EKINEX

Application manual

4/8-fold KNX 20venti series pushbutton

EK-E20-TP-…
EK-E20-TP-…-R

Summary

Revision	Modifications	Author	Date
1.0	First version	G. Schiochet	10/09/2020

1 Scope of the document

This application manual describes application details for ekinex® 4/8-fold pushbutton interface 20venti series EK-E20-TP-… (for round or square wall mounting box) and EK-E20-TP-…-R (for rectangular wall mounting box).

The document is aimed at the system configurator as a description and reference of device features and application programming. For installation, mechanical and electrical details of the device please refer to the technical description datasheet.

Application manual and application programs for ETS are available for download at www.ekinex.com.

Item	File name (## = release)	Version	Device rel.	Update
Product datasheet	STEKE20TP_EN.pdf	EK-E20-TP-… EK-E20-TP-…-R	1.0	09 / 2020
Application manual	MAEKE20TP_EN.pdf			
Application program	APEKE20TP##.knxprod			

You can access the most up-to-date version of the full documentation for the device using following QR codes:

EK-E20-TP-…:   EK-E20-TP-…-R:

2 Product description

The ekinex® 4/8-fold KNX 20venti series pushbutton unit is a S-mode KNX wall-mounting device for on/off switching of loads, dimming of lighting devices, control of motor drives or other programmable switching and control functions.

The pushbutton is equipped with an integrated temperature sensor and can act as a room probe or thermostat, both in heating and cooling mode.

When acting as a room thermostat, the device is not equipped with a user interface for displaying room conditions and modifying the setpoint temperature; therefore, it must be paired with an external supervision device. Terminals such as radiators, electrical radiators and radiant panels can be controlled.

The proximity sensor, where available, allows the activation of the backlight and other functions.

This unit is equipped with an integrated KNX bus communication module and is designed for wall installation; each button can be freely programmed to perform a bus function and has RGB LEDs, that can be used, for example, as a status signal or night orientation light.

For final use, this unit must be completed with frontal plates for commands and a set of either 4 or 8 rockers, which must be ordered separately in order to obtain the desired aesthetic look; regardless of the detail, several kinds of plates are available (square or rectangular) which can be combined in order to obtain different combinations.

The device is powered by the KNX bus line with a 30 VDC SELV voltage and does not require auxiliary power.

Product code	Nr. of rockers / position	Rockers type	Proximity sensor
EK-E20-TP-4LS	4 / left	LED	No
EK-E20-TP-4LD	4 / right		
EK-E20-TP-4TS-P	4 / left	Text / symbols	Yes
EK-E20-TP-4TD-P	4 / right		
EK-E20-TP-8L-… *	8	LED	No
EK-E20-TP-8L-… *		Text / symbols	Yes
(*) – Add "R" for the rectangular wall box version			

Table 1 – Product codes and features

The supply includes, inside the box:

- 1 adapter;
- 1 metallic support for mounting on round or square 60 mm (EK-E20-TP-…) or rectangular 83.5 mm (EK-E20-TP-…-R) box
- 2 pairs of fixing screws;
- 1 KNX terminal block for the connection of the bus line.

2.1 Completion of the device

For full installation and operation, the unit must be completed with:

- A set of rockers (depending on the number of buttons, 4 or 8, on the colour and finish);
- A square plate with at least 1 window (30 x 60 mm or 60 x 60 mm in size) or a rectangular plate with at least 1 window (60 x 60 mm in size) from the ekinex® series.

Rocker kit codes *	Type	Nr.	Shape / Dimensions L x H [mm]
EK-T4R-20-BT-…-YYY	Symbols / text	1	Rectangular / 30 x 15
EK-T4R-20-BL-…-YYY	LED + Symbols / text		
EK-T4R-20-…	-	4	
EK-T4R-20-BL-…	LED		
(*) – To be completed with the extension for colour and finishing			

Table 2 – Rockers codes

The rockers are available in plastic material and in several variants for colour and finishing:

• With RGB LEDs backlight only;

• With backlighting of text and/or symbols with RGB LEDs.

Plate code *	Shape	Window dimensions L x H [mm]	Serie
EK-SQT-… *	Square	30 x 60	*Surface*
EK-DQT-… *			*Deep*
EK-SQS-… *		60 x 60	*Surface*
EK-DQS-… *			*Deep*
EK-SRS-… *	Rectangular	60 x 60	*Surface*
EK-DRS-… *			*Deep*
(*) – To be completed with the extension for colour and finishing			

Table 3 – Plate codes

3 Main functional characteristics

The main functions of the device are:

- On/off switching of single loads or groups of loads
- Dimming of lighting devices
- Control of motor drives (for roller shutters, blinds, curtains, etc.)
- Measuring of room temperature through integrated sensor
- Backlight and other features activation through integrated sensor
- Room temperature regulation
- Logic functions
- Sending of values (temperature, etc.) on the bus
- Switching to forced functioning (lock)
- Recalling and saving of scenes
- Different functions programmable for short pressure / long pressure of a rocker
- Status feedback or orientation nightlight through programmable RGB LEDs

3.1 Rocker functions

Once a button is pushed, the device sends the telegram (or sequence) that was associated with it in the programming phase on the KNX bus.

3.2 LED indicators

Each button is provided with high efficiency RGB LEDs, that can be freely programmed (also with functions independent of those of the buttons), both as functional indications and for obtaining aesthetic effects or as night orientation lights.

For a more detailed description of the position of the LEDs and the relative configuration parameters, refer to the application section of the manual.

3.3 Customization of rocker plates

Rocker plates can be customized with predefined symbols and texts. On request, a customization is also possible with symbols and texts chosen by the customer. For more information see the standard library on the ekinex® catalogue or the website www.ekinex.com.

i	For further technical information, please also refer to the product datasheet available on the website www.ekinex.com.

3.4 Proximity sensor (available in symbols / text versions only)

Through the proximity sensor it is possible to keep the button in a stand-by state, setting a maximum level of brightness of the LEDs and reactivate the backlighting of the keys only when the user approaches the device. The detection distance and sensor sensitivity are configurable, to avoid false detections.

It is also possible to activate the sending of values or sequences when the sensor detects the presence of the user within the set range of action, so as to activate automations or signal the presence of a user (it is not however to be considered as a presence sensor).

3.5 Technical data

Characteristics	Value
Device	KNX bus S-mode
Communication	According to standard KNX TP1
Microcontroller	MSP430F541
Power supply	SELV 30 Vdc through KNX bus
Current consumptions	< 17 mA
Max power from bus	< 500 mW
Usage	Dry internal rooms
Environmental conditions	• Operating temperature: - 5 ... + 45°C • Storage temperature: - 25 ... + 55°C • Transport temperature: - 25 ... + 70°C • Relative humidity: 95% not condensing
Programming elements	Simultaneous pressing of 2 keys for at least 4 seconds
Displaying elements	Button backlighting via RGB LEDs
Integrated sensors	Temperature, proximity
Installation	Wall installation on a round or square flush-mounting box with a distance of 60 mm between the fixing holes, or rectangular with a distance of 83.5 mm between the fixing holes
Pollution degree	2 (according to IEC 60664-1)
Installazione	??
Protection degree	IP20
Dimensions (L x H x P)	80 x 80 x 20 mm (EK-E20-TP-…), 122 x 80 x 20 mm (EK-E20-TP-…-R)

4   Switching, display and connection elements

On the front of the device the supports for the interlocking assembly of the buttons, the LED light diffusers and the temperature and proximity sensors are located.

At the rear there are the terminals for connection to the KNX bus, the springs for hooking to the mounting support and the seat of the product label.
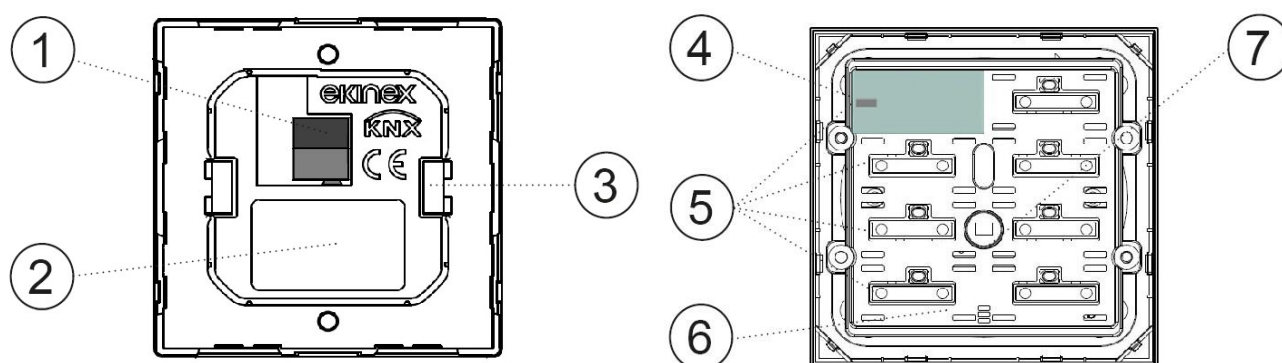
1. KNX bus line connection terminal
2. Label
3. Springs for attachment to the mounting support
4. Rocker 30 x 15 mm
5. LED lightguides
6. Position of temperature sensor
7. Position of proximity sensor

Table 4 - Device elements

4.1 Versions

The device is suitable for mounting on a round or square flush-mounting box, with fixing holes with a 60 mm center distance, or on a rectangular 3-position flush-mounting box according to Italian standard, equipped with fixing holes with a 83.5 mm distance. Each device is supplied with a metallic mounting support which, if necessary, can also be ordered separately; a plastic adapter and a connection terminal to the bus line complete the supply.
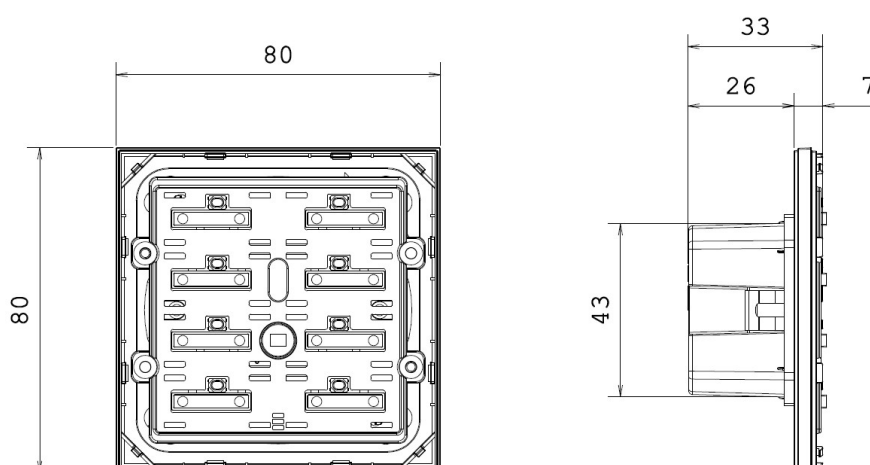
Figure 1 - EK-E20-TP-…: front and lateral sight
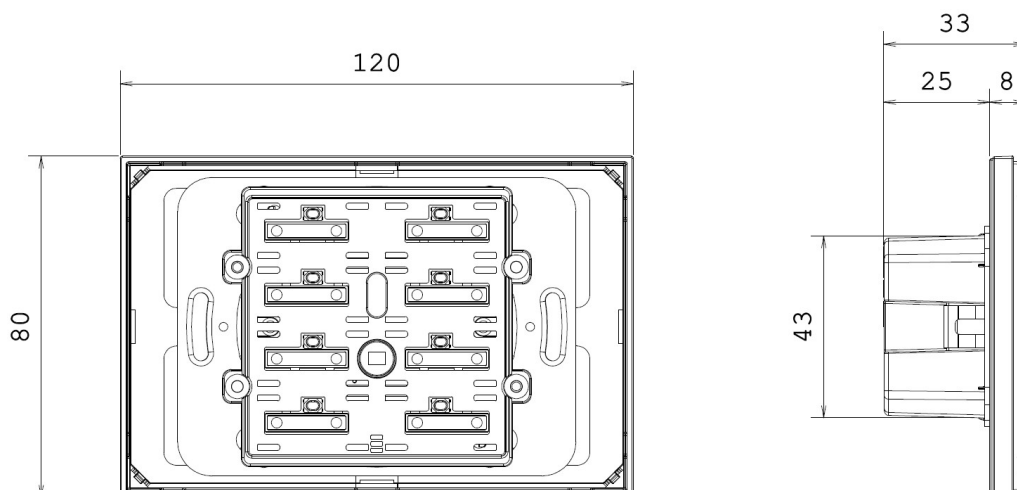
Figure 2 - EK-E20-TP-…-R: front and lateral sight

5 Configuration

The functionality of the device is determined by the settings done via software.

In order to configure the device, the ETS4 development tool (or later) and the ekinex® application program dedicated to the device (the name is **APEK20TPxx.knxprod**) is required. The updated version can be downloaded from the website www.ekinex.com.

The application program allows you to access, within the ETS4 environment, the configuration of all the working parameters of the device. The program must be loaded into ETS (alternatively, the entire database of ekinex® products can be loaded in one operation), after which all device specimens of the type considered can be added to the project being defined.

The configurable parameters for the device will be described in detail in the following paragraphs.

The configuration can be, and generally will be, defined completely in off-line mode; the transfer of the set configuration to the device will therefore take place in the programming phase, described in the following paragraph.

Product code	EAN	N. of buttons	ETS application program (## = revision)	Communication objects (Max n.)	Group addresses (Max n.)
EK-E20-TP-… EK-E20-TP-…-R		4 o 8	APEKE20TP##.knxprod	267	254

i	The configuration and commissioning of KNX devices require specialized skills. To acquire these skills, you should attend the workshops at KNX certified training centers. For further information, please visit www.knx.org.

6   Configuration and commissioning

After the device configuration has been defined within the ETS project according to the user's requirements, to carry out the programming it is necessary to do the following operations:

- electrically connect the device, as described in the data sheet, to the KNX bus in the final destination system or in a reduced system, specially composed for programming. In any case, the system will contain an interface device to the PC on which the KNX environment is installed;
- switch the device operation to programming mode by pressing simultaneously, for at least 4 seconds, the first and fourth buttons in the 4-key configuration or the first and fourth buttons on the left side for the 8-key configuration (see step 1 in Figure 3) ;
- releasing the buttons, all the LEDs flash red (step 2 in Figure 3); the device is in programming mode;
- from the ETS environment, start programming (which in case of first configuration must include the physical address to be given to the device).

At the end of the program download, the device automatically returns to operating mode; all LEDs start working as programmed. The device is now programmed and ready for operation in the system.
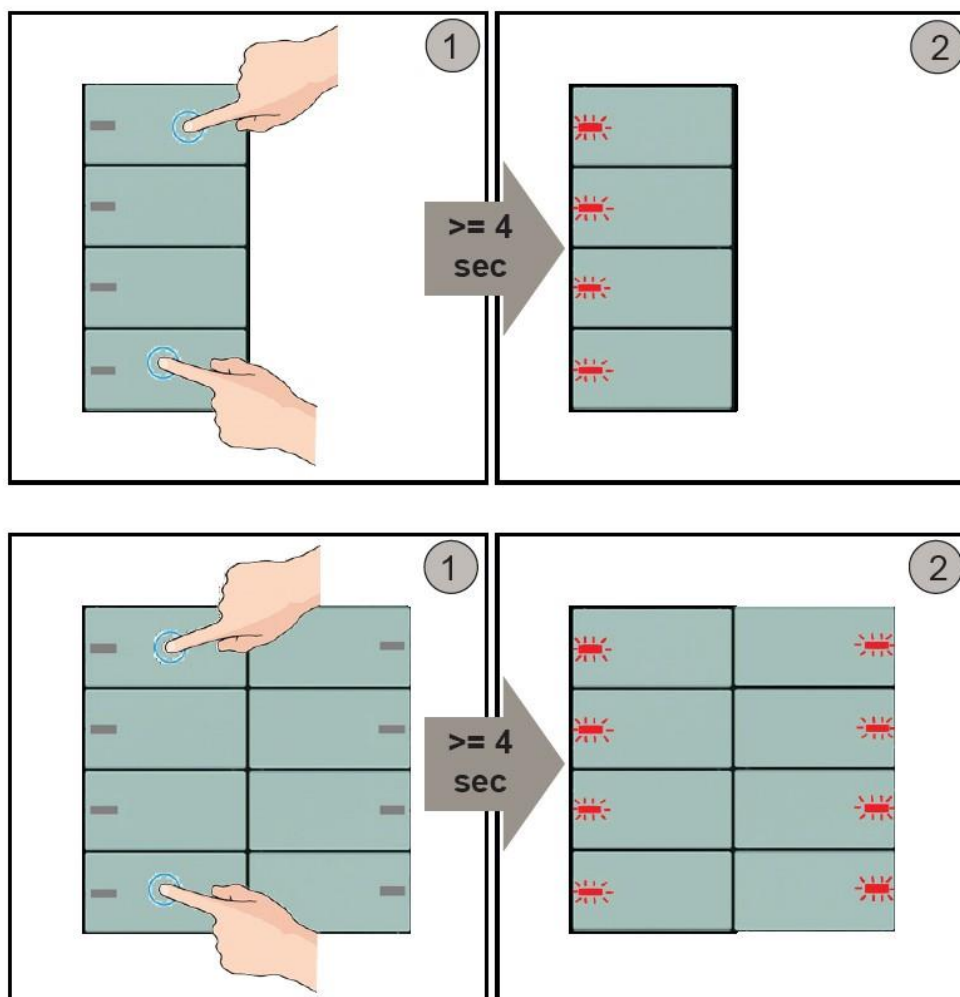
Figure 3 - Device application programming

6.1 Device reset

To reset the device, keep the same buttons combination for entering the programming mode pressed, for at least 10 seconds (Figure 3). All the LEDs start blinking, then they all switch off: the reset was carried out. Now you need to address and configure again the device via ETS.

7   Function description

After switching on the bus, which also acts as a power supply, the device becomes fully functional after a very short time needed for reinitialization. A delay is programmable for the device to become active on the bus in order to avoid a bus traffic overload during the first moments of start-up of the whole network.

In case of a bus power failure (voltage lower than 19 Vdc for 1 s or more), the device becomes unreactive: before the power supply becomes insufficient, the status is internally stored. The timing functions are not active, neither are the programmed group addresses.

As soon as the bus voltage is restored, the device will resume operation in its previous state (which is saved on power fail), unless different initialization settings are programmed.

7.1 Offline operation

A fully unprogrammed device does not operate in standby mode. Since the operation relies entirely on the exchange of information through communication objects, there is no part of the device that can operate independently from a KNX bus.

7.2 Online operation

In general the device works like a configurable digital sensor that is listening to own inputs or outputs of other devices. On input events the device performs output functionality over KNX bus like sending values or controlling external devices like KNX actuators.

7.3 Software operations

The main purpose of the software is following:

- Handle user pushbutton presses and generate bus telegrams according to the assigned functions;
- Implement pushbutton interlock and timing functions;
- Handle incoming bus messages in order to update the status of pushbutton activations and LED indicators;
- Respond to bus messages requesting feedback on the status of the inputs.

The status of the device and specifically of its entities (input activation status and LED indicators) relies on KNX *communication objects*, which can be freely defined and bound in various ways to the physical elements of the device; these communication objects acts as *state variables* for the device.

There are also special events on which it is possible to trigger additional features. These events are the bus failure and recovery, and the download of a new configuration with ETS.

7.4 Pushbutton inputs

The press of a pushbutton can be bound to different effects on a state variable.

7.4.1 Pushbutton input events

A button press can be handled either as an "on-off" event ("on" means when the button is pushed, "off" when it is released), or as a "short press - long press" event (whereby a time period can be defined to discriminate the duration of the "long" from the "short" press).

In both cases, for each of the two available events a separate action can be assigned that operates on a selected variable (actually, more than one; see below for details).

7.4.2 Lock function

For each input (or channel if inputs are coupled, see below), a lock feature can be enabled which allows to block the operation of an input through a message on a communication object.

When in a locked state, the input is effectively disabled.

A value (for each transition) can be specified to be assigned to the communication object upon entering or exiting the locked state.

The locked state can also be automatically activated when the bus is connected.

7.4.3 State variables (communication objects)

The variable that is changed by the input events can be one of the types available for KNX communication objects, i.e. for instance a 1-bit value (on-off), a 2-bit value or an integer value of larger size.

In all cases, each of the two events can:

- change the value of the variable to one of two definable values within its range (which is trivial in the case of the 1-bit value);
- toggle between the two defined values
- do nothing (value is unaffected)

This state variable, once assigned a group address, is actually a **KNX communication object**; as such, it undergoes the usual rules for communication objects, among which – for instance – the effect of flags to determine how the change of value affects the transmission of the objects.

7.4.4 Binding between Events and Communication objects

The above description is a little simplified in order to ease comprehension; as a matter of fact, to each event can be assigned not just one, but several communication objects (up to 8), even of different types. Each of these communication objects can have its own behaviour and its own associated value set.

7.4.5 Repeated send

For most features, is it possible to set the device to send a telegram not just when a value changes as a consequence of an input transition, but also at regular intervals whenever that value setting is active.

This behaviour, also referred to as Cyclical Transmission, can be set separately for each of the two values that are associated to an input (or both, or none of them).

If an input is set to "*send values or sequences*" mode, repeated send is not available if more than 1 Communication Object is assigned to that input.

7.4.6 Single or independent input mode

Each single input can be configured for one of following different features:

1. *Send values or sequences*
   An event triggers the transmission on the bus of configurable values or sequence of values.
   These values can be of a logical type or a numerical type with a different size.
   A sequence of values can be made of up to 8 communication objects of different value types.
   Time delays can set between values in the sequence.

2. *Dimmer control*
   This mode is intended to be used with dimming actuators for the control of lighting devices.
   The functionality is triggered on short press and long press events.
   On short press events, the device sends on/off telegrams to the dimming actuator.
   On long press events, the dimming percentage is varied up or down until the button is released.

3. *Shutter or Venetian blind control*
   This mode is intended to be used together with actuators for the control of motorized blinds, shutters and similar devices. These actuators have functions for blind opening and closing; two movement types are selectable, i.e. continuous movement and stepwise movement. On input events, the device sends operation telegrams to the actuators.
   The operation is configurable through following parameters:
   - If *toggle* mode is enabled, on each activation of the same input the movement direction is inverted; if it is disabled, the movement direction is fixed and it can be set to "up" or "down".
   - If *blinds* mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.

4. *Scene function output*
   This mode is intended to be used together with several KNX actuators that support using a scene function; this function allows storing and recalling a communication object value on an actuator.
   In this mode, the role of the device is to send a "store / recall scene" telegram to the actuator on a long / short press event.
   This mode has two possible configurations:
   - Activate pre-set scene on short press, and store current setting as scene value on long press
   - Activate two different scenes on long and short press.

7.4.7 Dimming function

The dimming function is a device application profile included in KNX specifics. Those specifics define the basic requirements for interface mechanisms, in addition to which some aspects regarding the operating modes, peculiar for each device (for both command or actuation devices) are to be considered.

> **i**
>
> *All the information contained in this section have the purpose of illustrating specific device functions, and therefore are not to be necessarily considered exhaustive or applicable to other cases. In order to obtain a complete or generally applicable documentation, please refer to the official KNX documentation.*
>
> *For further information, visit the website www.knx.org.*

The dimmer control type is essentially based on a 4-bit communication object, whose data has the following format:
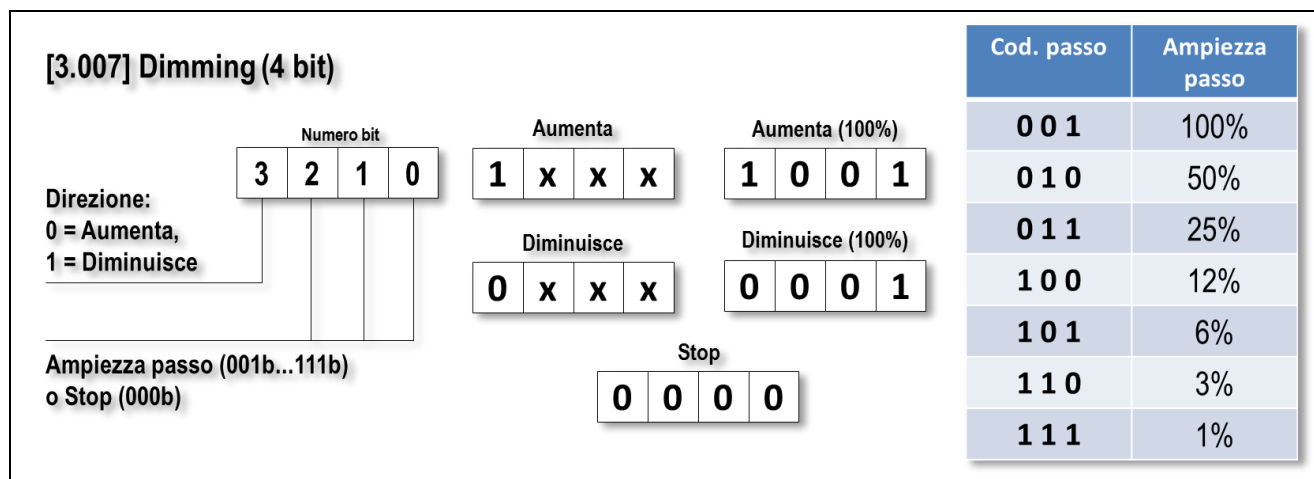
The transmission of telegrams containing data of such format tells the actuator to perform an increase or a decrease, by an amplitude equal to the specified step, or to stop an ongoing variation.

