

# CHORUS

# GEWISS

KNX Thermostat with humidity management



**GW 1x795H**

## 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 .....	17
4	"Heating" menu .....	28
4.1	Parameters.....	28
5	"Air cooling" menu.....	37
5.1	Parameters.....	37
6	"Temperature setpoint" menu .....	45
6.1	Parameters.....	45
7	"Temperature sensors" menu .....	50
7.1	Parameters.....	50
8	"Feedback" menu.....	56
8.1	Parameters.....	56
9	"Scenes" menu.....	62
9.1	Parameters.....	62
10	"KNX Remote devices control" menu .....	64
10.1	Parameters.....	64
11	"Input aux 1" and "Input aux 2" menu .....	67
11.1	Edge function (closing / opening).....	67
11.2	Short / long operation .....	71
11.3	1 push button + stop dimmer.....	75
11.4	Cyclic sending 1 push button dimmer .....	77
11.5	1 push button shutter control.....	79
11.6	Scene management .....	79
11.7	Window contact .....	81
12	"Auxiliary inputs 1/2" menu .....	82
12.1	"Dimming with STOP telegram" function.....	82
12.2	"Dimming with cyclic telegram" function.....	83
12.3	"Shutter control" function .....	85
13	"Auxiliary output" menu.....	86
13.1	Parameters.....	87
14	"Display" menu.....	90
14.1	Parameters.....	90
15	"Humidity" menu.....	93
15.1	Parameters.....	94
16	"Relative humidity threshold x" menu .....	97
16.1	Parameters.....	98
17	"Dew point" menu.....	105
17.1	Parameters.....	106
18	"Thermal comfort" menu .....	111
18.1	Parameters.....	111
19	Communication objects.....	114
20	ETS programming error feedback.....	127

# 1 Introduction

This manual explains the functions of the “**KNX flush-mounting thermostat with humidity management**” (GW1x795H), and how they are set and configured with the aid of the ETS configuration software.

## 2 Application

The KNX flush-mounting thermostat with humidity management 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 heating or air cooling elements (including the fan coils) and the humidification/dehumidification elements.

The thermostat can work in "autonomous" control mode, to autonomously manage the temperature adjustment system (or parts of it); when combined with the KNX flush-mounting timed thermostat, it can work in "Slave" control mode to create multi-area temperature adjustment systems. The thermostat allows you to visualise and independently modify the operating parameters of up to 4 KNX flush-mounting temperature adjustment probes.

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 device is 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%)

### Fan coil management

- control of fancoil speed, with ON/OFF selection commands or continuous adjustment (0% - 100%)
- 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 / KNX temperature adjustment probe / external temperature sensor with definition of the relative weight
- calculation of the dew temperature
- setting of 1 threshold associated with the dew temperature, with sending of BUS commands when the threshold is exceeded and restored

### Measuring relative humidity

- relative humidity measurement obtained from an external KNX sensor
- estimate of the relative humidity in the place where the thermostat is located
- 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 fronts management, short/long operation, dimmer with single push-button, roller shutters with single push-button, scenes and window contact
- auxiliary output for controlling the thermostat solenoid valve for heating/cooling, or used as a general output for executing ON/OFF commands, timed commands, priority commands and scene management
- 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. The basic structure of the menu is as follows:

Device: 1.5.3 KNX Thermostat with humidity - white

Main	Thermostat functioning	stand alone
Heating	Parameters local modification	disabled
Air cooling	Stand alone control type	setpoint
Temperature setpoint	Functioning type setting	by local command or from bus
Temperature sensors	Remote parameters setting	disable
Feedbacks	Allows local switching off	disable
Scenes	Input contacts function	two auxiliary inputs independent
KNX Remote devices control	Day and Hour modification via bus	disable
Display	Automatic change Daylight Saving Time/Standard Time	disable
Input aux 1	Control logic Heating/Air cooling	different (4 pipe)
Input aux 2	Heating control algorithm	two points ON-OFF
Auxiliary output	Air cooling control algorithm	two points ON-OFF
Humidity		
Relative humidity threshold 1		
Relative humidity threshold 2		
Relative humidity threshold 3		
Relative humidity threshold 4		
Relative humidity threshold 5		
Dew point		
Thermal comfort		

Fig. 3.1

## 3.1 Parameters

### ➤ 3.1.1 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.

### ➤ 3.1.2 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
- only general parameters
- general and operating parameters
- **general, operating and control parameters** (default value)

The various access levels are indicated below (partial/complete):

- **General parameters:** the following parameters are displayed if “**Parameters local modification**” is set to any value other than **disabled**:
  - *Hours*
  - *Minutes*
  - *Standard time/day-light saving time*
  - *°C/°F*
  - *Back to main page*
  - *Main page timeout*
  - *Backlight colour*
  - *Icons theme*
  - *Single-colour theme for icon colour*
  - *Backlight timeout*
  - *Backlight intensity*
  - *Backlight % value*
  - *Light sensitive sensor offset*
- **Operating parameters:** the following parameters are displayed if “**Parameters local modification**” is set to **general and operating parameters** or **general, operating and control parameters**:
  - *Heat/Cool/Auto*
  - *T1/SetP/Enable humidity threshold 1*
  - *T2/Enable humidity threshold 2*
  - *T3/Enable humidity threshold 3*
  - *Toff/Enable humidity threshold 4*
  - *Enable humidity threshold 5*
  - *Enable dew point alarm threshold*
  - *Humidity threshold 1*
  - *Humidity threshold 2*
  - *Humidity threshold 3*
  - *Humidity threshold 4*
  - *Humidity threshold 5*
  - *Dew point alarm feedback limit*
- **Control parameters:** the following parameters are also displayed if “**Parameters local modification**” is set to **general, operating and control parameters**:
  - *Proportional band*

- *Integration time*
- *Cycle time*
- *Min. % value for command sending*
- *Hysteresis amplitude*
- *Valve hysteresis*
- *V1 hysteresis*
- *V2 hysteresis*
- *V3 hysteresis*
- *V1 inertia*
- *V2 inertia*
- *V3 inertia*
- *2nd stage hysteresis*
- *Slave/stand alone*
- *Remote commands*

### ➤ 3.1.3 Stand alone control type

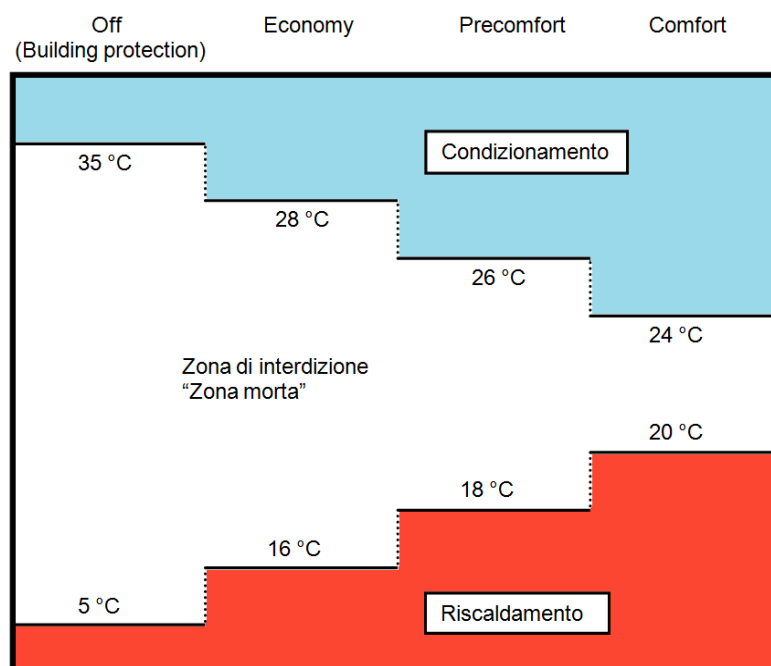
If the device functions as "stand alone", it is possible to define whether only one operating setpoint should be managed or whether the operation should be managed via the HVAC modes. The **"Stand alone control type"** parameter is used to set the type of control you want to use. The values that can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

- HVAC mode (default value)
- Setpoint

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

### ➤ 3.1.4 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 (autonomous 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)**
- only via the local menu

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).

### ➤ **3.1.5 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.

### ➤ **3.1.6 Remote parameters setting**

Even if the device functions as stand alone, it can still receive various commands from remote units via the BUS; the “**Remote parameters setting**” parameter 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 defines 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 “**Reception of 1 bit mode activation with priority > current**” and “**Reception of 1 bit mode activation with priority < current**” parameters, and the **HVAC off mode input**, **HVAC economy mode input**, **HVAC precomfort mode input**, **HVAC comfort mode input** and **HVAC auto mode input** communication objects (Data Point Type: 1.003 DPT\_Enable) which are used to activate the relative HVAC mode.

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

Priority	Object	Size
Maximum	Manual switch-off	-
	Reactivation after manual switch-off (if only manual reactivation)	-
	Window contact function aux input	-
	HVAC off mode input	1 bit
	HVAC economy mode input	1 bit
	HVAC precomfort mode input	1 bit
	HVAC comfort mode input	1 bit
Minimum	HVAC/Scene/Setpoint input/local commands mode input	1 byte/1 byte/2 byte/-
	Reactivation after manual switch-off (if manual reactivation and remote controls)	-

The 1-bit mode setting objects all have a higher priority than the 1-byte mode setting object and the local command; 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 way, 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 “**Reception of 1 bit mode activation with priority < current**” parameter.

The values that can be set for the “**Reception of 1 bit mode activation with priority > current**” parameter 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 priority lower than the one currently active, the new command is ignored (as if it had not been received).

The “**Modify HVAC mode setpoint via BUS**” parameter 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 the absolute value (°C)
- enable setting of the absolute value (°K)
- enable setting of the absolute value (°F)
- enable increase/decrease step regulation

Selecting a value other than **disable** 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) via 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), via which you can regulate the setpoints of each 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 in steps of 1 (default value 5)

Regardless of whether the mode setpoints are modified via the local menu or via a remote command via BUS, there is a value setting limit between the various setpoints belonging to the same functioning type, which is determined by the following relationship:

- $T_{\text{antifreeze}} \leq T_{\text{economy}} \leq T_{\text{precomfort}} \leq T_{\text{comfort}}$  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)

This constraint 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)

This constraint 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 in steps of 1 (default value 5)

As with the local push-buttons, you can temporarily force the current setpoint from the BUS 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 **Temporary setpoint forcing regulation** communication object (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 “**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 in steps of 1 (default value 5)

The “**Functioning type**” parameter 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 (e.g. for hotels) where the thermostat is requested to function as a Slave and the client can only change the operating setpoint slightly 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); this function is activated by pressing the MODE button key. The “**Allows local switching off**” parameter 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.

### ➤ 3.1.7 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)**
- two auxiliary inputs combined
- 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 auxiliary inputs independent**, a common menu (**Auxiliary input 1/2**) if the set value is **two auxiliary inputs combined** or a single menu (**Auxiliary input 1**) and the new parameter “**Auxiliary external sensor**” in the **Temperature sensors** menu if the value is **one auxiliary input and one temperature sensor**.

### ➤ 3.1.8 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.

Given that the device has no battery, 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.

### ➤ 3.1.9 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.

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

### ➤ 3.1.10 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
- fancoil with continuous 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** or **fancoil with continuous speed control** displays the “**Fancoil valve management**”, “**Valve regulation differential (tenth of °C)**”, “**Number of fancoil speeds**” and “**Fancoil speed status feedback**” parameters 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
- fan coil with ON-OFF speed control
- fan coil with continuous 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** or **fancoil with continuous speed control** displays the “**Fancoil valve management**”, “**Valve regulation differential (tenth of °C)**”, “**Number of fancoil speeds**” and “**Fancoil speed status feedback**” parameters 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
- fancoil with continuous 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 (air cooling 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 “**Selecting heating system (air cooling 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.

Select **fancoil with ON-OFF speed control** or **fancoil with continuous speed control** to display the “**Fancoil valve management**” and “**Valve regulation differential (tenth of °C)**” parameters, while in the **Heating** and **Air cooling** menus the “**Number of fancoil speeds**” and “**Fancoil speed status feedback**” parameters will be displayed.

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 defines 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/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. Selecting **2 points 0%-100%** displays 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 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 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 hysteresis defined in the "Control algorithms" paragraph. When the value “0” is received, 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.

When BUS voltage is restored, 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 enables the **Fancoil mode feedback** communication object via which the device indicates via BUS telegram the fancoil speed control mode (manual/automatic). The values that can be set are:

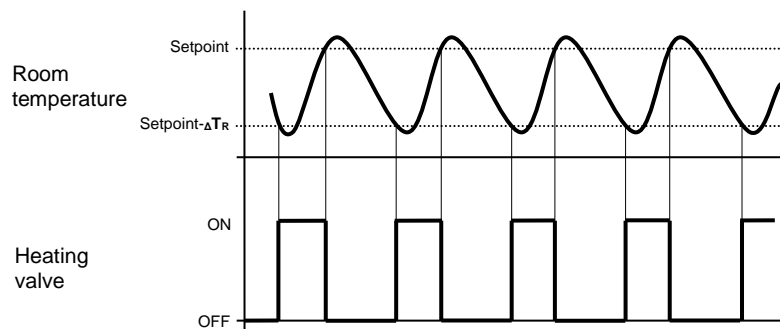
- **disable** (default value)
- enable



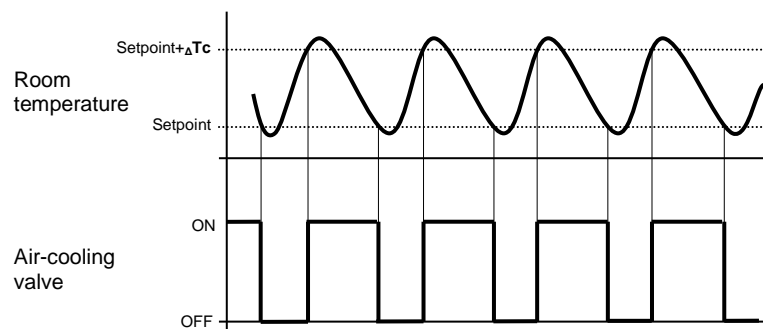
Selecting **enable** displays the **Fancoil mode feedback** communication object used to transmit the feedback. When fancoil speed control mode changes from MANUAL to AUTOMATIC, the device sends a telegram via BUS with the logical value "1"; when fancoil speed control mode changes from AUTOMATIC to MANUAL,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

### 3.2 Control algorithms

The algorithm used for controlling the temperature adjustment system is the classic type that is called 2 points 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- $\Delta TR$ " (where  $\Delta TR$  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 BUS 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-  $\Delta TR$ " 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.

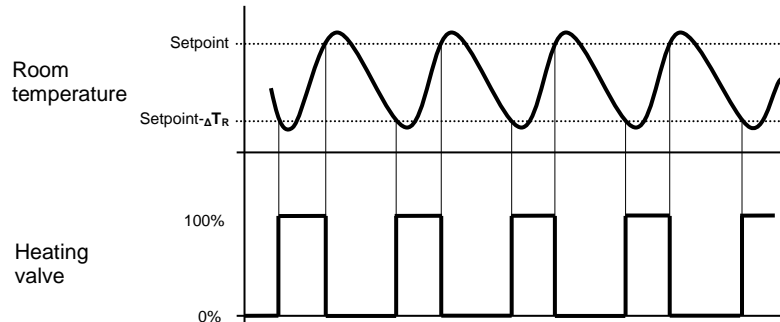


When the measured temperature is higher than the value "setpoint+  $\Delta Tc$ " (where  $\Delta Tc$  identifies the air cooling regulation differential) the device turns on the air cooling system, sending the relative BUS command to the actuator that manages it. When the detected temperature reaches the fixed setpoint value, the device turns off the air cooling system, sending the relative BUS command to the BUS 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+ $\Delta Tc$ " 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%**

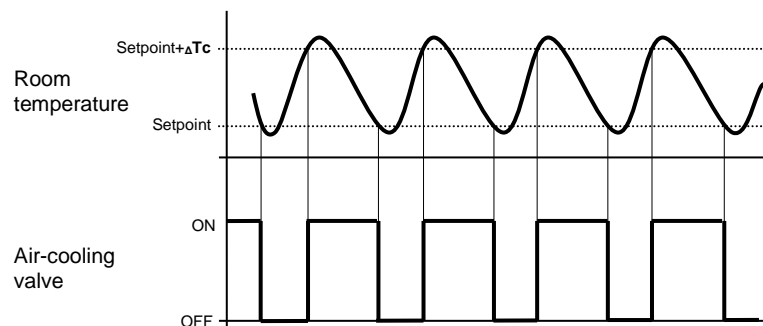
The algorithm used for controlling the temperature adjustment system is the classic type that is called 2 points control. This type of control involves the turning on and off of the temperature adjustment system following a hysteresis cycle. This means there isn't a single threshold that discriminates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

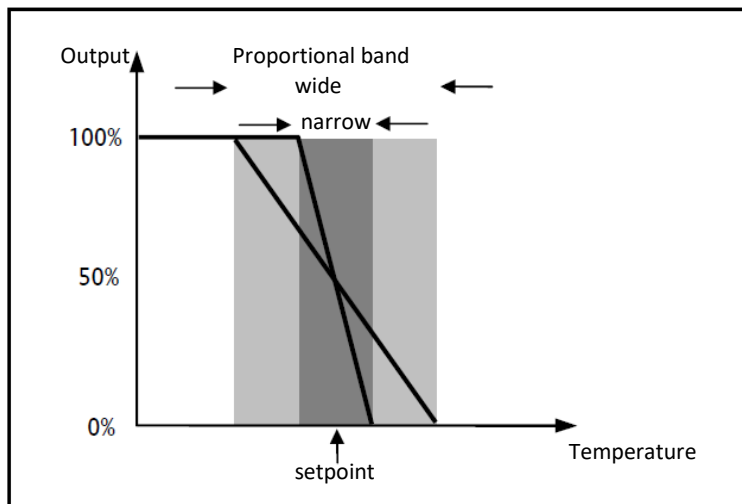
In the proportional component, the output function is proportional to the error (difference between setpoint and measured temperature).

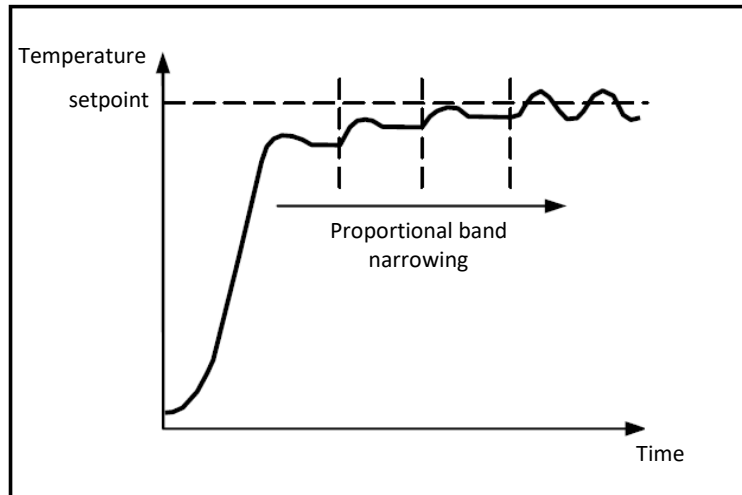
$$P_{out} = K_p e(t)$$

Once the proportional band is defined, the duty-cycle within the band varies between 0% and 100%; outside of 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.





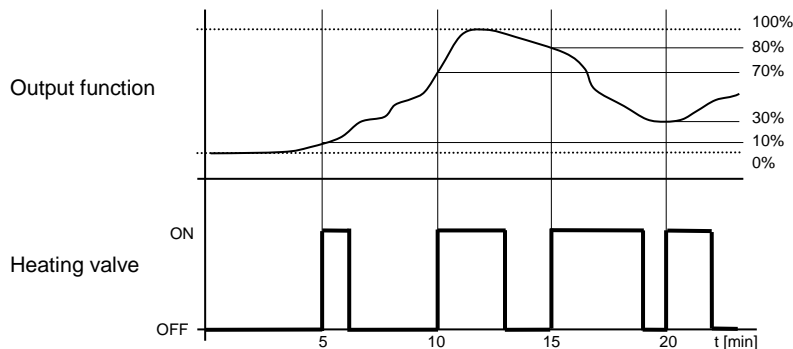
**Integral component**

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.

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

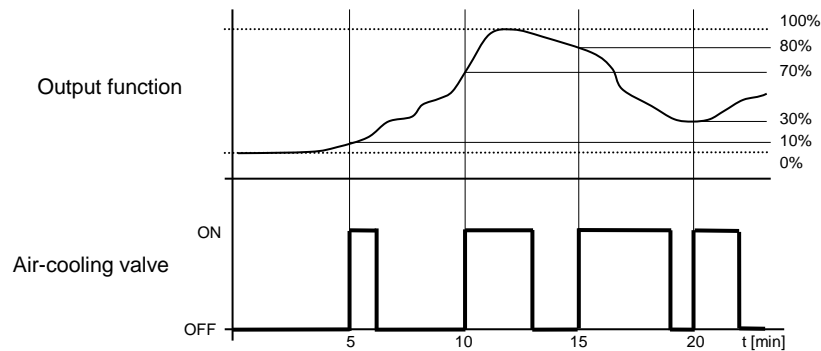
The integral period accelerates the dynamics of the process towards the setpoint and eliminates the residuals of the stationary error status that takes place with a pure proportional controller.

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 points 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 points 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 points 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

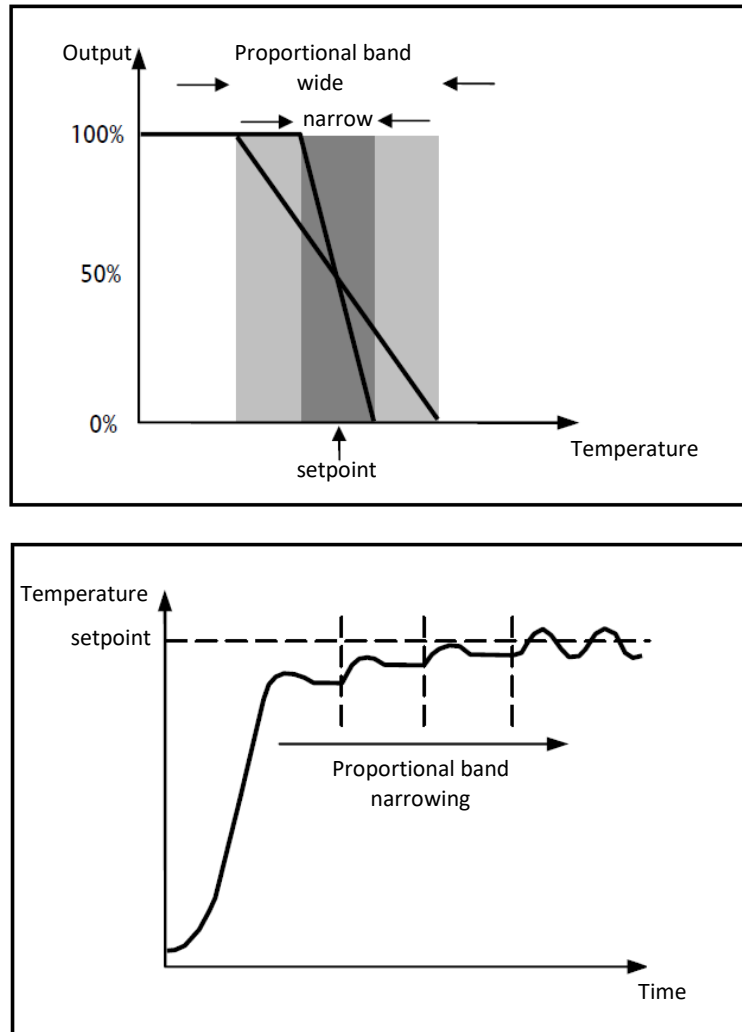
In the proportional component, the output function is proportional to the error (difference between setpoint and measured temperature).

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

Once the proportional band is defined, the output within the band varies between 0% and 100%; outside of 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.



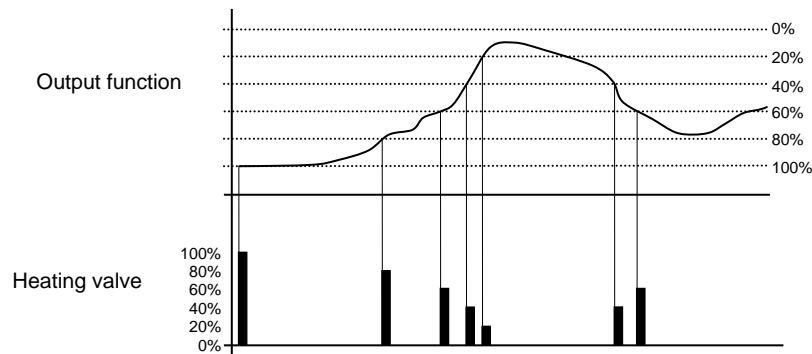
### Integral component

The contribution 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.

