ekinex

CONTROL YOUR LIVING SPACE

Application manual wall mounting KNX virtual Pushbuttons with touch display EK-EV2-TP SIGNUM

Sommario

Revision	Modifications	Date
1.0	First emission	12/09/2022
1.1	Modifications after the first ETS revision	22/11/2022

1 Scope of the document

This application manual describes application details for ekinex® virtual pushbutton with touch display EK-EV2-TP.

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	STEKEV2TP_IT.pdf	EK-EV2-TP	A1.0	11 / 2022
Application manual	MAEKEV2TP_IT.pdf	EK-EV2-TP		
Application program	APEKEV2TP##.knxprod	EK-EV2-TP		

The *Ekinex Tool* app, available for Apple and Android mobile devices, can be downloaded directly from their respective stores.

Download on the App Store

ANDROID APP ON Google play

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

KNX virtual pushbutton with touch display EK-EV2-TP SIGNUM

2 Product description

The EK-EV2-TP touch-display pushbutton control from the ekinex® SIGNUM series is a KNX S-mode wall-mounted device that is characterised by its ease of use and elegant, discreet appearance. The product is suitable for the on/off control of utilities, the dimming of luminaires, the control of motorised drives and other programmable command and control functions, such as the room temperature control graphical page and the public address device control graphical page.

The integrated temperature sensor also allows it to be used as a temperature controller for a room or zone. The device, designed for KNX-standard home and building automation systems, is equipped with a KNX bus communication module and is designed for mounting on a round or square wall flush-mounting box, fitted with fixing holes with 60 mm spacing. The button is to be completed with a plate from the Deep series.

The borderless IPS touch display, with dimensions of 2.65", 162 x 320 pixels with 262K colours and haptic feedback, can be configured to manage up to 4 buttons per screen and up to 3 screens in total. The IPS (In Plane Switching) technology adopted for the display allows wide viewing angles and high fidelity in the colour reproduction of themes and graphic elements. SIGNUM is capable of satisfying multiple aesthetic installation requirements and can be mounted indifferently with alignment of the IPS touch display on the left or right side, in front view.

Each button can be customised via a dedicated Ekinex Tool app via BLE (Bluetooth Low Energy), and certain attributes of each button can also be changed after installation, e.g. symbols and lettering. The proximity sensor integrated in the faceplate (to be ordered separately) enables the activation of backlighting and other functions, ensures maximum readability when using the button panel, and provides visual comfort at night.

The device is powered at SELV 30 Vdc via the KNX bus and does not require an auxiliary power supply, either for normal bus communication or during Bluetooth Low Energy wireless communication with the mobile app.

The supply includes, inside the box:

- 1 metal bracket for flush-mounting box (Code EK-SMQ-71-S)
- 2 pairs of fixing screws;
- 1 KNX terminal block for the connection of the bus line.

Mounting information

The screws supplied in the package are suitable for standard installations. For particular applications, in which the screws must be replaced, these must be of the flat-head type.

The supplied plastic screws (# 2) must only be used to hold the push-button panel in position, therefore they must not be tightened with excessive force (max. torque 0.4 Nm).

The screws for the metal support must be tightened with a max. torque of 1.0 Nm.

2.1 Additional codes

The device is completed with an ekinex® plate in metal (aluminium), Metal HT (chrome) or Fenix NTM®, with a 30 x 60 mm window and integrated proximity sensor. The plate is available for Deep series mounting.

Single square plate with proximity sensor with 30 x 60 mm window, part number EK-DQV-... (Deep)

(The code is to be completed with the extension for colour and finish).

Series	Window	Color	Finish	Part-number
Signum METAL (aluminium)	30 x 60 mm	aluminium	brushed	EK-DQV-GBQ
		nichel		EK-DQV-GBR
		titanium		EK-DQV-GBS
		carbon		EK-DQV-GBU
		brass		EK-DQV-GBB

Series	Window	Color	Finish	Part-number
Signum METAL HT	30 x 60 mm	chrome	chrome-plated	EK-DQV-CRO
		black metal		EK-DQV-CBM
		copper		EK-DQV-COP
		champagne		EK-DQV-CHA

Series	Window	Color	Finish
Signum **FENIX NTM®**	30 x 60 mm	malè white	EK-DQV-FBM
		luxor beige	EK-DQV-FBL
		castoro ottawa	EK-DQV-FCO
		cacao orinoco	EK-DQV-FCC
		efeso gray	EK-DQV-FGE
		bromine gray	EK-DQV-FGL
		london gray	EK-DQV-FGB
		comodoro green	EK-DQV-FVC
		ingo black	EK-DQV-FNI

i The plate to complete the luminaire must be ordered separately. For more information on available materials, colours and finishes, see the ekinex® product catalogue or go to www.ekinex.com.

3 Switching, display and connection elements

The device has a IPS touch display for activating the hand control functions and a proximity sensor integrated in the front cover for activating the backlight and other programmable functions.

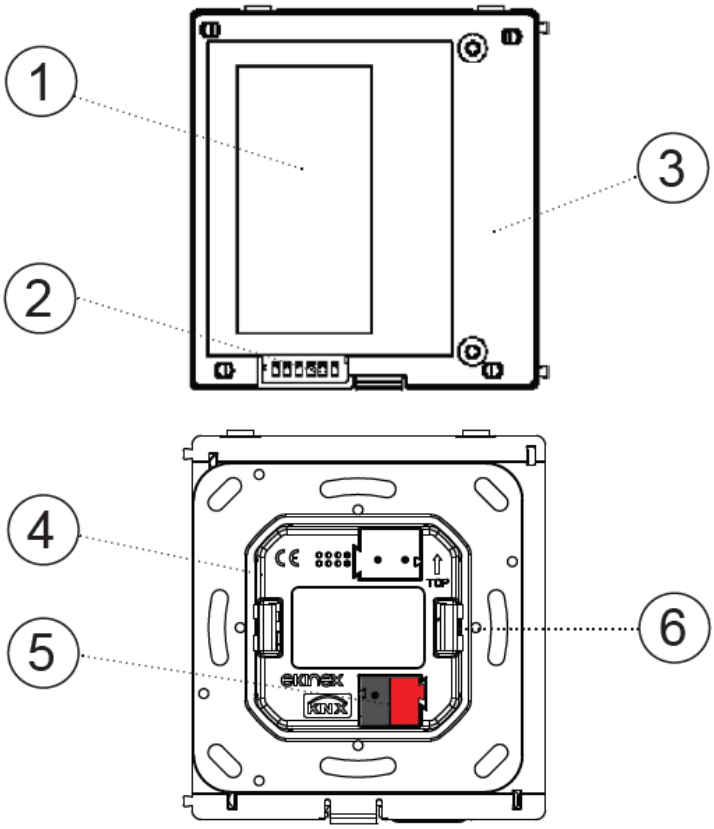


Fig. 2A - Elementi di commutazione e connessione EK-EV2-TP

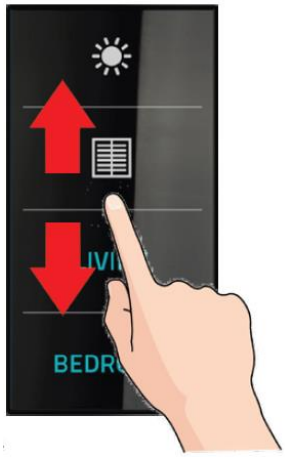
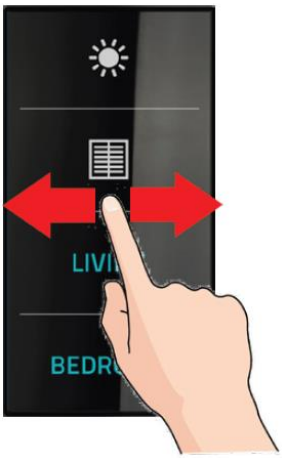
1) IPS touch display
2) Proximity sensor connection
3) Temperature sensor position (under cover)
4) Product label
5) KNX bus line connection terminal
6) Fixing springs

Switching elements

The touch display (1) can implement up to 12 buttons arranged on 3 pages, with a dedicated page for thermoregulation. The methods of interaction are as follows:

- touch or tap: pressing the area of a key with the fin¬ger for at least 15ms starts an activation or deactivation event;

- swipe: dragging the finger vertically from one side of the display to scroll through the function pages, while dragging it horizontally displays the settings and information pages;
- long press: prolonged touch (settable via ETS) of an area of the display activates the slider functions, such as light dimming or control of motor drives.

		
Touch / tap Switching On/off, sending values or sequences, scenes **Long press** Slider activation for dimmer, up / down, modification of thermostat setpoint	**Vertical swipe** Scrolling of function pages only	**Horizontal swipe** Scroll through the pages of functions, info / programming / reset and settings

The SIGNUM virtual pushbutton control is able to meet several aesthetic installation requirements and can be mounted indifferently with alignment of the IPS touch display on the left or right side, in front view. The different alignment of the display is achieved by rotating both the button and the plate by 180°. In this case, the proximity sensor and its slot will be located at the top of the button panel, instead of at the bottom.

The different physical mounting must be accompanied by the correct configuration of the parameter Product code (left or right) in the General tab of the ETS application.

4 Configuration

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

In order to configure the device, the ETS5 development tool (or later versions) and the ekinex® application programme dedicated to the device (the name is APEKEV2TPxx.knxprod) are required; these can be downloaded from the ekinex website www.ekinex.com.

The application programme allows access, within the ETS5 environment, to the configuration of all the working parameters of the device. The programme must be loaded into ETS (alternatively, the entire ekinex® product database can be loaded in a single operation), after which all device specimens of the type in question 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 then take place in the programming phase, described in the next paragraph.

Product code	EAN	Input Nr.	Application program ETS (## = revision)	Communication objects (Nr. max)	Group addresses (Nr. max)
EK-EV2-TP			APEKEV2TP##.knxprod	347	254

The configuration and programming of KNX devices requires specific knowledge; to acquire this knowledge, it is recommended to attend the appropriate training courses at a KNX consortium-certified centre.

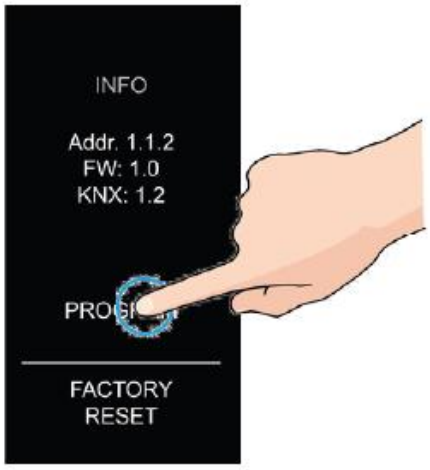
For further information, please visit www.knx.org.

After the device has been programmed for the first time, via the ETS5, it is possible to change some parameters via the *Ekinex Tool* mobile app, through the Bluetooth LE access point provided by the device.

5 Commissioning

For the commissioning of SIGNUM, the following activities are required:

- make the electrical connections as described above;
- turn on the KNX bus power supply;
- switch the device operation to the programming mode by scrolling the display pages to the INFO page and pressing the PROGRAM virtual button for 5s, which will turn red;
- download the physical address and configuration into the device using the ETS® program.



At the end of the download, operation of the device automatically returns to normal mode, displaying the first page of virtual buttons; SIGNUM is programmed and ready for operation.

Resetting the device

To reset the device, scroll through the pages of the display to the INFO page and press the virtual button marked FACTORY RESET for a few seconds until the brightness of the page dims significantly. At this point:

- if the PIN is enabled, the display shows a virtual keyboard for entry. The PIN for the FACTORY RESET is the same used for the connection from the App;
- if the PIN is disabled, the reset takes place directly.

The display will show the start-up page and the reset has been carried out. At this point, the addressing and configuration of the device via ETS® must be carried out again.

The reset resets the device to the factory delivery state. The addressing and parameter value set during configuration will be lost.

6 Display widgets

SIGNUM features a user interface designed to facilitate end-user use and consists of a number of pages and graphic components that can be configured via the ETS (and subsequently customised via the Ekinex Tool app). Depending on the control functions activated, the graphic components illustrated below are available on the device pages (assembly with display alignment to the left).

		
Page 1 virtual button It is possible to organise the buttons for each individual screen in 3 different ways: single button, double button and quadruple button.	**Page 2 virtual buttons** To best reflect every possible room structure, each button can be customised with text, symbols and colours.	**Page 4 virtual buttons** Buttons can be used to control lighting devices, scenarios and motorised drives. A long press can call up graphic pages for precise control of home automation elements.
		
Dimming page The slider allows precision control of dimmed actuators for lighting or motorised drives.	**Colour temperature adjustment page** The slider can be used to dim a lighting actuator from a cool to a warmer colour.	**Motorisation control page** The motorisation page allows you to control motorised locking elements using customised buttons with up and down symbols.
		
Room thermostat page It is possible to configure a simple single setpoint controller or a complete controller with support for HVAC operating modes. Manual/automatic speed control of a fan-coil is available.		

7 Connecting Ekinex Tool app

The user can customise certain functions and attributes of SIGNUM via the *Ekinex Tool* app, which is available for Android and iOS systems and can be downloaded via digital stores.

The setup procedure requires the Bluetooth connection of the device (smartphone or tablet). If it is deactivated, the app will ask for its activation through the settings options.