The increase or decrease of an intensity value by the actuator is not instantaneous but gradual; therefore, an increase / decrease command with interval equal to the maximum allowed value has the effect of starting the intensity variation in the desired direction, which will continue until the maximum (or minimum) value has been reached. Such variation can be stopped, once the desired intensity value has been reached, by sending a "stop" command.

It is normally possible, and desirable, to have the possibility to instantly switch on or off the load (i.e. to instantaneously bring its value from 0% to 100%). In order to achieve that, an "On / Off" command based on another object is used; this is the same object used for the normal load switch, which is present also in absence of a dimming mechanism.

The command device – in this case, the rocker unit – will define the operations to generate a sequence of commands with an opportune order and time interval, in order to achieve the desired command effect.

The defined operations and related commands are the following:
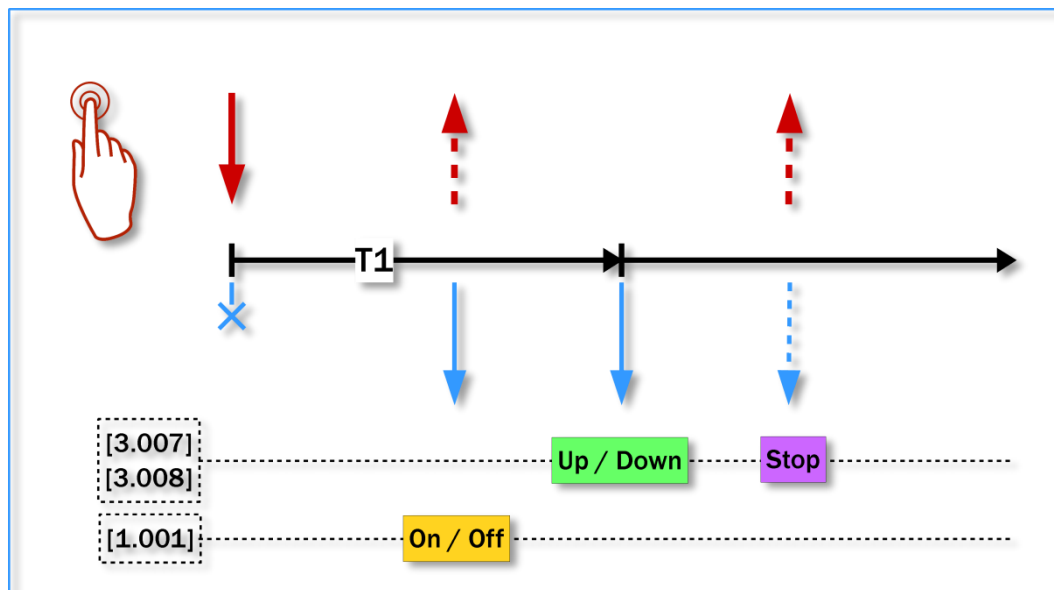
<div align="center">**Figure 4 - Dimmer mode command sequence**</div>

- Short press: instantaneous switch on / off (toggle on / off on a switch object);
- Long press: increase / decrease value until 100% / 0%;
- Release: stop increase / decrease.

Please note that the same mechanism can be applied to the shutter or venetian blind control (in that case, "maximum / minimum" is substituted with "open / close"). For this purpose, the data type (DPT) 3.008 exists, whose structure and values are identical to those already described; in order to control a shutter with the same mode, it is possible to connect a communication object type 3.007 command side, to an object type 3.008 actuator side (if foreseen). In this case, obviously, the object type "On / Off" which allows instantaneous switch on / off is not used.

7.4.8   Shutter / venetian blind function

The "Shutter / venetian blind" function is a bundle of application profiles included in KNX specifics. As for dimming function, such specifics define basic requirements related to interface mechanisms, in addition to which some aspects regarding the operating modes, peculiar for each device (for both command or actuation devices) are to be considered.

> **i**
>
> *All the information contained in this section have the purpose of illustrating specific device functions, and therefore are not to be necessarily considered exhaustive or applicable to other cases. In order to obtain a complete or generally applicable documentation, please refer to the official KNX documentation.*
>
> *For further information, visit the website www.knx.org.*

In case of shutters, the actuator brings a mechanic component from one point to another in a gradual way, with possibility to stop at intermediate points; the command is carried out by 2 lines which, when activated (one line at a time) make the actuator move in the corresponding direction.

A venetian blind is essentially a shutter that, in addition to the up / down movement, is also equipped with slats that can be opened / closed same way as a shutter (gradual movement between extreme points). The peculiarity is that normally the slat's movement and the up / down movement are controlled by the same two lines; therefore, the activation of the electromechanic device must be carried out according to a specific sequence. For further detail please check the actuator's documentation; in this document all we need to point out is that, command side, the control sequences can be considered as independent from these aspects.

The basic control for a shutter or a venetian blind is essentially based on three 1-bit communication objects:

- [1.008]   Move Up/Down
- [1.007]   Stop – Step Up/Down
- [1.017]   Dedicated Stop

The effect of the commands linked to these objects is the following:

- The command "Move", when received, starts the movement of the shutter in the indicated direction.
- The command "Stop – Step" has two functions: if the shutter is stopped, it moves by one step in the indicated direction (the duration is set in the actuator), if not, it stops the ongoing movement without doing anything else.
- The command "Stop" just stops the ongoing movement.

In addition, other types of control objects are normally available ("dimmer" type, absolute position, etc.) but they are not part of the basic control on which this manual is about; for further information please refer to the actuators' manual or KNX specifics.

In the simplest version, on command side:

- In order to control a shutter at least the objects "Move" and "Stop" are required (and present).
- In order to control a venetian blind at least the objects "Move" and "Stop – Step" are required (and present).

On actuator side – whether it is a shutter or a venetian blind – the presence of objects "Move" and "Stop – Step" must be guaranteed, while the presence of the object "Stop" is optional (but usually present).

As for the operations to perform on the command device, in our specific case the rocker unit, in order to generate a sequence of these commands with the proper order and time interval, there are multiple possibilities.

In case of ekinex input devices, two modes are available – indicated as "Shutter" and "Venetian blind" based on their typical destination – which are illustrated in the following figure.
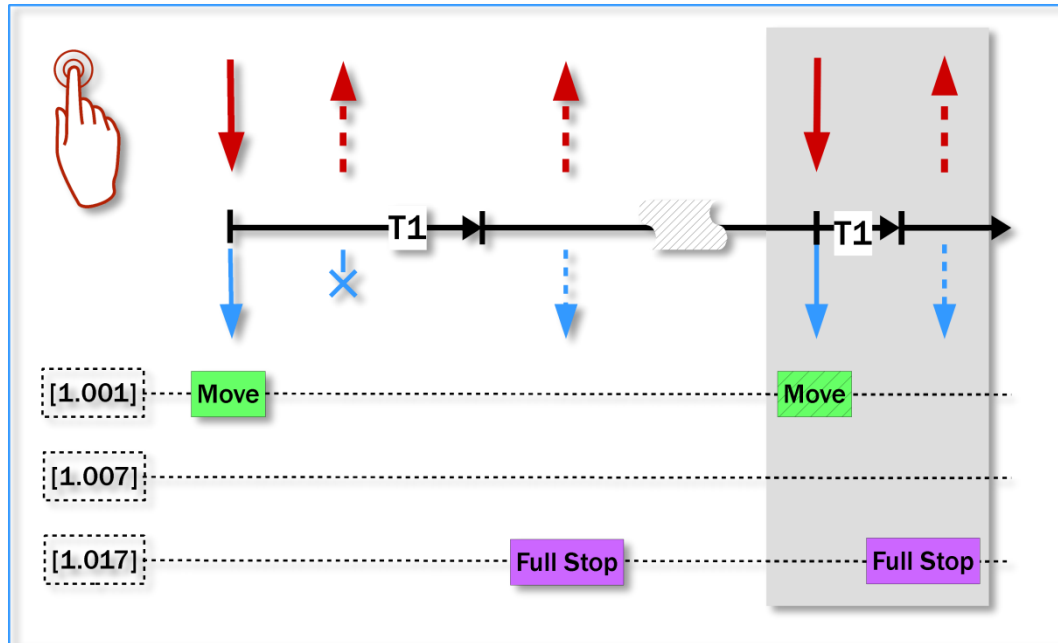
Figure 5 - "Shutter" mode command sequence

In "Shutter" mode, when a rocker is pressed – or a digital input is activated – the shutter starts moving in the corresponding direction (which can be alternatively in the two directions if the rocker is in independent mode and has been configured as *toggle*).

If the rocker is released quickly, the shutter will continue its run until full opening or closing; it is still possible to stop it by pressing again the rocker with a long press.

If the rocker is pressed with a long press, when it is released – which will be in correspondence with the desired position – the shutter will stop.
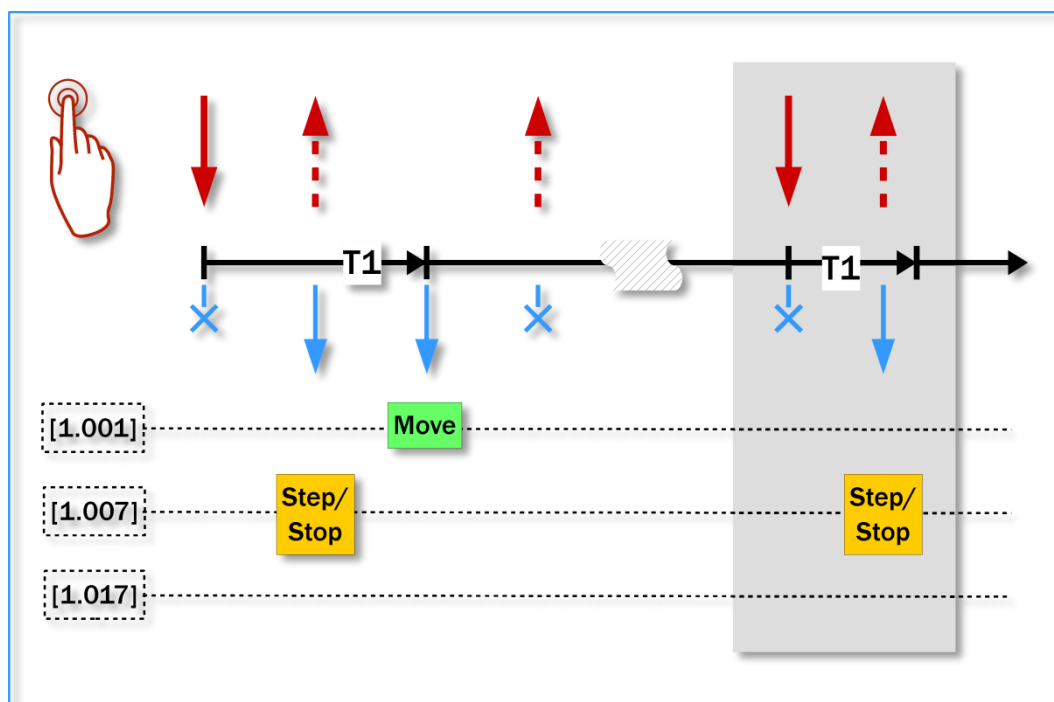
Figure 6 - "Venetian blind" mode command sequence

In "Venetian blind" mode, on release of a rocker after a short press, the venetian blind performs a step; this operation, normally – i.e. even if the actuator is indeed configured for a venetian blind – is used for the slats regulation.

If the rocker is pressed with a long press, when the threshold time is reached, a "Move" command is issued, which will bring the venetian blind to full open or close. In order to stop it at an intermediate position, the rocker needs to be pressed again (short press).

7.5 RGB LED indicators

The RGB LED indicators associated with each input are programmable from ETS.
To select the colour of an RGB LED, we have chosen to display a preview of the color using the drop-down selector.

In the configuration of any LED, by clicking on the symbol on the right [symbol], the palette for the selection of colours is displayed.
An example is shown in Figure 7.
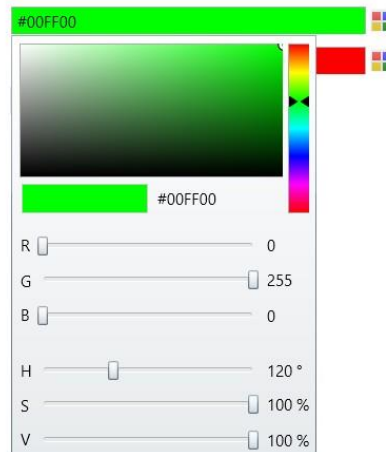
Figure 7 - Colour palette

A color reference is available at the following link:

https://www.rapidtables.com/web/color/html-color-codes.html

Whenever we discuss the color palette in this document, we will refer to this type of color selection.

7.5.1   Individual parameters

The activation of RGB LEDs can be set as follows:
- Fixed value (always on or always off) by specifying an intensity and hue value using the color palette from ETS;
- Dependent on the status of the corresponding input (open or closed). With this option it is possible to set one colour when the input is at rest and another colour when the input has been activated, with the possibility of returning to the rest state with a personalized delay;
- State from the bus with thresholds (described below).

By selecting the status from the threshold bus for the activation of RGB LEDs, it will be possible to select the type of data to be monitored, from the following options:
- 1-bit boolean;
- 2-bit Boolean;
- 1 signed byte;
- 1 unsigned byte;
- 2 signed bytes;

- 2 unsigned bytes;
- 2 byte float.

In the case of the 1-bit and 2-bit variable, the colour will change when the value matches the value. In the case of byte variables, the colour selection will be made by means of thresholds; this means that when the value received by the bus is between two thresholds it will assume the set colour.
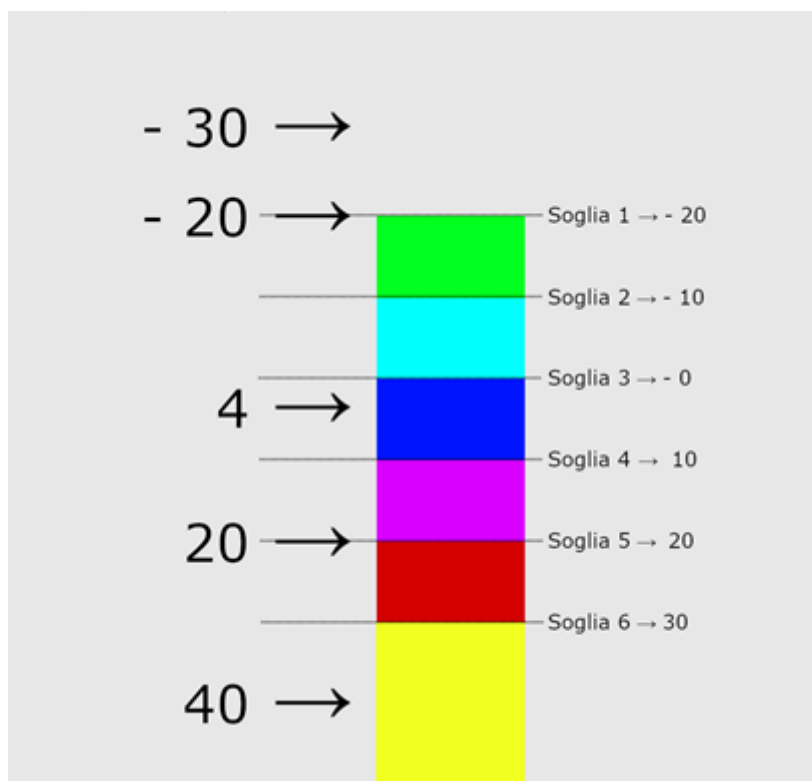
Figure 8 - Example of bus state operation with thresholds for RGB LEDs

Taking as a reference the graph above, the color of RGB LEDs will vary according to the command value in the following way (in the example, it is assumed to have set 6 thresholds, respectively at -20, -10, 0, 10, 20, 30):

- With value = -30 since no threshold covers this value, the LED will be off;
- With value = -20, since threshold 1 is set at -20, the LED will be green;
- At value = 4, threshold 3 starts, setting the blue color, then the LED will light up in blue;
- With value = 20, threshold 5 starts, which sets the color red, then the LED will light up red;
- At value = 40, since the threshold 6 starts at 30 and ends up to 127, therefore the color of the LED will be that expected by this threshold (in the example, yellow).

7.5.2 Technical Alarm indicator

A particular indication function can be activated on the device: if enabled, the RGB LEDs located at the four corners of the device can be activated in flashing mode by means of a KNX telegram. In particular, the activation of the technical alarm generates the switch on of the RGB LEDs in red colour.

The typical purpose of this indication is to signal an alarm condition, but it can also be used for any other signalling.

> **i**   **Note on LED flashing settings**
>
> When the LED status from bus with threshold mode is activated, an alarm can be notified by setting a flashing time for each LED. However, if the pushbutton is provided with a proximity sensor, as long as an alarm is active the Stand-By mode is disabled to allow better visibility of the flashing LEDs.

7.6 Temperature sensor

The value from the embedded temperature sensor, unless it is disabled, can be read from the bus by other devices. In addition, their behaviour can be modified through following parameters:

- The raw value read from the sensor can be corrected with a small offset (-5 °C to +5 °C in steps of 0.5 °C), in order to compensate for environmental factors and achieve a better precision.
- The sensor value can periodically be sent on the bus with a specified transmission interval, and whenever a specified variation occurs.

8 Room thermostat

Use of sensors

The temperature controller integrated inside the pushbutton allows the room temperature acquisition in the following ways:

1) from the temperature sensor integrated inside the device;
2) via bus from another KNX device, e.g. another ekinex® pushbutton

In order to optimize or correct the temperature regulation in particular cases (big rooms, when there is a strong asymmetry in temperature distribution, when the pushbutton is installed in wrong or unsuitable positions, etc.) the device can use a weighted mean between two temperature values. The weights are assigned according to the *Relative weight* parameter, which assigns a proportion to the values.

> **i** **Note on mounting position**
>
> If the integrated temperature regulator is used, the device must be preferably installed on an internal wall, at 1,5 m of height and at least 0,3 m of distance from doors. The device cannot be installed near heat sources such as radiators or domestic appliances or in positions subjected to direct solar irradiation. If necessary, for the regulation can be used a weighted mean value between the measured temperature acquired by the integrated sensor and a value received via bus from another KNX device.

Applications

The applications that can be configured are peculiar to thermal plants with a single stage and concern the following terminals: radiators, electric radiators and radiant panel systems.

The temperature control can be:

- two point control with hysteresis, ON-OFF command type;
- proportional-integral, with ON-OFF command, PWM or continuous type.

8.1 Control algorithms

Figure 9 shows the components of a common generic control system for ambient temperature. The room thermostat measures the actual temperature of the air mass (T_{eff}) and constantly compares it to the setpoint value (T_{set}).
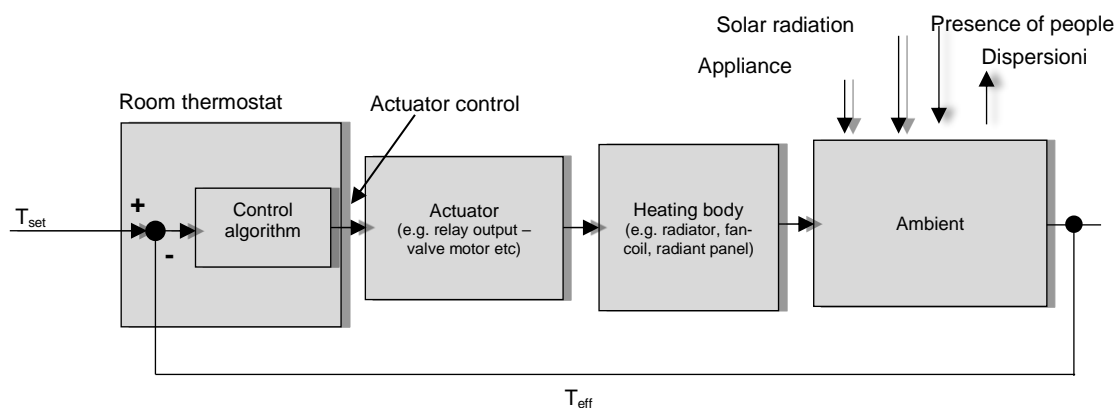
Figure 9 - Generic control system for room temperature regulation

The control algorithm, basing on the difference between T_{set} and T_{eff}, processes a command value which can be analogue or On / Off type; the command is represented by a CO that is transmitted via bus, periodically or event based, to a KNX actuator device.

The output of the actuator device is the driving variable of the control system, which can be e.g. a flow rate of water or air. The control system realized by the room thermostat is of feedback type, namely the algorithm takes into account the effects on the system in order to change the control action on the same entity.

8.2 Two-point control with hysteresis

This control algorithm, which is also known as On / Off, is the most classic and popular. The control provides for the on / off switching of the system following a hysteresis loop, i.e. two threshold levels are considered for the switching instead of a single one.
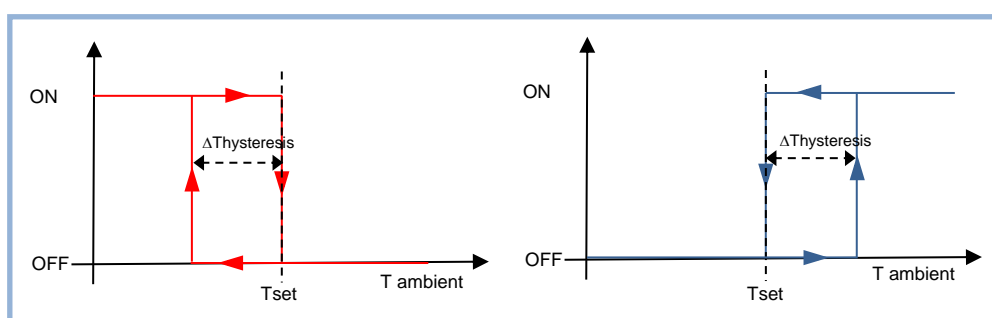
Figure 10 - Temperature control

Heating mode: when the measured temperature is lower than the value of the difference ($T_{set} - \Delta T_{hysteresis}$), whereby $\Delta T_{hysteresis}$ identifies the differential adjustment of the boilers, the device activates the heating system by sending a message or KNX telegram to the actuator that handles the heating system; when the measured temperature reaches the desired temperature (Setpoint), the device disables the heating system by sending another message. In this way, there are two decision thresholds for activation and deactivation of the heating, the first being the level ($T_{set} - \Delta T_{hysteresis}$) below which the device activates the system, whereas the second is the desired temperature above which the heating system is deactivated.

Cooling mode: When the measured temperature is higher than the value of the difference ($T_{set} + \Delta T_{hysteresis}$), whereby $\Delta T_{hysteresis}$ identifies the differential adjustment of the cooler, the device activates the air conditioning system by sending a message or KNX telegram to the actuator that handles it; when the measured temperature falls below the desired temperature T_{set} the device turns off the air conditioning system by sending another message. In this way, there are two decision thresholds for activation and deactivation of the cooling: the first being the level ($T_{set} + \Delta T_{hysteresis}$) above which the device activates the system, whereas the second is the desired temperature below which the air conditioning system is deactivated. In the ETS application program, two different parameters are available for the hysteresis value for both heating and cooling: the values usually differ depending on the system type and its inertia.

In those applications where floor or ceiling radiant panels are present, it is possible to realize a different 2-point room temperature control. This type of control must be paired either to a proper regulation system for flow temperature that takes into account all internal conditions or an optimizer that exploits the thermal capacity of the building to adjust the energy contributions. In this type of control the hysteresis ($\Delta T_{hysteresis}$) o the room

temperature high limit ($T_{set} + \Delta T_{hysteresis}$) represent the maximum level of deviation that the user is willing to accept during plant conduction.
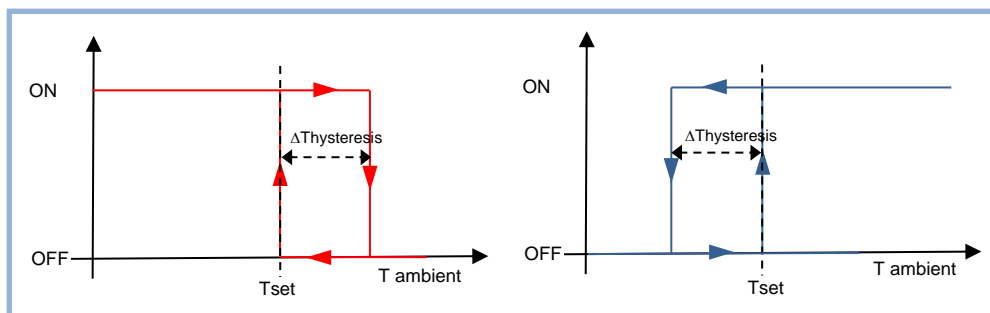
Figure 11 - Two-points temperature regulation

Heating mode – When the measured temperature is lower than the desired temperature T_{set}, the device activates the heating system by sending a message or KNX telegram to the actuator that handles it; when the measured temperature reaches the value ($T_{set} + \Delta T_{hysteresis}$), whereby $\Delta T_{hysteresis}$ identifies the differential adjustment of the boilers the device disables the heating system by sending another message. In this way, there are two decision thresholds for activation and deactivation of the heating, the first being the desired temperature T_{set} below which the device activates the system, whereas the second is the value ($T_{set} + \Delta T_{hysteresis}$), above which the heating system is deactivated.