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

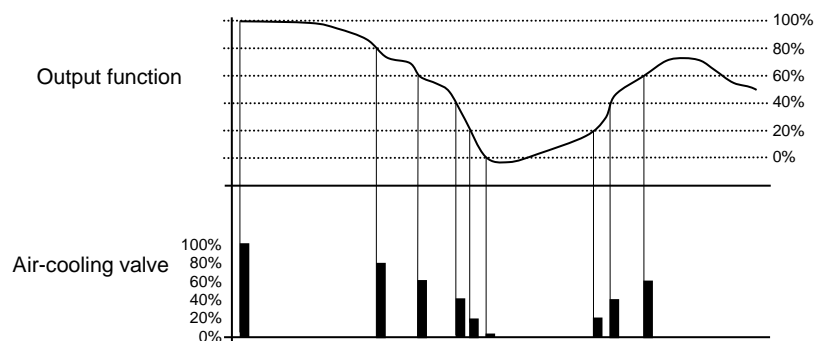
The integral period accelerates the dynamics of the process towards the setpoint and eliminates the residuals of the stationary error status that takes place with a pure proportional controller.

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 various levels with a distance equal to the value defined in **Px "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 between the calculated output function value and a specific 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 points 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.



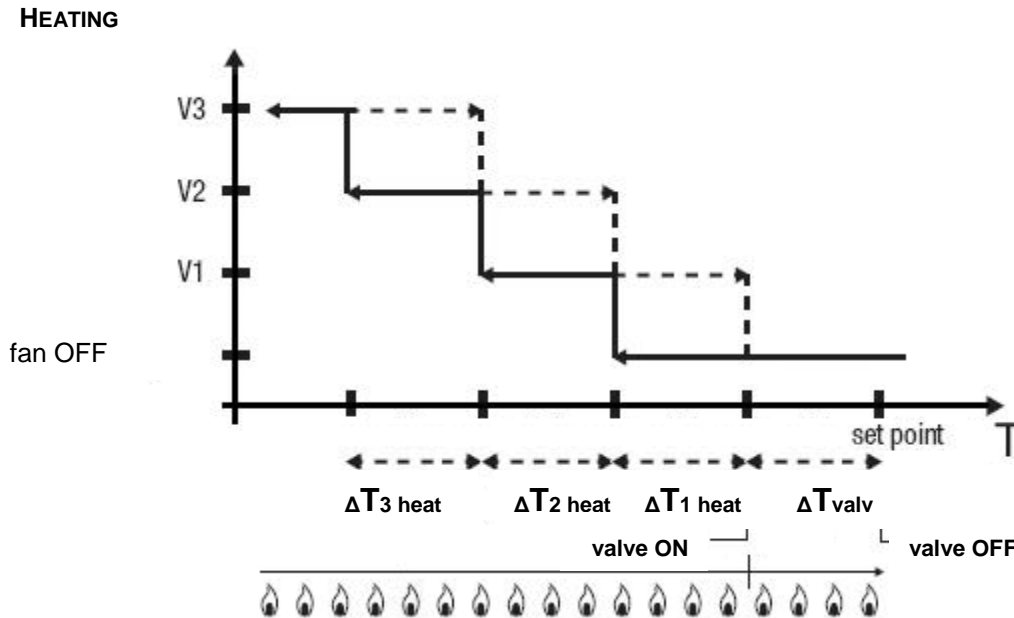
As can be seen in the figure, the device sends the commands to the actuator that manages the air cooling system based on the output function of the proportional-integral control. Along the vertical axis, the 0% - 100% interval of the output function of the proportional-integral control is divided into various levels with a distance equal to the value defined in **"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 between the calculated output function value and a specific 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 points 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.

• **fan coil with ON-OFF speed control**

The type of control that is applied when the fancoil control is enabled is similar to the 2 points 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). In short, this means that each stage corresponds to a speed and when the difference between the measured temperature and the setpoint that was set causes a certain speed to be turned on, this means that before turning on the new speed, the other two must absolutely be turned off.



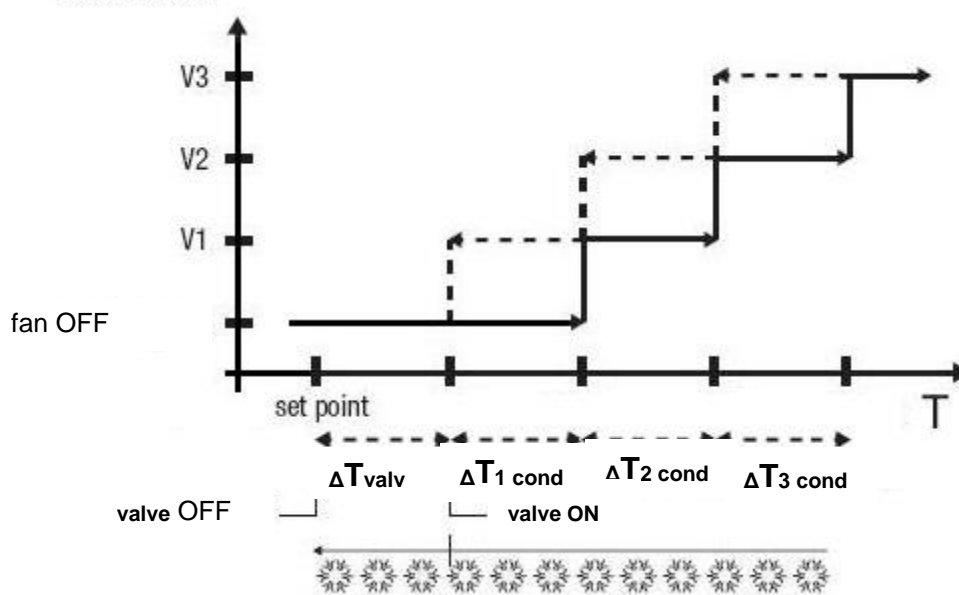
The figure refers to the control of the fancoil speeds with three operating stages for heating. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine its activation and deactivation. The thresholds are determined by values set for the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

- Speed V1 (1st stage): the speed is turned on when the temperature value is lower than the value “setpoint- $\Delta T_{\text{valv}}-\Delta T_{1 \text{ heat}}$ ” and turned off when the temperature value reaches the value “setpoint- $\Delta T_{\text{valv}}$ ” (or the “setpoint” value if  $\Delta T_{1 \text{ heat}}=0$ ). 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- $\Delta T_{\text{valv}}-\Delta T_{1 \text{ heat}}-\Delta T_{2 \text{ heat}}$ ” and turned off when the temperature value reaches the value “setpoint- $\Delta T_{\text{valv}}-\Delta T_{1 \text{ heat}}$ ”. 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- $\Delta T_{\text{valv}}-\Delta T_{1 \text{ heat}}-\Delta T_{2 \text{ heat}}-\Delta T_{3 \text{ heat}}$ ” and turned off when the temperature value reaches the value “setpoint- $\Delta T_{\text{valv}}-\Delta T_{1 \text{ heat}}-\Delta T_{2 \text{ heat}}$ ”

With regards the heating solenoid valve, once the measured temperature is lower than the value “setpoint- $\Delta T_{\text{valv}}$ ”, 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.



## AIR COOLING



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

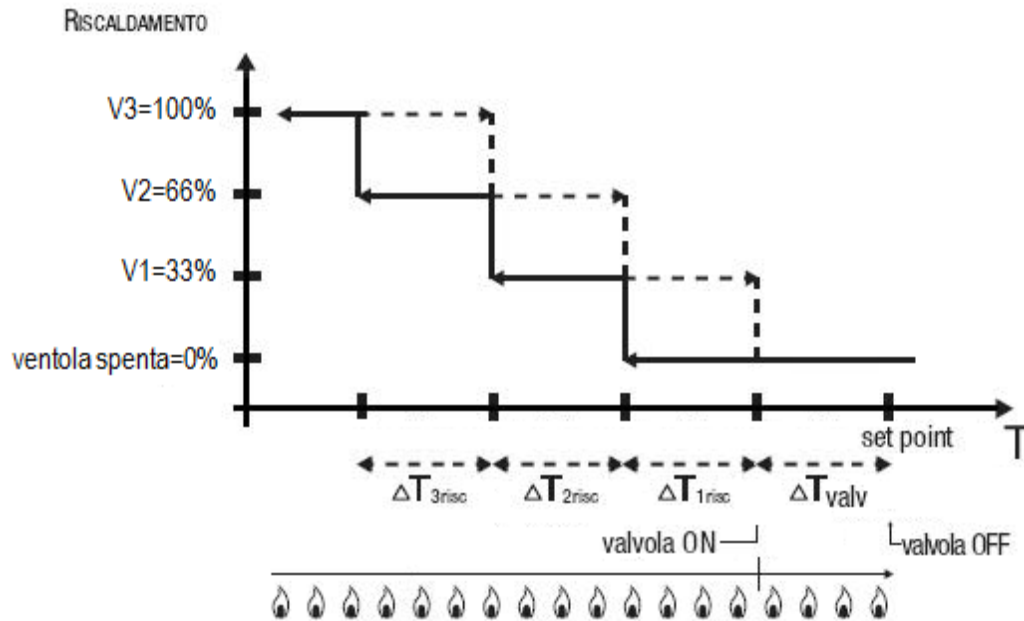
With regards the air cooling solenoid valve, once the measured temperature is higher than the value “setpoint+ $\Delta T_{valv}$ ”, the thermostat sends the activation command to the solenoid valve that manages the air cooling system; the solenoid valve is deactivated when the detected temperature reaches 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 fan coil 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.

### • fan coil with continuous speed control

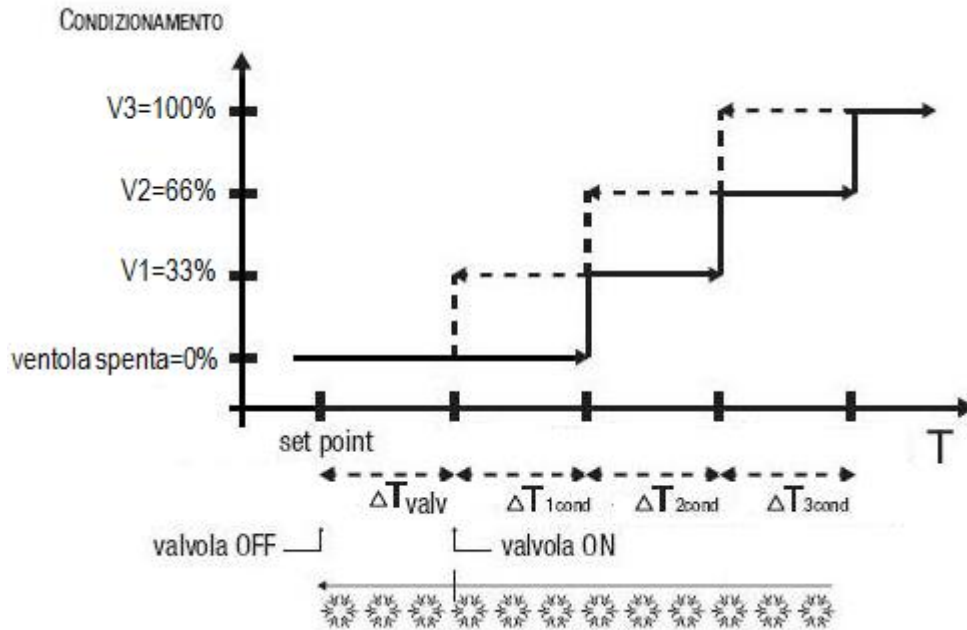
The substantial difference with the algorithm **fancoil with continuous speed control** is that in this case, there are no independent communication objects for managing the speeds and there is just one object, what will change is the value (1 byte) that is sent; this means that before turning on a speed, the others do not need to be turned off.



The figure refers to the control of the fancoil speeds with three operating stages for heating. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine the sending of the associated value. 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- $\Delta T_{valv}$ - $\Delta T_{1\text{ heat}}$ ” and turned off (sending the value “valve off”) when the temperature value reaches the value “setpoint- $\Delta T_{valv}$ ” (or the “setpoint” value if  $\Delta T_{1\text{ heat}}=0$ ). 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- $\Delta T_{valv}$ - $\Delta T_{1\text{ heat}}$ - $\Delta T_{2\text{ heat}}$ ” and turned off (sending speed V1) when the temperature value reaches the value “setpoint- $\Delta T_{valv}$ - $\Delta T_{1\text{ heat}}$ ”. 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- $\Delta T_{valv}$ - $\Delta T_{1\text{ heat}}$ - $\Delta T_{2\text{ heat}}$ - $\Delta T_{3\text{ heat}}$  and turned off (sending value V2) when the temperature value reaches the value “setpoint- $\Delta T_{valv}$ - $\Delta T_{1\text{ heat}}$ - $\Delta T_{2\text{ heat}}$ ”

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

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 fan coil 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.

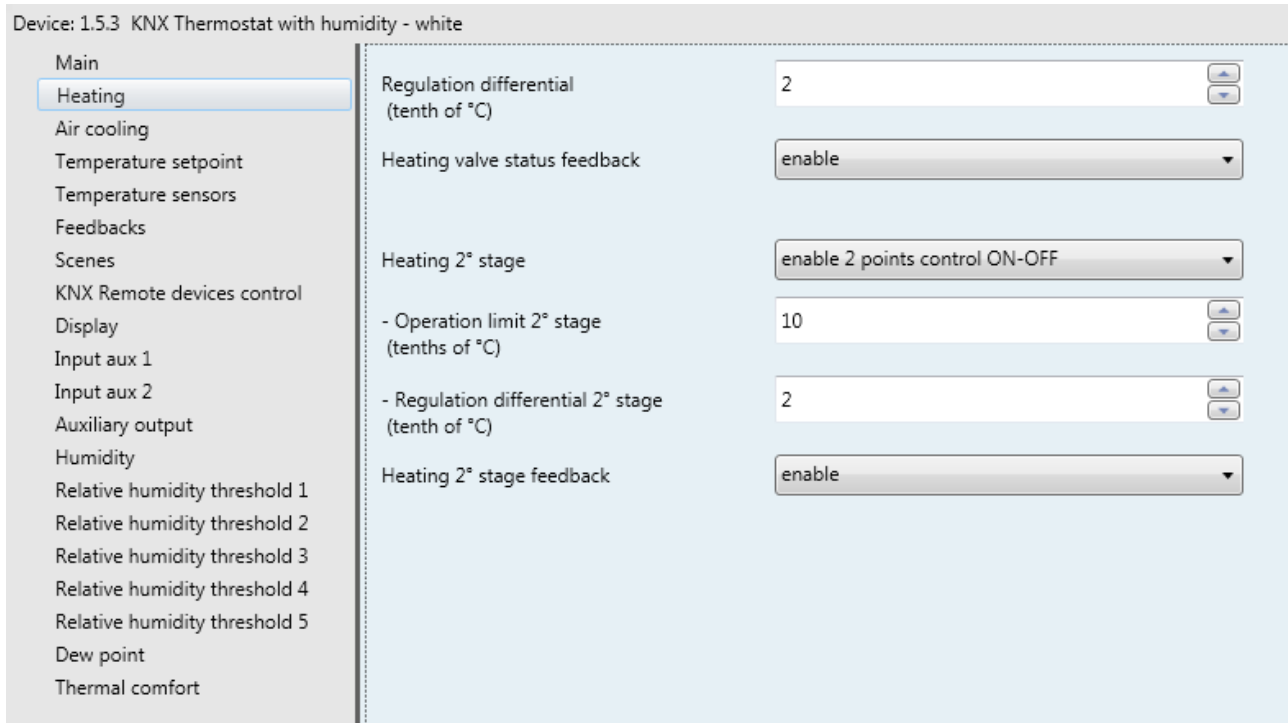


Fig. 4.1

### 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 in 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
- **floor 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 established 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 in 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 points 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 in 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 “**Heating control algorithm**” in the **Main** menu, the following communication objects are enabled:

- If the value set for the above item is **fancoil with ON-OFF speed control**, this enables the **Heating fan V1 switching** communication object (Data Point Type: 1.001 DPT\_Switch) for controlling the first and only fancoil speed.
- If the value set for the above item is **fancoil with continuous speed control**, this enables the **Heating fancoil speed % command** communication object (Data Point Type: 5.001 DPT\_Scaling) for controlling fancoil speed. In this case, the commands sent are percentage fancoil speed values, which can be summarised as follows:

<b>Fancoil speed</b>	<b>Sent percentage value</b>
<i>fan OFF</i>	0%
<i>first speed (V1)</i>	100%

- 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 “**Heating control algorithm**” in the **Main** menu, the following communication objects are enabled:

- If the value set for the above item is **fancoil with ON-OFF speed control**, this enables the **Heating fan V1 switching** and **Heating fan V2 switching** communication objects (Data Point Type: 1.001 DPT\_Switch) for respectively controlling the first and the second fancoil speed.
- If the value set for the above item is **fancoil with continuous speed control**, this enables the **Heating fancoil speed % command** communication object (Data Point Type: 5.001 DPT\_Scaling) for controlling fancoil speed. In this case, the commands sent are percentage fancoil speed values, which can be summarised as follows:

<b>Fancoil speed</b>	<b>Sent percentage value</b>
<i>fan OFF</i>	0%
<i>first speed (V1)</i>	50%
<i>second speed (V2)</i>	100%

- 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 “**Heating control algorithm**” in the **Main** menu, the following communication objects are enabled:

- If the value set for the above item is **fancoil with ON-OFF speed control**, this enables the **Heating fan V1 switching**, **Heating fan V2 switching** and **Heating fan V3 switching** communication objects (Data Point Type: 1.001 DPT\_Switch) for respectively controlling the first, second and third fancoil speed.
- If the value set for the above item is **fancoil with continuous speed control**, this enables the **Heating fancoil speed % command** communication object (Data Point Type: 5.001 DPT\_Scaling) for controlling fancoil speed. In this case, the commands sent are percentage fancoil speed values, which can be summarised as follows:

<b>Fancoil speed</b>	<b>Sent percentage value</b>
<i>fan OFF</i>	0%
<i>first speed (V1)</i>	33%
<i>second speed (V2)</i>	67%
<i>third speed (V3)</i>	100%

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- $\Delta T_{valv}$ ” value, determines the threshold value below which speed 1 of the fancoil is turned on. The values that can be set are:

- from 0 to 20 in steps of 1 (**default value 2**)

Setting the value **0** obtains the condition “ $\Delta T_{1 \text{ heat}} = \Delta T_{valv}$ ” 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 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- $\Delta T_{\text{valv}} - \Delta T_{1 \text{ heat}}$ ” value, determines the threshold value below which speed 2 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 in 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 “setpoint- $\Delta T_{\text{valv}} - \Delta T_{1 \text{ heat}} - \Delta T_{2 \text{ heat}}$ ” value, determines the threshold value below which speed 3 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 in 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 turning off speed 1 and turning on the new speed. The values that can be set are:

- from **0 (default value)** to 10, in 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 turning off speed 1 and turning on the new speed. The values that can be set are:

- from **0 (default value)** to 10, in 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 turning off speed 3 and turning on the new speed. The values that can be set are:

- from **0 (default value)** to 10, in 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:

- if the control algorithm is **fancoil with ON-OFF speed control**, the **Heating fan V1 status feedback**, **Heating fan V2 status feedback** and **Heating fan V3 status feedback** communication objects (Data Point Type: 1.001 DPT\_Switch), according to the number of fancoil speeds.
- if the valve control algorithm is **fancoil with continuous speed control**, the **Heating fancoil continuous control feedback** communication object (Data Point Type: 5.001 DPT\_Scaling), regardless of the number of fancoil speeds. The feedback received via this object can be different from 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 fancoil speed 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.

When 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** object in order to be updated about the activation status of the fancoil speeds.

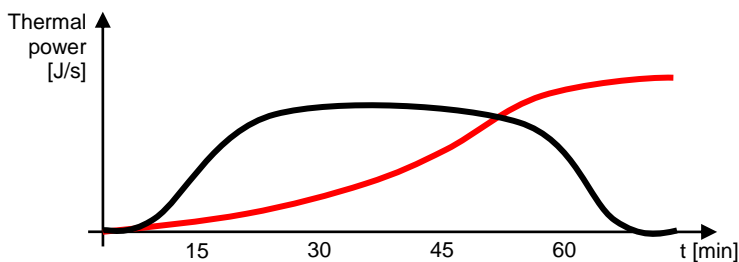
If the fancoil feedback is disabled, deactivation commands of the inactive speeds must be sent for every speed activation command. In the same way, every speed deactivation command must be sent together with deactivation commands for the other speeds.

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

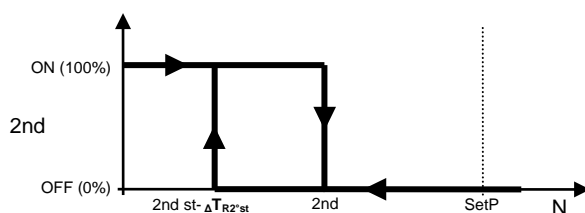
- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes** (default value)

If the control is **fancoil with ON-OFF speed control**, the commands are repeated on all the speed communication objects.

Some heating systems (for example, floor 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 points 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 the 2nd heating stage) the device activates the 2nd heating stage 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 the 2nd heating stage 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 the 2nd heating stage, 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 points control ON-OFF
- enable 2 points control 0%-100%

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

The “**Operation limit 2° stage (tenths of °C)**” parameter defines the intervention threshold for 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 system of the heating 2nd stage in the 2 points control is activated. The values that can be set are:

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

As for the basic heating algorithm, the “**Heating 2° 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 “**2° stage command repetition period without feedback**” parameter. Select **enable** to view the **Heating 2° stage feedback** communication object (Data Point Type: 1.001 DPT\_Switch) if the 2nd stage control algorithm is **2 points ON-OFF**, or **Heating 2° 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 2° stage feedback** or **Heating 2° stage valve % feedback** object, in order to be updated about the status of the heating second stage.

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 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 “**2° 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.

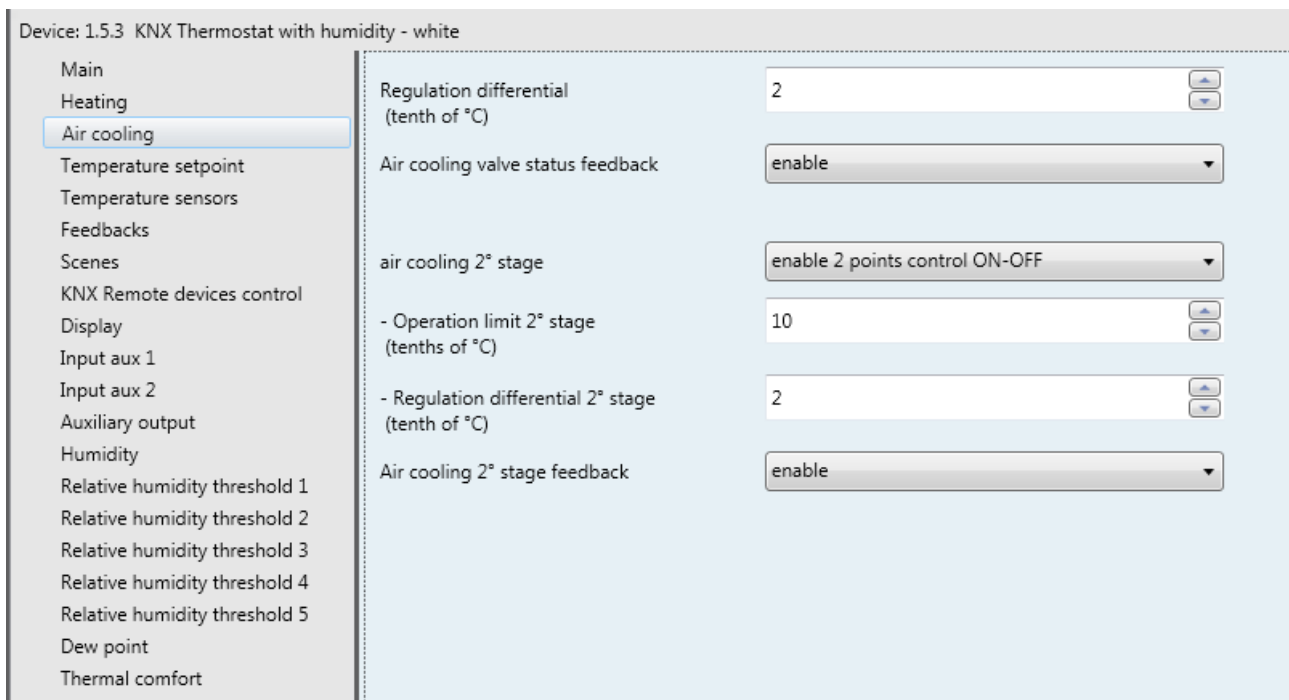


Fig. 5.1

### 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 in 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 air-cooling system (already mentioned in the "Control algorithms" section) which, when added to the established setpoint value, determines the upper limit of the proportional band used for the proportional integral control. The values that can be set are:

- 1.0°C
- 1.5°C
- **2.0°C** (default value)
- 2.5°C
- 3.0°C
- 3.5°C
- 4.0°C
- 4.5°C
- 5.0°C
- 5.5°C
- 6.0°C
- 6.5°C
- 7.0°C
- 7.5°C
- 8.0°C
- 8.5°C
- 9.0°C
- 9.5°C
- 10.0°C

The “**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 in 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 points 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 in 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, the following communication objects are enabled:

- if the value set for the above item is **fancoil with ON-OFF speed control**, this enables the **Air cooling fan V1 switching** communication object (Data Point Type: 1.001 DPT\_Switch) for controlling the first and only fancoil speed.
- if the value set for the above item is **fancoil with continuous speed control**, this enables the **Air cooling fancoil speed % command** communication object (Data Point Type: 5.001 DPT\_Scaling) for controlling fancoil speed. In this case, the commands sent are percentage fancoil speed values, which can be summarised as follows:

<b>Fancoil speed</b>	<b>Sent percentage value</b>
<i>fan OFF</i>	0%
<i>first speed (V1)</i>	100%

- 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:

- if the value set for the above item is **fancoil with ON-OFF speed control**, this enables the **Air cooling fan V1 switching** and **Air cooling fan V2 switching** communication objects (Data Point Type: 1.001 DPT\_Switch) for controlling the first and second fancoil speeds respectively.
- if the value set for the above item is **fancoil with continuous speed control**, this enables the **Air cooling fancoil speed % command** communication object (Data Point Type: 5.001 DPT\_Scaling) for controlling fancoil speed. In this case, the commands sent are percentage fancoil speed values, which can be summarised as follows:

<i>Fancoil speed</i>	<i>Sent percentage value</i>
<i>fan OFF</i>	0%
<i>first speed (V1)</i>	50%
<i>second speed (V2)</i>	100%

- **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:

- if the value set for the above item is **fancoil with ON-OFF speed control**, this enables the **Air cooling fan V1 switching**, **Air cooling fan V2 switching** and **Air cooling fan V3 switching** communication objects (Data Point Type: 1.001 DPT\_Switch) for controlling the first, second and third fancoil speeds respectively.
- if the value set for the above item is **1 byte (value %)**, this enables the **Air cooling fancoil speed % command** communication object (Data Point Type: 5.001 DPT\_Scaling) for controlling fancoil speed. In this case, the commands sent are percentage fancoil speed values, which can be summarised as follows:

<i>Fancoil speed</i>	<i>Sent percentage value</i>
<i>fan OFF</i>	0%
<i>first speed (V1)</i>	33%
<i>second speed (V2)</i>	67%
<i>third speed (V3)</i>	100%

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 “setpoint+ $\Delta T_{valv}$ ” value, determines the threshold value below which speed 1 of the fancoil is turned on. The values that can be set are:

- from 0 to 20 in steps of 1 (**default value 2**)

Setting the value **0** obtains the condition “ $\Delta T_{1\ cool} = \Delta T_{valv}$ ” for which the value of the speed 1 activation threshold is “setpoint+ $\Delta T_{valv}$ ” 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 “setpoint+ $\Delta T_{valv} + \Delta T_{1\ cool}$ ” value, determines the threshold value below which speed 2 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 in 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 “setpoint+ $\Delta T_{valv} + \Delta T_{1\ cool} + \Delta T_{2\ cool}$ ” value, determines the threshold value below which speed 3 of the fancoil is turned on. The values that can be set are:

- from 1 to 20 in 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 turning off speed 1 and turning on the new speed. The values that can be set are:

- from **0 (default value)** to 10, in 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 turning off speed 1 and turning on the new speed. The values that can be set are:

- from **0 (default value)** to 10, in 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 turning off speed 3 and turning on the new speed. The values that can be set are:

- from **0 (default value)** to 10, in 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 view the “**Fancoil speed command repetition period**” parameter and the following communication objects:

- if the control algorithm is **fancoil with ON-OFF speed control**, the ***Air cooling fan V1 status feedback, Air cooling fan V2 status feedback and Air cooling fan V3 status feedback*** communication objects (Data Point Type: 1.001 DPT\_Switch), according to the number of fancoil speeds.
- if the valve control algorithm is **fancoil with continuous speed control**, the ***Air cooling fancoil continuous control feedback*** communication object (Data Point Type: 5.001 DPT\_Scaling), regardless of the number of fancoil speeds. The feedback received via this object can be different from 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 fancoil speed 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.

