ekinex

CONTROL YOUR LIVING SPACE

Application manual

8-fold KNX
analog / binary input module
EK-CA1-TP

Contents

Revision	Modifications	Author	Date
1.0	First release	R. Rocco	08/02/2014
2.0	Update for new device release A2.0	G. Schiochet	07/05/2021

1. Scope of the document

This application manual describes application details for the A2.0 release of the ekinex® KNX analog/binary input EK-CA1-TP (8 inputs).

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
Technical datasheet	STEKCA1TP_EN.pdf	-	A2.0	05 / 2021
Application manual	MAEKCA1TP_EN.pdf	-		
Application program	APEKCA1TP##.knxprod	-		

2. Product description

The ekinex® binary input EK-CA1-TP is an S-mode KNX modular device for rail mounting that allows to connect switches and sensors of conventional type (not communicating natively on the KNX bus), equipped with potential-free contacts, to the KNX bus.

Through the binary input module is possible to employ normal switches, pushbuttons and sensors or binary signals made available by other devices to switch and control KNX bus functions. The device can be used as follows:

- up to 8 independent single inputs, e.g. for the connection of conventional control points or sensors to control loads via the KNX bus;
- up to 4 individually configurable inputs for NTC temperature probes;
- 4 independent 2-input coupled channels, e.g. for the connection of conventional double pushbuttons for the control of dimmer or motorized drives.

As an alternative to programming as a digital input, each of the inputs 1B, 2B, 3B, 4B can be programmed as an analog input for:

- temperature measurement of the ambient air mass using a passive temperature probe (NTC 10 kΩ at 25 ° C) to be connected to the input with the possibility of sending the value on the bus;
- 2-point (ON / OFF type) or proportional (PWM or continuous) room temperature regulation.

The device has 2 conduction modes (heating and cooling) with switching via bus and 4 operating modes (comfort, standby, economy and building protection) with separate setpoints for heating and cooling operation. There is an automatic switching function of the operating mode depending on the presence or opening of windows and the adjustment can be done by means of the weighted average between two temperature values.

Finally, it is possible to configure up to 4 logic functions (AND, OR or XOR) with 4 logic objects in input each.

The device is equipped with an integrated KNX bus communication module and is designed for rail mounting in distribution boards.

The device basically receives an input signal and converts it into a corresponding telegram sent on the bus; the telegram sent by the device is received and processed by one or more KNX actuators.

The device is powered by the KNX bus line with a 30 VDC SELV voltage and does not require auxiliary power; all required operation voltages for the input channels are produced inside the device.

2.1 Input action

Each of the inputs can correspond to an action or a physical button of the device. These actions, in relation to a given Input, will be indicated with the letters A and B.

By pressing one side of an input, the device sends the telegram (or sequence) associated with it during the programming phase on the KNX bus.

In the most common case, for example, one side of an input could send an "ON" status telegram for a light point, while the other side could send the "OFF" status telegram. Other examples of typical applications are the increase and decrease in brightness of a lamp controlled by a dimmer unit, or the raise / lower commands for a roller shutter or a motorized awning and so on.

The two actions associated with an input can also be programmed to activate exactly the same function, thus allowing you to use the entire activation surface of the Input as if it were a single button.

2.2 Temperature sensors and room controllers

By parameterizing the input as a probe via ETS, the device allows to enable and configure up to four independent room controllers.

For further technical information, please also refer to the product datasheet STEKCA1TP_EN.pdf available on the ekinex website www.ekinex.com.

3. Switching, display and connection elements

The device is equipped with:

- a programming pushbutton and a programming LED
- membrane pushbuttons
- LEDs for input status indication
- terminals for connecting the inputs
- terminals for connecting the KNX bus line

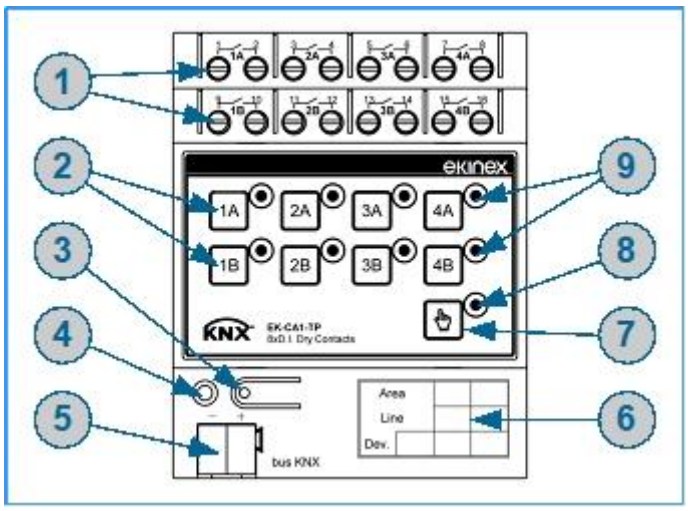


Figure 1 - Switching, display and connection elements

1) Terminal blocks for input channels	6) Field for physical address
2) Pushbuttons for forced operation of the inputs	7) Pushbutton for toggling manual / automatic mode
3) Programming pushbutton	8) LED for indication manual / automatic mode
4) Programming LED	9) LEDs for status indication of input channels
5) Terminal block for KNX bus line	

Input signals are normally taken from the terminal blocks; the switch devices to be connected must be capable of supplying a potential-free contact, either Normally Open or Normally Closed.

Voltage level signals (e.g. 24V signals) are not compatible with the device; a separation relay must be employed in case there is the need to interface such signals.

For convenience of operation, inputs can also be given manually by the user by means of the membrane buttons on the top panel. A pushbutton allows to switch to manual mode and back.

The status of all inputs (either from the terminal block or, in manual mode, from the buttons) is displayed through LEDs on the panel.

4. Configuration

The exact functionality of the device depends on the software settings.

In order to configure and commission the device you need ETS5 or later releases and the ekinex® application program **APEKCA1TP.knxprod** which can be downloaded from the ekinex website www.ekinex.com.

The application program allows the configuration of all working parameters for the device.

The device-specific application program has to be loaded into ETS or, as alternative, the whole ekinex® product database can be loaded; at this point, all the instances of the selected device type can be added to the project.

For every single device, ETS allows to set the operating parameters individually for each input as described in detail in the following chapters.

The configuration can, and usually will, be performed completely offline; the actual transfer of the programmed configuration to the device takes place in the commissioning phase.

Product code	EAN	No. of inputs	ETS application software (## = release)	Communication objects (max nr.)	Group adresses (max nr.)
EK-CA1-TP	8018417180958	8	APEKCA1TP##.knxprod	430	254

Configuration and commissioning of KNX devices require specialized skills. To acquire these skills, you should attend training courses at a training center certified by KNX.

For further information: www.knx.org

5. Commissioning

After the device has been configured within the ETS project according to user requirements, the commissioning of the device requires the following activities:

- electrically connect the device, as described in the product datasheet, to the bus line on the final network or through a purposely setup network for programming;
- apply power to the bus;
- switch the device operation to programming mode by pressing the programming pushbutton located on the front side of the housing. In this mode of operation, the programming LED is turned on steady;
- upload the configuration (including the physical address) to the device with the ETS program.

At the end of the upload, the operation of the device automatically returns to normal mode; in this mode the programming LED is turned off. Now the device is programmed and ready for use on the bus.

Device reset

To reset the device, remove the connection to the bus network by extracting the bus terminal from its seat. Keeping the programming button pressed, reinsert the bus terminal in its seat; the programming LED flashes quickly. Release the programming button and extract the terminal again; the reset has been carried out. At this point it is necessary to carry out the addressing and configuration of the device again via ETS.

Warning! The reset resets the device to the factory delivery state.

The addressing and the value of the parameters set during the configuration phase are lost!

6. 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 startup 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: 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.

6.1 Offline operation

A fully unprogrammed device causes no activity on the bus; it can be operated in manual mode so that the inputs are set as desired, but the input setting does not have effect on any other device.

6.2 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.

In the following chapters, for convenience, the inputs will sometimes be referred to as "buttons" or "switches" depending on the fact that one of the most frequent uses is interfacing with these control elements.

6.2.1 Software working cycle

The software working cycle can be described as follows:

- Handle input contacts or user pushbutton presses and generate bus telegrams according to the assigned functions;
- Implement input / 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.

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.

6.2.2 Inputs

6.2.2.1 Input types

The status of digital inputs corresponds to the status of connected physical contacts.

The device may be configured in two modes, in order to be interfaced to different contacts: these modes are named **NO** (normally open) and **NC** (normally closed).

Usually, the mode denomination clearly matches the type of the contact of the electrical device used at the input. From a logical point of view, this mode affects the interpretation of the “active” and “inactive” state of an input as follows:

- In NO mode, an open connection between the terminals (open contact) is associated to the inactive state, while a closed contact is associated to the active state;

- in NC mode, an open connection between the terminals (open contact) is associated to the active state, while a closed contact is associated to the inactive state.

6.2.2.2 Input events

The device recognizes two types of input events: "*close / open contact*" and "*short / long press*".

The first event type is a simple logical value change: "OPEN" is an alias for "inactive", whereas "CLOSE" means "active state".

> *It is very important to stress that the words "OPEN" and "CLOSE", although standard terms for input status conditions, are to be interpreted from a logic point of view, and that they are not to be confused with the physical contact status as used in the description of "NO" and "NC" input types.*
>
> *In other words, for example, a NO contact in active position is electrically AND logically CLOSEd, whereas a NC contact in active position is electrically open, but logically CLOSEd.*

The second type of event that can be associated with an input is the Short or Long Press; the term "press" is typically referred to user activated pushbuttons, although it also applies to signals originating from contacts of other devices.

The distinction is as follows:

- If an input remains active for a period shorter than a defined time duration, upon release a "Short Press" event is generated;
- If the input remains active for longer than the defined time duration, at the duration time point a "Long Press" event is generated. Thereafter, the input can remain active for as long as desired, and no more events are generated either during the rest of activation or at release (next event will occur after next activation).

Please refer to time diagrams in following figures for an illustration of the difference between these events.

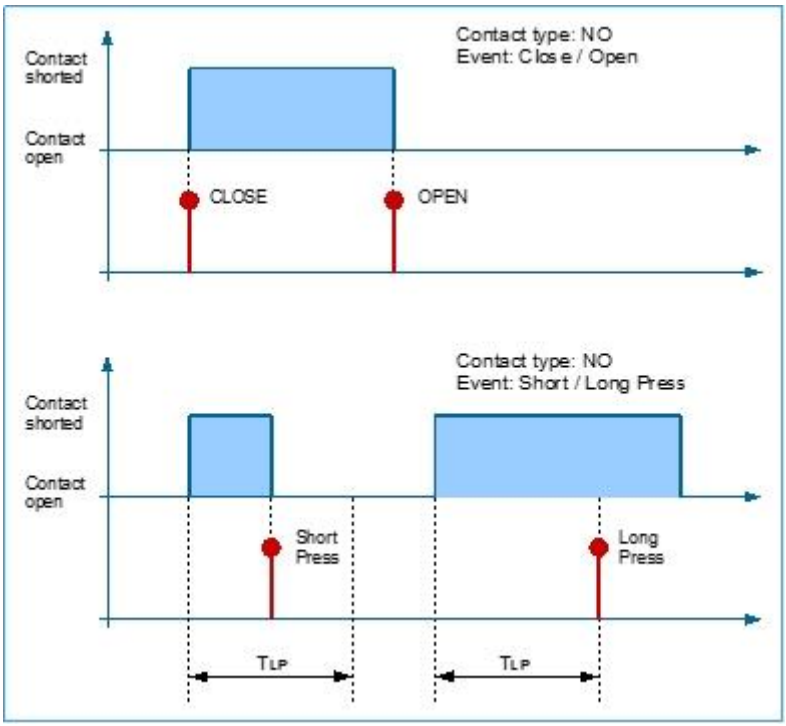


Figure 2 - time diagrams for Normally Open (NO) mode

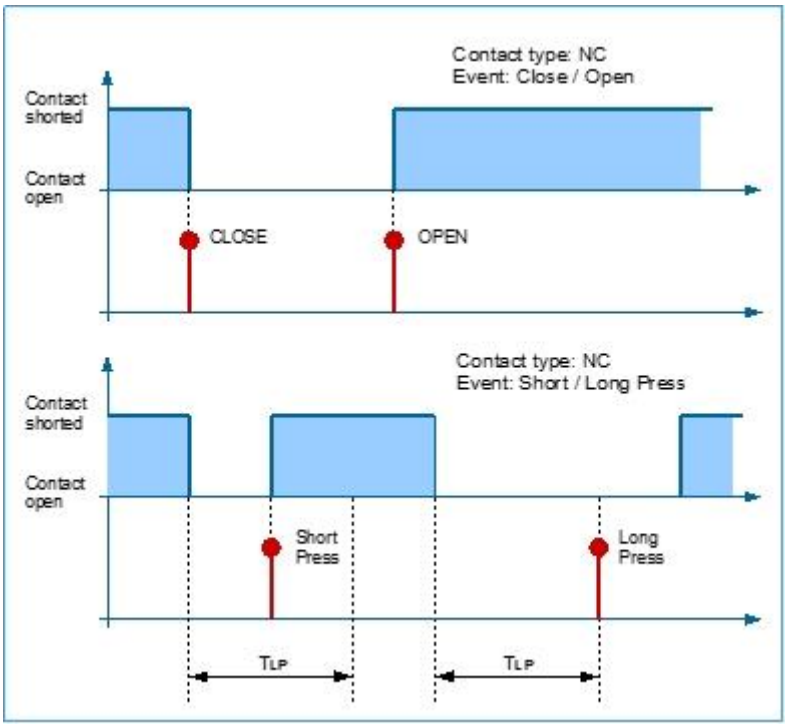


Figure 3 - time diagrams for NC (Normally Closed) modes

6.2.2.3 Lock function

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

When in a locked state, the input / channel is effectively disabled; the locked state can be deactivated by sending another telegram.

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.

6.2.2.4 Generic temperature value acquisition

The analog inputs can be used to acquire a generic temperature value by means of a traditional NTC type probe (10 kΩ at 25 ° C). The measured value can be sent on the bus and used by other KNX devices, for example for displaying or calculating a weighted average by a room thermostat.

6.2.3 State variables

6.2.3.1 State variables (Communication objects)

The variables that are 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.

This obviously implies that, if the value of a communication objects changes due to the effect of a bus telegram, the corresponding change will register in the device, according to its associated flags.

6.2.3.2 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 behavior and its own associated value set.

6.2.3.3 Cyclic sending

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 behavior, 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.

6.2.4 Input coupling

The 8 inputs of the device can be considered, and used, as independent; however, due to the physical structure of the device and the nature of the functions it most frequently performs, these inputs can be naturally grouped in pairs. In this case, a pair can consist of two inputs located above / below on the terminal block, indicated by the labels 1A and 1B on the front of the device, or Channel 1 - input1 and input2 on the ETS application.

Since the input pairs are numbered 1 to 4, the inputs are referred to as 1A / 1B for pair 1, 2A / 2B for pair 2, and so on. For consistency, the same indication is used regardless of whether all or some of the inputs are coupled.

In order to specify channel pairings, each input can be configured in two ways: single mode and coupled mode.

- In *single mode*, each input operates independently, has its own parameters and communication objects. This is the mode of operation described so far.
- In *coupled mode*, 2 inputs operate logically grouped under the same channel in order to perform a common functionality. Only inputs belonging to the same channel can be coupled, therefore the only combinations allowed for coupling are 1A with 1B, 2A with 2B, and so on.

It is possible to configure some of the inputs in single mode and the others in coupled mode, with the pairing constraints just described.

It should be noted that there is in fact a third way of configuring a pair of inputs, which is almost a cross between those described (even if in the application program it appears as a variation of the independent or single mode). The second input of a pair, for example 1B, can be configured in such a way as to have exactly the same function as the first corresponding one. In this way, the two inputs of a pair are effectively used "in parallel" as a single control (momentary button, switch or other depending on the programming).

Single and coupled modes have a similar functionality, but differ for the configuration. Therefore, they will be described separately.

In the rest of the document, as in the application program, for clarity the term Channel will be used to indicate the pair of inputs. For single inputs (not coupled) we will speak generically of “inputs”.

6.2.4.1 Independent or single input mode

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

1. *Send values or sequences*
   An event triggers the transmission on the bus of configurable values or sequence of values. These values can be of a logical type or a numerical type with a different size.

   A sequence of values can be made of up to 8 communication objects of different value types; time delays can set between values in the sequence.

2. *Dimmer control*
   This mode is intended to be used with dimming actuators for the control of lighting devices.

   The functionality is triggered on short press and long press events. On short press events, the device sends on/off telegrams to the dimming actuator; on long press events, the dimming percentage is varied up or down until the button is released.

3. *Shutter or Venetian blind control*
   This mode is intended to be used together with actuators for the control of motorized blinds, shutters and similar devices. These actuators have functions for blind opening and closing; two movement types are selectable, i.e. continuous movement and stepwise movement. On input events, the device sends operation telegrams to the actuators.

   The operation is configurable through following parameters:

   - If *toggle* mode is enabled, on each activation of the same input the movement direction is inverted; if it is disabled, the movement direction is fixed and it can be set to “up” or “down”.
   - If *blinds* mode is enabled, the device sends “full movement” telegrams on long press and “step” telegrams on short press; if it is disabled, the device sends “full movement” telegrams on long press and “stop” telegrams on short press.

