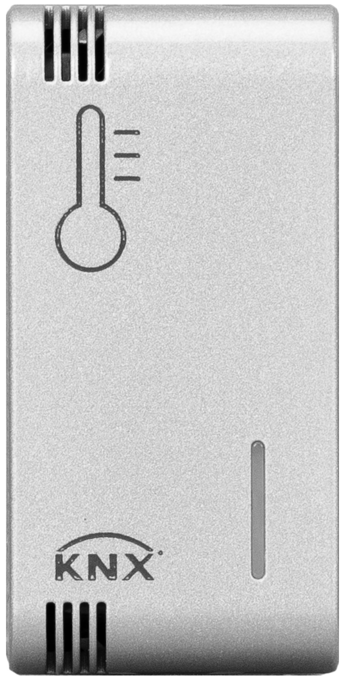
KNX system flush-mounting temperature adjustment sensor



GW 1x799

Technical Manual

Contents

1 Introduction

This manual explains the functions of the "**KNX System temperature adjustment sensor**" (GW1x799) device and how they are set and configured with the aid of the ETS configuration software.

2 Application

The KNX flush-mounting temperature adjustment sensor is used, with the aid of a KNX timed thermostat (GW 10 794 - GW 12 794 - GW 14 794) or a KNX thermostat (GW 10 795 - GW 12 795 – GW 14 795), to manage the temperature of the environment where it is installed or of another environment when used with an external temperature sensor.
The sensor is not equipped with its own visualisation and command elements, so it must be used with a KNX device (e.g. a KNX thermostat or a KNX timed thermostat) that can control its parameters (HVAC or Setpoint mode and operating type).
The temperature adjustment sensor has various 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 fan coil 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 1byte 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 sensor / external temperature sensor with definition of the relative weight.
- Underfloor sensor
  - setting of threshold value for floor temperature alarm.
- Temperature control for specific zones:
  - with the operating mode received from the Master device, and the use of a local setpoint;
  - with the Setpoint value received from the Master device, and local residual current device for temperature.
- Scenes
  - memorisation and activation of 8 scenes (value 0..63).
- Other functions:
  - setting of the Setpoint (OFF, ECONOMY, PRECOMFORT, COMFORT) from the BUS
  - setting of the operating Setpoint from the BUS;
  - setting of the functioning type (heating/air cooling) from the BUS;
  - transmission of the status information (mode, type), measured temperature and current setpoint on the BUS;
  - auxiliary input for fronts management, short/long operation, dimmer with single push-button, roller shutters with single push-button, scenes and window contact;

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 define the use of the input channels and the signalling leds. The basic structure of the menu is as follows:

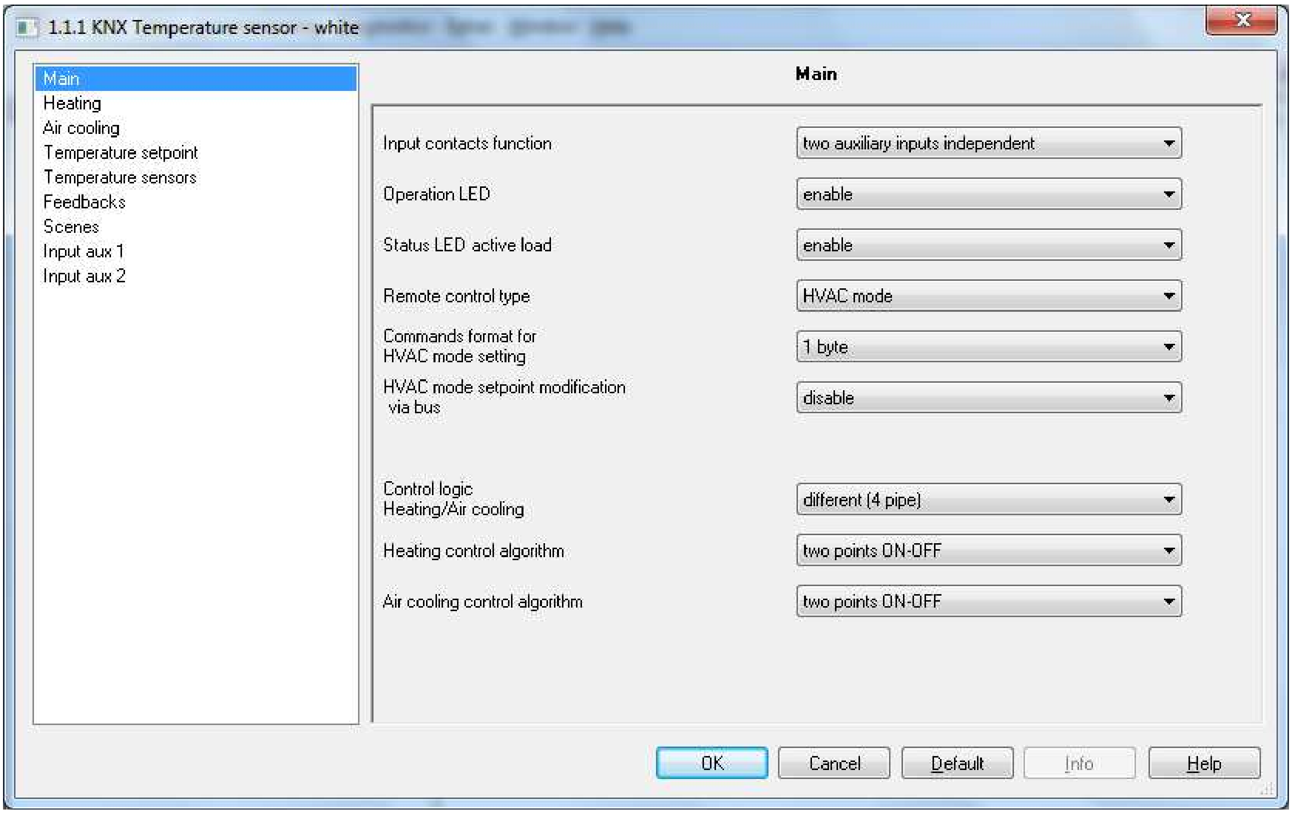
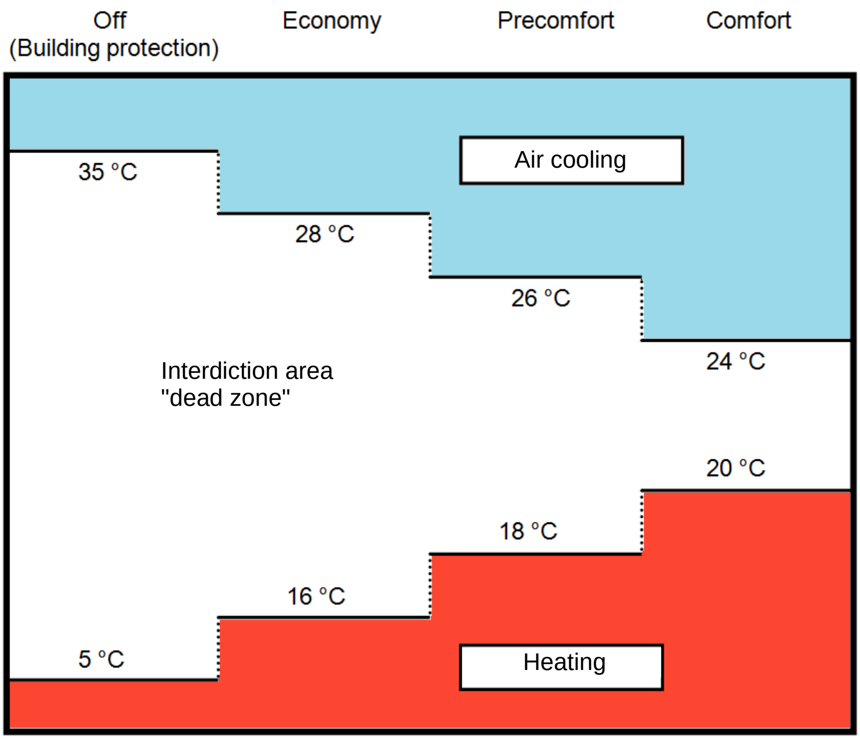


Fig. 3.1

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

The device's temperature adjustment functioning type (heating or air cooling) can be managed by a remote device via BUS command or independently by the device. 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 BUS command.

The object ***Operating type input*** (Data Point Type: 1.100 DPT_Heat/Cool) is used to change the functioning type remotely and the object ***Enable dead zone*** (Data Point Type: 1.003 DPT_Enable) is used to enable or disable the dead zone. When the dead zone is disabled via a BUS command on the object ***Enable dead zone***, the active functioning type remains the one set automatically and can be modified via a BUS command on the object ***Functioning type input***.
When the application is downloaded, the dead zone is disabled and the set functioning type is HEATING; this ensures that if the object ***Enable dead zone*** is not linked, it is possible to remotely modify the device's functioning type via the object ***Functioning type input***.

3.1 Parameters

➢ *3.1.1 Input contacts function*

The device has two input channels (two terminals plus common) for connecting the potential-free contacts; Each of the 2 input channels activated from the device can be autonomously managed by performing an autonomous function or can be combined with another one to produce a single, common function; the second channel can be used alternatively to connect an NTC external sensor that can be used as an external temperature sensor or as a floor sensor. The parameter **"Input contacts function"** 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 database structure will vary depending on 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 **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**.

➢ *3.1.2 Operation LED*

The parameter **"Operation LED"** is used to enable the luminous signalling of the green LED, which signals that the device is operating correctly; the values that can be set are:

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

➢ *3.1.3 Status LED active load*

The parameter **"Status LED active load"** is used to enable the luminous signalling of the red LED that identifies the activation of the active functioning type solenoid valve; the values that can be set are:

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

Selecting **enable** the red LED turns on when the control algorithm activates the relative temperature regulation valve; if the valve feedback is active, the red signal flashes if there is no valve status feedback.

➢ *3.1.4 Remote control type*

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

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

Selecting **HVAC mode** displays the parameters **"Format of the commands for setting HVAC mode"** and **"Change HVAC mode setpoint via BUS"**; selecting **setpoint** displays the parameters **"Object measure unit Setpoint input"** and **"Temporary setpoint forcing via BUS"** and the communication object ***Setpoint input***.

➢ *3.1.5 Commands format for HVAC mode setting*

This is used to define the format of the communication objects used for the remote control of the HVAC mode of the KNX sensor; the values that can be set are:
- 1 bit
- **1 byte (default value)**
- both

Selecting **1 byte** or **both**, displays the communication object ***HVAC mode input*** (Data Point Type: 20.102 DPT_HVACMode) via which the remote device changes the HVAC mode via a single command; selecting **1 bit** or **both** displays the parameters **"Reception of 1 bit mode activation with priority > current"**, **"Reception of 1 bit mode activation with priority < current"** and the communication objects ***HVAC off mode input***, ***HVAC economy mode input***, ***HVAC precomfort mode input***, ***HVAC comfort mode input*** and ***HVAC auto mode input*** (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	Window contact function aux input 1	-
	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 mode input/Setpoint input/Scene	1 byte/2 byte /1 byte

The 1 bit mode setting objects all have a higher priority than the 1 byte mode setting object; this is due to the fact that, by enabling both the mode setting possibilities, the 1 bit objects can be used to fix the mode if particular events occur.
Naturally there is also a command execution priority between the 1 bit mode setting objects, especially because if only the 1 bit format is set for setting the functioning mode and if multiple objects are enabled, it is necessary to determine which of these have greater priority to determine the active functioning mode on the device; due to the fact that multiple 1 bit objects can be enabled at the same time, it is possible to define the behaviour of the device when a communication object is received with a higher priority than the one currently active at that moment using the parameter **"Reception of 1 bit mode activation with priority > current"**; in the same manner, the behaviour of the device can be defined when a communication object is received with a lower priority than the one currently active at that moment using the parameter **"Reception of 1 bit mode activation with priority < current"**.
The values that can be set for the parameter **"Reception of 1 bit mode activation with priority > current"** are:

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

By selecting **maintain value of objects with lower priority**, when receiving a 1 bit HVAC mode activation command with a priority higher than the one currently active, the mode of the new object is set, but the activation status of the objects with a lower priority is maintained; instead by setting **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 parameter **"Reception of 1 bit mode activation with priority < current"** 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; instead by setting **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).

3.1.6 HVAC mode setpoint modification via BUS

This is used to enable the communication objects necessary for setting the setpoints of each device mode via BUS telegram; the values that can be set are:

- **disable** **(default value)**
- enable objects (°C)
- enable objects (°K)
- enable objects (°F)

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 a nd *9.027 DPT_Value_Temp_F* if °F) through which it is possible to set the setpoints of every device operating mode via BUS.

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

– $T_{antifreeze} \leq T_{economy} \leq T_{precomfort} \leq T_{comfort}$ in heating mode ("T" indicates the generic value of the mode setpoint)
– $T_{comfort} \leq T_{precomfort} \leq T_{economy} \leq T_{high\ temp.\ protection}$ in air cooling mode ("T" indicates the generic 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 remote control type is setpoint, between the various setpoints belonging to the same functioning type, there is a value setting limit that is determined by the following relationship:

– $T_{building\ protection} \leq T_{functioning}$ in heating mode ("T" indicates the generic value of the setpoint)
– $T_{functioning} \leq T_{building\ protection}$ in air cooling mode ("T" indicates the generic 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.

3.1.7 Object measure unit Setpoint input

This is used to set the measure unit with which the information received via the communication object ***Setpoint input*** will be decoded. the values that can be set are:

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

The value set for this parameter changes the coding of the communication object ***Setpoint input***: *9.001 DPT_Value_Temp* if the value is **degrees Celsius (°C)**, *9.002 DPT_Value_Tempd* if the value is **degrees Kelvin (°K)** and *9.027 DPT_Value_Temp_F* if the value is **degrees Fahrenheit (°F)**.

Some hotel applications require to be able to change the operating setpoint within a defined value range; to do this, the remote device that controls the KNX sensor sends setpoints that are not memorised by the sensor, but are used temporarily; the operating setpoint must not be modified by these telegrams, only the current setpoint is modified. This operation is the same that takes place when the operating setpoint is temporarily forced on the KNX Thermostat using the UP and DOWN push-buttons. The parameter **"Temporary setpoint forcing via BUS"** is used to enable the object ***Temporary setpoint forcing input*** through which the KNX sensor receives the temporary setpoint values to be used; the values that can be set are:

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

Select **enable** to view the communication object ***Temporary setpoint forcing input*** (Data Point Type: *9.001 DPT_Value_Temp* if the value of the parameter **"Object measure unit Setpoint input"** 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)**). At BUS voltage failure, the forced setpoint is not saved and as a result, after voltage recovery, the active setpoint will be the operating setpoint.

➢ *3.1.8 Heating control algorithm*

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 parameter **"Heating/air cooling control logic"** is used to define if the system control logic, and as a result the control communication object, is common for the heating and air cooling or if it is different; the values that can be set are:

- common
- **different** **(default value)**

Select **common** to view the parameters **"Heating/air cooling control algorithm"** and **"Heating/air cooling valve status feedback"**, whereas select **different** to view the parameters **"Heating control algorithm"** and **"Air cooling control algorithm"**.

➢ *3.1.9 Heating control algorithm*

This is used to define the control algorithm used for the heating system; the values that can be set are:

- **2 points ON-OFF** **(default value)**
- 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 parameter **"Regulation differential (tenth of °C)"** in the **Heating** menu and the communication object ***Heating valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **2 points 0%-100%** to view the parameter **"Regulation differential (tenth of °C)"** in the **Heating** menu and the communication object ***Heating valve % command***(Data Point Type: 5.001 DPT_Scaling) via which the device sends the command telegrams.
Select **PWM proportional integral** to view the parameters **"Select heating system"**, **"Proportional band"**, **"Integration time"** and **"Cycle time"** in the **Heating** menu and the communication object ***Heating valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **continuous proportional-integral** to view the parameters **"Select heating system"**, **"Proportional band"**, **"Integration time"** and **"Min. % variation for continuous command sending"** in the **Heating** menu and the communication object ***Heating valve % command*** (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 parameters **"Fancoil valve management"**, **"Valve regulation differential (tenth of °C)"** , **"Number of fancoil speeds"** and **"Fancoil speed status feedback"** in the **Heating** menu.

➢ *3.1.10 Air cooling control algorithm*

This is used to define the control algorithm used for the air cooling system; the values that can be set are:

- **2 points ON-OFF** **(default value)**
- 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 parameter **"Regulation differential (tenth of °C)"** in the **Air cooling** menu and the communication object ***Air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **2 points 0%-100%** to view the parameter **"Regulation differential (tenth of °C)"** in the **Air cooling** menu and the communication object ***Air cooling valve % command*** (Data Point Type: 5.001 DPT_Scaling) via which the device sends the command telegrams.
Select **PWM proportional integral** to view the parameters **"Select air cooling system"**, **"Proportional band"**, **"Integration time"** and **"Cycle time"** in the **Air cooling** menu and the communication object ***Air conditioning valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **continuous proportional-integral** to view the parameters **"Select air cooling system"**, **"Proportional band"**, **"Integration time"** and "**Min. % variation for continuous command sending"** in the **Air cooling** menu and the communication object ***Air cooling valve % command*** (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 parameters **"Fancoil valve management"**, **"Valve regulation differential (tenth of °C)"**, **"Number of fancoil speeds"** and **"Fancoil speed status feedback"** in the **Air cooling** menu.

➢ *3.1.11 Heating/air cooling control algorithm*

This is used to define the control algorithm used for the heating system as well as for the air cooling system, due to the fact that 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
- fan coil with ON-OFF speed control
- fan coil with continuous speed control

Select **2 points ON-OFF** in the **Heating** and **Air cooling** menus to display the parameters **"Regulation differential (tenth of °C)"** and the communication object ***Heating/air cooling valve switching*** (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 parameters **"Regulation differential (tenth of °C)"** and the communication object ***Heating/air cooling valve % command*** (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 parameters **"Select heating system**(**air cooling** in the **Air cooling menu)"**, **"Proportional band"**, **"Integration time"** and **"Cycle time"** and the communication object ***Heating/air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams.
Select **continuous proportional-integral** to display the parameters **"Selecting heating system**(**air cooling** in the **Air cooling menu)"**, **"Proportional band"**, **"Integration time"** and **"Min. % variation for continuous command sending"** in the **Heating** and **Air cooling** menus and the communication object ***Heating/air cooling valve % command*** (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 parameters **"Fancoil valve management"** and **"Valve regulation differential (tenth of °C)"** and in the

Heating and **Air cooling** the parameters **"Number of fancoil speeds"** and **"Fancoil speed status feedback"** are displayed.

If the control algorithm is fancoil, the format of the heating/air cooling solenoid valve commands (2-way system) is independent on that of the fancoil speed control; the parameter **"Fancoil valve management"** 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 communication object ***Heating/air cooling valve switching*** (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 communication object ***Heating/air cooling valve % command*** (Data Point Type: 5.001 DPT_Scaling) via which the device sends the command telegrams to the solenoid valve.

The parameter **"Heating/air cooling solenoid valve regulation differential (tenth of °C)"** is used to set the regulation differential value of the 2 points control of the fancoil operating solenoid valve, as mentioned in the Control algorithms paragraph; the value is the same for the heating system and for the air cooling system. The values that can be set are:

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

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

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

Select **disable** to view the parameter **"Command repetition period with disabled feedback"**; select **enable** to view the communication object ***Heating/air cooling valve status feedback*** (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 object ***Heating/air cooling valve status feedback*** or ***Heating/air cooling valve % feedback*** to be updated about the status of the heating/air cooling solenoid valve.