The activities to be performed to connect to the app are as follows:

- scroll to the SETTING MODE page on the SIGNUM display; the device is now set to perform the pairing operation via Bluetooth;
- download and install the “Ekinex Tool” app, possibly framing the QR code with the smartphone;
- at the first start, follow the steps of the app wizard to add a new device, by pressing on the virtual symbol “+”; then, by pressing on “SIGNUM”, the configuration starts;
- End the procedure by following the steps, as described in the Ekinex Tool app guide.

The main SIGNUM settings that can be customised via the *Ekinex Tool* app are:

- The background theme (Flat or Gradient) of the IPS display
- The texts associated to the virtual buttons with the intended use
- The icons associated to the virtual buttons, accessing a library now consisting of over 120 symbols
- The brightness of the display
- The intervention of the display haptic feedback
- The delay for inactivity before the display switches to standby mode

8 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 V 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.

8.1 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.

8.2 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.

8.3 Software working cycle

The main purpose of the software is following:

- Handle user virtual pushbutton presses and generate bus telegrams according to the assigned functions;
- Implement virtual 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.

8.4 Inputs commands

The touch display can represent up to 12 buttons arranged on up to 3 graphic pages; each graphic page can represent 1-2 or 4 buttons.

Two graphic themes are also available for button rendering: a linear rendering and a gradient rendering.

Left-hand display alignment			
Right-hand display alignment			

The 'tap' gesture on a button can be associated with different effects on a status variable.

8.4.1 Input events

A virtual button tap 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).

8.4.2 Lock function

For each input 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.

8.4.3 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.

8.4.4 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.

8.4.5 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.

The buttons can be configured independently or individually, each operating independently and possessing its own parameters and communication objects. It is not possible to configure buttons in coupled mode.

8.4.6 Single input command

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 function is only activated with long press events: a long press automatically opens the control widget.
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 function is only activated with long press events: a long press automatically opens the control widget.
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.

8.5 Proximity sensor

The proximity sensor is an active sensor with TOF (Time Of Flight) technology, calibrated to a detection distance of approximately 1 metre. Using this sensor, it is possible to keep the virtual pushbutton in a stand-by state by setting a maximum brightness level of display and only reactivate the backlighting of the virtual buttons when the user approaches the device. The sensitivity of the sensor is configurable by the ETS application, to avoid false detections.

In addition, it is possible to activate the sending of values or sequences when the sensor detects a user approaching within range, so as to trigger automations or signal the approach of a user. However, the proximity sensor is not a presence sensor and as such cannot be used to monitor the presence of people inside rooms and other closed places.

For correct operation of the proximity sensor, we recommend positioning the hand control at a distance from heat sources or draughts (air conditioners, fan coils, doors, windows).

8.6 Temperature sensor

The value from the embedded temperature sensor, unless it is disabled, can be read from the bus by other devices.

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.

8.7 Room controller

The SIGNUM push-button control offers 2 ways of managing the single-zone room thermostat function:

- with communication objects, without user interface: SIGNUM is used as a push-button control of the 'FF series, '71 series or Venti20 series, the user can interact with the controller via a supervisor such as ekinex® delégo, using the dedicated app for mobile devices provided.
- with communication objects and with a dedicated graphic component: in this configuration, SIGNUM presents a page with a graphic component dedicated to individual zone control.

8.7.1 SIGNUM thermostat widget

Depending on the requirements in terms of complexity and completeness of the information to be transferred to the customer, the system integrator can opt for one of 2 possible modes of representation of the graphic component:

- Single setpoint mode
- Absolute setpoint mode

The setpoint type and consequently the type of representation of the graphic component is chosen in the ETS application, in the *Settings* tab of the thermostat.

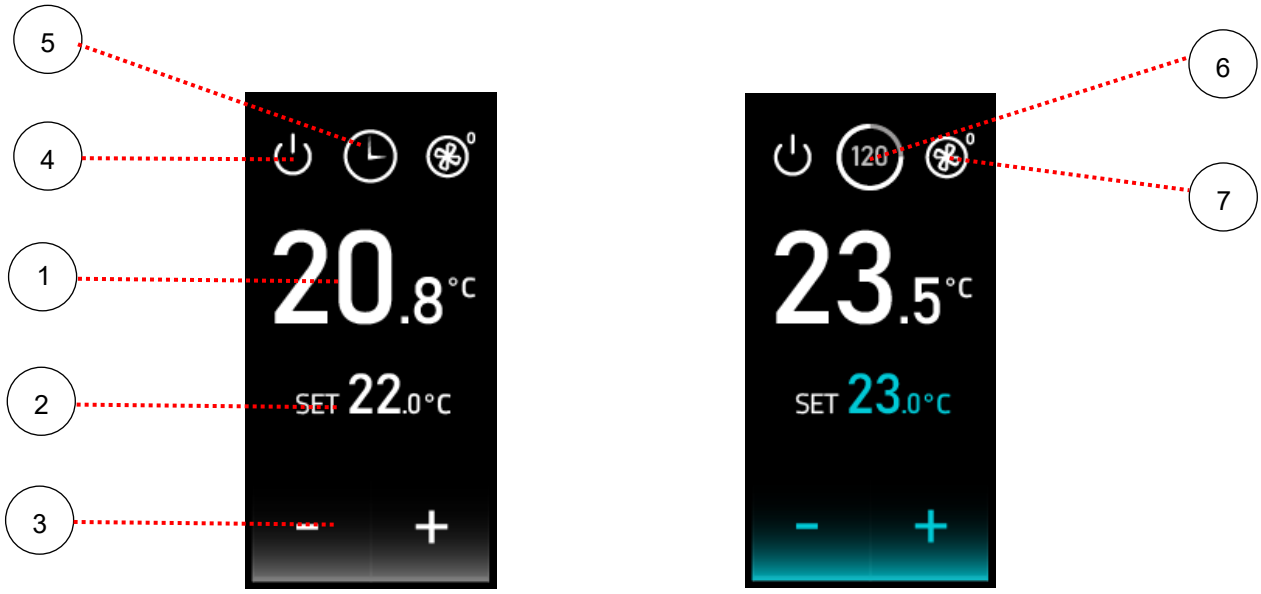
Single setpoint mode

The user can simply change the current setpoint, which is stored permanently. It is possible to use an external device to carry out hourly programming that directly changes the setpoint.

Absolute setpoint mode

The user can set one of the following operating modes:

- MANUAL: Equivalent to single setpoint operation mode. MANUAL operation is unlimited in duration
- CHRONO: the device receives the time schedule from an external supervisor, via a communication object of type HVAC_Mode
- BOOST: this is a manual forcing of the setpoint imposed by the time program, with limited duration. At the end of the forcing, the device automatically returns to CHRONO mode.



1. Room temperature
2. Actual setpoint
3. Setpoint increase/decrease keys
4. Icona / pulsante di spegnimento OFF termostato
5. Thermostat OFF icon / button
6. CHRONO mode inserted icon / button
7. BOOST mode inserted icon / button
8. MANUAL/AUTOMATIC fan mode icon / button

The heat or cool demand status is represented by the colour of the setpoint and the increase/decrease buttons: RED in heating conduction mode and BLUE in air conditioning conduction mode.

8.7.2 Use of temperature sensor

The temperature controller integrated inside the virtual 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.

Note on mounting position

If the integrated temperature regulator is used, the device must be preferably installed on an internal wall, at 0,7-1,5 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.

8.7.3 Applications

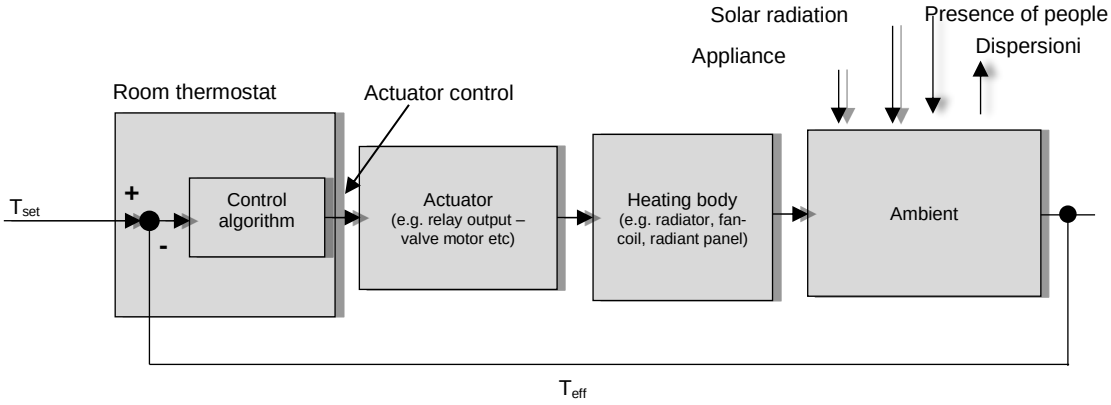
The applications that can be configured are peculiar to thermal plants with a double stage and concern the following terminals: radiators, electric radiators radiant panel systems and fan-coils.

The temperature control can be:

- two point control with hysteresis, ON-OFF command type;
- proportional-integral, with ON-OFF command, PWM or continuous type.
- 3 discrete velocity or continuous velocity for fan-coils

8.7.4 Control algorithms

The picture below 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}).

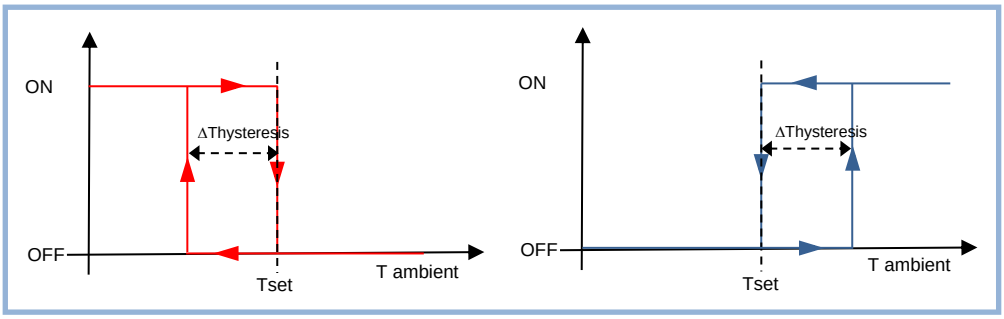


The control algorithm, basing on the difference between T_{set} and T_{eff}, processes a command value which can be analogue or On / Off type; the command is represented by a CO that is transmitted via bus, periodically or event based, to a KNX actuator device.

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.

8.7.4.1 Two-point control with hysteresis

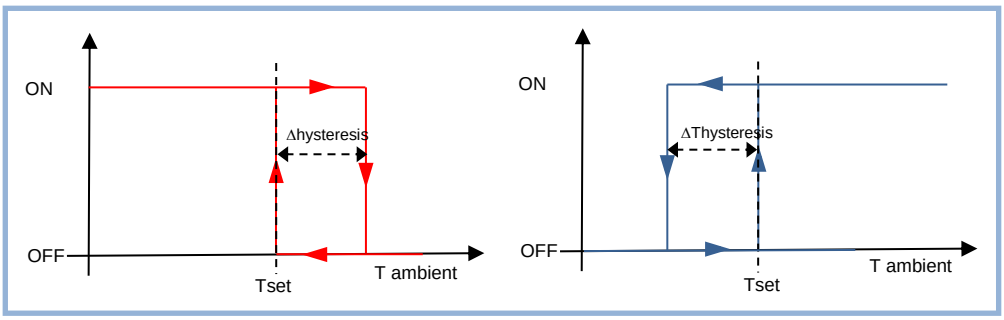
This control algorithm, which is also known as On / Off, is the most classic and popular. The control provides for the on / off switching of the system following a hysteresis loop, i.e. two threshold levels are considered for the switching instead of a single one.



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.



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.

8.7.4.2 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:

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

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

$Kp = proportional\ constant$

$Ki = intgral\ constant$

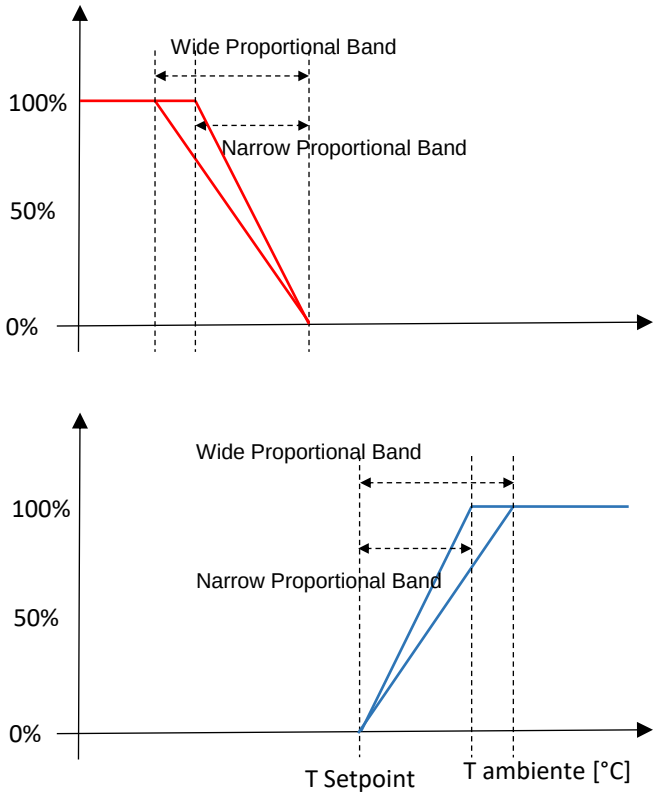
The control variable is composed by 2 numbers, one depending proportionally from the error and one depending from the integral of the error itself.