4. *Scene function output*
   This mode is intended to be used together with several KNX actuators that support using a scene function; this function allows to store and recall 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 preset scene on short press, and store current setting as scene value on long press
   - Activate two different scenes on long and short press.

5. *Pulse counter*
   In this mode the device can count the number of commutations at an input channel. The counter value can be read from a communication object which can be cyclically sent on specified time period. It is possible to set the counter’s value type and maximum reachable value.

6.2.4.2 Coupled Input mode

Each pair of coupled inputs can be configured for one of following different features (only the differences from the single mode are highlighted):

1. *Switch control*
   Both inputs in a pair are bound to the same communication object; unlike single mode, the object can only be of the 1-bit type (on-off), therefore building a conventional switching behaviour.

   The user can configure which of the two inputs sets the “off” or resp. “on” value.

2. *Dimmer control*
   The functionality is triggered on short press and long press events of the inputs in the pair.

   The user can configure which of the two inputs sets the “up” or resp. “down” value.

   On short press events, the input configured as “up” sends an “on” switching telegram to the dimming actuator, while the “down” input sends an “off” telegram.

   On long press events, the dimming percentage is varied up or down until the button is released.

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

The dimmer control type is essentially based on a 4-bit communication object, whose data has the following forrmat:

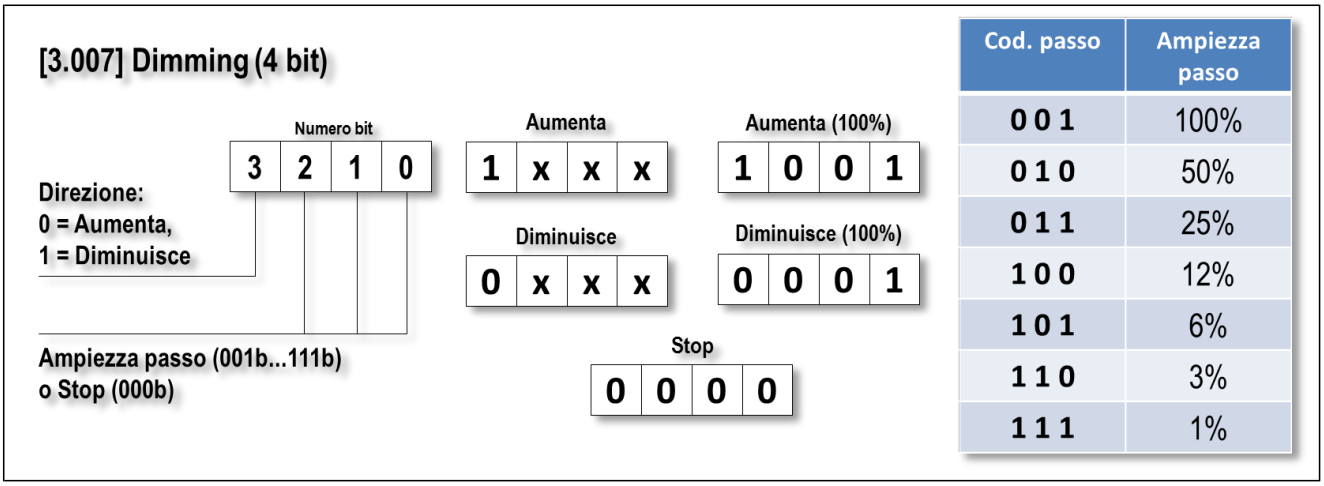


Figure 4 - Dimmer type CO

The transmission of telegrams containing data of such format tells the actuator to perform an increase or a decrease, by an amplitude equal to the specified step, or to stop an ongoing variation.

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

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

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

The defined operations and related commands are the following ones (please see Figure 5

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

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

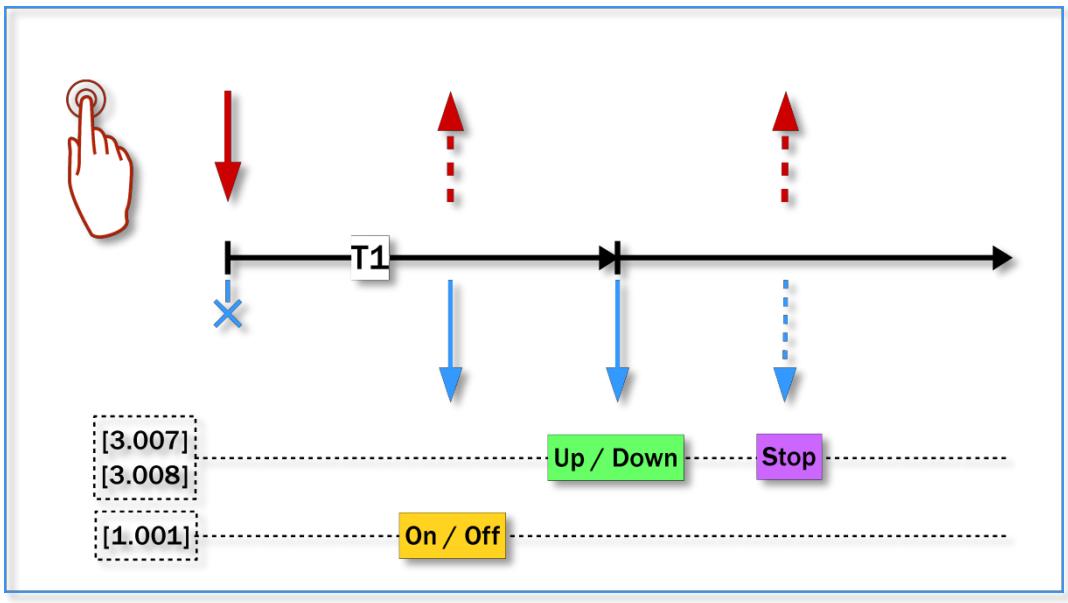


Figure 5 - Dimmer mode command sequence

6.2.4.3 Shutter or Venetian blind control

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

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

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

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

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

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

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

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

In the simplest version, on command side:

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

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

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

In case of ekinex input devices, two modes are available – indicated as "Shutter" and "Venetian blind" based on their typical destination – which are illustrated in the following figures.

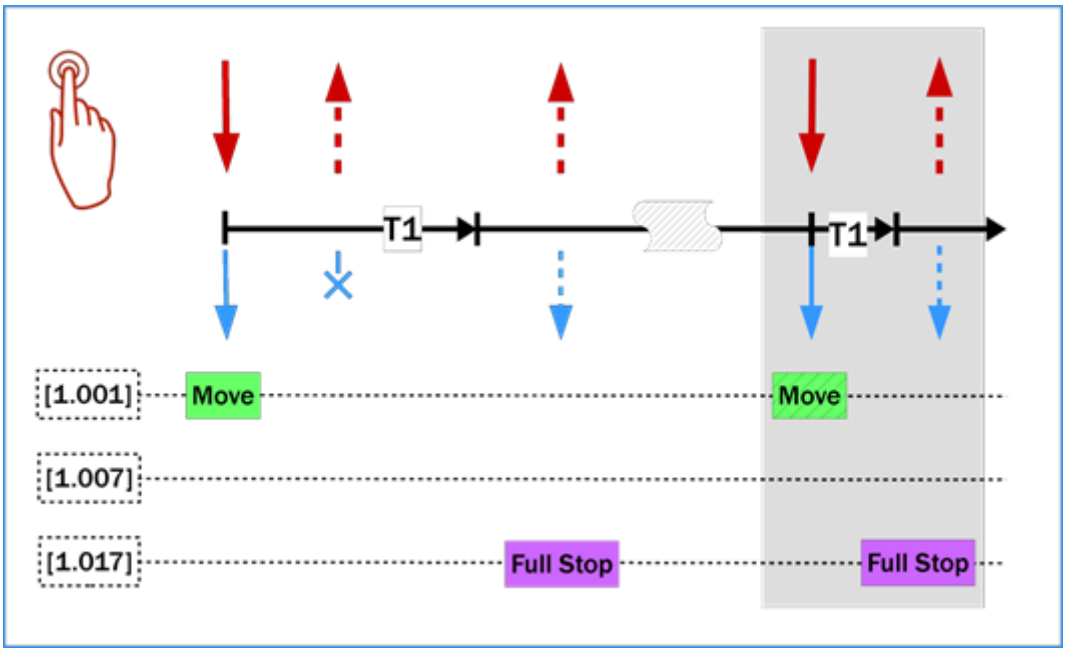


Figure 6 - "Shutter" mode command sequence

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

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

If the input is pressed with a long press, when it is released – which will be in correspondence with the desired position – the shutter will stop.

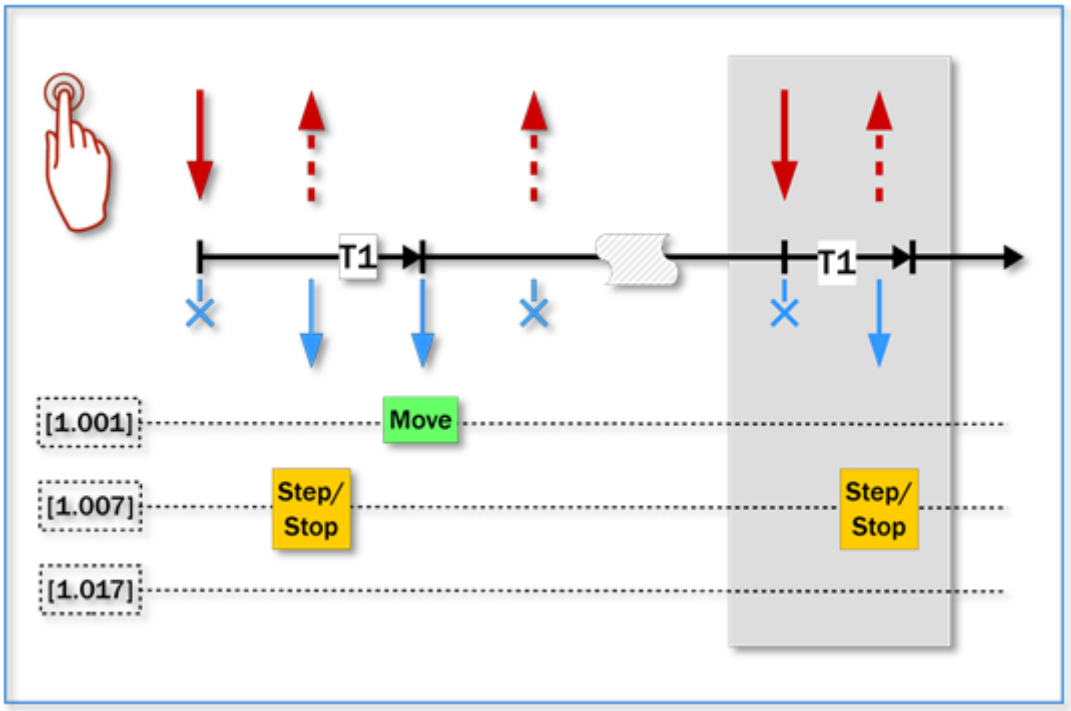


Figure 7 - "Venetian blind" mode command sequence

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

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

6.2.5 Logic functions

The KNX interface provides some useful logic functions (AND, OR, NOT and XOR) in order to implement complex functions in the building automation system. The following elements are available and configurable:

- 4 channels of logic 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

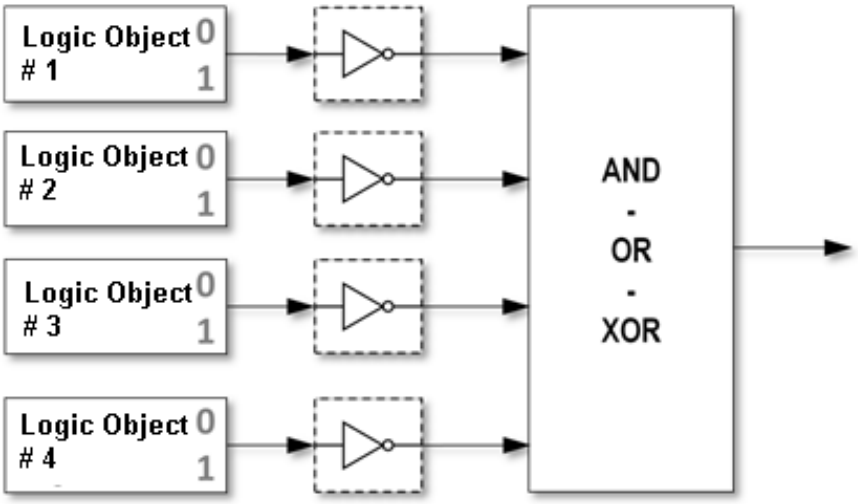


Figure 8 - combination logic function

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 next figures show the basic logic functions, assuming 2 inputs and only one logic communication object.

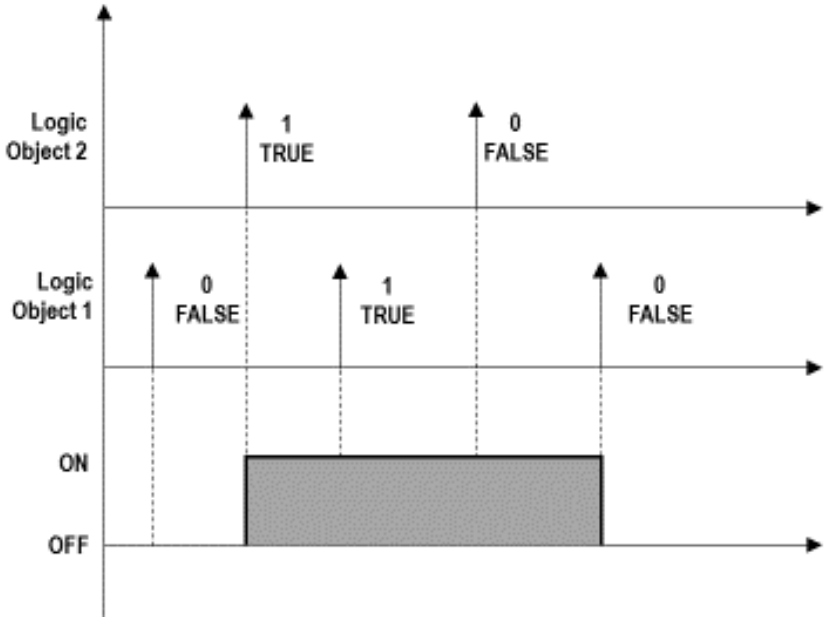


Figure 9 - logic function OR

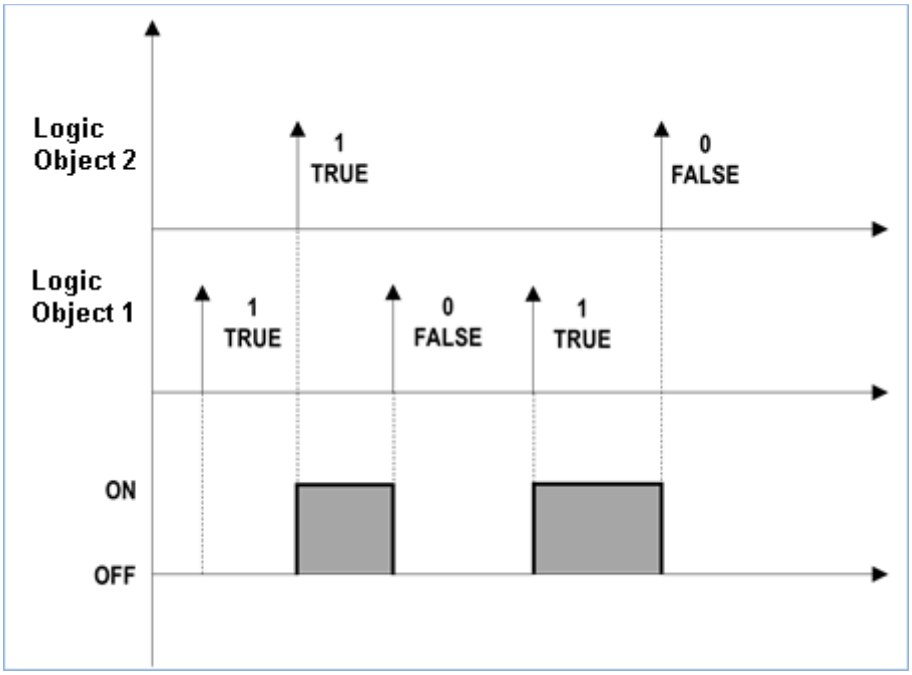


Figure 10 - Logic AND function

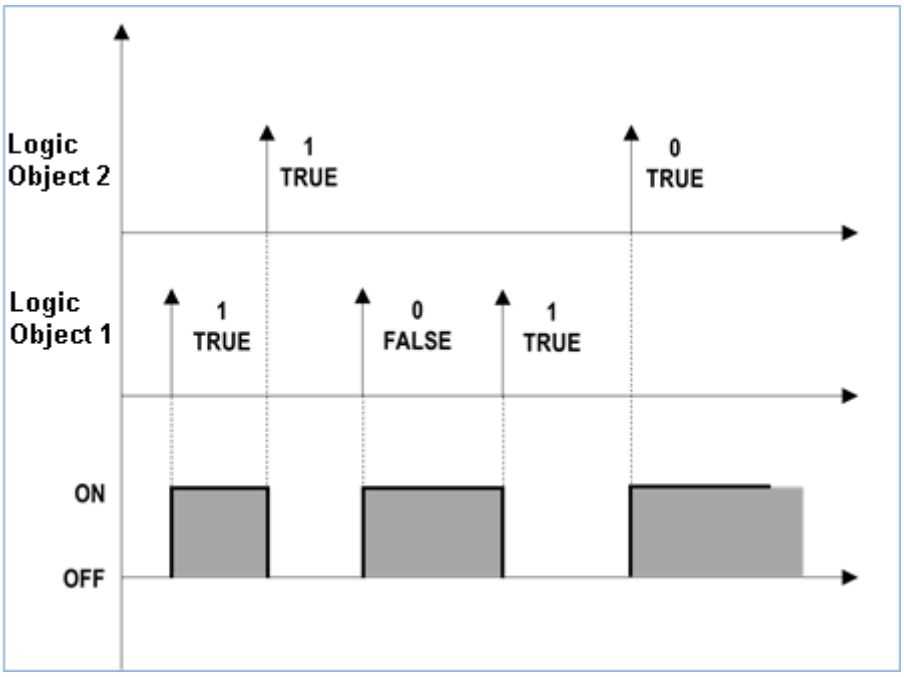


Figure 11 - Logic function Exclusive OR (XOR)

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.

