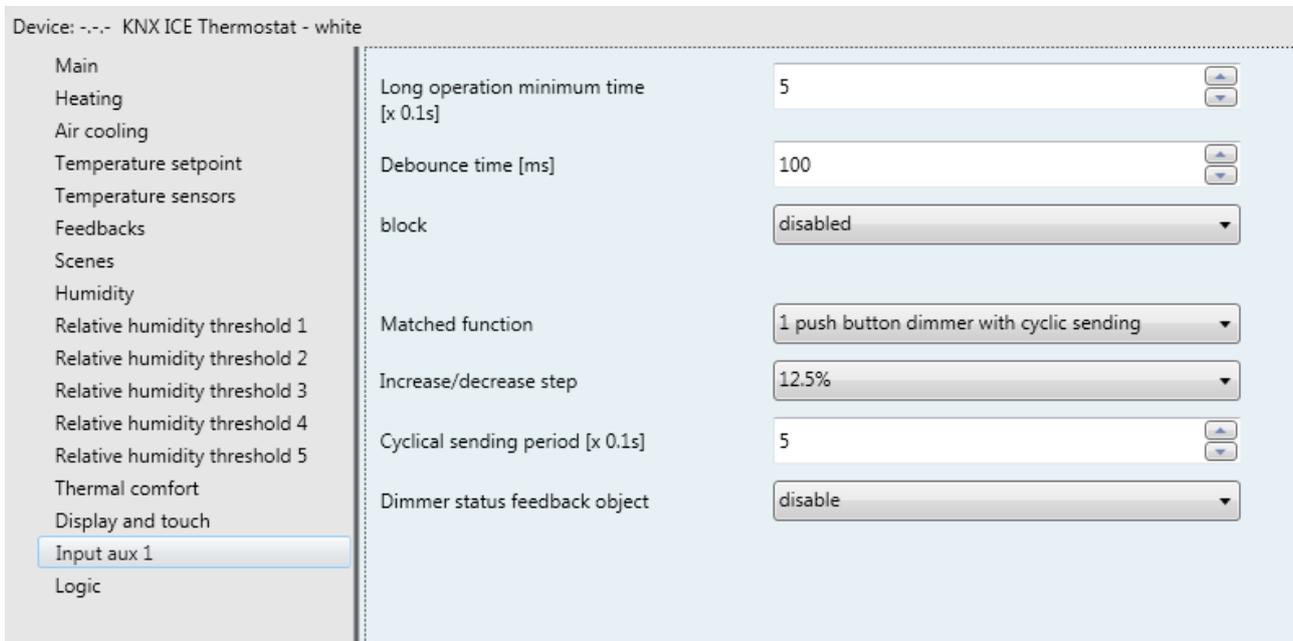


Fig. 14.4

### ➤ 14.4.1 Parameters

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

- 100%
- 50%

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

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

The values that can be set for the "**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 operation is maintained, 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

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 14.3 "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 "**Brightness control commands with dimmer on**" parameter 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 push-button panel about the status of the devices connected.

## **14.5 Matched function: "1 push-button roller shutter control"**

This is used to configure the auxiliary 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:

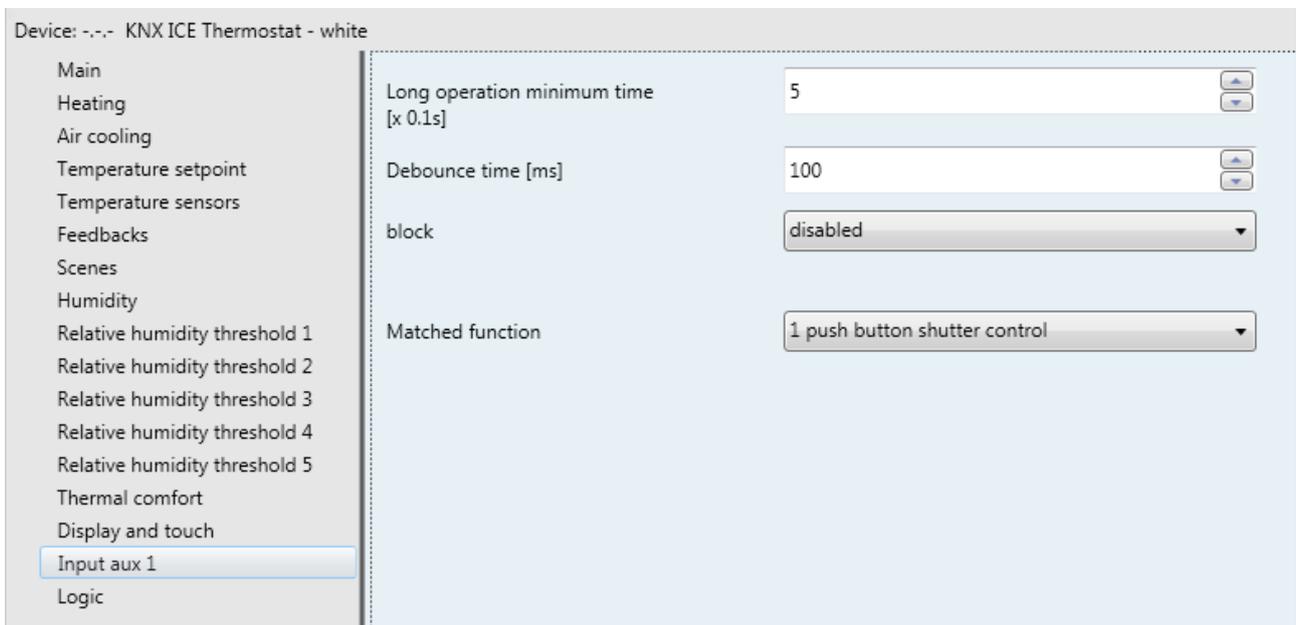


Fig. 14.5

No new parameters are enabled with this function.

## 14.6 Matched function: “2 push-button + stop dimmer”

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:

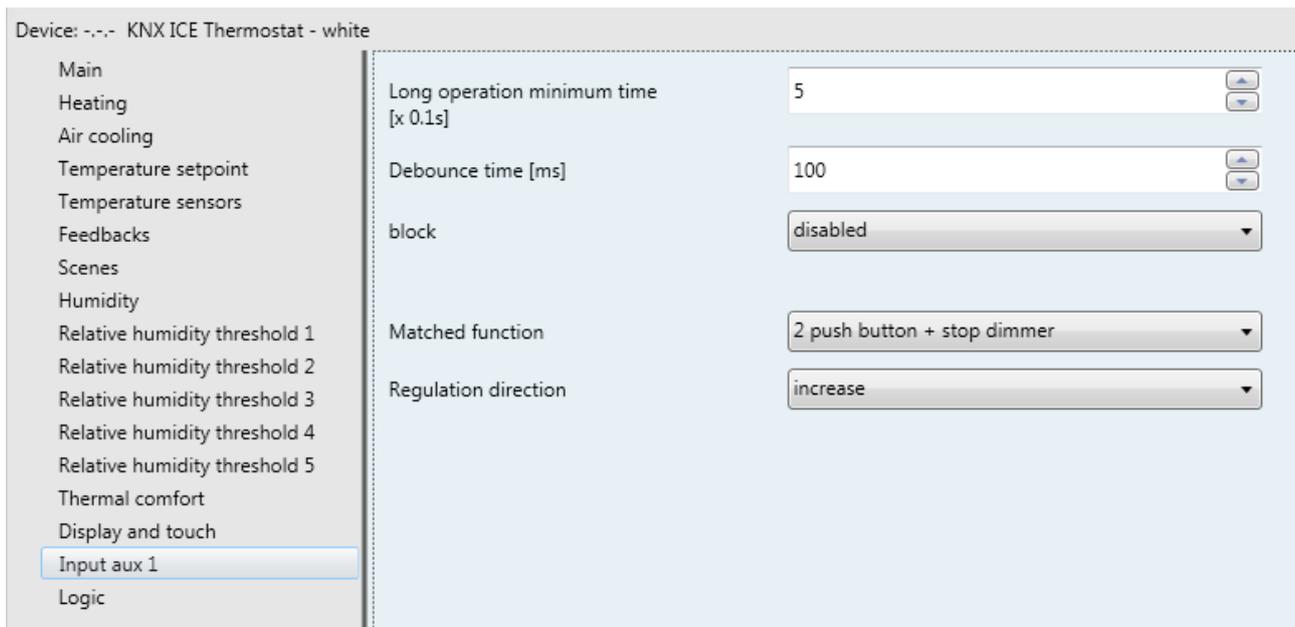


Fig. 14.6

### ➤ 14.6.1 Parameters

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.

## 14.7 Matched function: “cyclical sending 2 push-button dimmer”

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:

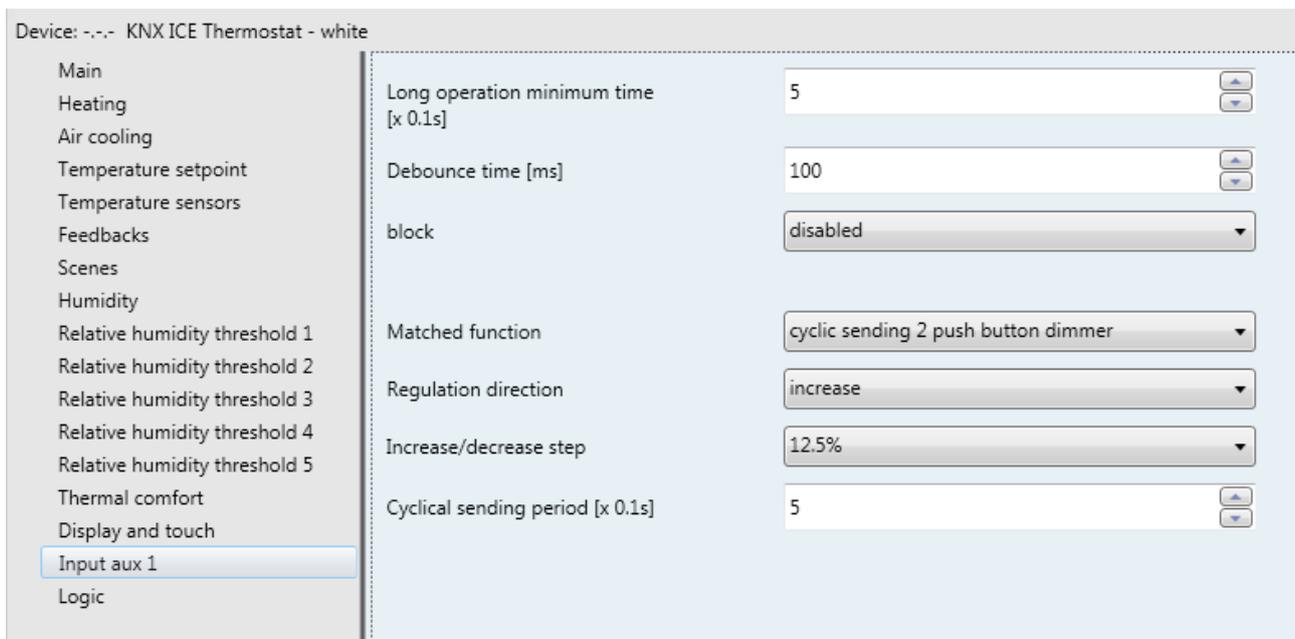


Fig. 14.7

### ➤ 14.7.1 Parameters

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.

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

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

If 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

## **14.8 Matched function: "2 push-button roller shutters control"**

This is used to configure the auxiliary 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:

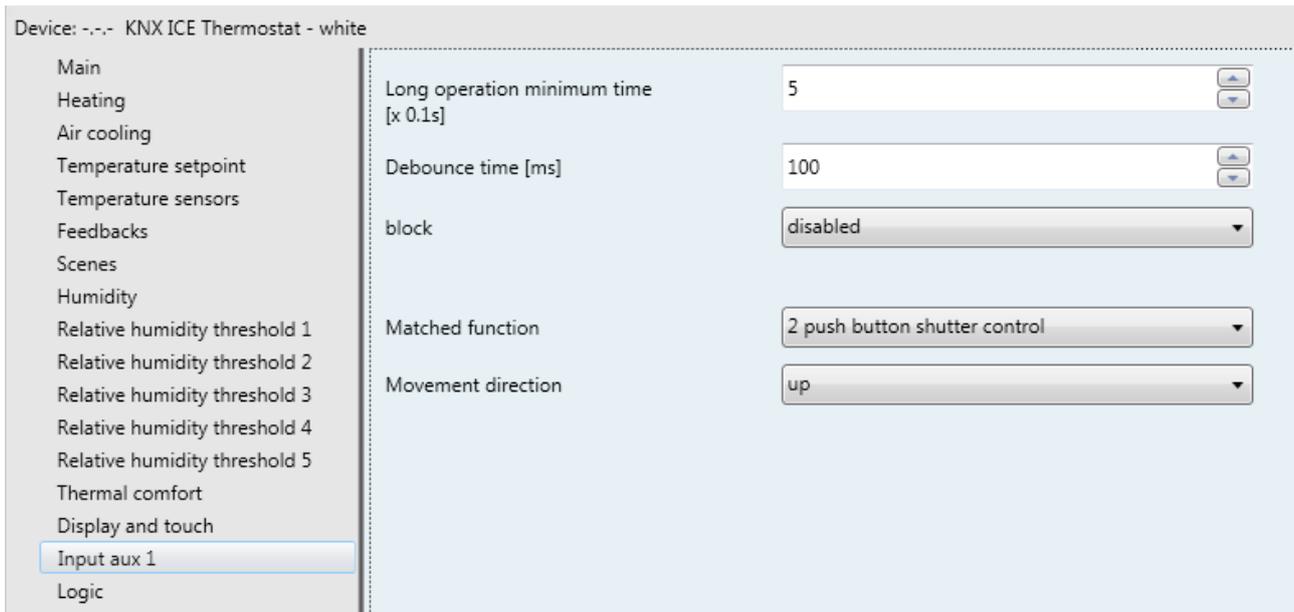


Fig. 14.8

### ➤ 14.8.1 Parameters

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).