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

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

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

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 parameter **"Modify fancoil speed via BUS"** 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 communication object ***Fancoil mode input*** (Data Point Type: 1.001 DPT_Switch) which makes it possible to receive fancoil mode selection commands; when value "1" is received, the speed is defined autonomously by the device (AUTO fancoil mode) according to the different hystereses defined in the Control algorithms paragraph. When the value "0" is received, 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 parameter **"Fancoil mode feedback (automatic/manual)"** is used to enable the communication object ***Fancoil mode feedback*** through 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

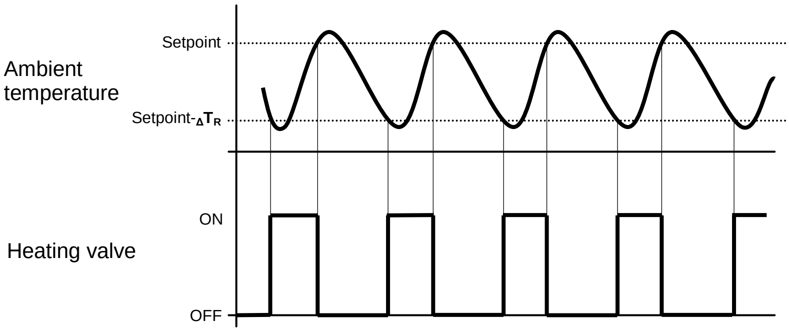
Selecting **enable** displays the communication object ***Fancoil mode feedback*** used to transmit the feedback. When fancoil speed control mode changes from MANUAL to AUTOMATIC, the device sends a telegram via BUS with the logical value "1"; when fancoil speed control mode changes from AUTOMATIC to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

3.2 Control algorithms

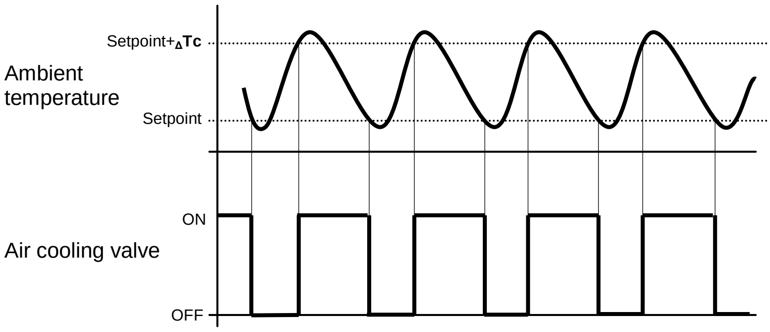
Below there is more information about the available control algorithms, regardless if the control logic is common or different for the functioning types (heating/air cooling):

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

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 that there is not a single threshold that discriminates between the turning on and off of the system, but there are two.



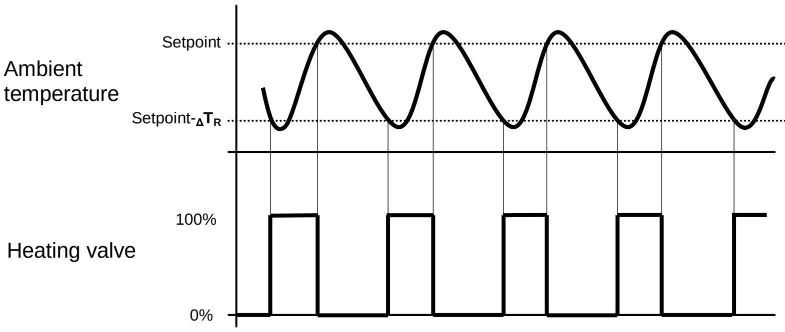
When the measured temperature is lower than the value “setpoint-ΔT_R” (where ΔT_R identifies the value of the heating regulation differential) the device turns on the heating system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches the fixed setpoint value, the device turns off the heating system, sending the relative BUS command to the 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- ΔT_R” below which the device turns on the system, the second consists of the setpoint value that was set, above which the device turns off the system.



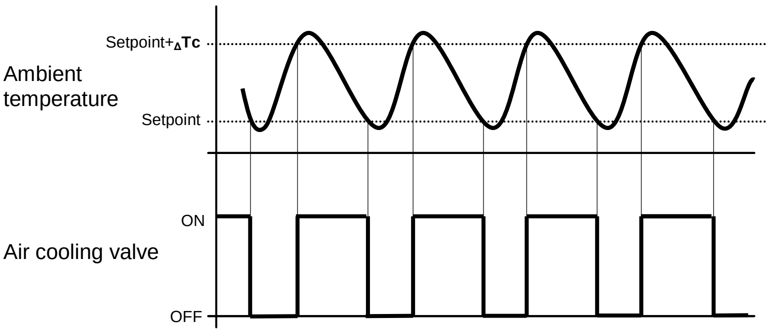
When the measured temperature is higher than the value “setpoint+ ΔTc” (where ΔTc identifies the air cooling regulation differential) the device turns on the air cooling system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches the fixed setpoint value, the device turns off the air cooling system, sending the relative BUS command to the 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+ΔTc” above which the device turns on the system.
To avoid the continuous switchovers 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 that there is not a single threshold that discriminates between the turning on and off of the system, but there are two.



When the measured temperature is lower than the value "setpoint-**ΔT_R**" (where **ΔT_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- **ΔT_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 switchovers 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+ **ΔTc**" (where **ΔTc** identifies the air cooling regulation differential) the device turns on the air cooling system, sending the relative BUS command to the actuator that manages it; when the detected temperature reaches the fixed setpoint value, the device turns off the air cooling system, sending the relative BUS command to the 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+**ΔTc**" 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.

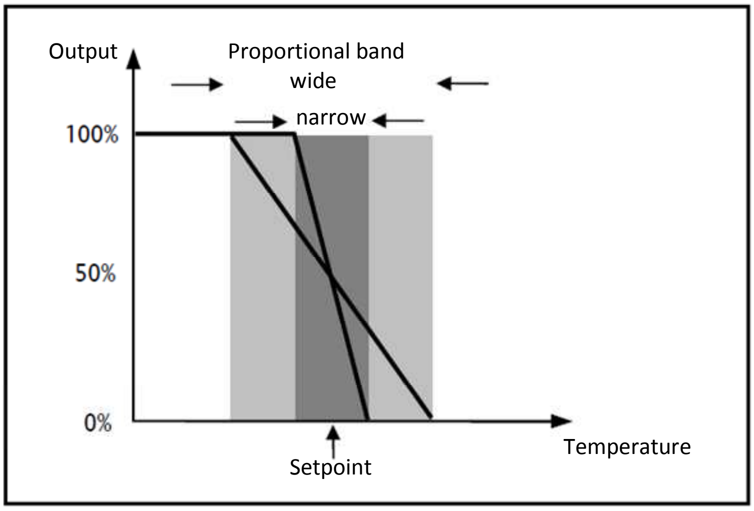
$$\mathrm{u}(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

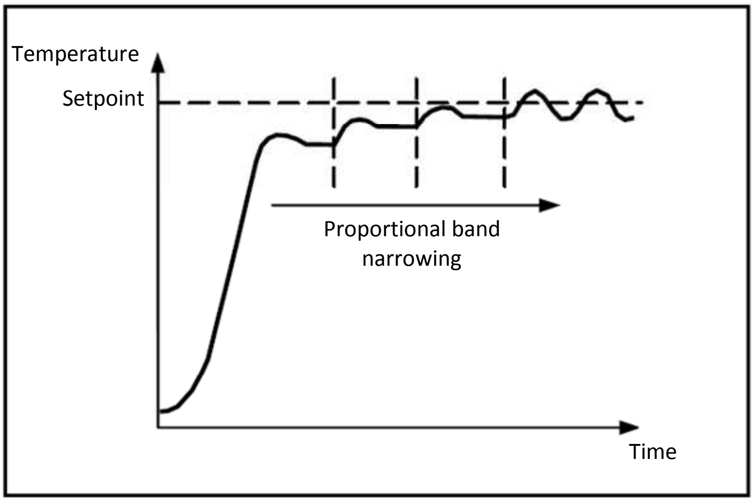
Proportional component

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

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

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.



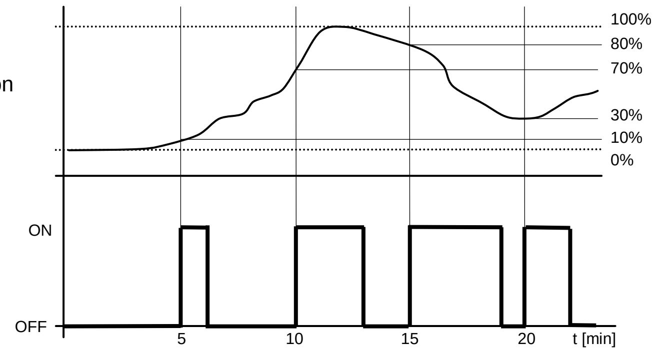


Integral component
The contribute of the integral period is proportional to the error (difference between the setpoint and the measured temperature) and its duration. The integral is the sum of the instantaneous error for every moment of time and provides the accumulated offset that should have been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

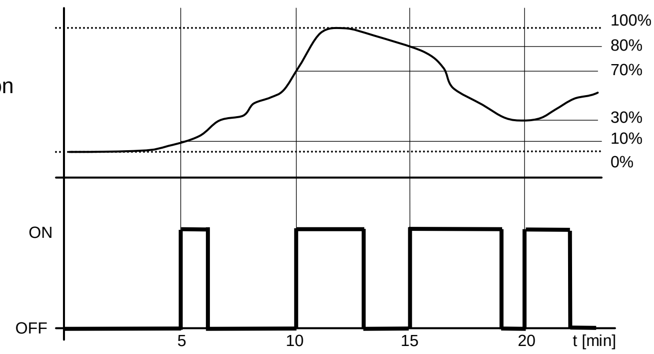
$$I_{\mathrm{out}} = K_i \int_0^t e(\tau)\, d\tau$$

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

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



The device keeps the heating system switched on for a cycle time percentage that depends on the output function of the proportional-integral control; the device continuously regulates the heating system, modulating the system turning on-off times with a duty-cycle (shown to the right along the vertical axis) that depends on the output function value calculated at every time interval equal to the cycle time. The cycle time is reinitialised every time the reference set-point is modified.
With this type of algorithm, there is no longer a hysteresis cycle on the heating device, so the inertia times (system heating and air cooling times) introduced by the 2 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.

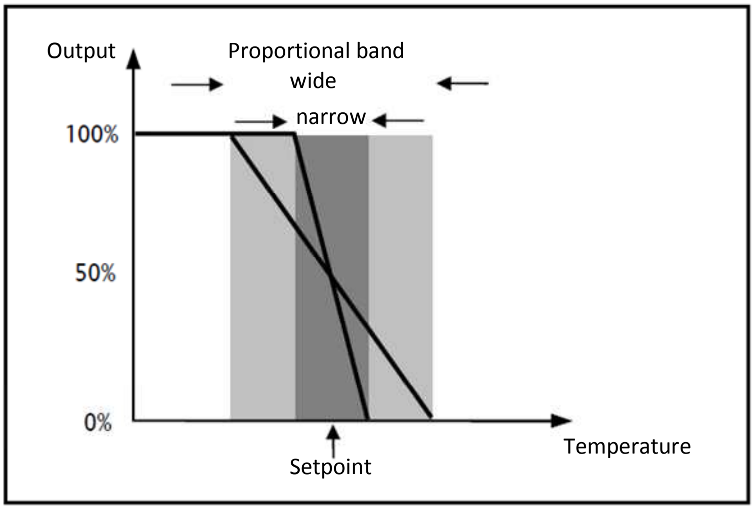
$$\mathrm{u}(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

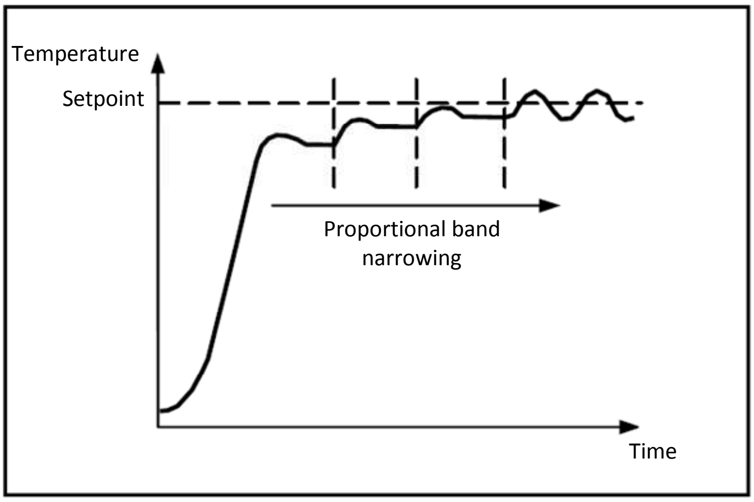
Proportional component

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

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

Once the proportional band is defined, the 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.



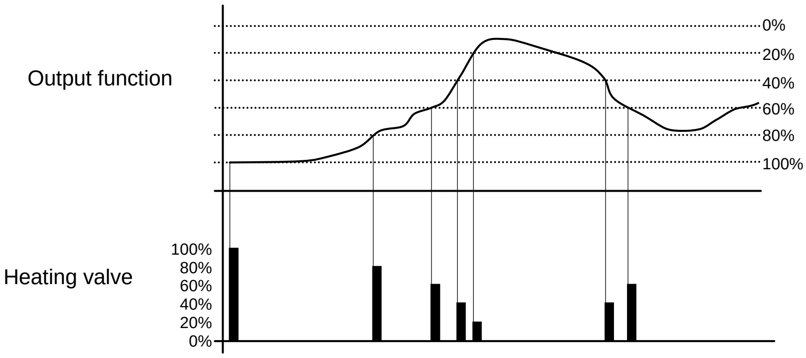


Integral component
The contribute of the integral period is proportional to the error (difference between the setpoint and the measured temperature) and its duration. The integral is the sum of the instantaneous error for every moment of time and provides the accumulated offset that should have been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

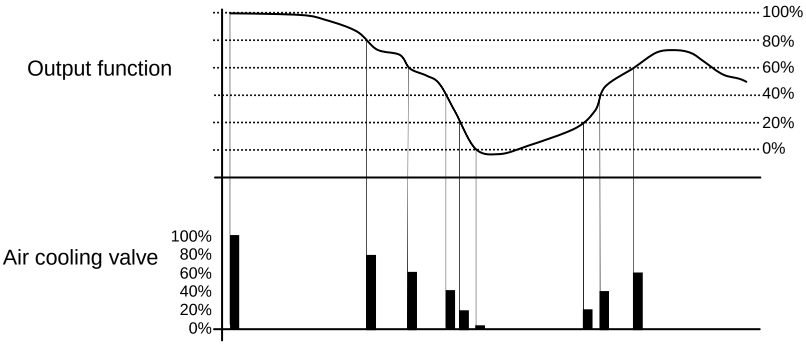
$$I_{\text{out}} = K_i \int_0^t e(\tau)\, d\tau$$

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 different levels with a distance equal to the value defined by the parameter **“Min. % variation for continuous command sending”** (in the figure, the value is **20%**) and the device continuously adjusts the heating system by sending percentage activation values of the solenoid valve (shown along the vertical axis) that depend on the intersection of the output function value calculated with a determined level. In this way, the KNX BUS will not be saturated with continuous telegrams.
With this type of algorithm, there is no longer a hysteresis cycle on the heating device, so the inertia times (system heating and air cooling times) introduced by the 2 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.

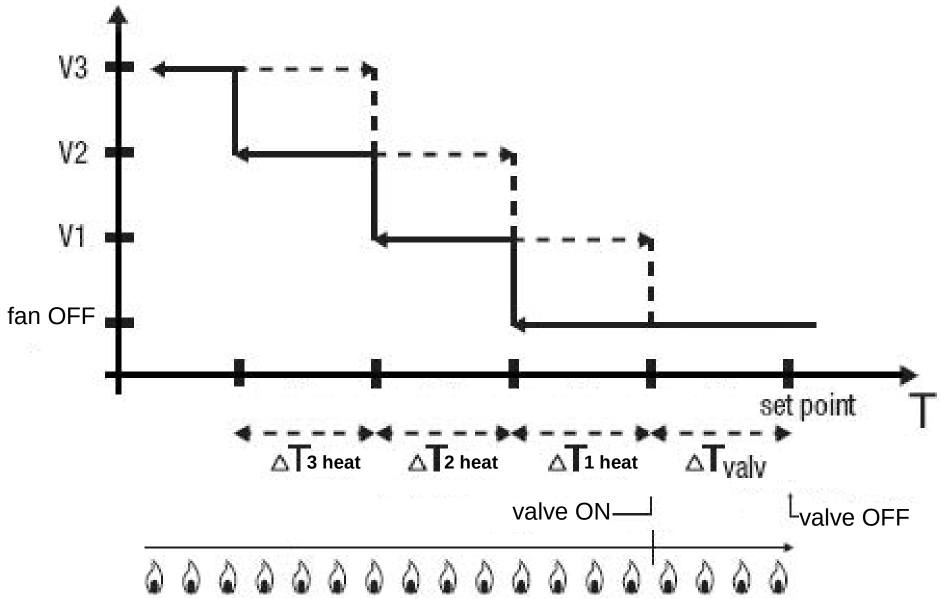


As can be seen in the figure, the device sends the commands to the actuator that manages the air cooling system based on the output function of the proportional-integral control; along the vertical axis, the 0% - 100% interval of the output function of the proportional-integral control is divided into different levels with a distance equal to the value defined by the parameter **“Min. % variation for continuous command sending”** (in the figure, the value is **20%**) and the device continuously adjusts the air cooling system by sending percentage activation values of the solenoid valve (shown along the vertical axis) that depend on the intersection of the output function value calculated with a determined level. In this way, the KNX BUS will not be saturated with continuous telegrams.
With this type of algorithm, there is no longer a hysteresis cycle on the cooling device, so the inertia times (system air cooling and heating times) introduced by the 2 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); substantially, this means that each stage corresponds to a speed and when the difference between the measured temperature and the setpoint that was set causes a certain speed to be turned on, this means that before turning on the new speed, the other two must absolutely be turned off.

HEATING

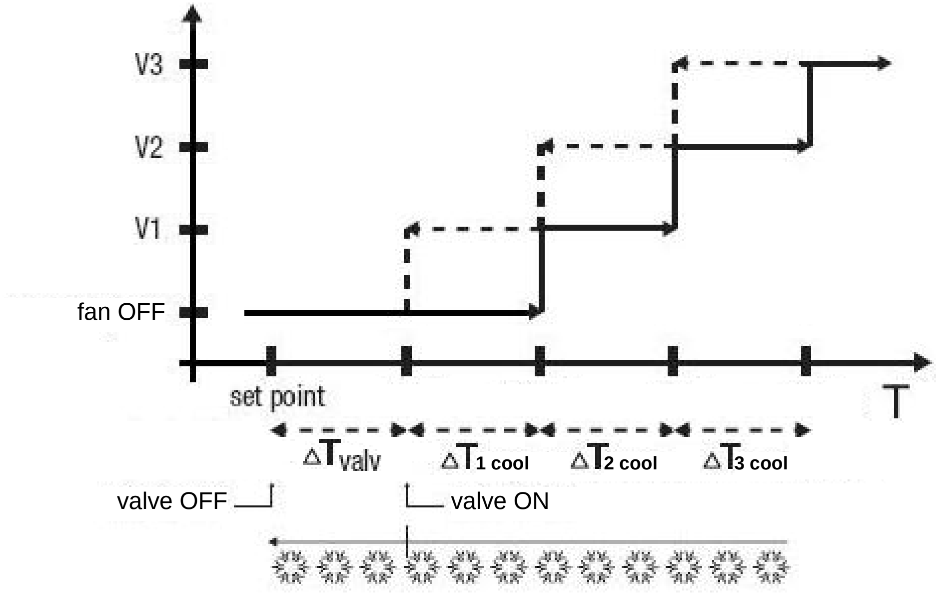


The figure refers to the control of the fancoil speeds with three operating stages for heating. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine its activation and deactivation. The thresholds are determined by values set for the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

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

With regard to the heating solenoid valve, once the measured temperature is lower than the value “setpoint-ΔT_{valv}”, the sensor 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 below refers to the control of the speeds of a fan coil with three operating stages for air cooling. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine its activation and deactivation. The thresholds are determined by values set for the various regulation differentials, and can be summarised as follows:

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

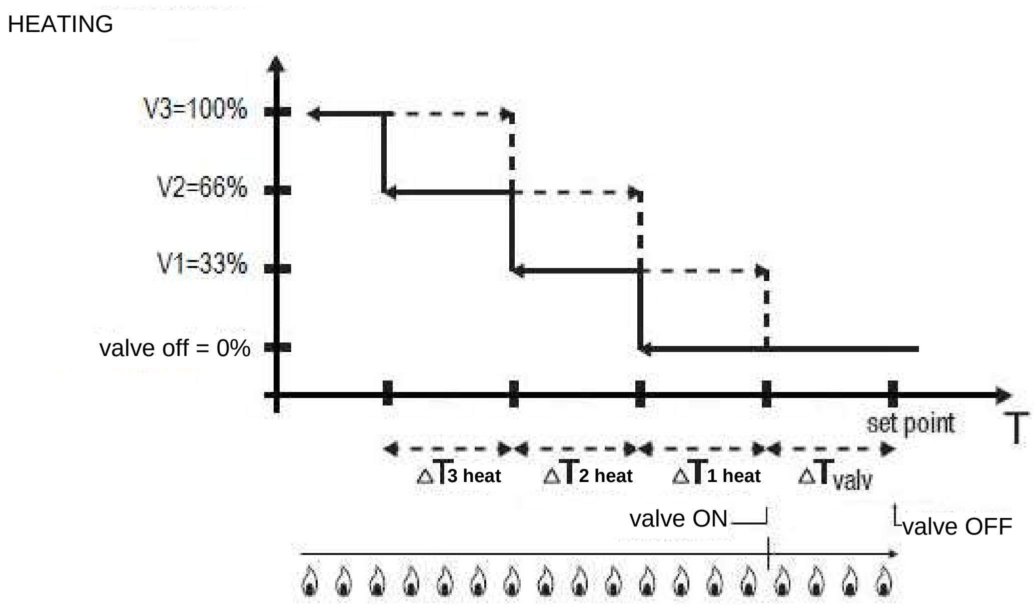
With regard to the air cooling solenoid valve, once the measured temperature is higher than the value "setpoint+ΔT_{valv}", the sensor sends the activation command to the solenoid valve that manages the air cooling system; the solenoid valve is deactivated when the detected temperature reaches the fixed 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 sensor 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.