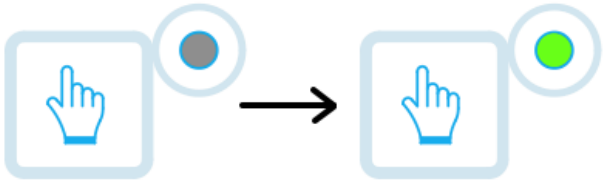
The communication function representing the logic function output is sent on the bus on event of change; alternatively, a cyclic sending can be set.

6.2.6 Manual operation

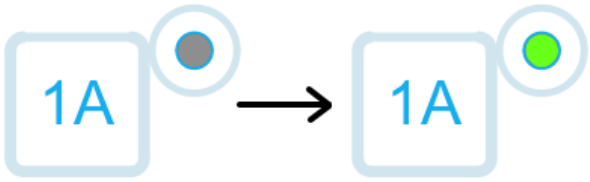
The manual operation works as an alternative to the physical inputs. When the manual operation is activated, any signal changes coming from the physical inputs will be not considered, and the device can only be operated via the membrane pushbuttons on the front side of the device. If group addresses have been assigned, telegrams will be sent on the bus. It is possible to control a channel through the membrane keypad. Each pushbutton press sends a telegram like they were acquired physically. The LED of each pushbutton shows if the contact is closed.

For switching the device to the manual operations mode proceed according to the following steps:

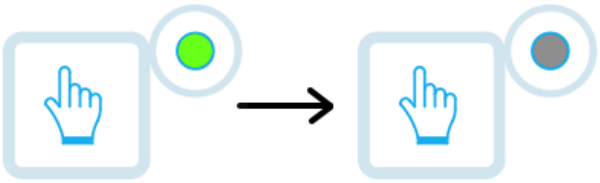
1) Press the manual mode pushbutton. In normal operation the LED is turned off. When the LED turns on, the whole membrane keypad is activated and the manual operations are allowed.



2) Press the pushbutton of the keypad corresponding to the channel that has to be operated (in the example: 1A).



3) After the operation, turn off the manual mode by pressing again the manual mode pushbutton. After switching off manual operations the input value will be accorded to the physical input.



6.3 Room Controller

The room air temperature control is performed thanks to the intercept valve(s) on the heat exchange coil(s), with an ON/OFF or PWM control algorithm. In order to control those valves, electrothermal actuators and/or servomotors can be used

6.3.1 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}).

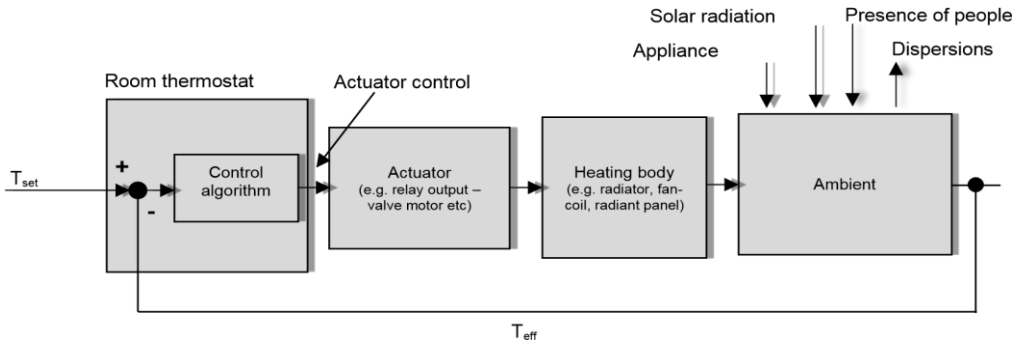


Figure 12 - Generic scheme for room temperature regulation

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

6.3.2 Two-point control with hysteresis

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

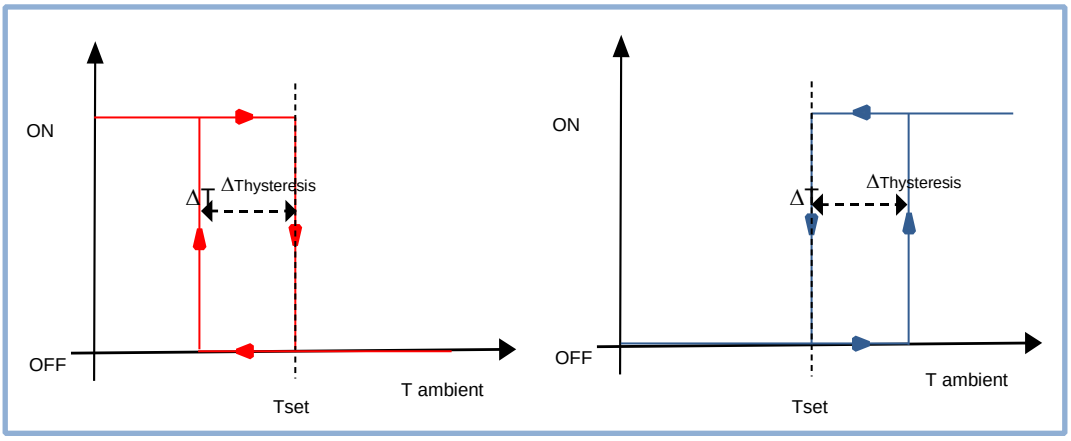


Figure 13 - Two-point control with hysteresis

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

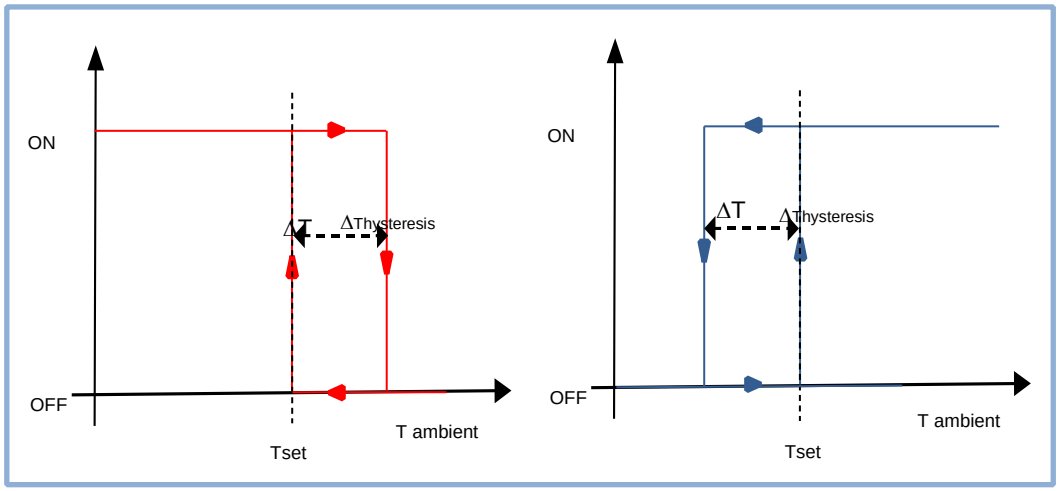


Figure 14 - Two-point control with hysteresis

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 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 multifunctionality 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 window opening for air refreshment; • Circuit deactivation when desired temperature is reached;
- Flow temperature reduction in case of partial load.

6.3.3 PWM Proportional-Integral control

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

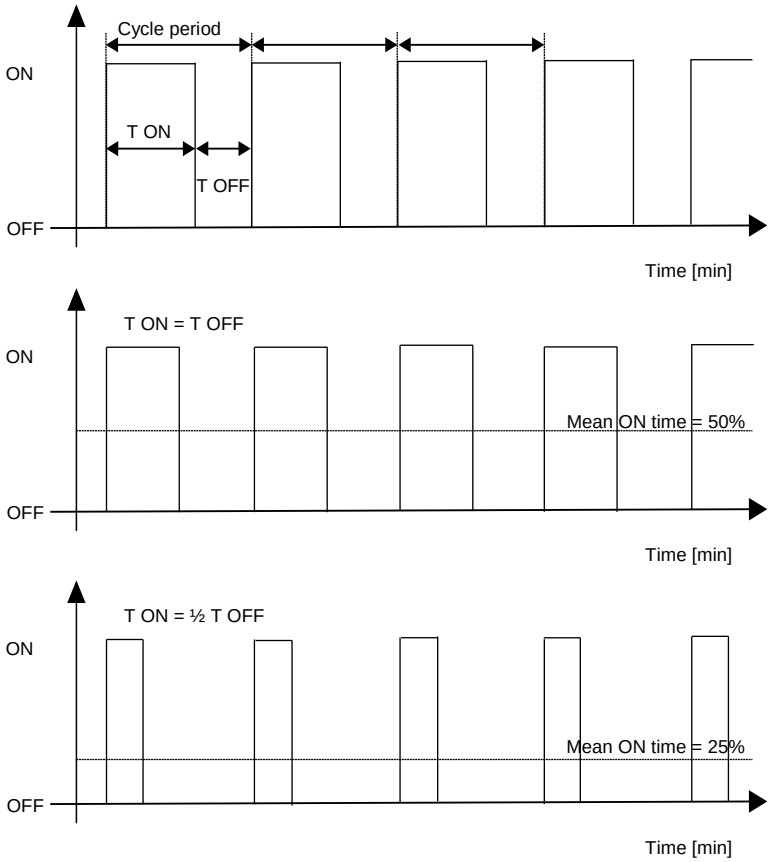


Figure 15 - PWM control

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.

6.3.4 Setpoint management

The device is not equipped with a local interface to control the integrated room thermostat, therefore the temperature setpoint modifications need to be performed by another KNX device (supervisor) and sent to this device through communication objects.

Three setpoint management modes are foreseen:

- Single setpoint;
- Relative setpoints;
- Absolute setpoints

.

6.3.4.1 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.

In case of configuration of a temperature control in both heating and cooling, it is necessary that the supervisor device also updates the conduction mode input object (Heating / cooling status in, [1.100] DPT_Heat_Cool) to switch coherently the type of action of the regulator.

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).

6.3.4.2 Relative setpoints mode

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 device the current operating mode (comm. obj. *HVAC mode in* [20.102] DPT_HVAC Mode). The default value for *HVAC mode in* corresponds to the comfort setpoint value.

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

6.3.4.3 Absolute Setpoint mode

In this mode, 3 communication objects are exposed, for each conduction mode:

- Comfort setpoint;
- Standby setpoint;
- Economy setpoint;
- Building protection setpoint.

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

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

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

6.3.5 Operating modes

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

- Temperature setpoint
- Building protection setpoint

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

In Relative and 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.

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

6.3.6 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 indicated through the communication object *Room temperature control alarm.*

6.3.7 Input from bus

6.3.7.1 Characteristics and timeout

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

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

The temperature controller allows the acquisition of the room temperature from an external temperature probe connected to the input of the device configured as analog (inputs 1B, 2B, 3B, 4B = enabled NTC).

In order to optimize or correct the room temperature regulation in particular cases (in large rooms, in the presence of strong asymmetry of the temperature distribution, when the thermostat is installed in an unsuitable position, etc.), the software can use the value acquired through the bus from another KNX device (e.g., from

an ekinex button), performing a weighted average between two temperature values. The weights are assigned by the *relative weight* parameter which assigns a proportion to the two values.

6.3.7.3 Surface temperature limitation function (external object)

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

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

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

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

The surface temperature limitation may be realized for several purposes:

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

Note that the warm water radiant panels are usually already equipped with a safety thermostat (with intervention on the hydraulic mixing group), while in the case of electrical power this device is not usable and it is common practice to realize a temperature limitation with a surface temperature sensor connected to the device.

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

6.3.7.4 Anticondensation protection function (external object)

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

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

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

6.3.7.5 Window contacts (external object)

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

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

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

6.3.7.6 Presence sensors (external objects)

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

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

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

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

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

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

Comfort extension. This function is only active if the actual operating mode is set on comfort; if, during this time, a presence is detected, the operating mode remains comfort even if the operating mode forced by the time scheduling function shifts to economy or standby. If a presence is not detected for a timeframe less than

a preset time, the operating mode does not change; vice versa, if a presence is not detected for a time period greater than the same preset time, the operating mode becomes the one forced by the time scheduling function.

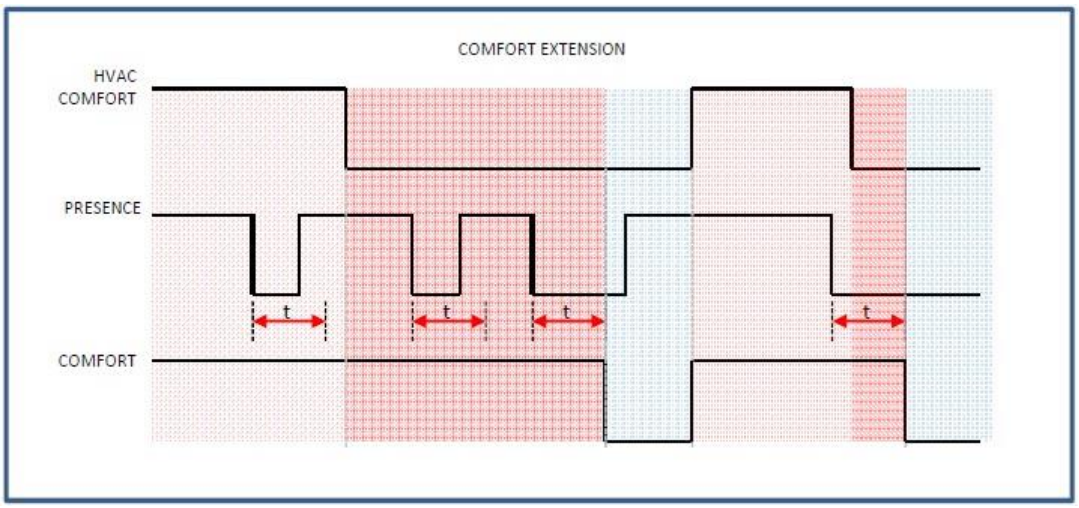


Figure 16 - Comfort extension

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

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

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

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

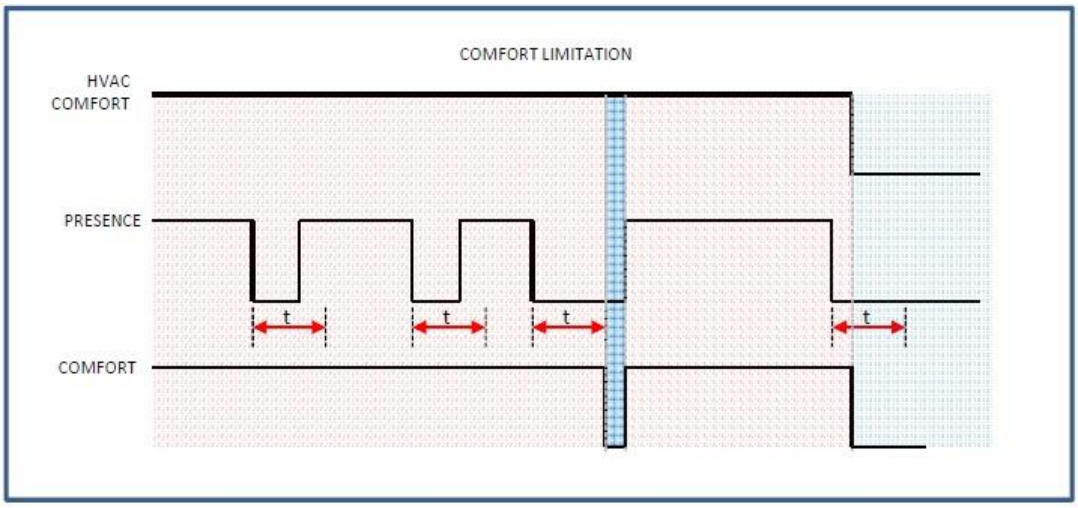


Figure 17 - Comfort limitation

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

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

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

7. 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 either by input or 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, 2B, etc.).