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.

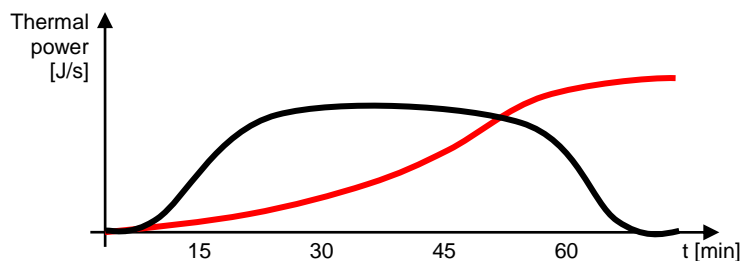
The “**Fancoil speed command repetition period**” parameter is used to define the time range of the cycling sending to the fancoil speeds. The values that can be set are:

- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes**

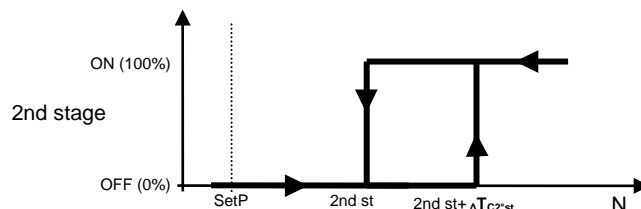
(default value)

If the control is **fancoil with ON-OFF speed control**, the commands are repeated on all the 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 only be a 2 points 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+  $\Delta T_{C2^{\circ}st}$ ” (where  $\Delta T_{C2^{\circ}st}$  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+  $\Delta T_{C2^{\circ}st}$ ” 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**
- enable 2 points control ON-OFF
- enable 2 points control 0%-100%

(default value)

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

The “**Operation limit 2° stage (tenths of °C)**” parameter is used to define the operation threshold for 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 2° 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 2nd stage air cooling in the 2 points control is activated. The values that can be set are:

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

As for the basic air cooling algorithm, the “**Air cooling 2° stage 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 “**2° stage command repetition period without feedback**” parameter. Select **enable** to view the ***Air cooling 2° stage 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 ***Air cooling 2° stage feedback*** or ***Air cooling 2° stage % feedback*** object, in order to be updated about the status of the 2nd 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.

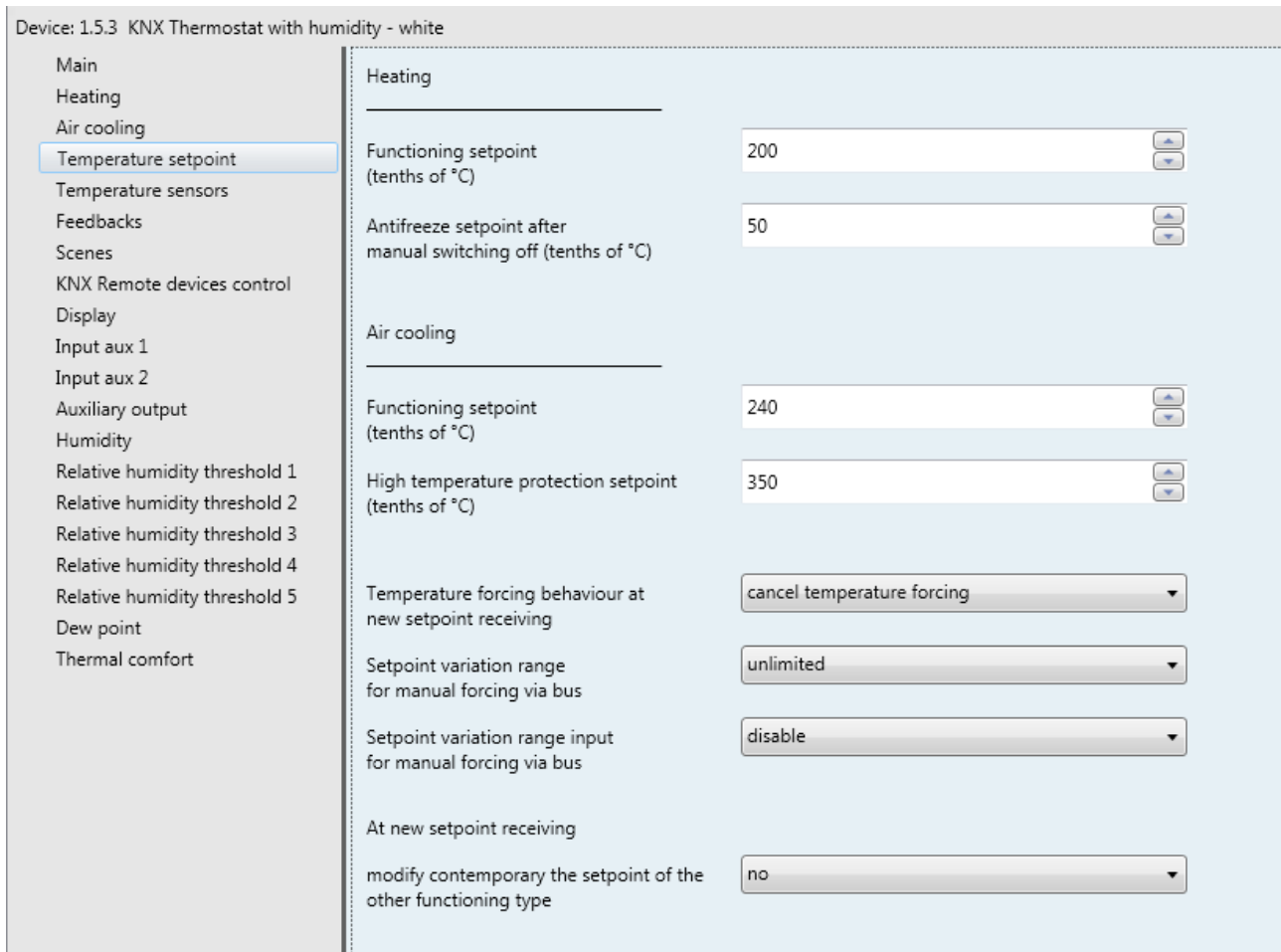
With 2nd stage air cooling status feedback disabled, it may be useful to cyclically repeat the command to the actuator so that if the first command telegram is lost, one of the subsequent ones will be received eventually. The “**2° stage command repetition period without feedback**” parameter is used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



Device: 1.5.3 KNX Thermostat with humidity - white

Section	Parameter	Value
Heating	Functioning setpoint (tenths of °C)	200
	Antifreeze setpoint after manual switching off (tenths of °C)	50
Air cooling	Functioning setpoint (tenths of °C)	240
	High temperature protection setpoint (tenths of °C)	350
	Temperature forcing behaviour at new setpoint receiving	cancel temperature forcing
	Setpoint variation range for manual forcing via bus	unlimited
	Setpoint variation range input for manual forcing via bus	disable
At new setpoint receiving	modify contemporary the setpoint of the other functioning type	no

Fig. 6.1

### 6.1 Parameters

#### ➤ 6.1.1 Heating section

The “**Comfort setpoint (tenth 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 in 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 “**Precomfort 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 “**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 in 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 “**Comfort setpoint (tenth of °C)**” and the value set for “**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 in 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 “**Precomfort setpoint (tenth of °C)**” and the value set for “**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 in 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 in 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 “**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 in 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 in 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 “**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 in 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 “**Comfort setpoint (tenth of °C)**” and the value set for “**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 in 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 “**Precomfort setpoint (tenth of °C)**” and the value set for “**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 in 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 in 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 “**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 in 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 forcing** (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 the Slave or stand alone operation of the device).



➤ **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, also the Air cooling Comfort setpoint is automatically modified and set equal to 25.5°C.

The parameter that makes it possible to enable the contemporary modification of the same mode for the two different functioning types is "**At new setpoint receiving modify 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 way, it may be useful to modify 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.

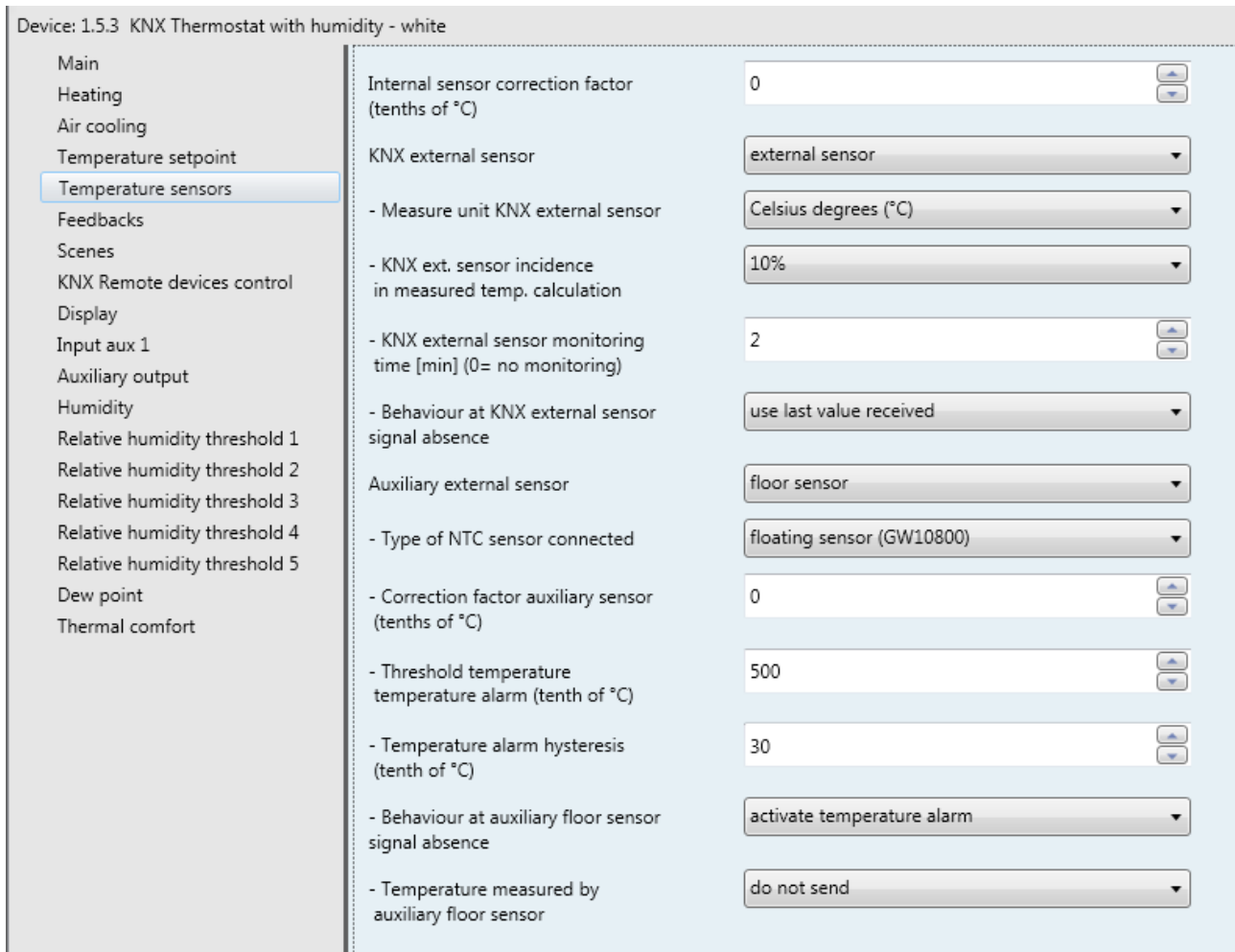
The parameter that makes it possible to enable the contemporary modification of the setpoints for the same functioning type is "**At new setpoint receiving modify the setpoints of the same functioning type**". The possible values are:

- **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 sensor. The structure of the menu is as follows:



Parameter	Value
Internal sensor correction factor (tenths of °C)	0
KNX external sensor	external sensor
- Measure unit KNX external sensor	Celsius degrees (°C)
- KNX ext. sensor incidence in measured temp. calculation	10%
- KNX external sensor monitoring time [min] (0= no monitoring)	2
- Behaviour at KNX external sensor signal absence	use last value received
Auxiliary external sensor	floor sensor
- Type of NTC sensor connected	floating sensor (GW10800)
- Correction factor auxiliary sensor (tenths of °C)	0
- Threshold temperature temperature alarm (tenth of °C)	500
- Temperature alarm hysteresis (tenth of °C)	30
- Behaviour at auxiliary floor sensor signal absence	activate temperature alarm
- Temperature measured by auxiliary floor sensor	do not send

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 in 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
- external sensor for humidity estimate

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 measure unit**”, “**Temperature alarm threshold temperature (tenth 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.

Selecting **external sensor for humidity estimate** displays the “**KNX external sensor measurement unit**”, “**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 in the same point where the humidity is measured. With this setting, the value received from the BUS is used merely to calculate the remote absolute humidity and therefore estimate the humidity at the temperature level measured by the device, without affecting the temperature measured by the device.

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 store the received value.

### ➤ 7.1.3 Measurement unit KNX external 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%, in 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 in 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:

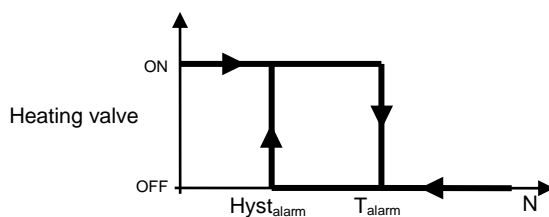
- use last value received (default value)
- exclude KNX sensor contribution

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 **KNX floor sensor input** communication object. 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 in 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 is used to define which of the possible sensors will be connected to the device contacts, in order to interface correctly with the sensor. The values that can be set are:

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

The “**Correction factor auxiliary sensor (tenths of °C)**” parameter is used to set the correction factor to be applied to the temperature value measured by the auxiliary sensor connected to the device to eliminate the contribution of heat generated by the site of installation. The values that can be set are:

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

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%, in steps of 10%

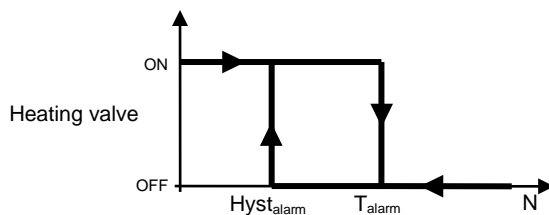
The auxiliary external sensor is always connected to the device, so we cannot speak - in this case - of sensor monitoring time; it is possible, however, to detect any potential NTC sensor malfunctioning and define how the device must react via the “**Behaviour at auxiliary external sensor signal absence**” 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 any potential NTC sensor malfunctioning 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 variation
- send periodically
- send on variation 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 variation** or **send on variation and periodically**, also the “**Minimum auxiliary sensor temperature variation for sending value  $\pm 0.1^{\circ}\text{C}$** ” parameter will be visible, whereas by selecting **send periodically** or **send on variation 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 ( $^{\circ}\text{C}$ )**      **(default value)**
- degrees Kelvin ( $^{\circ}\text{K}$ )
- degrees Fahrenheit ( $^{\circ}\text{F}$ )

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 ( $^{\circ}\text{C}$ )**, *9.002 DPT\_Value\_Tempd* if the value is **degrees Kelvin ( $^{\circ}\text{K}$ )** and *9.027 DPT\_Value\_Temp\_F* if the value is **degrees Fahrenheit ( $^{\circ}\text{F}$ )**.

The “**Minimum auxiliary sensor temperature variation for sending value  $\pm 0.1^{\circ}\text{C}$** ” parameter, 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 in steps of 1      **(default value 5)**

The “**Auxiliary sensor temperature sending period [minutes]**” parameter, 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 in 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:

Device: 1.5.3 KNX Thermostat with humidity - white

Main	Measured temperature	on variation and periodically
Heating	- Measure unit	Celsius degrees (°C)
Air cooling	- Minimum variation for sending value [± 0.1 °C]	5
Temperature setpoint	- Temperature sending period [minutes]	5
Temperature sensors	Functioning setpoint feedback	disabled
<b>Feedbacks</b>	Functioning type feedback	send on demand only
Scenes	Current setpoint report	disabled
KNX Remote devices control	Thermostat functioning feedback	disabled
Display	Feedbacks sending trigger object	disable
Input aux 1		
Auxiliary output		
Humidity		
Relative humidity threshold 1		
Relative humidity threshold 2		
Relative humidity threshold 3		
Relative humidity threshold 4		
Relative humidity threshold 5		
Dew point		
Thermal comfort		

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 variation
- send periodically
- send on variation 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 on variation** or **send on variation 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 variation 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 [ $\pm 0.1^{\circ}\text{C}$ ]

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 in 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 in 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 its 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 the **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\_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 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:

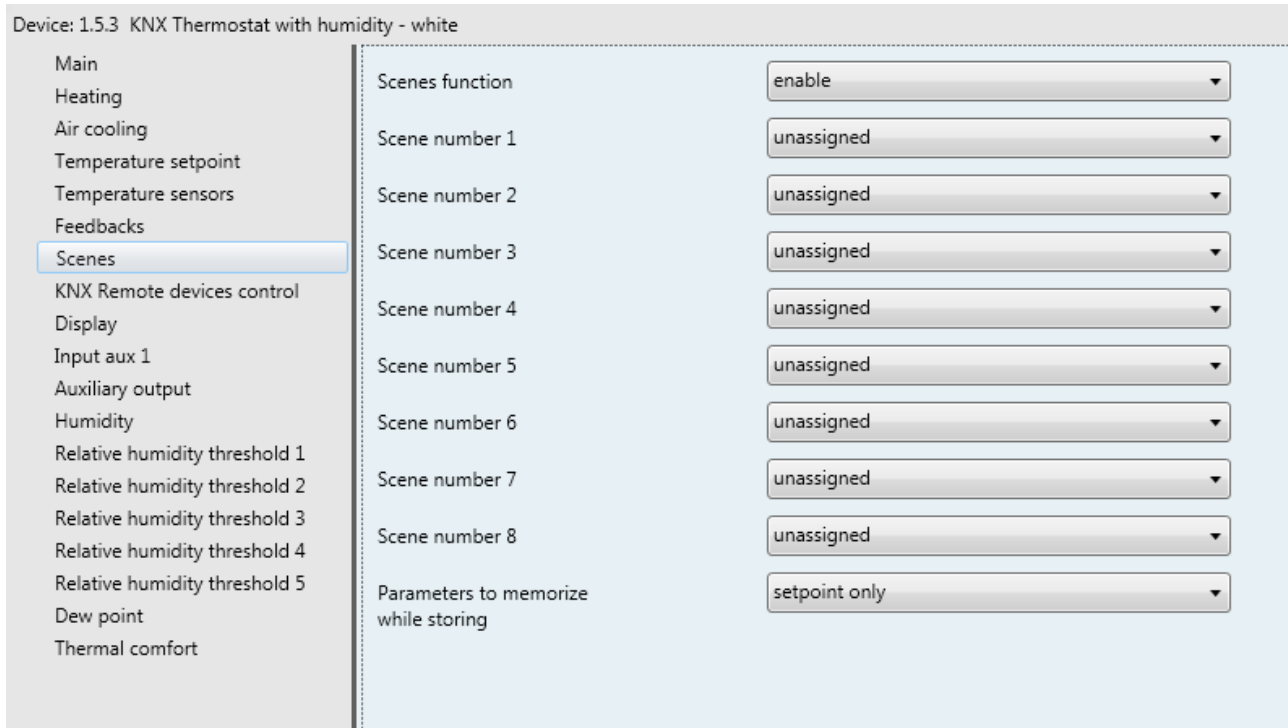


Fig. 9.1

### 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, you can set the numerical value for identifying and therefore executing/memorising the *i*-th scene. The possible values are:

- **not assigned** (default value)
- 0, 1.. 63

### ➤ 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 “KNX Remote devices control” menu

The **KNX Remote devices control** menu contains the parameters used to enable and define the control of remote devices, using the device to display the parameters of the remote elements.