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.

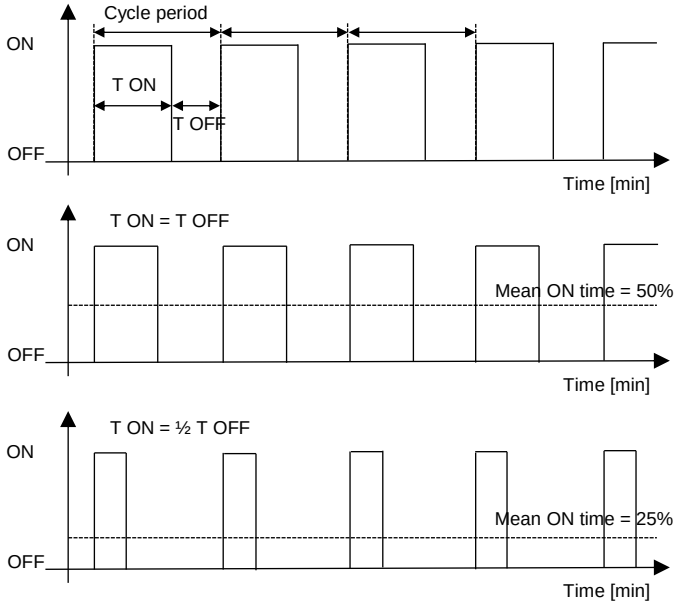


Integral Time is the amount of time necessary to repeat the value of the control variable of a purely proportional controller, when error is constant. For example, with a purely proportional controller with Proportional Band = 4 K, if Setpoint = 20°C and Measured Temperature = 18°C, the control variable 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.

8.7.4.3 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.



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

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.

8.7.5 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.

Five setpoint management modes are foreseen:

- Single setpoint
- Relative setpoints, heating/cooling switch over from bus
- Relative setpoints, automatic heating/cooling switch over
- Absolute setpoints, heating/cooling switch over from bus
- Absolute setpoints, automatic heating/cooling switch over

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.

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).

Absolute setpoints, heating/cooling switch over from bus

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, automatic heating/cooling switch over

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

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

Stand-by and economy setpoints are represented as attenuations to the comfort setpoint and can only be modified in the application program during commissioning: by uniquely modifying the comfort setpoint, references for attenuated modes and for comfort cooling setpoint mode (through switch over dead band) 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.

The switch over between operating modes is automatic and the information can be sent to other devices through communication object *Heating/cooling status out*, [1.100] DPT_Heat_Cool). Please refer to the section about heating/cooling switch over to learn more about switch over modes.

8.7.6 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 Absolute 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).

8.7.7 Heating/cooling switch over

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

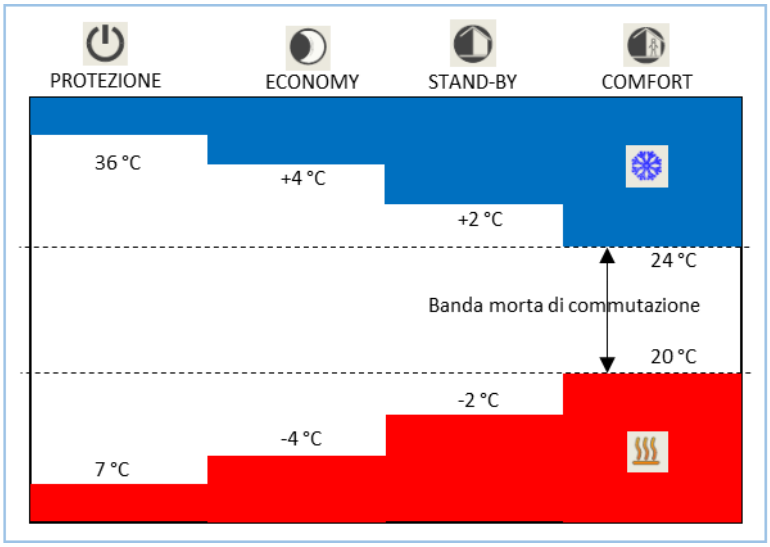
1. from KNX bus, through a communication object;
2. automatically, through a command from the internal logic of the device;

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

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.



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.

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.

8.7.8 Window switch management

Window switch management is an optional feature, oriented to energy saving, which becomes available only if the *Temperature control* function is enabled in the application program.

Whenever a condition of opened window is detected, the operating mode is forced to “building protection” and it remains forced as long as the open window condition is active. The program provides a time delay parameter for detection, in order to discriminate between an occasional short-term opening (e.g. to provide air exchange in the room) from an unintentional opening that justifies the power-saving function to be recalled.

The operating mode determined from Window switch management has priority on all operating mode settings imposed by the scheduler (in case *Setpoint management* = relative setpoint)

The physical detection of window openings is normally performed through switches that can be connected to KNX input devices; the pushbutton exposes up to 2 1-bit communication objects (*Temperature control* tab ⇒ External sensors) which can be synchronized to the switches’ states.

The internal logic performs a logical OR operation of the acquired contacts: the energy saving function is therefore activated if at least one window switch activation is detected. In order to determine the physical state of the contact corresponding to the “open window” state, two different options can be selected:

- NC (normally closed): open contact stands for closed window, closed contact stands for open window;
- NO (normally open): open contact stands for open window, closed contact stands for closed window;

8.7.9 Valve protection function

The function is suitable for both heating and cooling systems that use water as thermal conveying fluid and are provided with motorized valves for the interception of a zone or of a single room. Long periods of inactivity of the system can lead to the blockage of valves: to prevent this, the room temperature controller may periodically send a command to open / close the valve in the period of inactivity of the system. This possibility is further defined by the frequency and duration of the valve control.

8.7.10 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*.

8.8 Logic functions

The KNX virtual 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.

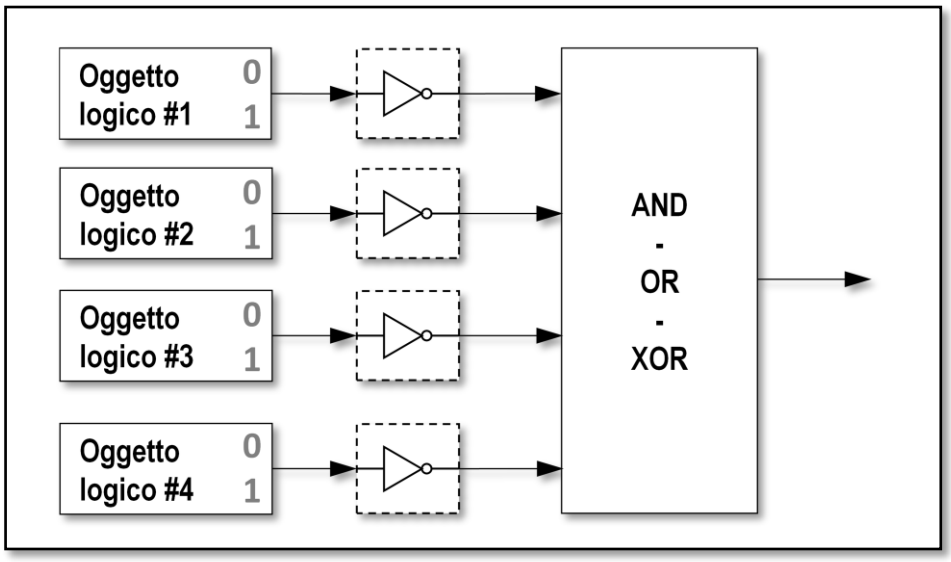
You can configure:

- 8 channels of 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 1 – 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:

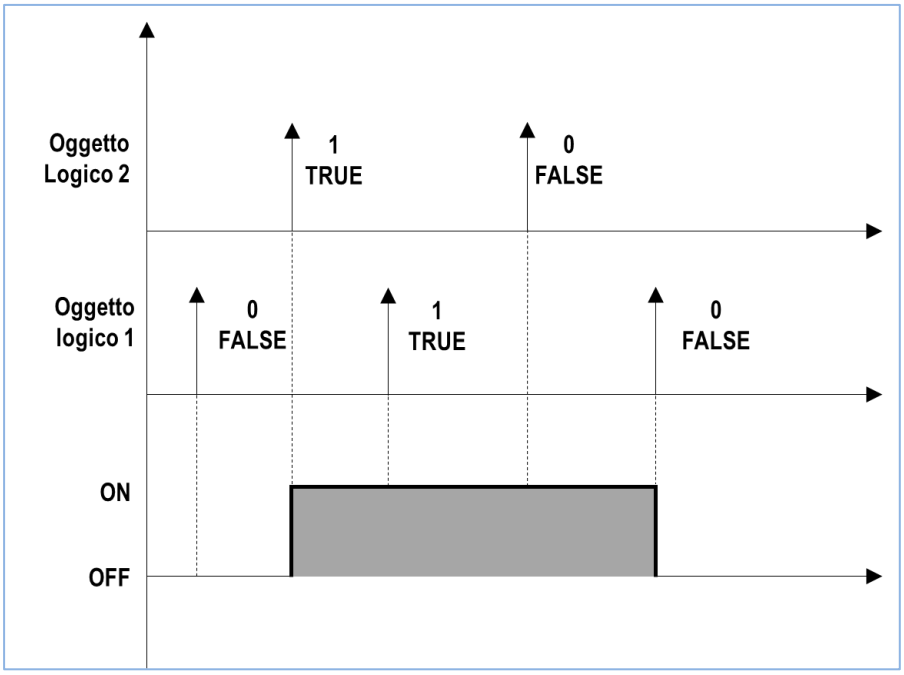


Figure 2 – Logic function OR

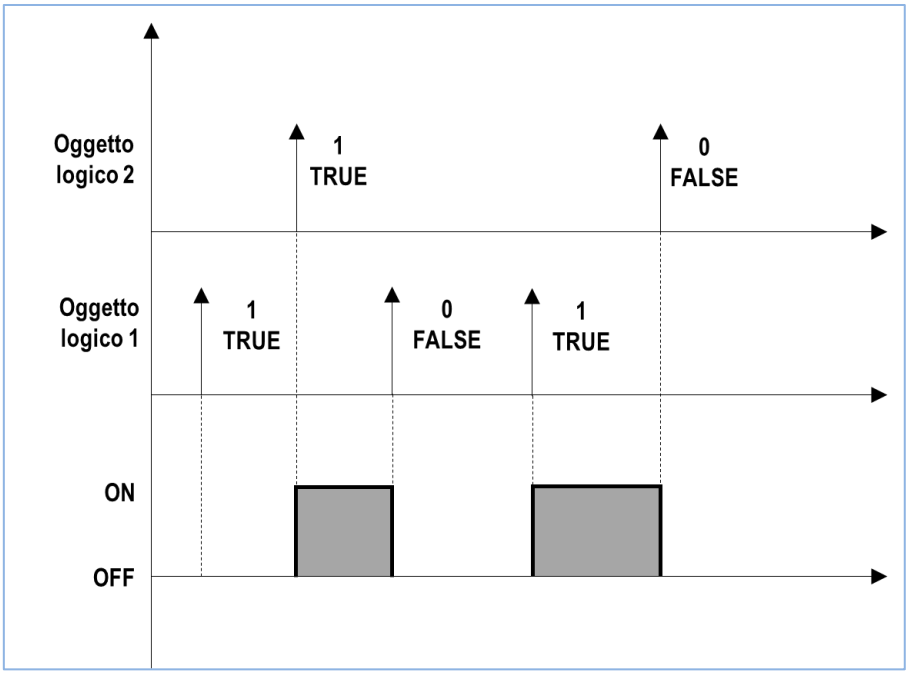


Figure 3 – Logic function AND

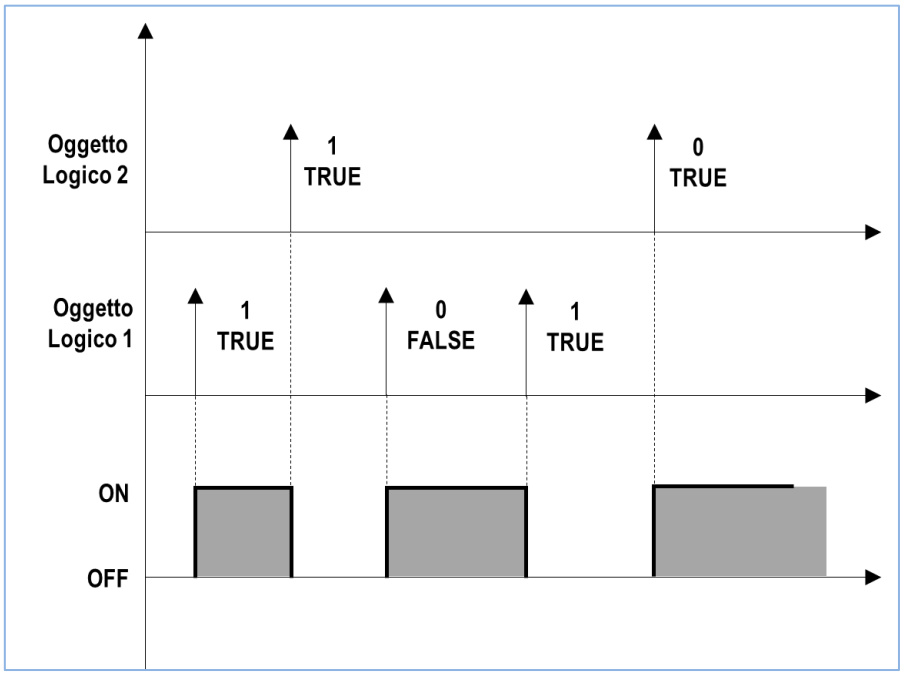


Figure 4 – Logic function Exclusive OR (XOR)

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.