## 14.9 Matched function: “scene management”

This is used to configure the auxiliary 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:

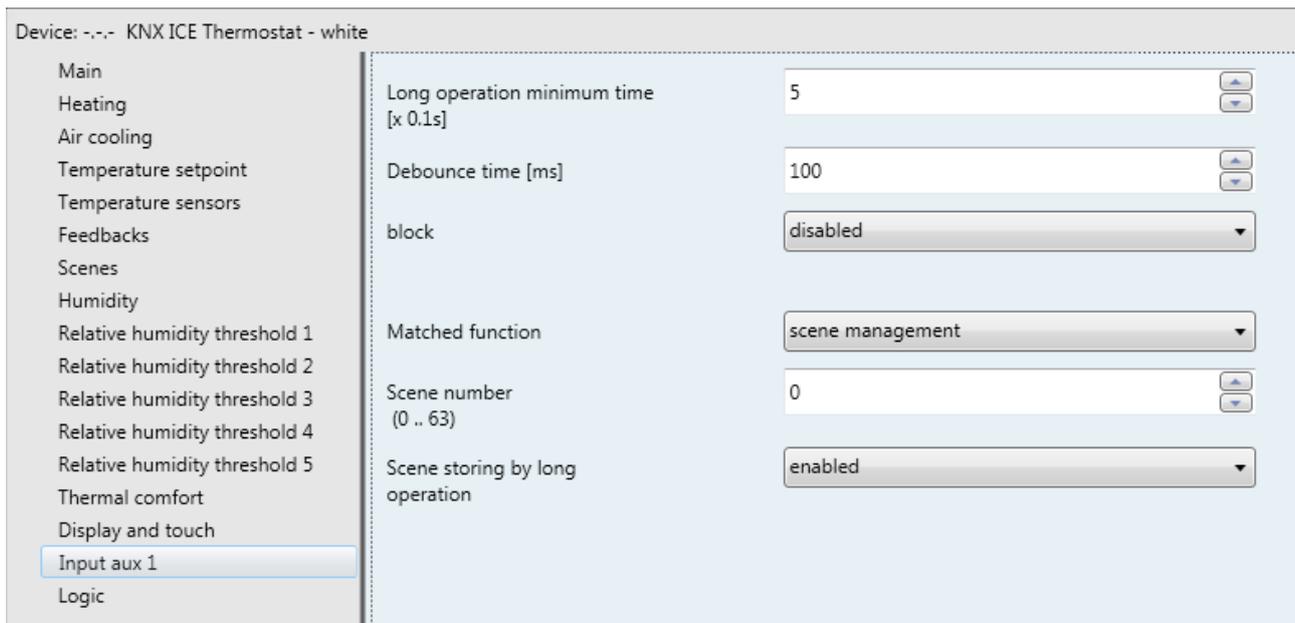


Fig. 14.9

### ➤ 14.9.1 Parameters

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

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.

## 14.10 Matched function: “window contact”

Used to configure auxiliary input 1 to carry out the “window contact” function of the device.

The structure of the menu is as follows:

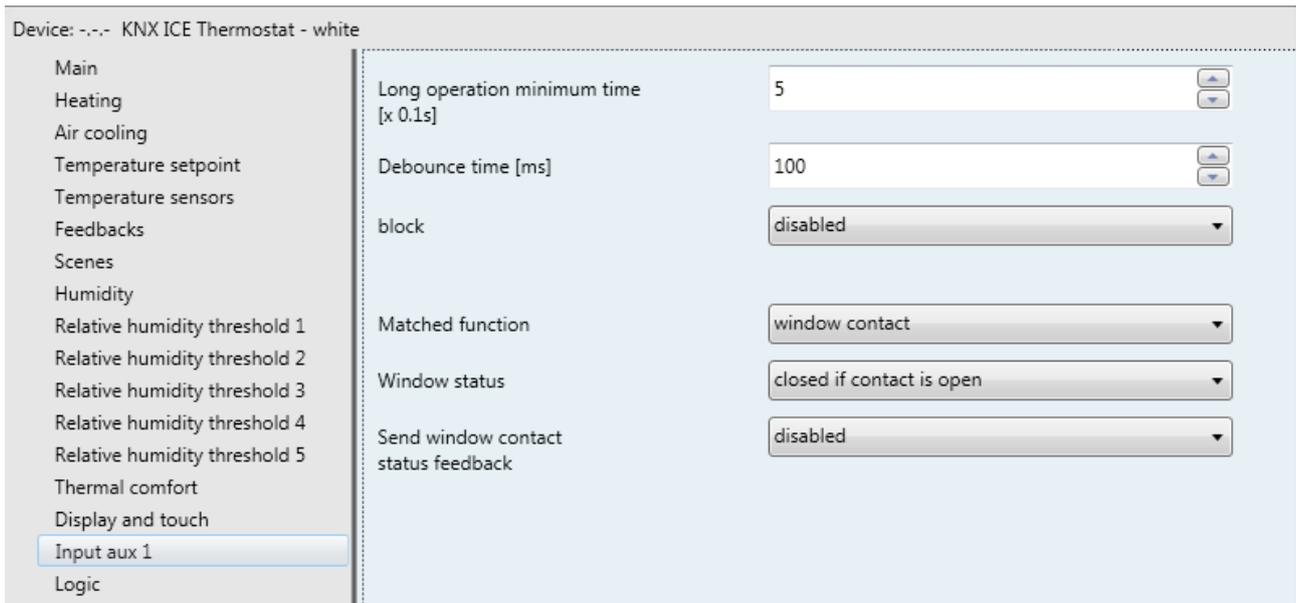


Fig. 14.10

This particular function allows you to force the device to OFF mode (if the control type is remote) or set the BUILDING PROTECTION setpoint (if the remote control type is setpoint) when the window opens, and reactivate normal operation when the window closes again. The priority of this command is higher than all remote commands (even if the device is a Slave), including 1-bit modes.

### ➤ 14.10.1 Parameters

The “**Window status**” parameter is used to associate the window status (closed/open) with the auxiliary contact status (closed/open). The values that can be set are:

- closed if contact is open
- **closed if contact is closed** (default value)

The status of the contact that identifies the “open” window is the opposite of the status that identifies the “closed” window.

The status of the contact can be sent on the BUS via the **Aux input 1 - Window status feedback** communication object; The parameter used to enable transmitting this information is “**Send window contact status feedback**”, which can have the following values:

- disabled
- on demand only
- **on change** (default value)

Selecting any value other than **disabled** displays the **Aux input 1 - Window status feedback** communication object (Data Point Type: 1.019 DPT\_Window\_Door), which is used to transmit via BUS status information concerning the status of the contact connected to the device.

If the status feedback takes place **on variation** the communication object is sent spontaneously when the status switches from CLOSED to OPEN or vice versa. If the set value is **on demand only**, the status will never be sent spontaneously by the device. Only when a status reading request is received from the BUS, the device sends a response telegram with the current load status.

The communication object has the value “1” when the window is OPEN (the effective status of the contact depends on the value set for the “**Window status**” parameter) and the value “0” when the window is CLOSED.

## 15 “Logic” menu

The **Logic** menu contains the parameters used to configure the logic operation to be executed, also defining the involved objects. This function can be very useful, for example, for activating the pump/boiler when at least one area valve is open.

The basic structure of the menu is as follows:

Device: --- KNX ICE Thermostat - white

Main	Logic function	enable
Heating	Logic input 1	enabled
Air cooling	- NOT operation for logic input 1	disable
Temperature setpoint	Logic input 2	enabled
Temperature sensors	- NOT operation for logic input 2	disable
Feedbacks	Logic input 3	disabled
Scenes	Operation between logic inputs	AND
Humidity	The logic operation is computed	whenever an input value is received
Relative humidity threshold 1	Generate the result of the logic operation	whenever the logic is evaluated
Relative humidity threshold 2	If the result of the logic operation is TRUE	sends "1"
Relative humidity threshold 3	If the result of the logic operation is FALSE	sends "1"
Relative humidity threshold 4		
Relative humidity threshold 5		
Thermal comfort		
Display and touch		
Input aux 1		
Logic		

Fig. 15.1

### ➤ 15.1.1 Parameters

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

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

Selecting **enabled** displays the function configuration parameters.