Cooling mode – When the measured temperature is higher than the desired temperature T_{set}, the device activates the air conditioning system by sending a message or KNX telegram to the actuator that handles it; when the measured temperature reaches the value ($T_{set} - \Delta T_{hysteresis}$), whereby $\Delta T_{hysteresis}$ identifies the differential adjustment of the air conditioning system, the device disables the air conditioning system by sending another message. In this way, there are two decision thresholds for activation and deactivation of the air conditioning system: he first being the desired temperature T_{set} above which the device activates the system, whereas the second is the value ($T_{set} - \Delta T_{hysteresis}$) below which the air conditioning system is deactivated.

In the ETS application program, two different parameters are available for the hysteresis value for both heating and cooling: the values usually differ depending on the system type and its inertia.

In the ETS application program, the default 2-point hysteresis control algorithm foresees inferior hysteresis for heating and superior for cooling. If Heating and/or cooling type = floor radiant panels or ceiling radiant panels, it is possible to select the hysteresis position according to the described second mode, i.e. with superior hysteresis for heating and inferior for cooling.

The desired temperature (T_{set}) is generally different for each one of the 4 operating modes and for heating/cooling modes. The different values are defined for the first time during ETS configuration and can be modified later on. In order to optimize energy saving (for each extra degree of room temperature, outbound dispersions and energy consumption go up 6%), it is possible to take advantage of the multi-functionality of the domotic system, for example with:
- Hour programming with automatic commutation of the operating mode by means of KNX supervisor;
- Automatic commutation of the operating mode according to presence of people in the room;
- Automatic commutation of the operating mode according to window opening for air refreshment;
- Circuit deactivation when desired temperature is reached;
- Flow temperature reduction in case of partial load.

Continuous Proportional-Integral control

The continuous proportional-integral (PI) controller is described by the following equation:

$$control\ variable(t) = Kp \times error(t) + Ki \times \int_0^t error(\tau)d\tau$$

whereby:

$error(t) = (Setpoint - Measured\ temperature)\ in\ heating$

$error(t) = (Measured\ temperature - Setpoint)\ in\ cooling$

$Kp = proportional\ constant$

$Ki = integral\ constant$

The control variable is composed by 2 numbers, one depending proportionally from the error and one depending from the integral of the error itself.

Practically, some more intuitive values are used:

$$Proportional\ Band\ BP\ [K] = \frac{100}{Kp}$$
$$Integral\ Time\ Ti\ [min] = \frac{Kp}{Ki}$$

The Proportional Band is the error value that determines the maximum span of the control variable at 100%.
For example, a controller with Proportional Band = 5 K regulates at 100% when Setpoint = 20°C and Measured Temperature is ≤ 15 °C in heating mode; in cooling mode, it regulates at 100% when Setpoint = 24°C and Measured Temperature is ≥ 29°C. As shown in figure, a controller with a narrow Proportional Band provides higher control variable values for smaller errors compared to a controller with a wider Proportional Band.
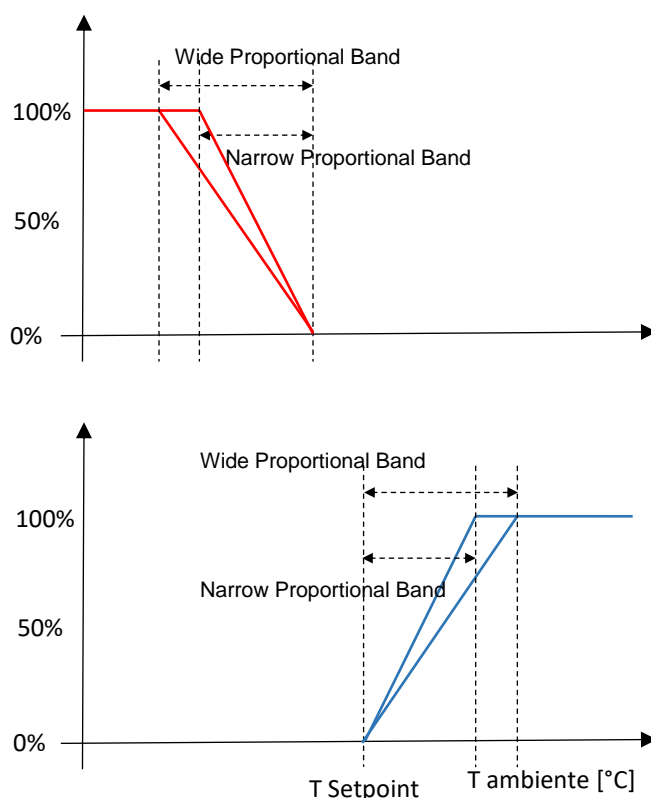
> **Integral Time is the amount of time necessary to repeat the value of the control variable of a purely proportional controller, when error is constant.** For example, with a purely proportional controller with Proportional Band = 4 K, if Setpoint = 20°C and Measured Temperature = 18°C, the control variable will be 50%. If Integral Time = 60 minutes, if error remains constant, the control variable will be 100% after 1 hour, i.e. the controller will add to the control variable a contribution equal to the value due to its proportional part.
>
> In heating and air conditioning systems, a purely proportional controller cannot guarantee reaching the Setpoint. An integral action is mandatory in order to reach the Setpoint: for this reason, the integral action is also called automatic reset.

8.3 PWM Proportional-Integral control

The proportional-integral PWM (Pulse Width Modulator) controller uses an analogue control variable to modulate the duration of the time intervals in which a binary output is in the On or Off state. The controller operates in a periodic manner over a cycle, and in each period it maintains the output to the On value for a time proportional to the value of the control variable. As shown in the figure, by varying the ratio between the ON time and the OFF time, the average time of activation of the output varies, and consequently the average intake of heating or cooling power supplied to the environment.
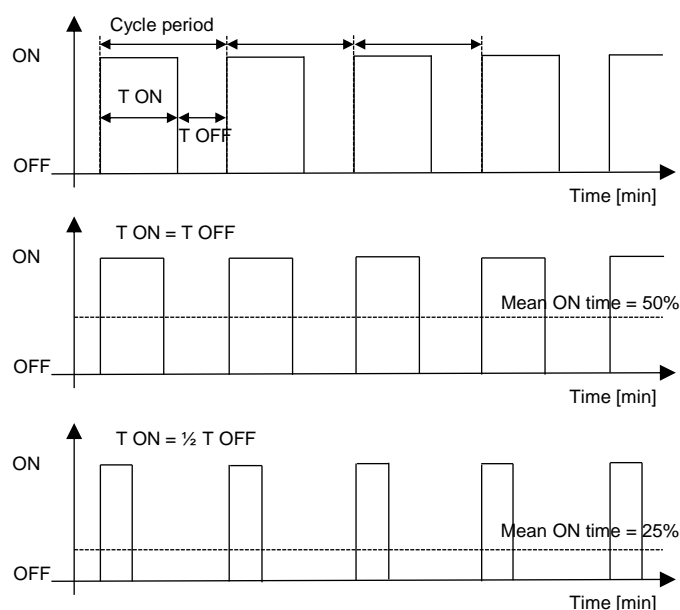
Figure 12 - PWM regulation scheme

This type of controller is well suited for use with On / Off type actuators, such as relays and actuators for zone valves, which are less expensive (both for electrical and mechanical components) than proportional actuators. A distinctive advantage of this type of controller, compared with the raw On / Off controller already described, is that it eliminates the inertia characteristics of the system: it allows significant energy savings, because you avoid unnecessary interventions on the system introduced by the 2-point control with hysteresis and it only provides the power required to compensate for losses in the building.

Every time the user or the supervisor changes the desired temperature setpoint, the cycle time is interrupted, the control output is reprocessed and the PWM restarts with a new cycle: this allows the system to reach its steady state more quickly.

Terminal type	Proportional Band [K]	Integral Time [min]	Cycle Period [min]
Radiators	5	150	15-20
Electrical heaters	4	100	15-20
Fan-coil	4	90	15-20
Floor radiant panels	5	240	15-20
Ceiling radiant panels	5	100	15-20

Guidelines for choosing the proper parameters of a PMW Proportional-Integral controller:

- **Cycle time**: for low-inertial systems such as heating and air conditioning systems, short cycle times must be chosen (10-15 minutes) to avoid oscillations of the room temperature.
- **Narrow proportional band**: wide and continuous oscillations of the room temperature, short setpoint settling time.
- **Wide proportional band**: small or no oscillations of the room temperature, long setpoint settling time.
- **Short integral time**: short setpoint settling time, continuous oscillations of the room temperature.
- **Long integral time**: long setpoint settling time, no oscillations of the room temperature.

8.4 Setpoint management

The pushbutton is not equipped with any local interface to control the integrated room thermostat, therefore the temperature setpoint modifications need to be managed through communication objects coming from a supervisory device.

Three setpoint management modes are foreseen:

- Single setpoint
- Relative setpoints;
- Absolute setpoints.

Single setpoint mode

In this mode, a unique communication object is exposed (*Input Setpoint*) to modify the desired temperature. This object can be updated cyclically or on event of change by the supervisory device. If power goes down, the last value is retained into the pushbutton's non-volatile memory. In case the object is not updated, the temperature controller acts anyway on default setpoints (both heating and cooling) set in the application program during commissioning.

> **i** If a temperature controller is set on both heating and cooling mode, it is necessary that the supervisory device also updates the input seasonal mode object (*Heating/cooling status in*, [1.100] DPT_Heat_Cool) in order to coherently switch over the controller's action.

If window contacts for energy saving are used, when detecting an open window the input setpoint freezes and the pre-set building protection setpoint is activated (the relative communication object is exposed and is different in heating or cooling mode).

Relative setpoints

In this mode, 4 communication objects are exposed, one for each operating mode:

- Comfort setpoint
- Stand-by offset
- Economy offset
- Building protection setpoint

Stand-by and economy setpoints are represented as attenuations to the comfort setpoint in order to facilitate the supervisor management: by uniquely modifying the comfort setpoint, references for attenuated modes are automatically transferred. The values modified from bus are retained in the pushbutton's non-volatile memory.

With this mode, the supervisory device can develop an hour-based time scheduling by sending to the pushbutton the current operating mode (comm. obj. *HVAC mode in* [20.102] DPT_HVACMode). The default value for *HVAC mode in* corresponds to the comfort setpoint value.

Same as single setpoint management, if the temperature controller is set as both heating and cooling mode, it is necessary that the supervisory device also updates the input seasonal mode object (*Heating/cooling status in*, [1.100] DPT_Heat_Cool) in order to coherently switch over the controller's action.

Absolute setpoints

In this mode, 3 communication objects are exposed, for all operating modes:

- Comfort heating setpoint
- Building protection heating setpoint
- Building protection cooling setpoint

All setpoint are absolute values: by modifying those values from bus through communication objects you need to keep the coherence among the values of the attentuated operating modes.

With this mode, the supervisory device can develop an hour-based time scheduling by sending to the device the current operating mode (comm. obj. *HVAC mode in* [20.102] DPT_HVAC Mode). The default value for *HVAC mode in* corresponds to the comfort setpoint value.

Same as single setpoint management, if the temperature controller is set as both heating and cooling mode with switch over from bus, it is necessary that the supervisory device also updates the input seasonal mode object (*Heating/cooling status in*, [1.100] DPT_Heat_Cool) in order to coherently switch over the controller's action.

8.5 Operating modes

In Single Setpoint mode, 2 levels for each operating mode are available:

- Temperature setpoint
- Building protection setpoint

Time scheduling for attenuation can be realized by the supervisor, by directly modifying the temperature setpoint.

In Relative Setpoint mode, 4 different operating modes are available, which are mutually exclusive to one another:

- comfort;
- stand-by;
- economy;
- building protection.

Through ETS application program, it is possible to assign 2 different setpoint values to each operating mode, for comfort and building protection level, and two different attenuation levels for stand-by and economy, corresponding to both heating and cooling. Stand-by and economy setpoints are represented as attenuations to the comfort setpoint in order to facilitate the supervisor management: by uniquely modifying the comfort setpoint, references for attenuated modes are automatically transferred.

Each setpoint, except when automatic heating/cooling switch over is active, is exposed through communication objects. Setpoints and attenuations can be modified remotely through the exposed communication objects. The building protection setpoint intervention must be planned in ETS application program, as these parameters concern the safety and protection of the plant's components (especially during heating).

8.6 Heating/cooling switch over

The switch over between both heating and cooling mode can take place in 2 ways:

1. from KNX bus, through a communication object;
2. automatically, through a command from the internal logic of the device;

Switchover from bus

In mode 1, the switch over command is issued through KNX bus and therefore it is performed by a different KNX device, e.g. the ekinex® Touch&See unit. The integrated temperature controller acts as a "slave": the switch over is carried out by input communication object [DPT 1.100 heat/cool].

Automatic switch over, based on the room temperature

Mode 2 is suitable for applications with heating / cooling systems with a 4-pipe configuration (e.g. fan-coils or radiant ceiling panels). Also in this case the information can be transmitted on the bus through an output communication object [DPT 1.100 heat/cool]; the difference with mode 1 is that the switch over is performed automatically by the machine, basing on the values of current temperature and setpoint. The automatic switch over is achieved by introducing a dead band as shown in the following figure.
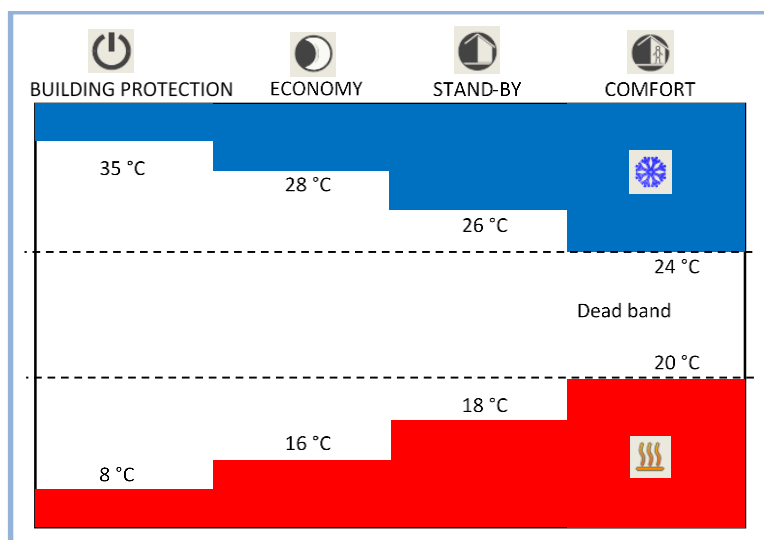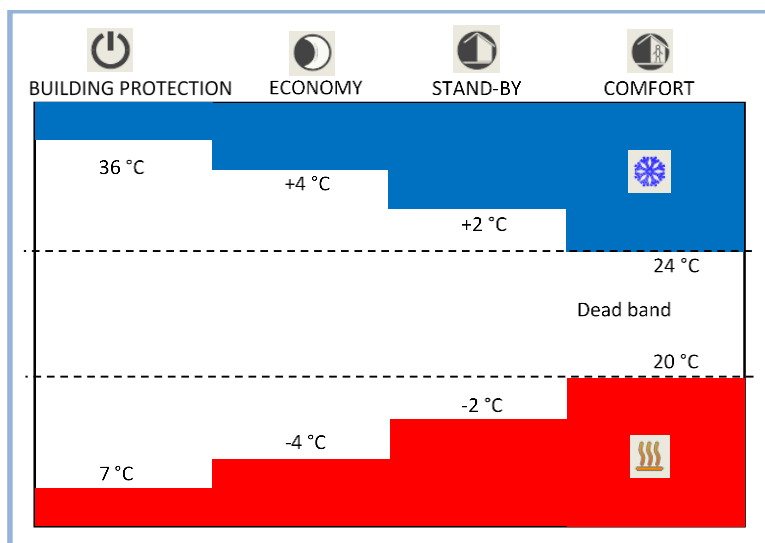
Figure 13 - Automatic switch-over, based on room temperature

The figure shows that, as long as the actual measured temperature is below the heating mode setpoint, the heating mode is selected; similarly, if the value is greater than the cooling setpoint, then cooling mode is selected. If the value is within the dead band, the operation mode remains unchanged until the value itself passes over the threshold value associated with the opposite mode.

> **i** The 4 setpoints for heating mode and the 4 setpoints for cooling mode are not exposed through communication objects to avoid inconsistencies between the different levels of temperature. In this case, a single communication object is published, which corresponds to the comfort heating setpoint. Every time this parameter is changed, the whole dead band changes with it, as well as all setpoints related to the 4 operating modes: the automatic switch over is then triggered outside the defined dead band.

8.6.1 Temperature control alarm

The integrated temperature controller can stop the internal control algorithm for one of the following reasons:

- For an external event, which can be configured and linked to the *Thermal generator lock* communication object;
- For an internal temperature sensor's fault (measured room temperature too low while NTC resistance value is too high or vice versa);
- For a timeout (data not updated by the bus) when a weighted mean between the internal sensor's value and an auxiliary external sensor's value is used.

When one of these events occur, the internal controller stops the control algorithm and the command output is taken to complete closing position (OFF or 0%): this state is signalled through the communication object *Temperature control alarm*.

9 Input from bus

9.1 Characteristics and timeout

When using the device with integrated temperature controller, variables acquired from the bus are available, which are different for each channel. All bus inputs allow to extend the device's functionality.

9.2 Room temperature (input) and weighted temperature (external object)

The temperature controller allows the acquisition of the room temperature from an external temperature probe connected to the input of the device configured as NTC.

In order to optimize or correct the room temperature regulation in particular cases (in large rooms, in the presence of strong asymmetry of the temperature distribution, when the thermostat is installed in an unsuitable position, etc.), the software can use the value acquired through the bus from another KNX device (e.g., from an ekinex button), performing a weighted average between two temperature values. The weights are assigned by the *relative weight* parameter which assigns a proportion to the two values.

9.3 Surface temperature limitation function (external object)

The floor heating system (warm water version) provides plastic pipes embedded in the concrete layer or placed directly under the final coating of the floor (light or "dry" system) filled by heated water. The water releases heat to the final coating that heats the room by radiation. The standard EN 1264 Floor heating (Part 3: Systems and components - Dimensioning) prescribes a maximum allowed temperature (T_{Smax}) for the surface of the floor that is physiologically correct defined as:

- $T_{Smax} \leq 29°C$ for zones of normal occupancy;
- $T_{Smax} \leq 35°C$ for peripheral zones of the rooms.

National standards may also limit these temperatures at lower values. Peripheral zones are strips generally located along the external walls with a maximum width of 1 m.

The floor heating system (electrically powered version) involves the laying under the floor coating of an electric cable powered by the mains voltage (230 V) or low voltage (for example 12 or 45 V), possibly already prepared in the form of rolls with constant distance between sections of cable. The powered cable releases heat to the overlying coating that heats the room by radiation. The regulation is based on measurement of the temperature of the air mass, but generally requires the monitoring and limiting of the surface temperature by using a NTC type sensor which is in contact with the floor surface.

The surface temperature limitation may be realized for several purposes:

- physiological compatibility (correct temperature at the height of the legs);
- when the system is used as auxiliary stage for heating. In this case, the heat losses to the exterior of the building are handled by the main heating stage, while the auxiliary stage only works to keep the floor temperature at a comfortable level (for example in bathrooms of residential buildings, sports centers , spas and thermal baths, etc.);
- protection against damages of the final coating due to an accidental overheating.

Note that the warm water radiant panels are usually already equipped with a safety thermostat (with intervention on the hydraulic mixing group), while in the case of electrical power this device is not usable and

it is common practice to realize a temperature limitation with a surface temperature sensor connected to the device.

The surface temperature limitation function closes the intercept valves on the distribution manifold when the temperature measured on the panel rises above set threshold (default value 29°C). The regular operation of the room thermostat resumes when the measured surface temperature drops below the hysteresis threshold (29°C - 0,3 K). For related alarms please refer to the Appendix.

9.4 Anticondensation protection function (external object)

The objective of this function is to prevent the condensation on the thermal exchange surfaces of the installation or building when cooling is working. This function is mainly used in systems with thermal exchange consisting in surface terminals such as for the floor and ceiling cooling radiant systems. In this case the hydraulic circuits contain refrigerated water; usually the latent loads (due to the increase of air humidity in the room) are handled by air-conditioning units and the temperature and humidity conditions are far from those that could cause condensation. If this is not done in a satisfactory manner, or in case of stop of the airconditioning units, it is necessary to provide additional safety measures to prevent or restrict the accidental formation of condensation on cold surfaces.

The alarm contact must be connected to an input channel of another KNX device, for example a pushbutton interface or a binary input. In this case the signal coming from the probe is sent to the channel of the device via bus, through a communication object.

In case of anticondensation alarm, if the temperature controller is in cooling mode and is demanding for fluid, the intercept valve is closed. It automatically comes back to normal mode as soon as the sensor returns to normal operation. For related alarms please refer to the Appendix.

9.5 Window contacts (external object)

In order to realize energy-saving functions, window contacts (to detect the opening of windows or doors) can be used. The device can acquire the status of a contact by means of a digital input or receive the status of two contacts connected to different KNX devices (binary inputs, pushbutton interfaces). When a window opens, the device automatically switches to *Building Protection* operating mode; when it closes, the device automatically returns to the previous operating mode. When acquiring two signals, they can be combined in logical OR.

The window contact management is an optional feature, oriented to energy saving, which is available only when the device is configured as integrated temperature controller. When an open window is detected, the operating mode is forced into building protection and remains forced until all windows are closed. The application program features a time parameter for opening delay to discriminate between an occasional, short opening and a long opening, which justifies the energy saving mode recall.

The window contact management has absolute priority over the operating mode forced by time scheduling, over the mode forced by presence sensors (if enabled) and over the HVAC mode forced by supervisor through the communication object *HVAC Forced mode in* DPT 20.102.

9.6 Presence sensors (external objects)

Presence sensors management includes a set of optional features, oriented to energy saving, which become available when the device is configured as integrated controller.

Generally speaking, if a human presence is detected and limited to the occupancy period, the comfort operating mode can be extended; vice versa, if no presence is detected, the comfort operating mode can be limited, because no longer necessary.

The occupancy status detection is performed by presence sensors which can be connected to KNX devices equipped with binary inputs; the device exposes two 1-bit communication objects for each one of the 4 channels; these objects are then synchronized to the situations detected by the sensors.

In order to determine which physical state corresponds to the presenc state, two different options can be selected:

- Not inverted (normally closed): an open contact corresponds to non-occupancy state, a close contact corresponds to detected presence;

- Inverted (normally open): an open contact corresponds to detected presence, a close contact corresponds to non-occupancy state;

There are three presence state management modes: comfort extension, comfort limitation and a combination of these two modes.

Comfort extension. This function is only active if the actual operating mode is set on comfort; if, during this time, a presence is detected, the operating mode remains comfort even if the operating mode forced by the time scheduling function shifts to economy or standby. If a presence is not detected for a timeframe less than a preset time, the operating mode does not change; vice versa, if a presence is not detected for a time period greater than the same preset time, the operating mode becomes the one forced by the time scheduling function.
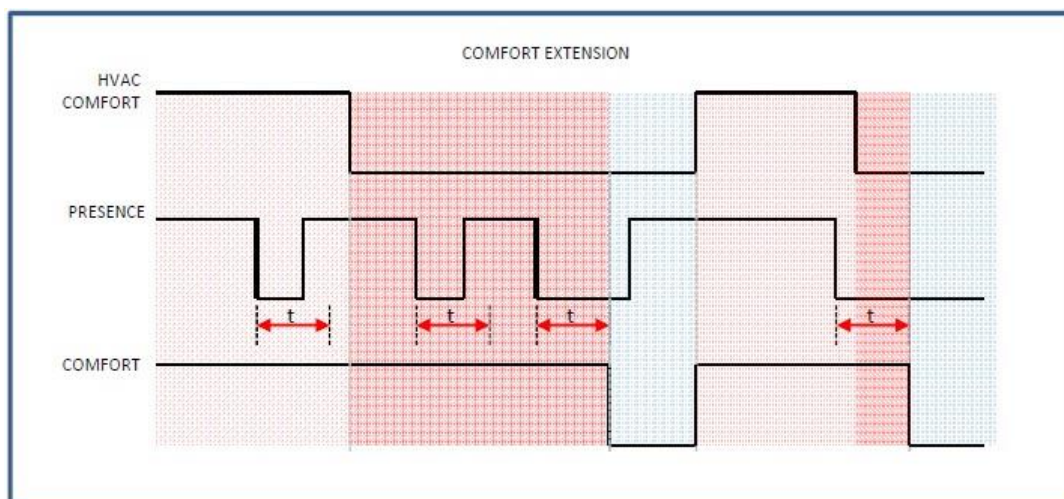
Figure 14 - Comfort extension

The figure above shows that, even if a presence is detected while the operating mode forced by the time scheduling function is not comfort, no change of operating mode is performed until the next programmed comfort event.

In case a forced HVAC mode is used by a supervisor through the communication object *HVAC forced mode in* DPT 20.102, the forced operating mode has a higher priority compared to the mode foreseen by the presence management, so it will prevail.

In case the energy saving management is carried out through window contacts, the latter has a higher priority compared tor both the forced mode and the mode foreseen by the presence management; whatever operating mode is forced by the time scheduling function, by presence management or by forced mode, the system switches to building protection mode when detecting an open window.