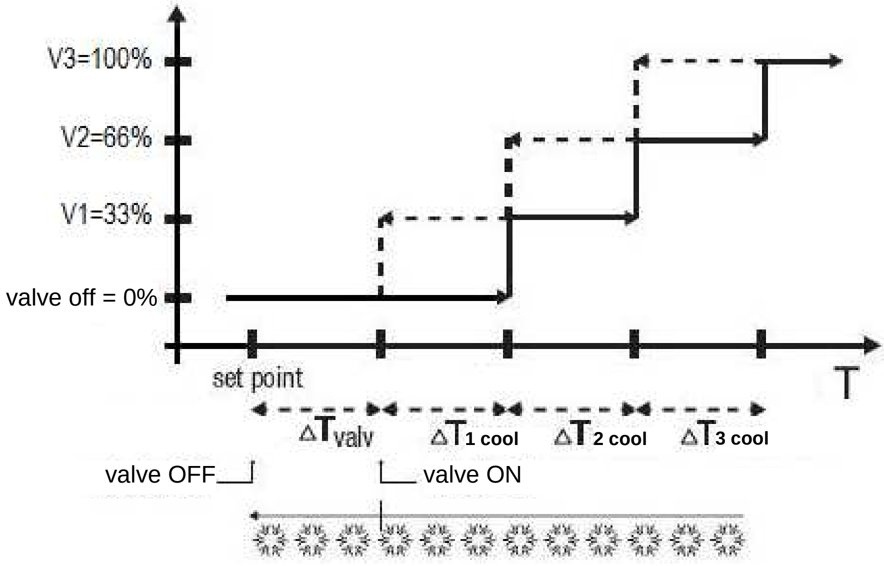


The figure refers to the control of the fancoil speeds with three operating stages for heating. The chart shows that each stage has a hysteresis cycle, and each speed is associated with two thresholds that determine the sending of the associated value. The thresholds are determined by values set for 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-ΔT_{valv}-$\Delta T_{1\ heat}$" and turned off (sending the value "valve off") when the temperature value reaches the value "setpoint-ΔT_{valv}" (or the "setpoint" value if $\Delta T_{1\ heat}$=0). The first speed is also turned off when a higher speed needs to be activated
- Speed V2 (2nd stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$" and turned off (sending speed V1) when the temperature value reaches the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$". The second speed is also turned off when the V3 speed needs to be activated
- Speed V3 (3rd stage): the speed is turned on when the temperature value is lower than the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$-$\Delta T_{3\ heat}$ and turned off (sending value V2) when the temperature value reaches the value "setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$"

With regard to the heating solenoid valve, once the measured temperature is lower than the value "setpoint-ΔT_{valv}", the sensor 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 below refers to the control of the speeds of a fan coil 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+ΔT_{valv}+ΔT_{1cool}" and turned off (sending the value "valve off") when the temperature value reaches the value "setpoint+ΔT_{valv}" (or the "setpoint" value if $\Delta T_{1\ cool}$=0). The first speed is also turned off when a higher speed needs to be activated
- Speed V2 (2nd stage): the speed is turned on when the temperature value is higher than the value "setpoint+ΔT_{valv}+ΔT_{1cool}+ΔT_{2cool} and turned off (sending value V1) when the temperature value reaches the value "setpoint+ΔT_{valv}+$\Delta T_{1\ cool}$". The second speed is also turned off when the V3 speed needs to be activated
- Speed V3 (3rd stage): the speed is turned on when the temperature value is higher than the value "setpoint+ΔT_{valv}+ΔT_{1cool}+ΔT_{2cool}+ΔT_{3cool} and turned off (sending the value V2) when the temperature value reaches the value "setpoint+ΔT_{valv}+ΔT_{1cool}+ΔT_{2cool}"

With regard to the air cooling solenoid valve, once the measured temperature is higher than the value "setpoint+ΔT_{valv}", the sensor 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 sensor 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. The basic structure of the menu is as follows:

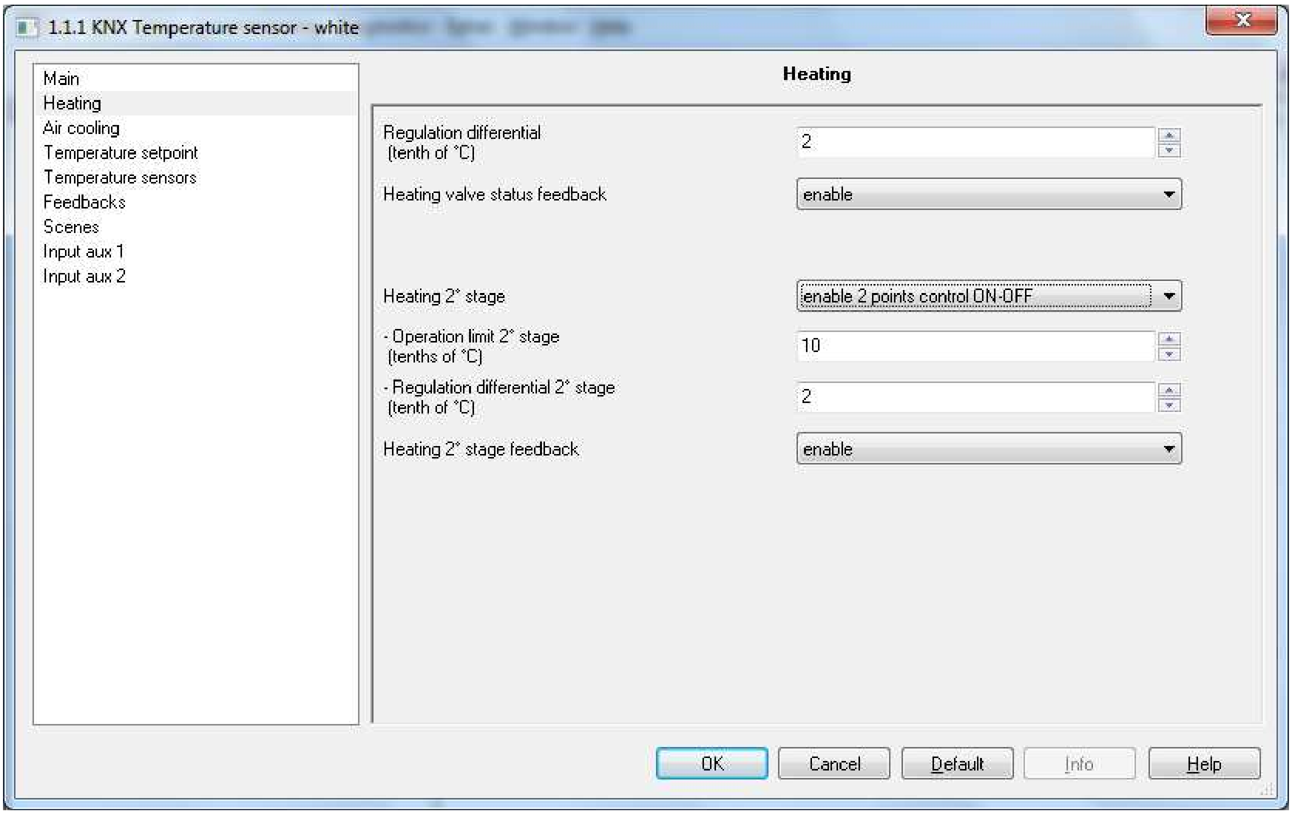


Fig. 4.1

4.1 Parameters

➢ *4.1.1 Regulation differential (tenth of °C)*

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

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

➢ *4.1.2 Select heating system*

The parameter **“Select heating system”** is used to automatically measure the operating parameters (Proportional band and Integration time) of the proportional integral algorithm based on the selected heating system. The values that can be set are:

- hot water heating
- **floor heating (default value)**
- fan coil unit
- electric heating
- customised

Selecting **hot water heating**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **5.0 °C** and **150** are displayed.

Selecting **floor heating**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **5.0 °C** and **240** are displayed.
Selecting **fan coil unit**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **4.0 °C** and **90** are displayed.

Selecting **electric heating**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified, and the values **4.0 °C** and **100** are displayed.

Selecting **customised**, the parameters **“Proportional band”** and **“Integration time (minutes)”** will be displayed but cannot be modified.
It is not necessary to store the parameter **“Select heating system”** in the memory.

The parameter **“Proportional band”** is used to set the width of the PWM proportional band of the heating **PWM proportional integral** or **continuous proportional integral** control algorithm, already mentioned in the Control algorithms paragraph, which, when subtracted from the value of the setpoint that was set, determines the lower limit of the proportional band used for the proportional integral control. The values that can be set are:

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

The parameter **“Integration time”** is used to set the contribution of the integral action in the proportional integral control (see the Control algorithms section). The values that can be set are:

- from 1 minute to 250 minutes with steps of 1 plus the value “no integral” (255), **60 (default value)**

Selecting **no integral**, the integral component is zero and the pure effect of proportional control is obtained.

The parameter **“Cycle time”** is used to set the value of the period within which the device carries out PWM modulation, modifying the duty-cycle. The values that can be set are:

- 5 minutes
- 10 minutes
- 15 minutes
- **20 minutes** **(default value)**
- 30 minutes
- 40 minutes
- 50 minutes
- 60 minutes

The parameter **“Min. % variation for continuous command sending”** 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 Control algorithms section); there is not a fixed number of proportional sub-bands, as this 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 parameter **“Fancoil valve management”** 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)**
- two points 0%-100%

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

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

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

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

- 1

setting this value, the number of stages for controlling the fancoil speeds is 1; this setting displays the parameters **“Speed 1 regulation differential (tenth of °C)”** and **“Speed 1 inertia time (seconds)”**.
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 communication object ***Heating fan V1 switching*** (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 communication object ***Heating fancoil speed % command*** (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:

Fancoil speed	***Sent percentage value***
fan OFF	0%
first speed (V1)	100%

- 2

Setting this value, the number of stages for controlling the fancoil speeds is 2; this setting displays the parameters **“Speed 1 regulation differential (tenth of °C)”** , **“Speed 2 regulation differential (tenth of °C)”** , **“Speed 1 inertia value (seconds)”** and **“Speed 2 inertia value (seconds)”**.
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 communication objects ***Heating fan V1 switching*** and ***Heating fan V2 switching*** (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 communication object ***Heating fancoil speed % command*** (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:

Fancoil speed	***Sent percentage value***
fan OFF	0%
first speed (V1)	50%
second speed (V2)	100%

- **3 (default value)**

Setting this value, the number of stages for controlling the fancoil speeds is 3; this setting displays the parameters **“Speed 1 regulation differential (tenth of °C)”** , **“Speed 2 regulation differential (tenth of °C)”** , **“Speed 3 regulation differential (tenth of °C)”** , **“Speed 1 inertia time (seconds)”**, **“Speed 2 inertia time (seconds)”** and **“Speed 3 inertia time (seconds)”**.
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 communication objects ***Heating fan V1 switching*** , ***Heating fan V2 switching*** and ***Heating fan V3 switching*** (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 communication object ***Heating fancoil speed % command*** (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:

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

The parameter **“Speed 1 regulation differential (tenth of °C)”** is used to set the value of the regulation differential of the first speed of the heating control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control** , already mentioned in the Control algorithms section; this value, subtracted from the “setpoint-ΔT_{valv}” determines the value of the threshold,under which speed 1 of the fancoil is turned on. The values that can be set are:

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

Setting the value **0** obtains the condition “$\Delta T_{1\ heat} = \Delta T_{valv}$” for which the value of the speed 1 activation threshold is “setpoint-ΔT_{valv}” and the off value is “setpoint”.

The parameter **“Speed 2 regulation differential (tenth of °C)”** is used to set the value of the regulation differential of the second speed of the heating control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control** , already mentioned in the Control algorithms section; this value, subtracted from the value “setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$” determines the value of the threshold, under which speed 2 of the fancoil is turned on. The values that can be set are:

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

The parameter **“Speed 3 regulation differential (tenth of °C)”** is used to set the value of the regulation differential of the third speed of the heating control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control** , already mentioned in the Control algorithms section; this value, subtracted from the value “setpoint-ΔT_{valv}-$\Delta T_{1\ heat}$-$\Delta T_{2\ heat}$” determines the value of the threshold, under which speed 3 of the fancoil is turned on. The values that can be set are:

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

When, according to the "fancoil with speed control” algorithm, the device must turn on any speed and speed 1 is on, a delay can be inserted between the moment in which feedback is received that speed 1 is turned off (or the moment the command to turn off speed 1 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **“Speed 1 inertia time (seconds)”** 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, with steps of 1

When, according to the "fancoil with speed control” algorithm, the device must turn on any speed and speed 2 is on, a delay can be inserted between the moment in which feedback is received that speed 2 is turned off (or the moment the command to turn off speed 2 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **“Speed 2 inertia time (seconds)”** 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, with steps of 1

When, according to the "fancoil with speed control” algorithm, the device must turn on any speed and speed 3 is on, a delay can be inserted between the moment in which feedback is received that speed 3 is turned off (or the moment the command to turn off speed 3 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **“Speed 3 inertia time (seconds)”** 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, with steps of 1

Defining the inertia times is useful for preserving the integrity of the fancoil, because the fact of turning off the power supply to the motor (turning off the actuator) of a fancoil speed does not guarantee that current is no longer circulating in the winding and the instantaneous supply of power to another winding could damage the fancoil (simultaneous powering of multiple windings).

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

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

Select **disable** to view the parameter **“Command repetition period with disabled feedback”**; select **enable** to view the communication object ***Heating valve status feedback*** (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 BUS voltage is restored, the device sends the read request command via the object ***Heating valve status feedback*** or ***Heating valve % feedback*** 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 sensor, 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 sensor, 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. ± 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 sensor, 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 parameter **“Command repetition period with disabled feedback”** is used to define the time range of the cyclical sending. The values that can be set are:

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

If the control algorithm is fancoil, more important than the valve feedback is the possibility to receive feedback about the fancoil speed ON status. 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 sensor, 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 sensor 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 parameter **“Fancoil speed status feedback”** is used to enable the device to receive feedback from the actuator that commands the fancoil speeds. The values that can be set are:

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

Select **disable** to view the parameter **“Fancoil speed command repetition period”**; select **enable** to view the following communication objects:

- if the control algorithm is **fancoil with ON-OFF speed control**, the communication objects ***Heating fan V1 status feedback***, ***Heating fan V2 status feedback*** *and* ***Heating fan V3 status feedback*** (Data Point Type: 1.001 DPT_Switch) according to the number of fancoil speeds.

if the valve control algorithm is **fancoil with continuous speed control**, the communication object ***Heating fancoil continuous control feedback*** (Data Point Type: 5.001 DPT_Scaling) independently of the number of fancoil speeds. The feedback received via this object can be different from the sent command by approx.

± 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 sensor, the operation is guaranteed and the process of periodically sending the command is not triggered.

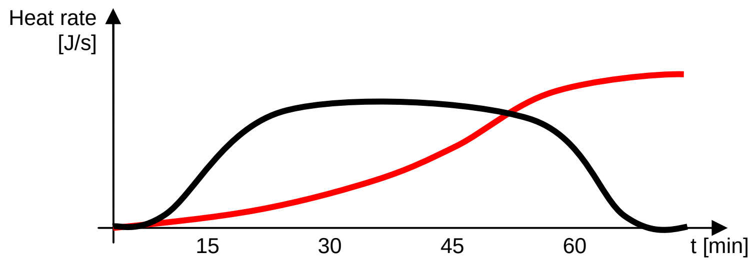
When BUS voltage is restored, the device sends the read request command via the ***Heating fan V1 status feedback***, ***Heating fan V2 status feedback***, ***Heating fan V3 status feedback*** or ***Heating valve % feedback*** objects to be updated about the activation status of the fancoil speeds.
If the fancoil feedback is disabled, deactivation commands of the inactive speeds must be sent for every speed activation command; in the same manner, every speed deactivation command must be sent together with deactivation commands for the other speeds.

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

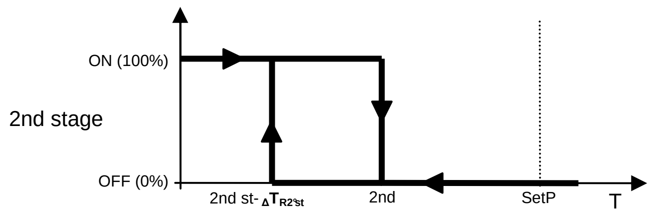
- no repetition
- 1 minute
- 2 minutes
- 3 minutes
- 4 minutes
- **5 minutes (default value)**

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

Some heating systems (for example, 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-$\Delta T_{R2nd\ st}$" under 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 Heating 2nd stage*

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 parameters **"Operating limit 2° stage (tenths of °C)"** , **"Regulation differential 2° stage (tenth of °C)"** and **"Heating 2° stage feedback"**. Selecting **enable 2 points control ON-OFF** displays the communication object ***Heating 2° stage switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams; selecting **enable 2 points control 0%-100%** displays the communication object ***Heating 2° stage % command*** (Data Point Type: 5.001 DPT_Scaling) via which the device sends the command telegrams.