The logic operation can have a maximum of 8 inputs; the parameters “**Logic input 1**”, ... “**Logic input 8**” can be used to enable each input to be used in the logic operation. The values that can be set for the parameters listed above are:

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

selecting **enabled**, displays the communication object **Logic input 1 ... Logic input 8** (Data Point Type:1.002 DPT\_Bool) via which the device receives the input values of the logic function and the parameter **“NOT operation for logic input 1” ... “NOT operation for logic input 8”** depending on which logic input is enabled.

It is possible to refuse the value from the logic inputs via the parameters **“NOT operation for logic input 1” ... “NOT operation for logic input 8”** (they are displayed depending on the number of enabled logic inputs), which can have the following values:

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

The logic operation to be executed between the logic inputs is defined via the parameter **“Operation between logic inputs”** which can have the following values:

- **AND** (default value)
- OR
- NAND
- NOR
- XOR
- XNOR

It is possible to determine the condition that generates the calculation of the logic operation selected via the parameter **“The logic operation is computed”**; The values that can be set are:

- **each time an input value is received** (default value)  
the logic is calculated each time an input value is received, independently of the fact if the new value is equal to or different than the previous value
- if at least one input value changes status  
the logic is calculated each time an input value is received only if the new value is different than the previous value
- periodically  
the logic is calculated at fixed time intervals, independently of the fact if new input values were received

selecting **periodically**, to display the parameter **“Calculation period”** which makes it possible to set the time interval with which the logic is calculated. The values that can be set are:

- from **1 second (default value)** to 255 seconds with step of 1

The result of the logic operation is transmitted on the KNX BUS via the communication object **Logic operation result**(Data Point Type:1.002 DPT\_Bool) ; The conditions for sending the telegram associated with the result of the logic operation is defined via the parameter **“Generate the result of the logic operation”**, which can have the following values:

- **each time the logic is evaluated** (default value)  
the telegrams are sent each time the logic is calculated, independently of the fact that the result of the new calculation is equal to or different than the previous result
- only if the outcome changes  
the telegrams are only sent if the result of the new calculation is different than the previous result

The result of the logic operation can be “true” or “false”; for each of these values, it is possible to define if a feedback telegram should be sent or not via the communication object **Logic operation result**. The parameter **“If the result of the logic operation is TRUE”** is used to activate the sending of the result feedback “true” whereas the parameter **“If the result of the logic operation is FALSE”** is used to activate the sending of the result feedback “false”; The possible values for these parameters are:

- do not send BUS commands
- **send “0”** (default value for a FALSE result)
- **send “1”** (default value for a TRUE result)

When the BUS or auxiliary voltage is restored, the device sends the read requests on the objects **Logic input 1 ... Logic input 8** to be updated about the current values of the inputs; as long as all the statuses of the inputs are not known, the device will not send any telegram via the object **Logic operation result** and will try to send the logic input read request every minute until a value is received.

## 16 ETS programming error feedback

The device is able to detect and therefore indicate various programming errors on the display using the three central digits:

Detected error	Information on the display
<p>The constraints between the setpoints of the various HVAC modes belonging to the same functioning type are not respected:</p> <ul style="list-style-type: none"> <li>- <math>T_{\text{anti-freeze}} \leq T_{\text{economy}} \leq T_{\text{precomfort}} \leq T_{\text{comfort}}</math> in heating mode</li> <li>- <math>T_{\text{comfort}} \leq T_{\text{precomfort}} \leq T_{\text{economy}} \leq T_{\text{high temp. protection}}</math> in air cooling mode</li> </ul> <p>or if the control type is setpoint, the constraints are</p> <ul style="list-style-type: none"> <li>- <math>T_{\text{antifreeze}} \leq T_{\text{operation}}</math> in heating mode</li> <li>- <math>T_{\text{operation}} \leq T_{\text{high temp. protection}}</math> in air cooling mode</li> </ul>	ER1
<p>The HVAC mode setpoints are out of the maximum range</p> <ul style="list-style-type: none"> <li>- The comfort/economy/precomfort setpoints are not between 5 °C and 40 °C</li> <li>- The antifreeze setpoint is not between 2 °C and 7 °C</li> <li>- The high temperature protection setpoint is not between 30 °C and 40 °C</li> </ul> <p>or if the control type is setpoint</p> <ul style="list-style-type: none"> <li>- The operating setpoint is not between 5 °C and 40 °C</li> <li>- The antifreeze setpoint is not between 2 °C and 7 °C</li> <li>- The high temperature protection setpoint is not between 30 °C and 40 °C</li> </ul>	ER2
<p>For every HVAC mode, the difference between the heating and air cooling setpoints is lower than 1 °C (only if the dead zone is enabled via ETS).</p>	ER3
<p>Connection error of the communication objects dedicated to sending commands to the actuation devices (valves and fancoils):</p> <ul style="list-style-type: none"> <li>- if different control logic is enabled but command or feedback objects for the heating and air cooling solenoid valves are connected in the same group address</li> <li>- 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 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.</li> <li>- if the fancoil algorithm is set and all the command objects for the fancoil speed are not connected</li> <li>- if the second stage is active and the command objects for the relative solenoid valves are not connected.</li> <li>- if the second stage is active for both functioning types and the command objects are connected to the same address (in the second stage the control logic is always different).</li> </ul>	ER4
<p>The total of the impact of the external temperature probes (KNX and auxiliary) exceeds 100%</p>	ER5
<p>Missing the links of the communication objects dedicated to receive the notification feedbacks (valves and fan coil speeds) from actuator devices:</p> <ul style="list-style-type: none"> <li>- if the valves status feedbacks are enabled but not connected</li> <li>- if the fan coil speed status feedbacks are enabled but not connected</li> <li>- if the valves status feedbacks of the second stage are enabled but not connected</li> </ul>	ER6

If multiple errors are detected, the error with the lowest index will be shown on the display.

Once the error is detected and shown on the display, the feedback will remain visible until the ETS application with the necessary corrections has been downloaded.