Comfort limitation. This function is only active if the actual operating mode is set on comfort; if, during this time, a presence is not detected for a time period greater than a preset time, the operating mode shifts to economy or standby. The attenuation modes can be selected in the application program and are independent from the modes foreseen by the time scheduling function.
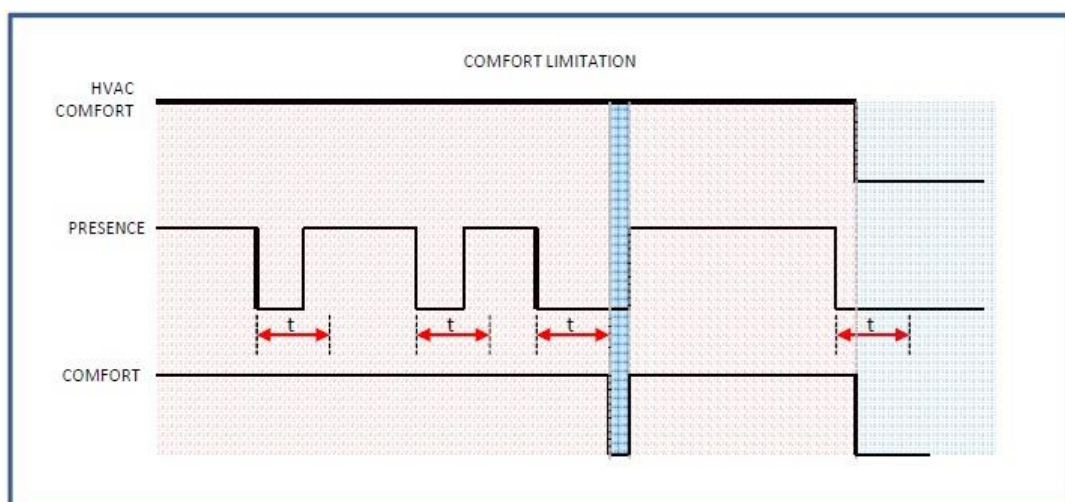
Figure 15 - Comfort limitation

Same as comfort extension, in case a forced HVAC mode is used by a supervisor through the communication object *HVAC forced mode in* DPT 20.102, the forced operating mode has a higher priority compared to the mode foreseen by the presence management, so it will prevail.

In case the energy saving management is carried out through window contacts, the latter has a higher priority compared tor both the forced mode and the mode foreseen by the presence management; whatever operating mode is forced by the time scheduling function, by presence management or by forced mode, the system switches to building protection mode when detecting an open window.

Comfort extension and comfort limitation. This mode is a combination of comfort extension and comfort limitation modes.

9.7 Logic functions

The KNX pushbutton allows to use some useful logic functions (AND, OR, NOT and exclusive OR) in order to implement complex functions in the building automation system.

You can configure:

- 8 logical functions
- 4 inputs for each channel

Each object value, if desired, can be individually inverted by inserting a NOT logic operator.

The inputs created by the objects are then logically combined as shown in the following figure:
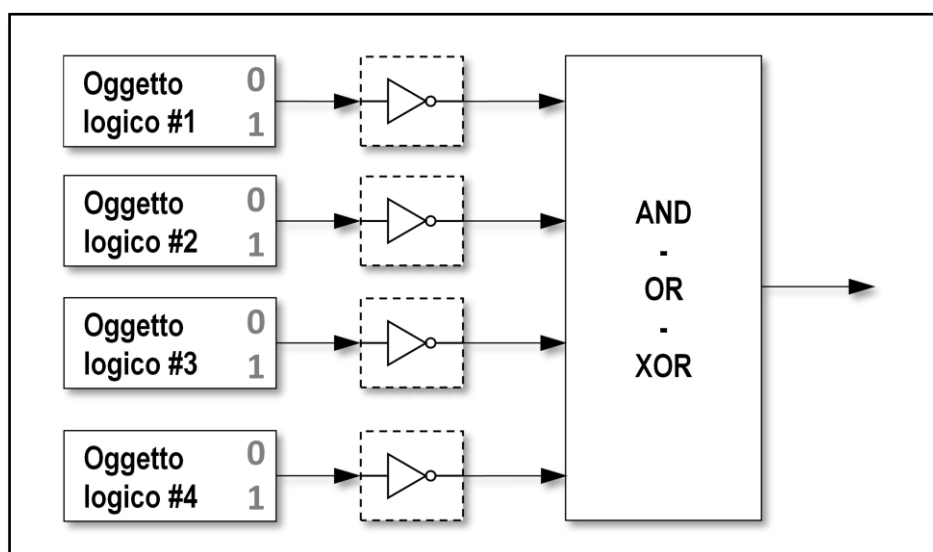
Figure 16 - Logic combination function

The logic block on the right side of the figure has the following function, based on the selected operation:

- OR – the output is ON if at least one input is ON;
- AND – the output is ON if all inputs are ON;
- XOR – the output is ON if an odd number of inputs is ON;

This last function is more intuitive when there are only 2 inputs: in this case, the output is ON when one input or the other one is ON, but not the two of them simultaneously.

Please note that in this description, with "input" and "output" we refer only to the logic block; for the device operation, the effective "inputs" are given by communication objects, so also the possible activation of NOT logic operators has to be considered.

The following figures show the basic logic functions, assuming 2 inputs and only one logic communication object:
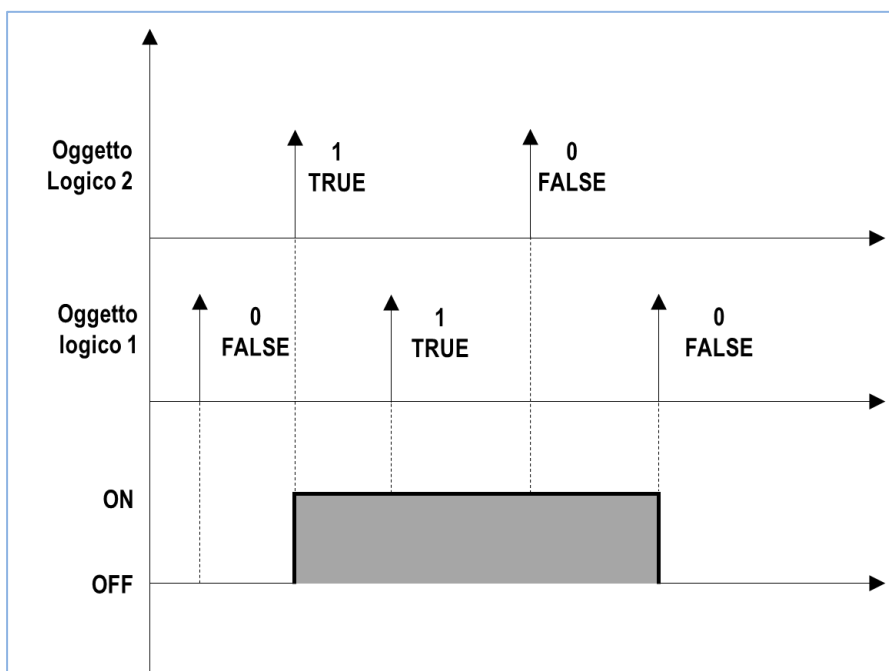
Figure 17 - Logic function OR
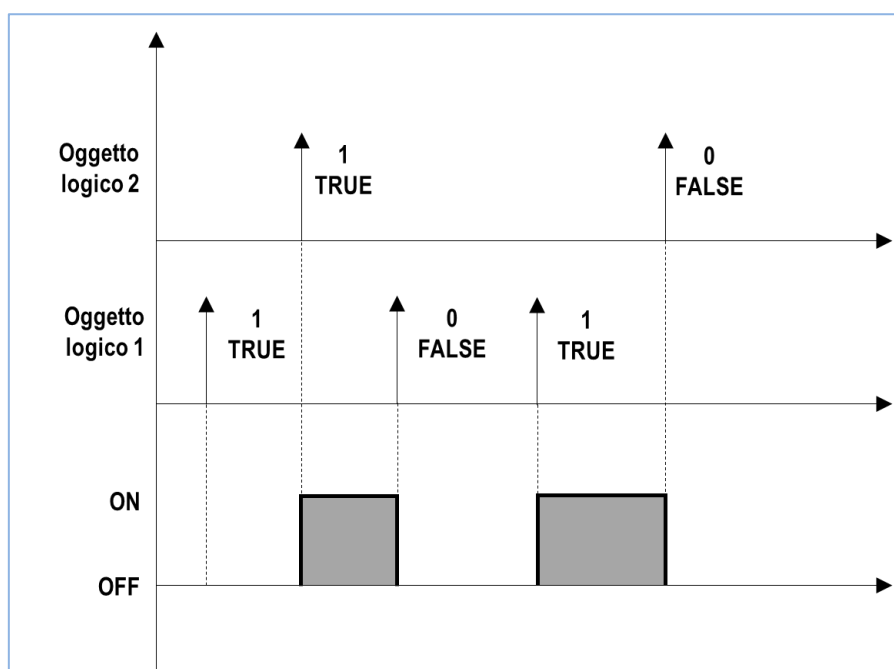
Figure 18 - Logic function AND
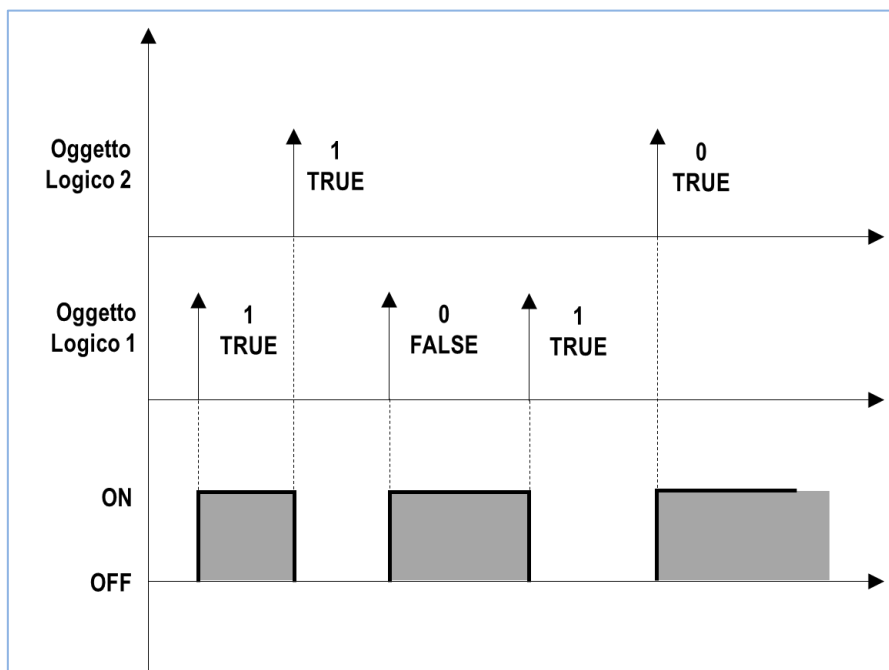
Figure 19 - Logic function XOR

For each channel, a parameter *Delay after bus voltage recovery* is available: this parameter represents the time interval between the bus voltage recovery and the first reading of the input communication objects for evaluating the logic functions.

The communication function representing the logic function output is sent on the bus on event of change; alternatively, a cyclic sending can be set.

10 Application program for ETS

In the following chapters, there is the list of folder, parameters and communication objects of the application program.

All buttons have the same set of communication objects and parameters, but they may all be independently configured.

Hereafter, all channel-specific settings are listed grouped by button; a generic button number is referenced as "x" (where x = 1…4 for the 4-buttons configuration and x = 1,…,8 for the 8-buttons configuration).

---

i	The parameter values highlighted in bold represent the default value.

The device settings are divided in two main groups: the *general* settings and the *tab-specific* settings. The settings are grouped in folders. The following figure shows the tree structure of the application program, with the main folders:

```
├── About EK-E20-TP
├── General
├── Internal sensors
│       ├── Temperature sensor
│       └── Proximity sensor
├── RGB LEDs configuration
│       ├── LED 1 left / right
│       ├── ...
│       └── LED 4 left / right
├── Buttons configuration
├── Button 1 left / right
│       ├── Settings
│       ├── Object 1
│       ├── ...
│       └── Object 8
├── ...
├── Button 4 left / right
├── Thermostat
│       ├── External sensors (from bus)
│       ├── Weighted temperature value
│       ├── Temperature control
│       ├── Relative humidity control
│       └── Energy saving
└── Logic functions
```

Figure 20 - Application program parameters

In order to use the device as a temperature sensor or as a room temperature controller it is enough to enable the temperature sensor in the *Internal sensors* folder. Consequently, also the *Temperature control* folder is activated: therefore, it is possible to select an auxiliary temperature sensor to perform a weighted mean with the main sensor and it is possible to configure the controller's options for room temperature.

10.1 About EK-E20-TP

The folder ***About EK-E20-TP*** is for information purposes only and does not contain parameters to be set. The information given is:

© Copyright Ekinex S.p.A. 2020
Application software for ETS4 and ETS5
Version 1.0 (or later)
EK-E20-TP – 4/8-fold KNX 20venti series pushbutton

Ekinex S.p.A.
Via Novara 37
I-28010 Vaprio d'Agogna (NO) Italy
www.ekinex.com
info@ekinex.com

10.2 General settings

The *General* settings tab shows the following parameters:

- Buttons layout;
- LEDS intensity from bus;
- LEDs intensity;
- Backlight duration;
- Technical alarm;
- Thermostat functionalities.

This Tab has no secondary sections.

The parameters in this section define the overall behaviour of the device, including the setting that defines which and how many channels are available.

10.2.1.1 Parameters and communication objects

Parameter name	Conditions	Values
Buttons layout		**4 left buttons** 4 right buttons 8 buttons
	Depending on the selected value, either 4 or 8 buttons/LEDs will be displayed.	
LEDs intensity from bus		**No** / yes
LEDs intensity	LEDs intensity from bus = No	**50** [range 0 … 100]
Delay after bus voltage recovery	-	**00:00:04.000 hh:mm:ss:fff** [range 00:00:04.000 … 00:10:55.350]
	Delay before bus telegrams can be sent after a recovery of the bus voltage. The delay time affects the transmission generated by an event as well as the cyclical transmission. For the cyclical transmission: after the delay time finished, the cycle restarts and the first telegram will be sent after the cycle time.	
Technical alarm	-	enabled / **disabled**
	Enables a communication objects that activates an alarm indication through a bus telegram. The indication is made by flashing the four LEDs at the corners of the device. *This indication is made available to the user for any purpose he sees fit (not necessarily an actual alarm).*	
Thermostat functionalities	-	enabled / **disabled**

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Technical alarm	Technical alarm = enabled	1 bit	C–W--	[1.005] alarm	1
Max LEDs intensity percentage	LEDs intensity from bus = Yes	1 byte	C–W--	[5.001] percentage (0…100%)	2

10.3   Internal sensors

The **Internal sensor** Tab contains the following items:

- Temperature sensor;
- Proximity sensor.

10.3.1.1 Parameters and communication objects

Parameter name	Conditions	Values
Temperature sensor		**enabled** disabled
	Enables the temperature sensor by making the corresponding communication object available.	

Parameter name	Conditions	Values
Proximity sensor		disabled **enabled** from bus

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Temperature value	Temperature sensor = enabled	2 Bytes	CR–T-	[9.001] temperature (°C)	196
Proximity sensor	Proximity sensor = enabled	1 bit	C-WTU	[1.017] trigger	3

10.3.2 Temperature sensor

The ***Temperature sensor*** Tab contains the following items:

- Filter type
- Temperature offset
- Minimum change of value to send [K]
- Cyclic sending interval
- Threshold 1
- Threshold 2

10.3.2.1 *Parameters and communication objects*

Parameter name	Conditions	Values
Filter type	Temperature sensor = enabled	low **average** high
	Low = average value every 4 measurements *Average = average value every 16 measurements* *High = average value every 64 measurements*	
Measured temperature correction	Temperature sensor = enabled	**0°C** [range -2,5°C … +2,5°C]
	Offset steps are 0.1°C in order to perform a more effective sensor calibration.	
Minimum change of value to send [K]	Temperature sensor = enabled	**0,5** [range 0 … 5]
	If the parameter is set to 0, no value is sent for change	
Cyclic sending interval	Temperature sensor = enabled	**no sending** [other values in range 30 s … 120 min]
Threshold 1	Temperature sensor = enabled	**not active** below above

Parameter name	Conditions	Values
Value [°C]	Temperature sensor = enabled, Threshold 1 = above or below	**7** [range 0 … 50]
Threshold 2	Temperature sensor = enabled	**not active** below above
Value [°C]	Temperature sensor = enabled, Threshold 2 = above or below	**45** [range 0 … 50]
Threshold value update from bus	Temperature sensor = enabled, Threshold 1 or 2 = above or below	**no** yes
Threshold lock enable	Temperature sensor = enabled, Threshold 1 or 2 = above or below	**no** yes
Behaviour at lock	Temperature sensor = enabled, Threshold 1 or 2 = above or below Threshold lock enable = yes	**none** off on
Behaviour at bus recovery	Temperature sensor = enabled, Threshold 1 or 2 = above or below Threshold lock enable = yes	**previous state** lock unlock
Hysteresis	Temperature sensor = enabled, Threshold 1 e/o Threshold 2 = above or below	**0,4 K** [other values in range 0,2 K ... 3 K]
Cyclic sending interval	Temperature sensor = enabled, Threshold 1 e/o Threshold 2 = above or below	**no sending** [other values in range 30 s … 120 min]

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Temperature threshold 1 - Switch	Temperature sensor = enabled, Threshold 1 = above or below	1 Bit	CR-T--	[1.001] switch	197
Temperature threshold 1 - Lock	Temperature sensor = enabled, Threshold 1 = above or below Threshold lock enable = yes	1 bit	C-W--	[1.001] switch	198
Temperature threshold 1 – Value (from bus)	Temperature sensor = enabled, Threshold 1 = above or below Threshold update value from bus = yes	2 bytes	C-W--	[9.001] temperature (°C)	199

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Temperature threshold 2 - Switch	Temperature sensor = enabled, Threshold 2 = above or below	1 Bit	CR-T--	[1.001] switch	200
Temperature threshold 2 - Lock	Temperature sensor = enabled, Threshold 2 = above or below Threshold lock enable = yes	1 bit	C-W--	[1.001] switch	201
Temperature threshold 2 – Value (from bus)	Temperature sensor = enabled, Threshold 2 = above or below Threshold update value from bus = yes	2 bytes	C-W--	[9.001] temperature (°C)	202

Acquisition filter

The acquisition filter calculates an average between a series of acquired values of the quantity measured before being sent on the bus. The parameter can take the values:

• low (average value calculated every 4 measurements);

• average (average value calculated every 16 measurements);

• high (average value calculated every 64 measurements).

Measured temperature correction

The sampling (indicative) of the temperature value takes place every 10 seconds; during configuration with ETS, the possibility of correcting the measured temperature value within the offset range - 5 ° C ... + 5 ° C (step 0.1 K) is left.

10.3.3 Proximity sensor

The Proximity Sensor tab contains the following parameters:

- Sensitivity
- Maximum standby intensity
- Send values or sequences
- Lock function
- Number of communication objects

10.3.3.1 Parameters and communication objects

Parameter name	Conditions	Values
Sensitivity	Proximity sensor = enabled	Low **Medium** high
	Select the sensor sensitivity Low = mean value every 4 measurements Medium = mean value every 16 measurements High = mean value every 64 measurements	
Maximum standby intensity	Proximity sensor = enabled or from bus	**80 %** [range 0 % … 100%]
	Sets the maximum backlit level when in stand-by state	

Parameter name	Conditions	Values
Backlight duration	Proximity sensor = enabled or from bus	**00:00:00 hh:mm:ss** [range 00:00:00 … 18:12:15
	Duration of the backlight function	
Send values or sequences	Proximity sensor = enabled or from bus	enabled **disabled**
	If the parameter is set to 0, no value is sent in case of change.	
Number of communication objects	Proximity sensor = enabled or from bus Send values or sequences = enabled	1…8 **1**
Object *n* – Send delay	Proximity sensor = enabled or from bus Send values or sequences = enabled Number of communication objects = *n*	hh:mm:ss.ff (**00:00:00.00**)
	Delay between the event and the transmission of the value on the bus. *By defining an individual delay for each object, it is possible to form a defined telegram sequence to be associated to the event.*	
Object *n* – Communication object dimension	Proximity sensor = enabled or from bus Send values or sequences = enabled Number of communication objects = *n*	**1 bit value** 2 bits value 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned value 2 bytes signed value 2 bytes floating value
	Defines the type and size of the any communication objects.	
Object *n* – reaction to presence detected	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension = 1 bit	none on off **toggle**
	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension = 2 bits	none disable enable off / up **enable on / down** enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension ≠ 1 bit or 2 bits	**none** send value 1 send value 2 send value 1 ↔ send value 2
	The configuration of the object for detecting proximity changes according to the type of communication object selected.	
Oggetto *n* – reaction to presence not detected	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension = 1 bit	**none** on off toggle
	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension = 2 bits	none **disable** enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension ≠ 1 bit or 2 bits	**none** send value 1 send value 2 send value 1 ↔ send value 2
	The configuration of the object for detecting proximity changes according to the type of communication object selected.	

Parameter name	Conditions	Values
Object *n* – Value 1	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension ≠ 1 bit or 2 bits	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 bytes unsigned value) -32768... 32767 (2 bytes signed value) -671088.64...670760.96 (2 bytes floating value)
	First value configured for association to events	
Object *n* – Value 2	Proximity sensor = enabled or from bus Send values or sequences = enabled Communication object dimension ≠ 1 bit or 2 bits	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 bytes unsigned value) -32768... 32767 (2 bytes signed value) -671088.64...670760.96 (2 bytes floating value)
	Second value configured for association to events	
Lock function	Proximity sensor = enabled or from bus Send values or sequences = enabled	enabled **/ disabled**
Invert lock device signal	Proximity sensor = enabled or from bus Send values or sequences = enabled Lock function = enabled	**Not inverted** / inverted
Lock after bus recovery	Proximity sensor = enabled or from bus Send values or sequences = enabled Lock function = enabled	**no** / yes
Behaviour at locking	Proximity sensor = enabled or from bus Send values or sequences = enabled Lock function = enabled	**None** / As presence detected / As presence not detected
Behaviour at unlocking	Proximity sensor = enabled or from bus Send values or sequences = enabled Lock function = enabled	**None** / As presence detected / As presence not detected

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Proximity - Switching status *[type]*, Object *n*	Proximity sensor = enabled Send values or sequences = enabled	*Depending on the configuration* (**1-bit**)	C-WTU	*Depending on the configuration* (**[1.001] switch**)	133, 134,135, 136,137, 138,139, 140
	Possono essere definiti fino a 8 oggetti da associare ad uno stesso evento.				
Proximity – lock command	Proximity sensor = enabled Send values or sequences = enabled Lock function = enabled	1 bit	C-W--	[1.003] enable	132

10.4   LEDs

The *RGB LEDs configuration* Tab contains the following parameters:

- LED n (where n = 1,…,4) left / right

10.4.1   LED 1 … LED 4 left/right

Any of the *LED 1 … LED 4 left / right* Tab shows the following parameters:

- Color ON;
- Color OFF;
- OFF delay;
- Color;
- Threshold type;
- Value n – blinking period (n = 0,…,3)
- Value n – color (n = 0,…,3)
- Number of thresholds
- Threshold n – blinking period (n = 1, …, 8)
- Threshold n - value (n = 1, …, 8)
- Threshold n - color (n = 1, …, 8)

10.4.1.1 Parameters and communication objects

Parameter name	Conditions	Values
LED *n* activation (n = 1,…,4 left / right)		fixed **on contact closed** status from bus with threshold
	It allows to select the behaviour for LED n	
LED *n* - color	LED *n* activation = fixed	**#FF0000** [range #000000, …, #FFFFFF]
	n = 1, …, 4 left / right	
LED *n* – color ON	LED *n* activation = on contact closed	**#FF0000** [range #000000, …, #FFFFFF]
	n = 1, …, 4 left / right	
LED *n* – color OFF	LED *n* activation = on contact closed	**#00D4FF** [range #000000, …, #FFFFFF]
	n = 1, …, 4 left / right	
LED *n* – OFF delay	LED *n* activation = on contact closed	**00:00:01.00** [range 00:00:00.00…01:49:13.50] hh:mm:ss.ff
	n = 1, …, 4 left / right	
LED *n* – threshold type	LED *n* activation = status from bus with threshold	**1 bit value** 2 bits value 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned value 2 bytes signed value 2 bytes floating value
	n = 1, …, 4 left / right	
LED *n* - value 0 – blinking period	LED *n* activation = status from bus with threshold LED *n* – threshold type = 1 bit or 2 bits	**None** 0,25s ON, 0,25s OFF 0,25s ON, 0,75s OFF 0,50s ON, 0,50s OFF 0,75s ON, 0,25s OFF 0,50s ON, 1,50s OFF 1s ON, 1s OFF 1,50s ON, 0,50s OFF 1s ON, 3s OFF 2s ON, 2s OFF 3s ON, 1s OFF
	n = 1, …, 4 left / right	
LED *n* - value 0 – color	LED *n* activation = status from bus with threshold LED *n* – threshold type = 1 bit or 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	n = 1, …, 4 left / right	
LED *n* - value 1 – blinking period	LED *n* activation = status from bus with threshold LED *n* – threshold type = 1 bit or 2 bits	**None** 0,25s ON, 0,25s OFF 0,25s ON, 0,75s OFF 0,50s ON, 0,50s OFF 0,75s ON, 0,25s OFF 0,50s ON, 1,50s OFF 1s ON, 1s OFF 1,50s ON, 0,50s OFF 1s ON, 3s OFF 2s ON, 2s OFF 3s ON, 1s OFF
	n = 1, …, 4 left / right	
LED *n* - value 1 – color	LED *n* activation = status from bus with threshold LED *n* – threshold type = 1 bit or 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]

	n = 1, …, 4 left / right	
LED *n* - value 2 – blinking period	LED *n* activation = status from bus with threshold LED *n* – threshold type = 2 bits	**None** 0,25s ON, 0,25s OFF 0,25s ON, 0,75s OFF 0,50s ON, 0,50s OFF 0,75s ON, 0,25s OFF 0,50s ON, 1,50s OFF 1s ON, 1s OFF 1,50s ON, 0,50s OFF 1s ON, 3s OFF 2s ON, 2s OFF 3s ON, 1s OFF
	n = 1, …, 4 left / right	
LED *n* - value 2 – color	LED *n* activation = status from bus with threshold LED *n* – threshold type = 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	n = 1, …, 4 left / right	
LED *n* - value 3 – blinking period	LED *n* activation = status from bus with threshold LED *n* – threshold type = 2 bits	**None** 0,25s ON, 0,25s OFF 0,25s ON, 0,75s OFF 0,50s ON, 0,50s OFF 0,75s ON, 0,25s OFF 0,50s ON, 1,50s OFF 1s ON, 1s OFF 1,50s ON, 0,50s OFF 1s ON, 3s OFF 2s ON, 2s OFF 3s ON, 1s OFF
	n = 1, …, 4 left / right	
LED *n* - value 3 – color	LED *n* activation = status from bus with threshold LED *n* – threshold type = 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	n = 1, …, 4 left / right	
LED *n* – number of thresholds	LED *n* activation = status from bus with threshold LED *n* – threshold type ≠ 1 bit, 2 bits	**1** (range 1…8)
	n = 1, …, 4 left / right	
LED *n* – threshold x - blinking period	LED *n* activation = status from bus with threshold LED *n* – threshold type ≠ 1 bit, 2 bits	**None** 0,25s ON, 0,25s OFF 0,25s ON, 0,75s OFF 0,50s ON, 0,50s OFF 0,75s ON, 0,25s OFF 0,50s ON, 1,50s OFF 1s ON, 1s OFF 1,50s ON, 0,50s OFF 1s ON, 3s OFF 2s ON, 2s OFF 3s ON, 1s OFF
	n = 1, …, 4 left / right *x = 1,…,8 (depending on the number of thresholds)*	
LED *n* - threshold x - value	LED *n* activation = status from bus with threshold LED *n* – threshold type ≠ 1 bit, 2 bits	**0** 1 signed byte: range -128…+127 1 unsigned byte: range 0…255 2 signed bytes: range -32768…+32767 2 unsigned bytes: range 0…65535 2 floating bytes: range -671088,64…670760,96