The structure of the menu is as follows:

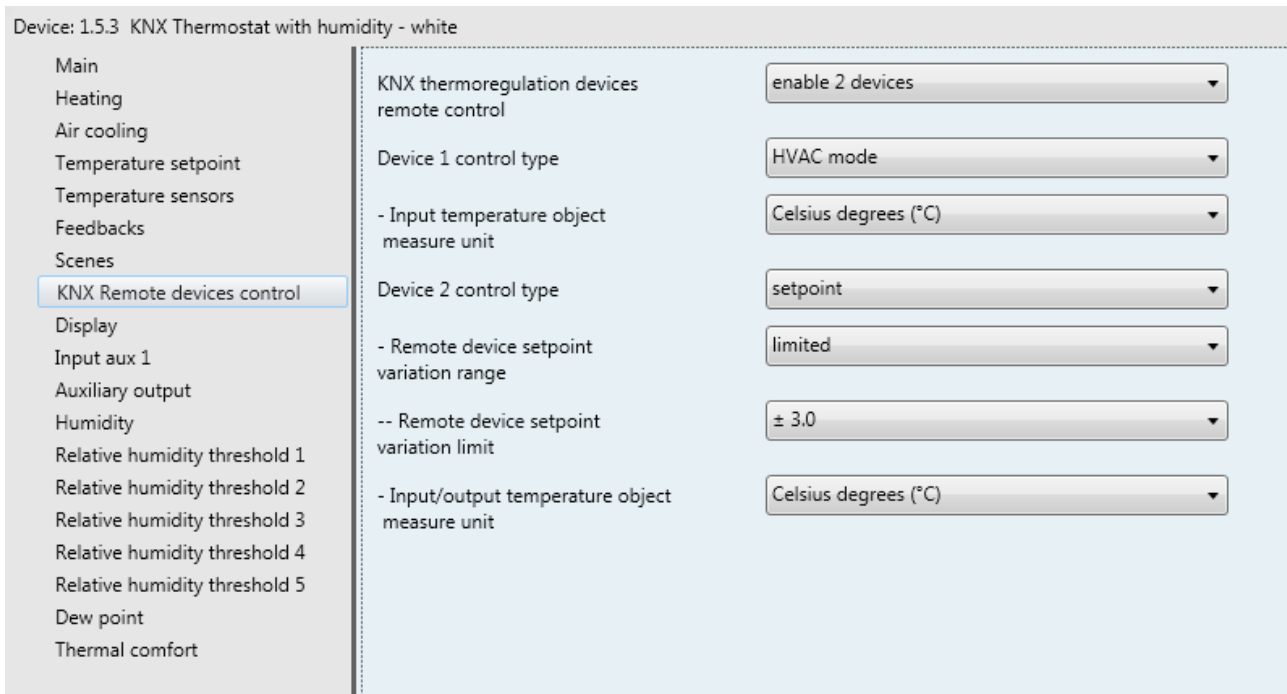


Fig. 10.1

### 10.1 Parameters

#### ➤ 10.1.1 KNX thermoregulation devices remote control

If the system in which the device is installed already has one or more elements with their own control logic (for example, KNX temperature adjustment sensor), the thermostat may also be used as a unit for displaying and setting operating parameters, delegating valve control to the other elements. The “**KNX thermoregulation devices remote control**” parameter defines whether (and how many) external elements with their own control logic can be controlled. The values that can be set are:

- **disabled** (default value)
- enable 1 device
- enable 2 devices
- enable 3 devices
- enable 4 devices

Depending on the value selected for these parameters, the “**Control device 1 with**”, “**Control device 2 with**”, “**Control device 3 with**” and “**Control device 4 with**” parameters will be displayed. These parameters are used to define if the thermostat controls the associated device through a single setpoint or by changing the active HVAC mode. The values that can be set are:

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



Select **HVAC mode** to display (for each device) the “**Input temperature object measure unit**” parameter and the **Device X - Send HVAC mode** communication objects (Data Point Type: 20.102 DPT\_HVACMode), **Device X - HVAC mode feedback** (Data Point Type: 20.102 DPT\_HVACMode), **Device X - Send functioning type**(Data Point Type: 1.100 DPT\_Heat/Cool), **Device X - Functioning type feedback**(Data Point Type: 1.100 DPT\_Heat/Cool), **Device X - Setpoint feedback** and **Device X - Measured temperature feedback**.

Select **setpoint** to display (for each device) the “**Input/output temperature object measure unit**” and “**Remote device setpoint variation range**” parameters and the **Device X - Send setpoint**, **Device X - Setpoint feedback**, **Device X - Send functioning type** communication objects (Data Point Type: 1.100 DPT\_Heat/Cool), **Device X - Functioning type feedback** (Data Point Type: 1.100 DPT\_Heat/Cool) and **Device X - Measured temperature feedback** (X = 1 .. 4).

The **Device X - Measured temperature feedback** communication object permits the device to receive and display the temperature measured by the potential KNX sensor associated with device X.

The **Device X - Functioning type feedback** communication object permits the device to receive and display the type of functioning active on the potential KNX sensor associated with device X.

The **Device X - HVAC mode feedback** communication object permits the device to receive and display the HVAC mode active on the potential KNX sensor associated with device X.

The **Device X - Setpoint feedback** communication object permits the device to receive and display the type of functioning setpoint active on the potential KNX sensor associated with device X.

For all the feedback objects mentioned above, status read request should be sent on BUS voltage recovery in order to update the relative information.

The **Device X - Send functioning type** communication object permits the device to send the functioning type to be used to the potential KNX sensor associated with Device X.

The **Device X - Send HVAC mode** communication object permits the device to send the HVAC mode to be used to the potential KNX sensor associated with Device X.

The **Device X - Send setpoint** communication object permits the device to send the functioning setpoint to be used to the potential KNX sensor associated with Device X.

### ➤ 10.1.2 Input temperature object measure unit

The “**Input temperature object measure unit**” parameter defines the measurement unit used to decode the information received via the **Device X - Measured temperature feedback** communication object. The values that can be set are:

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

The value set for this parameter changes the coding of the **Device X - Measured temperature feedback** 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)**.

### ➤ 10.1.3 Input/output temperature object measure unit

The “**Input/output temperature object measure unit**” parameter defines the measurement unit used to decode the information received via the **Device X - Measured temperature feedback**, **Device X - Setpoint feedback** and **Device X - Reference setpoint input** communication objects and to encode the information to be sent via the **Device X - Send setpoint** object. The values that can be set are:

- **degrees Celsius (°C)** 0 (default value)
- degrees Kelvin (°K) 1
- degrees Fahrenheit (°F) 2

The value set for this parameter changes the coding of the communication object indicated above: 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)**.

#### ➤ 10.1.4 Remote device setpoint variation range

If the remote device control type is setpoint, the adjustment range around the active setpoint on the remote device can be limited so that the thermostat does not change its operating setpoint to any great extent. This function is typically used in hotel applications where both the thermostat as well as the KNX sensor that controls it have a limited range for the temporary forcing of the operating setpoint. The “**Remote device setpoint variation range**” parameter is used to enable or disable the setpoint adjustment limit of the remote device. The values that can be set are:

- **unlimited**                    **(default value)**
- limited

Selecting **limited** displays the “**Remote device setpoint variation limit**” parameter and the **Device X - Reference setpoint input** communication object (Data Point Type: 9.001 *DPT\_Value\_Temp* if the value of “**Input/output temperature object measure unit**” 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)**) which is used by the device to be informed of the operating setpoint of the remote device to which the adjustment limit is applied while setting the parameters of the remote device.

The “**Remote device setpoint variation limit**” parameter sets 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:

- ± 0.1°C                    1
- ...
- **± 3.0°C**                    **30 (default value)**
- ...
- ± 5.0°C                    50

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.

EXAMPLE: if the setpoint variation range of the remote device is **± 3.0°C** and a value of 20°C is received via the **Device X - Reference setpoint input** object, then when setting the remote device setpoint, the value is limited to between 17 and 23°C. When the BUS voltage is restored, send a status read request in order to update the active functioning setpoint on the remote device; the setpoint of the remote device cannot be set unless a reply is received.

## 11 “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.

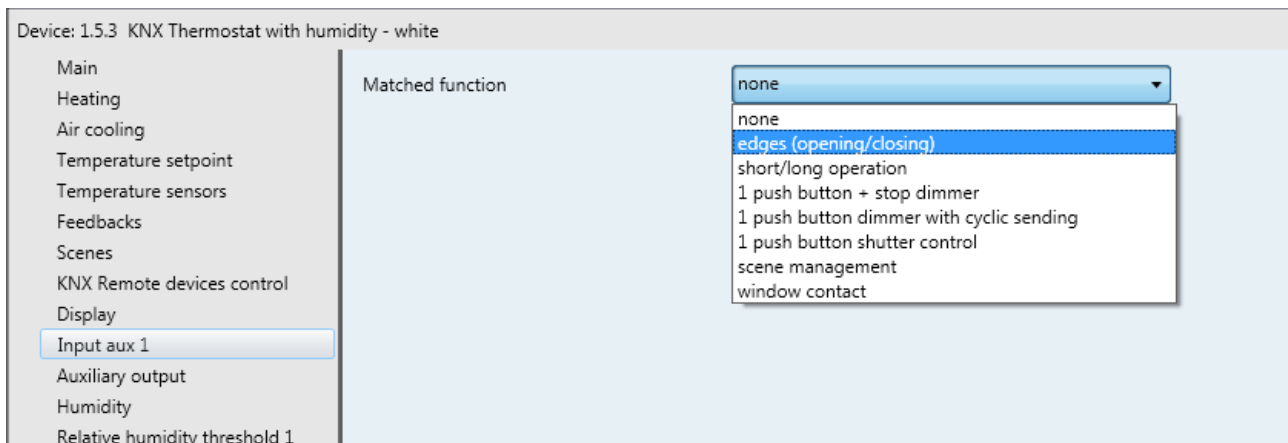


Fig. 11.1

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 takes on the values 1 and 2. The values that can be set are:

- **none** **(default value)**
- edges (closing/opening)  
(see paragraph 11.1 “Edges (closing/opening)” function)
- short/long operation  
(see paragraph 11.2 “Short/long operation” function)
- 1 push-button + stop dimmer  
(see paragraph 11.3 “1 push-button + stop dimmer” function)
- cyclic sending 1 push-button dimmer  
(see paragraph 11.4 “Push-button dimmer with cyclic sending” function)
- 1 push-button shutter control  
(see paragraph 11.5 “1 push-button shutter control” function)
- scene management  
(see paragraph 11.6 “Scene management” function)
- window contact  
(see paragraph 11.7 “Window contact” function)

### 11.1 Edge function (closing / opening)

This function is used to set the type of command to send following a variation in the contact status (edge). It is possible to differentiate between the type of command depending on the edge that is detected (from contact open to contact closed, and vice versa). The basic structure of the menu is as follows:

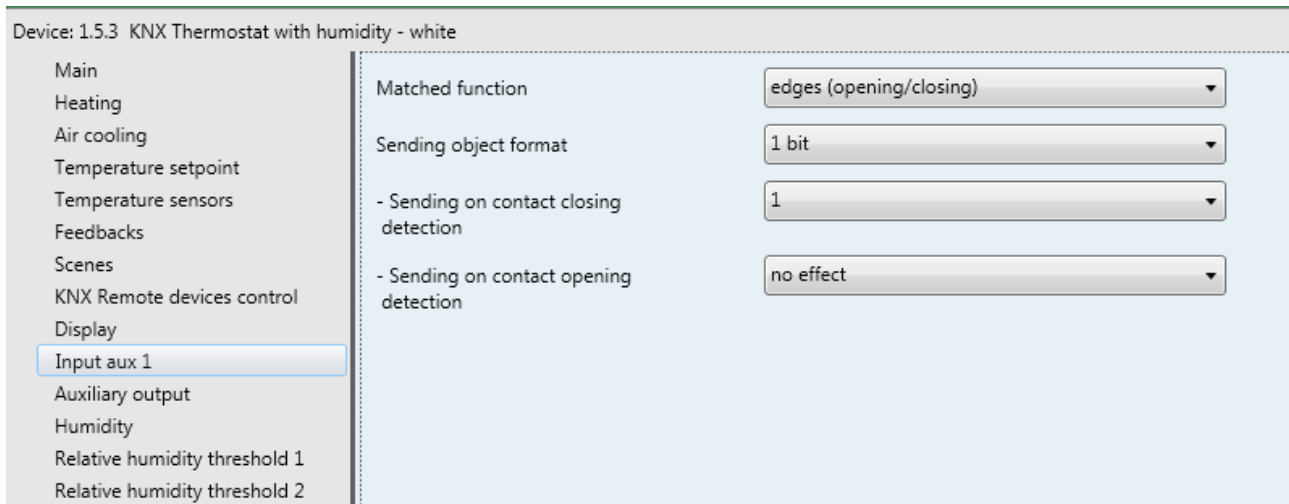


Fig. 11.1

### ➤ 11.1.1 Parameters

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

The value set for this item will cause the values set for the “**Sending on contact closing detection**” and “**Sending on contact opening detection**” parameters to change as a result.

The “**Sending on contact closing detection**” parameter is used to set the command or value to be sent following the detection of the contact closing edge (variation from open contact → closed contact).

The “**Sending on contact opening detection**” parameter is used to set the command or value to be sent following the detection of the contact opening edge (variation from contact closed → contact open).

- If the format of the object to send is **1 bit**, the **Aux input - 1 bit value** communication object will be visible (Data Point Type: 1.001 DPT\_Switch) and the values that can be set for the two parameters listed above are:

- **no effect** (opening detection default value)
- 0
- **1** (closing detection default value)
- cyclic switching

Selecting **cyclic switching** displays the “**Status feedback object**” parameter which is used to enable and display the **Aux input - 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 **Aux input - 1 bit value** when the event associated with the sending is detected will be the opposite of the value generated by the most recent event between the BUS value received on the **Aux input - status feedback** object and the last sent value (via **Aux input - 1 bit value**). The “**Status feedback object**” parameter may have the following values:

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

Selecting the value **enabled** displays the **Aux input - status feedback** communication object. In this case, every time the BUS voltage is restored you must send a status read request on this object in order to update the interface about the status of the devices connected.

- If the format of the object to send is **2 bit**, the **Aux input - 2 bit value** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control) and the values that can be set for the two parameters listed above are:

- **no effect** (default opening value)
- **active forcing on (down)** (default closing value)
- active forcing off (up)
- deactivate forcing [=forcing deactivation]
- forcing on/forcing off cyclical switching
- forcing on/forcing deactivation cyclical switching
- forcing off/forcing deactivation cyclical switching

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

- If the format of the object to send is **1 byte unsigned value**, the **Aux input - 1 byte value** 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 effect** (default opening value)
- **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, in steps of 1

- If the format of the object to send is **1 byte signed value**, the **Aux input - 1 byte value** 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 effect** (default opening value)
- **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 in steps of 1 (default value 0)

- If the format of the object to send is **1 byte percentage value**, the **Aux input - 1 byte value** 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 effect** (default opening value)
- **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, in steps of 1

- If the format of the object to send is **1 byte HVAC mode**, the **Aux input - 1 byte value** 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 effect** (default opening value)
- auto
- **comfort** (default closure 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) 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) 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 value**, the **Aux input 2 byte value** 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 effect** (opening detection default value)
- **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, in steps of 1

- If the format of the object to send is **2 byte signed value**, the **Aux input 2 byte value** 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 effect** (opening detection default value)
- **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 in steps of 1      **(default value 0)**

- If the format of the object to send is **4 byte unsigned value**, the **Aux input 4 byte value** 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 effect**                                      **(opening detection default value)**
- **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, in steps of 1

- If the format of the object to send is **4 byte signed value**, the **Aux input 4 byte value** 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 effect**                                      **(opening detection default value)**
- **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 in steps of 1      **(default value 0)**

- If the format of the object to send is **14 byte HVAC mode**, the **Aux input 14 byte value** 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 effect**                                      **(opening detection default value)**
- **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

## 11.2 Short / long operation

This function is used to set the type of command to send after detecting short or long pressing; it is possible to differentiate the command depending on the event that is detected (short or long pressing). The basic structure of the menu is as follows:

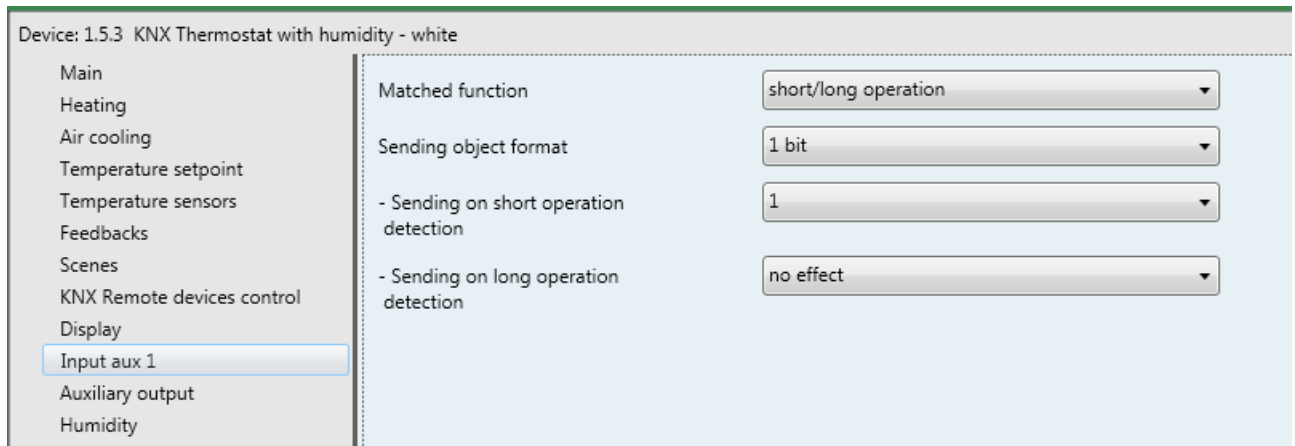


Fig. 11.2

If the auxiliary contact is closed for less than 0.5 seconds, a short operation is recognised and the relative command is set on the KNX BUS; if the auxiliary contact is closed for a period of time longer than or equal to 0.5 seconds, a long operation is recognised and the relative command is set on the KNX BUS.

### ➤ 11.2.1 Parameters

The “**Sending object format**” parameter makes it possible to set the format and code of the BUS telegram that will be sent by the device after detecting a short or long operation 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

The value set for this item will cause the values set for the “**Sending on short operation detection**” and “**Sending on long operation detection**” parameters to change as a result.

The “**Sending on short operation detection**” parameter is used to send the command or value to be sent when detecting the short operation of the contact (contact closure < 0.5 seconds).

The “**Sending on long operation detection**” parameter is used to send the command or value to be sent when detecting the long operation of the contact (contact closure ≥ 0.5 seconds).

- If the format of the object to send is **1 bit**, the **Aux input - 1 bit value** communication object will be visible (Data Point Type: 1.001 DPT\_Switch) and the values that can be set for the two parameters listed above are:
  - **no effect** (long operation detection default value)
  - 0
  - **1** (short operation detection default value)
  - cyclical switching



Selecting **cyclical switching** displays the “**Status feedback object**” parameter which is used to enable and display the **Aux input - 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 **Aux input - 1 bit value** when the event associated with the sending is detected will be the opposite of the value generated by the most recent event between the BUS value received on the **Aux input - status feedback** object and the last sent value (via **Aux input - 1 bit value**). The “**Status feedback object**” may have the following values:

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

Selecting the value **enabled** displays the **Aux input - status feedback** communication object. In this case, every time the BUS voltage is restored you must send a status read request on this object in order to update the interface about the status of the devices connected.

- If the format of the object to send is **2 bit**, the **Aux input - 2 bit value** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control) and the values that can be set for the two parameters listed above are:

- **no effect** (long operation default value)
- **activate forcing on(down)** (short operation default value)
- activate forcing off (up)
- deactivate forcing
- forcing on/ forcing off cyclical switching
- forcing on/forcing deactivation cyclical switching
- forcing off/forcing deactivation cyclical switching

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

- If the format of the object to send is **1 byte unsigned value**, the **Aux input - 1 byte value** 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 effect** (long operation default value)
- **send value** (short operation default 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, in steps of 1

- If the format of the object to send is **1 byte signed value**, the **Aux input - 1 byte value** 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 effect** (long operation default value)
- **send value** (short operation default 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 in steps of 1 (**default value 0**)

- If the format of the object to send is **1 byte percentage value**, the **Aux input - 1 byte value** 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 effect** (long operation default value)
- **send value** (short operation default 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, in steps of 1

- If the format of the object to send is **1 byte HVAC mode**, the **Aux input - 1 byte value** 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 effect** (long operation default value)
- **auto**
- **comfort** (pressing default value)
- **precomfort**
- **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 the value **cyclical switching (thermostat)**, each time the associated event is detected (short pressing/long pressing) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort*→ *Precomfort*→ *Economy*→ *Off*→ *Comfort* ... By selecting the value **cyclical switching (timed thermostat)**, each time the associated event is detected (short pressing/long pressing) the device sends a new temperature adjustment mode (HVAC), following the order *Comfort*→ *Precomfort*→ *Economy*→ *Off*→ *Auto*→ *Comfort* ...

- If the format of the object to send is **2 byte unsigned value**, the **Aux input 2 byte value** 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 effect** (long operation detection default value)
- **send value** (short operation 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, in steps of 1

- If the format of the object to send is **2 byte signed value**, the **Aux input 2 byte value** 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 effect** (long operation detection default value)
- **send value** (short operation 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 in steps of 1 (default value 0)

- If the format of the object to send is **4 byte unsigned value**, the **Aux input 4 byte value** 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 effect** (long operation detection default value)
- **send value** (short operation 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, in steps of 1

- If the format of the object to send is **4 byte signed value**, the **Aux input 4 byte value** 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 effect** (long operation detection default value)
- **send value** (short operation 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 in steps of 1 (**default value 0**)

- If the format of the object to send is **14 byte HVAC mode**, the **Aux input 14 byte value** 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 effect** (long operation detection default value)
- **send value** (short operation 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

### 11.3 1 push button + stop dimmer

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

For sending on/off telegrams and brightness control telegrams.

As there is only one contact to manage the On/Off and brightness control functions, the operation is managed by differentiating between short operations and long operations:

- a long operation (contact closure > 0.5 seconds) is transformed into 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 (contact closure < 0.5 seconds) is transformed into 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 **Input aux X - switching** (Data Point Type: 1.001 DPT\_Switch) and **Input aux X - Brightness control** (Data Point Type: 3.007 DPT\_Control\_Dimming).

The structure of the menu is as follows:

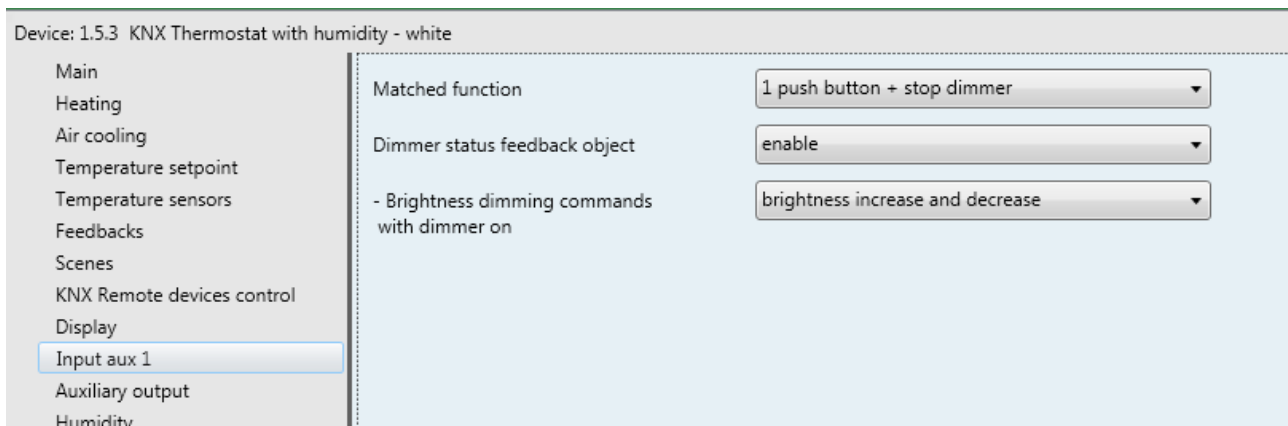


Fig. 11.3

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 **Aux input - Dimmer status feedback** communication object (Data Point Type: 1.001 DPT\_Switch), via the “**Dimmer status feedback object**” parameter.

### ➤ 11.3.1 Parameters

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

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

If **enabled** is selected, the “**Brightness control commands with dimmer on**” parameter is visualised, along with the **Aux input - Dimmer status feedback** communication object 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 “**Brightness control commands with dimmer on**” parameter, which can assume the following values:
  - only brightness increase
  - only brightness decrease
  - **brightness 2 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 button pad about the status of the devices connected.

## 11.4 Cyclic sending 1 push button dimmer

This is used to configure the auxiliary input to control a dimmer with a single button, increasing and decreasing dimmer brightness always using the same button, with defined and settable control steps.

As there is only channel to manage the On/Off and brightness control functions, the operation is managed in the following way: with each operation, the command sent is the opposite of the last one sent. Furthermore, a differentiation is made between short operations and long operations:

- a long operation (contact closure > 0.5 seconds) is transformed into a brightness control command. No telegram is sent when the contact is opened.
- a short operation (contact closure < 0.5 seconds) is transformed into an on/off command.

Unlike the function **1 push button + stop dimmer**, it is possible to define both the brightness various steps as well as the time that passes between sending different commands, if the long operation continues over time; therefore the sending of the control stop telegram is not necessary when releasing the button, even though the control follows the power/brightness characteristic curve, as it is the command that is sent from the push-button panel that determines the percentage variation. The communication objects enabled by this function are **Aux input - switching** (Data Point Type: 1.001 DPT\_Switch) and **Aux input - Brightness control** (Data Point Type: 3.007 DPT\_Control\_Dimming).

The structure of the menu is as follows:

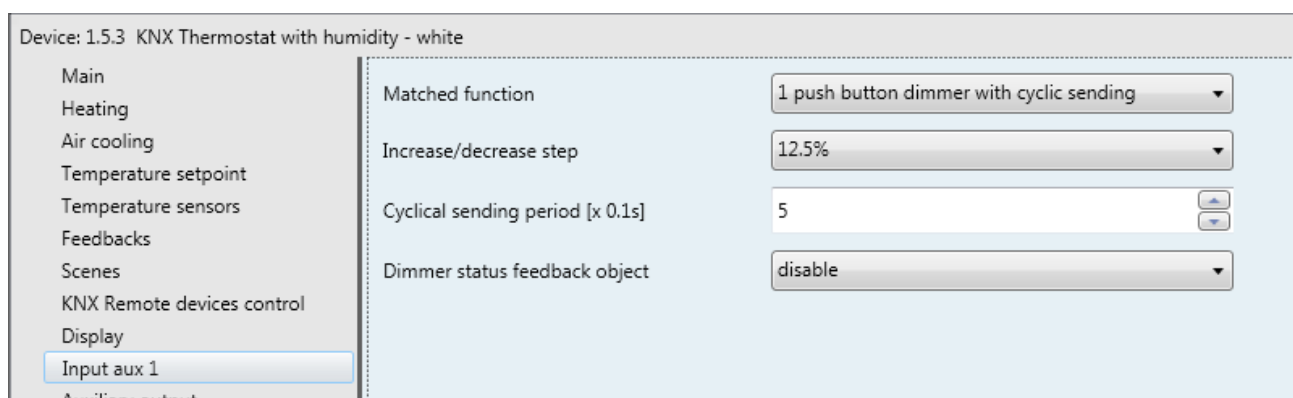


Fig. 11.4

### ➤ 11.4.1 Parameters

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 with a closed status, the device sends the command cyclically until it detects the release; the “**Cyclical sending period [x 0.1s]**” parameter defines the time that passes between sending an increase/decrease command and the other if the closure is maintained. When the contact is opened, no telegram is sent but only the cyclical sending of the brightness control commands is stopped.