9 Application program for ETS

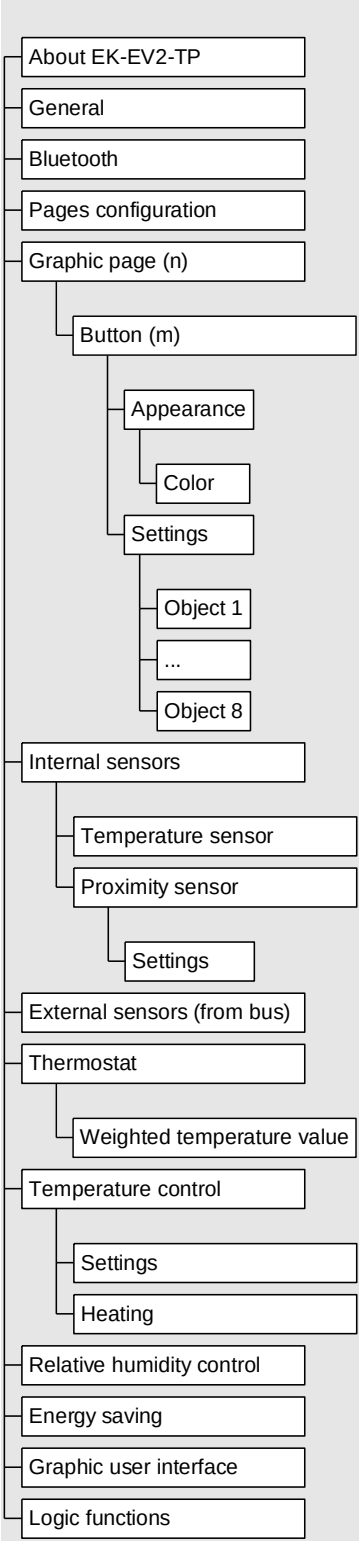
In the following chapters, there is the list of folder, parameters and communication objects of the application program.

Every channel, and every input or input pair under a channel, offers the same set of communication objects and parameters, but they may all be independently configured.

Hereafter, all channel-specific settings are listed grouped by channel; a generic channel number is referenced as “x” (where x = 1…4), while a generic input is referenced as “xx” (xx = 1A, 1B, 2A, ... 4B).

The parameter values highlighted in bold represent the default value.

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



In order to use the device as a temperature sensor or as a room temperature controller it is sufficient 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.

9.1 About EK-EV2-TP-…

The folder About EK-EV2-TP is for information purposes only and does not contain parameters to be set.
The information given is:

Application software for ETS5 and ETS6
Version 1.0 (or later)
EK-EV2-TP – Virtual pushbutton SIGNUM

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

9.2 General

The *General* tab contains the following parameters:

- device mounting type with left or right display alignment
- display backlight intensity
- display standby mode activation and activation delay
- haptic feedback activation
- communication activation delay when bus voltage is restored
- technical alarm activation
- activation of thermostat function and activation of widget on display

Parameter name	***Conditions***	***Values***
Keys layout		**left** right
	Determines the rotation of the display graphics. See the graphic preview	
Backlight brightness from bus		**no** / yes
Display backlight intensity	Backlight brightness from bus = no	**50** [range 20 … 100]
Haptic feedback activation		**no** / yes
Display standby mode		**no** / yes
Standby activation delay	Display standby mode = yes	**00:00:15.00 hh:mm:ss:ff** [range 00:00:00.00 … 01:49:13.50]
Standby backlight intensity	Display standby mode = yes	**80 %** [range 0 % … 100%]
Page after standby	Display standby mode = yes	Page 1 Page 2 Page 3 Thermostat page
[…]		
Delay after bus voltage recovery		**00:00:04.000 hh:mm:ss:fff**

Parameter name	*Conditions*	*Values*
		[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.	
[…]		
PIN enable		4 digits
Technical alarm		**disabled** enabled
	Enables a communication objects that activates an alarm indication through a bus telegram. The indication is made by flashing virtual buttons. *This indication is made available to the user for any purpose he sees fit (not necessarily an actual alarm).*	
Thermostat functionalities		**disabled** enabled

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Technical alarm	Technical alarm = enabled	1 bit	C–W--	[1.5] DPT_Alarm	7
Max display intensity percentage	Backlight brightness from bus = yes	1 byte	C–W--	[5.1] DPT_Scaling	20

9.3 Bluetooth

The Bluetooth tab contains the following parameters

- enabling of the graphical display page for manual activation of the Bluetooth LE AP (Access Point): customisation of texts and icons must be done via the Ekinex Tool app
- enabling of a 4-digit numeric PIN
- enabling of communication object for AP activation from bus
- enabling of communication object for AP locking from bus

Parameter name	*Conditions*	*Values*
Enable Bluetooth LE Access Point		Disabled **enabled**
Disable Bluetooth LE from bus	Enable Bluetooth LE Access Point = enabled	**Disabled** enabled
Lock from bus	Enable Bluetooth LE Access Point = enabled and Disable Bluetooth LE from bus = enabled	**Disabled** enabled
	The lock no longer allows you to enable the Bluetooth access point from the touch panel	

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Bluetooth enable	Enable Bluetooth LE Access Point = enabled	1 bit	C–WTU	[1.3] DPT_Enable	14
Lock Bluetooth function	Enable Bluetooth LE Access Point = enabled and Disable Bluetooth LE from bus = enabled	1 bit	C–W--	[1.3] DPT_Enable	15

9.4 Pages configuration

The tab allows you to define the number of graphic pages (max 3) for virtual button commands and the graphic theme to be applied. It is also possible to choose whether the ETS application download should overwrite settings (texts and icons) previously configured via the *Ekinex Tool* app.

Parameter name	*Conditions*	*Values*
Number of pages		**1,** 2, 3
Theme selection		**theme 1** theme 2
Override graphic after download (changes form app will be lost)		no / **yes**
	If the graphic appearance has been changed by an app with Bluetooth LE, by choosing the no option, the download of the ETS application does not overwrite the theme, text and icon associated with the button.	

9.5 Graphic page (n)

The number of virtual buttons (max. 4) for each configured graphics page (max. 3) is selected.

Parameter name	*Conditions*	*Values*
Number of button page (n)		1, 2, **4**
	See graphic preview	

The convention that is adopted for communication objects is to indicate as button 1 of each page the first from the top. If the button per page is 1, there is no misunderstanding; if > 1, then it always starts from the top.

9.6 Button (m)

The Tab *Button* includes the following parameters:

- Appearence
- Color
- Settings

Parameter name	*Conditions*	*Values*
Color activation	Button (m) – page (n) = enabled	Fixed **On contact closed** Status from bus with threshold
	Type of event to use as a trigger for an action.	
Type	Button (m) = enabled	**Send values or sequences** Dimming Shutter or venetian blind Scene

Parameter name	*Conditions*	*Values*
	Set the functionality associated with the button (m) - page (n). *Further parameters for the selected function appear in the individual configuration of the single button (see following paragraphs).*	

9.6.1 Appearence

Each virtual button can be customised either via the ETS or via the *Ekinex Tool* app via Bluetooth connection. The button can be customised in 2 ways:

- with a text distributed over 1-2 lines, with a maximum of 14 characters per line (the text is automatically centred and distributed on the graphic component)
- with a customised icon, extracted from a library of 120 components

Parameter name	*Conditions*	*Values*
View		**text** / icon
Number of text lines	View = text	**1** / 2
Text line 1	View = text, Number of text lines > = 1	[MAX 14 chars]
Text line 2	View = text, Number of text lines > = 1	[MAX 14 chars]
Button icon	View = icon	**0**, other values in range [1, …, 120]
QR code	View = icon	-
	By framing the QR with a smartphone or tablet, the Ekinex Tool app opens in the page of numbered icons, so that you can select the desired icon and enter the corresponding number in the "Symbol" field.	

Consult the general catalogue or web documentation to find the correspondence between the icon's nuneric index and the corresponding graphic rendering.

9.6.2 Color

The colour and flashing of the virtual button is determined by the colour activation parameter, with the following options:

- fixed colour
- at the button press event (via tap)
- according to the exceeding of predefined status/value thresholds of communication objects from the bus

Parameter name	*Conditions*	*Values*
Button (m) - color	Button (m) activation = fixed	**#FF0000** [range #000000, …, #FFFFFF]
	m = 1, …, 12	
Button (m) – color ON	Button (m) activation = on contact closed	**#FF0000** [range #000000, …, #FFFFFF]
	m = 1, …, 12	
Button (m) – color OFF	Button (m) activation = on contact closed	**#00D4FF** [range #000000, …, #FFFFFF]
	m = 1, …, 12	
Button (m) – OFF delay	Button (m) activation = on contact closed	**00:00:01.00** [range 00:00:00.00…01:49:13.50] hh:mm:ss.ff
	m = 1, …, 12	
Button (m) – threshold type	Button (m) 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
	m = 1, …, 12	
Button (m) - value 0 – blinking period	Button (m) activation = status from bus with threshold Button (m) – 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
	m = 1, …, 12	
Button (m) - value 0 – color	Button (m) activation = status from bus with threshold Button (m) – threshold type = 1 bit or 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	m = 1, …, 12	
Button (m) - value 1 – blinking period	Button (m) activation = status from bus with threshold Button (m) – 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
	m = 1, …, 12	

Parameter name	*Conditions*	*Values*
Button (m) - value 1 – color	Button (m) activation = status from bus with threshold Button (m) – threshold type = 1 bit or 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	m = 1, …, 12	
Button (m) - value 2 – blinking period	Button (m) activation = status from bus with threshold Button (m) – 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
	m = 1, …, 12	
Button (m) - value 2 – color	Button (m) activation = status from bus with threshold Button (m) – threshold type = 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	m = 1, …, 12	
Button (m) - value 3 – blinking period	Button (m) activation = status from bus with threshold Button (m) – 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
	m = 1, …, 12	
Button (m) - value 3 – color	Button (m) activation = status from bus with threshold Button (m) – threshold type = 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	m = 1, …, 12	
Button (m) – number of thresholds	Button (m) activation = status from bus with threshold Button (m) – threshold type ≠ 1 bit, 2 bits	**1** (range 1…8)
	m = 1, …, 12	
Button (m) – threshold x - blinking period	Button (m) activation = status from bus with threshold Button (m) – 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
	m = 1, …, 12	

Parameter name	*Conditions*	*Values*
Button (m) - threshold x - value	Button (m) activation = status from bus with threshold Button (m) – 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
	m = 1, …, 12	
Button (m) - threshold x - color	Button (m) activation = status from bus with threshold Button (m) – threshold type ≠ 1 bit, 2 bits	**#FFFF00** [range #000000, …, #FFFFFF]
	m = 1, …, 12	

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Color (m) - Command *[tipo]*	Button (m) activation = status from bus with threshold Color Button (m) – threshold type = 1 bit	1 bit	C-WTU	[1.1] DPT_Switch	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
	m = 1, …, 12				
Color (m) - Command *[tipo]*	Button (m) activation = status from bus with threshold Color Button (m) – threshold type = 2 bits	2 bits	C-WTU	[2.1] DPT_Switch_Control	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
	m = 1, …, 12				
Color (m) - Command *[tipo]*	Button (m) activation = status from bus with threshold Color Button (m) – threshold type = 1 signed byte	1 byte	C-WTU	[6.010] DPT_Value_1_Count counter pulses -128…+127	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
	m = 1, …, 12				
Color (m) - Command *[tipo]*	Button (m) activation = status from bus with threshold Color Button (m) – threshold type = 1 unsigned byte	1 byte	C-WTU	[5.010] DPT_Value_1_Ucount counter pulses 0…255	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
	m = 1, …, 12				
Color (m) - Command *[tipo]*	Button (m) activation = status from bus with threshold Color Button (m) – threshold type = 2 signed byte	2 bytes	C-WTU	[8.1] DPT_Value_2_Count pulses difference	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
	m = 1, …, 12				
Color (m) - Command *[tipo]*	Button (m) activation = status from bus with threshold Color Button (m) – threshold type = 2 unsigned byte	2 bytes	C-WTU	[7.1] DPT_Value_2_Ucount pulses	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
	m = 1, …, 12				
Color (m) - Command *[tipo]*	Button (m) activation = status from bus with threshold Color Button (m) – threshold type = 2 floating bytes	2 bytes	C-WTU	[9.0*] DPT_Value_x 2-byte float value	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
	m = 1, …, 12				

9.6.3 Settings (for send values or sequences)

Parameter name	*Conditions*	*Values*
Lock function	Button (m) = enabled	**disabled** enabled
Number of communication objects	Button (m), page (n) = enabled Type = send values or sequences	1…8 (**1**)
	Number of communication objects configured in association with the button event.	
Event	Button (m), page (n) = 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 (m), page (n) = 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.	