LED *n* - threshold x - color	*n = 1, ..., 4 left / right* *x = 1,...,8 (depending on the number of thresholds)*		
	LED *n* activation = status from bus with threshold LED *n* – threshold type ≠ 1 bit, 2 bits	**#FFFF00** [range #000000, ..., #FFFFFF]	
	n = 1, ..., 4 left / right *x = 1,...,8 (depending on the number of thresholds)*		

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
LED n – command 1 bit	LED *n* activation = status from bus with threshold LED *n* – threshold type = 1 bit	1 bit	C-WTU	[1.001] switch	148, 149, 150, 151, 152, 153, 154, 155.
	n = 1, ..., 4 left / right				
LED n – command 2 bit	LED *n* activation = status from bus with threshold LED *n* – threshold type = 2 bits	2 bits	C-WTU	[2.001] switch control	148, 149, 150, 151, 152, 153, 154, 155.
	n = 1, ..., 4 left / right				
LED n – command 1 signed byte	LED *n* activation = status from bus with threshold LED *n* – threshold type = 1 signed byte	1 byte	C-WTU	[6.010] counter pulses -128…+127	148, 149, 150, 151, 152, 153, 154, 155.
	n = 1, ..., 4 left / right				
LED n – command 1 unsigned byte	LED *n* activation = status from bus with threshold LED *n* – threshold type = 1 unsigned byte	1 byte	C-WTU	[5.010] counter pulses 0…255	148, 149, 150, 151, 152, 153, 154, 155.
	n = 1, ..., 4 left / right				
LED n – command 2 signed byte	LED *n* activation = status from bus with threshold LED *n* – threshold type = 2 signed byte	2 bytes	C-WTU	[8.001] pulses difference	148, 149, 150, 151, 152, 153, 154, 155.
	n = 1, ..., 4 left / right				

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
LED n – command 2 unsigned bytes	LED n activation = status from bus with threshold LED n – threshold type = 2 unsigned bytes	2 bytes	C-WTU	[7.001] pulses	148, 149, 150, 151, 152, 153, 154, 155.
	n = 1, …, 4 left / right				
LED n – command 2 floating bytes	LED n activation = status from bus with threshold LED n – threshold type = 2 floating bytes	2 bytes	C-WTU	[9.0*] 2-byte float value	148, 149, 150, 151, 152, 153, 154, 155.
	n = 1, …, 4 left / right				

10.5 Buttons configuration

The **Buttons configuration** tab allows you to configure each independent or single button. Contains the following parameters:

- Button 1, …, 4 left
- Button 1, …, 4 right

10.5.1 Button settings

Each button from the first to the fourth (or eighth) will be displayed only if it meets two conditions:

- the button is enabled;
- the parameter "Buttons layout" if set on 4 left buttons / 4 right buttons enables the first 4 buttons, while if set on 8 buttons it enables all 8 buttons;

Each button has its own Settings tab, which differs according to the type of function chosen (send values or sequences, dimming, shutter / venetian blind mode, scenes).

10.5.1.1 Parameters and communication objects

Parameter name	Conditions	Values
Button x - left		disabled **enabled**
	It enables the x (x = 1,…,4) buttons on the left side, according to the selected device configuration.	
Button x - right	-	disabled **enabled**
	It enables the x (x = 1,…,4) buttons on the right side, according to the selected device configuration.	

Parameter name	Conditions	Values
Type	Button x left / right = enabled	**send values or sequences** dimming shutter or venetian blind scene
	Determines the kind of function performed by the button x. *Further parameters for the selected function will appear in the individual rocker configuration sections (see below).*	

10.5.2 Send values or sequences

10.5.2.1 Parameters and communication objects

Parameter name	Conditions	Values
Number of communication objects	Button x = enabled Type = send values or sequences	1…8 (**1**)
	Number of communication objects configured in association with the button event.	
Lock function – Behaviour at locking	Button x = enabled Type = send values or sequences	**none** as close or short press as open or long press
	Allows performing the operation associated to the specified event when a locking command is received. You can choose between operations linked to two possible closing (or short press, depending on the configuration) or opening (or long press) events.	
Lock function – Behaviour at unlocking	Button x = enabled Type = send values or sequences	**none** as close or short press as open or long press
	Allows performing the operation associated to the specified event when an unlocking command is received. You can choose between operations linked to two possible closing (or short press, depending on the configuration) or opening (or long press) events.	
Event	Button x = enabled Type = send values or sequences	**activation / release** short / long press
	Type of event that should be used as trigger for an action.	
Long press time	Button x = enabled Type = send values or sequences Event = short / long press	hh:mm:ss.fff (**00:00:00.800**)
	Minimum push time for a press in order to be recognized as a long press.	
Object *n* – Send delay	Button x = enabled Type = send values or sequences	hh:mm:ss.ff (**00:00:00.00**)
	Delay before the object is transmitted on the bus. *By defining a delay after the event occurs and before the object value is sent, it is possible to associate a time defined sequence of values to an input event.*	
Object *n* – Send cyclically	Button x = enabled Type = send values or sequences Number of communication objects = 1	none **off / value 1** on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated. *The cyclical transmission is only available if the number of communication objects to link is 1.*	
Object *n* – Cyclic sending interval	Button x = enabled Type = send values or sequences Number of communication objects = 1 Send cyclically ≠ none	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	

Parameter name	*Conditions*	*Values*
Object *n* – Communication object dimension	Button x = enabled Type = send values or sequences	**1 bit value** 2 bits value 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned value 2 bytes signed value 2 bytes floating value
	Defines size and type of the values to be sent when an event occurs.	
Object *n* – Reaction to short press	Button x = enabled Type = send values or sequences c.o. dimension = 1 bit	none on off **toggle**
	Button x = enabled Type = send values or sequences c.o. dimension = 2 bit	none disable enable off / up **enable on / down** enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Button x = enabled Type = send values or sequences c.o. dimension = *any byte value*	none **send value 1** send value 2 send value 1 ↔ send value 2
	Value change behaviour caused by either a Close or a Short Press event (according to event configuration)	
Object *n* – Reaction to long press	Button x = enabled Type = send values or sequences c.o. dimension = 1 bit	**none** on off toggle
	Button x = enabled Type = send values or sequences c.o. dimension = 2 bit	none **disable** enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Button x = enabled Type = send values or sequences c.o. dimension = *any byte value*	**none** send value 1 send value 2 send value 1 ↔ send value 2
	Value change behaviour caused by either an Open or a Long Press event (according to event configuration)	
Object *n* – Value 1	Button x = enabled Type = send values or sequences c.o. dimension = *any byte value*	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 bytes unsigned value) -32768... 32767 (2 bytes signed value) -671088.64...670760.96 (2 bytes floating value)
	First value available for association in send events	
Object *n* – Value 2	Button x = enabled Type = send values or sequences c.o. dimension = *any byte value*	*same as value 1*
	Second value available for association in send events	

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Button x left / right – Switching status [type], object n	Button x = enabled Type = send values or sequences	*according to the configuration* (**1-bit**)	C-WTU	*according to the configuration* (**[1.001] switch**)	5, 21, 37, 53, 69, 85, 101, 117
	Up to 8 objects can be defined for binding with the same event.				
	The listed CO numbers are those referring to object nr.1; the COs for each subsequent object are following in sequence.				
	To obtain the CO numbers for object number n, just add (n-1) to the listed numbers.				
	E.g.: COs associated to Button 3 have numbers from 37 to 44. The number of CO no. 5 is therefore 37 + (5-1) = 41.				
	The size and type of the individual objects can be configured as described in following sections.				

Sizes and DPTs are as follows:

Size	DPT
1 bit	[1.001] switch
2 bit	[2.*] 1-bit controlled
1 byte unsigned value	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte percentage	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte signed value	[6.*] 8-bit signed value
2 bytes unsigned value	[7.*] 2-byte unsigned value
2 bytes signed value	[8.*] 2-byte signed value
2 bytes floating value	[9.*] 2-byte float value

Table 5 – Size and Data types for communication objects

Note on the range of 2-bytes floating point values

In case the data type is *2 bytes floating value*, the range of values is -273 ... 670760.96, instead of -671088.64 ... 670760.96. This because it is considered as a temperature range in °C by default, so the lower limit is absolute zero.

In order to assign negative values < -273, it is necessary to open the communication object *Properties* menu from the Group Objects tab and manually modify the DPT, choosing the most suitable one among those available for group 7. * 2-byte unsigned value (see example in Figure 21):
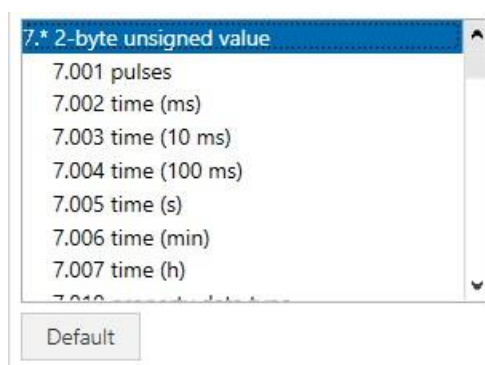
Figure 21 - DPT of 7.* 2-bytes floating point

10.5.3 Dimming

10.5.3.1 Parameters and communication objects

Parameter name	Conditions	Values
Long press time	Button x = enabled Type = dimming	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Toggle mode	Button x = enabled Type = dimming	**enabled /** disabled
	When enabled, causes the short press to toggle the on-off status of the destination CO; otherwise, a fixed status can be assigned to the short press.	
Reaction to long press	Button x = enabled Type = dimming Toggle mode = enabled	**darker** brighter darker ↔ brighter
	Defines the function to be assigned to the long press. If the toggle mode is enabled, the Short press action is already defined as toggle.	
Short / Long action	Button x = enabled Type = dimming Toggle mode = disabled	**off / darker** on / brighter off / darker ↔ brighter on / darker ↔ brighter
	Defines the function to be assigned to the long and short press.	
Send cyclically	Button x = enabled Type = dimming	**none** off / value 1 on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclic sending interval	Button x = enabled Type = dimming Send cyclically ≠ none	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at locking	Button x = enabled Type = dimming	**none** off on toggle
	Value to be assigned to the object when a locking command is received.	

Parameter name	Conditions	Values
Lock function – Behaviour at unlocking	Button x = enabled Type = dimming	**none** off on as previous
	Value to be assigned to the object when an unlocking command is received.	

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Button x left / right – Switching command	Button x = enabled Type = dimming	1 bit	C-WTU	[1.001] switch	13, 29, 45, 61, 77, 93, 109, 125
	Send a command to a dimming actuator to switch the light on or off. *The command is triggered by a short press on the input.* *The value sent can be a fixed value or it can be toggled at each input activation.*				
Button x left / right – Dimming up / down / stop command	Button x = enabled Type = dimming	4 bit	CR-T-	[3.*] 3-bit control	14, 30, 46, 62, 78, 94, 110, 126

	Send a command to a dimming actuator to change dimming intensity (brighter or darker).
	Three values are used which mean start increase, start decrease or stop the change.
	
	Increase Decrease
	`1 0 0 0` `0 0 0 1`
	
	Stop dimming
	`0 0 0 0`
	
	Increase/decrease values are sent when a long press action occurs and stop value on press release.
	The value sent can be a fixed value or it can be toggled at each input activation.				

10.5.4 Shutter or venetian blind

10.5.4.1 Parameters and communication objects

Parameter name	Conditions	Values
Long press time	Button x = enabled Type = shutter or venetian blind	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Toggle mode	Button x = enabled Type = shutter or venetian blind	**enabled /** disabled
	When enabled, causes the short press to toggle the on-off status of the destination CO; otherwise, a fixed status can be assigned to the short press.	
Up / Down action	Button x = enabled Type = shutter or venetian blind Toggle mode = disabled	**up** down
	Defines the movement direction to be assigned to the button press.	

Parameter name	Conditions	Values
Blind mode	Button x = enabled Type = shutter or venetian blind	enabled / **disabled**
	If blinds mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.	
Lock function – Behaviour at locking	Button x = enabled Type = shutter or venetian blind	**none** up down
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Rocker x = independent or single Type = shutter or venetian blind	**none** up down
	Operation to perform when an unlocking command is received.	

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Button x left / right – Dedicated stop command	Button x = enabled Type = shutter or venetian blind	1 bit	C—T-	[1.017] trigger	13, 29, 45, 61, 77, 93, 109, 125
	Immediately stop any movement of the blind. The object is sent on a short press if the blind mode is disabled and at the end of a long press if the venetian blind mode is enabled.				
Button x left / right – Stop – step up/down command	Button x = enabled Type = shutter or venetian blind Venetian blind mode = enabled	1 bit	CR-T-	[1.007] step	15, 31, 47, 63, 79, 95, 111, 127
	Move the blind to fully open or fully closed position. The object is sent at the end of a long press.				
Button x left / right – Move up / down command	Button x = enabled Type = shutter or venetian blind Venetian blind mode = enabled	1 bit	CRWTU	[1.008] up/down	16, 32, 48, 94, 80, 96, 112, 128
	Increase or decrease the opening of the blind stepwise. *The object is sent on a short press.*				

10.5.5 Scene

10.5.5.1 Parameters and communication objects

Parameter name	Conditions	Values
First scene number	Button x = enabled Type = scene	1…64 (**1**)
	Main scene number to be assigned to button press. It is named "first" for the case that an alternative scene number is used.	
Learning mode	Button x = enabled Type = scene	enabled / **disabled**

Parameter name	Conditions	Values
	When enabled, a long key press can be used to program the selected scene by storing the current parameters.	
Long press time	Button x = enabled Type = scene Learning mode = enabled	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Scene activation	Button x = enabled Type = scene Learning mode = disabled	**send first scene only** toggle between two scenes
	Allows the key to be used to alternate between two different scenes.	
Second scene number	Button x = enabled Type = scene Learning mode = disabled Scene activation = toggle between two scenes	1…64 (**2**)
	Alternate scene number to be assigned to button press.	
Lock function – Behaviour at locking	Button x = enabled Type = scene	**none** send first scene send second scene
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Button x = enabled Type = scene	**none** send first scene send second scene
	Operation to perform when an unlocking command is received.	

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Button x left / right – Scene number	Button x = enabled Type = scene	1 Byte	C--T-	[17.*] Scene number [18.*] Scene control	17, 33, 49, 65, 81, 97, 113, 129
	Store or recall a scene. The lowest 6 bits in the byte form the code of the scene, while the highest bit is the operation code (store or recall).				

10.5.6 Lock function

When the lock function is enabled, for each button the user can define an action to execute when a lock or unlock command is received.

Details are shown in the following sections; a resume of all options is shown in Table 6.

Function type	Behaviour at locking	Behaviour at unlocking
send values or sequences	**none** as close or short press as open or long press	
dimming	**none** off on toggle	**none** off on as previous
scene	**none** send first scene send second scene	
shutter or venetian blind	**none** up down	

Table 6 - Lock function

10.5.6.1 Parameters and communication objects

Parameter name	Conditions	Values
Lock function	-	enabled / **disabled**
	Enables or disables the possibility of blocking a button via a telegram on the bus	
Lock function – invert lock device signal	Button *x* = enabled Type = any	**Not inverted** / inverted
	It allows to commute a "block active" command as a "deactivate block" command and vice versa.	
Lock function – Lock after bus recovery	Button *x* = enabled Type = any	**no** / yes
	If active, when the bus voltage returns (i.e. when it is switched back on) the device will maintain the lock status, active or inactive, which it had when it was turned off. Otherwise, the device will always restart in the unlocked condition (default setting).	

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Button x – Lock function	Button x = enabled Lock function = enabled	1 bit	C-W--	[1.003] enable	4, 20, 36, 52, 68, 84, 100, 116

10.6   Thermostat

The *Temperature control* folder allows the pushbutton to be configured as a room temperature controller and also allows filtering, through a weighted mean average, the reading from the internal sensor with the reading from an auxiliary sensor added on the bus.

The folder is active only if in the *General* Tab the Thermostat functionalities option has been enabled.

It includes the following secondary folders:

- External sensors (from bus)
- Weighted temperature value
- Temperature control
- Relative humidity control
- Energy saving

10.6.1   External sensors (from bus)

Bus sensors are KNX devices (or traditional sensors interfaced to the bus by means of KNX devices) that send values or states to the pushbutton via the bus.

Sensors from bus are KNX devices (or traditional sensors interfaced to the bus by means of KNX devices) which send values or states to the pushbutton via bus.

> [i]   The internal pushbutton control system cyclically monitors the updating status of the values of the external sensors (from bus) when the timeout setting expires. In case no updated value has been received, the regulation function is suspended and the actuation valves are closed. An alarm is issued on the bus through communication object *252 – Room temperature control alarm* (please refer to *Settings* folder).

The External sensors (from bus) Tab is always active and contains the following parameters:

- Room temperature
- Relative humidity
- Antistratification temperature
- Outdoor temperature
- Coil temperature
- Floor surface temperature
- Flow temperature
- Analog sensors timeout
- Anticondensation
- Window contact X (X = 1, 2)
- Presence sensor X (X = 1, 2)
- Card holder contact
- Digital sensors timeout

10.6.1.1 Parameters and communication objects

Parameter name	Conditions	Values
Room temperature		**disabled** / enabled
	It enables a bus temperature sensor. The measured value can be used to calculate a weighted average value in combination with the temperature sensor integrated into the device or a temperature sensor connected to a device input.	
Cyclic reading interval	Room temperature = enabled	**no reading** [other values in the range 30 s … 120 min]
	If the parameter is set to "no reading", the corresponding communication object must be updated by the remote device sending data. With any different value, data are updated with a reading request by the room thermostat.	
Relative humidity		**disabled** / enabled
Humidity CO dimension	Relative humidity = enabled	1 byte (DPT 5.001) 2 **byte (DPT 9.007)**
Cyclic reading interval	Relative humidity = enabled	**no reading** [other values in the range 30 s … 120 min]
Antistratification temperature		**disabled** / enabled
	It enables a temperature bus sensor to carry out the antistratification function.	
Cyclic reading interval	Antistratification temperature = enabled	**no reading** [other values in the range 30 s … 120 min]
Outdoor temperature		**disabled** / enabled
	It enables an outdoor temperature bus sensor to display the measured value on the display. This is alternative to an outdoor temperature sensor connected to a device input: the parameter appears only if the external temperature sensor is disabled in the Inputs folder.	
Cyclic reading interval	Outdoor temperature = enabled	**no reading** [other values in the range 30 s … 120 min]
Coil temperature		**disabled** / enabled
	It enables a bus sensor for measuring the coil temperature of the conveying fluid for heat exchange. The acquisition of the value allows realizing the hot- start function of a fan.	
Cyclic reading interval	Coil temperature = enabled	**no reading** [other values in the range 30 s … 120 min]
Floor surface temperature		**disabled** / enabled
	It enables a bus sensor for measuring the surface temperature of a floor heating system. The acquisition of the value allows to realize the function of surface temperature limitation.	
Cyclic reading interval	Floor surface temperature = enabled	**no reading** [other values in the range 30 s … 120 min]

Parameter name	Conditions	Values
Flow temperature		**disabled** / enabled
	It enables a bus sensor for measuring the flow temperature of the conveying fluid. The acquisition of the value allows calculating the dew-point temperature to realize the active anticondensation protection function in surface cooling plants (floor or ceiling).	
Cyclic reading interval	Flow temperature = enabled	**no reading** [other values in the range 30 s … 120 min]
Analog sensors timeout		**00:05:00** hh:mm:ss [range 00:00:00 … 18:12:15]
	The field has format hh:mm:ss (hours : minutes : seconds): the default value 00:05:00 corresponds to a timeout of 5 minutes. The value 00:00:00 means that the timeout of the analogic sensors is disabled.	
Anticondensation		**disabled** / enabled
	It enables a bus sensor for detecting the condensation.	
Signal	Anticondensation = enabled	**not inverted** / inverted
Cyclic reading interval	Anticondensation = enabled	**no reading** [other values in the range 30 s … 120 min]
Windows contact 1		**disabled** / enabled
	It enables a bus sensor for detecting the state of opening / closing of a window or a door.	
Signal	Windows contact 1 = enabled	**not inverted** / inverted
Cyclic reading interval	Windows contact 1 = enabled	**no reading** [other values in the range 30 s … 120 min]
Windows contact 2		**disabled** / enabled
	It enables a bus sensor for detecting the state of opening / closing of a window or a door.	
Signal	Windows contact 2 = enabled	**not inverted** / inverted
Cyclic reading interval	Windows contact 2 = enabled	**no reading** [other values in the range 30 s … 120 min]
Presence sensor 1		**disabled** / enabled
	It enables a bus sensor for detecting the presence / absence of people within a room.	
Signal	Presence sensor 1 = enabled	**not inverted** / inverted
Cyclic reading interval	Presence sensor 1 = enabled	**no reading** [other values in the range 30 s … 120 min]
Presence sensor 2		**disabled** / enabled
	It enables a bus sensor for detecting the presence / absence of people within a room.	

Parameter name	Conditions	Values
Signal	Presence sensor 2 = enabled	**not inverted** / inverted
Cyclic reading interval	Presence sensor 2 = enabled	**no reading** [other values in the range 30 s … 120 min]
Card holder contact		**disabled** / enabled
	It enables a bus sensor for detecting the presence / absence of people in a hotel room provided with a card holder.	
Signal	Card holder contact = enabled	**not inverted** / inverted
Cyclic reading interval	Card holder contact = enabled	**no reading** [other values in the range 30 s … 120 min]
Digital sensors timeout		**00:05:00** hh:mm:ss [range 00:00:00 … 18:12:15]
	The field has format hh:mm:ss (hours : minutes : seconds): the default value 00:05:00 corresponds to a timeout of 5 minutes. The value 00:00:00 means that the timeout of the digital sensors is disabled.	

About sensor timeout

The internal control system of the thermostat cyclically monitors the updating status of the values of the external sensors (from bus) and the inputs when the timeout setting expires. In case no updated value has been received, the regulation function is suspended.

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Room temperature (from bus)	Temperature sensor = enabled	2 Bytes	C-WTU	[9.001] temperature (°C)	203
Humidity (2 bytes, from bus)	Relative humidity sensor = enabled, Humidity comm. obj. size = 2 byte	2 Bytes	C-WTU	[9.007] humidity (%)	204
Humidity (1 byte, from bus)	Relative humidity sensor = enabled, Humidity comm. obj. size = 1 byte	1 Byte	C-WTU	[5.001] percentage (0..100%)	205
Antistratification temperature (from bus)	enabled	2 Bytes	C-WTU	[9.001] temperature (°C)	206
Outdoor temperature (from bus)	enabled	2 Bytes	C-WTU	[9.001] temperature °C	207
Coil temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	208

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Floor temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	209
Flow temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	210
Anticondensation (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	216
Windows contact sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.019] window/door	211
Windows contact sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.019] window/door	212
Presence sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.018] occupancy	213
Presence sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.018] occupancy	214
Contact of card holder (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	215

10.6.2 Weighted temperature value

The **Weighted temperature value** tab can be used only if the acquisition of the room temperature is enabled both from the internal (main) sensor and from the bus.