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

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

To sum up, 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 **Increase/decrease step** is set to **12.5%** and the **Cyclical sending period [x 0.1s]** parameter is set to **3** (0.3 sec) and operation 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 a new 12.5% brightness increase/decrease command until the opening of the contact is detected
- when the contact is opened, no telegram is sent but the cyclical sending is 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 11.3 “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

Selecting **enable** displays the “**Brightness control commands with dimmer on**” parameter and the **Aux input - 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 button pad about the status of the devices connected.

## 11.5 1 push button shutter control

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

As only one channel manages the louvres up/down and control functions, the operation is managed so that at each contact closure, a command is sent that is the opposite of the last movement signal received by the actuator that manages the shutter. There is a difference between short and long operations:

- a long operation (contact closure > 0.5 seconds) is transformed into an up/down movement command. The new value to be sent is the opposite of the last value sent via the **Aux input - Shutter movement** object or the movement feedback received via the **Aux input - 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 (contact closure < 0.5 seconds) is transformed into a louvre control command. The new value to be sent depends on the last value sent via the **Ch.x - Shutter movement** object or the movement feedback received via the **Ch.x - Movement feedback** object, depending on which of the two events occurred last; if the last event that occurred is "upward movement feedback reception" or "sending upward movement command", the command will be a "closing louvres control" command and vice versa. If the shutter is moving, the louvre control command will only stop the shutter up/down movement.

The communication objects enabled by this function are: **Aux input - Shutter movement** (Data Point Type: 1.008 DPT\_UpDown) used to send the actuator up/down movement commands, **Aux input - Louvre stop/control** (Data Point Type: 1.007 DPT\_Step) to stop the movement in progress or to adjust the actuator louvres and **Aux input - Movement feedback** (Data Point Type: 1.008 DPT\_UpDown) used to receive feedback about the direction of the movement in progress.

The structure of the menu is as follows:

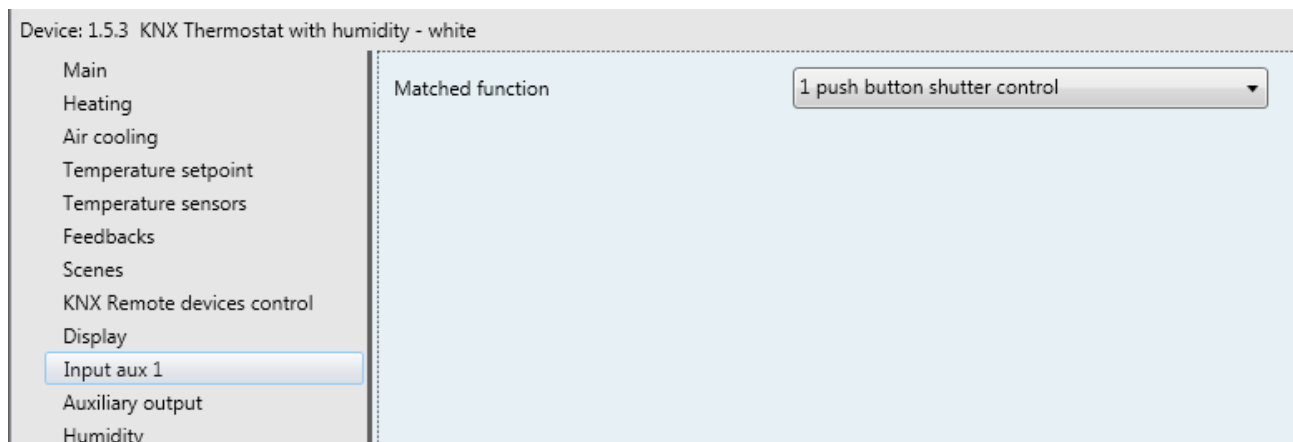


Fig. 11.5

No new parameters are enabled with this function.

## 11.6 Scene management

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

There is a difference between short and long operations:

- a long operation (contact closure > 3 seconds) is transformed into a scene storing command.
- a short operation (contact closure < 3 seconds) is transformed into a scene execution command.

The communication objects enabled by this function are **Aux input - Scene** (Data Point Type: 18.001 DPT\_SceneControl) and **Aux input - Scene storing trigger** (Data Point Type: 1.017 DPT\_Trigger).

The structure of the menu is as follows:

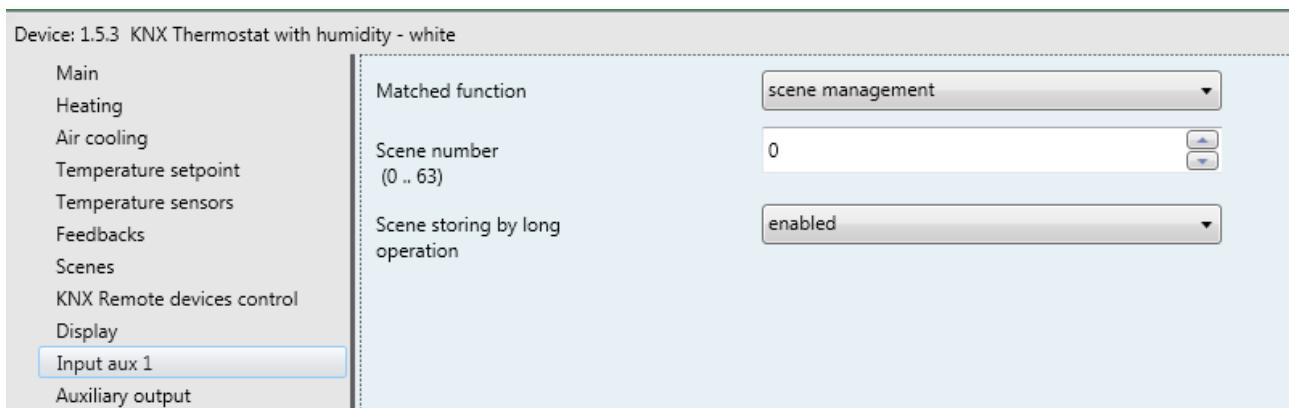


Fig. 11.6

### ➤ 11.6.1 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 **Aux input - Scene** object. The possible values are:

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

The **Scene storing by long operation** parameter 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. By selecting **disabled**, the long operation will not be recognised and the long closing of the contact will cause the sending of the scene execution command (as for the short operation). Regardless 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 **Aux input - Scene storing trigger** object (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.



## 11.7 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:

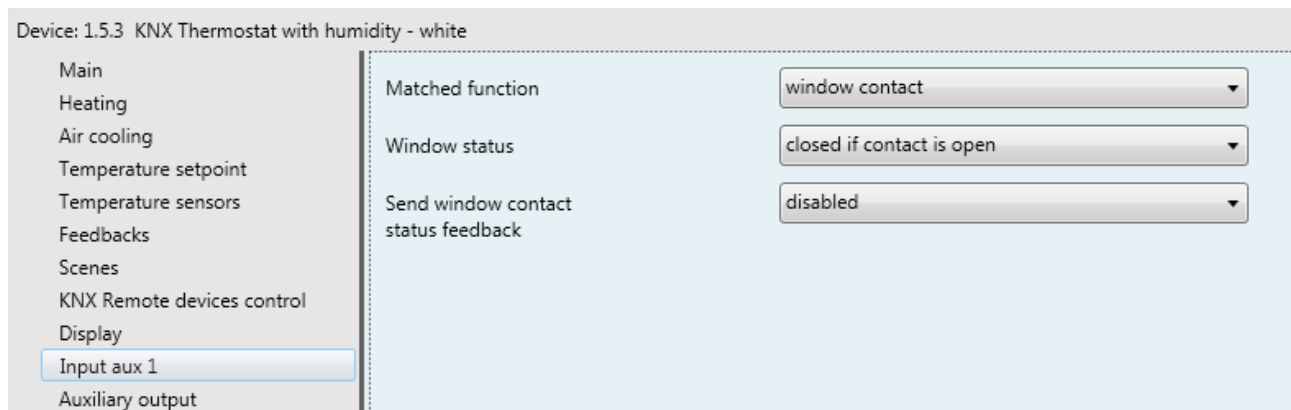


Fig. 11.7

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.

### ➤ 11.7.1 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 - 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 variation** (default value)

Selecting any value other than **disabled** displays the **Aux input - 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.

## 12 “Auxiliary inputs 1/2” 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 1/2** menu is displayed if the “**Input contacts function**” parameter in the **Main** menu has the value **two auxiliary inputs combined**.

This is the menu that appears in ETS:

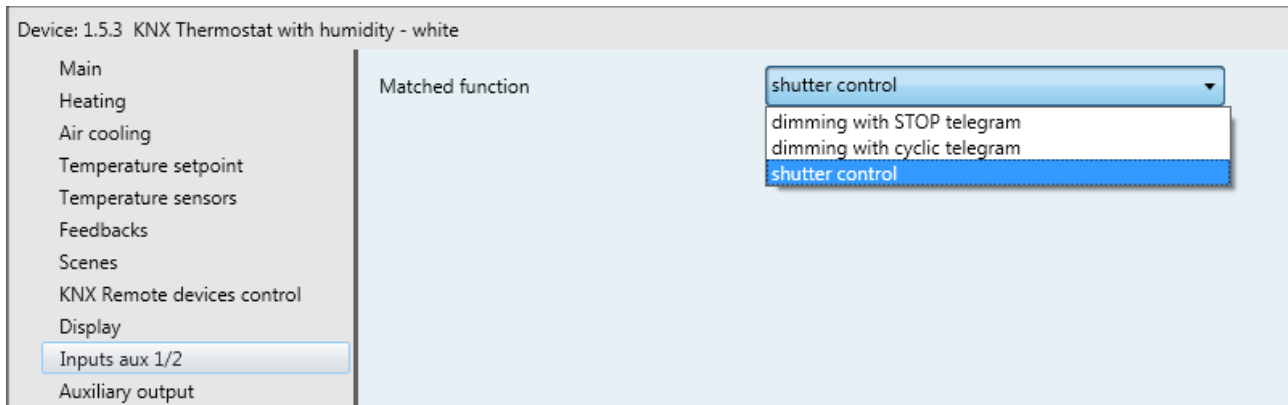


Fig. 12.1

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

- **dimming with STOP telegram** (default value)  
(see paragraph 12.1)
- dimming with cyclic telegram  
(see paragraph 12.2)
- shutter control  
(see paragraph 12.3)

### 12.1 “Dimming with STOP telegram” function

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

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

- a long operation (contact closure > 0.5 seconds) is transformed into a brightness control command. If this type of operation is recognised on auxiliary input 1, the device will send a command to increase brightness; on the other hand, if the operation is recognised on auxiliary input 2, the device will send a command to decrease brightness. In both cases, when release is detected, 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 (contact closure < 0.5 seconds) is transformed into an on/off command. If this type of operation is recognised on auxiliary input 1, the device will send a switching on command; on the other hand, if the operation is recognised on auxiliary input 2, the device will send a switching 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 **Input aux 1/2 - Switching** and **Input aux 1/2 - Brightness control**.

The structure of the menu is as follows:

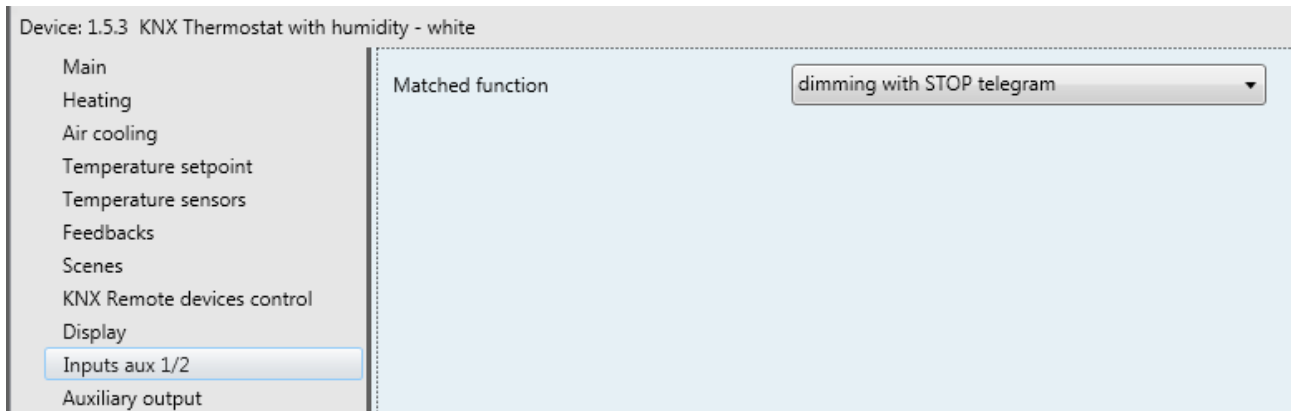


Fig. 12.2

No new parameters are enabled with this function.

## 12.2 “Dimming with cyclic telegram” function

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

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

- a long operation (contact closure > 0.5 seconds) is transformed into a brightness control command. If this type of operation is recognised on auxiliary input 1, the device will send a command to increase brightness; on the other hand, if the operation is recognised on auxiliary input 2, the device will send a command to decrease brightness. No telegram is sent when released.
- a short operation (contact closure < 0.5 seconds) is transformed into an on/off command. If this type of operation is recognised on auxiliary input 1, the device will send a switching on command; on the other hand, if the operation is recognised on auxiliary input 2, the device will send a switching off command.

Unlike the **dimming with telegram + STOP** 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. In this way, the sending of the control stop telegram is not necessary when releasing the push-button, even though the control follows the power/brightness characteristic curve, as it is the command that is sent from the touch push-button panel that determines the percentage variation. The communication objects enabled by this function are **Input aux 1/2 - Switching** and **Input aux 1/2 - Brightness control**.

The structure of the menu is as follows:

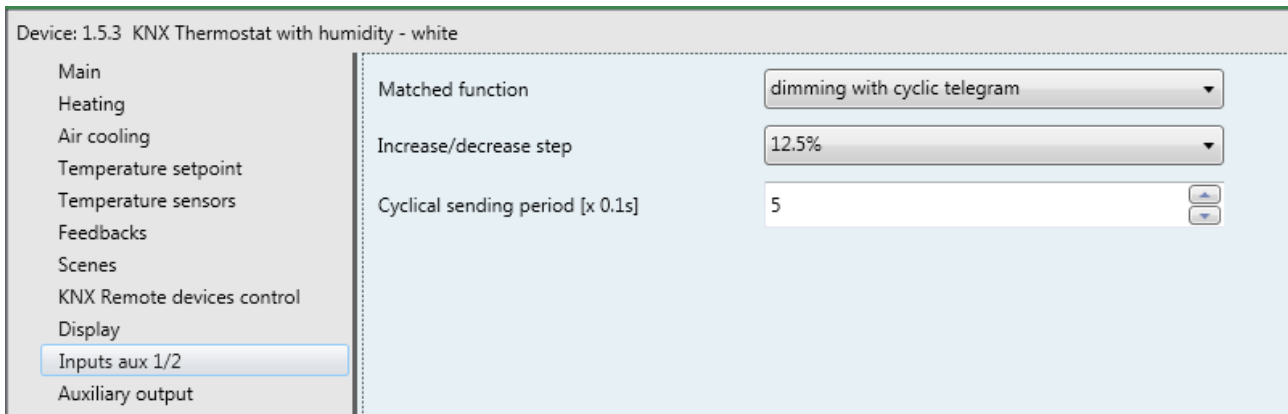


Fig. 12.3

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

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

If pressing is maintained, the device will cyclically send the command until release is detected; the “**Cyclical sending period [x 0.1s]**” parameter defines the time that passes between sending an increase/decrease command and the other if the pressure is maintained. When released, no telegram is sent but only the cyclical sending of the brightness control commands is stopped.

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

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

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

EXAMPLE: if 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 pressure is detected on the push-button:

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

### 12.3 "Shutter control" function

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

For sending up/down telegrams and louvres adjustment telegrams.

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

- a long operation (contact closure > 0.5 seconds) is transformed into an up/down command. If this type of operation is recognised on auxiliary input 1, the device will send an up command; otherwise, the device will send a down command. When released, the device will not perform any action.
- a short operation (contact closure < 0.5 seconds) is transformed into a louvres adjustment command. If this type of operation is recognised on auxiliary input 1, the device will send an opening louvres adjustment command; otherwise, the device will send a closing louvres adjustment command. If the shutter is moving, the louvre control command will only stop the shutter up/down movement; the louvres control is carried out when the shutter is stationary.

The communication objects enabled by this function are **Input aux 1/2 - Shutter movement** and **Input aux 1/2 - Louvres stop/control**.

The structure of the menu is as follows:

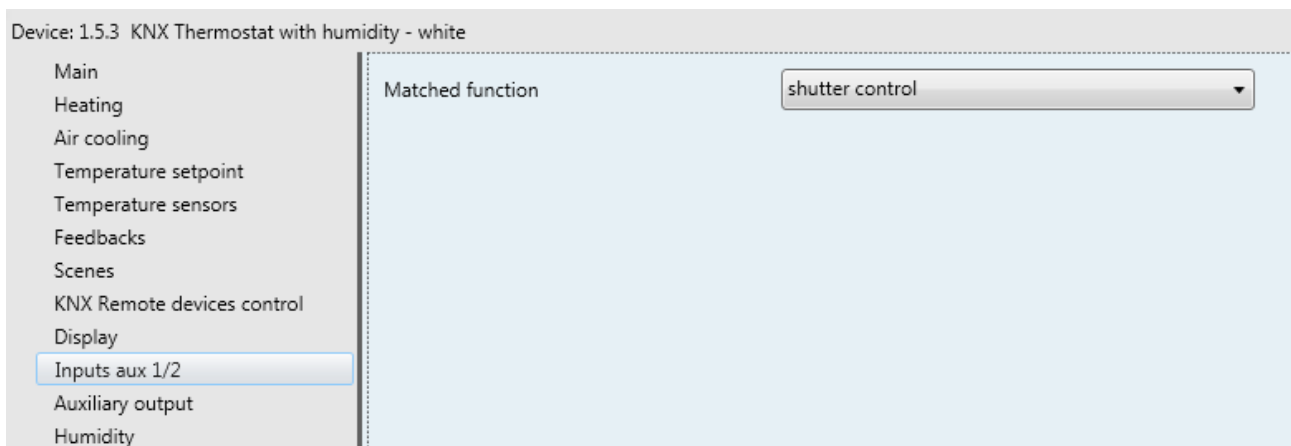


Fig. 12.4

No new parameters are enabled with this function.

## 13 "Auxiliary output" menu

The device is equipped with a relay with a change-over contact that can be used to control the heating/air cooling solenoid valve of the thermostat or can be freely commanded via the BUS with the possibility to associate at least one time profile with it.

The **Auxiliary output** menu contains the parameters used to set the function of the auxiliary output.

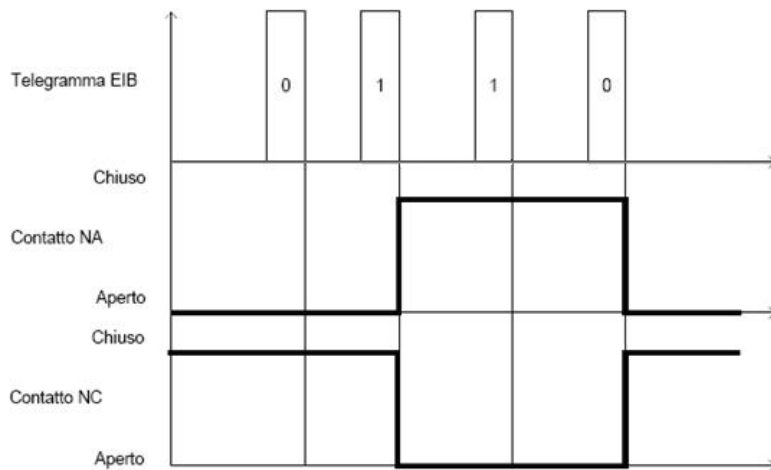
Device: 1.5.3 KNX Thermostat with humidity - white

Main	Status information transmission	sending on change
Heating	- Status transmission on bus voltage recovery	enable
Air cooling	Relay status at bus voltage failure	no change
Temperature setpoint	Relay status after bus voltage recovery	as before voltage drop
Temperature sensors	Stairs light	
Feedbacks	- Activation time	1 min
Scenes	- Warning time	no warning
KNX Remote devices control	Forced positioning	
Display	- Relay status on forced positioning end	follows last command received
Inputs aux 1/2	- Forced positioning status at bus voltage recovery	as before voltage drop
<b>Auxiliary output</b>		
Humidity		
Relative humidity threshold 1		
Relative humidity threshold 2		
Relative humidity threshold 3		
Relative humidity threshold 4		
Relative humidity threshold 5		
Dew point		
Thermal comfort		

Fig. 13.1

One of the auxiliary output operating modes is on/off switching, which involves switching the relay status according to the received commands; via the BUS, it is possible to control this operating mode via the **Aux output - Switching** communication object (Data Point Type: 1.001 DPT\_Switch).

When the device receives a telegram via the BUS on the **Aux output - Switching** object with a logical value equal to "1", it switches the relay with change-over contact to the status → NO contact closed/NC contact open; vice versa, when the logical value of "0" is received, the device switches the change-over contact to the conditions → NO contact open/NC contact closed. See figure below.



The auxiliary output implements the scene function, which is used to replicate a certain previously memorised status of the contact upon receipt of the scene execution command. The **Aux output - Scene** communication object (Data Point Type: 18.001 DPT\_SceneControl) is used to receive the scene execution and memorisation commands via the BUS. The device can manage a maximum of 8 scenes, with an index between 0 and 7.

## 13.1 Parameters

### ➤ 13.1.1 Status information transmission

The status of the on board relay and as a result the connected load can be transmitted on the BUS via the switching object, regardless of the output function. The parameter used to enable the transmission of this information is "**Status information transmission**", which can have the following values:

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

Selecting any value other than **disabled** displays the **Aux output - Status** communication object (Data Point Type: 1.001 DPT\_Switch), which is used to transmit via BUS the status information concerning the load connected to the device.

If the status signalling takes place **on variation**, the communication object is sent spontaneously when the status switches from ON to OFF 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 assumes the value 1 = ON when the NO contact (normally open) is closed and the NC contact (normally closed) is open, and the value 0 = OFF when the NO contact is open and the NC contact is closed.

When the **on variation** value is selected, the "**Status transmission on BUS voltage recovery**" parameter is visualised; this enables the transmission of the load status information when the BUS voltage is restored. The parameter may have the following values:

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

Selecting **enable**, the status feedback is sent when the device is turned on after BUS voltage recovery.

### ➤ 13.1.2 Stair light

The actuator output also implements the operating mode "activation time" (stair light) which permits the device to activate the load for a certain period of time, before deactivating it independently. The **Aux output - Timed switchover** communication object (Data Point Type: 1.010 DPT\_Start) is used to receive via BUS the timed activation start commands (value "1") and timing stop (value "0"); a timing start command with active timing will reset the activation time count.

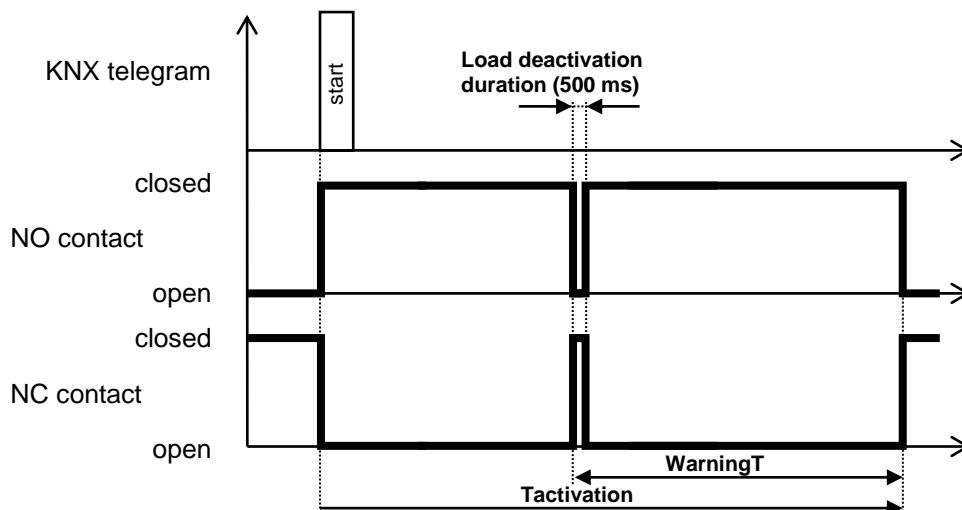
The "**Stairs light function activation time**" parameter is used to set the load activation time (Tatt). The values that can be set are:

- 1 s, 2 s, 3 s, 5 s, 10 s, 15 s, 20 s, 30 s, 45 s, **1 min (default value)**, 1 min 15 s, 1 min 30 s, 2 min, 2 min 30 s, 3 min, 5 min, 15 min, 20 min, 30 min, 1 h, 2 h, 3 h, 5 h, 12 h, 24 h.