9.6.3.1 Lock function for Send values or sequences

Parameter name	*Conditions*	*Values*
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).	
Lock function – Behaviour at locking	Button (m), page (n) = 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 (m), page (n) = 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.	

9.6.3.2 Object (x)

Parameter name	*Conditions*	*Values*
Object x – Send delay	Button (m), page (n) = 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.*	

Parameter name	*Conditions*	*Values*
Object x – Send cyclically	Button (m), page (n) = 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 x – Cyclic sending interval	Button (m), page (n) = enabled Type = send values or sequences Number of communication objects = 1 Send cyclically ≠ none	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Object x – Communication object dimension	Button (m), page (n) = 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 x – Reaction to activation	Button (m), page (n) = enabled Type = send values or sequences c.o. dimension = 1 bit	none on off **toggle**
	Button (m), page (n) = 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 (m), page (n) = 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 x – Reaction to long press	Button (m), page (n) = enabled Type = send values or sequences c.o. dimension = 1 bit	**none** on off toggle
	Button (m), page (n) = 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 (m), page (n) = 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 x – Value 1	Button (m), page (n) = 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	

Parameter name	*Conditions*	*Values*
Object n – Value 2	Button (m), page (n) = 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*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Button (m) - State *[type]*, object x*	Button (m) = enabled Type = send values or sequences	*according to the configuration* (**1-bit**)	C-WTU	*according to the configuration* (**[1.001] switch**)	45, …, 52, 61, …, 68, 77, …, 84, 93, …, 100, 109, …, 116, 125, …, 132, 141, …, 148, 157, …, 164, 173, …, 180, 189, …, 196, 205, …, 212, 221, …, 228
	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

Tabella 1 - Dimensioni e tipi di dati per gli oggetti di comunicazione

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 1):

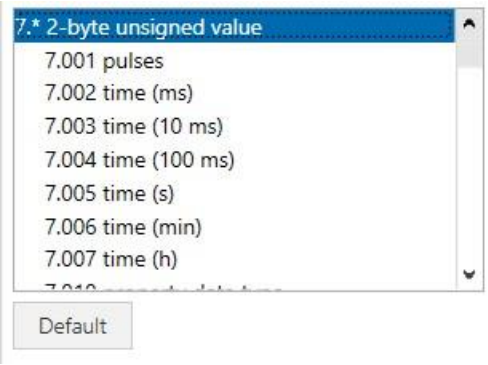


Figure 1 - DPT of 7.* 2-bytes floating point

9.6.4 Settings (for dimming)

Parameter name	*Conditions*	*Values*
Lock function	Button (m), page (n) = enabled	**disabled** enabled
Dimming type	Button (m), page (n) = enabled Type = dimming	**Brightness only** Brightness and color temperature
	Dimming type selection: brightness only or tunable white	

9.6.4.1 Lock function for dimming

Parameter name	*Conditions*	*Values*
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).	
Lock function – Behaviour at locking	Button (m), page (n) = 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 (m), page (n) = enabled Type = dimming	**none** off on as previous
	Value to be assigned to the object wen an unlocking command is received.	

Nome oggetto	*Condizioni*	*Dimens.*	*Flags*	*DPT*	*Nr. Ogg. Com.*
Button (m) - Switching command	Button (m) = enbled Type = dimming	1 bit	C-WTU	[1.1] DPT_Switch	53, 69, 85, 101, 117, 133, 149, 165, 181, 197, 213, 229
	Sends an on/off command to a dimmer actuator *The command is sent following a short press event on the button.* *The value sent can either be a permanently fixed value or alternate between the two possible values at each activation.*				
Button (m) - Dimming command value	Button (m) = enbled Type = dimming	1 Byte	C-W--	[5.1] DPT_Scaling	54, 70, 86, 102, 118, 134, 150, 166, 182, 198, 214, 230
Button (m) - Dimming status value	Button (m) = enbled Type = dimming	1 Byte	CRWTU	[5.1] DPT_Scaling	55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231
Button (m) - Color temperature command	Button (m) = enbled Type = dimming	1 Byte	C-W--	[5.1] DPT_Scaling	56, 72, 88, 104, 120, 136, 152, 168, 184, 200, 216, 232
Button (m) - Color temperature status	Button (m) = enbled Type = dimming	1 Byte	CRWTU	[5.1] DPT_Scaling	57, 73, 89, 105, 121, 137, 153, 169, 185, 201, 217, 233

9.6.5 Settings (for shutter or venetian blind)

Parameter name	*Conditions*	*Values*
Lock function	Button (m), page (n) = enabled	**disabled** enabled
Long press time	Button (m) = 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 (m) = 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 (m) = enabled Type = shutter or venetian blind Toggle mode = disabled	**up** down

Parameter name	Conditions	Values
	Defines the movement direction to be assigned to the button press.	
Venetian blind mode	Button (m) = 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.	

9.6.5.1 Lock function for shutter or venetian blind

Parameter name	Conditions	Values
Lock function – invert lock device signal	Button (m), page (n) = 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 (m), page (n) = 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).	
Lock function – Behaviour at locking	Button (m), page (n) = enabled Type = shutter or venetian blind	**none** up down
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Button (m), page (n) = enabled Type = shutter or venetian blind	**none** up down
	Operation to perform when an unlocking command is received.	

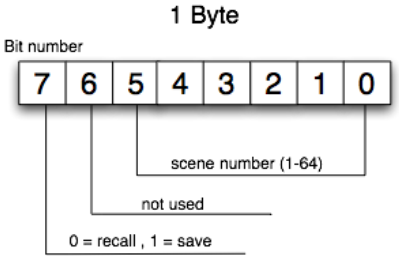
Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Button (m) - Dedicated stop command	Button (m) = enabled Type = shutter or venetian blind	1 bit	C- -T-	[1.17] DPT_Trigger	53, 69, 85, 101, 117, 133, 149, 165, 181, 197, 213, 229
	Immediately stop any movement of the blind. The object is sent on a short press if the blind mode is disabled and at the end of a long press if the venetian blind mode is enabled.				
Button (m) - Stop-step up / down command	Button (m) = enabled Type = shutter or venetian blind Venetian blind mode = enabled	1 bit	CR-T-	[1.7] DPT_Step	55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231
	Move the blind to fully open or fully closed position. The object is sent at the end of a long press.				
Button (m) - Move up / down command	Button (m) = enabled Type = shutter or venetian blind Venetian blind mode = enabled	1 bit	CRWTU	[1.8] DPT_UpDown	56, 72, 88, 104, 120, 136, 152, 168, 184, 200, 216, 232
	Increase or decrease the opening of the blind stepwise. *The object is sent on a short press.*				

9.6.6 Settings (for scene)

Parameter name	*Conditions*	*Values*
Lock function	Button (m), page (n) = enabled	**disabled** enabled
First scene number	Button (m) = 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 (m) = enabled Type = scene	enabled / **disabled**
	When enabled, a long key press can be used to program the selected scene by storing the current parameters.	
Long press time	Button (m) = 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 (m) = 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 (m) = enabled Type = scene Learning mode = disabled Scene activation = toggle between two scenes	1…64 (**2**)
	Alternate scene number to be assigned to button press.	

9.6.6.1 Lock function for scene

Parameter name	*Conditions*	*Values*
Lock function – invert lock device signal	Button (m), page (n) = 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 (m), page (n) = 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).	
Lock function – Behaviour at locking	Button (m) = 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 (m) = enabled Type = scene	**none** send first scene send second scene
	Operation to perform when an unlocking command is received.	

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Button (m) - Scene number	Button (m) = enabled Type = scene	1 Byte	C--T-	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	57, 73, 89, 105, 121, 137, 153, 169, 185, 201, 217, 233
	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).  				

General note on 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 1.

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 1 - Lock function

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

Parameter name	Conditions	Values
	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	Dim.	Flags	DPT	Comm. Obj. No.
Button (m) - Lock command	Button (m) = enabled Lock function = enabled	1 bit	C-W--	[1.3] DPT_Enable	44, 60, 76, 92, 108, 124, 140, 156, 172, 188, 204, 220

9.7 Internal sensors

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

- Temperature sensor;
- Proximity sensor (if available)

Parameter name	*Conditions*	*Values*
Temperature sensor		**enabled** disabled
	Enables the temperature sensor by making the corresponding communication object and the specific tab available.	
Proximity sensor (if available)		**disabled** enabled from bus
	Enables the proximity sensor by making the corresponding communication object and the specific tab available.	

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Temperature value	Temperature sensor = enabled	2 Bytes	CR–T-	[9.1] DPT_Value_Temp	234

9.7.1 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

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]

Parameter name	*Conditions*	*Values*
Threshold 1	Temperature sensor = enabled	**not active** below above
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*	*Comm. Obj. No.*
Temperature value	Temperature sensor = enabled	2 bytes	CR-T-	[9.1] DPT_Value_Temp	234
Temperature threshold 1 - switch	Temperature sensor = enabled, Threshold 1 = above or below	1 bit	CR-T-	[1.1] DPT_Switch	235
Temperature threshold 1 - Lock	Temperature sensor = enabled, Threshold 1 = above or below Threshold lock enable = yes	1 bit	C-W--	[1.1] DPT_Switch	236

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Temperature threshold 1 – Value (from bus)	Temperature sensor = enabled, Threshold 1 = above or below Threshold update value from bus = yes	2 bytes	C-W--	[9.1] DPT_Value_Temp	237
Temperature threshold 2 - switch	Temperature sensor = enabled, Threshold 2 = above or below	1bit	CR-T-	[1.1] DPT_Switch	238
Temperature threshold 2 - Lock	Temperature sensor = enabled, Threshold 2 = above or below Threshold lock enable = yes	1 bit	C-W--	[1.1] DPT_Switch	239
Temperature threshold 2 – Value (from bus)	Temperature sensor = enabled, Threshold 2 = above or below Threshold update value from bus = yes	2 bytes	C-W--	[9.1] DPT_Value_Temp	240

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.

9.7.2 Proximity sensor

La scheda ***Sensore di prossimità*** contiene i seguenti parametri:

- Sensibilità
- Intensità massima in stato di riposo
- Invio valori o sequenze
- Funzione di blocco
- Numero di oggetti di comunicazione

Parameter name	*Conditions*	*Values*
Sensibilità	Sensore di prossimità = abilitato	bassa **media** alta
	Seleziona la sensibilità del sensore *Basso = valore medio ogni 4 misurazioni* *Medio = valore medio ogni 16 misurazioni* *Alto = valore medio ogni 64 misurazioni*	
Intensità massima standby	Sensore di prossimità = abilitato o da bus	**80 %** [campo 0 % … 100%]
	Imposta la luminosità massima dei tasti quando in stato di riposo(stand-by)	
Durata retroilluminazione		**00:00:00 hh:mm:ss** [campo 00:00:00 … 18:12:15]
	Durata della retroilluminazione dopo la sua attivazione da parte del sensore.	
Invio valori o sequenze	Sensore di prossimità = abilitato o da bus	abilitato **disabilitato**
	Se il parametro è impostato al valore 0, nessun valore è inviato al cambiamento.	
Numero di oggetti di comunicazione	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato	1…8 **1**
Oggetto *n* – Ritardo di invio	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato Numero di oggetti di comunicazione = *n*	hh:mm:ss.ff (**00:00:00.00**)
	Ritardo fra l'evento e la trasmissione del valore sul bus. *Definendo un ritardo individuale per ogni oggetto è possibile formare una sequenza di telegrammi definita da associare all'evento.*	
Oggetto *n* – Invio ciclico	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato Numero di oggetti di comunicazione = *n*	**nessuno** off / valore 1 on / valore 2 entrambi off e on / entrambi i valori
Oggetto *n* – dimensione oggetto di comunicazione	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato Numero di oggetti di comunicazione = *n*	**valore a 1 bit** valore a 2 bit 1 byte senza segno 1 byte percentuale 1 byte con segno 2 byte senza segno 2 byte con segno valore con virgola mobile a 2 byte
Oggetto *n* – reazione alla rilevazione	*Definisce il tipo e la dimensione dei singoli oggetti di comunicazione.*	
	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. = 1 bit	nessuno on off **toggle**
	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. = 2 bits	nessuno disabilitare abilita off / salita **abilita on / discesa** abilita off / salita ↔ disabilita abilita on / discesa ↔ disabilita abilita off / salita ↔ abilita on / discesa
	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. ≠ 1 bit o 2 bits	nessuno **invio valore 1** invio valore 2 invio valore 1 ↔ invio valore 2
	La configurazione dell'oggetto per la rilevazione della prossimità cambia in funzione del tipo di oggetto di comunicazione selezionato.	
Oggetto *n* – reazione alla mancanza di rilevazione	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. = 1 bit	**nessuno** on off toggle