The parameter **"Operation limit 2° stage (tenths of °C)"** is used to define the operation threshold of the heating 2nd stage. 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) above which the latter is deactivated; the values that can be set are:

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

The parameter **"Regulation differential 2° stage (tenths of °C)"** is used to set the value of the regulation differential of the heating 2nd stage 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) under 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 with steps of 1, 2 (default value)

As for the basic heating algorithm, the parameter **"Heating 2° stage feedback"** is used to enable the device to receive feedback from the actuator that controls the heating 2nd stage; 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 parameter **"Command repetition period 2° stage without feedba ck"** ; select **enable** to view the communication object ***Heating 2° stage feedback*** (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 BUS voltage is restored, the device sends the read request command via the object ***Heating 2° stage feedback*** or ***Heating 2° stage valve % feedback*** 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 sensor, that forces its status, modifying it. In this case, the device repeats the switching command to realign the status of the actuator with the one determined by the control logic, triggering the process for waiting for confirmation and repeating the command until the confirmation is received.
With the heating 2nd stage 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 parameter **"2° stage command repetition period without feedbac k"** 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. The structure of the menu is as follows:

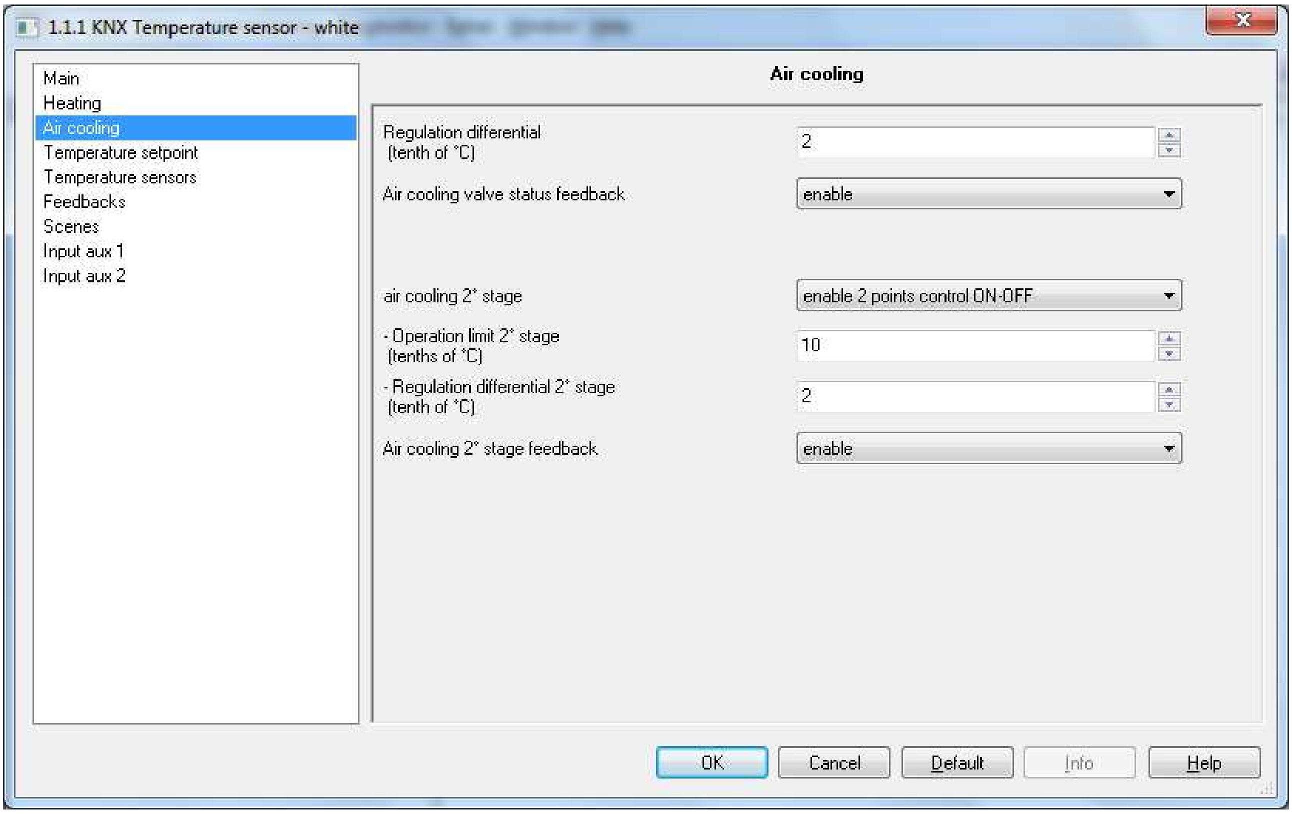


Fig. 5.1

5.1 Parameters

➢ *5.1.1 Regulation differential (tenth of °C)*

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

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

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)**
- fan coil unit
- customised

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

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

Selecting **customised**, the parameters **"Proportional band"** and **"Integration time (minutes)"** will be displayed but cannot be modified.
It is not necessary to store the parameter **"Select air cooling system"** in the memory.

The parameter **"Proportional band"** is used to set the width of the PWM proportional band of the air cooling **PWM proportional integral** or **continuous proportional integral** control algorithm, already mentioned in the Control algorithms paragraph, which, when added to the value of the setpoint that was set, determines the upper limit of the proportional band used for the proportional integral control. The values that can be set are:

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

The parameter **"Integration time (minutes)"** is used to set the contribution of the integral action in the proportional integral control (see the Control algorithms section). The values that can be set are:

- from 1 minute to 250 minutes with steps of 1 plus the value "no integral" (255), **60 (default value)**

Selecting **no integral**, the integral component is zero and the pure effect of proportional control is obtained.

The parameter **"Cycle time"** is used to set the value of the period within which the device carries out PWM modulation, modifying the duty-cycle. The values that can be set are:

- 5 minutes
- 10 minutes
- 15 minutes
- **20 minutes (default value)**
- 30 minutes
- 40 minutes
- 50 minutes
- 60 minutes

The parameter **“Min. % variation for continuous command sending”** 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 Control algorithms section); there is not a fixed number of proportional sub-bands, as this 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 parameter **“Fancoil valve management”** 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 communication object ***Air cooling valve switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams; selecting **2 points 0%-100%** displays the communication object ***Air cooling valve % command*** (Data Point Type: 5.001 DPT_Scaling) via which the device sends the command telegrams.

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

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

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

- 1

setting this value, the number of stages for controlling the fancoil speeds is 1; this setting displays the parameters **“Speed 1 regulation differential (tenth of °C)”** and **“Speed 1 inertia time (seconds)”**.
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 communication object ***Air cooling fan V1 switching*** (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 communication object ***Air cooling fancoil speed % command*** (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:

Fancoil speed	***Sent percentage value***
fan OFF	0%
first speed (V1)	100%

- 2

Setting this value, the number of stages for controlling the fancoil speeds is 2; this setting displays the parameters **“Speed 1 regulation differential (tenth of °C)”**, **“Speed 2 regulation differential (tenth of °C)”**, **“Speed 1 inertia value (seconds)”** and **“Speed 2 inertia value (seconds)”**.
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 communication objects ***Air cooling fan V1 switching*** and ***Air cooling fan V2 switching*** (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 communication object ***Air cooling fancoil speed % command*** (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:

Fancoil speed	*Sent percentage value*
fan OFF	0%
first speed (V1)	50%
second speed (V2)	100%

- **3 (default value)**

Setting this value, the number of stages for controlling the fancoil speeds is 3; this setting displays the parameters **"Speed 1 regulation differential (tenth of °C)"** , **"Speed 2 regulation differential (tenth of °C)"** , **"Speed 3 regulation differential (tenth of °C)"** , **"Speed 1 inertia time (seconds)"**, **"Speed 2 inertia time (seconds)"** and **"Speed 3 inertia time (seconds)"**.
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 communication objects ***Air cooling fan V1 switching***, ***Air cooling fan V2 switching*** and ***Air cooling fan V3 switching*** (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 ***1 byte (value %)***, this enables the communication object ***Air cooling fancoil speed % command*** (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:

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

The parameter **"Speed 1 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the first speed of the air cooling control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control** , already mentioned in the Control algorithms section; this value, added to the value "setpoint+ΔT_{valv}" determines the value of the threshold,under which speed 1 of the fancoil is turned on. The values that can be set are:

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

Setting the value **0** obtains the condition "$\Delta T_{1\ cool} = \Delta T_{valv}$" for which the value of the speed 1 activation threshold is "setpoint+ΔT_{valv}" and the off value is "setpoint".

The parameter **"Speed 2 regulation differential (tenth of °C)"** is used to set the value of the regulation differential of the second speed of the air cooling control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control** , already mentioned in the Control algorithms section; this value, subtracted from the value "setpoint+$\Delta T_{valv} + \Delta T_{1\ cool}$" determines the value of the threshold, under which speed 2 of the fancoil is turned on. The values that can be set are:

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

The parameter **“Speed 3 regulation differential (tenth of °C)”** is used to set the value of the regulation differential of the third speed of the air cooling control algorithm **fancoil with ON-OFF speed control** or **fancoil with continuous speed control** , already mentioned in the Control algorithms section; this value, subtracted from the value “setpoint+ΔT_{valv}+$\Delta T_{1\ cool}$+$\Delta T_{2\ cool}$” determines the value of the threshold, under which speed 3 of the fancoil is turned on. The values that can be set are:

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

When, according to the "fancoil with speed control” algorithm, the device must turn on any speed and speed 1 is on, a delay can be inserted between the moment in which feedback is received that speed 1 is turned off (or the moment the command to turn off speed 1 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **“Speed 1 inertia time (seconds)”** 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, with steps of 1

When, according to the "fancoil with speed control” algorithm, the device must turn on any speed and speed 2 is on, a delay can be inserted between the moment in which feedback is received that speed 2 is turned off (or the moment the command to turn off speed 2 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **“Speed 2 inertia time (seconds)”** 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, with steps of 1

When, according to the "fancoil with speed control” algorithm, the device must turn on any speed and speed 3 is on, a delay can be inserted between the moment in which feedback is received that speed 3 is turned off (or the moment the command to turn off speed 3 is sent if the fancoil speed feedback is disabled) and the instant in which the command for turning on the new speed is sent; the parameter **“Speed 3 inertia time (seconds)”** 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, with steps of 1

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

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

Select **disable** to view the parameter **“Command repetition period with disabled feedback”**; select **enable** to view the communication object ***Air cooling valve status feedback*** (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 BUS voltage is restored, the device sends the read request command via the object ***Air cooling valve status feedback*** or ***Air cooling valve % feedback*** 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 sensor, 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 sensor, 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. ± 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 sensor, 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 parameter **“Command repetition period with disabled feedback”** is used to define the time range of the cyclical sending; The values that can be set are:

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

If the control algorithm is fancoil, more important than the valve feedback is the possibility to receive feedback about the fancoil speed ON status. 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 sensor, 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 sensor 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 parameter **“Fancoil speed status feedback”** is used to enable the device to receive feedback from the actuator that commands the fancoil speeds. The values that can be set are:

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

Select **disable** to view the parameter **“Fancoil speed command repetition period”** and the following communication objects:

- if the control algorithm is **fancoil with ON-OFF speed control**, the communication objects ***Air cooling fan V1 status feedback***, ***Air cooling fan V2 status feedback*** *and* ***Air cooling fan V3 status feedback*** (Data Point Type: 1.001 DPT_Switch) according to the number of fancoil speeds.
- if the valve control algorithm is **fancoil with continuous speed control**, the communication object ***Air cooling fancoil continuous control feedback*** (Data Point Type: 5.001 DPT_Scaling) independently of the number of fancoil speeds. The feedback received via this object can be different from the sent command by approx. ± 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 sensor, 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 objects ***Air cooling fan V1 status feedback***, ***Air cooling fan V2 status feedback***, ***Air cooling fan V3 status feedback*** or ***Air cooling valve % feedback*** objects to be updated about the activation status of the fancoil speeds.

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

- no repetition
- 1 minute
- 2 minutes
- 3 minutes

- 4 minutes
- **5 minutes** **(default value)**

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

➢ *5.1.3 Air cooling 2nd stage*

This is used to enable and define the control algorithm for the second air cooling 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 parameters **"Operating limit 2° stage**, **"Regulation differential 2° stage (tenth of °C)"** and **"Air cooling 2° stage feedback"**. Selecting **enable 2 points control ON-OFF** displays the communication object ***Air cooling 2° stage switching*** (Data Point Type: 1.001 DPT_Switch) via which the device sends the command telegrams; selecting **enable 2 points control 0%-100%** displays the communication object ***Air cooling 2° stage % command*** (Data Point Type: 5.001 DPT_Scaling) via which the device sends the command telegrams.

The parameter **"Operation limit 2° stage (tenths of °C)"** is used to define the operation threshold of the air cooling 2nd stage. 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, with steps of 1

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

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

As for the basic air cooling algorithm, the parameter **"Air cooling 2° stage feedback"** is used to enable the device to receive feedback from the actuator that controls the air cooling 2nd stage; 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 parameter **"Command repetition period 2° stage without feedba ck"**; select **enable** to view the communication object ***Air cooling 2° stage feedback*** (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 BUS voltage is restored, the device sends the read request command via the object ***Air cooling 2° stage feedback*** or ***Air cooling 2° stage % feedback*** to be updated about the status of the air cooling 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 sensor, that forces its status, modifying it. In this case, the device repeats the switching command to realign the status of the actuator with the one determined by the control logic, triggering the process for waiting for confirmation and repeating the command until the confirmation is received.

With the air cooling 2nd stage 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 parameter **"2° stage command repetition period without feedback"** is used to define the time range of the cyclical sending; the values that can be set are:

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

6 *"Temperature setpoint"* menu

This contains the parameters used to configure the setpoint values of the various temperature adjustment modes of the two functioning types. The structure of the menu is as follows:

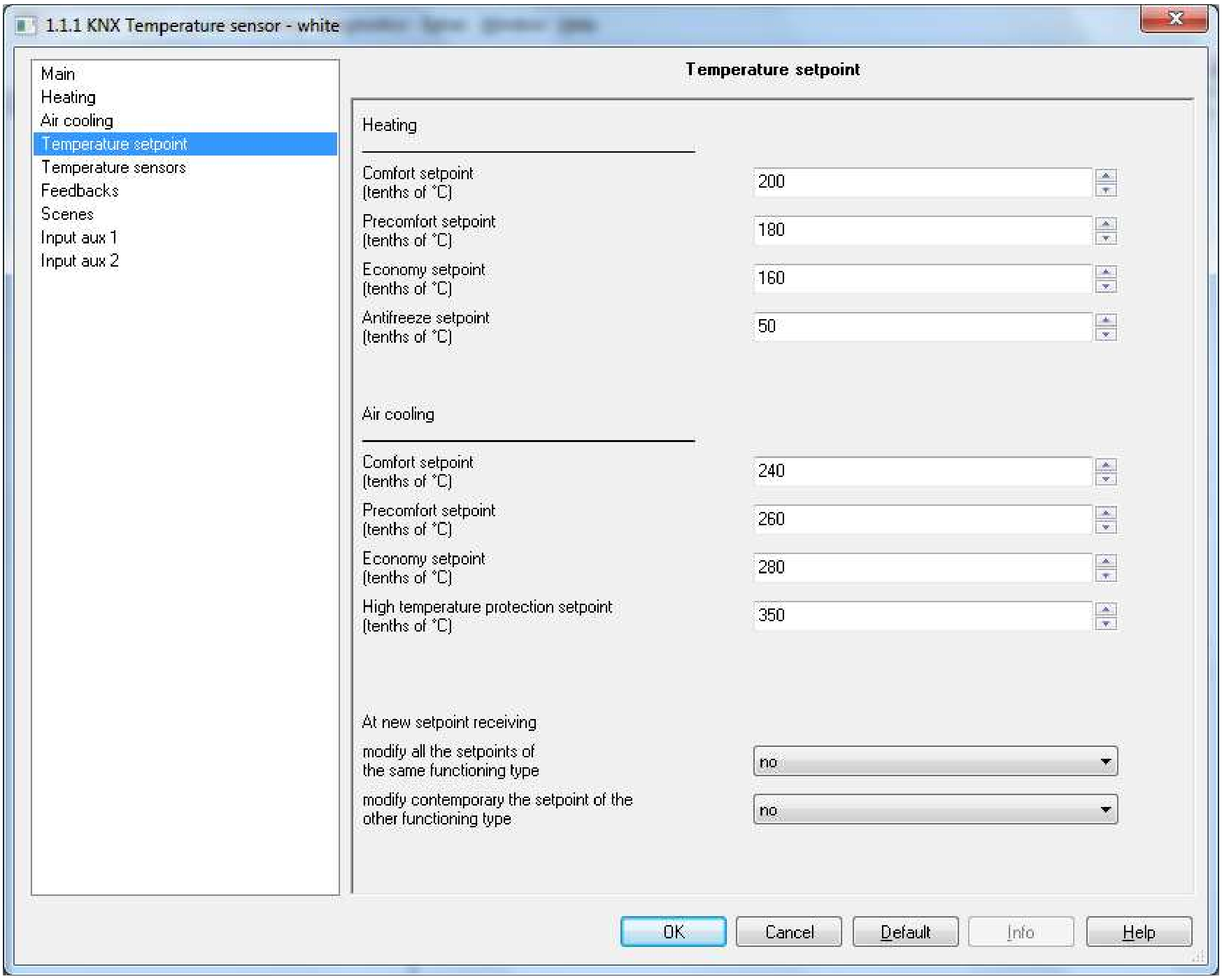


Fig. 6.1

6.1 Parameters

➢ 6.1.1 Heating section

The parameter **"Comfort setpoint (tenth of °C)"** is used to set the setpoint value of the COMFORT mode for HEATING operation; the values that can be set are:

- from 50 to 400 with steps of 1, **200 (default value)**

Remember that when setting this value, there is a constraint that it must be higher than the value set for the **"Precomfort setpoint (tenth of °C)"** for heating operation.
Remember that this value can 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 parameter **"Precomfort setpoint (tenth of °C)"** is used to set the setpoint value of the PRECOMFORT mode for HEATING operation; the values that can be set are:

- from 50 to 400 with steps of 1, **180 (default value)**

Remember than when setting this value, there is a constraint that it must lie between the value set for the **"Comfort setpoint (tenth of °C)"** and the value set for the **"Economy setpoint (tenth of °C)"** for heating operation.
Remember that this value can always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The parameter **"Economy setpoint (tenth of °C)"** is used to set the setpoint value of the ECONOMY mode for HEATING operation; the values that can be set are:

- from 50 to 400 with steps of 1, **160 (default value)**

Remember than when setting this value, there is a constraint that it must lie between the value set for the **"Precomfort setpoint (tenth of °C)"** and the value set for the **"Antifreeze setpoint (tenth of °C)"** for heating operation.
Remember that this value can always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

The parameter **"Antifreeze setpoint (tenth of °C)"** is used to set the setpoint value of the OFF mode for HEATING operation; the values that can be set are:

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

Remember that when setting this value, there is a constraint that it must be lower than the value set for the **"Economy setpoint (tenth of °C)"** for heating operation.
Remember that this value can always be modified by the user via the relative parameter in the device's local navigation menu and, if remote setpoint control is enabled, it can be modified by a BUS telegram on the communication object assigned to it.

➢ 6.1.2 Air cooling section

The parameter **"Comfort setpoint (tenth of °C)"** is used to set the setpoint value of the COMFORT mode for AIR COOLING operation; the values that can be set are:

- from 50 to 400 with steps of 1, **240 (default value)**

Remember that when setting this value, there is a constraint that it must be lower than the value set for the **“Precomfort setpoint (tenth of ℃)”** 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 parameter **“Precomfort setpoint (tenth of ℃)”** is used to set the setpoint value of the PRECOMFORT mode for AIR COOLING operation; the values that can be set are:

- from 50 to 400 with steps of 1, **260 (default value)**

Remember than when setting this value, there is a constraint that it must lie between the value set for the **“Comfort setpoint (tenth of ℃)”** and the value set for the **“Economy setpoint (tenth of ℃)”** 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 parameter **“Economy setpoint (tenth of ℃)”** is used to set the setpoint value of the ECONOMY mode for AIR COOLING operation; the values that can be set are:

- from 50 to 400 with steps of 1, **280 (default value)**

Remember than when setting this value, there is a constraint that it must lie between the value set for the **“Precomfort setpoint (tenth of ℃)”** and the value set for the **“High temperature protection setpoint (tenth of ℃)”** 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 parameter **“High temperature protection setpoint (tenth of ℃) ”** is used to set the setpoint value of the OFF mode for AIR COOLING operation; the values that can be set are:

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

Remember that when setting this value, there is a constraint that it must be higher than the value set for the **“Economy setpoint (tenth of ℃)”** 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.