The "**Warning time**" parameter can be used to enable signalling when the load is about to be automatically switched off, by deactivating and reactivating the load (blinking) for a very brief time (500 ms). The warning time is applied after the end of the activation time. The possible values are:

- **no warning (default value)**, 15 s, 30 s, 1 min.

The following figure shows the operating principle of the stairs light operation, with the various parameters listed above.



### ➤ 13.1.3 Forcing

The forcing function (priority commands) is used, based on the command received from the BUS, to force the relay contact to a determined condition until a forcing deactivation command is received; any command that is received during the period in which forcing is active will not be performed, as forcing has greater priority than any other BUS command.

The **Aux output - Priority command** communication object (Data Point Type: 2.001 DPT\_Switch\_Control) is used to receive via BUS the forcing deactivation or on/off forcing activation commands.

The semantics of the command received from the BUS follows what is shown in the following table:

bit1	bit 0	
0	0	Forcing deactivation
0	1	Forcing deactivation
1	0	Forcing OFF
1	1	Forcing ON



When receiving the priority command with the forcing ON activation value, the device switches the relay, closing the NO contact and opening the NC contact; vice versa, when receiving the priority command with the forcing OFF activation value, it switches the relay, opening the NO contact and closing the NC contact.

Upon receipt of the forcing deactivation command, the status to which the actuator switches the relay is defined by the “**Relay status after forcing**” parameter. The possible values are:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)
- No change
- **follows last command received (default value)**
- as prior to the forcing activation

If the parameter assumes the value **follows last command received**, the device follows the dynamics determined by the last command as if the execution of the command was initiated at the moment in which it was effectively received. Essentially, the command is executed in the background and is applied to the output in the moment in which forcing is ended. This behaviour applies, for example, to timed actuation commands with timing that has a duration that goes beyond the moment of forcing deactivation.

The “**Forcing status after BUS voltage recovery**” parameter determines the status of the forcing function on BUS voltage recovery. This parameter is useful if the function is active when the BUS voltage drops and you want to have the auxiliary output behaviour not be changed after voltage failure. The parameter may assume the following values:

- deactivated
- **as before voltage drop (default value)**

If the value **deactivated** is selected (and forcing was activated before the BUS voltage drop), when the BUS voltage is recovered the forcing function will be deactivated and the relay will take on the value determined by the “**Relay status after forcing**” parameter. If the value set for this last parameter is **follows last command received**, the actuator will execute the last command received before the BUS voltage drop that, as a result, must be stored to the non-volatile memory. If the last command received before voltage drop is a timed activation command, when the BUS voltage is recovered the command will not be executed and the relay will switch to the open (with NO)/closed (with NC) status.

If the value **as before voltage drop** is selected (and forcing was activated before BUS voltage drop), when the BUS voltage is recovered the forcing function is reactivated and the relay switches to the status prior to the voltage drop.

#### ➤ **13.1.4 Relay status at BUS voltage failure**

It is possible to define the status of the relay change-over contact (independently of the output function) following a BUS voltage drop with the front part inserted (no battery) via the “**Relay status at BUS voltage drop**” parameter, which can assume the following values:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)
- **no change (default value)**

#### ➤ **13.1.5 Relay status after bus voltage recovery**

It is possible to define the status of the relay change-over contact of the auxiliary output on BUS voltage recovery via the “**Relay status after bus voltage recovery**” parameter, which can assume the following values:

- open (with NO)/closed (with NC)
- closed (with NO)/open (with NC)
- **as before voltage drop (default value)**

When the BUS voltage is restored, the relay behaves as indicated in the **“Relay status after bus voltage recovery”** parameter if the output is configured as “free use”. Otherwise, the thermostat control algorithm is used.

## 14 "Display" menu

The structure of the menu is as follows:

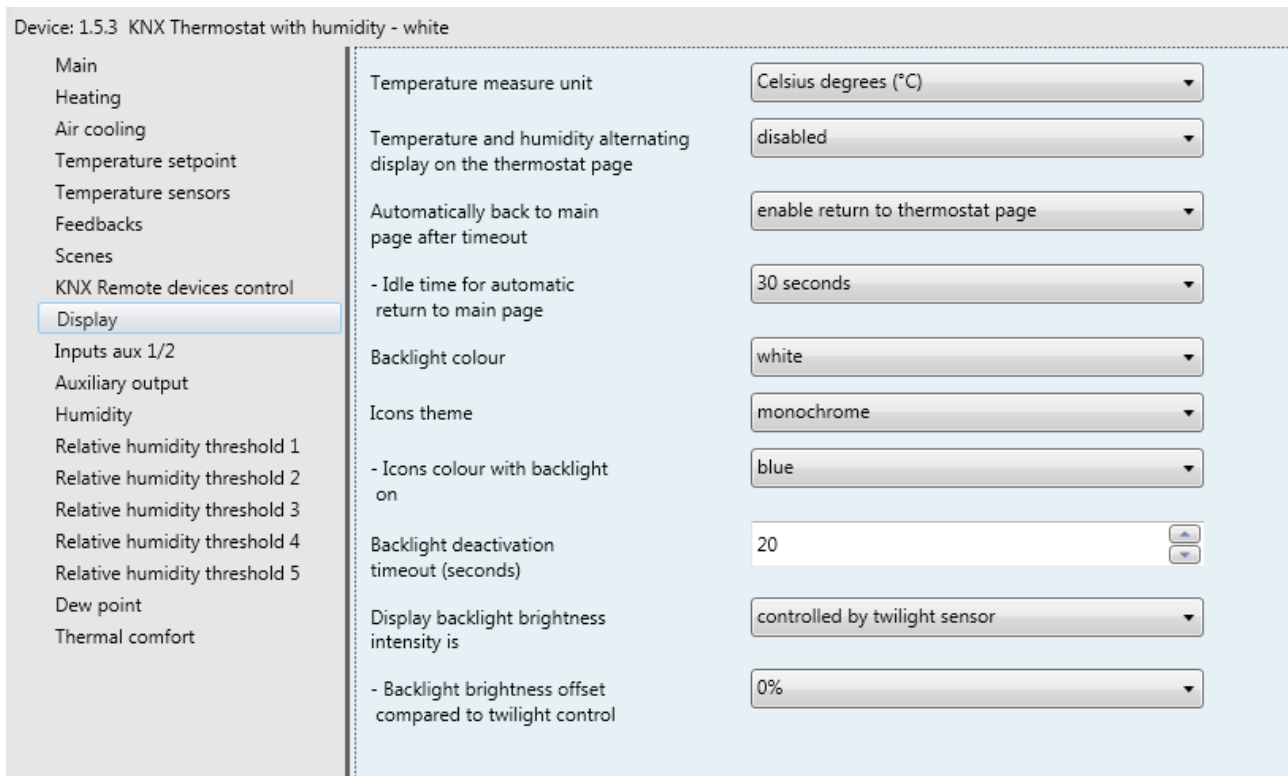


Fig. 14.1

### 14.1 Parameters

#### ➤ 14.1.1 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)

### ➤ 14.1.2 Temperature and humidity alternating display on the thermostat page

For the sake of convenience, it is possible to see the measured relative humidity value directly on the thermostat page. The value alternates with the measured temperature value. The “**Temperature and humidity alternating display on the thermostat page**” parameter is used to enable and define the time that the relative humidity value and the measured temperature value remain visible. The values that can be set are:

- **disabled** (default value)
- relative humidity 1s / temperature 4s
- relative humidity 2s / temperature 3s

### ➤ 14.1.3 Automatically back to main page after timeout

This is used to enable automatically back to the main page and at the same time define the main page itself. The values that can be set are:

- **disabled** (default value)
- enable return to thermostat page
- enable return to device page 1
- enable return to device page 2
- enable return to humidity page

Selecting any value other than **disabled** displays the “**Inactive period for automatically back to main page**” parameter.

The “**Inactive period for automatically back to main page**” parameter is used to define the duration of the user inactivity period on the device before the page defined as "main" will be automatically displayed. The values that can be set are:

- from “5 seconds” to “120 seconds” in steps of 1 second (**default value 30 seconds**)

### ➤ 14.1.4 Backlight colour

The display has back-lighting and this parameter is used to define the colour of the display's back-lighting (background). The values that can be set are:

- **white** (default value)
- yellow
- magenta
- red
- turquoise
- green
- blue
- red/blue
- no backlight

Selecting any value other than **no backlight** displays the “**Backlight deactivation timeout (seconds)**” and “**Display backlight brightness intensity is**” parameters.

Selecting **red/blue**, the icons are black and the background is white in the idle status (heating and air cooling valves deactivated), turning to red if the heating system is activated or blue if the air cooling system is activated (even before the actual system activation command).

By selecting **white**, the “**Icons theme**” parameter is also made visible.

### ➤ 14.1.5 Icons theme

If the selected backlight colour is **white** it is possible to select various colour themes with which the various icons shown on the display are represented when backlighting is on; if the backlight is not white, then the theme of the icons is MONOCHROME and BLACK. The “**Icons theme**” parameter is used to select the theme with which the icons are represented with white backlight on. The values that can be set are:

- **monochrome** (default value)
- theme 1
- theme 2
- theme 3
- theme 4
- theme 5

Selecting **monochrome** displays the “**Icons colour with backlight on**” parameter which is used to customise the colour of the icons with a monochrome theme and white backlight on; if backlighting is off, all the icons are black. The values that can be set are:

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

### ➤ 14.1.6 Backlight deactivation timeout (seconds)

The backlight turns on when any push-button is pressed, and then turns off automatically after a certain period of inactivity (energy saving). The “**Backlight deactivation timeout (seconds)**” parameter is used to set the minimum duration of the time the user is inactive on the display (pressing local push-buttons) before the backlight is automatically deactivated. When the backlight is deactivated, the icons are represented in BLACK regardless of the selected theme. The values that can be set are:

- from 10 to 180 in steps of 1 (default value 20)

### ➤ 14.1.7 Display backlight brightness intensity is

The backlight brightness intensity can be defined by the user or depend on the brightness of the environment in which the device is installed, which is measured by an internal light sensitive sensor. The “**Display backlight brightness intensity is**” parameter defines how the backlight intensity is managed. The values that can be set are:

- set manually
- **controlled by light sensitive sensor** (default value)

Selecting **set manually** displays the “**Percentage value of backlight brightness intensity**” parameter. Selecting **controlled by light sensitive sensor** displays the “**Backlight brightness offset compared to light sensitive control**” parameter.

The “**Percentage value of backlight brightness intensity**” parameter is used to select the percentage value of the brightness intensity that the display backlight must reach when turned on. The values that can be set are:

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

### ➤ 14.1.8 Backlight brightness offset compared to light sensitive control

If the backlight brightness intensity depends on the light sensitive control, the value calculated by the light sensitive control can be adjusted by adding a certain percentage contribution. The “**Backlight brightness offset compared to light sensitive control**” parameter is used to define the percentage of control contribution to the value calculated by the light sensitive sensor. The values that can be set are:

- - 10%
- **0% (default value)**
- + 10%

## 15 “Humidity” menu

The **Humidity** menu contains the parameters for configuring the operation of the external humidity sensor from which the thermostat receives the relative humidity values to be viewed and used for the various thresholds. The device does not have an external humidity sensor.

The structure of the menu is as follows:

Parameter	Value / Control
KNX external sensor monitoring time [min] (0= no monitoring)	2
Behaviour at KNX external sensor signal absence	use last value received
Relative humidity measured	do not send
Estimate the humidity at the temperature measured by the thermostat	enable
- Estimated relative humidity	do not send
Specific humidity	do not send

Fig. 15.1

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).



### ➤ 15.1.3 Estimated relative humidity and configuration parameters

Given that the relative humidity used by the device is obtained from the KNX external sensor, which can be located in a different part of the room (e.g. on the floor, in the case of underfloor cooling), it may be useful to know the relative humidity in the point where the device is located. Psychometric formulas can provide an estimate of relative humidity in the point where the device is installed. The calculation for obtaining this result is briefly explained below.

The temperature of the room in Kelvin (T) and the relative humidity ( $\varphi$ ) must be known.

Thanks to the semi-empirical correlation of Antoine, we can estimate the vapour tension, which we will call  $P_s(t)$

$$\ln P_s(t) = \frac{C_1}{T} + C_2 + C_3 \cdot T + C_4 \cdot T^2 + C_5 \cdot T^3 + C_6 \cdot T^4 + C_7 \cdot \ln(T)$$

So:

$$P_s(t) = e^{\frac{C_1}{T} + C_2 + C_3 \cdot T + C_4 \cdot T^2 + C_5 \cdot T^3 + C_6 \cdot T^4 + C_7 \cdot \ln(T)}$$

Coefficients	C1	C2	C3	C4	C5	C6	C7
T > 0°C	-5800.2206	1.3914993	-0.048640239	4.1764768x10 <sup>-5</sup>	-1.4452093x10 <sup>-8</sup>	0	6.5459673
T < 0°C	-5674.5359	6.3925247	-0.009677843	6.22115701x10 <sup>-7</sup>	2.0747825x10 <sup>-9</sup>	-9.484024x10 <sup>-13</sup>	4.1635019

Taking into account that:

$$\varphi = \frac{P_v}{P_s(t)}$$

Therefore:

$$P_v = \varphi * P_s(t)$$

The vapour pressure is constant throughout the room, as it depends on the amount of water released into the dry air.

Using the Antoine ratio again, we can obtain the vapour tension at the temperature measured, for example, on the floor  $P_s(Tr)$ . Bearing in mind that the vapour pressure has not changed (it depends on specific humidity alone), we can obtain the value of relative humidity on the floor

$$\varphi_{floor} = \frac{P_v}{P_s(Tr)}$$

The “**Estimate the relative humidity at the temperature measured by the thermostat**” parameter enables the calculation of the relative humidity at the temperature measured by the thermostat. The parameter may assume the following values:

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

Selecting **enable** displays the “**Estimated relative humidity**” parameter which defines the possible conditions for sending the value of the relative humidity estimated at the temperature measured by the thermostat. The parameter may assume the following values:

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

Selecting any value other than **do not send** displays the **Estimated relative humidity** communication object (Data Point Type: 9.007 DPT\_Value\_Humidity). Selecting **send on variation** or **send on variation and periodically** will display the “**Minimum variation for sending value [± RH]**” parameter, whereas selecting **send periodically** or **send on variation and periodically** displays the “**Estimated 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 estimated humidity value.

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

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

The “**Estimated relative humidity sending period [minutes]**” parameter, visible if the estimated relative 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 in steps of 1 **(default value 5)**

#### ➤ 15.1.4 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 variation
- send periodically
- send on variation 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** will display 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 (in relation to the last specific humidity value 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

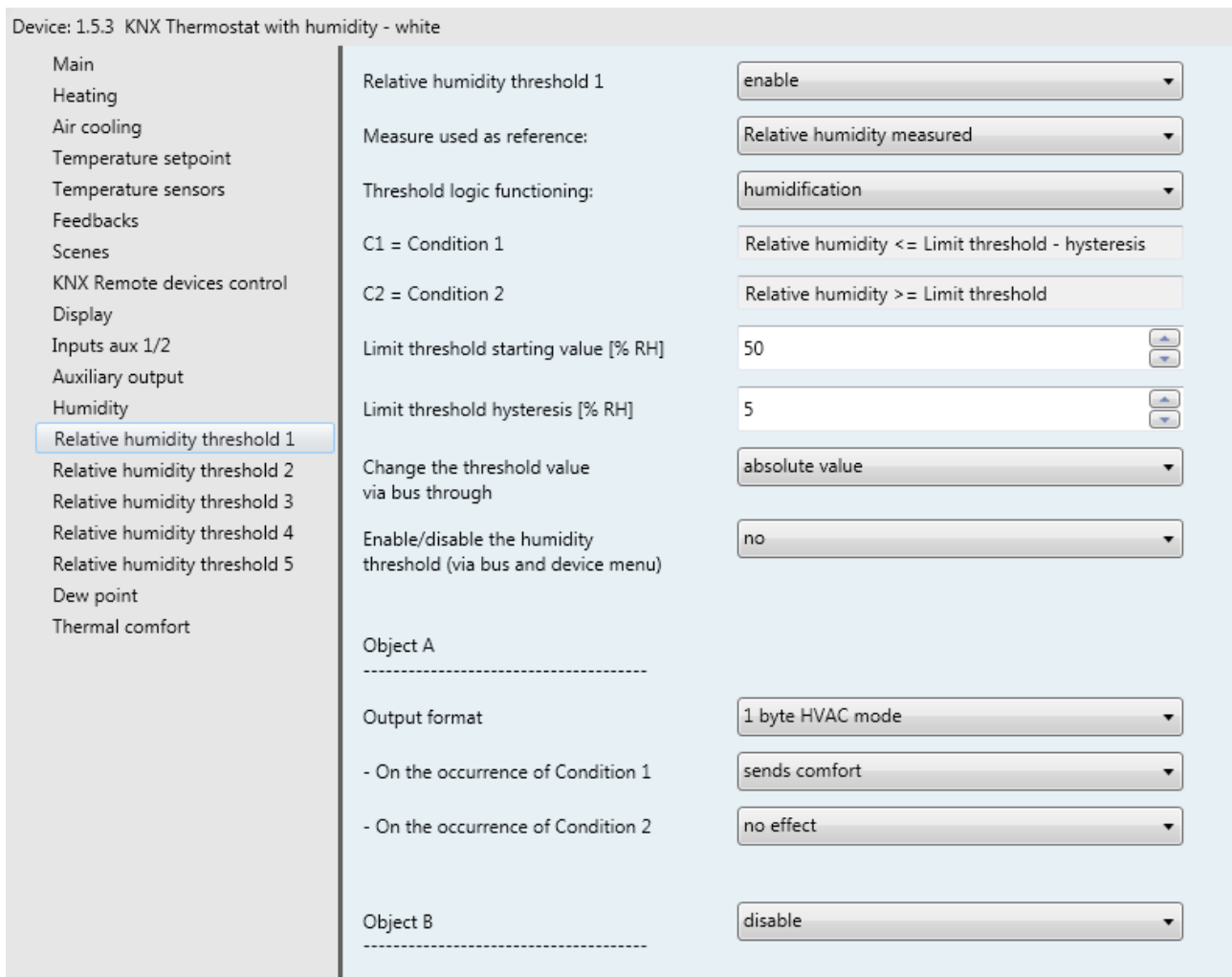
The “**Specific humidity sending period [minutes]**” parameter, visible if the specific humidity is sent periodically, is used to define the period with which the specific humidity feedback telegrams are sent spontaneously. The values that can be set are:

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

## 16 “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:



Device: 1.5.3 KNX Thermostat with humidity - white

Main	Relative humidity threshold 1	enable
Heating	Measure used as reference:	Relative humidity measured
Air cooling	Threshold logic functioning:	humidification
Temperature setpoint	C1 = Condition 1	Relative humidity <= Limit threshold - hysteresis
Temperature sensors	C2 = Condition 2	Relative humidity >= Limit threshold
Feedbacks	Limit threshold starting value [% RH]	50
Scenes	Limit threshold hysteresis [% RH]	5
KNX Remote devices control	Change the threshold value via bus through	absolute value
Display	Enable/disable the humidity threshold (via bus and device menu)	no
Inputs aux 1/2	Object A	
Auxiliary output	-----	
Humidity	Output format	1 byte HVAC mode
Relative humidity threshold 1	- On the occurrence of Condition 1	sends comfort
Relative humidity threshold 2	- On the occurrence of Condition 2	no effect
Relative humidity threshold 3	Object B	
Relative humidity threshold 4	-----	
Relative humidity threshold 5		disable
Dew point		
Thermal comfort		

Fig. 16.1

## 16.1 Parameters

### ➤ 16.1.1 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.

### ➤ 16.1.2 Measure used as reference

The “**Measure used as reference**” parameter allows you to select which of the two relative humidity measurements (measured or estimated) to use for the hysteresis check. The values that can be set are:

- **measured relative humidity** (default value)
- estimated relative humidity

if **estimated relative humidity** is selected, but this function has not been enabled in the **Humidity** menu, then the hysteresis threshold check will use the relative humidity measured.

### ➤ 16.1.3 Threshold functioning logic

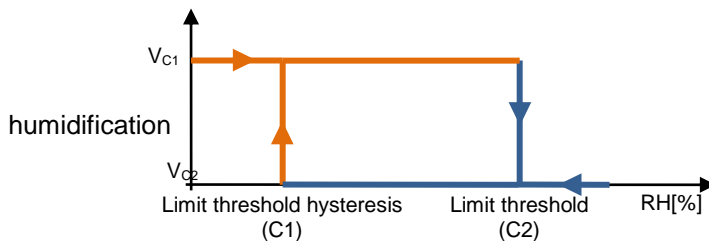
The “**Threshold operating logic**” parameter is used to define the type of hysteresis to be adopted and, as a result, the hysteresis limit values. The parameter may assume the following values:

- **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

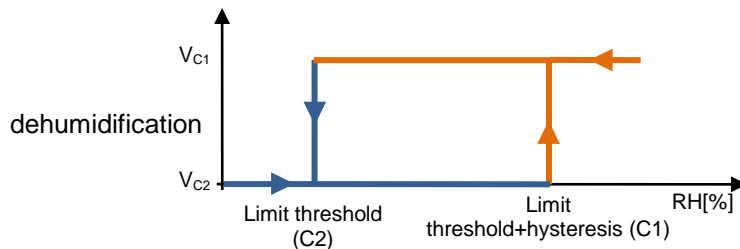


When the relative humidity of reference is lower than value C1 (limit threshold hysteresis), the device will send the command associated with Condition 1. When the relative humidity of reference reaches value C2 (limit threshold), the device will send the command associated with Condition 2.

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 send the command associated with Condition 1. When the relative humidity of reference reaches value C2 (limit threshold), the device will send the command associated with Condition 2.

### ➤ 16.1.4 Limit threshold starting value [%UR]

Via the “**Limit threshold starting value [%RH]**” parameter you can 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)**

### ➤ 16.1.5 Limit threshold hysteresis [%UR]

The “**Limit threshold hysteresis [%RH]**” parameter can be used to set the hysteresis value that, when added to or subtracted from the limit threshold, contributes towards defining the second limit value for sending the commands. This parameter may have the values:

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

### ➤ 16.1.6 Change the threshold value via bus through

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

- **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 in steps of 1                      **(default value 5)**

### ➤ 16.1.7 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 should activate the relative humidity threshold x. Receiving the value opposite to the one set for activation will deactivate the threshold. The possible values are:

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

The “**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)

### ➤ 16.1.8 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 (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** subgroup (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). By enabling this object, the command on the *Relative humidity threshold x output status feedback z* object will be repeated until the status received on the *Relative humidity threshold x output status feedback z* object coincides with the command. Possible repetition of the command will take place once a minute. The parameter may assume the following values:

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

Selecting **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, in 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 in 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, in 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”.

EXAMPLE: to switch from the “comfort” mode to “economy” mode, the offset must be “+2”; the set is not circular, therefore 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 in 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, in 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 in 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”.

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 in 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”.





## 17 “Dew point” menu

The **Dew point** menu contains parameters used to enable and configure the dew point temperature derived from the measurement of the measured relative humidity.

The structure of the menu is as follows:

Device: 1.5.3 KNX Thermostat with humidity - white

Main	Dew point temperature	do not send
Heating	Dew point temperature alarm	enable
Air cooling	Dew point temperature alarm feedback threshold	
Temperature setpoint	Alarm threshold = Dew point temperature + Feedback limit	
Temperature sensors	- Alarm feedback limit [tenths of °C]	50
Feedbacks	- Alarm feedback limit regulation step via bus [0.1°C]	5
Scenes	- Alarm threshold hysteresis [tenths of °C]	10
KNX Remote devices control	- Alarm threshold feedback measure unit	Celsius degrees (°C)
Display	- Enable/disable the alarm threshold (via bus and device menu)	no
Inputs aux 1/2	- Output format	1 bit
Auxiliary output	-- If Measure temperature < Threshold temperature	sends 1
Humidity	-- If Measure temperature > Threshold temperature + Hysteresis	no effect
Relative humidity threshold 1	- Output commands cyclical repetition	disable
Relative humidity threshold 2		
Relative humidity threshold 3		
Relative humidity threshold 4		
Relative humidity threshold 5		
<b>Dew point</b>		
Thermal comfort		

Fig. 17.1

Given a certain concentration of water in the air, defined as specific humidity  $X$ , the dew point temperature is the temperature at which saturation is reached, or where the relative humidity reaches 100%.

Specific humidity is directly linked with the dew temperature  $T_R$  via a table, so we can suppose there is a function of the  $T_R = f(X)$  type.

Once you know the coefficients of the polynomial that estimates the function, you can obtain the dew temperature starting from the specific humidity value.

In this specific case, the estimate polynomial is:

$$-21.8797827705 + 8831.8777713411 * X - 1056317.67502894 * X^2 + 81658587.5914781 * X^3 - 3592767185.52528 * X^4 + 82052448688.112 * X^5 - 753429044352.0 * X^6$$

The parameters for feedback configuration are explained below.