Default values for the parameters are highlighted in bold characters

The device parameters are divided into general and specific parameters, grouped into tabs. Figure 18 shows the tree structure of the application program with the main tabs.

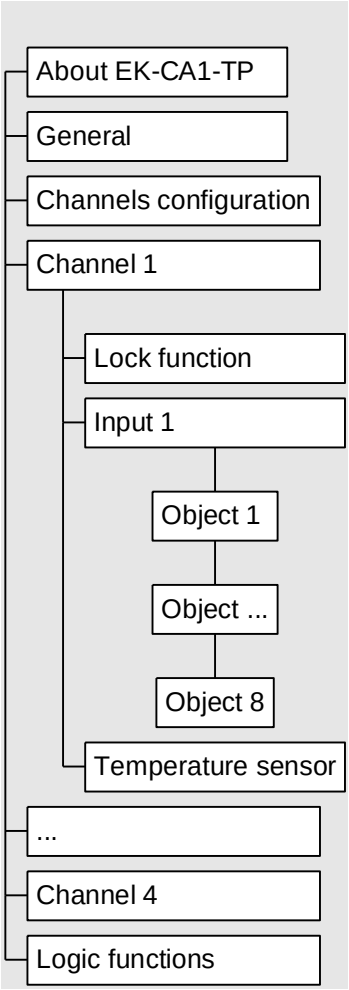


Figure 18 - ETS application main tabs

The folder ***About EK-CA1-TP*** is for information purposes only and does not contain parameters to be set.
The information given is:

Application software per ETS4/5
Version 1.00 (or higher)
Universal Interface 4 IN DI + 4 IN DI / NTC – EK-CA1-TP

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

7.1 General settings

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

Parameter name	*Conditions*	*Settings*
Manual operations	-	**enabled** / disabled
	This parameter enables or disables the membrane keypad of the device. *If it is set to "enabled", the manual operations mode is available and can be recalled pressing the corresponding pushbutton of the membrane keypad.* *If set to "disabled", the manual operations mode is unavailable.*	
Disable from bus	Manual operations = enabled	yes / **no**
	Enables or disables the capability of disabling manual operation of the inputs through a remote command (telegram).	
Restore auto mode time	Manual operations = enabled Disable from bus = no	hh:mm:ss (**00:15:00**)
	Allows to automatically switch off the manual operations mode after a time interval, in order to prevent the device to be unintentionally left offline. *The value 00:00:00 (zero) means that there is no automatic restore.*	
Debounce time	-	hh:mm:ss.fff (**00:00:00.050**)
	Sets a minimum time during which an input must remain stable in order to be considered valid, in order to avoid contact bounces or spikes .	
Delay after bus voltage recovery	-	hh:mm:ss.fff (**00:00:04.000**)

Parameter name	*Conditions*	*Settings*
	Delay before bus telegrams can be sent after a recovery of the bus voltage. The delay time affects the transmission generated by an event as well as the cyclical transmission. For the cyclical transmission: after the delay time finished, the cycle restarts and the first telegram will be sent after the cycle time.	
Technical alarm		**disabled** / enabled
	Enable or disable the generic malfunction alarm	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Disable front pushbuttons	Manual operations = enabled Disable from bus = yes	1 bit	C-W-	[1.001] switch	1

7.2 Channels configuration

These settings configure device channel behavior.

Parameter name	*Conditions*	*Settings*
Channel *x*	-	disabled **independent** coupled
	x = 1, 2, 3, 4. *Set operation mode for inputs corresponding to Channel x.*	
Input xx (xx = 1, 3, 5, 7)	**Channel *x* = independent or single**	disabled / **enabled DI**
	Enables or disables the first input of the x-channel (digital only). *The equivalence with the front panel and terminals is as follows:* • *Channel 1 – input 1 = 1A* • *Channel 2 – input 3 = 2A* • *Channel 3 – input 5 = 3A* • *Channel 4 – input 7 = 4A*	
Type	**Channel *x* = independent** **Input xx = enabled DI**	**send values or sequences** dimming shutter or venetian blind scene counter
	x = 1, 2, 3, 4. *xx = 1,…,8.* *Determines the kind of function performed by the specified input.* *Further parameters for the selected function will appear in the individual input configuration sections (see below).*	
Input xx (xx = 2, 4, 6, 8)	**Channel *x* = independent or single**	disabled **enabled DI** enabled NTC
	Enables or disables the second input of the x-channel (digital or analog NTC). *The equivalence with the front panel and terminals is as follows:* • *Channel 1 – input 2 = 1B* • *Channel 2 – input 4 = 2B* • *Channel 3 – input 6 = 3B* • *Channel 4 – input 8 = 4B*	

Parameter name	*Conditions*	*Settings*
Type	**Channel *x* = coupled**	switching **dimming** shutter or venetian blind
	x = 1, 2, 3, 4. *Determines the kind of function performed by the input pair.* *Further parameters for the selected function will appear in the individual rocker configuration sections (see below).*	

7.2.1 Independent or single: send values or sequences

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Switching status *[type]*, object *n*	Channel *x* = independent or single Input x*x* = enabled Type = send values or sequences	*According to configuration* (**1-bit**)	C−WTU	*According to configuration* (**[1.001] switch**)	4, 21 (1A, 1B) 38, 55 (2A, 2B) 72, 89 (3A, 3B) 106, 123 (4A, 4B)
	x = 1, 2, 3, 4. *xx = 1,…,8.* *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 input 3A (Channel 3 – input 1) have numbers from 72 to 79. The number of CO nr. 5 is therefore 72 + (5-1) = 76.* ***The size and type of the individual objects can be configured as described in following sections**.*				

7.2.2 Independent or single: dimming

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input x*x* – Switching command	Channel *x* = independent or single Input *xx* = enabled DI Type = dimming	1 bit	C−WTU	[1.001] switch	12, 29 (1A, 1B) 46, 63 (2A, 2B) 80, 97 (3A, 3B) 114, 131 (4A, 4B)
	x = 1, 2, 3, 4. *xx = 1,…,8.* *Send a command to a dimming actuator to switch the light on or off.* *The command is triggered by a short press on the input.* *The value sent can be a fixed value or it can be toggled at each input activation.*				
Input x*x* – Dimming up / down / stop command	Channel *x* = independent Input *xx* = enabled Type = dimming	4 bit	C−−T−	[3.*] 3-bit control	13, 30 (1A, 1B) 47, 64 (2A, 2B) 81, 98 (3A, 3B) 115, 132 (4A, 4B)

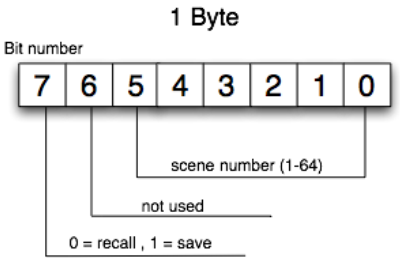
Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
	x = 1, 2, 3, 4. *xx = 1,…,8.* *Send a command to a dimming actuator to change dimming intensity (brighter or darker).* *Three values are used which mean start increase, start decrease or stop the change.*  *Increase/decrease values are sent when a long press action occurs and stop value on press release.* *The value sent can be a fixed value or it can be toggled at each input activation.*				

7.2.3 Independent: shutter or venetian blind

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Stop – step up/down command	Channel *x* = independent or single Input xx = enabled DI Type = shutter or venetian blind	1 bit	C--T-	[1.007] step	15, 32 (1A, 1B) 49, 66 (2A, 2B) 83,100 (3A, 3B) 117,134 (4A, 4B)
	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.				
Input *xx* – Move up / down command	Channel *x* = independent or single Input xx = enabled DI Type = shutter or venetian blind **Venetian blind mode = enabled**	1 bit	C--T-	[1.008] up/down	16,33 (1A, 1B) 50, 67 (2A, 2B) 84, 101 (3A, 3B) 118, 135 (4A, 4B)
	Move the blind to fully open or fully closed position. The object is sent at the end of a long press.				
Input *xx* – Dedicated stop command	Channel *x* = independent Input xx = enabled Type = shutter or venetian blind **Venetian blind mode = disabled**	1 bit	C--T-	[1.017] trigger	12, 29 (1A, 1B) 46, 63 (2A, 2B) 80, 97 (3A, 3B) 114, 131 (4A, 4B)
	Increase or decrease the opening of the blind stepwise. *The object is sent on a short press.*				

7.2.4 Independent: scene

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Scene number	Channel *x* = independent or single Input xx = enabled DI Type = scene	1 Byte	C--T-	[17.*] Scene number [18.*] Scene control	17, 34 (1A, 1B) 51, 68 (2A, 2B) 85, 102 (3A, 3B) 119, 136 (4A, 4B)

Object name	Conditions	Size	Flags	DPT	CO number(s)
	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). 				

7.2.5 Independent: counter

Object name	Conditions	Size	Flags	DPT	CO number(s)
Input x*x* – Counter value [1/2/4] byte	Channel *x* = independent or single Input xx = enabled DI Type = counter	*According to configuration* (**1-bit**)	C-R-T-	[12.001] Counter pulses [13.001] Counter pulses	17, 34 (1A, 1B) 51, 68 (2A, 2B) 85, 102 (3A, 3B) 119, 136 (4A, 4B)
	This object stores the current value of the input counter. *For each input, it is possible to select the counter size:* • *From 0 to 255 (1 byte)* • *From 0 to 65535 (2 bytes)* • From 0 to 4294967295 (4 bytes)				
Input x*x* – Counter reset command	Channel *x* = independent or single Input xx = enabled DI Type = counter	1-bit	C-W--	[1.015] reset	18, 35 (1A, 1B) 52, 69 (2A, 2B) 86, 103 (3A, 3B) 120, 137 (4A, 4B)
	It resets the counter by setting its value to zero				
Input x*x* – Counter runout	Channel *x* = independent or single Input xx = enabled DI Type = counter	1-bit	CRWT-	[1.055] alarm	19, 36 (1A, 1B) 53, 70 (2A, 2B) 87, 104 (3A, 3B) 121, 138 (4A, 4B)
	It sends an alarm bit when the counter reaches the maximum value according to the data size defined for the counter.				

7.2.6 Coupled: switch

Object name	Conditions	Size	Flags	DPT	CO number(s)
Inputs *x* and y – Switching command	Channel *x* = coupled Type = switching	1-bit	C-WTU	[1.001] switch	12, 46, 80, 114
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8. *Please see the notes of the same function for the independent input.*				

7.2.7 Coupled: dimming

Object name	Conditions	Size	Flags	DPT	CO number(s)
Inputs *x* and y – Switching command	Channel *x* = coupled **Type = dimming**	1 bit	C-WTU	[1.001] switch	12, 46, 80, 114
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8. *Please see the notes of the same function for the independent input.*				
Inputs *x* and y - Dimming up / down / stop command	Channel *x* = independent **Type = dimming**	4 bit	CR-T-	[3.*] 3-bit control	13, 47, 81, 115
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8. *Please see the notes of the same function for the independent input.*				

7.2.8 Coupled: shutter or venetian blind

Object name	Conditions	Size	Flags	DPT	CO number(s)
Inputs *x* and y - Dedicated stop command	Channel *x* = coupled **Type = shutter or venetian blind** **Blind mode = disabled**	1 bit	C--T-	[1.017] trigger	12, 46, 80, 114
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8. *Please see the notes of the same function for the independent input.*				
Inputs *x* and y - Stop – step up/down command	Channel *x* = coupled **Type = shutter or venetian blind** **Blind mode = enabled**	1 bit	CR-T-	[1.007] step	15, 49, 83, 117
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8. *Please see the notes of the same function for the independent input.*				
Inputs *x* and y - Move up / down command	Channel *x* = coupled **Type = shutter or venetian blind**	1 bit	C--T-	[1.008] up/down	16, 50, 84, 118
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8. *Please see the notes of the same function for the independent input.*				

7.2.9 Channel x: Input xA / xB configuration

7.2.9.1 Independent channels

For the *independent or* single channel setting, all parameters listed below are referred to either input A or input B (whichever are enabled).

In the following sections, it is implicitly understood that for the listed parameters to appear, the corresponding inputs xA and/or xB must be enabled.

The entries assigned to "Object *n*" are repeated so many times as the number of configured objects according to the *Number of Communication Objects* parameter.

For all Type values:

Parameter name	*Conditions*	*Settings*
Contact type	-	**NO (normally open)** NC (normally closed)
	In normally open (NO) mode the "active" state of the input is when input contacts shorted, and the "inactive" state is when they are disconnected. In normally closed (NC) mode the device behaviour is the opposite.	
Lock function	-	enabled / **disabled**
	Enables or disables the capability of locking the input through a remote command (telegram).	
Lock function – Invert lock device	Channel *x* = independent or single Lock function = enabled	**not inverted** / inverted
	It allows managing a "lock activate" telegram as unlock and vice-versa.	
Lock function – Lock after bus recovery	Channel *x* = independent or single Lock function = enabled	**no** / yes
	If active, after returning from a bus failure or power-off the device will retain the lock status it had before. Otherwise (in the default case), the device will restart in the non-locked condition.	

7.2.9.2 Independent: Lock function enabled

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Lock command	Channel *x* = independent or single Lock function = enabled	1 bit	C–W–U	[1.003] enable	3, 20 (1A, 1B) 37, 54 (2A, 2B) 71, 88 (3A, 3B) 105, 122 (4A, 4B)
	x = 1, 2, 3, 4. *xx = 1, …, 8.*				

When the lock function is enabled, for each input or channel a behaviour can be defined to be followed when the locking or unlocking command is received.

The details will be listed in the following sections; the different behaviours are summarized in the table below.

Channel mode	Input type	Behaviour at lock	Behaviour at unlock
Independent or single	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
	Shutter or venetian blind	**none** up down	
	Scene	**none** send first scene send second scene	
	Counter	NA	
Coupled	Switching	**none** off on toggle	**none** off on as previous
	Dimming	**none** off on toggle	**none** off on as previous
	Shutter or venetian blind	**none** up down	

7.2.9.3 Independent: send values or sequences

Parameter name	*Conditions*	*Settings*
Number of Communication Objects	Channel *x* = independent or single Type = send values or sequences	1…8 (**1**)
	Number of configuration objects configured in association with the input event.	
Lock function – Behaviour at locking	Channel *x* = independent or single Type = send values or sequences Lock function = enabled	**none** as close or short press as open or long press
	Allows to perform the operation associated to the specified event when a locking command is received.	
Lock function – Behaviour at unlocking	Channel *x* = independent or single Type = send values or sequences Lock function = enabled	**none** as close or short press as open or long press
	Allows to perform the operation associated to the specified event when an unlocking command is received.	
Event	Channel *x* = independent or single Type = send values or sequences	**close / open contact** short / long press
	Type of event that should be used as trigger for an action	
Long press time	Channel *x* = independent or single Type = send values or sequences Event = short / long press	hh:mm:ss.fff (**00:00:00.800**)
	Minimum push time for a press in order to be recognized as a long press.	
Object *n* – send delay	Channel *x* = independent or single Type = send values or sequences	hh:mm:ss.ff (**00:00:00.00**)
	Delay before the object is transmitted on the bus. *By defining a delay after the event occurs and before the object value is sent, it is possible to associate a time defined sequence of values to an input event.*	
Object *n* – Send cyclically	Channel *x* = independent or single Type = send values or sequences **Number of Comm. 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. *Cyclical transmission is only available if the number of Communication Objects is set to "1".*	
Object *n* – Cyclic sending interval	Channel *x* = independent or single Type = send values or sequences **Number of Comm. Objects = 1** **Send cyclically ≠ none**	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions. *Cyclical sending interval is only available if the number of Communication Objects is set to "1".*	
Object *n* – Communication Object dimension	Channel *x* = independent or single 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 point value
	Defines size and type of the values to be sent when an event occurs.	

Parameter name	*Conditions*	*Settings*
Object *n* – Reaction to Short Press	Channel *x* = independent or single Type = send values or sequences **send dimension = 1 bit value**	none on off **toggle**
	Channel *x* = independent or single Type = send values or sequences **send dimension = 2 bit value**	none disable enable off / up **enable on / down** enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Channel *x* = independent or single Type = send values or sequences **send dimension = *any byte value***	none **send value 1** send value 2 send value 1 ↔ send value 2
	Value change behaviour caused by either a Close or a Short Press event (according to event configuration)	
Object *n* – Reaction to Long Press	Channel *x* = independent or single Type = send values or sequences **send dimension = 1 bit value**	**none** on off toggle
	Channel *x* = independent or single Type = send values or sequences **send dimension = 2 bit value**	none **disable** enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Channel *x* = independent or single Type = send values or sequences **send dimension = *any byte value***	**none** send value 1 send value 2 send value 1 ↔ send value 2
	Value change behaviour caused by either an Open or a Long Press event (according to event configuration)	
Object *n* – Value 1	Channel *x* = independent or single Type = send values or sequences **send dimension = *any byte value***	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 bytes unsigned value) -32768... 32767 (2 bytes signed value) -671088.64...670760.96 (2 bytes floating value)
	First value available for association in send events	
Object *n* - Value 2	Channel *x* = independent or single Type = send values or sequences **send dimension = *any byte value***	*same as value 1*
	Second value available for association in send events	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Switching status *[type]* Object *n*	Channel *x* = independent or single Type = send values or sequences	*See table below*	C-WTU	*See table below*	4, 21 (1A, 1B) 38, 55 (2A, 2B) 72, 89 (3A, 3B) 106, 123 (4A, 4B)

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
	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 input 3A (Channel 3 – input 5) have numbers starting from 72. The number of CO nr. 5 is therefore 72 + (5-1) = 76.*				

Sizes and DPTs are as follows:

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

7.2.9.4 Independent: dimming

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

7.2.9.5 Independent: shutter or venetian blind

Parameter name	*Conditions*	*Settings*
Long press time	Channel *x* = independent or single 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	Channel *x* = independent or single Type = shutter or venetian blind	**enabled** / disabled
	When enabled, causes each subsequent press to invert the direction of movement; otherwise, a fixed direction can be assigned.	
Up / Down action	Channel *x* = independent or single Type = shutter or venetian blind **Toggle mode = disabled**	**up** down
	Defines the movement direction to be assigned to the button press.	
Venetian blind mode	Channel *x* = independent or single Type = shutter or venetian blind	enabled / **disabled**
	If blinds mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.	
Lock function – Behaviour at locking	Channel *x* = independent or single Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when a locking command is received.	
Lock function – Behaviour at unlocking	Channel *x* = independent or single Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when an unlocking command is received.	