Parameter name	*Conditions*	*Values*
	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. = 2 bit	nessuno **disabilitare** abilita off / salita abilita on / discesa abilita off / salita ↔ disabilita abilita on / discesa ↔ disabilita abilita off / salita ↔ abilita on / discesa
	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. ≠ 1 bit o 2 bit	**nessuno** invio valore 1 invio valore 2 invio valore 1 ↔ invio valore 2
	La configurazione dell'oggetto per la fine della rilevazione della prossimità cambia in funzione del tipo di oggetto di comunicazione selezionato.	
Oggetto *n* – Valore 1	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. ≠ 1 bit o 2 bit	0...255 (1 byte senza segno) 0…100 (1 byte percentuale) -128...127 (1 byte con segno) 0...65535 (2 byte senza segno) -32768... 32767 (2 byte con segno) -671088.64...670760.96 (2 byte virg. mobile)
	Primo valore configurato per l'associazione ad eventi	
Oggetto *n* – Valore 2	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato dimensione oggetto com. ≠ 1 bit o 2 bit	0...255 (1 byte senza segno) 0…100 (1 byte percentuale) -128...127 (1 byte con segno) 0...65535 (2 byte senza segno) -32768... 32767 (2 byte con segno) -671088.64...670760.96 (2 byte virg. mobile)
	Secondo valore configurato per l'associazione ad eventi	
Funzione di blocco	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato	abilita **/ disabilita**
Inverte segnale blocco dispositivo	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato Funzione di blocco = abilitata	**non invertita** / invertita
Blocco dopo ripristino del bus	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato Funzione di blocco = abilitata	**no** / si
Comportamento al blocco	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato Funzione di blocco = abilitata	**nessuno** / se presenza rilevata / se presenza non rilevata
Comportamento allo sblocco	Sensore di prossimità = abilitato o da bus Invio valori o sequenze = abilitato Funzione di blocco = abilitata	**nessuno** / se presenza rilevata / se presenza non rilevata

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Proximity - Switching status *[type]*, object *n**	Sensore di prossimità = abilitato Invio valori o sequenze = abilitato	*a seconda della configurazione* (**1-bit**)	C-WTU	*a seconda della configurazione* ([1.1] DPT_Switch)	24, 25, 26, 27, 28, 29, 30, 31
	Possono essere definiti fino a 8 oggetti da associare ad uno stesso evento.				
Proximity - Lock command	Sensore di prossimità = abilitato Invio valori o sequenze = abilitato Funzione di blocco = abilitata	1 bit	C-W--	[1.3] DPT_Enable	23

9.8 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 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.

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

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]

Parameter name	*Conditions*	*Values*
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]
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]

Parameter name	***Conditions***	***Values***
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.	
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.*
Thermostat - Room temperature (from bus)	Temperature sensor = enabled	2 Byte	C-WTU	[9.1] DPT_Value_Temp	241
Thermostat - Humidity (2 byte, from bus)	Relative humidity sensor = enabled, Humidity comm. obj. size = 2 byte	2 Byte	C-WTU	[9.7] DPT_Value_Humidity	242
Thermostat - Humidity (1 byte, from bus)	Relative humidity sensor = enabled, Humidity comm. obj. size = 1 byte	1 Byte	C-WTU	[5.1] DPT_Scaling	243
Thermostat - Antistratification temperature (from bus)	enabled	2 Byte	C-WTU	[9.1] DPT_Value_Temp	246
Thermostat - Outdoor temperature (from bus)	enabled	2 Byte	C-WTU	[9.1] DPT_Value_Temp	247
Thermostat - Coil temperature (from bus)	enabled	2 Byte	C-WTU	[9.1] DPT_Value_Temp	248
Thermostat - Floor temperature (from bus)	enabled	2 Byte	C-WTU	[9.1] DPT_Value_Temp	249
Thermostat - Flow temperature (from bus)	enabled	2 Byte	C-WTU	[9.1] DPT_Value_Temp	250
Thermostat - Anticondensation (from bus)	enabled	1 Bit	C-WTU	[1.1] DPT_Switch	256
Thermostat - Windows contact sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.019] DPT_Window_Door	251
Thermostat - Windows contact sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.019] DPT_Window_Door	252
Thermostat - Presence sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.018] DPT_Occupancy	253
Thermostat - Presence sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.018] DPT_Occupancy	254
Thermostat - Card holder contact (from bus)	enabled	1 Bit	C-WTU	[1.001] DPT_Switch	255

9.9 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

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.	
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*	*Comm. Obj. No.*
Thermostat - Weighted temperature	Cyclic sending interval ≠ no sending	2 Byte	CR-T--	[9.001] DPT_Value_Temp	257

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.

9.10 Temperature and relative humidity 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.

9.10.1 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
- Thermostat function 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

Parameter name	***Conditions***	***Values***
Setpoint type	General -> Thermostat functionalities = enabled	**Single** Absolute
	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	**from bus** / automatic

	Command Communication Object = separated Thermostat function = both heating and cooling	
	In case Setpoint management = single setpoint, or the Command Communication Object is unique, the heating-cooling switch over must be carried out from bus.	
Heating-cooling cyclic sending interval	Thermostat function = both heating and cooling	**no sending** [other values in range 30 s … 120 min]
Thermostat function 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*	*Comm. Obj. No.*
Thermostat - Actual setpoint		2 Bytes	CR-T--	[9.1] DPT_Value_Temp	268
Thermostat - Manual setpoint	Setpoint type = absolute	2 Bytes	C-W---	[9.1] DPT_Value_Temp	269

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Thermostat - Heating/cooling status out		1 Bit	CR-T--	[1.100] DPT_Heat_Cool	258
	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**				
Thermostat - Heating/cooling status in	Thermostat function = both heating and cooling, Heating–cooling changeover = from bus	1 Bit	C-W--	[1.100] DPT_Heat_Cool	259
	The communication object is received from the bus. At the switching event the internal regulator switches the conduction mode.				
Thermostat - HVAC mode in	Setpoint type = absolute	1 Byte	C-W--	[20.102] DPT_HVACMode	260
	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).*				
Thermostat - HVAC forced mode in	Setpoint type = absolute	1 Byte	C-W--	[20.102] DPT_HVACMode	261
	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.				
Thermostat - HVAC mode out	Setpoint type = absolute	1 Byte	CR-T-	[20.102] DPT_HVACMode	262
Thermostat - HVAC manual mode	Setpoint type = absolute	1 Byte	C-WTU	[20.102] DPT_HVACMode	263
Thermostat - Chrono active status	Setpoint type = absolute	1 Bit	CR-T-	[1.11] DPT_State	264
Thermostat - Room temperature controller status		1 Bit	CR-T-	[1.3] DPT_Enable	280
Thermostat - Alarm text		14 bytes	CR-T-	[16.0] DPT_String_ASCII	13
Thermostat - Manual setpoint active status	Setpoint type = absolute	1 Bit	CRWTU-	[1.11] DPT_State	270

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Thermostat - Input setpoint	Setpoint type = single	2 bytes	CRWTU	[9.1] DPT_Value_Temp	271
	The communication object allows to set and / or read the status of the setpoint (manual / forced).				
Thermostat - Room temperature control alarm		1 bit	CR-T--	[1.5] DPT_Alarm	10
Thermostat - Disable room temperature controller	Disable temperature controller from bus = yes	1 bit	C-W--	[1.3] DPT_Enable	279
Thermostat - Thermal generator lock		1 bit	C-W--	[1.5] DPT_Alarm	8
Thermostat - Building protection HVAC mode active		1 bit	CR-T--	[1.11] DPT_State	266

Note on system terminals for heating and cooling

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;
- fancoil;
- radiant panels;
- dehumidifiers;
- radiant panels + radiators (as an auxiliary stage);
- radiant panels + fancoil (as an auxiliary stage);
- radiant panels + dehumidifiers.

Changeover heating / cooling

Switching between the two plant conduction modes (heating and cooling) can take place as follows:

1. automatically on the initiative of the device;

2. from the KNX bus by means of a specific communication object.

Automatic switching (mode 1)

The automatic switching is suitable for a hydraulic configuration of the 4-pipe heating / air conditioning system (used for example to power fan-coil terminals or radiant ceiling panels). Also in this case the information can be sent on the bus with the output communication object [DPT 1.100 heat / cool]; switching is carried out automatically by the device based on the comparison between the values of the actual temperature and that of the setpoint.

Automatic switching is achieved with the introduction of a dead zone according to the scheme shown in Figure 11.

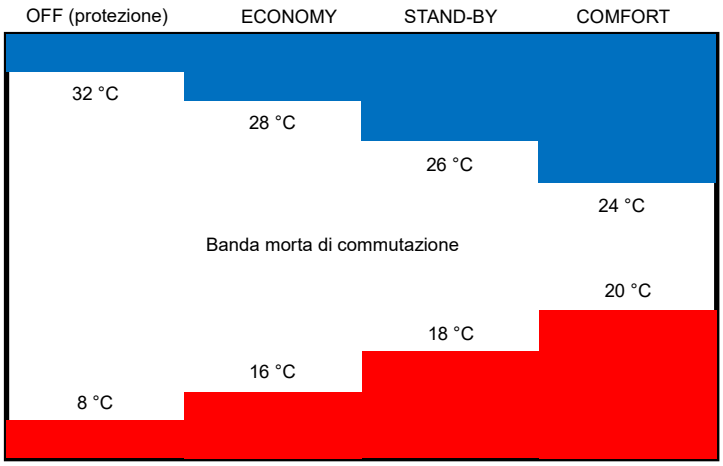
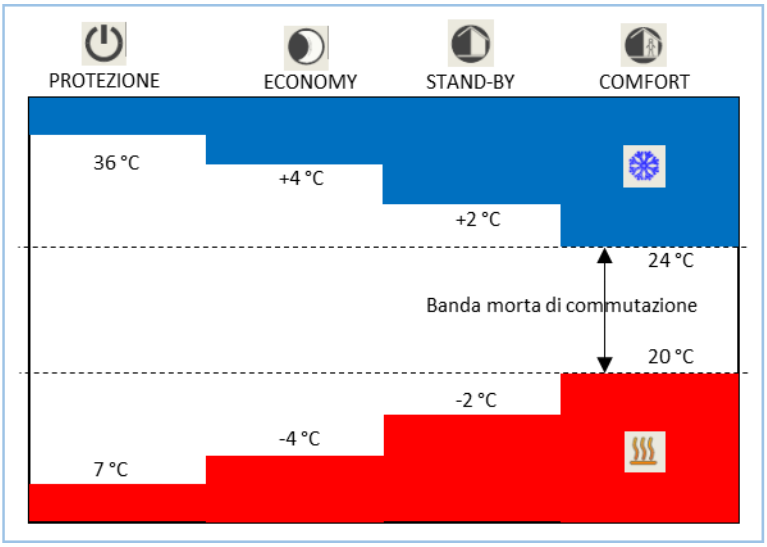


Figura 5 - Zona morta ed esempio di valori di setpoint coerentemente distribuiti



As long as the actual (measured) temperature is below the setpoint value for heating, the conduction mode remains heating; similarly, if the actual (measured) value is higher than the setpoint value for cooling, the conduction mode is cooling. If the actual (measured) temperature value is within the dead zone, the conduction mode remains the one previously active; the switching point of the heating / cooling conduction mode must occur in correspondence with the current setpoint of the set HVAC mode, in the same way the passage from cooling / heating must occur at the set heating setpoint.

Switching from the KNX bus (mode 2)

Switching from the bus requires that the command comes from another KNX device, such as another thermostat, a Touch & See unit or supervision software configured for the purpose. This acts as a "supervisor" device: switching takes place via the input communication object [DPT 1.100 heat / cool]. In this mode, manual switching by the user is inhibited. Thanks to this mode, the supervisor device is able to make the "slave" devices perform timed programs, extending their function to that of a chronothermostat (centrally controlled by the supervisor device).

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

The object Heating / cooling status in is exposed only when the function is both heating and cooling and the switching between modes is carried out by the bus.

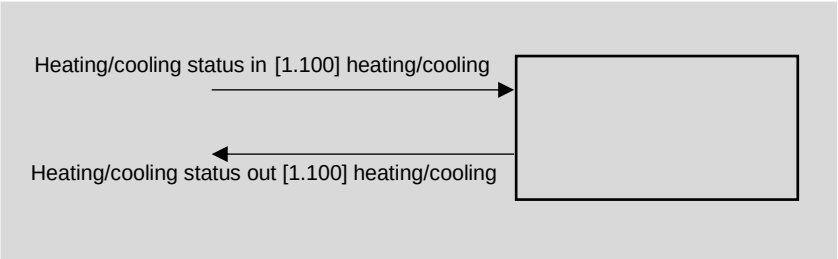


Figura 6 – Schema commutazione dal bus

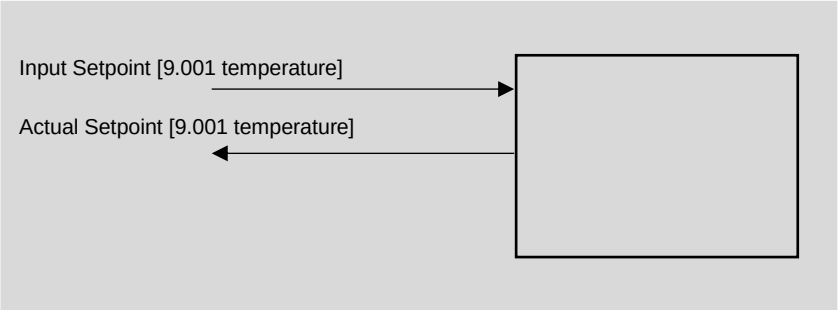
Valve protection function

The function is suitable for heating and cooling systems that use water as a heat transfer fluid and have motorized valves for the interception of a zone or a single environment. Long periods of system inactivity can lead to valve blocking: to prevent this possibility, the thermostat can periodically send a valve opening / closing command while the system is not in use. This possibility is made available in the application program by means of the "Valve protection function" parameter, further defined by the frequency and duration of the valve actuation.