## 17.1 Parameters

### ➤ 17.1.1 Dew point temperature

The “**Dew point temperature**” parameter is used to define the conditions for sending the dew point temperature value estimated based on the measured relative humidity. The values that can be set are:

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

selecting any value other than **do not send** displays the **Dew point temperature** communication object and the “**Feedback measure unit**” parameter. Selecting the value **send on variation** or **send on variation and periodically**, the “**Minimum variation for sending value [± 0.1°C]**” parameter will also be visible, whereas by selecting **send periodically** or **send on variation and periodically** the “**Feedback sending period [minutes]**” parameter will be visible.

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

The “**Feedback measure unit**” parameter is used to set the measurement unit for coding and sending the information via the **Dew point 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 **Dew point temperature** 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 variation for sending value [± 0.1°C]**” parameter, visible if the dew point temperature is sent on variation, defines the minimum temperature variation (in relation to the last temperature value sent) for generating the spontaneous sending of the new value measured. The values that can be set are:

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

The “**Sending feedback repetition time [minutes]**” parameter, visible if the dew point temperature is sent periodically, defines the time gap for spontaneously sending the measured temperature feedback telegrams. The values that can be set are:

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

### ➤ 17.1.2 Dew point temperature alarm feedback threshold

The “**Dew point temperature alarm feedback threshold**” parameter is used to enable the configuration of an alarm threshold for sending a BUS command when the fixed value is exceeded. The parameter may assume the following values:

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

Selecting **enable** displays the **Dew point temperature alarm limit regulation** (Data Point Type: 1.007 DPT\_Step), **Dew point temperature alarm feedback threshold**, **Dew point temperature alarm output** communication objects and the “**Alarm feedback limit [tenths of °C]**”, “**Alarm feedback limit regulation step via BUS [0.1°C]**”, “**Alarm threshold hysteresis [tenths of °C]**”, “**Alarm threshold feedback measure unit**”, “**Enable/disable the alarm threshold via BUS**” and “**Output format**” parameters.

The “**Alarm threshold feedback measure unit**” parameter is used to set the measurement unit for coding and sending the information via the **Dew point temperature alarm feedback threshold** communication object. The values that can be set are:

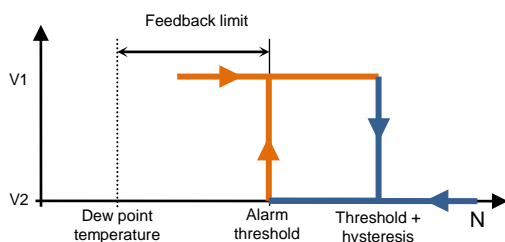
- **degrees Celsius (°C)** (default value)
- degrees Kelvin (°K)
- degrees Fahrenheit (°F)

The value set for this parameter changes the coding of the **Dew point temperature alarm feedback threshold** 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 conditions for spontaneously sending the feedback are every time the feedback limit via BUS is changed and each time the alarm threshold changes at least 0.5°C in comparison to the last value sent.

The alarm threshold is calculated as follows:

$$\text{Threshold alarm} = \text{Dew point temperature} + \text{Feedback limit}$$

This means the alarm threshold depends on two variables: the dew point temperature and the feedback limit.



The purpose of the feedback limit is to identify the approach of the dew point temperature, defining when prior to reaching the dew point temperature the command associated to the dew point temperature alarm must be sent. The initial value of the limit is defined via the “**Alarm feedback limit [tenths of °C]**” parameter, which can assume the following values:

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

The alarm feedback limit can be changed by the BUS via the **Dew point temperature alarm limit regulation** object (Data Point Type: 1.007 DPT\_Step). If the value “1” is received on this object, the limit value will be increased by the value defined by the “**Alarm feedback limit regulation step via BUS [0.1°C]**” parameter. If the value “0” is received on this object, the limit value will be decreased by the value defined by the “**Alarm feedback limit regulation step via BUS [0.1°C]**” parameter.

The “**Alarm feedback limit regulation step via BUS [0.1°C]**” parameter defines the increase/decrease step of the value of the feedback limit after receiving a command on the relative regulation object. The values that can be set are:

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

The “**Alarm threshold hysteresis [tenths of °C]**” parameter is used to set the hysteresis value that, when added to the feedback limit, contributes towards defining the dew point temperature alarm end condition and permits sending the associated command. This parameter may have the values:

- from 1 to 20 in steps of 1 (default value 10)

The “**Enable/disable alarm threshold via bus**” parameter is used to enable the possibility to activate/deactivate the dew point temperature alarm threshold 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 **Dew point temperature alarm enabling** (Data Point Type:1.002 DPT\_Boolean) and **Dew point temperature alarm 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 “**Alarm threshold enabling status when BUS voltage is restored**” parameters. The dew point temperature alarm 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 **Dew point temperature alarm enabling** communication object should activate the dew point temperature alarm threshold. Receiving the value opposite to the one set for activation will deactivate the threshold. The possible values are:

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

The “**Alarm threshold enabling status when BUS voltage is restored**” parameter is used to set the status of the dew point temperature alarm threshold after the BUS power supply voltage is restored. The possible values are:

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

The **Dew point temperature alarm enabling status** object signals the enabling status of the alarm threshold; this feedback occurs upon demand and spontaneously on variation and when BUS voltage is restored.

The “**Output format**” parameter is used to choose the format of the object to be sent in one of the two conditions. 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

Based on the value set for this item, the values that can be set for the “**If measured temperature < Threshold temperature**” and “**If measured temperature > Threshold temperature + hysteresis**” parameters will change as a result.

- If the format of the object to send is **1 bit**, the **Dew point temperature alarm output** communication object will be visible (Data Point Type: 1.001 DPT\_Switch), and the values that can be set for the two parameters listed above are:

- **no effect** (default value for  $T_{mes} > T_{threshold} + Hysteresis$ )
- 0
- 1 (default value for  $T_{mes} < T_{threshold}$ )

- If the format of the object to send is **2 bit**, the **Dew point temperature alarm output** communication object will be visible (Data Point Type: 2.001 DPT\_Switch\_Control), and the values that can be set for the two parameters listed above are:

- no effect **(default value Tmes > Tthreshold + Hysteresis)**
- Send forcing activation on (down) **(default value Tmes < Tthreshold)**
- send forcing activation off (up)
- deactivate forcing [=forcing deactivation]

- If the format of the object to send is **1 byte unsigned value**, the **Dew point temperature alarm output** 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 effect **(default value Tmes > Tthreshold + Hysteresis)**
- send value **(default value Tmes < Tthreshold)**

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, in steps of 1

- If the format of the object to send is **1 byte signed value**, the **Dew point temperature alarm output** 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 effect** **(default value Tmes > Tthreshold + Hysteresis)**
- send value **(default value Tmes < Tthreshold)**

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 in steps of 1 **(default value 0)**

- If the format of the object to send is **1 byte percentage value**, the **Dew point temperature alarm output** 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 effect **(default value Tmes > Tthreshold + Hysteresis)**
- send value **(default value Tmes < Tthreshold)**

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, in steps of 1

- If the format of the object to send is **1 byte HVAC mode**, the **Dew point temperature alarm output** 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 effect **(default value Tmes > Tthreshold + Hysteresis)**
- auto **(default value Tmes < Tthreshold)**
- comfort
- pre-comfort
- economy
- off (building protection)

- If the format of the object to send is **2 byte unsigned value**, the **Dew point temperature alarm output** 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 effect **(default value Tmes > Tthreshold + Hysteresis)**
- send value **(default value Tmes < Tthreshold)**

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, in steps of 1

- If the format of the object to send is **2 byte signed value**, the **Dew point temperature alarm output** 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 effect **(default value Tmes > Tthreshold + Hysteresis)**
- send value **(default value Tmes < Tthreshold)**

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 in steps of 1 **(default value 0)**

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 command. The values that can be set are:

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

## 18 “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: 1.5.3 KNX Thermostat with humidity - white

Main	Enable thermal comfort status feedback	enable
Heating	The season summer/winter	depends on the bus object input
Air cooling	- Logical value associated with the seasons	1 = summer / 0 = winter
Temperature setpoint	- Season active after an ETS download	summer
Temperature sensors	Limit conditions for comfort environment in the summer:	
Feedbacks	Maximum temperature [0.1 °C]	260
Scenes	Minimum temperature [0.1 °C]	240
KNX Remote devices control	Maximum relative humidity [% RH]	60
Display	Minimum relative humidity [% RH]	40
Inputs aux 1/2	Maximum specific humidity [% RH]	115
Auxiliary output	Limit conditions for comfort environment in the winter:	
Humidity	Maximum temperature [0.1 °C]	220
Relative humidity threshold 1	Minimum temperature [0.1 °C]	180
Relative humidity threshold 2	Maximum relative humidity [% RH]	60
Relative humidity threshold 3		
Relative humidity threshold 4		
Relative humidity threshold 5		
Dew point		
<b>Thermal comfort</b>		

Fig. 18.1

### 18.1 Parameters

#### ➤ 18.1.1 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.

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

To determine the thermal comfort status, it is necessary to know the current season (summer or winter); the “**Season summer/winter**” parameter is used to select the mode for setting the season. The possible values are:

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

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 input BUS object** 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.

### ➤ 18.1.3 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.

## 19 Communication objects

By enabling all the functions available, all the associated communication objects will be made visible.

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 input functions**

The following table contains all the objects with an input function.

#	Object name	Object function	Description	Datapoint type
0	Day/Hour input	Update day/hour	Receives updates about day of the week and hour of the day	10.001 DPT_TimeOfDay
1	Daylight saving time input	1= daylight saving/0= standard	Receives updates of hourly convention in course	1.001 DPT_Switch
2	Enabling slave function	Enable/Disable	Receives Slave function enabling commands	1.003 DPT_Enable
3	HVAC mode input	Eco/Precom/Comf/Off	Receives HVAC mode setting commands	20.102 DPT_HVACMode
3	Setpoint input	Value °C	Receives functioning setpoint values in degrees Celsius	9.001 DPT_Temp
3	Setpoint input	°K value	Receives functioning setpoint values in degrees Kelvin	9.002 DPT_Tempd
3	Setpoint input	°F value	Receives functioning setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F
3	Setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the functioning setpoint value	1.007 DPT_Step
4	Setpoint temporary forcing regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the setpoint temporary forcing value	1.007 DPT_Step
5	HVAC off mode input	Enable/Disable	Receives HVAC off (building protection) mode activation commands	1.003 DPT_Enable
6	HVAC economy mode input	Enable/Disable	Receives HVAC economy mode activation commands	1.003 DPT_Enable
7	HVAC precomfort mode input	Enable/Disable	Receives HVAC precomfort mode activation commands	1.003 DPT_Enable
8	HVAC comfort mode input	Enable/Disable	Receives HVAC comfort mode activation commands	1.003 DPT_Enable
9	Functioning type input	Heating/Air cooling	Receives commands of functioning type setting	1.100 DPT_Heat/Cool
10	Enable dead zone	Enable/Disable	Receives automatic (dead zone) functioning type setting enabling commands	1.003 DPT_Enable
11	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
11	Heating anti-freeze setpoint input	Value °C	Receives setpoint values heating off HVAC mode in degrees Celsius	9.001 DPT_Temp
11	Heating anti-freeze setpoint input	°K value	Receives setpoint values heating off HVAC mode in degrees Kelvin	9.002 DPT_Tempd
12	Heating anti-freeze setpoint input	°F value	Receives setpoint values heating off HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
12	Heating economy setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the heating HVAC mode economy setpoint	1.007 DPT_Step
12	Heating economy setpoint input	Value °C	Receives setpoint values heating economy HVAC mode in degrees Celsius	9.001 DPT_Temp

12	Heating economy setpoint input	°K value	Receives setpoint values heating economy HVAC mode in degrees Kelvin	9.002 DPT_Tempd
13	Heating economy setpoint input	°F value	Receives setpoint values heating economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
13	Heating precomfort setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the heating HVAC mode precomfort setpoint	1.007 DPT_Step
13	Heating precomfort setpoint input	Value °C	Receives setpoint values heating precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp
13	Heating precomfort setpoint input	°K value	Receives setpoint values heating precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
14	Heating precomfort setpoint input	°F value	Receives setpoint values heating precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
14	Heating comfort setpoint regulation	1 = Increase/0 = Decrease	Receives increase/decrease step commands of the heating HVAC mode comfort setpoint	1.007 DPT_Step
14	Heating comfort setpoint input	Value °C	Receives setpoint values heating comfort HVAC mode in degrees Celsius	9.001 DPT_Temp
14	Heating comfort setpoint input	°K value	Receives setpoint values heating comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
15	Heating comfort setpoint input	°F value	Receives setpoint values heating comfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
15	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
15	Air-cooling high temp. protection setpoint input	Value °C	Receives setpoint values air cooling off HVAC mode in degrees Celsius	9.001 DPT_Temp
15	Air-cooling high temp. protection setpoint input	°K value	Receives setpoint values air cooling off HVAC mode in degrees Kelvin	9.002 DPT_Tempd
16	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
16	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
16	Air-cooling economy setpoint input	Value °C	Receives setpoint values air cooling economy HVAC mode in degrees Celsius	9.001 DPT_Temp
16	Air-cooling economy setpoint input	°K value	Receives setpoint values air cooling economy HVAC mode in degrees Kelvin	9.002 DPT_Tempd
17	Air-cooling economy setpoint input	°F value	Receives setpoint values air cooling economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
17	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
17	Air-cooling precomfort setpoint input	Value °C	Receives setpoint values air cooling precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp
17	Air-cooling precomfort setpoint input	°K value	Receives setpoint values air cooling precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
18	Air-cooling precomfort setpoint input	°F value	Receives setpoint values air cooling precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
18	Air-cooling comfort setpoint regulation	1 = Increase/0 =	Receives increase/decrease step	1.007 DPT_Step

		Decrease	commands of the air cooling HVAC mode comfort setpoint	
18	Air-cooling comfort setpoint input	Value °C	Receives setpoint values air cooling comfort HVAC mode in degrees Celsius	9.001 DPT_Temp
18	Air-cooling comfort setpoint input	°K value	Receives setpoint values air cooling comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
18	Air-cooling comfort setpoint input	°F value	Receives setpoint values air cooling comfort HVAC mode in degrees Fahrenheit	9.002 DPT_Tempd
20	Heating valve status feedback	On/Off status	Receives feedback on electrovalve activation status 1st stage heating	1.001 DPT_Switch
20	Heating valve % feedback	% value	Receives feedback on electrovalve activation status 1st stage heating	5.001 DPT_Scaling
20	Heating/air cooling valve status feedback	On/Off status	Receives feedbacks on electrovalve activation status heating/air cooling	1.001 DPT_Switch
20	Heating/Air cooling valve % feedback	% value	Receives feedbacks on electrovalve activation status heating/air cooling	5.001 DPT_Scaling
24	Heating fancoil speed % feedback	% value	Receives feedbacks on fancoil % speed heating	5.001 DPT_Scaling
24	Heating fan V1 status feedback	On/Off status	Receives feedbacks on fancoil speed 1 activation status heating	1.001 DPT_Switch
25	Heating fan V2 status feedback	On/Off status	Receives feedbacks on heating fancoil speed 2 activation status	1.001 DPT_Switch
26	Heating fan V3 status feedback	On/Off status	Receives feedback on heating fancoil speed 3 activation status	1.001 DPT_Switch
28	Heating 2nd stage feedback	On/Off status	Receives feedbacks on electrovalve activation status 2nd stage heating	1.001 DPT_Switch
28	Heating 2° stage valve % feedback	% value	Receives feedbacks on electrovalve activation status 2nd stage heating	5.001 DPT_Scaling
30	Air cooling valve status feedback	On/Off status	Receives feedback on electrovalve activation status 1st stage air cooling	1.001 DPT_Switch
30	Air cooling valve % feedback	% value	Receives feedback on electrovalve activation status 1st stage air cooling	5.001 DPT_Scaling
34	Air cooling fancoil speed % feedback	% value	Receives feedbacks on fancoil % speed air cooling	5.001 DPT_Scaling
34	Air cooling fan V1 status feedback	On/Off status	Receives feedbacks on fancoil speed 1 activation status air cooling	1.001 DPT_Switch
35	Air cooling fan V2 status feedback	On/Off status	Receives feedbacks on air cooling fancoil speed 2 activation status	1.001 DPT_Switch
36	Air cooling fan V3 status feedback	On/Off status	Receives feedback on air cooling fancoil speed 3 activation status	1.001 DPT_Switch
38	Air cooling 2nd stage feedback	On/Off status	Receives feedback on electrovalve activation status 2nd stage air cooling	1.001 DPT_Switch
38	Air cooling 2° stage valve % feedback	% value	Receives feedback on electrovalve activation status 2nd stage air cooling	5.001 DPT_Scaling
39	Fancoil mode input	Automatic/manual	Receives commands of fancoil speed automatic mode selection or manual speed	1.001 DPT_Switch
41	KNX external sensor input	Value °C	Receives external KNX sensor values in degrees Celsius	9.001 DPT_Temp
41	KNX external sensor input	°K value	Receives external KNX sensor values in degrees Kelvin	9.002 DPT_Tempd
41	KNX floor sensor input	°F value	Receives floor external KNX	9.027 DPT_Temp_F

			sensor values in degrees Fahrenheit	
41	KNX floor sensor input	Value °C	Receives floor external KNX sensor values in degrees Celsius	9.001 DPT_Temp
41	KNX floor sensor input	°K value	Receives floor external KNX sensor values in degrees Kelvin	9.002 DPT_Tempd
41	KNX floor sensor input	°F value	Receives floor external KNX sensor values in degrees Fahrenheit	9.027 DPT_Temp_F
60	Feedbacks sending trigger	Send feedback	Receives trigger commands of feedback sending request	1.017 DPT_Trigger
61	Thermostat scene	Execute/Store	Receives thermostat function scene execution/storing commands	18.001 DPT_SceneControl
62	Input aux 1 – Dimmer status feedback	On/Off status	Receives status feedback by controlled dimmer	1.001 DPT_Switch
62	Input aux 1 – Status feedback	On/Off status	Receives status feedback by controlled actuator	1.001 DPT_Switch
62	Input aux 1 – Movement feedback	Increase/Decrease	Receives feedback of movement direction in progress by shutter actuator	1.008 DPT_UpDown
64	Input aux 1 – Scene storing trigger	Store	Receives trigger commands of scene storing commands sending request	1.017 DPT_Trigger
65	Input aux 2 – Dimmer status feedback	On/Off status	Receives status feedback by controlled dimmer	1.001 DPT_Switch
65	Input aux 2 – Status feedback	On/Off status	Receives status feedback by controlled actuator	1.001 DPT_Switch
65	Input aux 2 – Movement feedback	Increase/Decrease	Receives feedback of movement direction in progress by shutter actuator	1.008 DPT_UpDown
67	Input aux 2 – Scene storing trigger	Store	Receives trigger commands of scene storing commands sending request	1.017 DPT_Trigger
69	Output aux - Switching	On/Off	Receives output activation/deactivation commands	1.001 DPT_Switch
70	Output aux - Timed switching	Start/Stop	Receives timed activation start and exit timing stop commands	1.010 DPT_Start
71	Output aux - Priority command	On/Off forced positioning	Receives commands of on/off forced positioning and output forced positioning deactivation	2.001 DPT_Switch_Control
72	Output aux - Scene	Execute/Store	Receives commands of exit scene execution/storage	18.001 DPT_SceneControl
73	Setpoint regulation range setting	Value °C	Receives max setpoint regulation range in degrees Celsius	9.001 DPT_Temp
76	Device 1 - Functioning type status feedback	Heating/Air cooling	Receives functioning type from device 1	1.100 DPT_Heat/Cool
77	Device 1 - HVAC mode status feedback	Eco/Precom/Com/Off	Receives HVAC mode from device 1	20.102 DPT_HVACMode
77	Device 1 - Reference setpoint input	Value °C	Receives reference setpoint in degrees Celsius from device 1	9.001 DPT_Temp
77	Device 1 - Reference setpoint input	°K value	Receives reference setpoint in degrees Kelvin from device 1	9.002 DPT_Tempd
77	Device 1 - Reference setpoint input	°F value	Receives reference setpoint in degrees Fahrenheit from device 1	9.027 DPT_Temp_F
78	Device 1 - Setpoint status feedback	Value °C	Receives functioning setpoint in degrees Celsius from device 1	9.001 DPT_Temp
78	Device 1 - Setpoint status feedback	°K value	Receives functioning setpoint in degrees Kelvin from device 1	9.002 DPT_Tempd
78	Device 1 - Setpoint status feedback	°F value	Receives functioning setpoint in degrees Fahrenheit from device 1	9.027 DPT_Temp_F
79	Device 1 - Measured temperature status feedback	Value °C	Receives measured temperature in degrees Celsius from device 1	9.001 DPT_Temp
79	Device 1 - Measured temperature status feedback	°K value	Receives measured temperature in degrees Kelvin from device 1	9.002 DPT_Tempd
79	Device 1 - Measured temperature status feedback	°F value	Receives measured temperature in degrees Fahrenheit from	9.027 DPT_Temp_F

			device 1	
80	Device 1 - Measured relative humidity feedback	Value [% RH]	Receives measured relative humidity from device 1	9.007 DPT_Value_Humidity
83	Device 2 - Functioning type status feedback	Heating/Air cooling	Receives functioning type from device 2	1.100 DPT_Heat/Cool
84	Device 2 - HVAC mode status feedback	Eco/Precom/Comf/Off	Receives HVAC mode from device 2	20.102 DPT_HVACMode
84	Device 2 - Reference setpoint input	Value °C	Receives reference setpoint in degrees Celsius from device 2	9.001 DPT_Temp
84	Device 2 - Reference setpoint input	°K value	Receives reference setpoint in degrees Kelvin from device 2	9.002 DPT_Tempd
84	Device 2 - Reference setpoint input	°F value	Receives reference setpoint in degrees Fahrenheit from device 2	9.027 DPT_Temp_F
85	Device 2 - Setpoint status feedback	Value °C	Receives functioning setpoint in degrees Celsius from device 2	9.001 DPT_Temp
85	Device 2 - Setpoint status feedback	°K value	Receives functioning setpoint in degrees Kelvin from device 2	9.002 DPT_Tempd
85	Device 2 - Setpoint status feedback	°F value	Receives functioning setpoint in degrees Fahrenheit from device 2	9.027 DPT_Temp_F
86	Device 2 - Measured temperature status feedback	Value °C	Receives measured temperature in degrees Celsius from device 2	9.001 DPT_Temp
86	Device 2 - Measured temperature status feedback	°K value	Receives measured temperature in degrees Kelvin from device 2	9.002 DPT_Tempd
86	Device 2 - Measured temperature status feedback	°F value	Receives measured temperature in degrees Fahrenheit from device 2	9.027 DPT_Temp_F
87	Device 2 - Measured relative humidity feedback	Value [% RH]	Receives measured relative humidity from device 2	9.007 DPT_Value_Humidity
90	Device 3 - Functioning type status feedback	Heating/Air cooling	Receives functioning type from device 3	1.100 DPT_Heat/Cool
91	Device 3 - HVAC mode status feedback	Eco/Precom/Comf/Off	Receives HVAC mode from device 3	20.102 DPT_HVACMode
91	Device 3 - Reference setpoint input	Value °C	Receives reference setpoint in degrees Celsius from device 3	9.001 DPT_Temp
91	Device 3 - Reference setpoint input	°K value	Receives reference setpoint in degrees Kelvin from device 3	9.002 DPT_Tempd
91	Device 3 - Reference setpoint input	°F value	Receives reference setpoint in degrees Fahrenheit from device 3	9.027 DPT_Temp_F
92	Device 3 - Setpoint status feedback	Value °C	Receives functioning setpoint in degrees Celsius from device 3	9.001 DPT_Temp
92	Device 3 - Setpoint status feedback	°K value	Receives functioning setpoint in degrees Kelvin from device 3	9.002 DPT_Tempd
92	Device 3 - Setpoint status feedback	°F value	Receives functioning setpoint in degrees Fahrenheit from device 3	9.027 DPT_Temp_F
93	Device 3 - Measured temperature status feedback	Value °C	Receives measured temperature in degrees Celsius from device 3	9.001 DPT_Temp
93	Device 3 - Measured temperature status feedback	°K value	Receives measured temperature in degrees Kelvin from device 3	9.002 DPT_Tempd
93	Device 3 - Measured temperature status feedback	°F value	Receives measured temperature in degrees Fahrenheit from device 3	9.027 DPT_Temp_F
94	Device 3 - Measured relative humidity feedback	Value [% RH]	Receives measured relative humidity from device 3	9.007 DPT_Value_Humidity
97	Device 4 - Functioning type status feedback	Heating/Air cooling	Receives functioning type from device 4	1.100 DPT_Heat/Cool
98	Device 4 - HVAC mode status feedback	Eco/Precom/Comf/Off	Receives HVAC mode from device 4	20.102 DPT_HVACMode
98	Device 4 - Reference setpoint input	Value °C	Receives reference setpoint in degrees Celsius from device 4	9.001 DPT_Temp
98	Device 4 - Reference setpoint input	°K value	Receives reference setpoint in degrees Kelvin from device 4	9.002 DPT_Tempd
98	Device 4 - Reference setpoint input	°F value	Receives reference setpoint in degrees Fahrenheit from device 4	9.027 DPT_Temp_F
99	Device 4 - Setpoint status feedback	Value °C	Receives functioning setpoint in degrees Celsius from device 4	9.001 DPT_Temp
99	Device 4 - Setpoint status feedback	°K value	Receives functioning setpoint in degrees Kelvin from device 4	9.002 DPT_Tempd

99	Device 4 - Setpoint status feedback				°F value	Receives functioning setpoint in degrees Fahrenheit from device 4	9.027 DPT_Temp_F	
100	Device 4 - Measured temperature status feedback				Value °C	Receives measured temperature in degrees Celsius from device 4	9.001 DPT_Temp	
100	Device 4 - Measured temperature status feedback				°K value	Receives measured temperature in degrees Kelvin from device 4	9.002 DPT_Tempd	
100	Device 4 - Measured temperature status feedback				°F value	Receives measured temperature in degrees Fahrenheit from device 4	9.027 DPT_Temp_F	
101	Device 4 - Measured relative humidity feedback				Value [% RH]	Receives measured relative humidity from device 4	9.007 DPT_Value_Humidity	
102	Humidity external sensor input				Value [% RH]	Receives the relative humidity values from the external KNX humidity sensor	9.007 DPT_Value_Humidity	
107	Dew point temperature alarm enabling				Enable/Disable	Receives dew point temperature alarm threshold enabling commands	1.002 DPT_Boot	
109	Dew point temperature alarm limit regulation				1 = Increase/0 = Decrease	Receives increase/decrease step commands of the dew point temperature alarm limit	1.007 DPT_Step	
112	124	136	148	160	Relative humidity threshold x enabling	0=Enable/1=Disable	Receives relative humidity threshold x enabling commands	1.002 DPT_Boot
112	124	136	148	160	Relative humidity threshold x enabling	1=Enable/0=Disable	Receives relative humidity threshold x enabling commands	1.002 DPT_Boot
114	126	138	150	162	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
114	126	138	150	162	Relative humidity threshold x value input	Value [% RH]	Receives the relative humidity threshold x value	9.007 DPT_Value_Humidity
117	129	141	153	165	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
117	129	141	153	165	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
117	129	141	153	165	Output A relative humidity threshold x reference	Setpoint [°K]	Receives the reference setpoint values (°K) for the calculation of the output A associated with the relative humidity threshold x	9.002 DPT_Tempd
117	129	141	153	165	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
117	129	141	153	165	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
119	131	143	155	167	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
119	131	143	155	167	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
119	131	143	155	167	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
119	131	143	155	167	Output B relative humidity threshold x reference	Setpoint [°F]	Receives the reference setpoint values (°F) for the calculation of	9.027 DPT_Temp_F

					reference		the output B associated with the relative humidity threshold x	
119	131	143	155	167	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
121	133	145	157	169	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
121	133	145	157	169	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
121	133	145	157	169	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
121	133	145	157	169	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
121	133	145	157	169	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
123	135	147	159	171	Output D 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 D associated with the relative humidity threshold x	20.102 DPT_HVAC_Mode
123	135	147	159	171	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
123	135	147	159	171	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
123	135	147	159	171	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
123	135	147	159	171	Relative humidity threshold x output D status	On/Off status	Receives feedback about the state of activation of the output D associated with the relative humidity threshold x	1.001 DPT_Switch
172	Season selection input					1=Summer / 0=Winter	Receives current season setting for thermal comfort	1.002 DPT_Bool
172	Season selection input					1=Winter / 0=Summer	Receives current season setting for thermal comfort	1.002 DPT_Bool



## Communication objects with output functions

The following table contains all the objects with an output function.