It contains the following parameters:

- Relative weight
- Minimum change of value to send [K]
- Cyclic sending interval

10.6.2.1 Parameters and communication objects

Parameter name	Conditions	Values
Relative weight		100% main sensor 90% / 10% 80% / 20% 70% / 30% 60% / 40% **50% / 50%** 40% / 60% 30% / 70% 20% / 80% 10% / 90% 100% sensor from bus
	The main sensor is always the integrated sensor; the sensor from bus needs to be activated in Sensors from bus folder.	

Parameter name	Conditions	Values
Minimum change of value to send [K]		**0,5** [other values in range 0 … 5 K]
	If the parameter is set to 0, no value is sent for change	
Cyclic sending interval		**no sending** [other values in range 30 s … 120 min]

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Weighted temperature	Cyclic sending interval ≠ no sending	2 Byte	CR-T--	[9.001] temperature °C	217

10.6.3 Note on weighted temperature value

The device allows the acquisition of the room temperature in 2 ways:

    1)   from an external temperature sensor connected to a device input configured as NTC probe;

    2)   via bus from another KNX device, e.g. from an ekinex pushbutton (External sensors (from bus) `->` Room temperature = enabled).

To optimize or correct the room temperature regulation in special cases (in large rooms, in presence of strong asymmetry of the temperature distribution, when the installation of the device is in a position not suitable, etc.), the device can then use a weighted average between two temperature values. The weights are assigned by the parameter *Relative weight* that assigns a ratio of the two values.

Note: the value for "Weighted Temperature" communication object is set to 7F FF in case of the real value cannot be read from bus.

10.7 Temperature control

The *Temperature control* tab can be used only if the acquisition of the room temperature is enabled at least from the internal sensor. It contains the following secondary tabs:

- Settings
- Heating
- Cooling
- Ventilation

The *Cooling* tab is displayed only if the Thermostat function parameter is set to either *"heating and cooling"* or *"cooling"* value in the *Settings* tab.

The *Ventilation* tab is displayed in the following cases:

- if the Heating or Cooling type is set to "fancoil" in the Heating or Cooling tab, and/or
- if the *"Ventilation for auxiliary heating/cooling"* is set to *"enabled"* in the Heating/Cooling -> Auxiliary heating/cooling section.

10.7.1 Settings

The *Settings* includes the parameters to perform the basic configuration of the room temperature controller:

- Setpoint type
- Thermostat function
- Command communication object unique or separated (for 2 or 4-pipes systems)
- Heating/cooling changeover
- Heating/cooling cyclic sending interval
- HVAC mode after download
- Setpoint cyclic sending interval
- End of manual operation
- Disable temperature controller from bus
- Signal from bus
- Transmission delay after mode change
- Valve protection function
- Frequency
- Time interval

This folder is always active.

10.7.1.1 Parameters and communication objects

Parameter name	Conditions	Values
Setpoint type	General -> Thermostat functionalities = enabled	**Single** Absolute Relative
	In case the option "Single" setpoint is selected and Thermostat function = heating, the temperature controller acts on heating mode; in case Thermostat function = cooling, the temperature controller acts on cooling mode. *In case Thermostat function = both heating and cooling, the current seasonal mode needs to be specified by the proper communication object.*	
Thermostat function		**heating** cooling both heating and cooling
Command Communication Object	Thermostat function = both heating and cooling	**separated** / unique
Heating–cooling changeover	Setpoint type ≠ single Command Communication Object = separated Thermostat function = both heating and cooling	**from bus** / automatic
	In case Setpoint management = single setpoint, or the Command Communication Object is unique, the heating-cooling switch over must be carried out from bus.	

Parameter name	Conditions	Values
Heating-cooling cyclic sending interval	Thermostat function = both heating and cooling	**no sending** [other values in range 30 s … 120 min]
HVAC mode after download	Thermostat function = both heating and cooling Heating–cooling changeover = from bus	**no change** heating cooling
Setpoint cyclic sending interval		**no sending** [other values in range 30 s … 120 min]
	In case Setpoint management = single, the actual setpoint value takes only into account the actual state of the contacts window (if the corresponding function is enabled). *In case Setpoint management = relative, the actual setpoint value also depends on the operating mode set manually by the user or automatically by another KNX supervising device with the possibility of time scheduling.*	
End of manual operation	Setpoint management = Absolute setpoints or Relative setpoints	**till first telegram from bus** [other values in the range 30 min…48 h]
	It defines the end of manual operation modes.	
Disable temperature controller from bus		**no** / yes
Signal from bus	Disable temperature controller from bus = yes	**not inverted** inverted
Transmission delay after mode change		**00:00:04.000** [range 00:00:00.000 … 00:10:55.530 hh:mm:ss.fff]
	It defines a delay in transmission on the bus after a change of HVAC mode. *A null value (00:00:00) means that the transmission is immediate.*	
Valve protection function	Heating type ≠ electric AND Thermostat function ≠ heating	**disabled** enabled
	It enables the function that activates the drive for the valve control during periods of inactivity of the system.	
Frequency	Valve protection function = enabled	once a day **once a week** once a month
Time interval	Valve protection function = enabled	**10 s** [other values in range 5 s … 20 min]

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Actual setpoint		2 Bytes	CR-T--	[9.001] temperature (°C)	225

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Manual setpoint	Setpoint type = absolute or relative	2 Byte	C-W--	[9.001] temperature (°C)	226
Heating / cooling status out		1 Bit	CR-T--	[1.100] heating/cooling	218
	The communication object is sent over the bus after an internally elaborated switching event. The object is always exposed and contains information on the current conduction mode of the internal temperature controller. **[1.100] DPT Heat/Cool 1 Bit** **0 = Cooling** **1 = Heating**				
Heating / cooling status in	Thermostat function = both heating and cooling, Heating–cooling changeover = from bus	1 Bit	C-W--	[1.100] heating/cooling	219
	The communication object is received from the bus. At the switching event the internal regulator switches the conduction mode.				
HVAC mode in	Setpoint type = absolute or relative	1 Byte	C-W--	[20.102] HVAC mode	220
	Bits 5, 8 are reserved. **[20.102] DPT HVAC Mode 1 Byte** AUTO `0 0 0 0` COMFORT `0 0 0 1` STAND-BY `0 0 1 0` ECONOMY `0 0 1 1` PROTECTION `0 1 0 0` *The device receives the operating mode (HVAC mode) from a bus device with supervisor function. The operating mode received via this communication object can be subsequently changed by the user (in this case the room thermostat switches to manual control).*				
HVAC forced mode in	Setpoint type = absolute or relative	1 Byte	C-W--	[20.102] HVAC mode	221
	The communication object receives the operating mode in the same way as it happens with the communication object HVAC mode in; the difference is that the operating mode received via this object (except for the AUTO command) cannot be subsequently changed by the user. The user can change the mode only after the forced HVAC mode in has sent the AUTO command.				
HVAC mode out	Setpoint type = absolute or relative	1 Byte	CR-T-	[20.102] HVAC mode	222
HVAC manual mode	Setpoint type = absolute or relative	1 Byte	C-WTU	[20.102] HVAC mode	223
Chrono active status	Setpoint type = absolute or relative	1 Bit	CR-T-	[1.011] state	224

Room temperature controller status		1 Bit	CR-T-	[1.003] enable	235
Thermostat - Alarm text		14 bytes	CR-T-	[16.000] Character string (ASCII)	247
Manual setpoint active status	Setpoint type = absolute or relative	1 Bit	CRWTU-	[1.011] state	248
Input setpoint	Setpoint type = single	2 bytes	CRWTU	[9.001] temperature (°C)	227
	The communication object allows to set and / or read the status of the setpoint (manual / forced).				
Room temperature controller alarm		1 bit	CR-T--	[1.005] alarm	252
Disable room temperature controller	Disable temperature controller from bus = yes	1 bit	C-W--	[1.001] switch	253
Thermal generator lock		1 bit	C-W--	[1.005] alarm	264
Building protection HVAC mode active		1 bit	CR-T--	[1.011] state	265

Note on the heating and cooling system terminals

The application functions of the thermostat configurable with the ETS application are particularly suitable for command / control by means of KNX actuators (generic or dedicated) of the following system terminals:

- radiators;
- electric heaters;
- fancoils;
- floor or ceiling radiant panels;
- dehumidifiers;
- radiant panels + radiators (as an auxiliary stage);
- radiant panels + fancoil (as an auxiliary stage);
- radiant panels + dehumidifiers.

10.7.1.2 Seasonal mode switching (Heating / cooling)

The switch between heating and cooling mode can take place in 2 ways:

- automatically, depending on the room temperature (i.e., through a command from the internal logic of the device);
- from the KNX bus via communication object.

Automatic switch over, based on the room temperature

This mode is only suitable in applications for heating / cooling hydraulic systems with a 4-pipe configuration. Even in this case, the information can be sent on the bus with the output communication object [DPT 1,100 heat / cool]; the difference with mode 1 is that the switch over is performed automatically by the device, basing on the values of current temperature and setpoint. The automatic switch over is achieved by introducing a dead band, as shown in Figure 22**.**

.
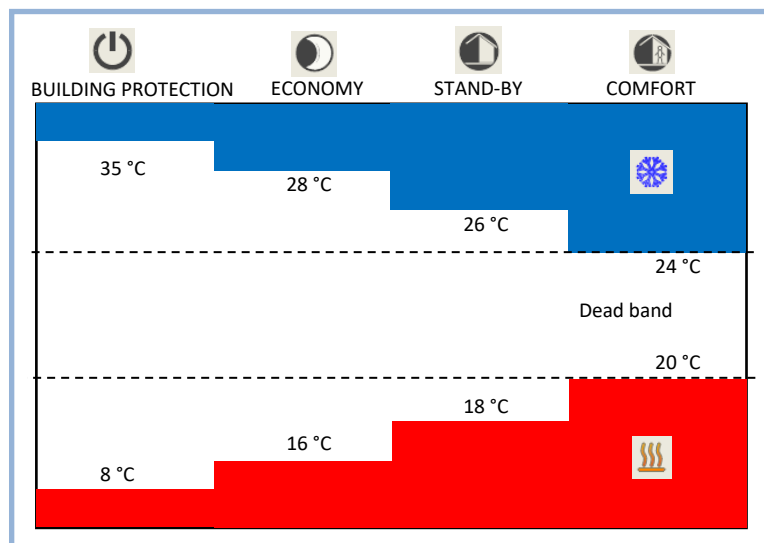
Figure 22 - Automatic switch over, based on the room temperature

As long as the actual measured temperature is below the heating mode setpoint, the heating mode is selected; similarly, if the value is greater than the cooling setpoint, then cooling mode is selected. If the value is within the dead band, the operation mode remains unchanged; the heating/cooling switchover point must correspond to the actual setpoint of the current HVAC mode, and in the same way the cooling/heating switchover must correspond to the actual heating setpoint.

Switch over from KNX bus

Switching from the bus requires that the command is sent out from another KNX device, for example another thermostat, a Touch & See unit or a supervision software configured for this purpose. This acts as a "supervisor" device: the switching takes place via the input communication object [DPT 1.100 heat / cool]. In this mode, manual switching by the user is not allowed. With this mode, the supervisor is able to let "slave" devices carry out timed programs by extending their function to a chronothermostat one (controlled centrally by the supervisor appliance).

The communication objects shown in the block diagram allow to monitor and modify the current conduction mode imposed on the temperature controller. Object 218 - *Heating / cooling status out* is always exposed, even when the thermostat function is set to heating or cooling only. In the event that the function is set to both heating and cooling, the cyclical sending of the object on the bus can be enabled; in all cases, information on the current mode of management can be acquired with a request to read this communication object.

Object 219 - *Heating / cooling status in* is exposed only when the thermostat function is set to both heating and cooling and the switching between the modes is performed by the bus.
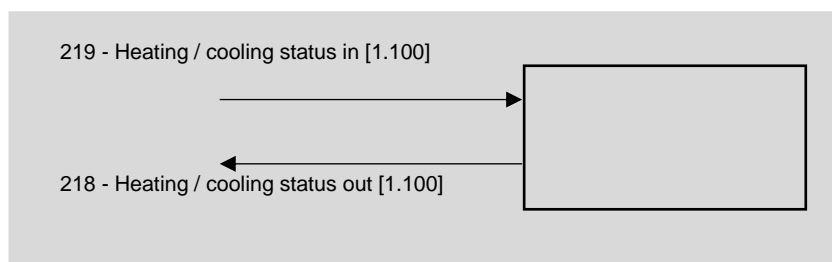
Figure 23 - Switch over from KNX bus

10.7.1.3 Valve protection function

This function is suitable for heating and cooling systems that use water as heat transfer fluid and have motorized valves for the interception of a zone or a single room. Long periods of inactivity of the system can lead to blocking of the valves: to prevent this eventuality, the thermostat can periodically send a valve opening / closing command during the period of inactivity of the system. This possibility is made available in the application program by means of the *"Valve protection function"* parameter, further defined through the frequency and duration of the valve actuation.

Note: this function is not available whenever the heating type is electric and at the same time the thermostat function is heating only.

10.7.1.4 Remote Setpoint modification

These communication objects allow to perform the same modifications remotely, for example from a supervisor software.
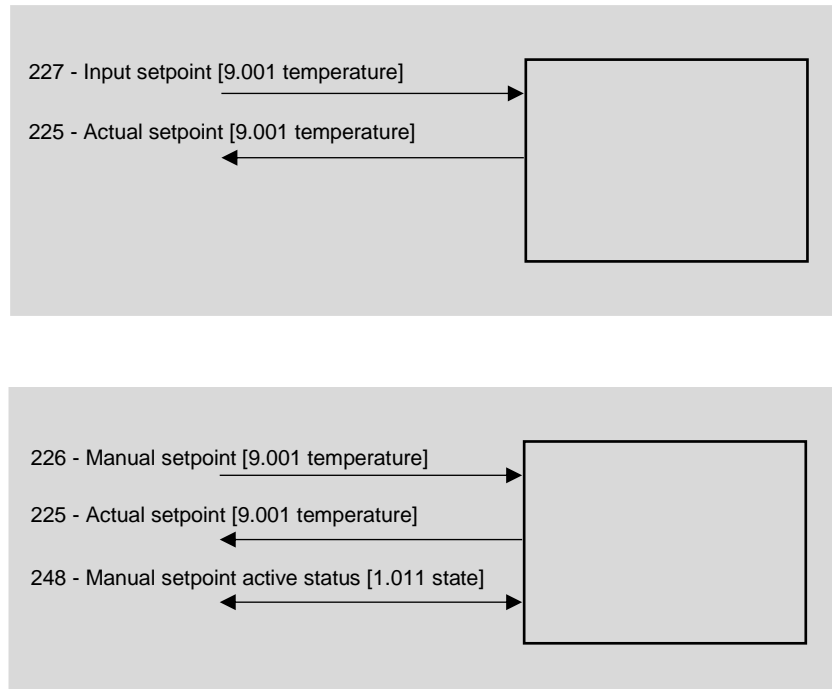
Figure 24 – Remote setpoint setting

Those objects are about the Setpoint forced modification: alternatively, the supervisor can act directly on the operating mode setpoints (C.O. from 227 to 234). The value of the C.O. *225 - Actual setpoint* represents the

current operative setpoint which the control algorithms are based on. The C.O. *248 - Manual setpoint active status* indicates (read request mode) if the forced mode is active (symbol M on the LCD display present). The supervisor can force at any time the actual setpoint by writing a new value directly into the C.O. *226 - Manual setpoint*. The C.O. *248 - Manual/forced setpoint active status* can also be used in writing mode, to exit the active forced mode.

10.7.1.5 Remote operative mode modification

The communication objects shown in Figure 25 allow to monitor the operating mode (comfort, standy, economy and building protection) modifications performed locally by the user when interacting with the LCD display and the touch buttons of the room thermostat, or the operating mode forced by chrono program.

220 - HVAC mode in [20.102]

223 - HVAC manual mode [20.102]

222 - HVAC out mode [20.102]

224 - Chrono active status [1.011 state]
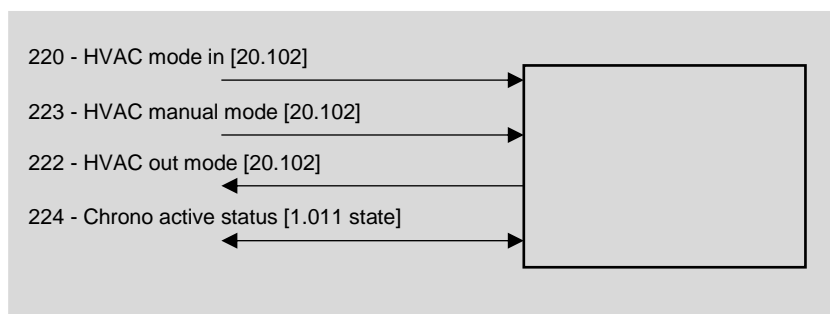
Figure 25 - Remote operative mode modification

The C.O. *220 - HVAC mode in* is associated to the chrono program. The C.O.s *222 - HVAC mode out* and *224 - HVAC chrono active status* allow the remote supervisor to detect the operating mode currently active on the room thermostat. Moreover, it also allows to understand if the chrono program is active or if attenuation is handled manually or not. The supervisor can set at any time a manual operating mode through C.O. *223 - HVAC manual mode*; to start the chrono program remotely, the C.O. *223 - HVAC manual mode* is to be set on value 0 = Automatic.

10.7.2 Heating

The *Heating* folder allows to set:

- The default value for single and relative setpoints (comfort, standby and economy setpoints);
- Control algorithm type (2-point hysteresis, PWM or continuous) and internal parameters;
- Building protection mode activation based on up to 2 window contacts' status.

Conditions of activation: *Settings* ⇒ Thermostat function = heating or both heating and cooling.

This folder contains the following parameters:

- Temperature setpoint [°C]
- Comfort temperature setpoint [°C]
- Standby temperature setpoint [°C]
- Economy temperature setpoint [°C]
- Standby temperature offset [0,1 K]
- Economy temperature offset [0,1 K]
- Building protection temperature setpoint [°C]
- Heating type
- Control type
- Hysteresis
- Hysteresis position
- Cyclic sending interval
- Min. change of value to send [%]
- PWM cycle time
- Proportional band [0.1 K]
- Integral time [min]
- Min control value [%]
- Max control value [%]
- Floor temperature limitation
- Temperature limit [°C]
- Hysteresis [K]
- Auxiliary heating
- Communication object
- Disabled from bus
- Offset from setpoint
- Hysteresis
- Cyclic sending interval
- Ventilation for auxiliary heating

10.7.2.1 Parameters and communication objects

Parameter name	Conditions	Values
Temperature setpoint [°C]	Setpoint type = Single	**21** [range 10 … 50]

Parameter name	Conditions	Values
Comfort temp. setpoint [°C]	Setpoint type = absolute or relative	**21** [range 10 … 50]
Standby temp. setpoint [°C]	Setpoint type = absolute	**18** [range 10 … 50]
	For proper operation, Standby temperature setpoint has to be < Comfort temperature setpoint.	
Economy temp. setpoint [°C]	Setpoint type = absolute	**16** [range 10 … 50]
	For proper operation, Economy temperature setpoint has to be < Standby temperature setpoint.	
Standby temp. offset [0,1 K]	Setpoint type = relative	**-30** [range -10 … -50]
Economy temp. offset [0,1 K]	Setpoint type = relative	**-50** [range -10 … -50]
	For proper operation, Standby temperature offset has to be < Comfort temperature offset.	
Building protection temp. setpoint [°C]		**7** [range 2 … 10]
Heating type		**radiators** electric fancoils floor radiant panels ceiling radiant panels
	It defines the terminal used for room heat exchange. The choice sets the default parameters proposed for the PWM control algorithm (proportional band and integral time) and the control options.	
Control type		**2 point hysteresis**, PWM (pulse width modulation) continuous
Hysteresis	Control type = 2 point hysteresis	**0,3 K** [other values in range 0,2 K … 3 K]
Hysteresis position	Heating type = floor radiant panels or ceiling radiant panels Control type = 2 point hysteresis	**below** above
	The above hysteresis is suitable in case of special applications requiring mixing group control.	
Cyclic sending interval	Control type = 2 point hysteresis, continuous	**no sending** [other values in range 30 s … 120 min]
Min. change of value to send [%]	Control type = continuous	**10** [range 0 … 100]
PWM cycle time	Control type = PWM (pulse width modulation)	**15 min** [range 5 … 240 min]

Parameter name	Conditions	Values
Proportional band [0,1 K]	Control type = continuous or PWM, Heating type = other	**40** [range 5 … 100]
	The value is represented in tenths of a Kelvin degree (0,1 K). *The field contains a preset value which depends on the type of heating selected (the value can be changed):* • *radiators: 50 (5 K)* • *electric: 40 (4K)* • *fan-coil: 40 (4K)* • *radiant floor: 50 (5 K)* • *radiant ceiling: 50 (5 K)* *The value of the Proportional Band parameter represents the maximum deviation between the desired and measured temperature which determines the maximum control output.*	
Integral time [min]	Control type = continuous or PWM, Heating type = other	**90** [range 0 … 255 min]
	This field contains a preset value which depends on the selected heating type (the value can be changed): • *radiators: 150 min* • *electric: 100 min* • *fancoils: 90 min* • *floor radiant panels: 240 min* • *ceiling radiant panels: 180 min*	
Min control value [%]	Control type = PWM or continuous	**15 %** [range 0 %…30 %]
Max control value [%]	Control type = PWM or continuous	**85 %** [range 70 %…100 %]
Floor temperature limitation	Heating type = floor radiant panels External sensors (from bus) ⇒ Floor surface temperature = enabled	**disabled** enabled
	This parameter enables the floor temperature limitation of a floor radiant panel. It is mandatory to measure the floor surface temperature by enabling the corresponding temperature sensor in "External sensors (from bus)" folder. *__Important!__ This function does not replace the over-temperature protection usually installed in hydronic floor systems, realized with the proper safety thermostat.*	
Temperature limit [°C]	Floor temperature limitation = enabled	**29** [range 20 … 40]
	According to EN 1264 a maximum allowed temperature is prescribed for the surface of a floor heating system: • *T(sup) max ≤ 29°C for normal occupancy zones;* • *T(sup) max ≤ 35°C for peripheral areas.* *National standard may limit those temperatures to lower values. As "peripheral areas" are defined bands generally located along the walls of the environment facing the outside of the building, with maximum width of 1 m.*	
Hysteresis [K]	Floor temperature limitation = enabled	**0,3 K** [other values in the range 0,2 K … 3 K]
	Before exiting the alarm status, the device waits until the surface temperature drops below the set threshold by an offset equal to the hysteresis value.	
Auxiliary heating		**disabled** / enabled
Communication object	Auxiliary heating = enabled	unique **separated**
Disabled from bus	Auxiliary heating = enabled	**no** / yes
	It enables the activation and deactivation of the function through a telegram sent on the bus by a supervising device.	

Parameter name	Conditions	Values
Offset from setpoint	Auxiliary heating = enabled	**0,6 K** [other values in the range 0 …3 K]
Hysteresis [K]	Auxiliary heating = enabled	**0,3 K** [other values in the range 0,2 K … 3 K]
Cyclic sending interval	Auxiliary heating = enabled	**no sending** [other values in the range 30 s … 120 min]
Ventilation for auxiliary heating	• Command Communication Object = unique • Heating type = floor radiant panels or ceiling radiant panels OR • Command Communication Object = separated • Heating type = radiators, electric, floor radiant panels or ceiling radiant panels	**disabled** / enabled
	This option allows to match a system with high inertia as the floor radiant panels (hydronic version) with a system with low inertia as the fan-coils.	

All communication objects are active if *Settings* ⇒ Thermostat function = heating or both heating and cooling.

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Comfort setpoint (heating)	Setpoint type = absolute or relative	2 Byte	CRWTU	[9.001] temperature (°C)	227
Standby setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU	[9.001] temperature (°C)	229
Standby offset (heating)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	229
Economy setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU	[9.001] temperature (°C)	231
Economy offset (heating)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	231
Building protection setpoint (heating)		2 Byte	CRWTU-	[9.001] temperature (°C)	233
Heating out command	Control type = 2 points hysteresis or PWM Command communication object = separated	1 Bit	CR-T-	[1.001] switch	236

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Heating out command	Control type = continuous Command communication object = separated	1 Byte	CR-T-	[5.001] percentage (0..100%)	236
Heating and cooling out command	Command communication object = unique, Control type = 2 points hysteresis or PWM	1 Bit	CR-T-	[1.001] switch	236
Heating and cooling out command	Command communication object = unique, Control type = continuous	1 Byte	CR-T-	[5.001] percentage (0..100%)	236
Auxiliary heating output command	Command communication object = separated, Auxiliary heating = enabled	1 Bit	CR-T-	[1.001] switch	238
Auxiliary heating and cooling output command	Command communication object = unique, Auxiliary heating = enabled	1 Bit	CR-T-	[1.001] switch	238
Auxiliary heating disable	Auxiliary heating = enabled, Disabled from bus = yes	1 Bit	C-W--	[1.003] enable	240

10.7.3 Cooling

The *Cooling* folder includes the following parameters:

- Temperature setpoint [°C]
- Comfort temperature setpoint [°C]
- Standby temperature setpoint [°C]
- Economy temperature setpoint [°C]
- Standby temperature offset [0,1 K]
- Economy temperature offset [0,1 K]
- Building protection temperature setpoint [°C]
- Cooling type
- Control type
- Hysteresis [K]
- Hysteresis position
- Cyclic sending interval
- Proportional band [0,1 K]
- Integral time [min]
- PWM cycle time
- Min. change of value to send [%]
- Min. control value [%]
- Max. control value [%]
- Anticondensation with probe

- Active anticondensation
- Flow temperature (project)
- Anticondensation hysteresis range
- Delay for alarm signal
- Auxiliary cooling
- Disabled from bus
- Offset from setpoint
- Hysteresis [K]
- Cyclic sending interval
- Ventilation for auxiliary cooling

10.7.3.1 Parameters and communication objects

Conditions: *Settings* ⇒ Thermostat function = cooling or both heating and cooling.