➢ *6.1.3 At new setpoint receiving modify contemporary the setpoint of the other functioning type*

By modifying the setpoint of a particular HVAC mode of a functioning type (if the remote control type is HVAC mode) or operating setpoint (if the remote control type is setpoint), it may be useful to modify in the same manner the setpoint of the same mode for the opposite functioning type (especially when the device functioning type is modified autonomously via the "dead" zone). EXAMPLE: Heating Comfort setpoint = 20 ℃ and Air cooling Comfort setpoint = 24 ℃; if a Heating Comfort setpoint value equal to 21.5 ℃ is received, also the Air cooling Comfort setpoint is automatically modified and set equal to 25.5 ℃.
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

➢ *6.1.4 At new setpoint receiving modify all the setpoints of the same functioning type*

In the same manner, if the remote control type is HVAC mode, may be useful to modify manner the setpoints of the same functioning type (excluding the OFF mode) after only one of them has been modified. EXAMPLE: Heating Comfort setpoint = 20 °C, Heating Precomfort setpoint = 18 °C and Heating Economy setpoint = 16 °C; if a Heating Comfort setpoint equ al to 21.5 °C is received, then automatically the H eating Precomfort setpoint becomes 19.5 °C and the Heating Economy setpoint becomes 17.5 °C.
The parameter that makes it possible to enable the contemporary modification of the setpoints for the same functioning type is **"At new setpoint receiving modify all 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: one KNX external sensor and one NTC external sensor. The structure of the menu is as follows:

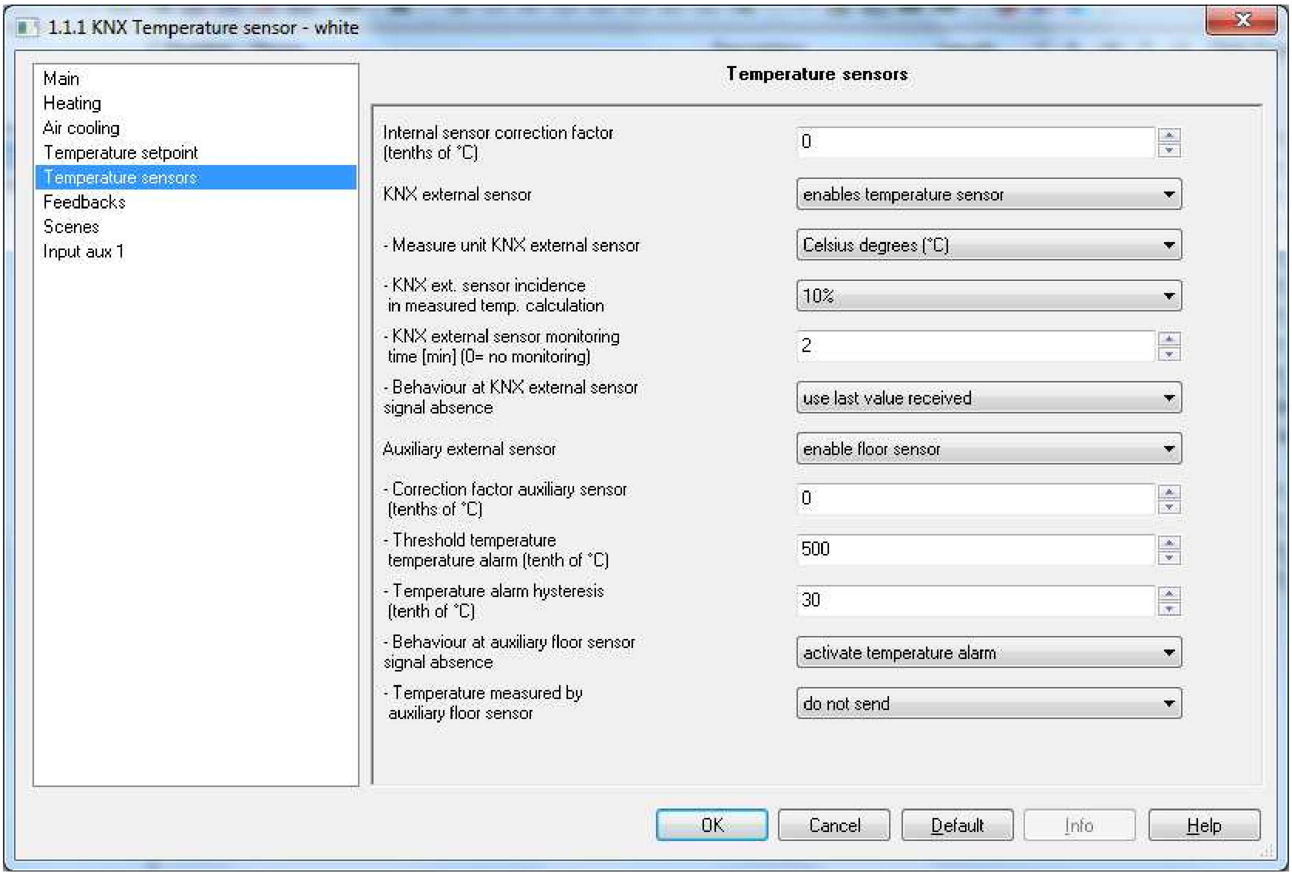


Fig. 7.1

7.1 Parameters

➢ 7.1.1 Internal sensor correction factor (tenths of °C)

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

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

➢ 7.1.2 KNX external sensor

This 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)**
- enables temperature sensor
- enable floor sensor

Selecting **enables temperature sensor** displays the parameters **"Measure unit KNX external sensor"**, **"KNX external sensor incidence in measured temp. calculation"**, **"KNX external sensor monitoring time [min] (0=no monitoring)"** and **"Behaviour at KNX external sensor signal absence"** and the communication object ***KNX external sensor input*** which is used to receive the temperature measured by the external sensor.

Selecting **enable floor sensor** displays the parameters **"Measure unit KNX floor sensor"**, **"Threshold temperature temperature alarm (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"** and the communication object ***KNX floor sensor input*** used to receive the temperature measured by the external sensor.

In both cases, when 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 object ***KNX external sensor input*** or ***KNX floor sensor input***, storing the received value.

➢ 7.1.3 Measure unit external sensor

The parameter **"Measure unit KNX external sensor"** (or **"Measure unit KNX floor sensor"**) is used to set the unit of measure used to decode the information received via the communication object ***KNX external sensor input*** (or ***KNX floor sensor input***); the values that can be set are:

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

The value set for this parameter changes the code of the communication object ***KNX external sensor input*** (or ***KNX floor sensor input***): *9.001 DPT_Value_Temp* if the value is **degrees Celsius (°C)**, *9.002 DPT_Value_Tempd* if the value is **degrees Kelvin (°K)** and *9.027 DPT_Value_Temp_F* if the value is **degrees Fahrenheit (°F)**.

➢ 7.1.4 KNX external sensor incidence in measured temp. 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 parameter **“KNX external sensor incidence in measured temp. calculation”** is used to determine the incidence of the value measured by the KNX external sensor when calculating the measured temperature, which ranges from a minimum of 10% to a maximum of 100% (external sensor measured value = measured temperature). The complete formula for the temperature calculation is:

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

The parameter may assume the following values:

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

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

The parameter **“KNX external sensor monitoring time [min] (0=no monitoring)”** is used to define the monitoring time of the KNX external sensor and can assume the following values:

- from 0 to 10 with steps of 1, **2 (default value)**

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 based on the setting of the parameter **“Behaviour at KNX external sensor signal absence”**. This parameter may have the following values:

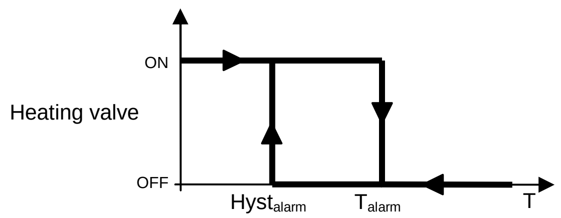
- **use last value received (default value)**
- exclude KNX sensor contribution

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

➢ 7.1.6 Threshold temperature temperature alarm (tenth of ℃)

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

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



7.1.7 Temperature alarm hysteresis (tenth of °C)

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

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

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

The parameter **"KNX floor sensor monitoring time [min] (0=no monitoring)"** is used to define the monitoring time of the KNX external floor sensor and can assume the following values:

- from 0 to 10 with steps of 1, **2 (default value)**

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 based on the setting of the parameter **"Behaviour at KNX floor sensor signal absence"**. This parameter may have the following values:

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

7.1.9 Auxiliary external sensor function

The parameter **"Auxiliary external sensor function"**, which is displayed if the parameter **"Input contacts function"** in the **Main** menu has the value **of an auxiliary input and a temperature sensor**, is used to configure the auxiliary input 2 to connect an NTC temperature sensor for measuring the room temperature or the floor temperature; the terminals of the auxiliary input 2 are used for this function. The values that can be set are:

- **enables temperature sensor (default value)**
- enable floor sensor

Selecting **enable temperature sensor** displays the parameters **"Correction factor auxiliary sensor [tenths of °C]"**, **"Auxiliary ext. sensor incidence in measured temp. calculation"**, **"Behaviour at auxiliary external sensor signal absence"** and **"Temperature measured by the auxiliary sensor"**.
Selecting **enable floor sensor** displays the parameters **"Correction factor auxiliary sensor [tenths of °C]"**,**Temperature measured by auxiliary floor sensor**, **"Threshold temperature temperature alarm (tenth of °C)"**, **"Temperature alarm hysteresis (tenth of °C)"**, **"Behaviour at auxiliary floor sensor signal absence"** and **"Temperature measured by the auxiliary sensor"**.

The parameter **"Correction factor auxiliary sensor [tenths of °C]"** 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 with steps of 1, **0 (default value)**

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 parameter **“Auxiliary ext. sensor incidence in measured temp. calculation”** 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:

$T_{measured} = T_{KNX\ external\ sensor} x\ Incidence_{KNX\ external\ sensor} + T_{auxiliary\ external\ sensor}\ x\ Incidence_{auxiliary\ external\ sensor} + T_{device\ sensor} x\ (100\%-Incidence_{auxiliary\ external\ sensor}-Incidence_{external\ sensor})$.

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

The parameter may assume the following values:

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

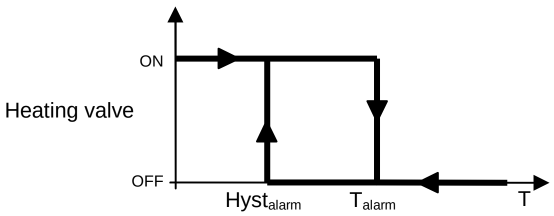
The auxiliary external sensor is always connected to the device, therefore there is no discussion in the case about sensor monitoring time; however it is possible to detect potential NTC sensor malfunction and define how the device must react via the parameter **“Behaviour at auxiliary external sensor signal absence”**. 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 parameter **“Temperature alarm threshold temperature (tenth of °C)”** 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 with steps of 1, **500 (default value)**



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

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

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

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

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

The parameter **"Measure unit auxiliary sensor"** is used to set the measure unit with which the information will be coded and sent via the communication object ***Temperature measured by auxiliary sensor***; the values that can be set are:

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

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

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

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

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

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

8 *"Feedbacks"* menu

The **Feedbacks** menu contains the parameters used to set the conditions for sending the feedback that the device sends via BUS telegrams. The structure of the menu is as follows:

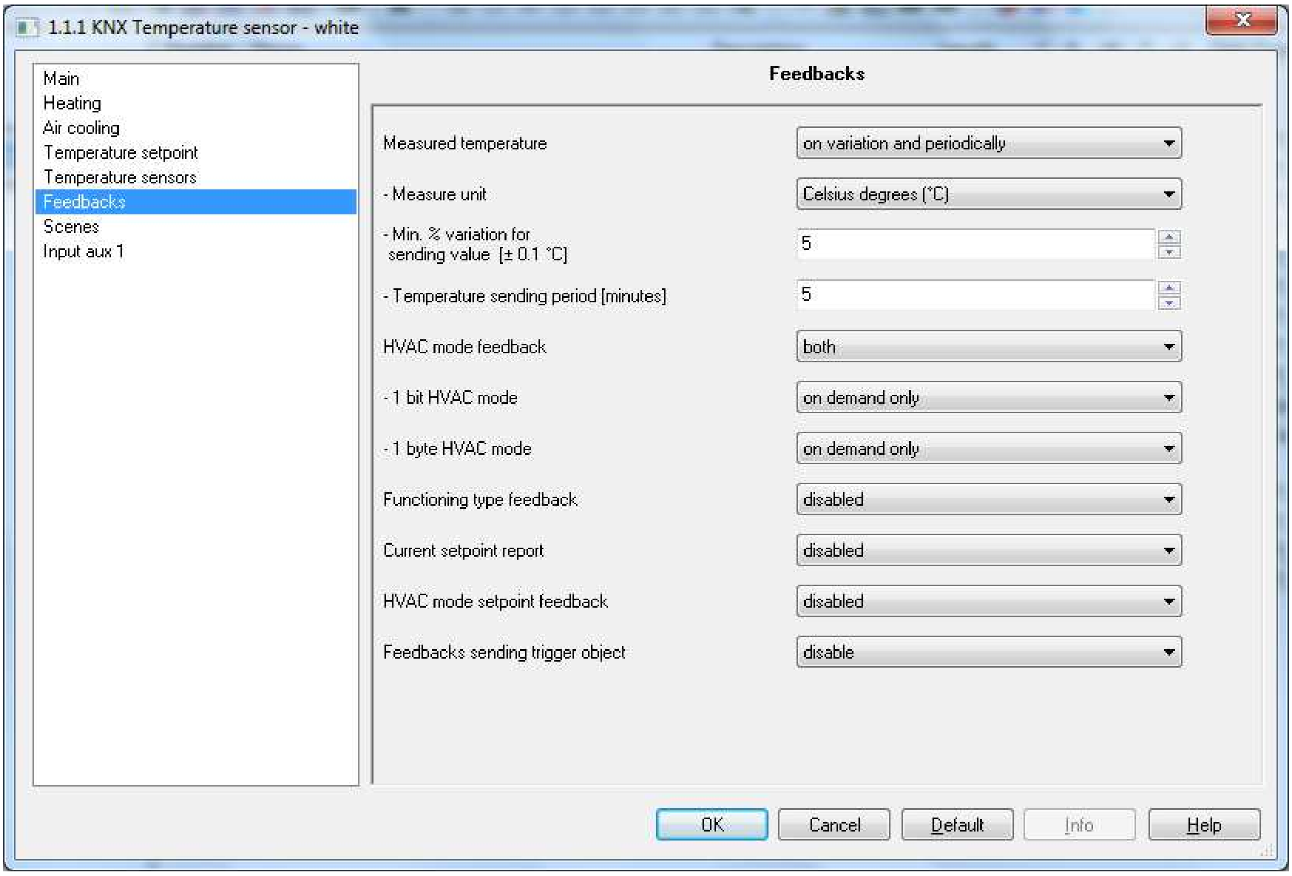


Fig. 8.1

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

8.1.2 Measure unit

This is used to set the measure unit with which the information will be coded and sent via the communication object ***Measured temperature***, the values that can be set are:

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

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

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

This is visible if the temperature is sent on variation, and is used to define the minimum temperature variation, with respect to the last sent temperature value, which causes the new measured value to be spontaneously sent, the values that can be set are:

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

8.1.4 Temperature sending period [minutes]

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

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

8.1.5 HVAC mode feedback

This is used to enable and set the format of the BUS telegrams the device uses to signal the active HVAC mode on the device. The values that can be set are:

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

Selecting **1 bit** or **both** displays the parameter **"1 bit HVAC mode"** and the communication objects ***HVAC off mode feedback***, ***HVAC economy mode feedback***, ***HVAC precomfort mode feedback***, ***HVAC comfort mode feedback*** and ***HVAC auto mode feedback*** (Data Point Type: 1.003 DPT_Enable) which are used to signal the active HVAC mode; when is 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 parameter **"1 byte HVAC mode"** and the communication object ***HVAC mode feedback*** (Data Point Type: 20.102 DPT_HVACMode) which are used to signal the active HVAC mode.
The parameter **"1 bit HVAC mode"** is used to set the conditions for sending the operating mode feedback via the communication objects ***HVAC off mode feedback***, ***HVAC economy mode feedback***, ***HVAC precomfort mode feedback***, ***HVAC comfort mode feedback*** and ***HVAC auto mode feedback*** 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 communication objects ***HVAC off mode feedback***, ***HVAC economy mode feedback***, ***HVAC precomfort mode feedback***, ***HVAC comfort mode feedback*** and ***HVAC auto mode feedback*** 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 one byte object. By selecting **on variation**, the operating mode feedback is sent spontaneously by the device via the communication objects ***HVAC off mode feedback***, ***HVAC economy mode feedback***, ***HVAC precomfort mode feedback***, ***HVAC comfort mode feedback*** and ***HVAC auto mode feedback*** 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.

The parameter **"1 byte HVAC mode"** is used to set the conditions for sending the HVAC mode feedback via the communication objects ***HVAC mode feedback*** 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 communication object ***HVAC mode feedback*** 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 communication object ***HVAC mode feedback*** with a 1 byte size, each time the mode is changed.

➢ *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 communication object ***Functioning type feedback*** (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 communication object ***Functioning type feedback***; in the case of a status reading request, it sends the requester a telegram in response to the received command, which includes information about the 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 communication object ***Functioning type feedback***, each time the functioning type changes.

➢ 8.1.7 Current setpoint feedback

This is used to enable and set the conditions for sending the feedback regarding the current setpoint value (which accounts for any temporarily active forcing) set on the device via the BUS telegram on the communication object ***Current setpoint feedback*** (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 communication object ***Current setpoint feedback***; in the case of a status reading request, it sends the requester a telegram in response to the received command, which includes information about the setpoint set on the device. 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 communication object ***Current setpoint feedback***, each time there is a variation in the setpoint itself (also following a temporary forcing).