#	Object name	Object function	Description	Datapoint type
19	Heating valve switch	On/Off	Sends heating electrovalve activation/deactivation commands 1st stage	1.001 DPT_Switch
19	Heating/Air cooling valve switch	On/Off	Sends heating/air cooling electrovalve activation/deactivation commands	1.001 DPT_Switch
19	% command valve heating	% value	Sends percentage commands to electrovalve heating 1st stage	5.001 DPT_Scaling
19	Heating/air cooling valve % command	% value	Sends percentage commands to electrovalve heating/air cooling	5.001 DPT_Scaling
21	V1 fan switching heating	On/Off	Sends heating fancoil speed 1 electrovalve activation/deactivation commands	1.001 DPT_Switch
21	% command fancoil speed heating	% value	Sends percentage commands to fancoil heating	5.001 DPT_Scaling
22	V2 fan switching heating	On/Off	Sends commands of activation/deactivation fancoil speed 2 heating	1.001 DPT_Switch
23	V3 fan switching heating	On/Off	Sends commands of activation/deactivation fancoil speed 3 heating	1.001 DPT_Switch
27	Heating 2° stage switching	On/Off	Sends heating electrovalve activation/deactivation commands 2nd stage	1.001 DPT_Switch
27	% command 2° stage heating	% value	Sends percentage commands to electrovalve heating 2nd stage	5.001 DPT_Scaling
29	Air cooling valve switch	On/Off	Sends air cooling electrovalve activation/deactivation commands 1st stage	1.001 DPT_Switch
29	Air cooling valve % command	% value	Sends percentage commands to electrovalve air cooling 1st stage	5.001 DPT_Scaling
31	V1 fan switching air cooling	On/Off	Sends air cooling fancoil speed 1 electrovalve activation/deactivation commands	1.001 DPT_Switch
31	% command fancoil speed air cooling	% value	Sends percentage commands to fancoil heating	5.001 DPT_Scaling
32	V2 fan switching air cooling	On/Off	Sends commands of activation/deactivation fancoil speed 2 air cooling	1.001 DPT_Switch
33	V3 fan switching air cooling	On/Off	Sends commands of activation/deactivation fancoil speed 3 air cooling	1.001 DPT_Switch
37	Air cooling 2° stage switching	On/Off	Sends air cooling electrovalve activation/deactivation commands 2nd stage	1.001 DPT_Switch
37	% command 2° stage air cooling	% value	Sends percentage commands to electrovalve air cooling 2nd stage	5.001 DPT_Scaling
40	Fan coil mode report	Automatic/manual	Sends feedback about fancoil speed automatic/manual mode	1.001 DPT_Switch
42	Auxiliary sensor measured temperature	Value °C	Sends temperature values in degrees Celsius measured by NTC sensor connected to input	9.001 DPT_Temp
42	Auxiliary sensor measured temperature	°K value	Sends temperature values in degrees Kelvin measured by NTC sensor connected to input	9.002 DPT_Tempd
42	Auxiliary sensor measured temperature	°F value	Sends temperature values in	9.027 DPT_Temp_F

			degrees Fahrenheit measured by NTC sensor connected to input	
43	Measured temperature	Value °C	Sends temperature values in degrees Celsius calculated by device	9.001 DPT_Temp
43	Measured temperature	°K value	Sends temperature values in degrees Kelvin calculated by device	9.002 DPT_Tempd
43	Measured temperature	°F value	Sends temperature values in degrees Fahrenheit calculated by device	9.027 DPT_Temp_F
44	HVAC mode feedback	Eco/Precom/Comf/Off	Sends feedback about HVAC mode set	20.102 DPT_HVACMode
44	Functioning setpoint feedback	Value °C	Sends functioning setpoint values in degrees Celsius	9.001 DPT_Temp
44	Functioning setpoint feedback	°K value	Sends functioning setpoint values in degrees Kelvin	9.002 DPT_Tempd
44	Functioning setpoint feedback	°F value	Sends functioning setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F
45	HVAC off mode feedback	Enable/Disable	Sends feedback about HVAC off (building protection) mode activation status	1.003 DPT_Enable
46	HVAC economy mode feedback	Enable/Disable	Sends feedback about HVAC economy mode activation status	1.003 DPT_Enable
47	HVAC precomfort mode feedback	Enable/Disable	Sends feedback about HVAC precomfort mode activation status	1.003 DPT_Enable
48	HVAC comfort mode feedback	Enable/Disable	Sends feedback about HVAC comfort mode activation status	1.003 DPT_Enable
49	Functioning type feedback	Heating/Air cooling	Sends feedback about functioning type set	1.100 DPT_Heat/Cool
50	Heating anti-freeze setpoint feedback	Value °C	Sends setpoint values heating off HVAC mode in degrees Celsius	9.001 DPT_Temp
50	Heating anti-freeze setpoint feedback	°K value	Sends setpoint values heating off HVAC mode in degrees Kelvin	9.002 DPT_Tempd
50	Heating anti-freeze setpoint feedback	°F value	Sends setpoint values heating off HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
51	Heating economy setpoint feedback	Value °C	Sends setpoint values heating economy HVAC mode in degrees Celsius	9.001 DPT_Temp
51	Heating economy setpoint feedback	°K value	Sends setpoint values heating economy HVAC mode in degrees Kelvin	9.002 DPT_Tempd
51	Heating economy setpoint feedback	°F value	Sends setpoint values heating economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
52	Heating precomfort setpoint feedback	Value °C	Sends setpoint values heating precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp
52	Heating precomfort setpoint feedback	°K value	Sends setpoint values heating precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
52	Heating precomfort setpoint feedback	°F value	Sends setpoint values heating precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
53	Heating comfort setpoint feedback	Value °C	Sends setpoint values heating comfort HVAC mode in degrees Celsius	9.001 DPT_Temp
53	Heating comfort setpoint feedback	°K value	Sends setpoint values heating comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
53	Heating comfort setpoint feedback	°F value	Sends setpoint values heating comfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
54	Air cond. high temp. protection setpoint feedback	Value °C	Sends setpoint values air cooling off HVAC mode in degrees	9.001 DPT_Temp

			Celsius	
54	Air cond. high temp. protection setpoint feedback	°K value	Sends setpoint values air cooling off HVAC mode in degrees Kelvin	9.002 DPT_Tempd
54	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
55	Air cooling economy setpoint feedback	Value °C	Sends setpoint values air cooling economy HVAC mode in degrees Celsius	9.001 DPT_Temp
55	Air cooling economy setpoint feedback	°K value	Sends setpoint values air cooling economy HVAC mode in degrees Kelvin	9.002 DPT_Tempd
55	Air cooling economy setpoint feedback	°F value	Sends setpoint values air cooling economy HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
56	Air cooling precomfort setpoint feedback	Value °C	Sends setpoint values air cooling precomfort HVAC mode in degrees Celsius	9.001 DPT_Temp
56	Air cooling precomfort setpoint feedback	°K value	Sends setpoint values air cooling precomfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
56	Air cooling precomfort setpoint feedback	°F value	Sends setpoint values air cooling precomfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
57	Air cooling comfort setpoint feedback	Value °C	Sends setpoint values air cooling comfort HVAC mode in degrees Celsius	9.001 DPT_Temp
57	Air cooling comfort setpoint feedback	°K value	Sends setpoint values air cooling comfort HVAC mode in degrees Kelvin	9.002 DPT_Tempd
57	Air cooling comfort setpoint feedback	°F value	Sends setpoint values air cooling comfort HVAC mode in degrees Fahrenheit	9.027 DPT_Temp_F
58	Current setpoint report	Value °C	Sends active setpoint values in degrees Celsius	9.001 DPT_Temp
58	Current setpoint report	°K value	Sends active setpoint values in degrees Kelvin	9.002 DPT_Tempd
58	Current setpoint report	°F value	Sends active setpoint values in degrees Fahrenheit	9.027 DPT_Temp_F
59	Thermostat functioning feedback	Slave/Stand-alone	Send Slave or stand-alone thermostat functioning feedback	1.001 DPT_Switch
63	Input aux 1 - Switching	On/Off	Sends dimmer on/off switching commands	1.001 DPT_Switch
63	Inputs aux 1/2 - Switching	On/Off	Sends dimmer on/off switching commands	1.001 DPT_Switch
63	Input aux 1 – Shutter movement	Up/Down	Sends up/down shutter movement commands	1.008 DPT_UpDown
63	Inputs aux 1/2 - Shutter movement	Up/Down	Sends up/down shutter movement commands	1.008 DPT_UpDown
63	Input aux 1 - Scene	Execute/Store	Sends scene storing/execution commands	18.001 DPT_SceneControl
63	Input aux 1 - 1 bit value	1/0 value	Sends 1/0 values	1.002 DPT_Bool
63	Input aux 1 - 2 bit value	On/Off forced positioning	Sends 2-bit values	2.001 DPT_Switch_Control
63	Input aux 1 - 1 byte value	Unsigned value	Sends unsigned values (0.. 255)	5.010 DPT_Value_1_Ucount
63	Input aux 1 - 1 byte value	Signed value	Sends signed values (-128.. +127)	6.010 DPT_Value_1_Count
63	Input aux 1 - 1 byte value	% value	Sends percentage values (0%.. 100%)	5.001 DPT_Scaling
63	Input aux 1 - 1 byte value	HVAC mode	Sends HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
63	Input aux 1 - 2 byte value	Unsigned value	Sends unsigned values (0.. 65535)	7.001 DPT_Value_2_Ucount
63	Input aux 1 - 2 byte value	Signed value	Sends signed values (-32768..	8.001

			+32767)	DPT_Value_2_Count
63	Input aux 1 - 4 byte value	Unsigned value	Sends unsigned values (0.. 4294967295)	12.001 DPT_Value_4_Ucount
63	Input aux 1 - 4 byte value	Signed value	Sends signed values (- 2147483648.. +2147483647)	13.001 DPT_Value_4_Count
63	Input aux 1 - 14 byte value	Characters ISO 8859-1	Sends characters encoded according to ISO 8859-1 standard	16.001 DPT_String_8859_1
63	Input aux 1 – Window status feedback	Open/Closed	Sends window status feedback according to input aux status	1.019 DPT_Window_Door
64	Input aux 1 – Brightness control	Increase/Decrease	Sends relative brightness control commands	3.007 DPT_Control_Dimmin g
64	Inputs aux 1/2 - Brightness control	Increase/Decrease	Sends relative brightness control commands	3.007 DPT_Control_Dimmin g
64	Input aux 1 – Shutter stop/Louvres control	Stop/Step	Sends movement stop/shutter control commands	1.007 DPT_Step
64	Inputs aux 1/2 - Shutter stop/Louvres control	Stop/Step	Sends movement stop/shutter control commands	1.007 DPT_Step
66	Input aux 2 - Switching	On/Off	Sends dimmer on/off switching commands	1.001 DPT_Switch
66	Input aux 2 – Shutter movement	Up/Down	Sends up/down shutter movement commands	1.008 DPT_UpDown
66	Input aux 2 – Scene	Execute/Store	Sends scene storing/execution commands	18.001 DPT_SceneControl
66	Input aux 2 - 1 bit value	1/0 value	Sends 1/0 values	1.002 DPT_Bool
66	Input aux 2 - 2 bit value	On/Off forced positioning	Sends 2-bit values	2.001 DPT_Switch_Control
66	Input aux 2 - 1 byte value	Unsigned value	Sends unsigned values (0.. 255)	5.010 DPT_Value_1_Ucount
66	Input aux 2 - 1 byte value	Signed value	Sends signed values (-128.. +127)	6.010 DPT_Value_1_Count
66	Input aux 2 - 1 byte value	% value	Sends percentage values (0%.. 100%)	5.001 DPT_Scaling
66	Input aux 2 - 1 byte value	HVAC mode	Sends HVAC modes (auto/comfort/precomfort/economy/off)	20.102 DPT_HVACMode
66	Input aux 2 - 2 byte value	Unsigned value	Sends unsigned values (0.. 65535)	7.001 DPT_Value_2_Ucount
66	Input aux 2 - 2 byte value	Signed value	Sends signed values (-32768.. +32767)	8.001 DPT_Value_2_Count
66	Input aux 2 - 4 byte value	Unsigned value	Sends unsigned values (0.. 4294967295)	12.001 DPT_Value_4_Ucount
66	Input aux 2 - 4 byte value	Signed value	Sends signed values (- 2147483648.. +2147483647)	13.001 DPT_Value_4_Count
66	Input aux 2 - 14 byte value	Characters ISO 8859-1	Sends characters encoded according to ISO 8859-1 standard	16.001 DPT_String_8859_1
67	Input aux 2 – Brightness control	Increase/Decrease	Sends relative brightness control commands	3.007 DPT_Control_Dimmin g
67	Input aux 2 – Shutter stop/Louvres control	Stop/Step	Sends movement stop/shutter control commands	1.007 DPT_Step
68	Output aux - Status	On/Off	Sends on/off output status feedback	1.001 DPT_Switch
74	Device 1 - Send HVAC mode	Eco/Precom/Com/Off	Sends functioning HVAC mode to device 1	20.102 DPT_HVACMode
74	Device 1 - Setpoint sending	Value °C	Sends functioning setpoint in degrees Celsius to device 1	9.001 DPT_Temp
74	Device 1 - Setpoint sending	°K value	Sends functioning setpoint in degrees Kelvin to device 1	9.002 DPT_Tempd
74	Device 1 - Setpoint sending	°F value	Sends functioning setpoint in degrees Fahrenheit to device 1	9.027 DPT_Temp_F
75	Device 1 - Send functioning type	Heating/Air cooling	Send functioning type to device 1	1.100 DPT_Heat/Cool
81	Device 2 - Send HVAC mode	Eco/Precom/Com/Off	Sends functioning HVAC mode to device 2	20.102 DPT_HVACMode

81	Device 2 - Setpoint sending	Value °C	Sends functioning setpoint in degrees Celsius to device 2	9.001 DPT_Temp
81	Device 2 - Setpoint sending	°K value	Sends functioning setpoint in degrees Kelvin to device 2	9.002 DPT_Tempd
81	Device 2 - Setpoint sending	°F value	Sends functioning setpoint in degrees Fahrenheit to device 2	9.027 DPT_Temp_F
82	Device 2 - Send functioning type	Heating/Air cooling	Send functioning type to device 2	1.100 DPT_Heat/Cool
88	Device 3 - Send HVAC mode	Eco/Precom/Comf/Off	Sends functioning HVAC mode to device 3	20.102 DPT_HVACMode
88	Device 3 - Setpoint sending	Value °C	Sends functioning setpoint in degrees Celsius to device 3	9.001 DPT_Temp
88	Device 3 - Setpoint sending	°K value	Sends functioning setpoint in degrees Kelvin to device 3	9.002 DPT_Tempd
88	Device 3 - Setpoint sending	°F value	Sends functioning setpoint in degrees Fahrenheit to device 3	9.027 DPT_Temp_F
89	Device 3 - Send functioning type	Heating/Air cooling	Sends functioning type to device 3	1.100 DPT_Heat/Cool
95	Device 4 - Send HVAC mode	Eco/Precom/Comf/Off	Sends functioning HVAC mode to device 4	20.102 DPT_HVACMode
95	Device 4 - Setpoint sending	Value °C	Sends functioning setpoint in degrees Celsius to device 4	9.001 DPT_Temp
95	Device 4 - Setpoint sending	°K value	Sends functioning setpoint in degrees Kelvin to device 4	9.002 DPT_Tempd
95	Device 4 - Setpoint sending	°F value	Sends functioning setpoint in degrees Fahrenheit to device 4	9.027 DPT_Temp_F
96	Device 4 - Send functioning type	Heating/Air cooling	Sends functioning type to device 4	1.100 DPT_Heat/Cool
103	Measured relative humidity	Value [% RH]	Sends the relative humidity values calculated by the device	9.007 DPT_Value_Humidity
104	Relative humidity estimated	Value [% RH]	Sends the relative humidity values estimated at the temperature received by the auxiliary probe	9.007 DPT_Value_Humidity
105	Specific humidity	Value [g/kg]	Sends specific humidity values in g/kg	14.005 DPT_Value_Amplitude
106	Dew point temperature	Value [°C]	Sends the dew point temperature values (°C)	9.001 DPT_Value_Temp
106	Dew point temperature	Value [°K]	Sends the dew point temperature values (°K)	9.002 DPT_Value_Tempd
106	Dew point temperature	Value [°F]	Sends the dew point temperature values (°F)	9.027 DPT_Value_Temp_F
108	Dew point temperature alarm enabling status	Enabled/Disabled	Sends dew point temperature alarm threshold enabling status	1.003 DPT_Enable
110	Dew point temperature alarm threshold feedback	Value [°C]	Sends the dew point temperature alarm threshold actual values (°C)	9.001 DPT_Value_Temp
110	Dew point temperature alarm threshold feedback	Value [°K]	Sends the dew point temperature alarm threshold actual values (°K)	9.002 DPT_Value_Tempd
110	Dew point temperature alarm threshold feedback	Value [°F]	Sends the dew point temperature alarm threshold actual values (°F)	9.027 DPT_Value_Temp_F
111	Dew point temperature alarm output	1/0 value	Sends the values 1/0 associated with dew point temperature alarm	1.001 DPT_OnOff
111	Dew point temperature alarm output	On/Off forced positioning	Sends the 2 bit values associated with dew point temperature alarm	2.001 DPT_Switch_Control
111	Dew point temperature alarm output	Value [0 .. 255]	Sends the unsigned values (0..255) associated with dew point temperature alarm	5.010 DPT_Value_1_Ucount
111	Dew point temperature alarm output	Value [-128 .. 127]	Sends the signed values (-128..127) associated with dew point temperature alarm	6.010 DPT_Value_1_Count
111	Dew point temperature alarm output	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated with dew point temperature alarm	5.001 DPT_Percentage

111	Dew point temperature alarm output				HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated with dew point temperature alarm	20.102 DPT_HVAC_Mode	
111	Dew point temperature alarm output				Value [0 .. 65535]	Sends the unsigned values (0..65535) associated with dew point temperature alarm	7.001 DPT_Value_2_Ucount	
111	Dew point temperature alarm output				Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated with dew point temperature alarm	8.001 DPT_Value_2_Count	
113	125	137	149	161	Relative humidity threshold x enabling status	Enabled/Disabled	Sends relative humidity threshold x enabling status	1.003 DPT_Enable
115	127	139	151	163	Relative humidity threshold x feedback	Value [% RH]	Sends the relative humidity threshold x current value	9.007 DPT_Value_Humidity
116	128	140	152	164	Relative humidity threshold x output A	1/0 value	Sends the values 1/0 associated with the output A of relative humidity threshold x	1.001 DPT_Switch
116	128	140	152	164	Relative humidity threshold x output A	On/Off forced positioning	Sends the 2 bit values associated with the output A of relative humidity threshold x	2.001 DPT_Switch_Control
116	128	140	152	164	Relative humidity threshold x output A	Value [0 .. 255]	Sends the unsigned values (0..255) associated with the output A of relative humidity threshold x	5.010 DPT_Value_1_Ucount
116	128	140	152	164	Relative humidity threshold x output A	Value [-128 .. 127]	Sends the signed values (-128..127) associated with the output A of relative humidity threshold x	6.010 DPT_Value_1_Count
116	128	140	152	164	Relative humidity threshold x output A	Value [0% .. 100%]	Sends the percentage values (0%..100%) associated with the output A of relative humidity threshold x	5.001 DPT_Percentage
116	128	140	152	164	Relative humidity threshold x output A	HVAC Mode (com/precom/eco/off)	Sends the HVAC Mode values (comfort/precomfort/economy/off) associated with the output A of relative humidity threshold x	20.102 DPT_HVAC_Mode
116	128	140	152	164	Relative humidity threshold x output A	Value [0 .. 65535]	Sends the unsigned values (0..65535) associated with the output A of relative humidity threshold x	7.001 DPT_Value_2_Ucount
116	128	140	152	164	Relative humidity threshold x output A	Value [-32768 .. 32767]	Sends the signed values (-32768..32767) associated with the output A of relative humidity threshold x	8.001 DPT_Value_2_Count
116	128	140	152	164	Relative humidity threshold x output A	Setpoint value [°C]	Sends the setpoint values (°C) associated with the output A of relative humidity threshold x	9.001 DPT_Value_Temp
116	128	140	152	164	Relative humidity threshold x output A	Setpoint value [°K]	Sends the setpoint values (°K) associated with the output A of relative humidity threshold x	9.002 DPT_Value_Tempd
116	128	140	152	164	Relative humidity threshold x output A	Setpoint value [°F]	Sends the setpoint values (°F) associated with the output A of relative humidity threshold x	9.027 DPT_Value_Temp_F
118	130	142	154	166	Relative humidity threshold x output B	1/0 value	Sends the values 1/0 associated with the output B of relative humidity threshold x	1.001 DPT_Switch
120	132	144	156	168	Relative humidity threshold x output C	1/0 value	Sends the values 1/0 associated with the output C of relative humidity threshold x	1.001 DPT_Switch
122	134	146	158	170	Relative humidity threshold x output D	1/0 value	Sends the values 1/0 associated with the output D of relative humidity threshold x	1.001 DPT_Switch
173	Thermal comfort status feedback				1=Comfortable/0=Uncomfortable	Sends the thermal comfort status feedback	1.002 DPT_Bool	

The object variants shown in blue in the table above are not shown for objects B (objects 118/130/142/154/166), C (objects 120/132/144/156/168), D (object 122/134/146/158/170), but are still present.

## 20 ETS programming error feedback

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

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.

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 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 the linked to two addresses that are the same. In this manner, if linked to two different addresses, also the pairs for speeds 2 and 3 must be linked to two different addresses. If a solenoid valve is connected (heating or air cooling), also the respective speeds must be connected, and vice versa. In common logic mode, and if connecting the heating fan speeds, they must also be connected for air cooling, and vice versa.</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

Punto di contatto indicato in adempimento ai fini delle direttive e regolamenti UE applicabili:

*Contact details according to the relevant European Directives and Regulations:*

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