Parameter name	Conditions	Valori
Temperature setpoint [°C]	Setpoint type = single	**23** [range 10 … 50]
Comfort temperature setpoint [°C]	Setpoint type = absolute or relative	**23** [range 10 … 50]
Standby temperature setpoint [°C]	Setpoint type = absolute	**26** [range 10 … 50]
	For a correct operation of the device the standby temperature setpoint has to be > comfort temperature setpoint.	
Economy temperature setpoint [°C]	Setpoint type = absolute	**28** [range 10 … 50]
	For a correct operation of the device the economy temperature setpoint has to be > standby temperature setpoint.	
Standby temperature offset [0,1 K]	Setpoint type = relative	**30** [range 10 … 50]
Economy temperature offset [0,1 K]	Setpoint type = relative	**50** [range 10 … 80]
	For a correct operation of the device the economy temperature offset has to be > standby temperature offset.	
Building protection temp. setpoint [°C]		**36** [range 30 … 50]

Parameter name	Conditions	Valori
Neutral zone [0,1 K]	Settings ⇒ Thermostat function = both heating and cooling Settings ⇒ Command Communication Object = separated Settings ⇒ Heating–cooling changeover = automatic Setpoint type = absolute or relative	**20** [range 10 … 80]
	It defines the width of the neutral zone, in case it is requested to automatically switch between heating and cooling starting from the Comfort heating setpoint, as soon as this zone is exceeded.	
Cooling type		**fancoils** floor radiant panels ceiling radiant panels
	If in Settings the parameter Thermostat function = both heating and cooling and Command communication object = unique, the parameter Cooling type is bound to the selection done for Heating.	
Control type		**isteresi a 2 punti**, PWM (modulazione ad ampiezza d'impulso), continuo
Hysteresis	Control type = 2 point hysteresis	**0,3 K** [other values in 0,2 K … 3 K range]
Hysteresis position	Cooling type = floor radiant panels or ceiling radiant panels Control type = 2 point hysteresis	**below** above
	The below hysteresis is suitable in case of special applications requiring mixing group control.	
Cyclic sending interval	Control type = 2 point hysteresis, continuous	**No sending** [[other values in 30 s … 120 min range]

Parameter name	Conditions	Valori
Proportional band [0,1 K]	Control type = continuous or PWM	**50** [range 5 … 100]
	The value is in tenths of Kelvin (K) degree. **) The field contains a preset value that depend on the selected heating type (the value can be modified):* • *radiators: 50 (5 K)* • *electric: 40 (4 K)* • *fan-coils: 40 (4 K)* • *floor radiant panels: 50 (5 K)* • *ceiling radiant panels: 50 (5 K)* The value of the parameter Proportional band represents the max difference between the setpoint temperature and the measured temperature that causes the max control output.	
Integral time [min]	Control type = continuous or PWM	**100** [other values in the range 0 … 255 min]
	The field contains a preset value that depend on the selected heating type (the value can be modified): • *fan-coils: 90 min* • *floor radiant panels: 240 min* • *ceiling radiant panels: 180 min*	
PWM cycle time	Control type = PWM (pulse width modulation)	**15 min** [range 5 … 240 min]
Min. change of value to send [%]	Control type = continuous	**10** [range 0 … 100]
Min control value [%]	Control type = continuos or PWM	**15** [range 0…30]
Max control value [%]	Control type = continuos or PWM	**85** [range 70…100]
Anticondensation with probe	Cooling type = fllor radiant panels or ceiling radiant panels External sensors (from bus) ⇒ Anticondensation = enabled	Enabled **disabled**
Active anticondensation	Cooling type = floor radiant panels or ceiling radiant panels, External sensors (from bus) ⇒ Relative humidity = enabled	**disabled** enabled (project temperature)
	Cooling type = floor radiant panels or ceiling radiant panels, External sensors (from bus) ⇒ Relative humidity = enabled External sensors (from bus) ⇒ Flow temperature = enabled	**disabled** enabled (comparison between flow temperature and dew-point)
	If flow temperature is lower than calculated dew point, the operating mode is cooling and the room thermostat is in flow request, then the thermostat will close the valve and send an alarm message over the bus.	
Flow temperature (project)	Active anticondensation = enabled (project temperature)	**14 °C** [other values in the range 14 °C … 20 °C]
	Only displayed if the flow temperature from external sensor (from bus) is not enabled.	

Parameter name	Conditions	Valori
Anticondensation hysteresis range	Active anticondensation = enabled (project temperature) External sensors (from bus) ⇒ Flow temperature = enabled	0,2 K / 0,3 K / 0,4 K / **0,5** / 0,6 K 0,8 K / 1 K / 1,5 K / 2 K / 2,5 K / 3 K
	Before exiting the alarm condition, it is expected that the calculated dew-point temperature will drop below the delivery temperature by an offset equal to the hysteresis value.	
Delay for alarm signal	Active anticondensation = enabled (project temperature), or Anticondensation with probe = enabled	**30 s** [other values in the range 30 s … 120 min]
Auxiliary cooling		**disabled** / enabled
Disabled from bus	Auxiliary cooling = enabled	**no** / yes
	This parameter enables the activation and deactivation of the function through a telegram from a bus device with supervising function.	
Offset from setpoint	Auxiliary cooling = enabled	0 K / 0,2 K / 0,4 K / **0,6 K** 0,8 K / 1 K / 1,5 K / 2 K / 2,5 K / 3 K
Hysteresis	Auxiliary cooling = enabled	0,2 K / **0,3 K** / 0,4 K / 0,5 / 0,6 K 0,8 K / 1 K / 1,5 K / 2 K / 2,5 K / 3 K
Ventilation for auxiliary cooling	Cooling type = floor radiant panels or ceiling radiant panels	**disabled** / enabled
	This option allows to combine a high-inertial system as the floor radiant panels to a low-inertial one as the fan-coils.	

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Comfort setpoint (cooling)	Setpoint type = absolute or relative	2 bytes	CRWTU	[9.001] temperature (°C)	228
Standby setpoint (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.001] temperature (°C)	230
Standby offset (cooling)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	230
Standby offset (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.001] temperature (°C)	232
Economy offset (cooling)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	232
Building protection setpoint (cooling)		2 bytes	CRWTU	[9.001] temperature (°C)	234
Cooling out command	Command Communication Object = separated Control type = 2 point hysteresis or PWM	1 Bit	CR-T-	[1.001] switch	237

Object name	Conditions	Dim.	Flags	DPT	No. Comm. Obj.
Cooling out command	Command Communication Object = separated Control type = continuous	1 Byte	CR-T-	[5.001] percentage (0..100%)	237
Auxiliary cooling output command	Auxiliary cooling = enabled	1 bit	CR-T-	[1.001] switch	239
Auxiliary cooling disable	Auxiliary cooling = enabled Disabled from bus = yes	1 bit	C-W--	[1.003] enable	241
Anticondensation alarm	Anticondensation with probe or Active anticondensation = enabled	1 bit	CR-T-	[1.005] alarm	263

About anticondensation protection function

The purpose of this function is to prevent the condensation on the thermal exchange surfaces of the installation or building when cooling is working. This function is mainly used in systems with thermal exchange consisting in surface terminals such as for the floor and ceiling cooling radiant systems. In this case the hydraulic circuits contain refrigerated water; usually the latent loads (due to the increase of air humidity in the room) are handled by air-conditioning units and the temperature and humidity conditions are far from those that could cause condensation. If this is not done in a satisfactory manner, or in case of stop of the air-conditioning units, it is necessary to provide additional safety measures to prevent or restrict the accidental formation of condensation on cold surfaces.

From a general point of view, the anticondensation protection function can be realized:

- by installing a proper room anticondensation probe; when this is active, the hydraulic circuit closes down. It is a passive protection, because the intervention takes place when condensation has already started;
- by calculating the dew-point temperature and confronting it with the conveying fluid flow temperature. If the critical condition for condensation is approaching, you can intervene by closing down the hydraulic circuit or adjusting the mixing conditions of the conveying fluid. This is an active protection because the goal is to prevent the condensation.

Nr.	Type	Denomination	Description
1a	Passive	Anticondensation protection by probe (via bus)	The thermostat receives the information about condensation via bus from a different KNX device through communication object 33: Anticondensation (from bus) [DPT 1.001 switch].
2a	Active	Anticondensation protection with comparison between flow temperature (constant projected value, set as parameter on ETS and dew-point temperature (calculated by the thermostat)	Software protection that intervenes by closing down the room cooling circuit when the flow temperature defined in the hydronic project (as set in the corresponding ETS parameter) is lower than dew-point temperature calculated by the room thermostat using temperature and relative humidity values. The communication object involved is 57: Cooling out command [DPT 1.001 switch].
2b	Active	Anticondensation protection with comparison between flow temperature (constant projected value, set as parameter on ETS and dew-point temperature (calculated by the thermostat)	Software protection that intervenes by closing down the room cooling circuit when the actual measured flow temperature and received via bus from a different KNX device is lower than dew-point temperature calculated by the room thermostat using temperature and relative humidity values. The communication objects involved are 27 at input: Flow temperature (from bus) [DPT 9.001 temperature °C] and 57: Cooling out command [DPT 1.001 switch].

3	Active	Anticondensation protection with dew-point temperature sending over the bus and adjustment of the flow temperature	Software protection that foresees the sending on the bus of the dew-point temperature calculated by the room thermostat using temperature and relative humidity values to a KNX device capable of controlling the mixing condition of the conveying fluid for the cooling circuit. The regulation is performed by the KNX device receiving the dew-point temperature sent by the thermostat. The communication object involved is 73: Dew-point temperature [DPT 9.001 temperature °C].

Table 7 - Anticondensation protection modes

If an anticondensation sensor is used, it is necessary use a device provided with a potential-free signalling contact. It is possible to connect the signalling contact to an input channel of another KNX device, e.g. a pushbutton interface or a binary input (External sensors (from bus) ⇒ Anticondensation sensor = enabled). In this case the signal of the sensor is transmitted to the room temperature controller through the status of a communication object (case 1b described in Table 7).

If the comparison between dew-point temperature calculated by the thermostat and flow temperature of the conveying fluid is used, there are 3 options:

- if the flow temperature value is not available (case 2a of Table 7), you can insert the value used in the project (parameter Flow temperature (projected));
- if the flow temperature value is available (case 2b of Table 7), you enable the Anticondensation Active parameter for comparison;
- if an bus actuator capable of intervention on the conveying fluid's mixing is available, the thermostat sends on the bus the calculated value of the dew-point temperature; this parameter has to be enabled in the *Relative humidity control ⇒ Calculated psychrometric values* tab.
  The actuator compares this value with the flow temperature and, if necessary, modifies the mixing conditions in order to prevent the risk for condensation formation.

The proper anticondensation protection mode needs to be evaluated during the thermical plant design and depends on many factors such as type of building, continuity of service and desired comfort level, available KNX devices, and so on.

10.7.4 Main and auxiliary ventilation

The *Ventilation* folder includes the following parameters:

- Ventilation function
- Control type
- Threshold first speed [0,1 K]
- Threshold second speed [0,1 K]
- Threshold third speed [0,1 K]
- Speed control hysteresis [K]
- Speed proportional band [0,1 K]
- Min. change of value to send [%]
- Manual operation
- Coil temperature usage for fan activation (Hot start)
- Antistratification function
- Disable ventilation from bus

- Signal from bus
- Fan start delay
- Fan stop delay

The conditions for the appearance of the **Ventilation** folder are:

Heating ⇒ Heating type = fan-coils or Cooling type = fan-coils

or a combination of the two conditions:

Heating ⇒ Heating type = floor radiant panels or ceiling radiant panels and *Heating* ⇒ Ventilation for auxiliary heating = enabled

Cooling ⇒ Cooling type = floor radiant panels or ceiling radiant panels and *Cooling* ⇒ Ventilation for auxiliary cooling = enabled

This way two types of installations can be controlled: i) fancoil terminals or ii) radiant panels as main stage and fancoil terminals as auxiliary stage.

10.7.4.1 *Parameter and communication object tables*

Parameter name	Conditions	Values
Control type		**1 speed** 2 speeds 3 speeds continuous regulation
Threshold first speed [0,1 K]	Control type ≥ 1 speed	**0** [range 0 … 255]
	The value is represented in tenths of Kelvin degrees. If the parameter Thermostat function = both heating and cooling, the threshold value is valid for both seasonal modes.	
Threshold second speed [0,1 K]	Control type ≥ 2 speeds	**10** [range 0 … 255]
	The value is represented in tenths of Kelvin degrees. If the parameter Thermostat function = both heating and cooling, the threshold value is valid for both seasonal modes. For a correct operation of the ventilation, Threshold second speed > Threshold first speed.	
Threshold third speed [0,1 K]	Control type = 3 speeds	**20** [range 0 … 255]
	The value is represented in tenths of Kelvin degrees. If the parameter Thermostat function = both heating and cooling, the threshold value is valid for both seasonal modes. For a correct operation of the ventilation, Threshold third speed > Threshold second speed.	
Speed control hysteresis	Control type = 1, 2 or 3 speeds	**0,3 K** [other values in the range 0,2 K …3 K]
Speed proportional band [0,1 K]	Control type = continuous regulation	**30** [range 5 … 100]
	The value is represented in tenths of Kelvin degrees. If the parameter Thermostat function = both heating and cooling, the threshold value is valid for both seasonal modes.	
Min. change of value to send [%]	Control type = continuous regulation	**10** [range 2 … 40]
	Please refer to the Control Algorithms chapter for further information about the meaning of this parameter.	

Parameter name	Conditions	Values
Manual operation		**not depending on the temperature** depending on the temperature
	If the parameter = not depending on the temperature, the fan speed set by the user is not changed even when the temperature setpoint is reached; if the parameter = depending on the temperature, the fan stops when the temperature setpoint is reached.	
Hot start	Thermostat function = heating or both heating and cooling, Heating type = fancoils External sensors (from bus) $\Rightarrow$ coil temperature = enabled	**no** / yes
	To carry out the function, a sensor for measuring the temperature of the heat exchanger of the fan coil has to be enabled. To this purpose, an external sensor (from bus) can be used.	
Min. temp.to start ventilation [°C]	Hot start = yes	**35** [range 28 …40]
	If enabled, the function is active only in heating mode.	
Antistratification function	External sensors (from bus) $\Rightarrow$ Antistratification temperature = enabled	**disabled** / enabled
	To carry out the function, at least a sensor for measuring a second temperature value must be enabled at a different height than that of the room temperature controller. To this purpose, an external sensor (from bus) can be used.	
Antistratification temp. differential	Antistratification function = enabled	**2 [K/m]** [other values in the range 0,25 … 4,00 K/m]
	The DIN 1946 recommends a max temperature gradient of 2 K/m for rooms with standard height (between 2,70 and 3 m).	
Hysteresis	Antistratification function = enabled	**0,5 K** [other values in the range 0,2 … 3 K]
Disable ventilation from bus		**no** / yes
Signal from bus	Disable ventilation from bus = yes	**not inverted** inverted
Fan start delay		**0 s** [other values in the range 10 s … 12 min]
	This parameter is also available if the hot-start function is active (through measuring of the conveying fluid temperature at the battery for the thermal exchange). The function is active in both seasonal modes (heating and cooling).	
Fan stop delay		**0 s** [other values in the range 10 s … 12 min]
	The function allows prolonging the operation of the ventilator, dissipating in the room the residual heat or cool present in battery for the thermal exchange. The function is active in both seasonal modes (heating and cooling).	
Cyclic sending interval		**no sending** [other values in the range 30 s … 120 min]

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Fan continuous speed	Control type = continuous regulation	1 byte	CR-T-	[5.001] percentage (0..100%)	242
Fan speed 1	Control type = 1, 2 or 3 speeds	1 bit	CR-T-	[1.001] switch	243
Fan speed 2	Control type = 2 or 3 speeds	1 bit	CR-T-	[1.001] switch	244
Fan speed 3	Control type = 3 speeds	1 bit	CR-T-	[1.001] switch	245
Fan control disable	Disable ventilation from bus = yes	1 bit	C-W--	[1.002] boolean	246
Fan manual speed		1 byte	CRW-U	[5.010] counter pulses (0…255)	249
Fan speed		1 byte	CR-T-	[5.010] counter pulses (0…255)	250
Fan manual active status		1 bit	CRWT-	[1.011] state	251
Fan manual speed percentage		1 byte	CR-T-	[5.001] percentage	266
Fan manual speed off status		1 bit	CR-T-	[1.011] state	267

10.7.4.2 Delayed fan start ("hot-start")

This function is used in case the fan forces in the room air passing through a heat exchange coil (as in the case of the terminals to the fan-coil). In the heating mode of operation, to avoid possible discomfort caused by the dispatch of cold air in the room, the room temperature controller does not start the fan until the fluid has not reached a sufficiently high temperature. This situation normally occurs at the first start or after long periods of inactivity. The function can be carried out by:

1. a temperature control (through a temperature sensor on the coil exchange battery);
2. a delayed start (function approximated);

In the first case the temperature of the heat conveying fluid is acquired at the exchange battery. The function then has an effective temperature control, but for the execution is necessary that the heat exchange coil is equipped with a sensor of minimum water temperature that acquires the temperature of the heat conveying fluid.

The effectiveness of the function depends on a field measurement of the time actually required to have sufficiently warm air from the terminal.

10.7.4.3 Antistratification function

This function is used in the case of heating systems with thermal exchange of convective type for rooms with height and volume much higher than usual (atriums, fitness facility, commercial buildings, etc.). Because of

the natural convection - with warm air rising to the highest altitudes of the room - the phenomenon of air stratification occurs, with energy waste and discomfort for the occupants at the same time. The function opposes to the air stratification, forcing the warm air downwards.

The antistratification function requires:

- rooms of great height;
- availability of ventilation devices able to force the air movement downwards (opposed to the natural convective movement of warm air);
- measuring of the temperature at two heights through the installation of a second temperature sensor at an adequate height in order to measure the actual air stratification (the main room temperature controller is supposed to be installed at 1.5 m).

For rooms with ordinary height (2,70 ÷ 3,00 m) the DIN 1946 standard recommends not to exceed 2 K/m in order to have an adequate comfort; this gradient may be bigger in higher rooms.

10.7.4.4 2-stage configuration with fan-coils as auxiliary stage

The fan-coil units may be used both as a main stage and secondary stage. As main stage they can be combined only to radiators as auxiliary stage. If, however, the main stage is done with (floor or ceiling) radiant panels, the fan-coils can be used as auxiliary stage. In the latter case they work in automatic mode with a configurable offset with respect to the temperature setpoint for the main stage, and then carry out their compensation function while the main stage is brought in temperature with bigger inertia.

The *Ventilation* folder, that is unique, configures a main or a auxiliary stage depending on the settings chosen in the *Heating* and *Cooling* folders. Similarly, the display interface will act on manual / automatic and manual forcing of the only fan-coil.

A particular case occurs when a fan-coil unit works in a season as auxiliary stage and in the other one as main stage. It is for example the case of:

- a radiant panels system that works only for heating and has a fan-coil as auxiliary stage; the same fan-coil works as main stage for cooling;
- a radiator system that has a fan-coil as auxiliary stage for heating; the same fan coil unit functions as main stage for cooling.

In these cases with the configuration adopted, the following steps are necessary:

1. Settings $\Rightarrow$ Thermostat function = both heating and cooling. This configuration enables both folders (heating and cooling)
2. Heating $\Rightarrow$ Heating type = floor radiant panels or ceiling radiant panels
3. Heating $\Rightarrow$ Command communication object = separated (if unique is choosen, the parameter Cooling $\Rightarrow$ Cooling type does not appear)
4. Heating $\Rightarrow$ Auxiliary heating = enabled
5. Auxiliary heating $\Rightarrow$ Communication object = separated
6. Heating $\Rightarrow$ Ventilation for auxiliary heating = enabled
7. Cooling $\Rightarrow$ Cooling type = fancoils

> **i** **Important!** If the fan-coil system has a 2-pipe hydraulic configuration, the objects Auxiliary heating output command (1 bit) and Cooling out command (1 bit) have to be set in logical OR in the actuator for controlling the fan-coil which in this case is unique.

An alternative solution that avoids the setting of a logic OR can be realized by configuring a main stage for heating and cooling with radiant panels through separate valves and an auxiliary stage for heating and cooling fan coil through combined valves. The offset of the auxialiary stage for cooling is set to the value 0 (zero); this corresponds to a configuration for main stage. The object Cooling out command (1 byte) is not connected so that the radiant panel system works only for heating.

10.7.4.5 Remote fan speed modification

The communication objects shown in Figure 26 allow to monitor the actual fan speed forced automatically by the temperature controller or or by means of a supervisor system. The communication objects (from now on: C.O.) also allow to perform the same modifications remotely, for example from a supervisor software.

249 – Fan manual speed [5.010 counter pulses]

250 – Fan speed [5.010 counter pulses]

251 – Fan manual active status [1.011 state]
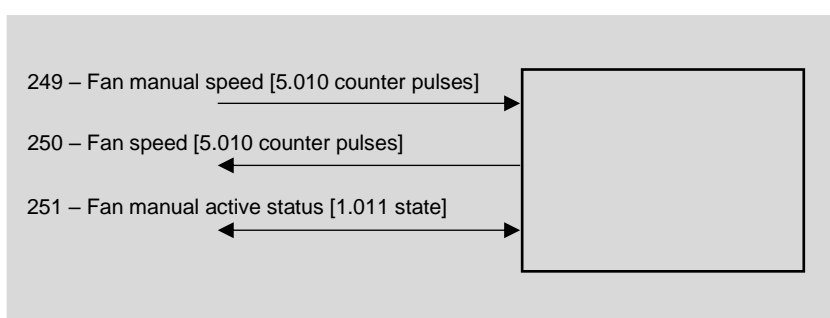
Figure 26 - Remote fan speed modification objects

The C.O. *250 – Fan speed* allows to evaluate the actual fan speed; the C.O. 251 – *Fan manual active status* contains the information about automatic (=0, not active) or manual (=1, active) operating mode. By modifying the C.O. 249 – *Fan manual speed*, the fan automatically switches to the setpoint speed; to return to automatic mode (A), the supervisor must exit from manual mode by modifying the C.O. 251 – *Fan manual active status* (=0, not active).

Accepted values for C.O.s 249, 250 and 251 depend on the number of speeds set in ETS for the Ventilation Control type.

If *Control Type* parameter in Ventilation folder is = 1, 2 or 3 speeds, C.O.s with DPT [5.010 counter pulses] accept the following values:

- = 0: OFF
- = 1: speed 1
- = 2: speed 2 (if *Control Type* > 1 speed)
- = 3: speed 3 (if *Control Type* > 2 speed)

If *Control Type* parameter in Ventilation folder is = continuous regulation, the values of the C.O.s with DPT [5.010 counter pulses] match the following percentage of the maximum speed:

- = 0: OFF
- = 1: 20%
- = 2: 40%
- = 3: 60%
- = 4: 80%
- = 5: 100%

10.7.5 Relative humidity control

The **Relative humidity control** folder includes the following secondary folders:

- Dehumidification
- Humidification
- Calculated psychrometric values

The secondary folders **Dehumidification, Humidification** and **Calculated psychrometric values** appear only if an external (from bus) humidity sensor is enabled.

The sensor acquires the air humidity value inside the room, which can be used for the following purposes:

- Sending over the bus (for information purpose) through DPT [9.007] percentage (%);
- Use of detected value for dew-point temperature calculations and sending on the bus through corresponding DPTs;
- Use for room ventilation through ventilation start, external intakes opening, window opening through motorized actuators. Control is performed upon thresholds;
- Use for control of thermoigrometric comfort conditions of radiant panel cooling systems equipped with integration of latent heat (starting of dedicated terminals without modification of cooling water flow temperature);
- Use for safety control in radiant panel cooling systems not equipped with integration of latent heat through calculation of critical thermoigrometric conditions (dew point) and corresponding modification of cooling water flow temperature.