# 17 Communication objects

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

## Communication objects with output functions

#	Object name	Object function	Description	DPT type
61	Day/Hour sending	Send day/hour	Sends updates about week day and day time	[10.1] DPT_TimeOfDay
81	Heating valve switch	On/Off	Sends heating electrovalve activation/deactivation commands 1st stage	[1.1] DPT_Switch
81	Heating/Air cooling valve switch	On/Off	Sends heating/air cooling electrovalve activation/deactivation commands	[1.1] DPT_Switch
81	% command valve heating	% value	Sends percentage commands to electrovalve heating 1st stage	[5.1] DPT_Scaling
81	Heating/Air cooling valve % command	% value	Sends percentage commands to electrovalve heating/air cooling	5.001 DPT_Scaling
83	V1 fan switching heating	On/Off	Sends heating fancoil speed 1 electrovalve activation/deactivation commands	1.001 DPT_Switch
84	V2 fan switching heating	On/Off	Sends commands of activation/deactivation fancoil speed 2 heating	1.001 DPT_Switch
85	V3 fan switching heating	On/Off	Sends commands of activation/deactivation fancoil speed 3 heating	1.001 DPT_Switch
89	Heating 2nd stage switching	On/Off	Sends the 2nd stage heating solenoid valve on/off commands	1.001 DPT_Switch
89	Heating 2° stage switching	On/Off	Sends heating electrovalve activation/deactivation commands 2nd stage	5.001 DPT_Scaling
91	Air cooling valve switch	On/Off	Sends air cooling electrovalve activation/deactivation commands 1st stage	1.001 DPT_Switch
91	Air cooling valve % command	% value	Sends percentage commands to electrovalve air cooling 1st stage	5.001 DPT_Scaling
93	V1 fan switching air cooling	On/Off	Sends air cooling fancoil speed 1 electrovalve activation/deactivation commands	1.001 DPT_Switch
94	V2 fan switching air cooling	On/Off	Sends commands of activation/deactivation fancoil speed 2 air cooling	1.001 DPT_Switch
95	V3 fan switching air cooling	On/Off	Sends commands of activation/deactivation fancoil speed 3 air cooling	1.001 DPT_Switch
99	Air cooling 2° stage switching	On/Off	Sends air cooling electrovalve activation/deactivation commands 2nd stage	1.001 DPT_Switch
99	% command 2° stage air cooling	% value	Sends percentage commands to electrovalve air cooling 2nd stage	5.001 DPT_Scaling
102	Fan coil mode report	Automatic/manual	Sends feedback about fancoil speed automatic/manual mode	1.001 DPT_Switch
104	Auxiliary sensor measured temperature	Value °C	Sends temperature values in degrees Celsius measured by NTC sensor connected to input	9.001 DPT_Temp

104	Auxiliary sensor measured temperature	°K value	Sends temperature values in degrees Kelvin measured by NTC sensor connected to input	9.002 DPT_Tempd
104	Auxiliary sensor measured temperature	°F value	Sends temperature values in degrees Fahrenheit measured by NTC sensor connected to input	9.027 DPT_Temp_F
105	Measured temperature	Value °C	Sends temperature values in degrees Celsius calculated by device	9.001 DPT_Temp
105	Measured temperature	°K value	Sends temperature values in degrees Kelvin calculated by device	9.002 DPT_Tempd
105	Measured temperature	°F value	Sends temperature values in degrees Fahrenheit calculated by device	9.027 DPT_Temp_F
106	HVAC mode feedback	Eco/Precom/Comf/Off	Sends feedback about HVAC mode set	20.102 DPT_HVACMode
106	Functioning setpoint feedback	Value °C	Sends functioning setpoint values in degrees Celsius	9.001 DPT_Temp
106	Functioning setpoint feedback	°K value	Sends functioning setpoint values in degrees Kelvin	9.002 DPT_Tempd
106	Functioning setpoint feedback	°F value	Sends functioning setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F
107	HVAC off mode feedback	Enable/Disable	Sends feedback about HVAC off (building protection) mode activation status	1.003 DPT_Enable
108	HVAC economy mode feedback	Enable/Disable	Sends feedback about HVAC economy mode activation status	1.003 DPT_Enable
109	HVAC precomfort mode feedback	Enable/Disable	Sends feedback about HVAC precomfort mode activation status	1.003 DPT_Enable
110	HVAC comfort mode feedback	Enable/Disable	Sends feedback about HVAC comfort mode activation status	1.003 DPT_Enable
111	Functioning type feedback	Heating/Air cooling	Sends feedback about functioning type set	1.100 DPT_Heat/Cool
112	Heating anti-freeze setpoint feedback	Value °C	Sends setpoint values heating off HVAC mode in degrees Celsius	9.001 DPT_Temp
112	Heating anti-freeze setpoint feedback	°K value	Sends setpoint values heating off HVAC mode in degrees Kelvin	9.002 DPT_Tempd
112	Heating anti-freeze setpoint feedback	°F value	Sends setpoint values heating off HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
113	Heating economy setpoint feedback	Value °C	Sends setpoint values heating economy HVAC mode in degrees Celsius	9.001 DPT_Temp
113	Heating economy setpoint feedback	°K value	Sends setpoint values heating economy HVAC mode in degrees Kelvin	9.002 DPT_Tempd
113	Heating economy setpoint feedback	°F value	Sends setpoint values heating economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
114	Heating precomfort setpoint feedback	Value °C	Sends setpoint values heating precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp
114	Heating precomfort setpoint feedback	°K value	Sends setpoint values heating precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
114	Heating precomfort setpoint feedback	°F value	Sends setpoint values heating precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
115	Heating comfort setpoint feedback	Value °C	Sends setpoint values heating comfort HVAC mode in degrees Celsius	9.001 DPT_Temp
115	Heating comfort setpoint feedback	°K value	Sends setpoint values heating comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
115	Heating comfort setpoint feedback	°F value	Sends setpoint values heating	9.027 DPT_Temp_F

						comfort HVAC mode in degrees Fahrenheit		
116	Air cond. high temp. protection setpoint feedback				Value °C	Sends setpoint values air cooling off HVAC mode in degrees Celsius	9.001 DPT_Temp	
116	Air cond. high temp. protection setpoint feedback				°K value	Sends setpoint values air cooling off HVAC mode in degrees Kelvin	9.002 DPT_Tempd	
116	Air cond. high temp. protection setpoint feedback				°F value	Sends setpoint values air cooling off HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F	
117	Air cooling economy setpoint feedback				Value °C	Sends setpoint values air cooling economy HVAC mode in degrees Celsius	9.001 DPT_Temp	
117	Air cooling economy setpoint feedback				°K value	Sends setpoint values air cooling economy HVAC mode in degrees Kelvin	9.002 DPT_Tempd	
117	Air cooling economy setpoint feedback				°F value	Sends setpoint values air cooling economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F	
118	Air cooling precomfort setpoint feedback				Value °C	Sends setpoint values air cooling precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp	
118	Air cooling precomfort setpoint feedback				°K value	Sends setpoint values air cooling precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd	
118	Air cooling precomfort setpoint feedback				°F value	Sends setpoint values air cooling precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F	
119	Air cooling comfort setpoint feedback				Value °C	Sends setpoint values air cooling comfort HVAC mode in degrees Celsius	9.001 DPT_Temp	
119	Air cooling comfort setpoint feedback				°K value	Sends setpoint values air cooling comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd	
119	Air cooling comfort setpoint feedback				°F value	Sends setpoint values air cooling comfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F	
120	Current setpoint report				Value °C	Sends active setpoint values in degrees Celsius	9.001 DPT_Temp	
120	Current setpoint report				°K value	Sends active setpoint values in degrees Kelvin	9.002 DPT_Tempd	
120	Current setpoint report				°F value	Sends active setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F	
121	Thermostat functioning feedback				Slave/Stand-alone	Sends Slave or stand-alone thermostat functioning feedback	1.001 DPT_Switch	
125	Measured relative humidity				Value [% RH]	Sends the relative humidity values calculated by the device	9.007 DPT_Value_Humidity	
126	Specific humidity				Value [g/kg]	Sends specific humidity values in g/kg	14.005 DPT_Value_Amplitude	
128	140	152	164	176	Relative humidity threshold 1 enabling status	Enabled/Disabled	Sends relative humidity threshold 1 enabling status	1.003 DPT_Enable
130	142	154	166	178	Relative humidity threshold 1 feedback	Value [% RH]	Sends the relative humidity threshold 1 current value	9.007 DPT_Value_Humidity
131	143	155	167	179	Relative humidity threshold 1 output A	1/0 value	Sends the values 1/0 associated to the output A of relative humidity threshold 1	1.001 DPT_Switch
131	143	155	167	179	Relative humidity threshold 1 output A	On/Off forced positioning	Sends the 2 bit values associated to the output A of relative humidity threshold 1	2.001 DPT_Switch_Control
131	143	155	167	179	Relative humidity threshold 1 output A	Value [0 .. 255]	Sends the unsigned values (0..255) associated to the output A of relative humidity threshold 1	5.010 DPT_Value_1_Ucount
131	143	155	167	179	Relative humidity threshold 1 output A	Value [-128 .. 127]	Sends the signed values (-128..127) associated to the	6.010 DPT_Value_1_Count