➢ 8.1.8 HVAC mode setpoint feedback

The parameter **"HVAC mode setpoint feedback"**, which is visible if the remote control type is HVAC mode, is used to enable sending the setpoint value of the HVAC mode via the objects ***Heating antifreeze setpoint feedback***, ***Heating economy setpoint feedback***, ***Heating antifreeze precomfort feedback***, ***Heating antifreeze comfort feedback***, ***Air cooling high temperature protection setpoint feedback***, ***Air cooling economy setpoint feedback***, ***Air cooling precomfort setpoint feedback*** and ***Air cooling comfort setpoint feedback*** (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 communication objects ***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***; 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 communication objects ***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*** each time the functioning type changes;

➢ *8.1.9 Feedbacks sending trigger object*

The parameter **"Feedbacks sending trigger object"** is used to enable the input object ***Feedback sending trigger*** (Data Point Type: 1.017 DPT_Trigger); when this object receives a BUS telegram with the logical value "0" or "1", the device automatically sends all the 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:

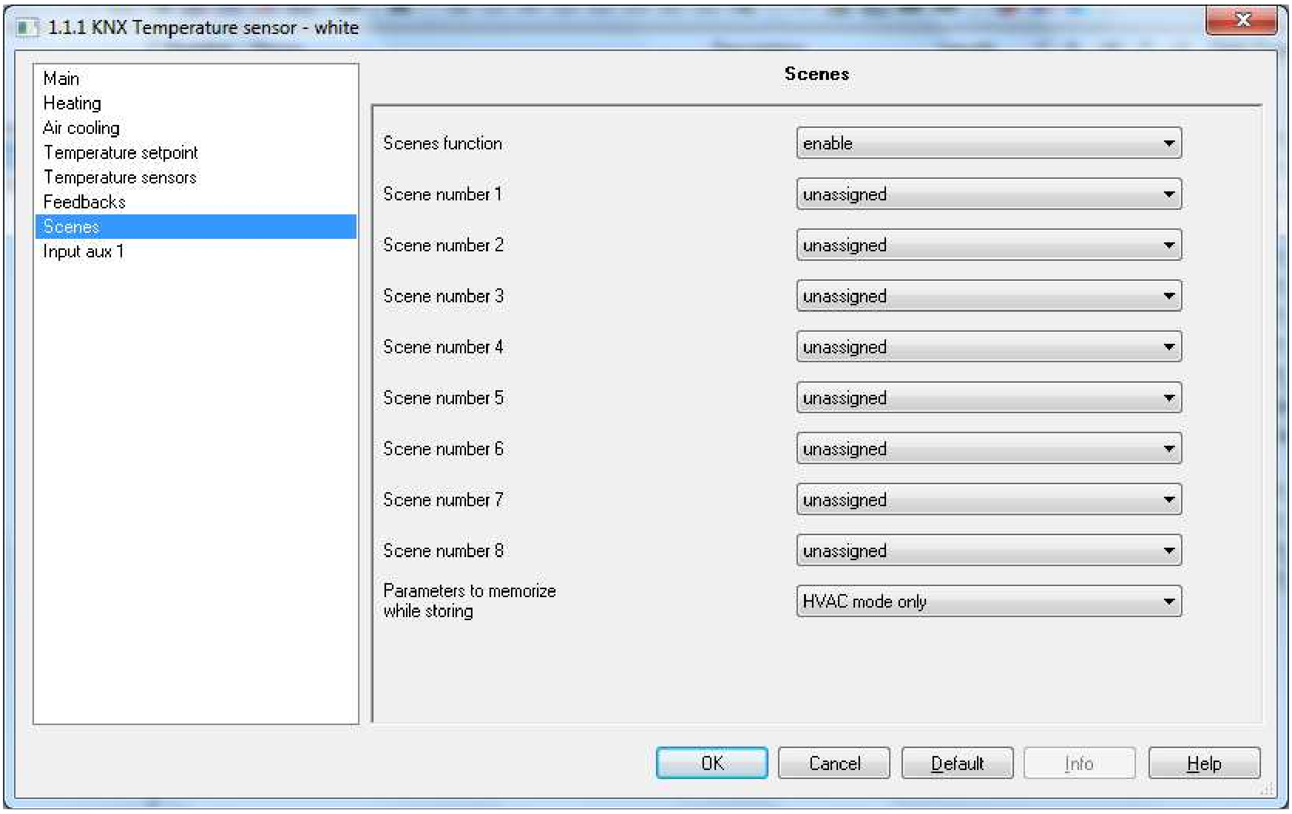


Fig. 9.1

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 communication object ***KNX sensor scene*** (Data Point Type: 18.001 DPT_SceneControl).
The scene function is used to send two possible commands to the device:

- execute scene, which is a command to assume a determined condition
- store scene, which is a command to memorise the current status (the moment the command is received) of the different functional parameters of the device defined in the configuration phase.

This function provides 8 scenes, for which the device can store/reproduce 8 different conditions of these functional parameters. The values that can be set are:

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

Selecting **enable** displays the parameters , **"Scene number 1"**, **"Scene number 2"**, **"Scene number 3"**, **"Scene number 4"**, **"Scene number 5"**, **"Scene number 6"**, **"Scene number 7"**, **"Scene number 8"** and **"Parameters to memorize while storing"** and the communication object ***KNX sensor scene***, through which the scene execution/memorise telegrams are received.

➢ 9.1.2 Scene number i

With the parameter **"Scene number i"** (1 ≤ i ≤ 8) it is possible to set the numerical value that is used to identify and therefore execute/memorise the i-th scene; the possible values are:

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

➢ 9.1.3 Parameters to memorize while storing

As the sensor has various operating parameters that can change during its operation, this parameter can be used to configure which of them should be memorised while storing the scene, to then be replicated following an execution command.
The values that can be set 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 remote 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.

10 *"Input aux 1" and "Inpux aux 2"* menu

The device has two terminals for the connection of a potential-free contact that can be used as a generic input (auxiliary input 1) and two terminals for the connection of a potential-free contact that can be used as a generic input (auxiliary input 2) or as an auxiliary external sensor.
The **Input aux 1** menu is displayed if the parameter **"Input contacts function"** 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 parameter **"Input contacts function"** 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.
The structure of the menu is as follows:

Fig. 10.1

10.1 Parameters

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 communications objects will refer to the 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 10.1 Function "edges (closing/opening)")
- short/long operation
  (See paragraph 10.2 Function "short/long operation")
- 1 push button + stop dimmer
  (See paragraph 10.3 Function "1 push button + stop dimmer")
- 1 push button dimmer with cyclic sending
  (See paragraph 10.4 Function "1 push button dimmer with cyclic sending")

- 1 push button shutter control
  (See paragraph 10.5 Function "1 push button shutter control")
- scene management
  (See paragraph 10.6 Function "scene management")
- window contact
  (See paragraph 10.7 Function "window contact")
  The function **window contact** is only available for **Input aux 1**.

10.2 Edge function

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:

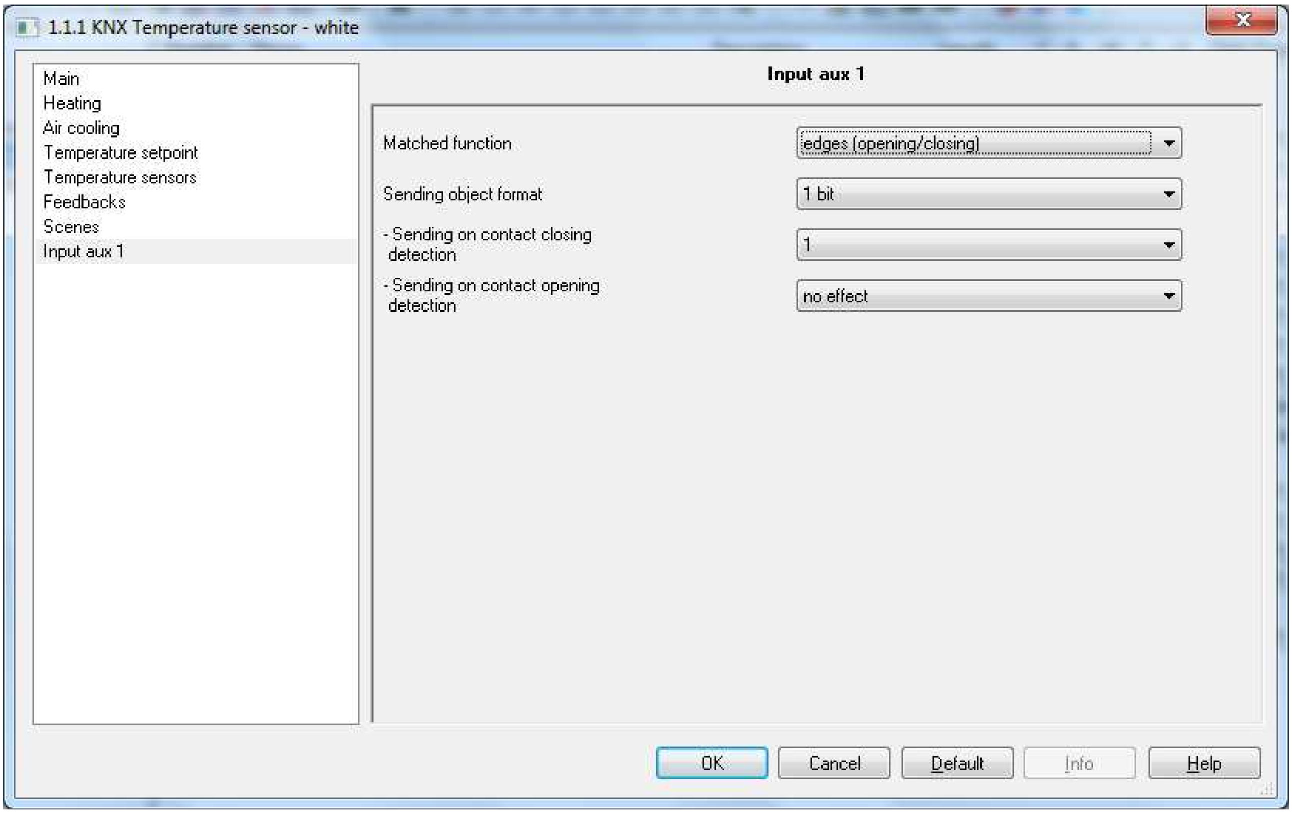


Fig. 10.1

➢ *11.1.1 Parameters*

The parameter **“Sending object format”** 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 parameters **“Sending on contact closing detection”** and **“Sending on contact opening detection”** to change as a result.
The parameter **“Sending on contact closing detection”** 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 parameter **“Sending on contact opening detection”** 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 communication object ***Aux input - 1 bit value*** 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**, the parameter **"Status feedback object"** will be shown which is used to enable and display the communication object ***Aux input - status feedback*** (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 object ***Aux input - status feedback*** and the last sent value (via the object ***Aux input - 1 bit value***). The **"Status feedback object"** may have the following values:
  - **disabled** **(default value)**
  - enabled

  Selecting the value **enabled** displays the communication object ***Aux input - status feedback***.

- If the format of the object to send is **2 bit**, the communication object ***Aux input - 2 bit value*** 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)**
  - **on forcing active (down)** **(default closing value)**
  - forcing activation off (up)
  - forcing deactivation
  - forcing on/forcing off cyclic switching
  - forcing on/forcing deactivation cyclic switching
  - forcing off/forcing deactivation cyclic 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 communication object ***Aux input - 1 byte value*** 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 new displayed parameter **"Value (0 .. 255)"** which can assume the following values:

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

- If the format of the object to send is **1 byte signed value**, the communication object ***Aux input - 1 byte value*** 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 new displayed parameter **"Value (-128 .. 127)"** which can assume the following values:

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

- If the format of the object to send is **1 byte percentage value**, the communication object ***Aux input - 1 byte value*** 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 new displayed parameter **"Value (0% .. 100%)"** which can assume the following values:

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

- If the format of the object to send is **1 byte HVAC mode**, the communication object ***Aux input - 1 byte value*** 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** **(closure default value)**
  - precomfort
  - economy
  - off (building protection)
  - cyclic switching (thermostat)
  - cyclic 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 **cyclic 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 **cyclic 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 communication object ***Aux input 2 byte value*** 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 new displayed parameter **"Value (0 .. 65535)"** which can assume the following values:

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

- If the format of the object to send is **2 byte signed value**, the communication object ***Aux input 2 byte value*** 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 new displayed parameter **"Value (-32768 .. +32767)"** which can assume the following values:

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

- If the format of the object to send is **4 byte unsigned value**, the communication object ***Aux input 4 byte value*** 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 new displayed parameter **"Value (0 .. 4294967295)"** which can assume the following values:

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

- If the format of the object to send is **4 byte signed value**, the communication object ***Aux input 4 byte value*** 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 new displayed parameter **"Value (-2147483648 .. 2147483647)"** which can assume the following values:

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

- If the format of the object to send is **14 byte HVAC mode**, the communication object ***Aux input 14 byte value*** 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 new displayed parameter **"Value (ISO characters 8859-1)"** which can assume the following values:

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

10.3 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:

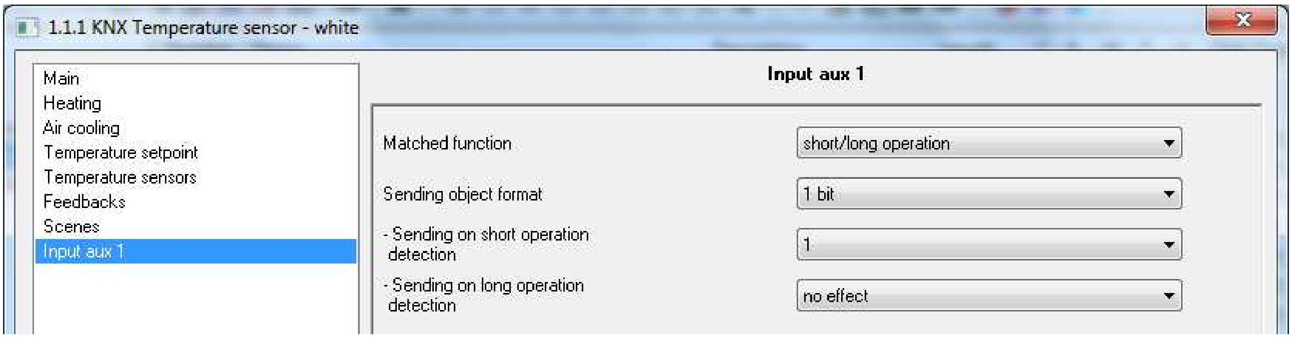


Fig. 10.2

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 parameter **“Sending object format”** 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 parameters **“Sending on short operation detection”** and **“Sending on long operation detection”** to change as a result.
The parameter **“Sending on short operation detection”** 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 parameter **“Sending on long operation detection”** 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 communication object ***Aux input - 1 bit value*** 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)**
  - cyclic switching

  Selecting **cyclic switching**, the parameter **“Status feedback object”** will be shown which is used to enable and display the communication object ***Aux input - status feedback*** (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 object ***Aux input - status feedback*** and the last sent value (via the object ***Aux input - 1 bit value***). The **“Status feedback object”** may have the following values:

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

  Selecting the value **enabled** displays the communication object ***Aux input - status feedback***.

- If the format of the object to send is **2 bit**, the communication object ***Aux input - 2 bit value*** 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)**
  - **forcing activation on (down)** **(short operation default value)**
  - forcing activation off (up)
  - deactivate forcing

- forcing on/ forcing off cyclical switching
- forcing on/forcing deactivation cyclic switching
- forcing off/forcing deactivation cyclic 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 communication object ***Aux input - 1 byte value*** 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 new displayed parameter **“Value (0 .. 255)”** which can assume the following values:

  - from **0 (default value)** to 255, with steps of 1
- If the format of the object to send is **1 byte signed value**, the communication object ***Aux input - 1 byte value*** 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 new displayed parameter **“Value (-128 .. 127)”** which can assume the following values:

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

- If the format of the object to send is **1 byte percentage value**, the communication object ***Aux input - 1 byte value*** 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 new displayed parameter **“Value (0% .. 100%)”** which can assume the following values:

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

- If the format of the object to send is **1 byte HVAC mode**, the communication object ***Aux input - 1 byte value*** 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)
  - cyclic switching (thermostat)
  - cyclic 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 **cyclic 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 **cyclic 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 communication object ***Aux input 2 byte value*** 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 new displayed parameter **"Value (0 .. 65535)"** which can assume the following values:

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

- If the format of the object to send is **2 byte signed value**, the communication object ***Aux input 2 byte value*** 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 new displayed parameter **"Value (-32768 .. +32767)"** which can assume the following values:

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

- If the format of the object to send is **4 byte unsigned value**, the communication object ***Aux input 4 byte value*** 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 new displayed parameter **"Value (0 .. 4294967295)"** which can assume the following values:

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

- If the format of the object to send is **4 byte signed value**, the communication object ***Aux input 4 byte value*** 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 new displayed parameter **"Value (-2147483648 .. 2147483647)"** which can assume the following values:

  - from -2147483648 to 2147483647 with steps of 1, **0 (default value)**
- If the format of the object to send is **14 byte HVAC mode**, the communication object ***Aux input 14 byte value*** 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 new displayed parameter **"Value (ISO characters 8859-1)"** which can assume the following values:

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

10.4 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 ***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:

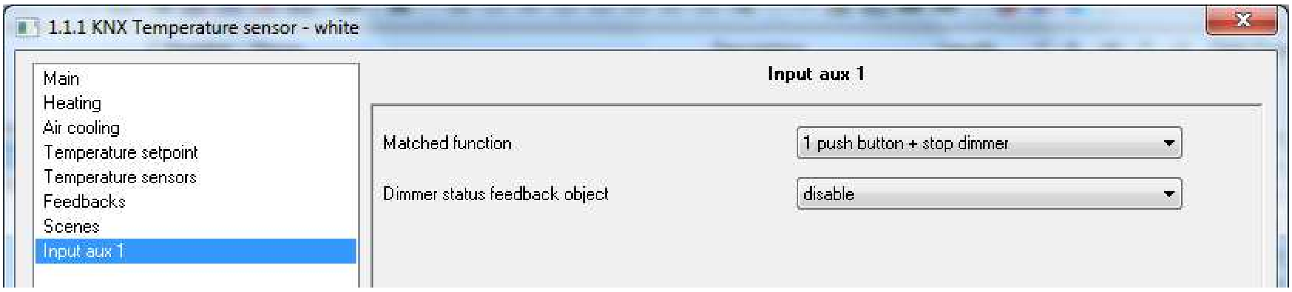


Fig. 10.3

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 communication object ***Aux input - Dimmer status feedback*** (Data Point Type: 1.001 DPT_Switch), via the parameter **"Dimmer status feedback object"**.

➢ *11.3.1 Parameters*

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

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

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

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

  - only brightness increase
  - only brightness decrease
  - **brightness 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.

10.5 1 Push button dimmer with cyclic sending

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:

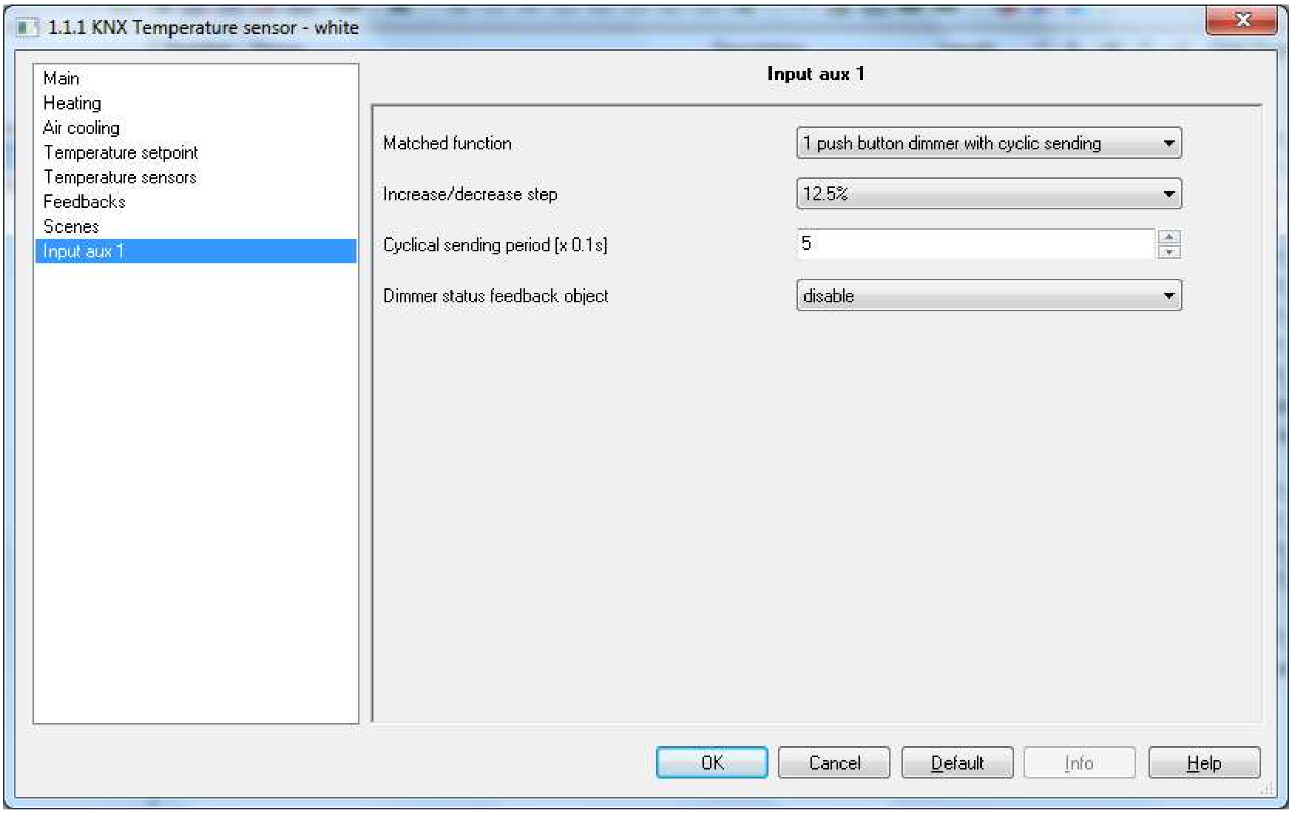


Fig. 10.4

➢ *11.4.1 Parameters*

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

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

If the contact remains with a closed status, the device sends the command cyclically until it detects the release, the parameter **"Cyclical sending period [x 0.1s]"** is used to set 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 parameter **"Cyclical sending period [x 0.1s]"** are:

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

To sum up, when a long operation is detected, the device sends the first increase/decrease command with the set percentage and, if the 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 parameter **Cyclical sending period [x 0.1s]** 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 function ***1 push button + stop dimmer***, it is possible to enable the dimmer status feedback object by changing the behaviour of the switching and control commands as described in paragraph 12.3 Function ”1 push button + stop dimmer”.
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 parameter **“Brightness control commands with dimmer on”** and the communication object ***Aux input - Dimmer status feedback*** (Data Point Type: 1.001 DPT_Switch), which is used to receive the status feedback from the controlled dimmer actuator.
The parameter **“Brightness control commands with dimmer on”** can have the following values:

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

10.6 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 louvres opening/closing.
As only one channel manages the louvres up/down and adjustment 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 object ***Aux input - Shutter movement*** or the movement feedback received via the object ***Aux input - Movement feedback***, 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 adjustment command. The new value to be sent depends on the last value sent via the object ***Ch.x - Shutter movement*** or the movement feedback received via the object ***Ch.x - Movement feedback***, depending on which of the two events occurred last; if the last event that occurred is “upward movement feedback reception” or “sending upward movement command”, the command will be a "closing louvres adjustment" command and vice versa. If the shutter is moving, the louvre adjustment command will only stop the shutter up/down movement.