7.2.9.6 Independent: scene

Parameter name	*Conditions*	*Settings*
First scene number	Channel *x* = independent or single 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	Channel *x* = independent or single 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	Channel *x* = independent or single 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	Channel *x* = independent or single 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	Channel *x* = independent or single Type = scene **Learning mode = disabled** **Scene activation = toggle between two scenes**	1…64 (**2**)
	Alternate scene number to be assigned to button press.	
Lock function – Behaviour at locking	Channel *x* = independent or single Type = scene	**none** send first scene send second scene
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Channel *x* = independent or single Type = scene	**none** send first scene send second scene
	Operation to perform when an unlocking command is received.	

7.2.9.7 Independent: counter

Parameter name	*Conditions*	*Settings*
Cyclical transmission interval	Channel *x* = independent or single Type = counter	hh:mm:ss (**00:02:00**) (values in the [00:00:00 – 18:12:15] range)
	Interval between cyclical transmissions. A zero value (00:00:00) means no cyclical transmission. ***Note:** values less than 10 seconds are considered equal to 10.*	
Counter dimension	Channel *x* = independent or single Type = counter	**from 0 to 255 (1 byte)** from 0 to 65535 (2 bytes) from 0 to 4294967295 (4 bytes)
	Value type of the counter. Unsigned integer value of 1, 2 or 4 bytes.	
Max value	Channel *x* = independent or single Type = counter	Depending on the counter dimension: 0…**255** 0…**65535** 0…**4294967295** (default value is the maximum value of the selected interval)
	Limit value for the counter. When this value is reached,a "runout" telegram is sent and the counter value is reset to zero.	

7.2.9.8 Coupled inputs

For a pair of coupled inputs, all parameters refer to the only menu item present relating to inputs xA and xB.

In the following sections, it is implicitly understood that for the listed parameters to appear, the corresponding inputs xA and xB must be in the enabled state.

For all Type values:

Parameter name	*Conditions*	*Settings*
Lock function	Channel *x* = coupled	enabled / **disabled**
	Enables or disables the capability of locking the input through a remote command (telegram).	

7.2.9.9 Coupled: Lock function enabled

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input x and y - Lock command	Channel *x* = coupled **Lock function = enabled**	1 bit	C-W-U	[1.003] enable	3, 37, 71, 105
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8.				

7.2.9.10 Coupled: switching

Parameter name	*Conditions*	*Settings*
x and y use	Channel *x* = coupled Type = switching	**A on, B off** A off, B on
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8.	
Send cyclically	Channel *x* = coupled Type = switching	**none** off / value 1 on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclic sending interval	Channel *x* = coupled Type = switching **Send cyclically ≠ none**	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at locking	Channel *x* = coupled Type = switching	**none** off on toggle
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Channel *x* = coupled Type = switching	**none** off on previous
	Operation to perform when an unlocking command is received.	

7.2.9.11 Coupled: dimming

Parameter name	*Conditions*	*Settings*
Long press time	Channel *x* = coupled Type = dimming	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
x and y use	Channel *x* = coupled Type = dimming	**A on, B off** A off, B on
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8.	
Send cyclically	Channel *x* = coupled Type = dimming	**none** off / value 1 on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclic sending interval	Channel *x* = coupled Type = dimming **Send cyclically ≠ none**	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at locking	Channel *x* = coupled Type = dimming	**none** off on toggle
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Channel *x* = coupled Type = dimming	**none** off on toggle
	Operation to perform when an unlocking command is received.	

7.2.9.12 Coupled: shutter or venetian blind

Parameter name	*Conditions*	*Settings*
Long press time	Channel *x* = coupled 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.	
x and y use	Channel *x* = coupled Type = shutter or venetian blind	**A up, B down** A down, B up
	x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8.	
Venetian blind mode	Channel *x* = coupled Type = shutter or venetian blind	enabled / **disabled**
	If blinds mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.	
Lock function – Behaviour at locking	Channel *x* = coupled Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when a locking command is received.	
Lock function – Behaviour at unlocking	Channel *x* = coupled Type = shutter or venetian blind	**none** up down
	Allows to perform the specified operation when an unlocking command is received.	

For other communication objects related to *coupled* mode, please refer to the general *Channels Configuration* section.

7.2.10 Temperature sensor (inputs 2, 4, 6, 8 only)

For the B inputs of each channel (indicated with input 2, 4, 6, 8), it is possible to select an NTC probe.

This choice enables a Temperature Sensor tab in the corresponding channel menu.

7.2.10.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Temperature sensor	Input 2 = enabled NTC, or Input 4 = enabled NTC, or Input 6 = enabled NTC, or Input 8 = enabled NTC	-
Filter type	Temperature sensor = enabled	low **medium** high
	Low = average value every 4 measurements *Medium = average value every 16 measurements* *High = average value every 64 measurements*	
Temperature offset	Temperature sensor = enabled	**0°C** [range -5°C … +5°C]
	Corrective value applied to the measured temperature.	
Minimum change of value to send [K]	Temperature sensor = enabled	**0,5** [range 0 … 5 °C]
	If the parameter is set to 0 (zero), no value is sent after a change.	
Cyclic sending interval	Temperature sensor = enabled	**No sending** [other values in the 30 s … 120 min range]
Threshold 1	Temperature sensor = enabled	**not active** below above
Value [°C]	Temperature sensor = enabled, Threshold 1 = below or above	**7** [range 0 … 50 °C]
Threshold 2	Temperature sensor = enabled	**not active** below above
Value [°C]	Temperature sensor = enabled, Threshold 2 = below or above	**45** [range 0 … 50 °C]

Parameter name	*Conditions*	*Settings*
Hysteresis	Temperature sensor = enabled, Threshold 1 and/or Threshold 2 = below or above	**0,4 K** [other values in the 0,2…3 K range]
Cyclic sending interval	Temperature sensor = enabled, Threshold 1 and/or Threshold 2 = below or above	**No sending** [other values in the 30 s … 120 min range]
Thermostat functionalities	Temperature sensor = enabled	**Disabled /** enabled
	Enables or disabled the thermostat functionality.	

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm.*
Input xx - Temperature value	Temperature sensor = enabled	2 Bytes	CR-T--	[9.001] temperature (°C)	159, 227, 295, 363
	xx = 2, 4, 6, 8. *Temperature value read by the NTC sensor.*				
Temperature threshold1 - Switch	Temperature sensor = enabled, Threshold 1 = below or above	1 Bit	CR-T--	[1.001] switch	160, 228, 296, 364
	xx = 2, 4, 6, 8.				
Temperature threshold 2- Switch	Temperature sensor = enabled, Threshold 2 = below or above	1 Bit	CR-T--	[1.001] switch	161, 229, 297, 365
	xx = 2, 4, 6, 8.				

7.2.11 Acquisition filter

The acquisition filter calculates an average with a series of measured values before sending on the bus. The parameter can have the following values:

- low = average value every 4 measurements;
- medium = average value every 16 measurements;
- high = average value every 64 measurements.

7.2.12 Correction of the measured temperature

The sampling of the temperature value occurs every 10 seconds, while the display is updated every minute. During the configuration with ETS the opportunity is given to correct the measured temperature value within the offset range of - 5 °C ... + 5 °C (step: 0.1 K).

7.3 Thermostat function

The Thermostat function is enabled within the Temperature Sensor board, for each input with an enabled NTC probe.

This function allows to configure the device as a room temperature regulator and also allows you to filter, with a weighted average, the reading of the NTC input probe with the reading of an added sensor acquired from the bus.

The following secondary tabs are enable:

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

7.3.1 External sensors (from bus)

As “external sensors” are intended KNX-devices (or conventional sensors interfaced to the bus through KNX devices) which send states or values to the room temperature controller via the bus.

The internal control system of the device cyclically monitors the update status of the external sensor values (from the bus) when the value set for the timeout expires. If no updated value has been received, the regulation function is suspended and the actuation valves are closed. An alarm is issued on the bus via communication objects *211, 279, 347, 415 - Room temperature control alarm* (refer to the Settings Tab).

The folder ***External sensors (from bus)*** includes 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

7.3.1.1 Parameter and communication object tables

Parameter name	***Conditions***	***Values***
Room temperature		**disabled** / enabled

	It enables a bus temperature sensor. The measured value can be used to calculate a weighted average value in combination with the temperature sensor integrated into the device or a temperature sensor connected to a device input.	
Cyclic reading interval	Room temperature = enabled	**no reading** [other values in the range 30 s … 120 min]
	If the parameter is set to "no reading", the corresponding communication object must be updated by the remote device sending data. With any different value, data are updated with a reading request by the room thermostat.	
Relative humidity		**disabled** / enabled
Humidity CO dimension	Relative humidity = enabled	1 byte (DPT 5.001) 2 **byte (DPT 9.007)**
Cyclic reading interval	Relative humidity = enabled	**no reading** [other values in the range 30 s … 120 min]
Antistratification temperature		**disabled** / enabled
	It enables a temperature bus sensor to carry out the antistratification function.	
Cyclic reading interval	Antistratification temperature = enabled	**no reading** [other values in the range 30 s … 120 min]

Outdoor temperature		**disabled** / enabled
	It enables an outdoor temperature bus sensor to display the measured value on the display. This is alternative to an outdoor temperature sensor connected to a device input: the parameter appears only if the external temperature sensor is disabled in the Inputs folder.	
Cyclic reading interval	Light sensor = 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]

Window contact 1		**disabled** / enabled
	It enables a bus sensor for detecting the state of opening / closing of a window or a door.	
Signal	Window contact 1= enabled	**not inverted** / inverted
Cyclic reading interval	Window contact 1= enabled	**no reading** [other values in the range 30 s … 120 min]
Window contact 2		**disabled** / enabled
	It enables a bus sensor for detecting the state of opening / closing of a window or a door.	
Signal	Window contact 2= enabled	**not inverted** / inverted
Cyclic reading interval	Window 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, an alarm is displayed on the display through the symbol and the corresponding alarm code.

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Room temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	162, 230, 298, 366
Humidity (2 bytes, from bus)	Relative humidity sensor = enabled, Humidity comm. obj. size = 2 byte	2 Byte	C-WTU	[9.007] humidity (%)	163, 231, 299, 367
Humidity (1 byte, from bus)	Relative humidity sensor = enabled, Humidity comm. obj. size = 1 byte	1 Byte	C-WTU	[5.001] percentage (0..100%)	164, 232, 300, 368
Antistratification temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	165, 233, 301, 369
Outdoor temperature (from bus)	Outdoor temperature = enabled	2 Byte	C-WTU	[9.001] temperature °C	166, 234, 302, 370
Coil temperature (from bus)	Coil temperature = enabled	2 Byte	C-WTU	[9.001] temperature (°C)	167, 235, 303, 371
Floor temperature (from bus)	Floor temperature = enabled	2 Byte	C-WTU	[9.001] temperature (°C)	168, 236, 304, 372
Flow temperature (from bus)	Flow temperature = enabled	2 Byte	C-WTU	[9.001] temperature (°C)	169, 237, 305, 373
Anticondensation (from bus)	Anticondensation = enabled	1 Bit	C-WTU	[1.001] switch	175, 243, 311, 379
Windows contact sensor 1 (from bus)	Windows contact 1 = enabled	1 Bit	C-WTU	[1.019] window/door	170, 238, 306, 374

Windows contact sensor 2 (from bus)	Windows contact 2 = enabled	1 Bit	C-WTU	[1.019] window/door	191, 259, 327, 395
Presence sensor 1 (from bus)	Presence sensor 1 = enabled	1 Bit	C-WTU	[1.018] occupancy	192, 260, 328, 396
Presence sensor 2 (from bus)	Presence sensor 2 = enabled	1 Bit	C-WTU	[1.018] occupancy	193, 261, 329, 397
Contact of card holder (from bus)	Card holder contact = enabled	1 Bit	C-WTU	[1.001] switch	194, 262, 330, 398

7.3.2 Weighted temperature value

The ***Weighted temperature value*** folder appears only if two sensors for measuring the room temperature are enabled and includes the following parameters:

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

7.3.2.1 Parameter and communication object

Parameter name	***Conditions***	***Values***
Relative weight		100% main sensor 90% / 10% 80% / 20% 70% / 30% 60% / 40% **50% / 50%** 0% / 60% 30% / 70% 20% / 80% 10% / 90% 100% sensor from bus

Minimum change of value to send [K]	Relative weight ≠ 100% main sensor Relative weight ≠ 100% additional sensor	**0,5** [other values in the range 0 … 5 K]
	If the parameter is set to 0 (zero), no value is sent at the change.	
Cyclic sending interval	Relative weight ≠ 100% main sensor Relative weight ≠ 100% additional sensor	**no sending** [other values in the range 30 s … 120 min]

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Weighted temperature	Cyclic sending interval ≠ no sending	2 Byte	CR-T--	[9.001] temperature °C	176, 244, 312, 380

7.3.2.2 About weighted temperature

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

1) from the temperature sensor integrated in the device;
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 the communication object “Weighing temperature” is set to 7F FF in case the value cannot be read from the bus.

7.3.3 Temperature control

The ***Temperature control*** folder includes the following secondary folders:

- Settings
- Heating
- Cooling
- Ventilation

The ***Cooling*** and ***Ventilation*** secondary folders appear only if in the ***Settings*** folder the parameter Thermostat function is set to the value *both heating and cooling* or *cooling*.

The ***Cooling*** secondary folder is displayed only if in the ***Settings*** folder the Thermostat function parameter is set to "*both heating and cooling*" or "*cooling*".

The Ventilation secondary folder is displayed in the following cases:

- if in the **Heating** or **Cooling** tab the type is set to *fancoil*, or
- if in the **Heating** or **Cooling** tab, the *auxiliary heating* or *auxiliary cooling* is enabled and *ventilation for auxiliary heating* or *cooling* is enabled.