							output A of relative humidity threshold 1	
131	143	155	167	179	Relative humidity threshold 1 output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated to the output A of relative humidity threshold 1	5.001 DPT_Percentage
131	143	155	167	179	Relative humidity 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 relative humidity threshold 1	20.102 DPT_HVAC_Mode
131	143	155	167	179	Relative humidity threshold 1 output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated to the output A of relative humidity threshold 1	7.001 DPT_Value_2_Ucount
131	143	155	167	179	Relative humidity threshold 1 output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated to the output A of relative humidity threshold 1	8.001 DPT_Value_2_Count
131	143	155	167	179	Relative humidity threshold 1 output A	Setpoint value [°C]	Sends the setpoint values (°C) associated to the output A of relative humidity threshold 1	9.001 DPT_Value_Temp
131	143	155	167	179	Relative humidity threshold 1 output A	Setpoint value [°K]	Sends the setpoint values (°K) associated to the output A of relative humidity threshold 1	9.002 DPT_Value_Tempd
131	143	155	167	179	Relative humidity threshold 1 output A	Setpoint value [°F]	Sends the setpoint values (°F) associated to the output A of relative humidity threshold 1	9.027 DPT_Value_Temp_F
133	145	157	169	181	Relative humidity threshold 1 output B	1/0 value	Sends the values 1/0 associated to the output B of relative humidity threshold 1	1.001 DPT_Switch
135	147	159	171	183	Relative humidity threshold 1 output C	1/0 value	Sends the values 1/0 associated to the output C of relative humidity threshold 1	1.001 DPT_Switch
137	149	161	173	185	Relative humidity threshold 1 output D	1/0 value	Sends the values 1/0 associated to the output D of relative humidity threshold 1	1.001 DPT_Switch
188	Thermal comfort status feedback					1=Comfortable/0=Uncomfortable	Sends the thermal comfort status feedback	1.002 DPT_Bool
190	215	IN.x - Switch				On/Off	Sends dimmer on/off switching commands	1.001 DPT_Switch
190	215	IN.x - Shutter movement				Up/Down	Sends up/down shutter movement commands	1.008 DPT_UpDown
190	215	IN.x - Scene				Execute/Store	Sends scene storing/execution commands	18.001 DPT_SceneControl
190	215	IN.x - A object 1 bit value				1/0 value	Send 1/0 values associated to object A	1.002 DPT_Bool
190	215	IN.x - A object 2 bits value				On/Off forced positioning	Send 1/0 values associated to object A	1.002 DPT_Switch_Control
190	215	IN.x - A object 1 byte value				Unsigned value	Send unsigned values (0..255) associated to object A	5.010 DPT_Value_1_Ucount
190	215	IN.x - A object 1 byte value				Signed value	Send signed values (-128..127) associated to object A	6.010 DPT_Value_1_Count
190	215	IN.x - A object 1 byte value				% value	Send percent values (0%..100%) associated to object A	5.001 DPT_Scaling
190	215	IN.x - A object 1 byte value				HVAC mode	Sends HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
190	215	IN.x - A object 2 bytes value				Unsigned value	Send unsigned values (0..65535) associated to object A	7.001 DPT_Value_2_Ucount
190	215	IN.x - A object 2 bytes value				Signed value	Send signed values (-32768..32767) associated to object A	8.001 DPT_Value_2_Count
190	215	IN.x - A object 3 byte value				RGB color	Send RGB color components value associated to object A	232.600 DPT_Colour_RGB
190	215	IN.x - A object 4 bytes value				Unsigned value	Sends unsigned values (0..4294967295) associated to object A	12.001 DPT_Value_4_Ucount

190	215	IN.x - A object 4 bytes value	Signed value	Sends signed values (-2147483648.. 2147483647) associated to object A	13.001 DPT_Value_4_Count
190	215	IN.x - A object 14 bytes value	Characters ISO 8859-1	Sends characters encoded according to ISO 8859-1 standard	16.001 DPT_String_8859_1
191	216	IN.x - Brightness dimming	Increase/Decrease	Sends relative brightness control commands	3.007 DPT_Control_Dimming
191	216	IN.x - Shutter stop/Louvres control	Stop/Step	Sends movement stop/shutter control commands	1.007 DPT_Step
191	216	IN.x - B object 1 bit value	1/0 value	Send 1/0 values associated to object B	1.002 DPT_Bool
192	217	IN.x - C object 1 bit value	1/0 value	Send 1/0 values associated to object C	1.002 DPT_Bool
193	218	IN.x - D object 1 bit value	1/0 value	Send 1/0 values associated to object D	1.002 DPT_Bool
247		Logic operation output	Logic	Logic function output	1.002 DPT_Bool

The object variants shown in blue in the table above are not shown for objects B (objects 133/145/157/169/181), C (objects 135/147/159/171/183), D (object 137/149/161/173/185), but are still present.

The object variants shown in light blue in the table above are not shown for objects B (objects 191/216), C (objects 192/217) and D (objects 193/218) due to space problems, but are still present.

### **Communication objects with input functions– from here**

#	Object name	Object function	Description	Datapoint type
59	Day/Hour input	Update day/hour	Receives updates about day of the week and hour of the day	10.001 DPT_TimeOfDay
60	Daylight saving time input	1= daylight saving/0= standard	Receives updates of hourly convention in course	1.001 DPT_Switch
62	Glass cleaning	Switching On/Off	It allows to activate/deactivate the "Glass cleaning" function	1.001 DPT_Switch
63	Enabling slave function	Enable/Disable	Receives Slave function enabling commands	1.003 DPT_Enable
64	HVAC mode input	Eco/Precom/Comfort/Off	Receives HVAC mode setting commands	20.102 DPT_HVACMode
64	Setpoint input	Value °C	Receives functioning setpoint values in degrees Celsius	9.001 DPT_Temp Celsius
64	Setpoint input	°K value	Receives functioning setpoint values in degrees Kelvin	9.002 DPT_Tempd Kelvin
64	Setpoint input	°F value	Receives functioning setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F Fahrenheit
64	Setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the functioning setpoint value	1.007 DPT_Step