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

Remote modification of the Setpoint

In the single setpoint configuration, the communication objects (O.C. hereafter) indicated in the block diagram of Figure 13 allow you to manually monitor and modify the Setpoint.



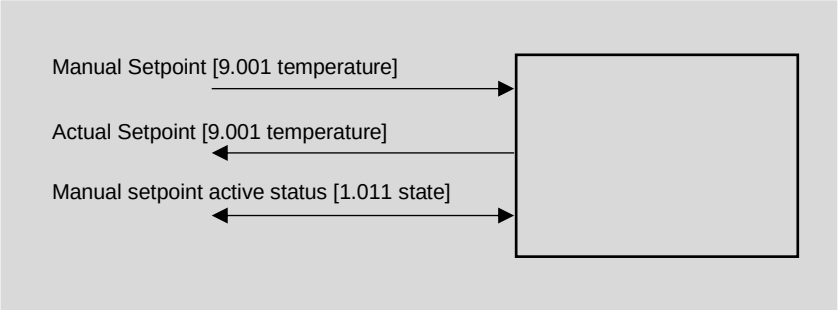


Figura 7 – Modifica manuale del setpoint

The objects refer to the manual modification of the Setpoint: alternatively, the supervisor can act directly on the Setpoints of the operating modes. The value of the O.C. Current setpoint represents the current operational setpoint on which the control algorithms operate. The O.C. Manual / forced setpoint status inserted indicates in reading if the forced mode is inserted. The supervisor can force the current setpoint at any time by writing a new value directly into the O.C. Manual setpoint. The O.C. Manual / forced setpoint status entered can also be used in writing to exit the active forced mode.

Remote modification of operating modes

The communication objects indicated in the block diagram in the figure allow you to monitor changes in the operating mode (comfort, stand-by, economy and building protection) made in manual / forced mode, or the operating mode imposed by the time programming. The communication objects (O.C. hereafter) also allow the same changes to be made remotely, for example via a system supervisor.

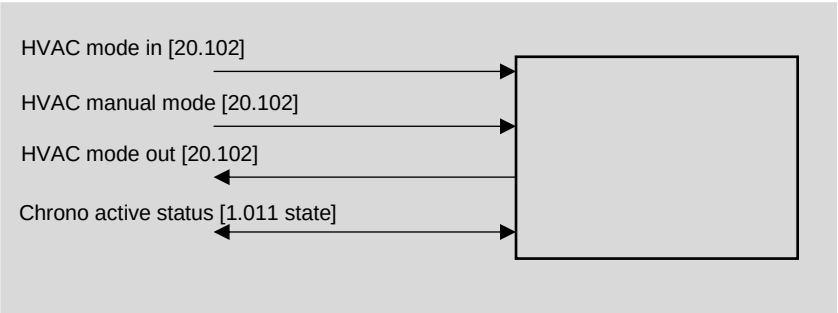


Figura 8 – Modifica modi operativi da remoto

The O.C. HVAC mode in is associated with the system time program. The O.C. HVAC out mode and time program status entered allow the remote supervisor to reconstruct the active mode on the room thermostat and allow you to understand if the time program is entered or the attenuation is managed manually. The supervisor can set a manual operating mode at any time through the O.C. Manual HVAC mode; to enter the time program in progress remotely, simply set the O.C. Time program status entered at value 0 = Automatic.

9.10.2 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
- Proportional band [0.1 K]
- Integral time [min]
- PWM cycle time
- Min. change of value to send [%]
- Min control value [%]
- Max control value [%]
- Floor temperature limitation
- Temperature limit [°C]
- Hysteresis [K]
- Auxiliary heating
- Communication object type
- Disable from bus
- Offset from setpoint
- Hysteresis
- Cyclic sending interval
- Ventilation for auxiliary heating

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.	
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.	

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Thermostat - Input setpoint	Setpoint type = single	2 bytes	CRWTU	[9.1] DPT_Value_Temp	271
Thermostat - Comfort setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU	[9.1] DPT_Value_Temp	271
Thermostat - Standby setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU	[9.1] DPT_Value_Temp	273
Thermostat - Economy setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU-	[9.1] DPT_Value_Temp	275
Thermostat - Building protection setpoint (heating)		2 Byte	CRWTU-	[9.1] DPT_Value_Temp	277
Thermostat - Heating out command	Control type = 2 points hysteresis or PWM Command communication object = separated	1 Bit	CR-T-	[1.1] DPT_Switch	281
Thermostat - Heating out command	Control type = continuous Command communication object = separated	1 Byte	CR-T-	[5.1] DPT_Scaling	281
Thermostat - Heating and cooling out command	Command communication object = unique, Control type = 2 points hysteresis or PWM	1 Bit	CR-T-	[1.1] DPT_Switch	281

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Thermostat - Heating and cooling out command	Command communication object = unique, Control type = continuous	1 Byte	CR-T--	[5.1] DPT_Scaling	281
Thermostat - Auxiliary heating output command	Command communication object = separated, Auxiliary heating = enabled	1 Bit	CR-T--	[1.1] DPT_Switch	283
Thermostat - Auxiliary heating and cooling output command	Command communication object = unique, Auxiliary heating = enabled	1 Bit	CR-T--	[1.1] DPT_Switch	283
Thermostat - Auxiliary heating disable	Auxiliary heating = enabled, Disabled from bus = yes	1 Bit	C-W---	[1.3] DPT_Enable	285

9.10.3 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

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	**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.	

Parameter name	*Conditions*	*Valori*
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.	
Building protection temp. setpoint [°C]		**36** [range 30 … 50]
Neutral zone [0,1 K]	Settings ⇒ Thermostat function = both heating and cooling Settings ⇒ Command Communication Object = separated Settings ⇒ Heating–cooling changeover = automatic Setpoint type = 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		**2-point hysteresis**, PWM (Pulse Width Modulation), continuous
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	Comm. Obj. No.
Thermostat - Input setpoint	Setpoint type = single	2 bytes	CRWTU	[9.1] DPT_Value_Temp	271
Thermostat - Comfort setpoint (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.1] DPT_Value_Temp	272
Thermostat - Standby setpoint (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.1] DPT_Value_Temp	274
Thermostat - Economy setpoint (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.1] DPT_Value_Temp	276
Thermostat - Building protection setpoint (cooling)		2 bytes	CRWTU	[9.1] DPT_Value_Temp	278
Thermostat - Cooling out command	Command Communication Object = separated Control type = 2 point hysteresis or PWM	1 bit	CR-T-	[1.1] DPT_Switch	282

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Thermostat - Cooling out command	Command Communication Object = separated Control type = continuous	1 byte	CR-T-	[5.1] DPT_Scaling	282
Thermostat - Auxiliary cooling output command	Auxiliary cooling = enabled	1 bit	CR-T-	[1.1] DPT_Switch	283
Thermostat - Auxiliary cooling disable	Auxiliary cooling = enabled Disabled from bus = yes	1 bit	C-W--	[1.003] enable	286
Thermostat - Anticondesation alarm	Anticondensation with probe or Active anticondensation = enabled	1 bit	CR-T-	[1.5] DPT_Alarm	11

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: 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: 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-	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. Flow

		point temperature (calculated by the thermostat)	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. Dew-point temperature [DPT 9.001 temperature °C].

Table 2 - 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 2).

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 2), you can insert the value used in the project (parameter Flow temperature (projected));
- if the flow temperature value is available (case 2b of Table 2), 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.

9.10.4 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.

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.	

Parameter name	*Conditions*	*Values*
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.	
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) ⇒ 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) ⇒ 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

Parameter name	*Conditions*	*Values*
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.
Thermostat - Fan continuous speed	Control type = continuous regulation	1 byte	CR-T-	[5.1] DPT_Scaling	287
Thermostat - Fan speed 1	Control type = 1, 2 or 3 speeds	1 bit	CR-T-	[1.1] DPT_Switch	293
Thermostat - Fan speed 2	Control type = 2 or 3 speeds	1 bit	CR-T-	[1.1] DPT_Switch	294
Thermostat - Fan speed 3	Control type = 3 speeds	1 bit	CR-T--	[1.1] DPT_Switch	295
Thermostat - Fan control disable	Disable ventilation from bus = yes	1 bit	C-W---	[1.2] DPT_Bool	296
Thermostat - Fan manual speed		1 byte	CRW-U	[5.10] DPT_Value_1_Ucount	290
Thermostat - Fan speed		1 byte	CR-T-	[5.10] DPT_Value_1_Ucount	288
Thermostat - Fan manual active status		1 bit	CRWTU	[1.11] DPT_State	291
Thermostat - Fan manual speed percentage		1 byte	CR-T-	[5.1] DPT_Scaling	289
Thermostat - Fan manual speed off status		1 bit	CR-T-	[1.11] DPT_State	292

9.10.4.1 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.

9.10.4.2 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.

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

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.

9.10.4.4 Remote fan speed modification

The communication objects shown in Figure 2 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.

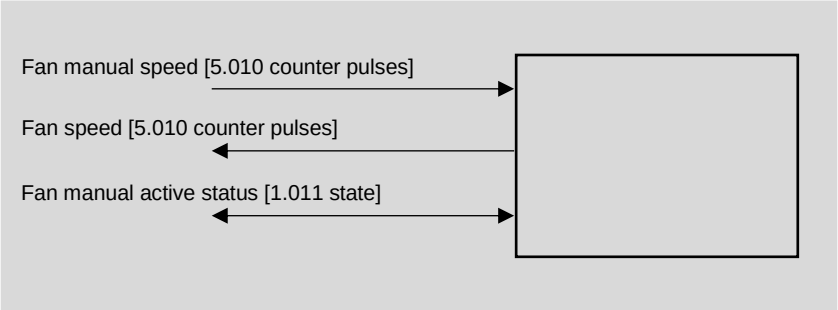


Figure 2 - Remote fan speed modification objects

The C.O. *Fan speed* allows to evaluate the actual fan speed; the C.O. *Fan manual active status* contains the information about automatic (=0, not active) or manual (=1, active) operating mode. By modifying the C.O. – *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. *Fan manual active status* (=0, not active).

Accepted values for C.O.s 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%

9.10.5 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.

9.10.5.1 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 dehumidification control from bus
- Signal from bus
- Dehumidification subordinated to temperature control
- Dehumidification start delay
- Sensible heat integration function
- Temperature difference for integration
- Hysteresis for integration

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 = cooling	**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.
Thermostat - Relative humidity setpoint for dehumidification		2 bytes	CRWTU	[9.7] DPT_Value_Humidity	300
Thermostat - Dehumidification command		1 bit	CR-T-	[1.1] DPT_Switch	302

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Thermostat - Dehumidification water battery command	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	1 bit	CR-T-	[1.1] DPT_Switch	303
Thermostat - 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--	[1.1] DPT_Switch	304
	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.				
Thermostat - Dehumidification control disable	Disable dehumidification control from bus = yes	1 bit	C-W--	[1.2] DPT_Bool	306

9.10.5.2 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

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 = cooling	**disabled** / cooling only
	Parameter that selects the humidification function.	

Parameter name	*Conditions*	*Values*
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.
Thermostat - Relative humidity setpoint for humidification	Humidification function ≠ disabled	2 bytes	CRWTU	[9.7] DPT_Value_Humidity	301
Thermostat - Humidification command	Humidification function ≠ disabled	1 bit	CR-T-	[1.1] DPT_Switch	305
Thermostat - Humidification control disable	Disable humidification control from bus = yes	1 bit	C-W--	[1.2] DPT_Bool	307

9.10.6 Comfort

9.10.6.1 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]

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.
Thermostat - Dew-point temperature	Dew-point temperature = enabled	2 bytes	CR-T-	[9.1] DPT_Value_Temp	297

9.10.7 Energy saving

In order to implement 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.*

9.10.7.1 Window open 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

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.
Thermostat - Windows contact sensor 1 (from bus)	Window contacts function = enabled	1 bit	C-WTU	[1.19] DPT_Window_Door	251
Thermostat - Windows contact sensor 2 (from bus)	Window contacts function = enabled	1 bit	C-WTU	[1.19] DPT_Window_Door	252

9.10.7.2 Presence sensors contacts

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), or EK-DF2-TP, EK-DG2-TP, EK-DH4-TP presence sensor.

The following condition has to be true:

- *External sensors (from bus) ⇒ Presence sensor 1 and/or Presence sensor 2 = enabled,* and
- *Settings ⇒ Setpoint type = absolute*

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.**
Thermostat - Presence sensor 1 (from bus)	Presence sensor function= enabled	1 bit	C-WTU	[1.18] DPT_Occupancy	253
Thermostat - Presence sensor 2 (from bus)	Presence sensor function= enabled	1 bit	C-WTU	[1.18] DPT_Occupancy	254

9.10.7.3 Card holder contact

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

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.**
Thermostat - Card holder contact (from bus)	Card holder function = enabled	1 bit	C-WTU	[1.18] DPT_Occupancy	255

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).