The communication objects enabled by this function are: ***Aux input - Shutter movement*** (Data Point Type: 1.008 DPT_UpDown) used to send the actuator up/down movement commands, ***Aux input - Louvre stop/adjustment*** (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:

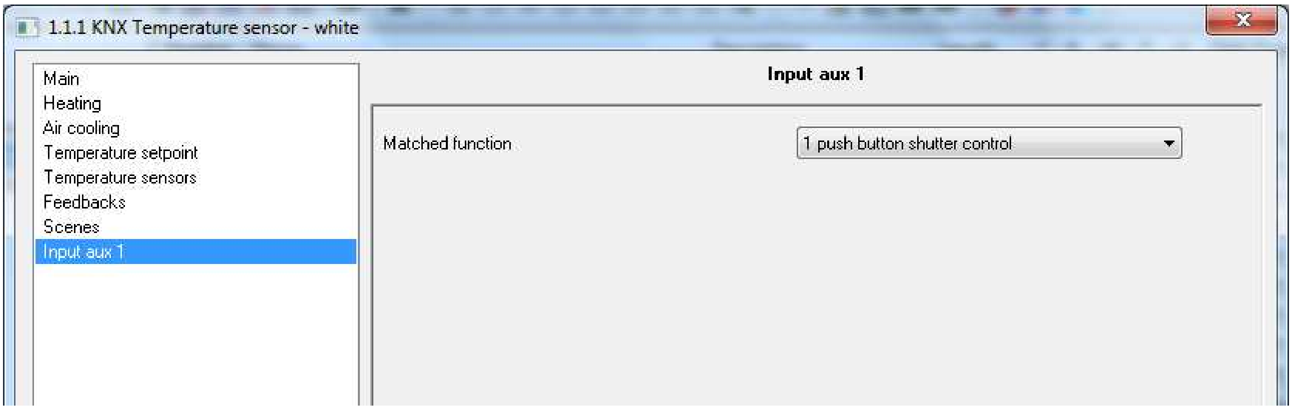


Fig. 10.5

No new parameters are enabled with this function.

10.7 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:

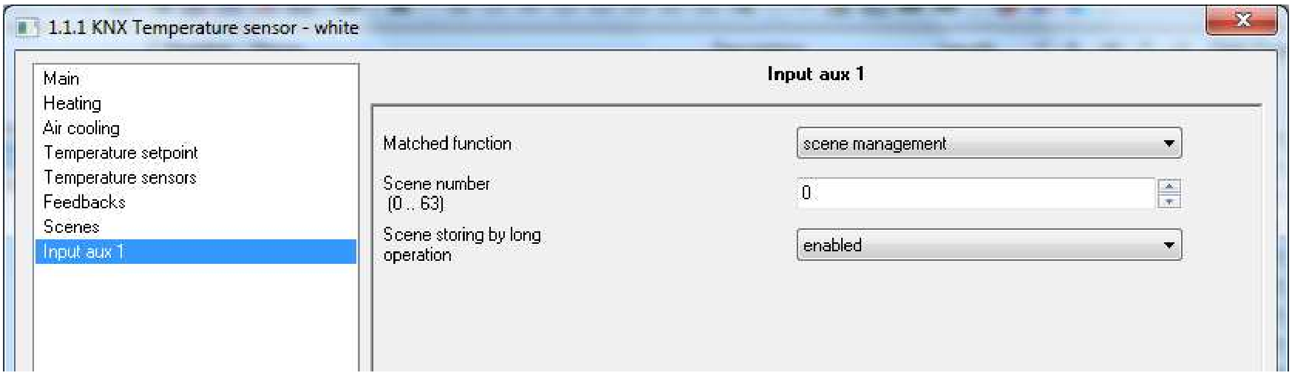


Fig. 10.6

➢ *11.6.1 Parameters*

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

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

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

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

Only if **enabled** is selected, the device will send the scene storing command when a long operation is detected; 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). Independently of the value set for the above parameter, it is possible to indirectly generate the sending of the scene memorising command after receiving a BUS telegram on the object ***Aux input - Scene storing trigger*** (both with value “1” as well as with value “0”); each time the device receives a telegram on that object, it must immediately send a scene memorisation telegram.

10.8 Window contact

This is used to configure the channel for performing the "window contact" function of the sensor.
The structure of the menu is as follows:

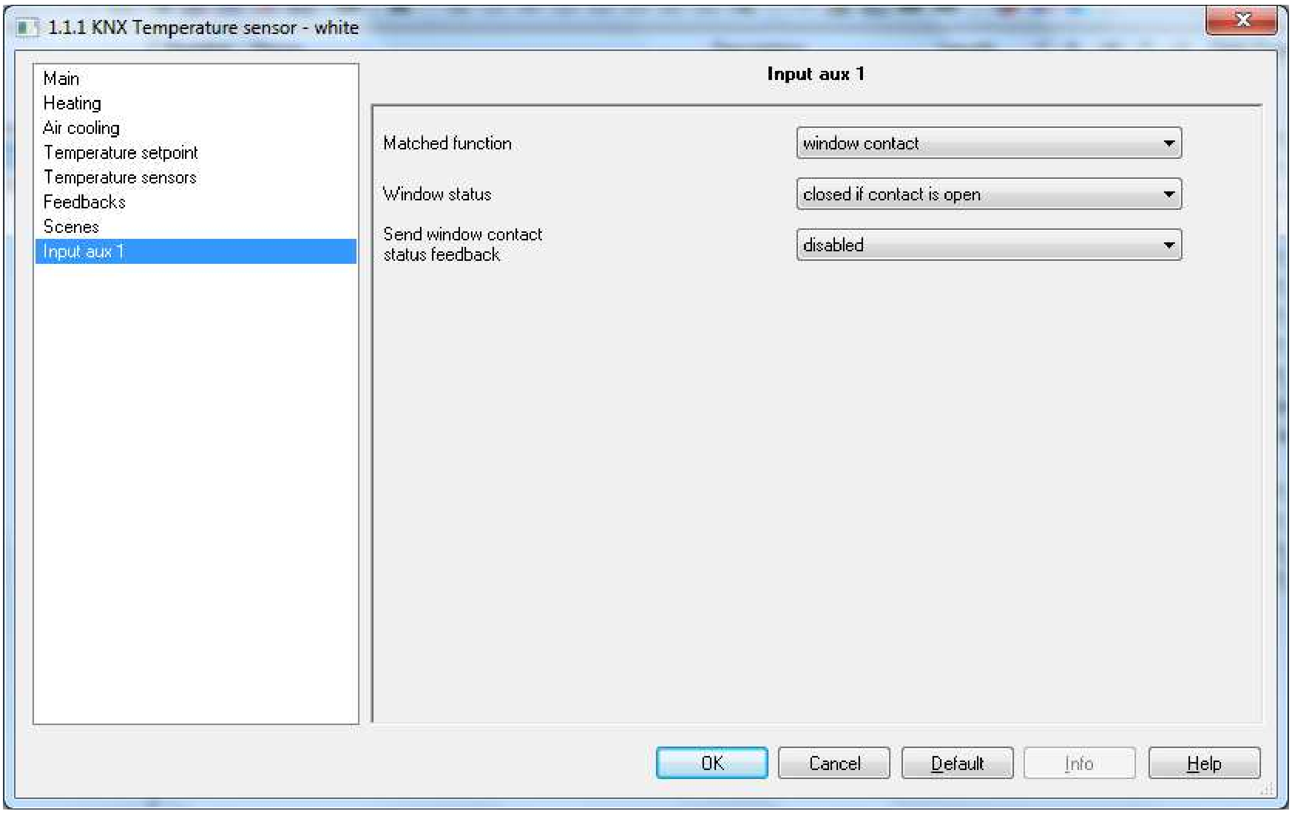


Fig. 10.7

This particular function is used to force the device in OFF mode when the window opens and to reactivate normal operation when the window is closed; the priority of this command is higher than all remote commands, including 1 bit modes.

➢ *11.7.1 Parameters*

The parameter **“Window status”** 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 communication object ***Aux input - Window status feedback***; 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 communication object ***Aux input - Window status feedback*** (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 parameter **“Window status”**) and the value “0” when the window is CLOSED.

10.9 Inputs aux 1/2

The device has two terminals for the connection of a potential-free contact that can be used as a generic input (auxiliary input 1) and two terminals for the connection of a potential-free contact that can be used as a generic input (auxiliary input 2) or as an auxiliary external sensor.
The **Input aux 1/2** menu is displayed if the parameter **“Input contacts function”** in the **Main** menu has the value **two auxiliary inputs combined**. The structure of the menu is as follows:

Fig. 10.8

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 10.9.1)
- dimming with cycling telegram
  (See paragraph 10.9.2)
- shutter control
  (See paragraph 10.9.3)

➢ 10.9.1 Dimming with STOP telegram

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:

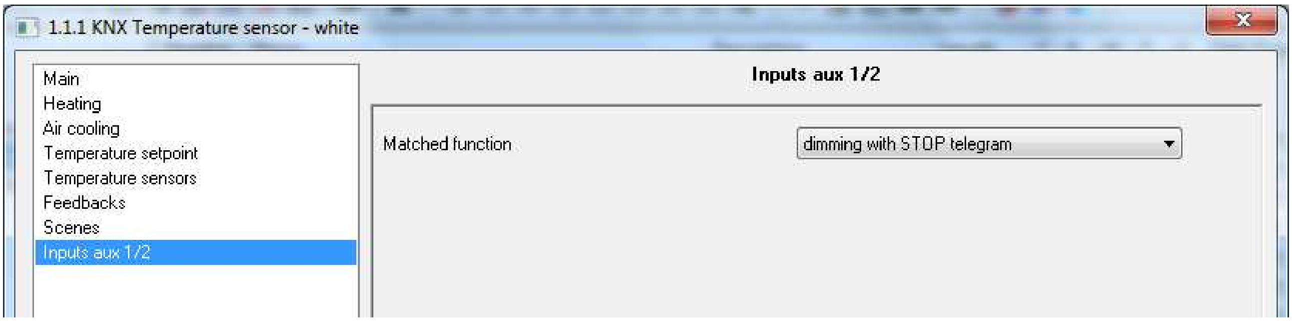


Fig. 10.9

No new parameters are enabled with this function.

➢ 10.9.2 Dimming with cyclic telegram

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. 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 function ***dimming with telegram + STOP***, it is possible to define both the brightness variation steps of the brightness increase/decrease commands as well as the time that must elapse between the sending of one command and another when the push-button remains pressed; 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:

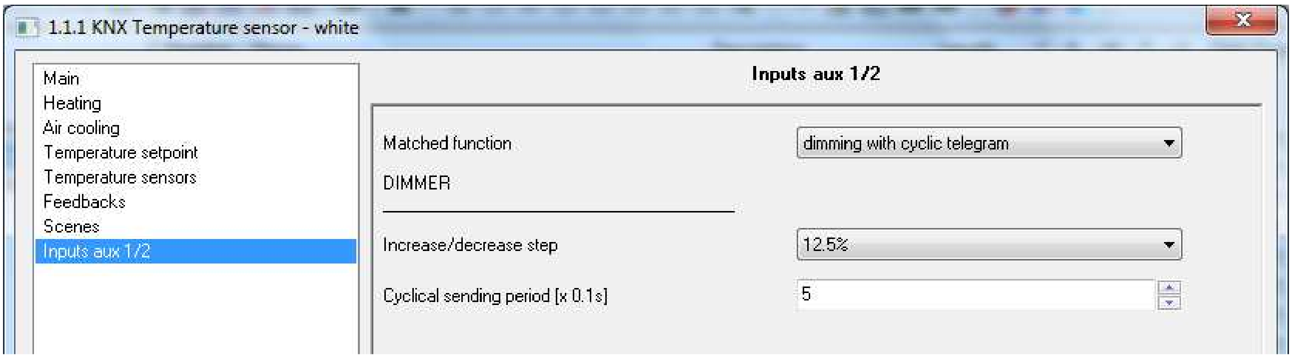


Fig. 10.10

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

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

If pressing is maintained, the device will cyclically send the command until release is detected; the parameter **"Cyclical sending period [x 0.1s]"** is used to set 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 parameter **"Cyclical sending period [x 0.1s]"** are:

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

In summary, 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 parameter **Increase/decrease step** is set to **12.5%** and the parameter **Cyclical sending period [x 0.1s]** is set to **3** (0.3 sec) and the pressure on the push-button is detected:
– 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

➢ *10.9.3 Dimming with cyclic telegram*

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

- a long operation (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:

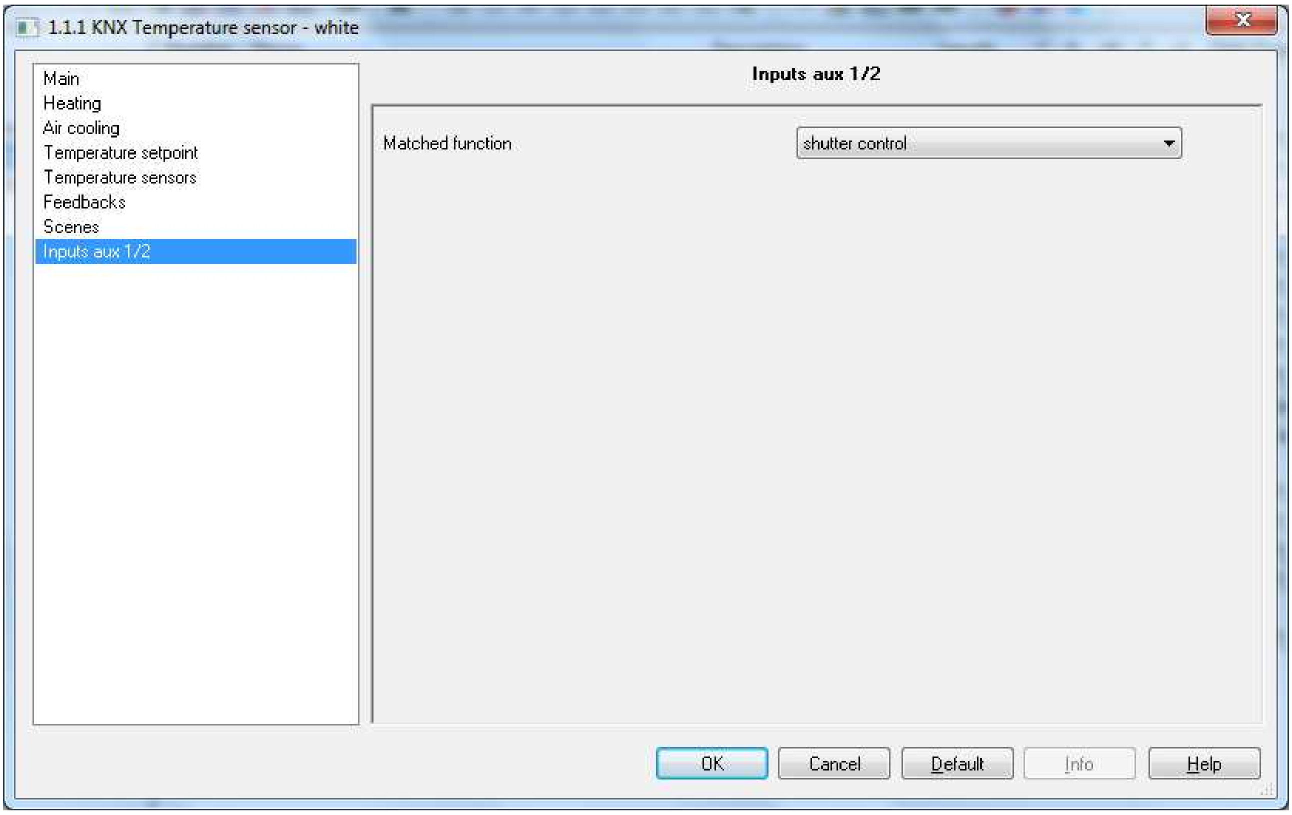


Fig. 10.11

No new parameters are enabled with this function.

11 Communication objects

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

11.1 Communication object table

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.