Dehumidification

The secondary folder **Dehumidification**, when the related function is enabled, includes the following parameters:

- Operating modes where dehumidification is active
- Relative humidity setpoint for dehumidification control [%]
- Dehumidification control hysteresis [%]
- Cyclic sending interval
- Disable from bus
- Subordinated to temperature control
- Dehumidification start delay
- Dehumidification secondary to temperature control
- Temperature difference for integration
- Hysteresis for integration

10.7.5.1 Parameter and communication object tables

Parameter name	Conditions	Values
Dehumidification function	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling	**disabled** cooling heating both cooling and heating
	Temperature control ⇒ Settings ⇒ Thermostat function = heating	**disabled** / heating only

Parameter name	Conditions	Values
	Temperature control ⇒ Settings ⇒ Thermostat function = heating	**disabled** / cooling only
	Parameter that selects the dehumidification function.	
Humidity setpoint [%]	Dehumidification function ≠ disabled	**55** [range 20 … 80]
Humidity hysteresis	Dehumidification function ≠ disabled	**0,8 %** [other values in the range 0,5 … 4%]
Cyclic sending interval	Dehumidification function ≠ disabled	**no sending** [other values in the range 30 s … 120 min]
Disable dehumidification control from bus	Dehumidification function ≠ disabled	**no** / yes
Signal from bus	Disable dehumidification control from bus = yes	**not inverted** / inverted
Subordinated to temperature control	Temperature control ⇒ Settings ⇒ Thermostat function = cooling or both heating and cooling, Temperature control ⇒ cooling ⇒ cooling type = floor radiant panels or ceiling radiant panels, Relative humidity ⇒ dehumidification ⇒ dehumidification function = cooling only	no / **yes**
Dehumidification start delay	Subordinated to temperature control = no	**00:05:00 hh:mm:ss** [range 00:00:00 … 18:12:15]
	Value 00:00:00 means that the start delay is disabled.	
Integration		**no** / yes
Temperature difference for integration	Integration = yes	**1,5°C** [other values in the range 0,5 °C … 3 °C]
Hysteresis for integration	Integration = yes	**0,5 K** [other values in the range 0,2 K … 3 K]

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Relative humidity setpoint for dehumidification		2 bytes	CRWTU	[9.007] humidity (%)	255
Dehumidification command		1 bit	CR-T-	[1.001] switch	257
Dehumidification water battery command	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling,	1 bit	CR-T-	[1.001] switch	258

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
	Temperature control ⇒ cooling ⇒ cooling type = floor radiant panels or ceiling radiant panels, Relative humidity control ⇒ dehumidification ⇒ dehumidification function = cooling only				
Dehumidification integration control	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ cooling type = floor radiant panels or ceiling radiant panels, Relative humidity control ⇒ dehumidification ⇒ dehumidification function = cooling only Integration = yes	1 bit	CR-T-	[1.001] switch	259
	This object switches ON if (simultaneously) the relative humidity is greater than the relative humidity setpoint and the room temperature is greater than the setpoint of the parameter Temperature difference for integration.				
Dehumidification control disable	Disable dehumidification control from bus = yes	1 bit	C-W--	[1.002] boolean	260

Humidification

The secondary folder *Humidification* includes the following parameters:

- Operating modes where humidification is active
- Relative humidity setpoint for humidification control [%]
- Dehumidification control hysteresis [%]
- Cyclic sending interval
- Disable from bus

10.7.5.2 Parameter and communication object tables

Parameter name	Conditions	Values
Humidification function	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling	**disabled** cooling heating both cooling and heating
	Temperature control ⇒ Settings ⇒ Thermostat function = heating	**disabled** / heating only
	Temperature control ⇒ Settings ⇒ Thermostat function = heating	**disabled** / cooling only
	Parameter that selects the humidification function.	
Humidity setpoint	Humidification function ≠ disabled	**35** [range 20 … 80 %]
Humidity hysteresis [%]	Humidification function ≠ disabled	**0,8 %** [other values in the range 0,5 … 4%]
Cyclic sending interval	Humidification function ≠ disabled	**no sending** [other values in the range 30 s … 120 min]
Disable humidification control from bus	Humidification function ≠ disabled	**no** / yes
Signal from bus	Humidification function ≠ disabled Disable humidification control from bus = yes	**not inverted** / inverted

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Relative humidity setpoint for humidification	Humidification function ≠ disabled	2 bytes	CRWTU	[9.007] humidity (%)	256
Humidification command	Humidification function ≠ disabled	1 bit	CR-T-	[1.001] switch	261
Humidification control disable	Disable humidification control from bus = yes	1 bit	C-W--	[1.002] boolean	262

Calculated psychrometric values

The secondary folder *Calculated psychrometric values* includes the following parameters:

- Dew-point temperature
- Cyclic sending interval
- Min. change of value to send [K]

10.7.5.3 Parameter and communication object tables

Parameter name	Conditions	Values
Dew-point temperature		**disabled** / enabled
	The dew-point temperature, if sent on the bus, allows to implement an active anticondensation protection with recalibration of the flow conditions of the conveying fluid if each mixing group has its own control device. If the thermostat is installed in an environment where no air conditioning is foreseen (e.g. toilets), it is better to exclude that environment from the control by disabling the dew-point temperature parameter.	
Cyclic sending interval	Dew-point temperature = enabled	**no sending** [other values in the range 30 s … 120 min]
Min. change of value to send [K]	Dew-point temperature = enabled	**0,2 K** / no sending [other values in the range 0,2 … 3 K]

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Dew-point temperature	Dew-point temperature = enabled	2 bytes	CR-T-	[9.001] temperature °C	254

10.7.6 Energy saving

In order to realize energy-saving functions, window contacts (to detect the opening of windows or doors), presence and movement sensors and card holders can be used.

The *Energy saving* folder includes the following secondary folders:

- Window contacts
- Presence sensors
- Card holder

The folder is available if the following conditions are satisfied:

- *Internal sensors ⇒ Temperature sensor = enabled, or*
- *External sensors (from bus) ⇒ Room Temperature sensor = enabled.*

Window contacts

The *Window contacts* secondary folder appears if at least one sensor dedicated to this function is enabled i.e. if the following condition is verified:

- *External sensors (from bus) ⇒ Windows contact sensor 1 and/or 2 = enabled.*

The *Window contacts* folder includes the following parameters:

- Window contacts function
- Wait time to building protection mode

10.7.6.1 Parameter and communication object tables

Parameter name	Conditions	Values
Window contacts function		**disabled** / enabled
	This parameter enables the window contact function.	
Wait time to building protection mode	Window contacts function = enabled	**00:01:00 hh:mm:ss** [range 00:00:00 … 18:12:15]
	Time interval before the automatic switching of the device to the Building protection operating mode	

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Windows contact sensor 1 (from bus)	Window contacts function = enabled	1 bit	C-WTU	[1.019] window/door	211
Windows contact sensor 2 (from bus)	Window contacts function = enabled	1 bit	C-WTU	[1.019] window/door	212

Presence sensors

The *Presence sensors* folder includes the following parameters:

- Presence sensors function
- Presence sensors use
- Thermostat modes
- Absence time to switch HVAC mode

For this function external sensors (from bus) can be used, such as the ekinex EK-SM2-TP movement sensor or the ekinex EK-DX2-TP (X = B, C, D, E) presence sensor. The following condition has to be true:

○ *External sensors (from bus) ⇒ Presence sensor 1 and/or Presence sensor 2 = enabled*

10.7.6.2 Parameter and communication object tables

Parameter name	Conditions	Values
Presence sensors function		**disabled** / enabled
	Parameter that enables the presence sensor function.	
Presence sensors use	Presence sensor function = enabled	**comfort extension** comfort limitation comfort extension and comfort limitation
Thermostat modes	Presence sensor function = enabled, Presence sensors use = comfort extension and comfort limitation, or comfort limitation	**comfort-standby** comfort-economy
Absence time to switch HVAC mode	Presence sensor function = enabled	**00:01:00 hh:mm:ss** [range 00:00:00 … 18:12:15]
	Time interval before the automatic switching of the operating mode set in the Thermostat modes parameter.	

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Presence sensor 1 (from bus)	Presence sensor function= enabled	1 bit	C-WTU	[1.018] occupancy	213
Presence sensor 2 (from bus)	Presence sensor function = enabled	1 bit	C-WTU	[1.018] occupancy	214

Card holder

The **Card holder** secondary folder appears only if the corresponding sensor is enabled, i.e. if the following condition is true:

- ○ *External sensors (from bus) ⇒ Card holder contact = enabled*

The **Card holder** folder includes the following parameters:

- Card holder function
- On card insertion switch HVAC mode to
- Activation delay on card insertion
- On card removal switch HVAC mode to
- Activation delay on card removal

10.7.6.3 Parameter and communication object tables

Parameter name	Conditions	Values
Card holder function		**disabled** / enabled
	Parameter that enables the card holder function.	
On card insertion switch HVAC mode to (*)	Card holder function = enabled	none **comfort** standby economy
	This parameter defines to which operating mode the device should automatically switch, after inserting the card into the holder. *(*) Note: if Setpoint type is set to **single** in the "Temperature control" tab, this parameter is set to "none", since the operating modes are not managed.*	
Activation delay on card insertion	Card holder function = enabled	**00:00:00 hh:mm:ss** [range 00:00:00 … 18:12:15]
	Time interval before the automatic switching to the new operating mode, after inserting the card into the holder.	
On card removal switch HVAC mode to (*)	Card holder function = enabled	none **standby** economy building protection
	This parameter defines to which operating mode the device should automatically switch, after removing the card from the holder. **) Note: if Setpoint type is set to **single** in the "Temperature control" tab, this parameter is set to "**Building protection**", since the operating modes are not managed.*	
Activation delay on card removal	Card holder function = enabled	**00:00:00 hh:mm:ss** [range 00:00:00 … 18:12:15]
	Time interval before the automatic switching to the new operating mode, after removing the card from the holder.	

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Contact of card holder (from bus)	Card holder function = enabled	1 bit	C-WTU	[1.018] occupancy	215

Note on card holder function

The information of card insertion/removal in/from a card holder allows you to directly control the temperature by means of the room thermostat, while sending the object value on the bus allows you to control other room functions with KNX (lighting, electrical loads, feedback status for the hotel reception, etc.) depending on the configuration done with ETS. The value of the setpoint temperature and the switching have to be defined with the hotel responsible in accordance with the target of energy saving and level of service to be offered to the guests.

Conventional (not KNX) card holder

With a conventional card holder the status (card present or absent) of a signal contact is detected through an input of the device configured as *[DI] card holder contact sensor*. This way you can detect only the insertion and extraction of the card, but it cannot be detected e.g. the access of users with different profiles (guests, service staff, maintenance workforce).

KNX card holder

With a KNX card holder you can differentiate the switching to be carried out; this is not resolved by the parameters of the room temperature controller, but through the definition of scenes that are received by the device. Depending on the available device, advanced functions are possible (e.g. different user profiles).

10.8   Logic functions

10.8.1   General information

The pushbutton interfaces EK-E20-TP-… and EK-E20-TP-…-R allow to use some useful logic functions (AND, OR, NOT and exclusive OR) in order to implement complex functions in the building automation system.

It is possible to configure up to:

- 8 logical functions
- 4 inputs for each channel

Each object value, if desired, can be individually inverted by inserting a NOT logic operator.

For each function, a parameter *Delay after bus voltage recovery* is available: this parameter represents the time interval between the bus voltage recovery and the first reading of the input communication objects for evaluating the logic functions.

> **i** In case of uncorrect connection of the input communication object or electrical trouble on bus resulting in a failed input reading request, the logic output of the corresponding channel can be calculated by setting the input values to default.

The communicaton function representing the logic function output is sent on the bus on event of change; alternatively, a cyclic sending can be set.

10.8.1.1   Parameter and communication object tables

The following condition has to be true: *General ⇒ Logic functions* = enabled.

Parameter name	Conditions	Values
Logic function		**disabled** / enabled
Logic operation	Logic function = enabled	**OR** / AND / XOR
	XOR (eXclusive OR)	
Delay after bus voltage recovery	Logic function = enabled	**00:00:04.000 hh:mm:ss.fff** [range 00:00:00.000 … 00:10:55.350]
	Time interval between the bus voltage recovery and the first reading of the input communication objects for evaluating the logic functions	
Output cyclic sending interval	Logic function = enabled	**no sending** [other value in range 30 s … 120 min]
	No sending means that the output state of the logic function is updated on the bus only on change. Different values imply cyclic sending on the bus of the output state.	
Output send	Logic function = enabled	**both values** only value 0 only value 1
Output updating	Logic function = enabled	**on value change** on change of value or input
Logic object x	Logic function = enabled	**disabled** / enabled

Parameter name	Conditions	Values
	x = 1, 2, 3, 4	
Logic object x - Negated	Logic function = enabled Logic object x = enabled	**no** / yes
	Negando lo stato logico dell'ingresso corrispondente, è possibile realizzare logiche combinatorie articolate. Esempio: Output=(NOT(Oggetto logico 1) OR Oggetto logico 2). x = 1, 2, 3, 4	
Logic object x - Read at startup	Logic function = enabled Logic object x = enabled	**no** / yes
	x = 1, 2, 3, 4	
Logic object x - Default value	Logic function = enabled Logic object x = enabled	**none** / off / on
	x = 1, 2, 3, 4	

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Logic function X – Input 1	Logic function X = enabled Logic object 1 = enabled	1 Bit	C-W--	[1.001] switch	156, 161, 166, 171, 176, 181, 186, 191
	x = 1, 2, 3, 4				
Logic function X – Input 2	Logic function X = enabled Logic object 2 = enabled	1 Bit	C-W--	[1.001] switch	157, 162, 167, 172, 177, 182, 187, 192
	x = 1, 2, 3, 4				
Logic function X – Input 3	Logic function X = enabled Logic object 3 = enabled	1 Bit	C-W--	[1.001] switch	158, 163, 168, 173, 178, 183, 188, 193
	x = 1, 2, 3, 4				
Logic function X – Input 4	Logic function X = enabled Logic object 4 = enabled	1 Bit	C-W--	[1.001] switch	159, 164, 169, 174, 179, 184, 189, 194
	x = 1, 2, 3, 4				
Logic function X – Output	Logic function X = enabled At least one Logic object = enabled	1 Bit	C-W--	[1.001] switch	160, 165, 170, 175, 180, 185, 190, 195
	x = 1, 2, 3, 4				

11 Appendix

11.1 Summary of KNX communication objects

The following list contains the KNX communication objects for all corresponding *Data Point Types* (DPT) defined by the application program according to the performed configurations.

The list is ordered by object number; if the same object is linked to different inputs, the first input or rocker is referenced.

Object name	Size	Flags	DPT	No. Comm. Obj.
Technical alarm	1 Bit	C-W--	[1.5] DPT_Alarm	1
Max LEDs intensity percentage	2 Byte	CR-T-	[5.1] DPT_Scaling	2
Presence sensor	1 bit	C-WTU	[1.17] DPT_Trigger	3
Button x left / right – Lock function	1 Bit	C-W--	[1.3] DPT_Enable	4, 20, 36, 52, 68, 84, 100, 116
Button x left / right – Switching status *[type]*, object n *	*See Table 8*	C-WTU	*See Table 8*	5, …, 12 21, …, 28 37, …, 44 53,…, 60 69, …, 76 85, …, 92 101, …, 108 117, …, 124
	* The numbers of listed C.O.s are referred to the first of 8 objects (for each input); the next C.O.s are sequential. In order to obtain the C.O. number for the n-th object, simply add (n-1) to the referred numbers. *E.g.: he C.O.s linked to button 3 have numbers starting from 37. The number of the 5th C.O. linked to that input will be 37 + (5-1) = 41.*			
Button x left / right – Switching command	1 Bit	C-WTU	[1.1] DPT_Switch	13, 29, 45, 61, 77, 93, 109, 125
Button x left / right – Dedicated stop command	1 Bit	C--T-	[1.17] DPT_Trigger	13, 29, 45, 61, 77, 93, 109, 125
Button x left / right – Dimming up / down / stop command	4 Bit	CR-T-	[3.*] DPT_Control_Dimming, DPT_Control_Blinds	14, 30, 46, 62, 78, 94, 110, 126
Button x left / right – Stop – step up/down command	1 Bit	C--T-	[1.7] DPT_Step	15, 31, 47, 63, 79, 94, 111, 127
Button x left / right – Move up / down command	1 Bit	C--T-	[1.8] DPT_UpDown	16, 32, 48, 64, 80, 95, 112, 128
Button x left / right – Scene number	1 Byte	C--T-	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	17, 33, 49, 65, 81, 96, 113, 129
Proximity – Lock command	1 bit	C-W--	[1.3] DPT_Enable	132
Proximity – Switching status *[type]*, object n (n = 1,…,8)	*See Table 8*	C-WTU	*See Table 8*	133, 134, 135, 136, 137, 138, 139, 140
LED x left / right – Command 1 bit	1 bit	C-WTU	[1.1] DPT_Switch	148, 149, 150, 151, 152, 153, 154, 155

Object name	Size	Flags	DPT	No. Comm. Obj.
Logic function x – Input 1 (x = 1,…,4)	1 bit	C-WTU	[1.1] DPT_Switch	156, 161, 166, 171, 176, 181, 186, 191
Logic function x – Input 2 (x = 1,…,4)	1 bit	C-WTU	[1.1] DPT_Switch	157, 162, 167, 172, 177, 182, 187, 192
Logic function x – Input 3 (x = 1,…,4)	1 bit	C-WTU	[1.1] DPT_Switch	158, 163, 168, 173, 178, 183, 188, 193
Logic function x – Input 4 (x = 1,…,4)	1 bit	C-WTU	[1.1] DPT_Switch	159, 164, 169, 174, 179, 184, 189, 194
Logic function x – Output (x = 1,…,4)	1 bit	CR-T-	[1.1] DPT_Switch	160, 165, 170, 175, 180, 185, 190, 195
Temperature value	2 Byte	CR-T-	[9.1] DPT_Value_Temp	196
Temperature threshold 1 - Switch	1 Bit	CR-T-	[1.1] DPT_Switch	197
Temperature threshold 1 - Lock	1 bit	C-W--	[1.1] DPT_Switch	198
Temperature threshold 1 – Value (from bus)	2 bytes	C-W--	[9.1] DPT_Value_Temp	199
Temperature threshold 2 - Switch	1 Bit	CR-T-	[1.1] DPT_Switch	200
Temperature threshold 2 - Lock	1 bit	C-W--	[1.1] DPT_Switch	201
Temperature threshold 2 – Value (from bus)	2 bytes	C-W--	[9.1] DPT_Value_Temp	202
Room temperature (from bus)	2 bytes	C-WTU	[9.1] DPT_Value_Temp	203
Humidity (1 byte, from bus)	2 bytes	C-WTU	[9.7] DPT_Value_Humidity	204
Humidity (2 bytes, from bus)	1 byte	C-WTU	[5.1] DPT_Scaling	205
Antistratification temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	206
Outdoor temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	207
Coil temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	208
Floor temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	209
Flow temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	210
Windows contact sensor 1 (from bus)	2 bytes	C-WTU	[1.019] DPT_Window_Door	211
Windows contact sensor 2 (from bus)	2 bytes	C-WTU	[1.019] DPT_Window_Door	212
Presence sensor 1 (from bus)	1 bit	C-WTU	[1.018] DPT_Occupancy	213
Presence sensor 2 (from bus)	1 bit	C-WTU	[1.018] DPT_Occupancy	214
Contact of card holder (from bus)	1 bit	C-WTU	[1.018] DPT_Occupancy	215
Anticondensation (from bus)	1 bit	C-WTU	[1.001] DPT_Switch	216
Weighted temperature	2 bytes	CR-T-	[9.001] DPT_Value_Temp	271
Heating/cooling status out	1 bit	CR-T-	[1.100] DPT_Heat_Cool	218
Heating/cooling status in	1 bit	C-W--	[1.100] DPT_Heat_Cool	219
HVAC mode in	1 byte	C-W--	[20.102] DPT_HVACMode	220
HVAC forced mode in	1 byte	C-W--	[20.102] DPT_HVACMode	221
HVAC mode out	1 byte	CR-T-	[20.102] DPT_HVACMode	222
HVAC manual mode	1 byte	C-WTU	[20.102] DPT_HVACMode	223
Chrono active status	1 bit	CR-T-	[1.11] DPT_State	224
Actual setpoint	2 bytes	CR-T-	[9.1] DPT_Value_Temp	225
Manual setpoint	2 bytes	C-W--	[9.1] DPT_Value_Temp	226

Object name	Size	Flags	DPT	No. Comm. Obj.
Input setpoint	2 bytes	CRWTU	[9.1] DPT_Value_Temp	227
Comfort setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	227
Comfort setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	228
Standby setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	229
Offset standby (heating)	2 bytes	CRWTU	[9.2] DPT_Value_Tempd	229
Standby setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	230
Offset standby (cooling)	2 bytes	CRWTU	[9.2] DPT_Value_Tempd	230
Economy setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	231
Offset economy (heating)	2 bytes	CRWTU	[9.2] DPT_Value_Tempd	231
Economy setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	232
Offset economy (cooling)	2 bytes	CRWTU	[9.2] DPT_Value_Tempd	232
Building protection setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	233
Building protection setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	234
Room temperature controller status	1 bit	CR-T-	[1.3] DPT_Enable	235
Heating out command	1 bit	CR-T-	[1.1] DPT_Switch	236
Heating out command	1 byte	CR-T-	[5.1] DPT_Scaling	236
Heating out command	1 bit	CR-T-	[1.1] DPT_Switch	236
Heating and cooling out command	1 byte	CR-T-	[5.1] DPT_Scaling	236
Cooling out command	1 bit	CR-T-	[1.1] DPT_Switch	237
Cooling out command	1 byte	CR-T-	[5.1] DPT_Scaling	237
Auxiliary heating output command	1 bit	CR-T-	[1.1] DPT_Switch	238
Auxiliary heating and cooling output command	1 bit	CR-T-	[1.1] DPT_Switch	238
Auxiliary cooling output command	1 bit	CR-T-	[1.1] DPT_Switch	239
Auxiliary heating disable	1 bit	C-W--	[1.3] DPT_Enable	240
Auxiliary cooling disable	1 bit	C-W--	[1.3] DPT_Enable	241
Fan continuous speed	1 byte	CR-T-	[5.1] DPT_Scaling	242
Fan speed 1	1 bit	CR-T-	[1.1] DPT_Switch	243
Fan speed 2	1 bit	CR-T-	[1.1] DPT_Switch	244
Fan speed 3	1 bit	CR-T-	[1.1] DPT_Switch	245
Fan control disable	1 bit	C-W--	[1.2] DPT_Bool	246
Thermostat – Alarm text	14 bytes	CR-T-	[16.0] DPT_String_ASCII	247
Manual setpoint active status	1 bit	CRWTU	[1.11] DPT_State	248
Fan manual speed	1 byte	CRW-U	[5.10] DPT_Value_1_Ucount	249
Fan speed	1 byte	CR-T-	[5.10] DPT_Value_1_Ucount	250
Fan manual active status	1 bit	CRWT-	[1.11] DPT_State	251
Room temperature control alarm	1 bit	CR-T-	[1.5] DPT_Alarm	252
Disable room temperature controller	1 bit	C-W--	[1.1] DPT_Switch	253
Dew-point temperature	2 bytes	CR-T-	[9.1] DPT_Value_Temp	254
Relative humidity setpoint for dehumidification	2 bytes	CRWTU	[9.7] DPT_Value_Humidity	255
Relative humidity setpoint for humidification	2 bytes	CRWTU	[9.7] DPT_Value_Humidity	256

Object name	Size	Flags	DPT	No. Comm. Obj.
Dehumidification command	1 bit	CR-T-	[1.1] DPT_Switch	257
Dehumidification water battery command	1 bit	CR-T-	[1.1] DPT_Switch	258
Dehumidification integration control	1 bit	CR-T-	[1.1] DPT_Switch	259
Dehumidification control disable	1 bit	C-W--	[1.2] DPT_Bool	260
Humidification command	1 bit	CR-T-	[1.1] DPT_Switch	261
Humidification control disable	1 bit	C-W--	[1.2] DPT_Bool	262
Anticondensation alarm	1 bit	CR-T-	[1.5] DPT_Alarm	263
Thermal generator lock	1 bit	C-W--	[1.5] DPT_Alarm	264
Building protection HVAC mode active	1 bit	CR-T-	[1.11] DPT_State	265
Fan manual speed percentage	1 byte	CR-T-	[5.1] DPT_Scaling	266
Fan manual speed off status	1 bit	CR-T-	[1.11] DPT_State	267

Size	DPT
1 bit	[1.001] switch
2 bit	[2.*] 1-bit controlled
1 byte unsigned value	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte percentage	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte signed value	[6.*] 8-bit signed value
2 bytes unsigned value	[7.*] 2-byte unsigned value
2 bytes signed value	[8.*] 2-byte signed value
2 bytes floating value	[9.*] 2-byte float value

Table 8 - Size e DPT for C.O.s with independent inputs

11.2 Diagnostic table

Alarm code	Message
A01	Room controller alarm
A02	Thermal generator lock
A03	internal temperature sensor alarm
A04	Floor surface temperature sensor alarm
A05	Anticondensation sensor alarm
Error code	
E01	Temperature sensor failure
E02	Humidity sensor failure
E03	Antistratification temperature sensor failure
E04	Outdoor temperature sensor failure
E05	Coil temperature sensor failure
E06	Floor surface temperature sensor failure
E07	Flow temperature sensor failure
Analogic timeout	
W01	Analogic temperature sensor timeout
W02	Analogic humidity sensor timeout
W03	Analogic antistratification temperature sensor timeout
W04	Analogic outdoor temperature sensor timeout
W05	Analogic coild temperature sensor timeout
W06	Analogic floor surface temperature sensor timeout
W07	Analogic flow temperature sensor timeout
Digital timeout	
W09	Digital anticondensation sensor timeout
W10	Digital window contact 1 timeout
W11	Digital window contact 2 timeout
W12	Digital presence sensor 1 timeout
W13	Digital presence sensor 2 timeout
W14	Digital card holder contact timeout

Table 9 - Diagnostic messages

11.3 Warning

- Installation, electrical connection, configuration and commissioning of the device can only be carried out by qualified personnel.
- Opening the housing of the device causes the immediate end of the warranty period.
- ekinex® KNX defective devices must be returned to the manufacturer at the following address:

EKINEX S.p.A. Via Novara 37, I-28010 Vaprio d'Agogna (NO) Italy.

11.4 Other information

- This application manual is aimed at installers, system integrators and planners
- For further information on the product, please contact the ekinex® technical support at the e-mail address: support@ekinex.com or visit the website www.ekinex.com
- KNX® and ETS® are registered trademarks of KNX Association cvba, Brussels

© EKINEX S.p.A. The company reserves the right to make changes to this documentation without notice.