65	Setpoint temporary forcing regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the setpoint temporary forcing value	1.007 DPT_Step
66	HVAC off mode input	Enable/Disable	Receives HVAC off (building protection) mode activation commands	1.003 DPT_Enable
67	HVAC economy mode input	Enable/Disable	Receives HVAC economy mode activation commands	1.003 DPT_Enable
68	HVAC precomfort mode input	Enable/Disable	Receives HVAC precomfort mode activation commands	1.003 DPT_Enable
69	HVAC comfort mode input	Enable/Disable	Receives HVAC comfort mode activation commands	1.003 DPT_Enable
70	Functioning type input	Heating/Air cooling	Receives commands of functioning type setting	1.100 DPT_Heat/Cool
71	Enable dead zone	Enable/Disable	Receives automatic (dead zone) functioning type setting enabling commands	1.003 DPT_Enable
72	Heating anti-freeze setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the heating HVAC mode off setpoint	1.007 DPT_Step
72	Heating anti-freeze setpoint input	Value °C	Receives setpoint values heating off HVAC mode in degrees Celsius	9.001 DPT_Temp
72	Heating anti-freeze setpoint input	°K value	Receives setpoint values heating off HVAC mode in degrees Kelvin	9.002 DPT_Tempd
73	Heating anti-freeze setpoint input	°F value	Receives setpoint values heating off HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
73	Heating economy setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the heating HVAC mode economy setpoint	1.007 DPT_Step
73	Heating economy setpoint input	Value °C	Receives setpoint values heating economy HVAC mode in degrees Celsius	9.001 DPT_Temp
73	Heating economy setpoint input	°K value	Receives setpoint values heating economy HVAC mode in degrees Kelvin	9.002 DPT_Tempd
74	Heating economy setpoint input	°F value	Receives setpoint values heating economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
74			Receives	1.007 DPT_Step

	Heating precomfort setpoint regulation	1 = Increase/0 = Decrease	increase/decrease step commands of the heating HVAC mode precomfort setpoint	
74	Heating precomfort setpoint input	Value °C	Receives setpoint values heating precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp
74	Heating precomfort setpoint input	°K value	Receives setpoint values heating precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
75	Heating precomfort setpoint input	°F value	Receives setpoint values heating precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
75	Heating comfort setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the heating HVAC mode comfort setpoint	1.007 DPT_Step
75	Heating comfort setpoint input	Value °C	Receives setpoint values heating comfort HVAC mode in degrees Celsius	9.001 DPT_Temp
75	Heating comfort setpoint input	°K value	Receives setpoint values heating comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
76	Heating comfort setpoint input	°F value	Receives setpoint values heating comfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
76	Air-cooling high temp. protec. setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the air cooling HVAC mode off setpoint	1.007 DPT_Step
76	Air-cooling high temp. protection setpoint input	Value °C	Receives setpoint values air cooling off HVAC mode in degrees Celsius	9.001 DPT_Temp
76	Air-cooling high temp. protection setpoint input	°K value	Receives setpoint values air cooling off HVAC mode in degrees Kelvin	9.002 DPT_Tempd
77	Air-cooling high temp. protection setpoint input	°F value	Receives setpoint values air cooling off HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
77	Air-cooling economy setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the air cooling HVAC mode economy setpoint	1.007 DPT_Step
77	Air-cooling economy setpoint input	Value °C	Receives setpoint values air cooling economy HVAC mode in degrees Celsius	9.001 DPT_Temp
77	Air-cooling economy setpoint input	°K value	Receives setpoint values	9.002 DPT_Tempd

			air cooling economy HVAC mode in degrees Kelvin	
78	Air-cooling economy setpoint input	°F value	Receives setpoint values air cooling economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
78	Air-cooling precomfort setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the air cooling HVAC mode precomfort setpoint	1.007 DPT_Step
78	Air-cooling precomfort setpoint input	Value °C	Receives setpoint values air cooling precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp
78	Air-cooling precomfort setpoint input	°K value	Receives setpoint values air cooling precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
79	Air-cooling precomfort setpoint input	°F value	Receives setpoint values air cooling precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
79	Air-cooling comfort setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the air cooling HVAC mode comfort setpoint	1.007 DPT_Step
79	Air-cooling comfort setpoint input	Value °C	Receives setpoint values air cooling comfort HVAC mode in degrees Celsius	9.001 DPT_Temp
79	Air-cooling comfort setpoint input	°K value	Receives setpoint values air cooling comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
79	Air-cooling comfort setpoint input	°F value	Receives setpoint values air cooling comfort HVAC mode in degrees Fahrenheit	9.002 DPT_Temp_F
80	Setpoint regulation range setting	Value °C	Receives max setpoint regulation range in degrees Celsius	9.001 DPT_Temp
82	Heating valve status feedback	On/Off status	Receives feedback on electrovalve activation status 1st stage heating	1.001 DPT_Switch
82	Heating valve % feedback	% value	Receives feedback on electrovalve activation status 1st stage heating	5.001 DPT_Scaling
82	Heating/air cooling valve status feedback	On/Off status	Receives feedbacks on electrovalve activation status heating/air cooling	1.001 DPT_Switch
82	Heating/air cooling valve % feedback	% value	Receives feedbacks on electrovalve activation	5.001 DPT_Scaling

			status heating/air cooling	
86	Heating fan V1 status feedback	On/Off status	Receives feedbacks on fancoil speed 1 activation status heating	1.001 DPT_Switch
87	Heating fan V2 status feedback	On/Off status	Receives feedbacks on heating fancoil speed 2 activation status	1.001 DPT_Switch
88	Heating fan V3 status feedback	On/Off status	Receives feedbacks on heating fancoil speed 3 activation status	1.001 DPT_Switch
90	Heating 2nd stage feedback	On/Off status	Receives feedbacks on electrovalve activation status 2nd stage heating	1.001 DPT_Switch
90	Heating 2° stage valve % feedback	% value	Receives feedbacks on electrovalve activation status 2nd stage heating	5.001 DPT_Scaling
92	Air cooling valve status feedback	On/Off status	Receives feedback on electrovalve activation status 1st stage air cooling	1.001 DPT_Switch
92	Air cooling valve % feedback	% value	Receives feedback on electrovalve activation status 1st stage air cooling	5.001 DPT_Scaling
96	Air cooling fan V1 status feedback	On/Off status	Receives feedbacks on fancoil speed 1 activation status air cooling	1.001 DPT_Switch
97	Air cooling fan V2 status feedback	On/Off status	Receives feedbacks on air cooling fancoil speed 2 activation status	1.001 DPT_Switch
98	Air cooling fan V3 status feedback	On/Off status	Receives feedbacks on air cooling fancoil speed 3 activation status	1.001 DPT_Switch
100	Air cooling 2nd stage feedback	On/Off status	Receives feedback on electrovalve activation status 2nd stage air cooling	1.001 DPT_Switch
100	Air cooling 2° stage valve % feedback	% value	Receives feedback on electrovalve activation status 2nd stage air cooling	5.001 DPT_Scaling
101	Fancoil mode input	Automatic/manual	Receives commands of fancoil speed automatic mode selection or manual speed	1.001 DPT_Switch
103	KNX external sensor input	Value °C	Receives external KNX sensor values in degrees Celsius	9.001 DPT_Temp
103	KNX external sensor input	°K value	Receives external KNX sensor values in degrees Kelvin	9.002 DPT_Tempd
103	KNX external sensor input	°F value	Receives external KNX sensor values in degrees Fahrenheit	9.027 DPT_Temp_F