➢ 11.1.1 Communication objects with input functions

#	Object name	Object function	Description	Datapoint type
0	HVAC mode input	Eco/Precom/Comf/Off	Receives the HVAC mode setting commands	20.102 DPT_HVACMode
0	Setpoint input	Value °C	Receives the operating setpoint values expressed in degrees Celsius	9.001 DPT_Temp
0	Setpoint input	Value °K	Receives the operating setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
0	Setpoint input	Value °F	Receives the operating setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
1	HVAC off mode input	Enable/disable	Receives the HVAC off mode enabling commands (building protection)	1.003 DPT_Enable
1	Temporary setpoint forcing input	Value °C	Receives the values of the temporary active setpoint forcing expressed in degrees Celsius	9.001 DPT_Temp
1	Temporary setpoint forcing input	Value °K	Receives the values of the temporary active setpoint forcing expressed in degrees Kelvin	9.002 DPT_Tempd
1	Temporary setpoint forcing input	Value °F	Receives the values of the temporary active setpoint forcing expressed in degrees Fahrenheit	9.027 DPT_Temp_F
2	HVAC economy mode input	Enable/disable	Receives the HVAC economy mode enabling commands	1.003 DPT_Enable
3	HVAC precomfort mode input	Enable/disable	Receives the HVAC precomfort mode enabling commands	1.003 DPT_Enable
4	HVAC comfort mode input	Enable/disable	Receives the HVAC comfort mode enabling commands	1.003 DPT_Enable
5	Functioning type input	Heating/air cooling	Receives the functioning type setting commands	1.100 DPT_Heat/Cool
6	Enable dead zone	Enable/disable	Receives the automatic functioning type setting enabling commands (dead zone)	1.003 DPT_Enable
7	Heating antifreeze setpoint input	Value °C	Receives the heating HVAC off mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
7	Heating antifreeze setpoint input	Value °K	Receives the heating HVAC off mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
7	Heating antifreeze setpoint input	Value °F	Receives the heating HVAC off mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
8	Heating economy setpoint input	Value °C	Receives the heating HVAC economy mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
8	Heating economy setpoint input	Value °K	Receives the heating HVAC economy mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
8	Heating economy setpoint input	Value °F	Receives the heating HVAC economy mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
9	Heating precomfort setpoint	Value °C	Receives the heating HVAC	9.001 DPT_Temp

	input		precomfort mode setpoint values expressed in degrees Celsius	
9	Heating precomfort setpoint input	Value °K	Receives the heating HVAC precomfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
9	Heating precomfort setpoint input	Value °F	Receives the heating HVAC precomfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
10	Heating comfort setpoint input	Value °C	Receives the heating HVAC comfort mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
10	Heating comfort setpoint input	Value °K	Receives the heating HVAC comfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
10	Heating comfort setpoint input	Value °F	Receives the heating HVAC comfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
11	Air cooling high temperature protection setpoint input	Value °C	Receives the air cooling HVAC off mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
11	Air cooling high temperature protection setpoint input	Value °K	Receives the air cooling HVAC off mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
11	Air cooling high temperature protection setpoint input	Value °F	Receives the air cooling HVAC off mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
12	Air cooling economy setpoint input	Value °C	Receives the air cooling HVAC economy mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
12	Air cooling economy setpoint input	Value °K	Receives the air cooling HVAC economy mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
12	Air cooling economy setpoint input	Value °F	Receives the air cooling HVAC economy mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
13	Air cooling precomfort setpoint input	Value °C	Receives the air cooling HVAC precomfort mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
13	Air cooling precomfort setpoint input	Value °K	Receives the air cooling HVAC precomfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
13	Air cooling precomfort setpoint input	Value °F	Receives the air cooling HVAC precomfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
14	Air cooling comfort setpoint input	Value °C	Receives the air cooling HVAC comfort mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
14	Air cooling comfort setpoint input	Value °K	Receives the air cooling HVAC comfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
14	Air cooling comfort setpoint input	Value °F	Receives the air cooling HVAC comfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
16	Heating valve status feedback	On/Off status	Receives feedback about the first stage heating solenoid valve activation status	1.001 DPT_Switch
16	Heating valve % feedback	% Value	Receives feedback about the first stage heating solenoid valve activation status	5.001 DPT_Scaling
16	Heating/air cooling valve status feedback	On/Off status	Receives feedback about the heating/air cooling solenoid valve activation status	1.001 DPT_Switch
16	Heating/air cooling valve % feedback	% Value	Receives feedback about the heating/air cooling solenoid valve activation status	5.001 DPT_Scaling
20	Heating fancoil speed % feedback	% Value	Receives feedback about the % speed of the heating fancoil	5.001 DPT_Scaling
20	Heating fan V1 status feedback	On/Off status	Receives feedback about the heating fancoil speed 1 activation status	1.001 DPT_Switch
21	Heating fan V2 status feedback	On/Off status	Receives feedback about the heating fancoil 2 speed activation status	1.001 DPT_Switch

22	Heating fan V3 status feedback	On/Off status	Receives feedback about the heating fancoil 3 speed activation status	1.001 DPT_Switch
24	Heating 2nd stage status feedback	On/Off status	Receives feedback about the second stage heating solenoid valve activation status	1.001 DPT_Switch
24	Heating 2nd stage valve % feedback	% Value	Receives feedback about the second stage heating solenoid valve activation status	5.001 DPT_Scaling
26	Air cooling valve status feedback	On/Off status	Receives feedback about the first stage air cooling solenoid valve activation status	1.001 DPT_Switch
26	Air cooling valve % feedback	% Value	Receives feedback about the first stage air cooling solenoid valve activation status	5.001 DPT_Scaling
30	Air cooling fancoil speed % feedback	% Value	Receives feedback about the % speed of the air cooling fancoil	5.001 DPT_Scaling
30	Air cooling fan V1 status feedback	On/Off status	Receives feedback about the air cooling 1 speed activation status	1.001 DPT_Switch
31	Air cooling fan V2 status feedback	On/Off status	Receives feedback about the air cooling 2 speed activation status	1.001 DPT_Switch
32	Air cooling fan V3 status feedback	On/Off status	Receives feedback about the air cooling 3 speed activation status	1.001 DPT_Switch
34	Air cooling 2nd stage status feedback	On/Off status	Receives feedback about the second stage air cooling solenoid valve activation status	1.001 DPT_Switch
34	Air cooling 2nd stage valve % feedback	% Value	Receives feedback about the second stage air cooling solenoid valve activation status	5.001 DPT_Scaling
35	Fancoil mode input	Automatic/manual	Receives the automatic fancoil speed or manual speed mode selection commands	1.001 DPT_Switch
37	KNX external sensor input	Value ℃	Receives the values from the KNX external sensor expressed in degrees Celsius	9.001 DPT_Temp
37	KNX external sensor input	Value K	Receives the values from the KNX external sensor expressed in degrees Kelvin	9.002 DPT_Tempd
37	KNX external sensor input	Value ℉	Receives the values from the KNX external sensor expressed in degrees Fahrenheit	9.027 DPT_Temp_F
37	KNX floor sensor input	Value ℃	Receives the values from the KNX external floor sensor expressed in degrees Celsius	9.001 DPT_Temp
37	KNX floor sensor input	Value K	Receives the values from the KNX external floor sensor expressed in degrees Kelvin	9.002 DPT_Tempd
37	KNX floor sensor input	Value ℉	Receives the values from the KNX external floor sensor expressed in degrees Fahrenheit	9.027 DPT_Temp_F
55	Feedback sending trigger	Send feedback	Receives the feedback sending request trigger commands	1.017 DPT_Trigger
56	KNX sensor scene	Execute/Store	Receives the KNX sensor function scene execution/store commands	18.001 DPT_SceneControl
57	Aux input 1 - Dimmer status feedback	On/Off status	Receives the status feedback from the controlled dimmer	1.001 DPT_Switch
57	Aux input 1 - Status feedback	On/Off status	Receives the status feedback from the controlled actuator	1.001 DPT_Switch
57	Aux input 1 - Movement feedback	Up/down	Receives feedback about the current direction of movement from the controlled motor command actuator	1.008 DPT_UpDown
59	Aux input 1 - Scene storing trigger	Store	Receives the scene storing command sending request trigger commands	1.017 DPT_Trigger
60	Aux input 2 - Dimmer status feedback	On/Off status	Receives the status feedback from the controlled dimmer	1.001 DPT_Switch
60	Aux input 2 - Status feedback	On/Off status	Receives the status feedback from the controlled actuator	1.001 DPT_Switch
60	Aux input 2 - Movement feedback	Up/down	Receives feedback about the current direction of movement from the controlled motor command actuator	1.008 DPT_UpDown
62	Aux input 2 - Scene storing trigger	Store	Receives the scene storing command sending request trigger commands	1.017 DPT_Trigger

11.1.2 Communication objects with output functions

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

#	Object name	Object function	Description	Datapoint type
15	Heating valve switching	On/Off	Sends the 1st stage heating solenoid valve on/off commands	1.001 DPT_Switch
15	Heating/air cooling valve switching	On/Off	Sends the heating/air cooling solenoid valve on/off commands	1.001 DPT_Switch
15	Heating valve % command	% Value	Sends the percentage commands to the 1st stage heating solenoid valve	5.001 DPT_Scaling
15	Heating/air cooling valve % command	% Value	Sends the percentage commands to the heating/air cooling solenoid valve	5.001 DPT_Scaling
17	Heating fan V1 switching	On/Off	Sends the heating fancoil speed 1 on/off commands	1.001 DPT_Switch
17	Heating fancoil speed % command	% Value	Sends the percentage commands to the heating fancoil	5.001 DPT_Scaling
18	Heating fan V2 switching	On/Off	Sends the heating fancoil speed 2 on/off commands	1.001 DPT_Switch
19	Heating fan V3 switching	On/Off	Sends the heating fancoil speed 3 on/off commands	1.001 DPT_Switch
23	Heating 2nd stage switching	On/Off	Sends the 2nd stage heating solenoid valve on/off commands	1.001 DPT_Switch
23	Heating 2nd stage command	% Value	Sends the percentage commands to the 2nd stage heating solenoid valve	5.001 DPT_Scaling
25	Air cooling valve switching	On/Off	Sends the 1st stage air cooling solenoid valve on/off commands	1.001 DPT_Switch
25	Air cooling valve % command	% Value	Sends the percentage commands to the 1st stage air cooling solenoid valve	5.001 DPT_Scaling
27	Air cooling fan V1 switching	On/Off	Sends the air cooling fancoil speed 1 on/off commands	1.001 DPT_Switch
27	Air cooling fancoil speed % command	% Value	Sends the percentage commands to the air cooling fancoil	5.001 DPT_Scaling
28	Air cooling fan V2 switching	On/Off	Sends the air cooling fancoil speed 2 on/off commands	1.001 DPT_Switch
29	Air cooling fan V3 switching	On/Off	Sends the air cooling fancoil speed 3 on/off commands	1.001 DPT_Switch
33	Air cooling 2nd stage switching	On/Off	Sends the 2nd stage air cooling solenoid valve on/off commands	1.001 DPT_Switch
33	Air cooling 2nd stage command	% Value	Sends the percentage commands to the 2nd stage air cooling solenoid valve	5.001 DPT_Scaling
36	Fancoil mode feedback	Automatic/manual	Sends feedback about fancoil speed automatic/manual mode	1.001 DPT_Switch
38	Temperature measured by the auxiliary sensor	Value ℃	Sends the temperature values expressed in degrees Celsius measured by the NTC sensor connected to the auxiliary sensor input	9.001 DPT_Temp
38	Temperature measured by the auxiliary sensor	Value K	Sends the temperature values expressed in degrees Kelvin measured by the NTC sensor connected to the auxiliary sensor input	9.002 DPT_Tempd
38	Temperature measured by the auxiliary sensor	Value ℉	Sends the temperature values expressed in degrees Fahrenheit measured by the NTC sensor connected to the auxiliary sensor input	9.027 DPT_Temp_F
39	Measured temperature	Value ℃	Sends the temperature values expressed in degrees Celsius calculated by the device	9.001 DPT_Temp
39	Measured temperature	Value K	Sends the temperature values expressed in degrees Kelvin	9.002 DPT_Tempd

			calculated by the device	
39	Measured temperature	Value ℉	Sends the temperature values expressed in degrees Fahrenheit calculated by the device	9.027 DPT_Temp_F
40	HVAC mode feedback	Eco/Precom/Comf/Off	Sends feedback about the set HVAC mode	20.102 DPT_HVACMode
40	Operating setpoint feedback	Value ℃	Sends the operating setpoint values expressed in degrees Celsius	9.001 DPT_Temp
40	Operating setpoint feedback	Value K	Sends the operating setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
40	Operating setpoint feedback	Value ℉	Sends the operating setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
41	HVAC off mode feedback	Enable/disable	Sends feedback about the activation status of the HVAC off mode (building protection)	1.003 DPT_Enable
42	HVAC economy mode feedback	Enable/disable	Sends feedback about the activation status of the HVAC economy mode	1.003 DPT_Enable
43	HVAC precomfort mode feedback	Enable/disable	Sends feedback about the activation status of the HVAC precomfort mode	1.003 DPT_Enable
44	HVAC comfort mode feedback	Enable/disable	Sends feedback about the activation status of the HVAC comfort mode	1.003 DPT_Enable
45	Functioning type feedback	Heating/air cooling	Sends feedback about the set functioning type	1.100 DPT_Heat/Cool
46	Heating antifreeze setpoint feedback	Value ℃	Sends the heating HVAC off mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
46	Heating antifreeze setpoint feedback	Value K	Sends the heating HVAC off mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
46	Heating antifreeze setpoint feedback	Value ℉	Sends the heating HVAC off mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
47	Heating economy setpoint feedback	Value ℃	Sends the heating HVAC economy mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
47	Heating economy setpoint feedback	Value K	Sends the heating HVAC economy mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
47	Heating economy setpoint feedback	Value ℉	Sends the heating HVAC economy mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
48	Heating precomfort setpoint feedback	Value ℃	Sends the heating HVAC precomfort mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
48	Heating precomfort setpoint feedback	Value K	Sends the heating HVAC precomfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
48	Heating precomfort setpoint feedback	Value ℉	Sends the heating HVAC precomfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
49	Heating comfort setpoint feedback	Value ℃	Sends the heating HVAC comfort mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
49	Heating comfort setpoint feedback	Value K	Sends the heating HVAC comfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
49	Heating comfort setpoint feedback	Value ℉	Sends the heating HVAC comfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
50	Air cooling high temperature protection setpoint feedback.	Value ℃	Sends the air cooling HVAC off mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
50	Air cooling high temperature protection setpoint feedback.	Value K	Sends the air cooling HVAC off mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
50	Air cooling high temperature protection setpoint feedback.	Value ℉	Sends the air cooling HVAC off mode setpoint values expressed in	9.027 DPT_Temp_F

			degrees Fahrenheit	
51	Air cooling economy setpoint feedback	Value °C	Sends the air cooling HVAC economy mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
51	Air cooling economy setpoint feedback	Value °K	Sends the air cooling HVAC economy mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
51	Air cooling economy setpoint feedback	Value °F	Sends the air cooling HVAC economy mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
52	Air cooling precomfort setpoint feedback	Value °C	Sends the air cooling HVAC precomfort mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
52	Air cooling precomfort setpoint feedback	Value °K	Sends the air cooling HVAC precomfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
52	Air cooling precomfort setpoint feedback	Value °F	Sends the air cooling HVAC precomfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
53	Air cooling comfort setpoint feedback	Value °C	Sends the air cooling HVAC comfort mode setpoint values expressed in degrees Celsius	9.001 DPT_Temp
53	Air cooling comfort setpoint feedback	Value °K	Sends the air cooling HVAC comfort mode setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
53	Air cooling comfort setpoint feedback	Value °F	Sends the air cooling HVAC comfort mode setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
54	Current setpoint feedback	Value °C	Sends the active setpoint values expressed in degrees Celsius	9.001 DPT_Temp
54	Current setpoint feedback	Value °K	Sends the active setpoint values expressed in degrees Kelvin	9.002 DPT_Tempd
54	Current setpoint feedback	Value °F	Sends the active setpoint values expressed in degrees Fahrenheit	9.027 DPT_Temp_F
58	Aux input 1 - Switching	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
58	Aux inputs 1/2 - Switching	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
58	Aux input 1 - Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
58	Aux inputs 1/2 - Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
58	Aux input 1 - Scene	Execute/Store	Sends scene memorising/execution commands	18.001 DPT_SceneControl
58	Aux input 1 - 1 bit value	1/0 value	Send values 1/0	1.002 DPT_Bool
58	Aux input 1 - 2 bit value	On/Off forced positioning	Send 2 bit values	2.001 DPT_Switch_Control
58	Aux input 1 - 1 byte value	Unsigned value	Sends unsigned values (0..255)	5.010 DPT_Value_1_Ucount
58	Aux input 1 - 1 byte value	Signed value	Sends signed values (-128..127)	6.010 DPT_Value_1_Count
58	Aux input 1 - 1 byte value	% Value	Sends the percentage values (0%..100%)	5.001 DPT_Scaling
58	Aux input 1 - 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off)	20.102 DPT_HVACMode
58	Aux input 1 - 2 byte value	Unsigned value	Sends unsigned values (0..65535)	7.001 DPT_Value_2_Ucount
58	Aux input 1 - 2 byte value	Signed value	Sends signed values (-32768..32767)	8.001 DPT_Value_2_Count
58	Aux input 1 - 4 byte value	Unsigned value	Sends unsigned values (0.. 4294967295)	12.001 DPT_Value_4_Ucount
58	Aux input 1 - 4 byte value	Signed value	Sends signed values (-2147483648.. 2147483647)	13.001 DPT_Value_4_Count
58	Aux input 1 - 14 byte value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
58	Aux input 1 - Window status feedback	Open/Closed	Sends window status feedback according to auxiliary input status	1.019 DPT_Window_Door

59	Aux input 1 - Brightness control	Increase/decrease	Sends brightness dimming commands	3.007 DPT_Control_Dimming
59	Aux inputs 1/2 - Brightness control	Increase/decrease	Sends brightness dimming commands	3.007 DPT_Control_Dimming
59	Aux input 1 - Louvres stop/adjustment	Stop/Step	Sends stop movement/louvres adjustment commands	1.007 DPT_Step
59	Aux inputs 1/2 - Louvres stop/adjustment	Stop/Step	Sends stop movement/louvres adjustment commands	1.007 DPT_Step
61	Aux input 2 - Switching	On/Off	Sends dimmer on/off commands	1.001 DPT_Switch
61	Aux input 2 - Shutter movement	Up/down	Sends shutter up/down movement commands	1.008 DPT_UpDown
61	Aux input 2 - Scene	Execute/Store	Sends scene memorising/execution commands	18.001 DPT_SceneControl
61	Aux input 2 - 1 bit value	1/0 value	Send values 1/0	1.002 DPT_Bool
61	Aux input 2 - 2 bit value	On/Off forced positioning	Send 2 bit values	2.001 DPT_Switch_Control
61	Aux input 2 - 1 byte value	Unsigned value	Sends unsigned values (0..255)	5.010 DPT_Value_1_Ucount
61	Aux input 2 - 1 byte value	Signed value	Sends signed values (-128..127)	6.010 DPT_Value_1_Count
61	Aux input 2 - 1 byte value	% Value	Sends the percentage values (0%..100%)	5.001 DPT_Scaling
61	Aux input 2 - 1 byte value	HVAC mode	Sends the HVAC modes (auto/comfort/pre-comfort/economy/off)	20.102 DPT_HVACMode
61	Aux input 2 - 2 byte value	Unsigned value	Sends unsigned values (0..65535)	7.001 DPT_Value_2_Ucount
61	Aux input 2 - 2 byte value	Signed value	Sends signed values (-32768..32767)	8.001 DPT_Value_2_Count
61	Aux input 2 - 4 byte value	Unsigned value	Sends unsigned values (0.. 4294967295)	12.001 DPT_Value_4_Ucount
61	Aux input 2 - 4 byte value	Signed value	Sends signed values (-2147483648.. 2147483647)	13.001 DPT_Value_4_Count
61	Aux input 2 - 14 byte value	ISO 8859-1 characters	Sends characters codified with ISO 8859-1 standard	16.001 DPT_String_8859_1
62	Aux input 2 - Brightness control	Increase/decrease	Sends brightness dimming commands	3.007 DPT_Control_Dimming
62	Aux input 2 - Louvres stop/adjustment	Stop/Step	Sends stop movement/louvres adjustment commands	1.007 DPT_Step

12 ETS programming error feedback

The device is able to detect and therefore indicate various programming errors via the alternative flashing for 500 ms of the green LED and 500 ms of the red LED.
If multiple errors are detected, the error signalling (green/red flashing) will continue until the ETS application is downloaded again with the necessary corrections.

12.1 Error table

Possible errors
The constraints between the setpoints of the various HVAC modes belonging to the same functioning type are not respected: - $T_{anti-freeze} \leq T_{economy} \leq T_{precomfort} \leq T_{comfort}$ in heating mode - $T_{comfort} \leq T_{precomfort} \leq T_{economy} \leq T_{high\ temp.\ protection}$ in air cooling mode or if the control type is setpoint, the constraints are - $T_{antifreeze} \leq T_{operation}$ in heating mode - $T_{operation} \leq T_{high\ temp.\ protection}$ in air cooling mode
The HVAC mode setpoints are out of the maximum range - The comfort/economy/precomfort setpoints are not between 5 °C and 40 °C - The antifreeze setpoint is not between 2 °C and 7 °C - The high temperature protection setpoint is not between 30 °C and 40 °C or if the control type is setpoint - The operating setpoint is not between 5 °C and 40 °C - The antifreeze setpoint is not between 2 °C and 7 °C - The high temperature protection setpoint is not between 30 °C and 40 °C 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).
Connection error of the communication objects dedicated to sending commands to the actuation devices (valves and fancoils): - if different control logic is enabled but command objects for the heating and air cooling solenoid valves are connected in the same group address - Coherency between the connection of the speeds from the fancoil. With regard to the command objects for 1 bit fancoil speeds, they can be linked to the same address or to different addresses, however it is important to maintain coherency between the pairs: if the two objects for controlling heating and air cooling speed 1 are linked to two addresses that are the same, also the pairs for speeds 2 and 3 must be the linked to two addresses that are the same. In this manner, if linked to two different addresses, also the pairs for speeds 2 and 3 must be linked to two different addresses. If a solenoid valve is connected (heating or air cooling), also the respective speeds must be connected, and vice versa. In common logic mode, and if connecting the heating fan speeds, they must also be connected for air cooling, and vice versa. - if the second stage is active and the command objects for the relative solenoid valves are not connected. 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).