7.3.3.1 Settings

The ***Settings*** folder includes the following parameters:

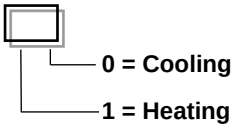
- Setpoint type
- Thermostat function
- Command Communication Object, unique or separated command (2 or 4 pipes system)
- Heating – cooling changeover
- Heating – cooling cyclic sending interval
- Thermostat function after download
- Setpoint cyclic sending interval
- End of manual operation
- Disable temperature controller from bus
- Signal from bus
- Transmission delay after mode change
- Valve protection function
- Frequency
- Time interval

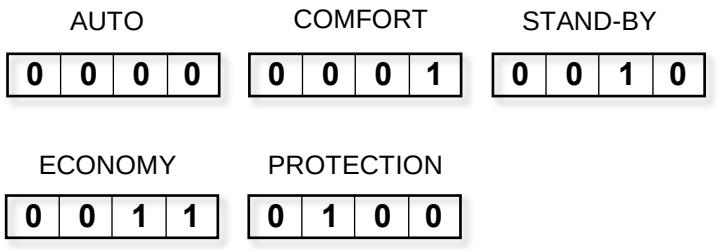
7.3.3.2 Parameter and communication object tables

Parameter name	***Conditions***	***Values***
Setpoint type	Thermostat functionalities = enabled	**single** absolute relative
	In case the option "Single" setpoint is selected and Thermostat function = heating, the temperature controller acts on heating mode; in case Thermostat function = cooling, the temperature controller acts on cooling mode. *In case Thermostat function = both heating and cooling, the current seasonal modeneeds to be specified by the proper communication object.*	
Thermostat function	Thermostat functionalities = enabled	**heating** cooling both heating and cooling
Command Communication Object	Thermostat function = both heating and cooling	**separated** / unique

Heating–cooling changeover	Setpoint type ≠ single Thermostat function = both heating and cooling Command Communication Object = separated	**from bus** automatic
	In case Setpoint type = 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 the 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 the range 30 s … 120 min]
	The setpoint value that can be sent cyclically is the actual one, depending on the operating mode set manually by the user or automatically by another KNX supervising device with the possibility of time scheduling. The actual setpoint value takes also into account the actual state of the contacts window and presence detection (if the corresponding functions are enabled).	
End of manual operation	Setpoint type = Absolute or Relative	**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
	It allows to disable the temperature controller functions from bus.	
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 the range 5 s … 20 min]

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Actual setpoint		2 Byte	CR-T--	[9.001] temperature (°C)	184, 252, 320, 388
Manual setpoint	Setpoint type = absolute or relative	2 Byte	C-W--	[9.001] temperature (°C)	185, 253, 321, 389
Heating / cooling status out		1 Bit	CR-T--	[1.100] heating/cooling	177, 245, 313, 381
	The communication object is sent over the bus after an internally elaborated switching event. The object is always exposed and contains information on the current conduction mode of the internal temperature controller. **[1.100] DPT Heat/Cool 1 Bit**  **0 = Cooling** **1 = Heating**				
Heating / cooling status in	Thermostat function = both heating and cooling, Heating–cooling changeover = from bus	1 Bit	C-W--	[1.100] heating/cooling	178, 246, 314, 382
	The communication object is received from the bus. At the switching event the internal regulator switches the conduction mode.				
HVAC mode in	Setpoint type = absolute or relative	1 Byte	C-W--	[20.102] HVAC mode	179, 247, 315, 383

	Bits 5, 8 are reserved. **[20.102] DPT HVAC Mode 1 Byte**  *The device receives the operating mode (HVAC mode) from a bus device with supervisor function. The operating mode received via this communication object can be subsequently changed by the user (in this case the room thermostat switches to manual control).*				
HVAC forced mode in	Setpoint type = absolute or relative	1 Byte	C-W--	[20.102] HVAC mode	180, 248, 316, 384
	The communication object receives the operating mode in the same way as it happens with the communication object HVAC mode in; the difference is that the operating mode received via this object (except for the AUTO command) cannot be subsequently changed by the user. The user can change the mode only after the forced HVAC mode in has sent the AUTO command.				
HVAC mode out	Setpoint type = absolute or relative	1 Byte	CR-T-	[20.102] HVAC mode	181, 249, 317, 385
HVAC manual mode	Setpoint type = absolute or relative	1 Byte	C-WTU	[20.102] HVAC mode	182, 250, 318, 386
Chrono active status	Setpoint type = absolute or relative	1 Bit	CR-T-	[1.011] state	183, 251, 319, 387
	It sends information about the current chrono program status (active or inactive)				
Room temperature controller status		1 Bit	CR-T-	[1.003] enable	194, 262, 330, 398
Thermostat - Alarm text		14 bytes	CR-T-	[16.000] Character string (ASCII)	206, 274, 342, 410

Manual setpoint active status	Setpoint type = absolute or relative	1 Bit	CRWTU-	[1.011] state	207, 275, 343, 411
Input setpoint	Setpoint type = single	2 bytes	CRWTU	[9.001] temperature (°C)	186, 254, 322, 390
	The communication object allows to set and / or read the status of the setpoint (manual / forced).				
Room temperature controller alarm		1 bit	CR-T--	[1.005] alarm	211, 279, 347, 415
Disable room temperature controller	Disable temperature controller from bus = yes	1 bit	C-W--	[1.001] switch	212, 280, 348, 416
Thermal generator lock		1 bit	C-W--	[1.005] alarm	223, 291, 359, 427
Building protection HVAC mode active		1 bit	CR-T--	[1.011] state	224, 292, 360, 428

About heating/cooling terminals

The application functions of the room temperature controller configurable with ETS are particularly suitable for the control through general-purpose or dedicated KNX actuators of the following heating/cooling terminals:

- radiators;
- electric heaters;
- fancoils;
- radiant panels;
- dehumidification units;
- radiant panels + radiators (as auxiliary system);
- radiant panels + fancoils (as auxiliary system);
- radiant panels + dehumidification units.

7.3.3.3 Seasonal mode switching (Heating / cooling)

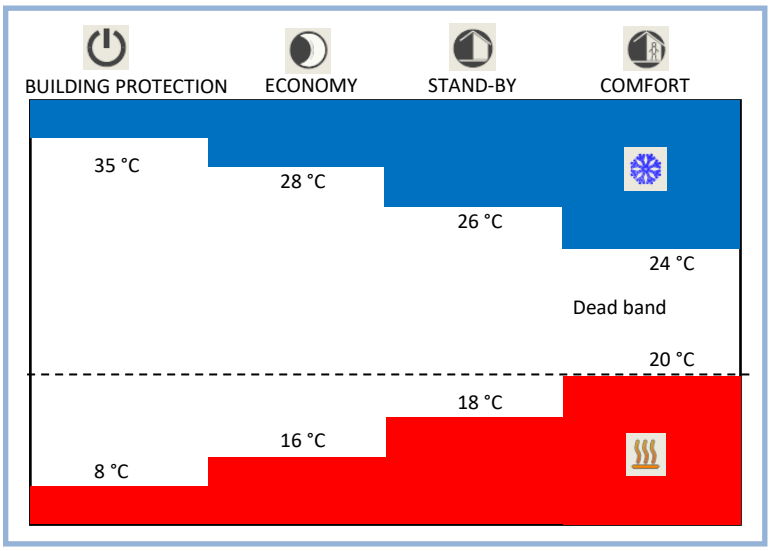
The switch between heating and cooling mode can take place in 2 ways:

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

Automatic switch over, based on the room temperature (mode 1)

This mode is only suitable in applications for heating / cooling hydraulic systems with a 4-pipe configuration. Even in this case, the information can be sent on the bus with the output communication object [DPT 1,100 heat / cool]; the difference with mode 1 is that the switch over is performed automatically by the device, basing on the values of current temperature and setpoint. The automatic switch over is achieved by introducing a dead band, as shown in Figure 19

..



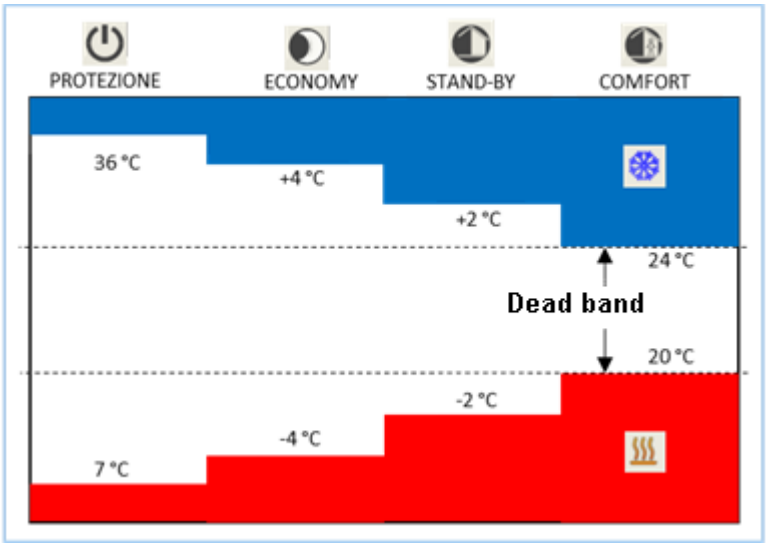


Figure 19 - Automatic switch over, based on the room temperature

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

Switch over from KNX bus (mode 2)

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

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

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

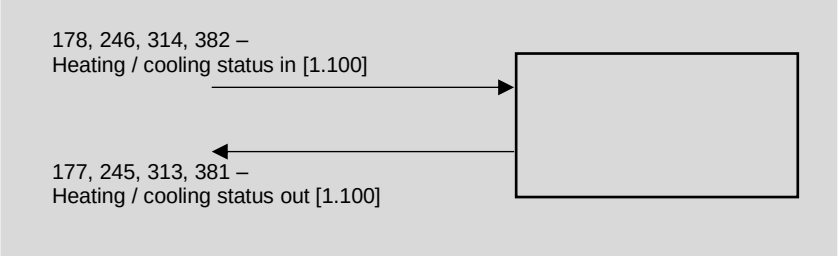


Figure 20 - Switch over from KNX bus

7.3.3.4 Valve protection function

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

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

7.3.3.5 Remote Setpoint modification

These communication objects allow to perform the same modifications remotely, for example from a supervisor software.

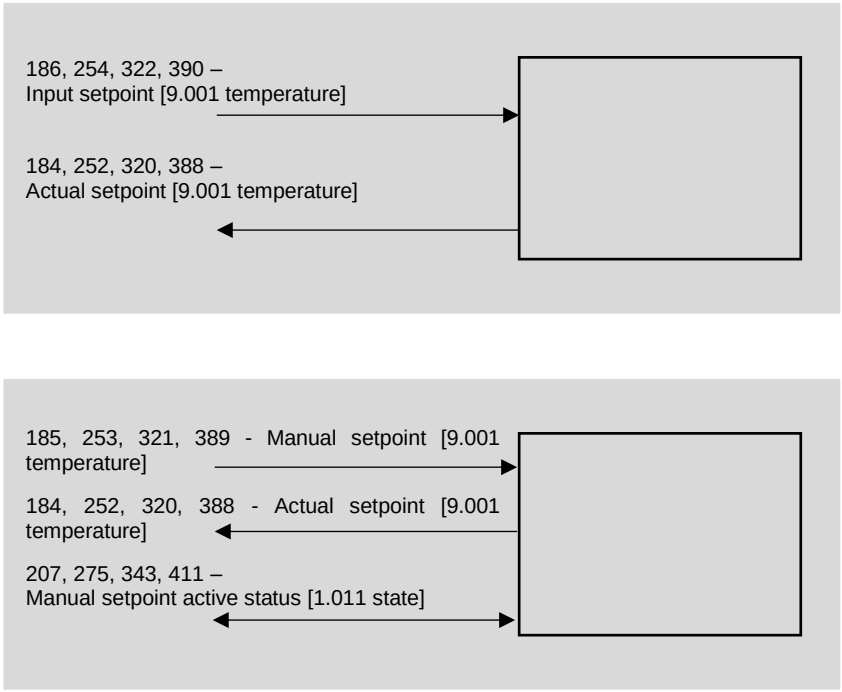


Figure 21 – Remote setpoint setting

Those objects are about the Setpoint forced modification: alternatively, the supervisor can act directly on the operating mode setpoints (C.O from *186* to *193,* from *254* to *261*, from *322* to *329* and from *390* to *397*)). The values of the C.O. *184, 252, 320, 388 - Actual setpoint* represent the current operative setpoints which the control algorithms are based on. The C.O. *207, 275, 343, 411- Manual setpoint active status* indicates (read request mode) if the forced mode is active (symbol M on the LCD display present). The supervisor can force at any time the actual setpoint by writing a new value directly into the C.O. *185, 253, 321, 389 - Manual setpoint*. The C.O. *207, 275, 343, 411 - Manual/forced setpoint active status* can also be used in writing mode, to exit the active forced mode.

7.3.3.6 Remote operative mode modification

The communication objects shown in Figure 22 allow to monitor the operating mode (comfort, standy, economy and building protection) modifications performed locally by the user when interacting with the LCD display and the touch buttons of the room thermostat, or the operating mode forced by chrono program.

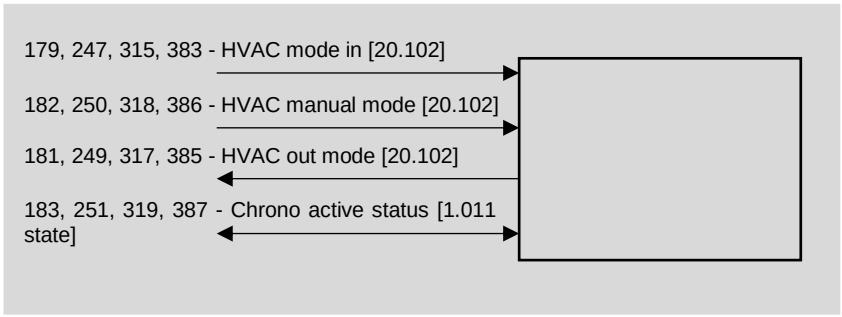


Figure 22 - Remote operative mode modification

The C.O. *179, 247, 315, 383 - HVAC mode in* are associated to the chrono program. The C.O.s *181, 249, 317, 385 - HVAC mode out* and *183, 251, 319, 387 - HVAC chrono active status* allow the remote supervisor to detect the operating mode currently active on the room thermostat. Moreover, it also allows to understand if the chrono program is active or if attenuation is handled manually or not. The supervisor can set at any time a manual operating mode through C.O. *182, 250, 318, 386 - HVAC manual mode*; to start the chrono program remotely, the C.O. *HVAC manual mode* is to be set on value 0 = Automatic.

7.3.4 Heating

The ***Heating*** folder allows to set:

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

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

This folder contains the following parameters:

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

7.3.4.1 Parameter and communication object tables

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

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

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 a correct operation of the device the standby temperature setpoint has to be < Comfort temperature setpoint.	
Economy temp. setpoint [°C]		**16** [range 10 … 50]
	For a correct operation of the device the economy temperature setpoint has to be < Standby temperature setpoint.	
Standby temp. offset [0,1 K]	Setpoint type = relative	**-30** [range -10 … -50]
Economy temp. offset [0,1 K]	Setpoint type = relative	**-50** [range -10 … -50]
	For proper operation, Standby temperature offset has to be < Comfort temperature offset.	
Building protection temp. setpoint [°C]		**7** [range 2 … 10]
Heating type		**radiators** electric fan-coils floor radiant panels ceiling radiant panels
	It defines the terminal used for the thermal exchange in the room. The choice affects the parameters of 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 the range 0,2 K … 3 K]

Hysteresis position	Heating type = floor radiant panels, 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 the range 30 s … 120 min]
Min. change of value to send [%]	Control type = continuous	**10** [range 0 … 100]
PWM cycle time	Control type = PWM	**15 min** [range 5 … 240 min]
Proportional band [0,1 K]	Control type = continuous or PWM	***** [range 0 … 255]
	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	**240** [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): • *radiators: 150 min* • *electric: 100 min* • *fan-coils: 90 min* • *floor radiant panels: 240 min* • *ceiling radiant panels: 180 min*	
Min control value [%]	Control type = continuous or PWM	**15** [range 0 … 30]
Max control value [%]	Control type = continuous or PWM	**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 per le zone di normale occupazione;* • *T(sup) max ≤ 35°C per le zone periferiche degli ambienti.* *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	**separated** unique
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.	
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 matching a system with high inertia as the floor radiant panels (hydronic version) to a system with low inertia as the fan-coils.	

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Comfort setpoint (heating)	Setpoint type = absolute or relative	2 Byte	CRWTU	[9.001] temperature (°C)	186, 254, 322, 390
Standby setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU	[9.001] temperature (°C)	188, 256, 324, 392
Standby offset (heating)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	188, 256, 324, 392
Economy setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU	[9.001] temperature (°C)	190, 258, 326,

					394
Economy offset (heating)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	190, 258, 326, 394
Building protection setpoint (heating)		2 Byte	CRWTU	[9.001] temperature (°C)	192, 260, 328, 396
Heating out command	Control type = 2 points hysteresis or PWM, Command communication object = separated	1 Bit	CR-T-	[1.001] switch	195, 263, 331, 399
Heating out command	Control type = continuous, Command communication object = separated	1 Byte	CR-T-	[5.001] percentage (0..100%)	195, 263, 331, 399
Heating and cooling out command	Control type = 2 points hysteresis or PWM, Command communication object = unique	1 Bit	CR-T-	[1.001] switch	195, 263, 331, 399
Heating and cooling out command	Control type = continuous, Command communication object = unique	1 Byte	CR-T-	[5.001] percentage (0..100%)	195, 263, 331, 399
Auxiliary heating output command	Auxiliary heating = enabled, Command communication object = separated	1 Bit	CR-T-	[1.001] switch	197, 265, 333,

					401
Auxiliary heating and cooling output command	Auxiliary heating = enabled, Command communication object = unique	1 Bit	CR-T-	[1.001] switch	197, 265, 333, 401
Auxiliary heating disable	Auxiliary heating = enabled, Disabled from bus = yes	1 Bit	C-W--	[1.003] enable	199, 267, 335, 403

7.3.5 Cooling

The ***Cooling*** folder includes the following parameters:

- Temperature setpoint [°C]
- Comfort temperature setpoint [°C]
- Standby temperature setpoint [°C]
- Economy temperature setpoint [°C]
- 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 [%]
- 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

7.3.5.1 Parameter and communication object tables

Conditions: *General* → Thermostat function = cooling or both heating and cooling.

Parameter name	***Conditions***	***Values***
Temperature setpoint [°C]	Setpoint type = single	**23** [range 10 … 50]
Comfort temp. setpoint [°C]	Setpoint type = absolute or relative	**23** [range 10 … 50]
Standby temp. setpoint [°C]	Setpoint type = absolute	**26** [range 10 … 50]
	For a correct operation of the device the standby temperature setpoint has to be > comfort temperature setpoint.	
Economy temp. setpoint [°C]	Setpoint type = absolute	**28** [range 10 … 50]
	For a correct operation of the device the standby temperature setpoint has to be > economy temperature setpoint.	
Standby temperature offset [0,1 K]	Setpoint type = relative	**30** [range 10 … 50]
Economy temperature offset [0,1 K]	Setpoint type = relative	**50** [range 10 … 80]
	For a correct operation of the device the economy temperature offset has to be > standby temperature offset.	
Building protection temp. setpoint [°C]		**36** [range 30 … 50]
Neutral zone [0,1 K]	Settings ⇒ Thermostat function = both heating and cooling Settings ⇒ Command Communication Object = separated Settings ⇒ Heating–cooling changeover =	**20** [range 10 … 80]

	automatic Setpoint type = absolute or relative	
	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		**fan-coils** 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 choice done for Heating.	
Control type		**2 points 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]
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.	
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.