103	KNX floor sensor input				Value °C	Receives floor external KNX sensor values in degrees Celsius	9.001 DPT_Temp
103	KNX floor sensor input				°K value	Receives floor external KNX sensor values in degrees Kelvin	9.002 DPT_Tempd
103	KNX floor sensor input				°F value	Receives floor external KNX sensor values in degrees Fahrenheit	9.027 DPT_Temp_F
122	Feedback sending trigger				Send feedback	Receives trigger commands of feedback sending request	1.017 DPT_Trigger
123	Thermostat scene				Execute/Store	Receives thermostat function scene execution/storing commands	18.001 DPT_SceneControl
124	Humidity external sensor input				Value [% RH]	Receives the relative humidity values from the external KNX humidity sensor	9.007 DPT_Value_Humidity
127	139	151	163	175	Relative humidity threshold enabling 0=Enable/1=Disable	Receives relative humidity threshold x enabling commands	1.002 DPT_Bool
127	139	151	163	175	Relative humidity threshold enabling 1=Enable/0=Disable	Receives relative humidity threshold x enabling commands	1.002 DPT_Bool
129	141	153	165	177	Relative humidity threshold x value regulation 1 = Increase/0 = Decrease	Receives increase/decrease step commands of the relative humidity threshold x value	1.007 DPT_Step
129	141	153	165	177	Relative humidity threshold x value input Value [% RH]	Receives the relative humidity threshold x value	9.007 DPT_Value_Humidity
132	144	156	168	180	Output A relative humidity threshold x reference HVAC Mode (com/precom/eco/off)	Receives the reference HVAC mode values (comfort/precomfort/economy/off) for the calculation of the output A associated with the relative humidity threshold x	20.102 DPT_HVAC_Mode
132	144	156	168	180	Output A relative humidity threshold x reference Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output A associated with the relative humidity threshold x	9.001 DPT_Temp
132	144	156	168	180	Output A relative humidity threshold x reference Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output A associated with	9.002 DPT_Tempd

							the relative humidity threshold x	
132	144	156	168	180	Output A relative humidity threshold x reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of the output A associated with the relative humidity threshold x	9.027 DPT_Temp_F
132	144	156	168	180	Relative humidity threshold x output A status	On/Off status	Receives feedback about the state of activation of the output A associated with the relative humidity threshold x	1.001 DPT_Switch
134	146	158	170	182	Output B relative humidity threshold x reference	HVAC Mode (com/precom/eco/off)	Receives the reference HVAC mode values (comfort/precomfort/economy/off) for the calculation of the output B associated with the relative humidity threshold x	20.102 DPT_HVAC_Mode
134	146	158	170	182	Output B relative humidity threshold x reference	Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output B associated with the relative humidity threshold x	9.001 DPT_Temp
134	146	158	170	182	Output B relative humidity threshold x reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output B associated with the relative humidity threshold x	9.002 DPT_Tempd
134	146	158	170	182	Output B relative humidity threshold x reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of the output B associated with the relative humidity threshold x	9.027 DPT_Temp_F
134	146	158	170	182	Relative humidity threshold x output B status	On/Off status	Receives feedback about the state of activation of the output B associated with the relative humidity threshold x	1.001 DPT_Switch
136	148	160	172	184	Output C relative humidity threshold x reference	HVAC Mode (com/precom/eco/off)	Receives the reference HVAC mode values (comfort/precomfort/economy/off) for the calculation of the output C associated with the relative humidity threshold x	20.102 DPT_HVAC_Mode

136	148	160	172	184	Output C relative humidity threshold x reference	Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output C associated with the relative humidity threshold x	9.001 DPT_Temp
136	148	160	172	184	Output C relative humidity threshold x reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output C associated with the relative humidity threshold x	9.002 DPT_Tempd
136	148	160	172	184	Output C relative humidity threshold x reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of the output C associated with the relative humidity threshold x	9.027 DPT_Temp_F
136	148	160	172	184	Relative humidity threshold x output C status	On/Off status	Receives feedback about the state of activation of the output C associated with the relative humidity threshold x	1.001 DPT_Switch
138	150	162	174	186	Output D relative humidity threshold x reference	HVAC Mode (com/precom/economy/off)	Receives the reference HVAC mode values (comfort/precomfort/economy/off) for the calculation of the output D associated with the relative humidity threshold x	20.102 DPT_HVAC_Mode
138	150	162	174	186	Output D relative humidity threshold x reference	Setpoint [°C]	Receives the reference setpoint values (°C) for the calculation of the output D associated with the relative humidity threshold x	9.001 DPT_Temp
138	150	162	174	186	Output D relative humidity threshold x reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output D associated with the relative humidity threshold x	9.002 DPT_Tempd
138	150	162	174	186	Output D relative humidity threshold x reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of the output D associated with the relative humidity threshold x	9.027 DPT_Temp_F
138	150	162	174	186	Relative humidity threshold x output D status	On/Off status	Receives feedback about the state of activation of the output D associated	1.001 DPT_Switch

							with the relative humidity threshold x	
187	Season selection input				1=Summer 0=Winter	/	Receives current season setting for thermal comfort	1.002 DPT_Bool
187	Season selection input				1=Winter 0=Summer	/	Receives current season setting for thermal comfort	1.002 DPT_Bool
189	214	IN.x - Block			Switching On/Off		Allows to activate/deactivate the block function	1.003 DPT_Enable
191	216	IN.x - Scene storing trigger			Store		Receives trigger commands of scene storing commands sending request	1.017 DPT_Trigger
210	235	IN.x - Dimmer status feedback			On/Off status		Receives status feedback by controlled dimmer	1.001 DPT_Switch
210	235	IN.x - A object status feedback			On/Off status		Receive status feedback from actuator for cyclical commutation function of object A	1.001 DPT_Switch
210	235	IN.1 - Movement feedback			Increase/Decrease		Receives feedback of movement direction in progress by shutter actuator	1.008 DPT_UpDown
211	236	IN.1 - B object status feedback			On/Off status			1.001 DPT_Switch
212	237	IN.1 - C object status feedback			On/Off status			1.001 DPT_Switch
213	238	IN.1 - D object status feedback			On/Off status			1.001 DPT_Switch
239	Logic input 1				Logic input	function	Logic function input	1.002 DPT_Bool
240	Logic input 2				Logic input	function	Logic function input	1.002 DPT_Bool
241	Logic input 3				Logic input	function	Logic function input	1.002 DPT_Bool
242	Logic input 4				Logic input	function	Logic function input	1.002 DPT_Bool
243	Logic input 5				Logic input	function	Logic function input	1.002 DPT_Bool
244	Logic input 6				Logic input	function	Logic function input	1.002 DPT_Bool
245	Logic input 7				Logic input	function	Logic function input	1.002 DPT_Bool
246	Logic input 8				Logic input	function	Logic function input	1.002 DPT_Bool

Ai sensi delle Decisioni e delle Direttive Europee applicabili, si informa che il responsabile dell'immissione del prodotto sul mercato Comunitario è:  
*According to the applicable Decisions and European Directives, the responsible for placing the apparatus on the Community market is:*

**GEWISS S.p.A. Via A.Volta, 1 IT-24069 Cenate Sotto (BG) Italy Tel: +39 035 946 111 Fax: +39 035 946 270 E-mail: [qualitymarks@gewiss.com](mailto:qualitymarks@gewiss.com)**



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



**+39 035 946 260**



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