9.11 Logic functions

The virtual 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.

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.

In case of wrong 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.

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
	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	

Parameter name	*Conditions*	*Values*
	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-WTU	[1.1] DPT_Switch	308, 313, 318, 323, 328, 333, 338, 343
	X = 1, …, 8				
Logic function X, Input 2	Logic function X = enabled Logic object 2 = enabled	1 bit	C-WTU	[1.1] DPT_Switch	309, 314, 319, 324, 329, 334, 339, 344
	X = 1, …, 8				
Logic function X, Input 3	Logic function X = enabled Logic object 3 = enabled	1 bit	C-WTU	[1.1] DPT_Switch	310, 315, 320, 325, 330, 335, 340, 345
	X = 1, …, 8				
Logic function X, Input 4	Logic function X = enabled Logic object 4 = enabled	1 bit	C-WTU	[1.1] DPT_Switch	311, 316, 321, 326, 331, 336, 341, 346
	X = 1, …, 8				
Logic function X, Output	Logic function X = enabled At least one Logic object = enabled	1 bit	CR-T-	[1.1] DPT_Switch	312, 317, 322, 327, 332, 337, 342, 347
	X = 1, …, 8				

10 Appendix

10.1 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	7
Thermostat - Thermal generator lock	1 bit	C-W--	[1.5] DPT_Alarm	8
Thermostat - Room temperature control alarm	1 bit	CR-T-	[1.5] DPT_Alarm	10
Thermostat - Anticondesation alarm	1 bit	CR-T-	[1.5] DPT_Alarm	11
Thermostat - Alarm text	14 bytes	CR-T-	[16.0] DPT_String_ASCII	13
Bluetooth enable	1 bit	C-WTU	[1.3] DPT_Enable	14
Lock Bluetooth function	1 bit	C-W--	[1.3] DPT_Enable	15
Enable Fahrenheit display	1 bit	CRWT-	[1.3] DPT_Enable	19
Max display intensity percentage	1 byte	C-W--	[5.1] DPT_Scaling	20
Proximity - Lock command	1 bit	C-W--	[1.3] DPT_Enable	23
Proximity - Switching status *[type]*, object *n**	*See Tabel 9*	C-WTU	*See Tabel 9*	24, 25, 26, 27, 28, 29, 30, 31
Color (m) - Command *[tipo]*	*See Tabel 9*	C-WTU	*See Tabel 9*	32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43
Button (m) - Lock command	1 Bit	C-W--	[1.3] DPT_Enable	44, 60, 76, 92, 108, 124, 140, 156, 172, 188, 204, 220
Button (m) - State *[type]*, object *n**	*See Tabel 9*	C-WTU	*See Tabel 9*	45, …, 52, 61, …, 68, 77, …, 84, 93, …, 100, 109, …, 116, 125, …, 132, 141, …, 148, 157, …, 164, 173, …, 180, 189, …, 196, 205, …, 212, 221, …, 228
	** The O.C. numbers listed refer to the first of these 8 objects (for each of the inputs); the O.C.'s of subsequent objects are sequential. To obtain the O.C. number for the n-th object, simply add (n-1) to the numbers listed. E.g.: the O.C.'s associated with button 3 have numbers starting from 37. The number of the 5th O.C. associated with that input will therefore be 37+ (5-1) = 41.*			
Button (m) - Switching command	1 Bit	C-WTU	[1.1] DPT_Switch	53, 69, 85, 101, 117, 133, 149, 165, 181, 197, 213, 229
Button (m) - Dimming command value	1 Byte	C-W--	[5.1] DPT_Scaling	54, 70, 86, 102, 118, 134, 150, 166, 182, 198, 214, 230
Button (m) - Dimming status value	1 Byte	CRWTU	[5.1] DPT_Scaling	55, 71, 87, 103, 119, 135, 151, 167,

Object name	Size	Flags	DPT	No. Comm. Obj.
				183, 199, 215, 231
Button (m) - Color temperature command	1 Byte	C-W--	[5.1] DPT_Scaling	56, 72, 88, 104, 120, 136, 152, 168, 184, 200, 216, 232
Button (m) - Color temperature status	1 Byte	CRWTU	[5.1] DPT_Scaling	57, 73, 89, 105, 121, 137, 153, 169, 185, 201, 217, 233
Button (m) - Dedicated stop command	1 Bit	C–T-	[1.17] DPT_Trigger	53, 69, 85, 101, 117, 133, 149, 165, 181, 197, 213, 229
Button (m) - Move up / down command	1 Bit	C–T-	[1.8] DPT_UpDown	56, 72, 88, 104, 120, 136, 152, 168, 184, 200, 216, 232
Button (m) - Stop-step up / down command	1 Bit	C–T-	[1.7] DPT_Step	55, 71, 87, 103, 119, 135, 151, 167, 183, 199, 215, 231
Button (m) - Scene number	1 Byte	C-WTU	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	57, 73, 89, 105, 121, 137, 153, 169, 185, 201, 217, 233
Temperature value	2 Byte	CR-T-	[9.1] DPT_Value_Temp	234
Temperature threshold 1 - switch	1 Bit	CR-T-	[1.1] DPT_Switch	235
Temperature threshold 1 - Lock	1 bit	C-W--	[1.1] DPT_Switch	236
Temperature threshold 1 – Value (from bus)	2 bytes	C-W--	[9.1] DPT_Value_Temp	237
Temperature threshold 2 - switch	1 Bit	CR-T-	[1.1] DPT_Switch	238
Temperature threshold 2 - Lock	1 bit	C-W--	[1.1] DPT_Switch	239
Temperature threshold 2 - Value (from bus)	2 bytes	C-W--	[9.1] DPT_Value_Temp	240
Thermostat - Room temperature (from bus)	2 bytes	C-WTU	[9.1] DPT_Value_Temp	241
Thermostat - Humidity (2 byte, from bus)	2 bytes	C-WTU	[9.7] DPT_Value_Humidity	242
Thermostat - Humidity (1 byte, from bus)	1 byte	C-WTU	[5.1] DPT_Scaling	243
Thermostat - Antistratification temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	246
Thermostat - Outdoor temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	247
Thermostat - Coil temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	248
Thermostat - Floor temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	249
Thermostat - Flow temperature (from bus)	2 bytes	C-WTU	[9.001] DPT_Value_Temp	250
Thermostat - Windows contact sensor 1 (from bus)	1 bit	C-WTU	[1.019] DPT_Window_Door	251
Thermostat - Windows contact sensor 2 (from bus)	1 bit	C-WTU	[1.019] DPT_Window_Door	252
Thermostat - Presence sensor 1 (from bus)	1 bit	C-WTU	[1.018] DPT_Occupancy	253
Thermostat - Presence sensor 2 (from bus)	1 bit	C-WTU	[1.018] DPT_Occupancy	254
Thermostat - Card holder contact (from bus)	1 bit	C-WTU	[1.018] DPT_Occupancy	255
Thermostat - Anticondensation (from bus)	1 bit	C-WTU	[1.001] DPT_Switch	256
Thermostat - Weighted temperature	2 bytes	CR-T-	[9.001] DPT_Value_Temp	257
Thermostat - Heating/cooling status out	1 bit	CR-T-	[1.100] DPT_Heat_Cool	258
Thermostat - Heating/cooling status in	1 bit	C-W--	[1.100] DPT_Heat_Cool	259
Thermostat - HVAC mode in	1 byte	C-W--	[20.102] DPT_HVACMode	260
Thermostat - HVAC forced mode in	1 byte	C-W--	[20.102] DPT_HVACMode	261
Thermostat - HVAC mode out	1 byte	CR-T-	[20.102] DPT_HVACMode	262
Thermostat - HVAC manual mode	1 byte	C-WTU	[20.102] DPT_HVACMode	263
Thermostat - Chrono active status	1 bit	CRWT-	[1.11] DPT_State	264

Object name	Size	Flags	DPT	No. Comm. Obj.
Thermostat - Building protection HVAC mode active	1 bit	CR-T-	[1.11] DPT_State	266
Thermostat - Actual setpoint	2 bytes	CR-T-	[9.1] DPT_Value_Temp	268
Thermostat - Manual setpoint	2 bytes	C-W--	[9.1] DPT_Value_Temp	269
Thermostat - Manual setpoint active status	1 bit	CRWTU	[1.11] DPT_State	270
Thermostat - Input setpoint	2 bytes	CRWTU	[9.1] DPT_Value_Temp	271
Thermostat - Comfort setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	271
Thermostat - Comfort setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	272
Thermostat - Standby setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	273
Thermostat - Standby setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	274
Thermostat - Economy setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	275
Thermostat - Economy setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	276
Thermostat - Building protection setpoint (heating)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	277
Thermostat - Building protection setpoint (cooling)	2 bytes	CRWTU	[9.1] DPT_Value_Temp	278
Thermostat - Disable room temperature controller	1 bit	C-W--	[1.3] DPT_Enable	279
Thermostat - Room temperature controller status	1 bit	CR-T-	[1.3] DPT_Enable	280
Thermostat - Heating out command	1 bit	CR-T-	[1.1] DPT_Switch	281
Thermostat - Cooling out command	1 byte	CR-T-	[5.1] DPT_Scaling	281
Thermostat - Heating and cooling out command	1 bit	CR-T-	[1.1] DPT_Switch	281
Thermostat - Heating and cooling out command	1 byte	CR-T-	[5.1] DPT_Scaling	281
Thermostat - Cooling out command	1 bit	CR-T-	[1.1] DPT_Switch	282
Thermostat - Cooling out command	1 byte	CR-T-	[5.1] DPT_Scaling	282
Thermostat - Auxiliary heating output command	1 bit	CR-T-	[1.1] DPT_Switch	283
Thermostat - Auxiliary heating and cooling output command	1 bit	CR-T-	[1.1] DPT_Switch	283
Thermostat - Auxiliary cooling output command	1 bit	CR-T-	[1.1] DPT_Switch	284
Thermostat - Auxiliary heating disable	1 bit	C-W--	[1.3] DPT_Enable	285
Thermostat - Auxiliary cooling disable	1 bit	C-W--	[1.3] DPT_Enable	286
Thermostat - Fan continuous speed	1 byte	CR-T-	[5.1] DPT_Scaling	287
Thermostat - Fan speed	1 byte	CR-T-	[5.10] DPT_Value_1_Ucount	288
Thermostat - Fan manual speed percentage	1 byte	CR-T-	[5.1] DPT_Scaling	289
Thermostat - Fan manual speed	1 byte	CRW-U	[5.10] DPT_Value_1_Ucount	290
Thermostat - Fan manual active status	1 bit	CRWTU	[1.11] DPT_State	291
Thermostat - Fan manual speed off status	1 bit	CR-T-	[1.11] DPT_State	292
Thermostat - Fan speed 1	1 bit	CR-T-	[1.1] DPT_Switch	293
Thermostat - Fan speed 2	1 bit	CR-T-	[1.1] DPT_Switch	294
Thermostat - Fan speed 3	1 bit	CR-T-	[1.1] DPT_Switch	295
Thermostat - Fan control disable	1 bit	C-W--	[1.2] DPT_Bool	296
Thermostat - Dew-point temperature	2 bytes	CR-T-	[9.1] DPT_Value_Temp	297
Thermostat - Relative humidity setpoint for dehumidification	2 bytes	CRWTU	[9.7] DPT_Value_Humidity	300
Thermostat - Relative humidity setpoint for humidification	2 bytes	CRWTU	[9.7] DPT_Value_Humidity	301

Object name	Size	Flags	DPT	No. Comm. Obj.
Thermostat - Dehumidification command	1 bit	CR-T-	[1.1] DPT_Switch	302
Thermostat - Dehumidification water battery command	1 bit	CR-T-	[1.1] DPT_Switch	303
Thermostat - Dehumidification integration control	1 bit	CR-T-	[1.1] DPT_Switch	304
Thermostat - Humidification command	1 bit	CR-T-	[1.1] DPT_Switch	305
Thermostat - Dehumidification control disable	1 bit	C-W--	[1.2] DPT_Bool	306
Thermostat - Humidification control disable	1 bit	C-W--	[1.2] DPT_Bool	307
Logic function X, Input 1	1 bit	C-WTU	[1.1] DPT_Switch	308, 313, 318, 323, 328, 333, 338, 343
Logic function X, Input 2	1 bit	C-WTU	[1.1] DPT_Switch	309, 314, 319, 324, 329, 334, 339, 344
Logic function X, Input 3	1 bit	C-WTU	[1.1] DPT_Switch	310, 315, 320, 325, 330, 335, 340, 345
Logic function X, Input 4	1 bit	C-WTU	[1.1] DPT_Switch	311, 316, 321, 326, 331, 336, 341, 346
Logic function X, Output	1 bit	CR-T-	[1.1] DPT_Switch	312, 317, 322, 327, 332, 337, 342, 347

Table A1. Dimensions e DPT for C.O.s with independent inputs:

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

10.2 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.

10.3 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