Parameter name	***Conditions***	***Values***
Hysteresis	Control type = 2 point hysteresis	**0,3 K** [other values in the range 0,2 K … 3 K]
Hysteresis position	Cooling type = floor radiant panels, 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 or continuous	**no sending** [other values in the range 30 s … 120 min]
Min. change of value to send [%]	Control type = continuous	**10** [range 0 … 100]
PWM cycle time	Control type = PWM (puls width modulation)	**15 min** [range 5 … 240 min]
Proportional band [0,1 K]	Control type = continuous or PWM	***** [range 0 … 255]
	The value is in tenths of Kelvin (K) degree. **) The field contains a preset value that depend on the selected cooling type (the value can be modified):* • *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	***** [range 0 … 255 min]

	**) The field contains a preset value that depend on the selected cooling type (the value can be modified): fan-coils: 90 min* • *floor radiant panels: 240 min* • *ceiling radiant panels: 180 min*	
Min control value [%]	Control type = continuous or PWM	**15** [range 0 … 30]
Max control value [%]	Control type = continuous or PWM	**85** [range 70 … 100]
Anticondensation with probe	Cooling type = floor radiant panels or ceiling radiant panels, External sensors (from bus) Anticondensation = enabled	**disabled** / enabled

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Comfort setpoint (cooling)		2 Byte	CRWTU	[9.001] temperature (°C)	187, 255, 323, 391
Standby setpoint (cooling)		2 Byte	CRWTU	[9.001] temperature (°C)	189, 257, 325, 393
Standby offset (cooling)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	189, 257, 325, 393
Economy setpoint (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.001] temperature (°C)	191, 259, 327, 395

Economy offset (cooling)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	191, 259, 327, 395
Building protection setpoint (cooling)		2 Byte	CRWTU-	[9.001] temperature (°C)	193, 261, 329, 397
Cooling out command	Command Communication Object = separated Control type = 2 point hysteresis or PWM	1 Bit	CR-T-	[1.001] switch	196, 264, 332, 400
Cooling out command	Command Communication Object = separated Control type = continuous	1 Byte	CR-T--	[5.001] percentage (0..100%)	196, 264, 332, 400
Auxiliary cooling output command	Auxiliary cooling = enabled	1 Bit	CR-T--	[1.001] switch	198, 266, 334, 402
Auxiliary cooling enable	Auxiliary cooling = enabled, Disabled from bus = yes	1 Bit	C-W--	[1.003] enable	200, 268, 336, 404
Anticondensation alarm	Anticondensation with probe or Active anticondensation = enabled	1 Bit	CR-T-	[1.005] alarm	222, 290, 358, 426

About anticondensation protection function

The objective of this function is to prevent the condensation on the thermal exchange surfaces of the installation or building when cooling is working. This function is mainly used in systems with thermal exchange

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

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

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

Nr.	Type	Denomination	Description
1a	Passive	Anticondensation protection by probe (via bus)	The thermostat receives the information about condensation via bus from a different KNX device through communication objects 175, 243, 311, 379: 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 dewpoint temperature (calculated by the thermostat)	Software protection that intervenes by closing down the room cooling circuit when the flow temperature defined in the hydronic project (as set in the corresponding ETS parameter) is lower than dew-point temperature calculated by the room thermostat using temperature and relative humidity values. The communication objects involved are 196, 264, 332, 400: 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 dewpoint temperature (calculated by the thermostat)	Software protection that intervenes by closing down the room cooling circuit when the actual measured flow temperature and received via bus from a different KNX device is lower than dew-point temperature calculated by the room thermostat using temperature and relative humidity values. The communication objects involved are 169, 237, 305, 373 at input: Flow temperature (from bus) [DPT 9.001 temperature °C] and 196, 264, 332, 400: Cooling out command [DPT 1.001 switch].
3	Active	Anticondensation protection with dew-point temperature sending over the bus and adjustment of the flow temperature	Software protection that foresees the sending on the bus of the dew-point temperature calculated by the room thermostat using temperature and relative humidity values to a KNX device capable of controlling the mixing condition of the conveying fluid for the cooling circuit. The regulation is performed by the KNX device receiving the dew-point temperature sent by the thermostat. The communication objects involved are 213, 281, 349, 417: Dew-point temperature [DPT 9.001 temperature °C].

Table 1 - 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 of Table 1).

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

7.3.6 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]
- Proportional band [0,1 K]
- Minimum change of value to send [%]
- Hot start
- Min. temp.to start ventilation [°C]
- Antistratification function
- Antistratification temp. differential
- Hysteresis
- Disable ventilation from bus
- Signal from bus
- Fan start delay
- Fan stop delay
- Cyclic sending interval

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.

7.3.6.1 Parameter and communication object tables

Parameter name	***Conditions***	***Values***
Control type		**1 speed** 2 speeds 3 speeds continuous regulation
Threshold first speed [0,1 K]	Control type ≥ 1 speed	**0** [range 0 … 255]
	The value is represented in tenths of Kelvin degrees. If the parameter Thermostat function = both heating and cooling, the threshold value is valid for both seasonal modes.	

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

Parameter name	*Conditions*	*Values*
Signal from bus	Disable ventilation from bus = yes	**not inverted** inverted
Fan start delay		**0 s** [other values in the range 10 s … 12 min]
	This parameter is also available if the hot-start function is active (through measuring of the conveying fluid temperature at the battery for the thermal exchange). The function is active in both seasonal modes (heating and cooling).	
Fan stop delay		**0 s** [other values in the range 10 s … 12 min]
	The function allows prolonging the operation of the ventilator, dissipating in the room the residual heat or cool present in battery for the thermal exchange. The function is active in both seasonal modes (heating and cooling).	
Cyclic sending interval		**no sending** [other values in the range 30 s … 120 min]

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Fan continuous speed	Control type = continuous regulation	1 Byte	CR-T-	[5.001] percentage (0..100%)	201, 269, 337, 405
Fan speed 1	Control type ≥ 1 speed	1 Bit	CR-T--	[1.001] switch	202, 270, 338, 406
Fan speed 2	Control type ≥ 2 speed	1 Bit	CR-T--	[1.001] switch	203, 271, 339, 407
Fan speed 3	Control type = 3 speeds	1 Bit	CR-T-	[1.001] switch	204, 272, 340,408

Fan control disable	Disable ventilation from bus = yes	1 Bit	C-W--	[1.002] boolean	205, 273, 341, 409
Fan manual speed		1 Byte	CRWTU	[5.010] counter pulses (0…255)	208, 276, 344, 412
Fan speed status		1 Byte	CR-T--	[5.010] counter pulses (0…255)	209, 277, 345, 413
Fan manual active status		1 Bit	CRWTU	[1.011] state	210, 278, 346, 414
Fan manual speed percentage		1 Byte	CR-T--	[5.001] percentage	225, 293, 361, 429
Fan manual speed off status		1 Bit	CR-T--	[1.011] state	226, 294, 362, 430

7.3.6.2 Delayed fan start ("hot-start")

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

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

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

equipped with a sensor of minimum water temperature that acquires the temperature of the heat conveying fluid.

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

7.3.6.3 Antistratification function

This function is used in the case of heating systems with thermal exchange of convective type for rooms with height and volume much higher than usual (atriums, fitness facility, commercial buildings, etc.). Because of the natural convection - with warm air rising to the highest altitudes of the room - the phenomenon of air stratification occurs, with energy waste and discomfort for the occupants at the same time. The function opposes to the air stratification, forcing the warm air downwards.

The antistratification function requires:

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

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

7.3.6.4 2-stage configuration with fan-coils as auxiliary stage

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

The ***Ventilation*** folder, that is unique, configures a main or a auxiliary stage depending on the settings 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:

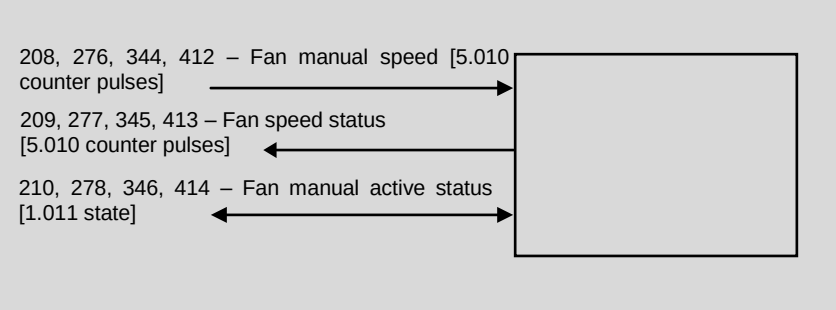
1) Settings ⇒ Thermostat function = both heating and cooling. This configuration enables both folders (heating and cooling)
2) Heating ⇒ Heating type = floor radiant panels or ceiling radiant panels
3) Heating ⇒ Command communication object = separated (if unique is choosen, the parameter Cooling ⇒ Cooling 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

i	**Important!** If the fan-coil system has a 2-pipe hydraulic configuration, the objects Auxiliary heating output command (1 bit) and Cooling out command (1 byte) 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.

7.3.6.5 Remote fan speed modification

The communication objects shown in figure allow to monitor the actual fan speed forced automatically (A) by the temperature controller or set locally by the user when interacting with the LCD display and the touch buttons of the room thermostat. The communication objects (from now on: C.O.) also allow to perform the same modifications remotely, for example from a supervisor software.



The C.O.s *209, 277, 345, 413 – Fan speed status* allows to evaluate the actual fan speed; the C.O.s *210, 278, 346, 414 – Fan manual active status* contains the information about automatic (=0, not active) or manual (=1, active) operating mode. By modifying the C.O.s *208, 276, 344, 412– 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.s *210, 278, 346, 414 – Fan manual active status* (=0, not active).

Accepted values for C.O.s mentioned above depend on the number of speeds set in ETS.

If *Control Type* parameter in Ventilation folder is = 1, 2 or 3 speeds, C.O.s with DPT [5.010 counter] 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] match the following percentage of the maximum speed:

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

7.3.7 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.

NOTE: if the sensor is not able to read the RU value from bus, the value 7F FF is sent, until the real value is available.

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

7.3.7.1 Parameter and communication object tables

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

Relative humidity setpoint for dehumidification		2 Byte	CRWTU	[9.007] humidity (%)	214, 282, 350, 418
Dehumidification command		1 Bit	CR-T-	[1.001] switch	216, 284, 352, 420
Dehumidification command - water battery	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-	[1.001] switch	217, 285, 353, 421
Dehumidification integration control	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ cooling type = floor radiant panels or ceiling radiant panels, Relative humidity control ⇒ dehumidification ⇒ dehumidification function = cooling only Integration = yes	1 Bit	CR-T-	[1.001] switch	218, 286, 354, 422
	This object switches ON if (simultaneously) the relative humidity is greater than the relative humidity setpoint and the room temperature is greater than the setpoint of the parameter Temperature difference for integration.				
Dehumidification control disable	Disable dehumidification control from bus = yes	1 Bit	C-W--	[1.002] boolean	219, 287, 355, 423

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

7.3.7.2 Parameter and communication object tables

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

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Relative humidity setpoint for humidification		2 Byte	CRWTU	[9.007] humidity (%)	215, 283, 351, 419
Humidification command		1 Bit	CR-T-	[1.001] switch	220, 288, 356,

					424
Humidification control disable	Disable humidification control from bus = yes	1 Bit	C-W--	[1.002] boolean	221, 289, 357, 425

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]

7.3.7.3 Parameter and communication object tables

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

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Dew-point temperature	Dew-point temperature = enabled	2 bytes	CR-T-	[9.001] temperature °C	213, 281, 349, 417

7.3.8 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.*

Window contacts

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

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

The ***Window contacts*** folder includes the following parameters:

- Window contacts function
- Wait time to building protection mode

7.3.8.1 Parameter and communication object tables

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

Nome oggetto	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Windows contact sensor 1 (from bus)	Window contacts function = enabled, Window contact 1 = enabled	1 Bit	C-W--	[1.019] window/door	170, 238, 306, 374
Windows contact sensor 2 (from bus)	Window contacts function = enabled, Window contact 2 = enabled	1 Bit	C-W--	[1.019] window/door	171, 239, 307, 375

Presence sensors

The ***Presence sensors*** folder includes the following parameters:

- Presence sensor 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* or *relative*

7.3.8.2 Parameter and communication object tables

Parameter name	*Conditions*	*Values*
Presence sensors function		**disabled** / enabled
	Parameter that enables the presence sensor function.	
Presence sensors use	Presence sensor function = enabled	**comfort extension** comfort limitation comfort extension and comfort limitation

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

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Presence sensor 1 (from bus)	Presence sensor function= enabled	1 Bit	C-W---	[1.018] occupancy	172, 240, 308, 376
Presence sensor 2 (from bus)	Presence sensor function = enabled	1 Bit	C-W---	[1.018] occupancy	173, 241, 309, 377

Card holder

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

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

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

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

7.3.8.3 Parameter and communication object tables

Parameter name	***Conditions***	***Values***
Card holder function		**disabled** / enabled
	Parameter that enables the card holder function.	
On card insertion switch HVAC mode to	Card holder function = enabled	none **comfort** standby economy
	This parameter defines to which operating mode the device should automatically switch inserting the card into the holder.	
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 of the operating mode, 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 removing the card from the holder.	
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 of the operating mode, removing the card from the holder.	

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm. Obj. No.***
Card holder contact sensor (from bus)	Card holder function = enabled	1 Bit	CR-T--	[1.018] occupancy	174, 242, 310, 378

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).

7.4 Logic functions

7.4.1 General information

The KNX device 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:

- 4 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.

7.4.1.1 Parameter and communication object tables

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

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

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

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm. Obj. No.*
Logic function X – Input 1	Logic function X = enabled Logic object 1 = enabled	1 Bit	C-WTU	[1.001] switch	139, 144, 149, 154
	x = 1, 2, 3, 4				
Logic function X – Input 2	Logic function X = enabled Logic object 2 = enabled	1 Bit	C-WTU	[1.001] switch	140, 145, 150, 155
	x = 1, 2, 3, 4				
Logic function X – Input 3	Logic function X = enabled Logic object 3 = enabled	1 Bit	C-WTU	[1.001] switch	141, 146, 151, 156
	x = 1, 2, 3, 4				
Logic function X – Input 4	Logic function X = enabled Logic object 4 = enabled	1 Bit	C-WTU	[1.001] switch	142, 147, 152, 157
	x = 1, 2, 3, 4				
Logic function X – Output	Logic function X = enabled At least one Logic object = enabled	1 Bit	CR-T-	[1.001] switch	143, 148, 153, 158
	x = 1, 2, 3, 4				

8. Appendix

8.1 Communication objects table

Following is a summary of all KNX Communication Objects (CO) and corresponding Data Point Types (DPT) defined by the application program according to configuration options. The listing order is generally by CO number (in case of COs repeated by channel, the CO number for the first channel is taken as relevant).

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Disable front pushbuttons	Manual operations = enabled Disable from bus = yes	1 bit	C−W−−	[1.001] switch	1
Input *xx* – **Lock command** (xx = 1,…,8)	Channel *x* = Independent, Input xx Lock function = enabled	1 bit	C−W−U	[1.003] enable	3, 20, 37, 54, 71, 88, 105, 122
	Channel *x* = coupled Lock function = enabled				
Input *xx* – **Switching status *[type]* Object *n**** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = send values or sequences**	***See*** Table 2	C−WTU	***See table A1***	4...11, 21…28, 38…45, 55...62, 72…79, 89…96, 106…113, 123…130
	** The numbers of listed C.O.s are referred to the first of 8 objects (for each input); the next C.O.s are sequential. In order to obtain the C.O. number for the n-th object, simply add (n-1) to the referred numbers.* E.g.: he C.O.s linked to input 3A have numbers starting from 72. The number of the 5th C.O. linked to that input will be 72 + (5-1) = 76.				
Input *xx* – **Switching command** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = dimming**	1 bit	C−WTU	[1.001] switch	12, 29, 46, 63, 80, 97, 114, 131
Input *x and y* – **Switching command** (x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8)	Channel *x* = coupled, **Type = switching**	1 bit	C−WTU	[1.001] switch	12, 46, 80, 114
	Channel *x* = coupled **Type = dimming**				
Input *xx* – **Dedicated stop command** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = shutter or venetian blind** **Venetian blind mode = disabled**	1 bit	C−−T−	[1.017] trigger	12, 29, 46, 63, 80, 97, 114, 131f
Input *x and y* – **Dedicated stop command** (x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8)	Channel *x* = coupled, **Type = shutter or venetian blind** **Venetian blind mode = disabled**	1 bit	C−−T−	[1.017] trigger	12, 46, 80, 114

Input *xx* / **Dimming up / down / stop command** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = dimming**	4 bit	C--T-	[3.*] 3-bit control	13, 30, 47, 64, 81, 98, 115, 132
Input *x and y* – **Dimming up / down / stop command** (x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8)	Channel *x* = coupled, **Type = dimming**	4 bit	C--T-	[3.*] 3-bit control	13, 47, 81, 115
Input *xx* - **Stop – step up/down command** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = shutter or venetian blind** **Venetian blind mode = enabled**	1 bit	C--T-	[1.007] step	15, 32, 49, 66, 83, 100, 117, 134
Input *x and y* – **Stop – step up/down command** (x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8)	Channel *x* = coupled, **Type = shutter or venetian blind** **Venetian blind mode = enabled**	1 bit	C--T-	[1.007] step	15, 49, 83, 117
Input *xx* - **Move up / down command** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = shutter or venetian blind** **Venetian blind mode = enabled**	1 bit	C--T-	[1.008] up/down	16, 33, 50, 67, 84, 101, 118, 135
Input *x and y* – **Move up / down command** (x and y = 1 and 2, 3 and 4, 5 and 6, 7 and 8)	Channel *x* = coupled, **Type = shutter or venetian blind**	1 bit	C--T-	[1.008] up/down	16, 50, 84, 118
Input *xx* – **Scene number**	Channel *x* = Independent, Input xx **Type = scene**	1 Byte	C--T-	[17.*] Scene number [18.*]	17, 34, 51, 68, 85, 102,

(xx = 1,…,8)				Scene control	119, 136
Input *xx* – **Counter value [1/2/4] byte** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = counter**	1 Byte 2 Bytes 4 Bytes	C-R-T-	[5.010] Counter pulses [7.001] Pulses [12.001] Counter pulses (unsigned)	17, 34, 51, 68, 85, 102, 119, 136
Input *xx* – **Counter reset command** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = counter**	1 bit	C-W--	[1.015] reset	18, 35, 52, 69, 86, 103, 120, 137
Input *xx* – **Counter runout** (xx = 1,…,8)	Channel *x* = Independent, Input xx **Type = counter**	1 bit	CRWT-	[1.055] alarm	19, 36, 53, 70, 87, 104, 121, 138
Logic function x – Input 1 (x = 1,…,4)	Logic function x = enabled Logic object 1 = enabled	1 bit	C-WTU	[1.1] DPT_Switch	139, 144, 149, 154
Logic function x – Input 2 (x = 1,…,4)	Logic function x = enabled Logic object 2 = enabled	1 bit	C-WTU	[1.1] DPT_Switch	140, 145, 150, 155
Logic function x – Input 3 (x = 1,…,4)	Logic function x = enabled Logic object 3 = enabled	1 bit	C-WTU	[1.1] DPT_Switch	141, 146, 151, 156
Logic function x – Input 4 (x = 1,…,4)	Logic function x = enabled Logic object 24= enabled	1 bit	C-WTU	[1.1] DPT_Switch	142, 147, 152, 157
Logic function x – Output (x = 1,…,4)	Logic function x = enabled	1 bit	C-R-T-	[1.1] DPT_Switch	143, 148, 153, 158
Temperature value (x=2, 4, 6, 8)	**Input x = enabled NTC**	2 bytes	C-R-T-	[9.001] Temperature	159, 227, 295, 363
Temperature threshold 1 - Switch	**Threshold 1 ≠ not active**	1 Bit	C-R-T-	[1.001] switch	160, 228, 296, 364
Temperature threshold 2 - Switch	**Threshold 2 ≠ not active**	1 Bit	C-R-T-	[1.001] switch	161, 229, 297, 365

Thermostat x – Room temperature (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – room temperature = enabled**	2 bytes	C-WTU	[9.001] Temperature	162, 230, 298, 366
Thermostat x - Humidity (2 byte, from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – relative humidity = enabled Humidity CO dimension = 2 bytes**	2 bytes	C-WTU	[9.007] Humidity (%)	163, 231, 299, 367
Thermostat x - Humidity (1 bytes, from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – relative humidity = enabled Humidity CO dimension = 1 bytes**	1 byte	C-WTU	[5.001] percentage (0..100%)	164, 232, 300, 368
Thermostat x – antistratification temperature (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – Antistratification temperature = enabled**	2 bytes	C-WTU	[9.001] Temperature	165, 233, 301, 369
Thermostat x – outdoor temperature (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – outdoor temperature = enabled**	2 bytes	C-WTU	[9.001] Temperature	166, 234, 302, 370
Thermostat x – coil temperature (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – coil temperature = enabled**	2 bytes	C-WTU	[9.001] Temperature	167, 235, 303, 371
Thermostat x – floor surface temperature (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – floor surface temperature = enabled**	2 bytes	C-WTU	[9.001] Temperature	168, 236, 304, 372
Thermostat x – flow temperature (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – flow temperature = enabled**	2 bytes	C-WTU	[9.001] Temperature	169, 237, 305, 373

Thermostat x – window contact sensor 1 (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – window contact 1 = enabled**	1 bit	C-WTU	[1.019] window/door	170, 238, 306, 374
Thermostat x – window contact sensor 2 (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – window contact 2 = enabled**	1 bit	C-WTU	[1.019] window/door	171, 239, 307, 375
Thermostat x – presence sensor 1 (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – presence sensor 1 = enabled**	1 bit	C-WTU	[1.018] occupancy	172, 240, 308, 376
Thermostat x – presence sensor 2 (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – presence sensor 2 = enabled**	1 bit	C-WTU	[1.018] occupancy	173, 241, 309, 377
Thermostat x – contact of card holder (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – card holder contact = enabled**	1 bit	C-WTU	[1.018] occupancy	174, 242, 310, 378
Thermostat x – anticondensation (from bus) (x=2, 4, 6, 8)	**External sensors (from bus) – anticondensation = enabled**	1 bit	C-WTU	[1.001] switch	175, 243, 311, 379
Thermostat x – weighted temperature (x=2, 4, 6, 8)	**External sensors (from bus) – room temperature = enabled**	2 bytes	CR-T-	[9.001] temperature	176, 244, 312, 380
Thermostat x – Heating/cooling status out (x=2, 4, 6, 8)	**Thermostat function = both heating and cooling Heating/cooling changeover = from bus**	1 bit	CR-T-	[1.100] heating/cooling	177, 245, 313, 381

Thermostat x – Heating/cooling status in (x=2, 4, 6, 8)	**Thermostat function = both heating and cooling** **Heating/cooling changeover = from bus**	1 bit	C-W--	[1.100] heating/cooling	178, 246, 314, 382
Thermostat x – HVAC mode in (x=2, 4, 6, 8)	**Setpoint type = absolute or relative**	1 byte	C-W--	[20.102] HVAC mode	179, 247, 315, 383
Thermostat x – HVAC forced mode in (x=2, 4, 6, 8)	**Thermostat function = both heating and cooling** **Setpoint type = absolute or relative**	1 byte	C-W--	[20.102] HVAC mode	180, 248, 316, 384
Thermostat x – HVAC mode out (x=2, 4, 6, 8)	**Setpoint type = absolute or relative**	1 byte	CR-T-	[20.102] HVAC mode	181, 249, 317, 385
Thermostat x – HVAC manual mode (x=2, 4, 6, 8)	**Setpoint type = absolute or relative**	1 byte	C-WTU	[20.102] HVAC mode	182, 250, 318, 386
Thermostat x – Chrono active status (x=2, 4, 6, 8)	**Setpoint type = absolute or relative**	1 bit	CR-T-	[1.011] state	183, 251, 319, 387
Thermostat x – Actual setpoint (x=2, 4, 6, 8)		2 bytes	CR-T-	[9.001] temperature	184, 252, 320, 388
Thermostat x – Manual setpoint (x=2, 4, 6, 8)	**Setpoint type = absolute or relative**	2 bytes	C-W--	[9.001] temperature	185, 253, 321, 389
Thermostat x – Input setpoint (x=2, 4, 6, 8)	**Setpoint type = single**	2 bytes	CRWTU	[9.001] temperature	186, 254, 322, 390
Thermostat x – Comfort setpoint (heating)	**Setpoint type = absolute or relative**	2 bytes	CRWTU	[9.001] temperature	186, 254, 322, 390

(x=2, 4, 6, 8)					
Thermostat x – Comfort setpoint (cooling) (x=2, 4, 6, 8)	**Setpoint type = absolute or relative**	2 bytes	CRWTU	[9.001] temperature	187, 255, 323, 391
Thermostat x – Standby setpoint (heating) (x=2, 4, 6, 8)	**Setpoint type = absolute**	2 bytes	CRWTU	[9.001] temperature	188, 256, 324, 392
Thermostat x – Offset standby (heating) (x=2, 4, 6, 8)	**Setpoint type = relative**	2 bytes	CRWTU	[9.002] temperature difference (K)	188, 256, 324, 392
Thermostat x – Standby setpoint (cooling) (x=2, 4, 6, 8)	**Setpoint type = absolute**	2 bytes	CRWTU	[9.001] temperature	189, 257, 325, 393
Thermostat x – Offset standby (cooling) (x=2, 4, 6, 8)	**Setpoint type = relative**	2 bytes	CRWTU	[9.002] temperature difference (K)	189, 257, 325, 393
Thermostat x – Economy setpoint (heating) (x=2, 4, 6, 8)	**Setpoint type = absolute**	2 bytes	CRWTU	[9.001] temperature	190, 258, 326, 394
Thermostat x – Offset economy (heating) (x=2, 4, 6, 8)	**Setpoint type = relative**	2 bytes	CRWTU	[9.002] temperature difference (K)	190, 258, 326, 394
Thermostat x – Economy setpoint (cooling) (x=2, 4, 6, 8)	**Setpoint type = absolute**	2 bytes	CRWTU	[9.001] temperature	191, 259, 327, 395

Thermostat x – Offset economy (cooling) (x=2, 4, 6, 8)	**Setpoint type = relative**	2 bytes	CRWTU	[9.002] temperature difference (K)	191, 259, 327, 395
Thermostat x – Building protection setpoint (heating) (x=2, 4, 6, 8)		2 bytes	CRWTU	[9.001] temperature	192, 260, 328, 396
Thermostat x – Building protection setpoint (cooling) (x=2, 4, 6, 8)		2 bytes	CRWTU	[9.001] temperature	193, 261, 329, 397
Thermostat x – Room temperature controller status (x=2, 4, 6, 8)		1 bit	CR-T-	[1.003] enable	194, 262, 330, 398
Thermostat x – Heating out command (x=2, 4, 6, 8)	Control type = 2 points hysteresis or PWM Command Communication Object = separated	1 bit	CR-T-	[1.001] switch	195, 263, 331, 399
Thermostat x – Heating out command (x=2, 4, 6, 8)	Control type = continuous Command Communication Object = separated	1 byte	CR-T-	[5.001] percentage (0..100%)	195, 263, 331, 399
Thermostat x – Heating out command (x=2, 4, 6, 8)	Control type = 2 points hysteresis or PWM Command Communication Object = unique	1 bit	CR-T-	[1.001] switch	195, 263, 331, 399
Thermostat x – Heating and cooling out command (x=2, 4, 6, 8)	Control type = continuous Command Communication Object = unique	1 byte	CR-T-	[5.001] percentage (0..100%)	195, 263, 331, 399

Thermostat x – Cooling out command	Control type = 2 points hysteresis or PWM Command Communication Object = separated	1 bit	CR-T-	[1.001] switch	196, 264, 332, 400
Thermostat x – Cooling out command (x=2, 4, 6, 8)	Control type = continuous Command Communication Object = separated	1 byte	CR-T-	[5.001] percentage (0..100%)	196, 264, 332, 400
Thermostat x – Auxiliary heating output command (x=2, 4, 6, 8)	**Auxiliary heating = enabled** Communication Object = separated	1 bit	CR-T-	[1.001] switch	197, 265, 333, 401
Thermostat x – Auxiliary heating and cooling output command (x=2, 4, 6, 8)	**Auxiliary heating = enabled** Communication Object = unique	1 bit	CR-T-	[1.001] switch	197, 265, 333, 401
Thermostat x – Auxiliary cooling output command (x=2, 4, 6, 8)	**Auxiliary cooling = enabled**	1 bit	CR-T-	[1.001] switch	198, 266, 334, 402
Thermostat x – Auxiliary heating disable (x=2, 4, 6, 8)	**Auxiliary heating = enabled** **Disabled from bus = yes**	1 bit	C-W--	[1.003] enable	199, 267, 335, 403
Thermostat x – Auxiliary cooling disable (x=2, 4, 6, 8)	**Auxiliary cooling = enabled** **Disabled from bus = yes**	1 bit	C-W--	[1.003] enable	200, 268, 336, 404
Thermostat x – Fan continuous speed (x=2, 4, 6, 8)	**Ventilation function = enabled** **Control type = continuous regulation**	1 byte	CR-T-	[5.001] percentage (0..100%)	201, 269, 337, 405
Thermostat x – Fan speed 1 (x=2, 4, 6, 8)	**Ventilation function = enabled** **Control type ≥ 1 speed**	1 bit	CR-T-	[1.001] switch	202, 270, 338, 406

Thermostat x – Fan speed 2 (x=2, 4, 6, 8)	**Ventilation function = enable** **Control type ≥ 2 speeds**	1 bit	CR-T-	[1.001] switch	203, 271, 339, 407
Thermostat x – Fan speed 3 (x=2, 4, 6, 8)	**Ventilation function = enabled** **Control type ≥ 3 speeds**	1 bit	CR-T-	[1.001] switch	204, 272, 340, 408
Thermostat x – Fan control disable (x=2, 4, 6, 8)	**Ventilation function = enabled** **Disable ventilation from bus = yes**	1 bit	C-W--	[1.002] boolean	205, 273, 341, 409
Thermostat x – Thermostat – Alarm text (x=2, 4, 6, 8)	**Ventilation function = enabled**	14 bytes	CR-T-	[16.000] Character string (ASCII)	206, 274, 342, 410
Thermostat x – Manual setpoint active status (x=2, 4, 6, 8)	**Setpoint type = absolute or relative**	1 bit	CRWTU	[1.011] state	207, 275, 343, 411
Thermostat x – Fan manual speed (x=2, 4, 6, 8)	**Ventilation function = enabled**	1 byte	CRWTU	[5.010] counter pulses (0…255)	208, 276, 344, 412
Thermostat x – Fan speed (x=2, 4, 6, 8)	**Ventilation function = enabled**	1 byte	CR-T-	[5.010] counter pulses (0…255)	209, 277, 345, 413
Thermostat x – Fan manual active status (x=2, 4, 6, 8)	**Ventilation function = enabled**	1 bit	CRWTU	[1.011] state	210, 278, 346, 414
Thermostat x – Room temperature control alarm (x=2, 4, 6, 8)	**Thermostat function = enabled**	1 bit	CR-T-	[1.005] alarm	211, 279, 347, 415

Thermostat x – Disable room temperature controller (x=2, 4, 6, 8)	**Disable temperature controller from bus = yes**	1 bit	C-W--	[1.001] switch	212, 280, 348, 416
Thermostat x – Dew-point temperature (x=2, 4, 6, 8)	**External sensors (from bus) – relative humidity = enabled Dew-point temperature = enabled**	2 bytes	CR-T-	[9.001] temperature	213, 281, 349, 417
Thermostat x – Relative humidity setpoint for dehumidification (x=2, 4, 6, 8)	**Dehumidification function ≠ disabled**	2 bytes	CRWTU	[9.007] humidity (%)	214, 282, 350, 418
Thermostat x – Relative humidity setpoint for humidification (x=2, 4, 6, 8)	**Humidification function ≠ disabled**	2 bytes	CRWTU	[9.007] humidity (%)	215, 283, 351, 419
Thermostat x – Dehumidification command (x=2, 4, 6, 8)	**Dehumidification function ≠ disabled**	1 bit	CR-T-	[1.001] switch	216, 284, 352, 420
Thermostat x – Dehumidification command - water battery (x=2, 4, 6, 8)	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ cooling type = floor radiant panels or ceiling radiant panels	1 bit	CR-T-	[1.001] switch	217, 285, 353, 421
Thermostat x – Dehumidification integration control (x=2, 4, 6, 8)	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ cooling type = floor radiant panels or ceiling radiant panels Integration = yes	1 bit	CR-T-	[1.001] switch	218, 286, 354, 422
Thermostat x – Dehumidification control disable	**Disable dehumidification control from bus = yes**	1 bit	C-W--	[1.002] boolean	219, 287, 355, 423

(x=2, 4, 6, 8)					
Thermostat x – Humidification command (x=2, 4, 6, 8)	**Humidification function ≠ disabled**	1 bit	CR-T-	[1.001] switch	220, 288, 356, 424
Thermostat x – Humidification control disable (x=2, 4, 6, 8)	**Disable humidification control from bus = yes**	1 bit	C-W--	[1.002] boolean	221, 289, 357, 425
Thermostat x – Anticondensation alarm (x=2, 4, 6, 8)	**Anticondensation with probe = enabled,** or **Active anticondensation = enabled**	1 bit	CR-T-	[1.005] alarm	222, 290, 358, 426
Thermostat x – Thermal generator lock (x=2, 4, 6, 8)		1 bit	C-W--	[1.005] alarm	223, 291, 359, 427
Thermostat x – Building protection HVAC mode active (x=2, 4, 6, 8)		1 bit	CR-T-	[1.011] state	224, 292, 360, 428
Thermostat x – Fan manual speed percentage (x=2, 4, 6, 8)		1 byte	CR-T-	[5.001] percentage	225, 293, 361, 429
Thermostat x – Fan manual speed off status (x=2, 4, 6, 8)		1 bit	CR-T-	[1.011] state	226, 294, 362, 430

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

Table 2 - Independent/single channel object sizes and DPTs

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

8.3 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